JunosE™ Software for E Series™ Broadband Services Routers

Link Layer Configuration Guide

Release
14.3.x

Published: 2013-07-17

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

JunosE™ Software for E Series™ Broadband Services Routers Link Layer Configuration Guide
Release 14.3.x

Revision History
July 2013—FRS JunosE 14.3.x

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Abbreviated Table of Contents

About the Documentation xxxi

Part 1 Chapters

Chapter 1 Configuring ATM 3
Chapter 2 Configuring Frame Relay 103
Chapter 3 Configuring Multilink Frame Relay 129
Chapter 4 Configuring Upper-Layer Protocols over Static Ethernet Interfaces 149
Chapter 5 Configuring VLAN and S-VLAN Subinterfaces 165
Chapter 6 Monitoring VLAN and S-VLAN Subinterfaces 183
Chapter 7 Configuring 802.3ad Link Aggregation and Link Redundancy 193
Chapter 8 Configuring IEEE 802.3ah OAM Link-Fault Management 225
Chapter 9 Configuring Point-to-Point Protocol 261
Chapter 10 Configuring Multilink PPP 315
Chapter 11 Configuring Multiclass Multilink PPP 361
Chapter 12 Configuring Packet over SONET 375
Chapter 13 Configuring Point-to-Point Protocol over Ethernet 387
Chapter 14 Monitoring and Troubleshooting Point-to-Point Protocol over Ethernet 425
Chapter 15 Configuring Bridged IP 445
Chapter 16 Configuring Bridged Ethernet 451
Chapter 17 Configuring Transparent Bridging 473
Chapter 18 Configuring Cisco HDLC 509
Chapter 19 Configuring Upper-Layer Dynamic Interfaces 519
Chapter 20 Monitoring Upper-Layer Dynamic Interfaces 593
Chapter 21 Configuring Dynamic Interfaces Using Bulk Configuration 619
Chapter 22 Monitoring Dynamic Interfaces and Profiles 691

Part 2 Index

Index 725

Table of Contents

About the Documentation . . . xxxi

- E Series and JunosE Documentation and Release Notes . . . xxxi
- Audience . . . xxxi
- E Series and JunosE Text and Syntax Conventions . . . xxxi
- Obtaining Documentation . . . xxxiii
- Documentation Feedback . . . xxxiii
- Requesting Technical Support . . . xxxiii
  - Self-Help Online Tools and Resources . . . xxxiv
  - Opening a Case with JTAC . . . xxxiv

Part 1 Chapters

Chapter 1 Configuring ATM . . . 3

- Overview . . . 3
  - ATM Interfaces . . . 4
  - ATM Physical Connections . . . 4
  - ATM Virtual Connections . . . 5
    - Virtual Channel Connection . . . 5
    - Virtual Path Connection . . . 5
  - ATM SVCs . . . 5
  - ATM Adaptation Layer . . . 5
  - Local ATM Passthrough . . . 6
  - VCC Cell Relay Encapsulation . . . 6
  - Traffic Management . . . 7
    - Connection Admission Control . . . 7
  - ILMI . . . 9
  - VPI/VCI Address Ranges . . . 9
  - VP Tunneling . . . 9
- Platform Considerations . . . 10
  - Module Requirements . . . 10
  - Interface Specifiers . . . 10
- References . . . 11
- Supported Features . . . 11
  - Module Capabilities . . . 12
  - Virtual Channel Support . . . 13
- ATM NBMA . . . 13
  - ARP Table . . . 13
    - Static Map Versus Inverse ARP . . . 14
    - Aging . . . 14
    - Removing Circuits . . . 14

Operations, Administration, and Management of ATM Interfaces . . . 14
End-to-End and Segment Endpoints . . . 15
Fault Management . . . 15
How the ATM Interface Handles AIS Cells . . . 15
How the ATM Interface Handles RDI Cells . . . 16
Continuity Verification . . . 16
Activation and Deactivation Cells . . . 16
Activating CC Cell Flow . . . 16
Deactivating CC Cell Flow . . . 17
After CC Cell Flow Is Enabled . . . 17
Loopback . . . 17
VC Integrity . . . 17
F4 OAM Cells . . . 18
ATM Ping . . . 18
How the ATM Interface Handles Loopback Cells Received . . . 19
Automatic Disabling of F5 OAM Services . . . 19
Rate Limiting for F5 OAM Cells . . . 20
Before You Configure ATM . . . 20
Configuration Tasks . . . 21
Creating a Basic Configuration . . . 21
Setting Optional Parameters . . . 23
Optional Tasks on ATM 1483 Subinterfaces . . . 25
Configuring OAM . . . 31
Configuring F4 OAM . . . 32
Configuring F5 OAM . . . 33
Setting a Loopback Location ID . . . 35
Enabling OAM Flush . . . 35
Running ATM Ping . . . 36
Configuring an NBMA Interface . . . 37
Creating an NBMA Static Map . . . 38
Assigning Descriptions to Interfaces . . . 40
Sending Interface Descriptions to AAA . . . 41
Assigning Descriptions to Virtual Paths . . . 42
Exporting ATM 1483 Subinterface Descriptions . . . 42
Configuring Individual ATM PVC Parameters . . . 43
Benefits . . . 44
Creating Control PVCs . . . 44
Creating Data PVCs . . . 45
Configuring the Service Category for Data PVCs . . . 46
Configuring Encapsulation for Data PVCs . . . 48
Configuring F5 OAM for Data PVCs . . . 49
Configuring Inverse ARP for Data PVCs . . . 52
Configuring ATM VC Classes . . . 52
Benefits . . . 53
Precedence Levels . . . 54
Precedence Levels for Static PVCs . . . 54
Precedence Levels for Dynamic PVCs . . . 54
Precedence Level Examples . . . 55
Upgrade Considerations . . . 55

Configuring VC Classes 56
Assigning VC Classes to Individual PVCs 61
Assigning VC Classes to ATM Major Interfaces 62
Assigning VC Classes to Static ATM 1483 Subinterfaces 63
Assigning VC Classes to Base Profiles for Bulk-Configured VC Ranges 64
Precedence Level Examples for Assigning VC Classes 64
Example 1: Explicitly Changing the Service Category 64
Example 2: Changing the Encapsulation Method in the VC Class 65
Example 3: Effect of Using the atm pvc Command 65
Example 4: Overriding RADIUS Values 66
Configuring Dynamic ATM 1483 Subinterfaces 66
Monitoring ATM 66
Setting Statistics Baselines 67
Displaying Interface Rate Statistics for ATM VCs and ATM VPs 68
Using ATM show Commands 71

Chapter 2 Configuring Frame Relay 103

Overview 103
Framing 103
Error Frames 104
Unicast and Multicast Addressing 104
User-to-Network and Network-to-Network Interfaces 104
Platform Considerations 105
Module Requirements 105
Interface Specifiers 105
References 106
Before You Configure Frame Relay 106
Configuring Frame Relay 106
Configuring IPv6 over Frame Relay Interfaces 114
Configuring an IPv6 Address for a Frame Relay That Operates as a DCE, DTE, or NNI 114
Configuring an IPv6 Address for a Frame Relay PVC 115
End-to-End Fragmentation and Reassembly 116
Frame Fragmentation 116
Frame Reassembly 116
Map Class 117
Configuring End-to-End Fragmentation 117
Monitoring Frame Relay 120

Chapter 3 Configuring Multilink Frame Relay 129

Overview 129
T1/E1 Connections 129
MLFR Link Integrity Protocol 130
Interface Stacking 131
Platform Considerations 131
Module Requirements 132
Interface Specifiers 132
References 132
Supported MLFR Features 132
Unsupported MLFR Features 133

Before You Configure MLFR . . . 134
Configuration Tasks . . . 134
Configuration Example . . . 135
Configuring Frame Relay Versus MLFR . . . 135
Monitoring MLFR . . . 136

Chapter 4 Configuring Upper-Layer Protocols over Static Ethernet Interfaces . . . 149

Upper-Layer Protocols over Static Ethernet Overview . . . 149
Upper-Layer Protocols over Static Ethernet Platform Considerations . . . 150
Module Requirements . . . 151
Interface Specifiers . . . 151
Upper-Layer Protocols over Static Ethernet References . . . 151
Configuring IP over a Static Ethernet Interface . . . 151
Configuring PPPoE over a Static Ethernet Interface . . . 152
Configuring IP and MPLS over a Static Ethernet Interface . . . 153
Configuring IP, MPLS, and PPPoE over Ethernet . . . 153
L2TP and Ethernet . . . 154
Multinetting and Ethernet . . . 155
Monitoring Upper-Level Protocols over Ethernet . . . 155

Chapter 5 Configuring VLAN and S-VLAN Subinterfaces . . . 165

VLAN Overview . . . 165
S-VLAN Overview . . . 166
VLAN and S-VLAN Platform Considerations . . . 167
Module Requirements . . . 167
Interface Specifiers . . . 168
VLAN and S-VLAN References . . . 168
Configuring VLAN Subinterfaces . . . 168
Creating a VLAN Major Interface . . . 169
Configuring IP over VLAN . . . 169
Configuring PPPoE over VLAN . . . 170
Configuring MPLS over VLAN . . . 171
Configuring IP over VLAN and PPPoE over VLAN . . . 172
Configuring S-VLAN Subinterfaces . . . 175
Configuring an S-VLAN Subinterface . . . 175
Configuring PPPoE over an S-VLAN . . . 175
Example: Configuring S-VLAN Tunnels for Layer 2 Services over MPLS . . . 177
S-VLAN Oversubscription . . . 179
ACI-based VLAN Subinterfaces per S-VLAN Overview . . . 180
Configuring the Number of ACI-based VLAN Subinterfaces per S-VLAN . . . 181

Chapter 6 Monitoring VLAN and S-VLAN Subinterfaces . . . 183

Monitoring VLAN Subinterfaces . . . 183
Monitoring Interface Rate Statistics for VLAN Subinterfaces . . . 186
Monitoring Fast Ethernet VLAN or S-VLAN Subinterfaces . . . 188
Monitoring Gigabit Ethernet or 10-Gigabit Ethernet VLAN or S-VLAN Subinterfaces . . . 190

Chapter 7 Configuring 802.3ad Link Aggregation and Link Redundancy 193
Understanding IEEE 802.3ad Link Aggregation 193
LACP 194
Higher-Level Protocols 194
Load Balancing and QoS 195
Ethernet Link Aggregation and MPLS 195
IPv6 Packets and LAG 195
802.3ad Link Aggregation Platform Considerations 196
Module Requirements 196
Interface Specifiers 196
802.3ad Link Aggregation References 197
Configuring 802.3ad Link Aggregation 197
Configuring an Ethernet Physical Interface 197
Configuring a LAG Bundle 198
Configuring IP for a LAG Bundle 198
Configuring IPv6 for a LAG Bundle 198
Configuring a VLAN Subinterface for a LAG Bundle 199
Configuring a PPPoE Subinterface for a LAG Bundle 199
Configuring MPLS for a LAG Bundle 200
Example: Configuring 802.3ad Link Aggregation 200
Ethernet Link Redundancy Overview 206
Ethernet Link Redundancy Configuration Models 207
Ethernet Link Redundancy Configuration Diagrams 208
GE-2 Line Module Configurations 208
FE-8 Line Module Configurations 209
E120 and E320 Routers Configurations 210
Dual-Homed Configurations with LAG Disabled 211
Ethernet Link Redundancy Behavior Overview 211
Link Failure and Acquisition 211
Protecting Against Physical Link Failure 211
Protecting Against Virtual Link Failure 212
Reverting After a Failover 212
LACP Configuration and Member Link Behavior 213
Member Link with Non-LAG Partner 213
Ethernet Link Redundancy and RSTP 214
Acquiring Initial Links 214
Detecting Failures 215
Failing Over 215
Configuring Ethernet Link Redundancy 215
Monitoring a Specified Ethernet Member Link in an IEEE 802.3ad LAG Bundle . . 216
Monitoring Ethernet Member Links in all IEEE 802.3ad LAG Bundles 221

Chapter 8 Configuring IEEE 802.3ah OAM Link-Fault Management 225
Ethernet OAM Link-Fault Management Overview 226
Ethernet OAM Link-Fault Management Platform Considerations 227
Module Requirements 227
Interface Specifiers 227
Ethernet OAM Link-Fault Management References 228
OAM Messages 228

OAM Elements Overview . . . 229
OAM Client . . . 230
OAM Sublayer . . . 230
Control Block . . . 231
Multiplexer . . . 231
Parser . . . 231
OAM Feature Overview . . . 232
OAM Discovery Feature . . . 232
Information OAM PDU Components . . . 233
Transmission Settings for Information OAM PDUs . . . 233
OAM Link Monitoring Feature . . . 234
Supported Error Events for Tracking Link Faults . . . 235
Actions Performed on Exceeding Threshold Values . . . 235
OAM Remote Fault Detection Feature . . . 236
Link Fault . . . 236
Dying Gasp . . . 236
Critical Event . . . 236
OAM Remote and Local Loopback Feature . . . 237
Interrelationship of OAM Link-Fault Management with Ethernet Subsystems . . 238
Guidelines for Configuring 802.3ah OAM Link-Fault Management . . . 239
Configuring 802.3ah OAM Link-Fault Management . . . 240
Example: Configuring 802.3ah OAM Link-Fault Management and Enabling Remote Failure Monitoring on an Interface . . . 246
Example: Enabling Remote Loopback Support on the Local Interface . . . 247
Monitoring OAM Link-Fault Management Discovery Settings for an Interface . . 247
Monitoring OAM Link-Fault Management Statistics for an Interface . . . 250
Monitoring OAM Link-Fault Management Configuration for an Interface . . . 252
Monitoring OAM Link-Fault Management Sessions on All Configured Interfaces . . . 255

Chapter 9 Configuring Point-to-Point Protocol . . . 261

Understanding PPP . . . 262
Framing . . . 262
Error Frames . . . 263
Understanding PPP Link Control Protocol . . . 263
LCP Negotiation Parameters . . . 263
Validation of LCP Peer Magic Number . . . 264
Broadband Remote Access Support for PPP Overview . . . 265
Authentication . . . 266
Rate Limiting for PPP Control Packets . . . 266
Understanding Extensible Authentication Protocol . . . 267
EAP Types . . . 267
EAP Packet Retransmission . . . 268
EAP Behavior in an L2TP Environment . . . 269
When the E Series Router Acts as a LAC . . . 269
When the E Series Router Acts as an LNS . . . 269
Limitations . . . 269
Performance . . . 270
Remote Peer Scenarios During Negotiation of PPP Options . . . 271

IPCP Lockout and Local IP Address Pool Restoration Overview 272
IPCP Negotiation with Optional Peer IP Address Overview 273
Processing NCP Negotiations in a Dual-Stack Environment Overview 273
Enabling IPCP and IPv6CP Negotiations for IPv4 and IPv6 Clients Based on RADIUS-Returned Attributes 274
Sending the Acct-Start and Acct-Update Messages After NCP Negotiation Is Completed 275
Overview of Processing IPCP Negotiations for Dual-Stack Subscribers 275
Guidelines for Configuring the AAA Server for Release of IPv4 Addresses for Dual-Stack Subscribers 276
Releasing IPv4 Addresses During Termination of PPP Sessions 277
IPCP Renegotiation of IPv4 Addresses for Dual-Stack Subscribers 277
Rate Limit on IPCP Negotiations 279
PPP Platform Considerations 279
Module Requirements 280
Interface Specifiers 280
PPP References 281
Configuring PPP over a Serial Interface 281
Overview of Sequencing NCP Packets for POS Interfaces with PPP Encapsulation 283
Sequencing NCP Packets for POS Interfaces with PPP Encapsulation 283
Configuring Optional PPP Configuration Tasks 284
Adding a Description or Alias for a Static PPP Interface 285
Configuring the IPCP Lockout Option for Each PPP Interface 286
Configuring the IPCP Netmask Option for Each PPP Interface 286
Configuring the KeepAlive Timeout for an Interface 287
Disabling the Negotiation of the Local Magic Number 287
Configuring the Router to Ignore a Mismatch of the Peer Magic Number 288
Configuring the Maximum Number of Renegotiation Attempts from a PPP Client 288
Configuring the Maximum Receive Unit for PPP 289
Forcing the PPP Interface into a Passive Mode 290
Configuring the PPP Peer to Take Precedence of DNS and WINS Addresses . . . 290
Configuring the PPP IP Address as Optional in an IPCP Request 291
Terminating the PPP Session 291
Configuring PPP Authentication 291
Requesting Authentication from a PPP Peer 292
Configuring IPCP Renegotiations in a Dual-Stack Network for Optimal Utilization of Released IPv4 Addresses 294
Modifying the Challenge Length for CHAP Authentication 295
Configuring the Maximum Retries for PAP and CHAP Authentication 295
PPP Accounting Statistics Overview 296
Setting a Baseline for PPP Interface Statistics 297
Monitoring PPP Interfaces 297
Monitoring Multilinked and Nonmultilinked PPP Interfaces 309
Monitoring the Status of an IP Address in IPCP Configuration 311
Monitoring AAA IPv4 Address Saving 311
Troubleshooting PPP Interfaces 312
Monitoring the Maximum Timeout of PPP Sessions 313

Chapter 10 Configuring Multilink PPP 315
MLPPP Overview 316
Application 316
MLPPP LCP Extensions 317
MLPPP Link Selection Overview 318
MLPPP Platform Considerations 318
MLPPP Interface Specifiers 319
MLPPP Module Requirements 320
MLPPP References 320
Supported MLPPP Features 320
MLPPP Unsupported Features 325
Configuring Static MLPPP 325
Before You Configure Static MLPPP 326
Configuring Static MLPPP and Adding Member Links to a Multilink Bundle 326
Example: Configuring Fragmentation and Reassembly for Static MLPPP 327
MLPPP Contextual Command Differences 328
Configuring MLPPP Authentication and Other Attributes 329
Configuring Hash-Based MLPPP Link Selection 329
Example: Configuring Hash-Based MLPPP Link Selection 330
Configuring Dynamic MLPPP 332
MLPPP Fragmentation and Reassembly 332
MLPPP Fragmentation and Reassembly Overview 333
Application 333
MLPPP Fragmentation and Reassembly Supported Parameters 333
MLPPP Fragmentation and Reassembly Module Requirements 334
MLPPP Fragmentation and Reassembly Configuration Parameters 334
MLPPP Bundle Validation and Configuration Guidelines 334
MLPPP Fragmentation Guidelines 335
MLPPP Reassembly Guidelines 335
MLPPP Bundle Validation Failure 336
Recovering from MLPPP Bundle Validation Failure 336
Configuring Fragmentation and Reassembly for Dynamic MLPPP 336
Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over PPPOE 338
Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over L2TP 339
Example: Configuring MLPPP Fragmentation and Reassembly for an MLPPP Bundle 341
Multiclass MLPPP Overview 342
Monitoring MLPPP 342
Monitoring Baseline Statistics for MLPPP Interfaces 342
Monitoring MLPPP Information Configured on PPP Interfaces 348
Monitoring MLPPP Summary Information 359

Chapter 11 Configuring Multiclass Multilink PPP 361
Multiclass MLPPP Overview 361
Multiclass MLPPP Fragmentation and Reassembly 361
Multiclass MLPPP Configuration Guidelines 362
Multiclass MLPPP Traffic Classes Overview 362
Multiclass MLPPP LCP Extensions Overview 363
Multiclass MLPPP Platform Considerations 363
Module Requirements 364
Interface Specifiers 364
Multiclass MLPPP References 364
Configuring Multiclass MLPPP 364
Enabling Multiclass MLPPP 365
Configuring Traffic Classes on Multiclass MLPPP Interfaces 366
Configuring Fragmentation on Multiclass MLPPP Interfaces 366
Configuring Reassembly on Multiclass MLPPP Interfaces 367
Example: Configuring Multiclass MLPPP on a Dynamic Interface 368
Example: Configuring Multiclass MLPPP on a Static Interface 369
Monitoring Multiclass MLPPP 369
Chapter 12 Configuring Packet over SONET 375
Overview 375
POS Features 375
SONET/SDH 376
Platform Considerations 376
Module Requirements 377
Interface Specifiers 377
References 377
Before You Configure POS 378
Configuration Tasks 378
Monitoring POS 382
Chapter 13 Configuring Point-to-Point Protocol over Ethernet 387
Understanding PPPoE 388
PPPoE Stages 388
Discovery 388
Session 389
PPPoE MTU Configuration 389
PPPoE Service Name Tables Overview 390
Features 391
Table Structure 391
Enabling the Service Name Table for Use 392
Identification of Subscribers Using the PPPoE Remote Circuit Identifier 392
Application 393
PPPoE Remote Circuit ID Capture 393
PPPoE Remote Circuit ID Format 393
Remote Circuit ID Delimiter 394
Format for dsl-forum-1 Keyword 394
Format Examples for dsl-forum-1 Keyword 395
Use by RADIUS or L2TP 396

System Event Log 397
PPPoE Platform Considerations 397
Module Requirements 398
Interface Specifiers 398
PPPoE References 398
Access Nodes in Ethernet Aggregation Networks Overview 399
ATM-to-Ethernet Interworking Overview 400
Configuring PPPoE over ATM 402
Guidelines for Overriding the PPPoE Maximum Session Value 403
Overview of IWF PPPoE Sessions with Duplicate MAC Addresses 404
Guidelines for Configuring Duplicate Protection for IWF PPPoE Sessions 405
Single DSLAM Connected to a PPPoE Access Concentrator 406
Multiple DSLAMs Connected to a PPPoE Access Concentrator 407
PPPoE with Ethernet Modules Configuration Overview 408
PPPoE Interface and Subinterface Limits 410
Configuring IPv4 and IPV6 over static PPPoE with VLANs 410
Configuring IPv4 over PPPoE Without VLANs 411
PADM Messages Overview 412
Configuring MOTM Messages from Privileged Exec Mode 413
Configuring MOTM Messages from Interface Configuration Mode 413
Configuring MOTM Messages from Profile Configuration Mode 414
Configuring URL Messages from Interface Configuration Mode 414
Configuring URL Messages from Profile Configuration Mode 415
PADN Messages Overview 415
Configuring PADN Messages 416
Creating and Populating PPPoE Service Name Tables 416
Enabling Usage of PPPoE Service Name Tables with Static Interfaces over ATM 418
Enabling Usage of PPPoE Service Name Tables with Static Interfaces over Ethernet 419
Enabling Usage of PPPoE Service Name Tables with Dynamic Interfaces 419
Configuring PADS Packet Content 420
Configuring PPPoE Remote Circuit ID Capture 421

Chapter 14 Monitoring and Troubleshooting Point-to-Point Protocol over Ethernet 425

Setting a Baseline for PPPoE Interface Statistics 425
Monitoring Tunnel Parameters Configured for L2TP Tunnel Definitions 426
Monitoring the PPPoE Interface 427
Monitoring the PPPoE Service Name Table 433
Monitoring the PPPoE Subinterface 434
Monitoring the PPPoE Profile 436
Monitoring the RADIUS Override Settings 441
Troubleshooting PPPoE Interfaces 441

Chapter 15 Configuring Bridged IP 445
Overview 445
Proxy ARP 445
DHCP 446
Platform Considerations 446
Module Requirements 446
Interface Specifiers 447
References 447
Before You Configure Bridged IP 447
Configuring Bridged IP 448
Chapter 16 Configuring Bridged Ethernet 451
Overview 451
Bridged Ethernet Application 452
Assigning MAC Addresses 452
VLAN and S-VLAN Configurations 452
Platform Considerations 454
Module Requirements 454
Interface Specifiers 454
References 455
Configuring Bridged Ethernet 455
Configuring IP with PPPoE Terminated at the Router 455
Alternative Configuration 460
Configuring VLANs over Bridged Ethernet 460
Configuring VLAN Subinterfaces over Bridged Ethernet 461
Configuring Higher-Level Protocols over VLANs 461
Configuring IP over VLAN 462
Configuring PPPoE over VLAN 462
Configuring MPLS over VLAN 462
Configuring S-VLANs over Bridged Ethernet 465
Configuring S-VLAN Subinterfaces over Bridged Ethernet 466
Configuring Higher-Level Protocols over S-VLANs 467
Configuring the MTU Size for Bridged Ethernet 468
Monitoring Bridged Ethernet 468
Chapter 17 Configuring Transparent Bridging 473
Overview 473
How Transparent Bridging Works 473
Bridge Groups and Bridge Group Interfaces 474
Bridge Interface Types and Supported Configurations 475
Subscriber Policies 476
Concurrent Routing and Bridging 477
Transparent Bridging and VPLS 478
Unsupported Features 478
Platform Considerations 478
Module Requirements 478
Interface Specifiers 479
References 479
Before You Configure Transparent Bridging 479

Configuration Tasks 480
Creating Bridge Groups 481
Configuring Optional Bridge Group Attributes 481
Configuring Bridge Group Interfaces 484
Configuring Subscriber Policies 486
Enabling Concurrent Routing and Bridging 491
Configuring Explicit Routing 491
Configuration Examples 493
Example 1: Bridging with Bridged Ethernet 493
Example 2: Bridging with VLANs 494
Monitoring Transparent Bridging 495
Setting Statistics Baselines 496
Removing Dynamic MAC Address Entries 496
Monitoring Bridge Groups 498
Monitoring Bridge Interfaces 505
Monitoring Subscriber Policies 507

Chapter 18 Configuring Cisco HDLC 509

Overview 509
Framing 510
Error Frames 510
SLARP Keepalive 510
Platform Considerations 510
Module Requirements 510
Interface Specifiers 511
Before You Configure Cisco HDLC 511
Configuration Tasks 511
Optional Tasks 512
Configuration Example 514
Configuring IPv6 over Cisco HDLC Interfaces 514
Monitoring Cisco HDLC 515

Chapter 19 Configuring Upper-Layer Dynamic Interfaces 519

Dynamic Interfaces Overview 519
Autodetection 520
Types of Dynamic Interfaces 520
Upper-Layer Dynamic Interface Configurations 521
Profiles 522
RADIUS Authentication 523
ATM Oversubscription for Dynamic Interfaces 523
How Oversubscription Works 524
Static ATM 1483 Subinterfaces 524
Bulk-Configured VC Ranges 524
Combination of Static ATM 1483 Subinterfaces and Bulk-Configured VC Ranges 525

Ethernet Oversubscription for Dynamic Interfaces 525
Upper-Layer Dynamic Interfaces Platform Considerations 525
Module Requirements 525
Interface Specifiers 526
RADIUS References for Upper-Layer Dynamic Interfaces 526
Upper-Layer Dynamic Interfaces over Static ATM Overview 527
How to Configure Upper-Layer Dynamic Interfaces Using the RADIUS Server . . 527
Configuring a Local Subscriber for a Dynamic IPoA or Bridged Ethernet Interface 528
Subscriber Authentication on Dynamic Bridged Ethernet over Static ATM Interfaces 528
Configuration Method Using subscriber Command 529
Configuration Method Using Subscriber Management Application . . . 529
Dynamic IP Route Insertion in the Routing Table Overview 530
Configuring a Dynamic Interface over an ATM 1483 Subinterface 530
Dynamic Encapsulation Type Lockout 532
How Encapsulation Type Lockout Works 532
Benefits of Encapsulation Type Lockout 533
Encapsulation Type Lockout Based on DSL Forum VSAs for IWF PPPoE Sessions 534
Guidelines for Configuring Encapsulation Type Lockout for PPPoE Sessions 535
Guidelines for Configuring Encapsulation Type Lockout for IWF PPPoE Sessions 535
Encapsulation Type Lockout for IWF PPPoE Sessions Overview 536
Creating a PVC on an ATM 1483 Subinterface 537
Dynamic PPP and PPPoE Interfaces over Static ATM 538
Dynamic PPP and PPPoE Interfaces over Static ATM Overview 538
Configuring a PPP or PPPoE Dynamic Interface over an ATM 1483 Subinterface 539
Overview of Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations 540
Dynamic PPPoE Interfaces over PPPoE Static Interfaces 542
Configuring Dynamic PPPoE over Static PPPoE with ATM Interface Columns 543
Configuring Dynamic PPPoE over Static PPPoE with Ethernet Interface Columns 545
Configuring Dynamic PPPoE over Static PPPoE with Ethernet and VLAN Interface Columns 546
Configuring IPv4 and IPv6 Interface Columns over Static and Dynamic PPPoE 548

Configuring Dynamic PPPoE over Static PPPoE with Ethernet and S-VLAN Interface Columns . . . 549
S-VLAN Oversubscription for Dynamic PPPoE Interfaces over Static PPPoE Overview . . . 550
Encapsulation Type Lockout for PPPoE Clients Overview . . . 550
Differences from Lockout Configuration for PPPoE over Static ATM . . . 551
Configuring Encapsulation Type Lockout for PPPoE Clients . . . 552
Configuring and Verifying Lockout for PPPoE Clients . . . 552
Clearing the Lockout Condition for a PPPoE Client . . . 553
Dynamic IPoA Interfaces Overview . . . 556
Configuring a Dynamic IPoA Interface . . . 556
Dynamic Bridged Ethernet Interfaces . . . 558
Dynamic Bridged Ethernet Interfaces Overview . . . 559
Configuring a Dynamic Bridged Ethernet Interface . . . 559
Configuring Subscriber Management for IP Subscribers on Dynamic Bridged Ethernet Interfaces . . . 561
Example: Configuring IP Subscribers on a Dynamic Bridged Ethernet Interface Using the subscriber Command . . . 562
Example: Configuring IP Subscribers on a Dynamic Bridged Ethernet Interface Using IP Subscriber Management . . . 563
Dynamic Interface Configuration Using a Profile . . . 565
Profile Considerations for Dynamic Interfaces . . . 566
Profile Characteristics . . . 566
Bridged Ethernet Characteristics . . . 566
IP Characteristics . . . 566
IPv6 Characteristics . . . 567
L2TP Characteristics . . . 568
MLPPP and PPP Characteristics . . . 568
PPPoE Characteristics . . . 570
VLAN Characteristics . . . 570
How to Work with Profiles . . . 571
Creating a Profile for Dynamic Interfaces . . . 571
Assigning a Profile to a Dynamic Interface . . . 572
Example: Configuring a Profile for Dynamic Interfaces . . . 573
Configuring Profile Characteristics . . . 576
Configuring Bridged Ethernet Characteristics for a Profile . . . 576
Configuring IPv4 Characteristics for a Profile . . . 577
Configuring IPv6 Characteristics for a Profile . . . 580
Configuring L2TP Characteristics for a Profile . . . 582
Configuring MLPPP and PPP Characteristics for a Profile . . . 582
Configuring PPPoE Characteristics for a Profile . . . 586
Configuring VLAN Characteristics for a Profile . . . 588
Scripts and Macros for Dynamic Interfaces Overview . . . 589
Reassigning a Debug Profile Before Troubleshooting PPP and PPPoE Dynamic Interfaces . . . 590

Chapter 20 Monitoring Upper-Layer Dynamic Interfaces . . . 593

Monitoring Configuration Information of an ATM AAL5 Interface . . . 593
Monitoring Status or Summary Information for ATM Subinterfaces . . . 595

Monitoring Summary Information for ATM VCs and Reserved VC Ranges 600
Monitoring Total Static and Dynamic Interface Counts for Interface Columns . . 602
Monitoring Summary Information About the Encapsulation Type Lockout for PPPoE Clients 603
Monitoring Detailed Information About the Current Encapsulation Type Lockout Condition for PPPoE Clients 604
Monitoring the Source MAC Address of a PPPoE Client 605
Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces . . . 606
Monitoring Status and Configuration Information for VLAN Subinterfaces 614

Chapter 21 Configuring Dynamic Interfaces Using Bulk Configuration 619

Configuring Dynamic Interfaces Using Bulk Configuration Overview 620
Bulk Dynamic Interface Configurations 621
Profiles 622
ATM Oversubscription for Bulk-Configured VC Ranges 622
Bulk-Configured VC Ranges 623
Combination of Static ATM 1483 Subinterfaces and Bulk-Configured VC Ranges 623
Configuring Dynamic Interfaces Using Bulk Configuration Platform Considerations 624
Module Requirements 624
Interface Specifiers 625
Configuring Dynamic Interfaces Using Bulk Configuration References 625
Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview 626
ATM Base Profile Overview 627
Nested Profile Assignments Overview 628
Additional Profile Characteristics for Upper Interfaces 629
Bulk Configuration of VC Ranges Overview 629
Bulk Configuration and CAC Overview 630
Changing VC Subranges Overview 631
Bulk Configuration and VC Classes Overview 631
Dynamic ATM 1483 Subinterface Creation Overview 632
Overriding ATM 1483 Base Profile Assignments Overview 632
Static ATM Interfaces Within VC Subranges Overview 633
Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations Overview 634
Subscriber Authentication on Dynamic Bridged Ethernet over Dynamic ATM Interfaces 635
Configuration Method Using atm atm1483 subscriber Command 635
Configuration Method Using Subscriber Management Application 635
Configuring a Dynamic ATM 1483 Subinterface 636
Creating a Base Profile to Configure Attributes for a Dynamic ATM 1483 Subinterface 639
Configuring a Base Profile for Dynamic ATM 1483 Subinterface 640
Configuring a Base Profile to Set Up an Advisory Receive Speed for a Dynamic ATM 1483 Subinterface 640
Specifying the Types of Dynamic Upper-Interface Encapsulations Accepted or Detected by a Dynamic ATM 1483 Subinterface 640

Configuring a Base Profile to Assign a Text Description for a Dynamic ATM 1483 Subinterface . . . 641
Adding a Nested Profile Assignment to a Base Profile for a Dynamic ATM 1483 Subinterface . . . 641
Configuring a Local Subscriber for a Dynamic Upper-Interface Encapsulation Type Configured over a Dynamic ATM 1483 Subinterface . . . 641
Assigning a Previously Configured VC Class to a Base Profile . . . 642
Configuring a Base Profile to Apply Encapsulation, Traffic-Shaping and OAM Parameters to a Bulk Range of PVCs Configured on an ATM AAL5 Interface . . . 643
Creating a Bulk-Configured VC Range on a Static ATM AAL5 Interface . . . 644
Configuring the Static ATM AAL5 Interface to Support Autodetection of an ATM 1483 Dynamic Encapsulation Type . . . 646
Configuring Overriding Profile Assignments . . . 646
Assigning an Overriding Profile to an ATM PVC Within a Bulk-Configured VC Subrange . . . 646
Removing an Overriding Profile Assignment from an ATM PVC . . . 648
Removing an Overriding Profile Assignment from a VC Range or VC Subrange . . . 649
Assigning the Base Profile Configured for a Dynamic ATM 1483 Subinterface to the VC Range Configured on a Static ATM AAL5 Interface . . . 649
Assigning an Overriding Profile to a Single ATM PVC that Exists Within a Bulk-Configured VC Subrange . . . 650
Disabling a Dynamic ATM 1483 Interface . . . 651
Changing VC Subranges . . . 651
Adding New VC Subranges to an Existing VC Range . . . 651
Removing VC Subranges from an Existing VC Range . . . 652
Modifying a VC Subrange by Shortening or Expanding the Subrange Values . . . 653
Merging Multiple VC Subranges Belonging to an Existing VC Range . . . 653
Changing the Administrative State of VC Subranges . . . 654
Configuring Static ATM Interfaces Within VC Subranges . . . 655
Creating a Static ATM Interface Within an Existing Bulk-Configured VC Subrange . . . 655
Creating a Bulk-Configured VC Subrange that Includes an Existing Static ATM Interface . . . 656
Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview . . . 657
VLAN Base Profile Overview . . . 660
VLAN Nested Profile Assignments . . . 661
VLAN Additional Profile Characteristics for Upper Interfaces . . . 661
Bulk Configuration of VLAN Ranges Overview . . . 662
Changing VLAN Subranges Overview . . . 662
Bulk Configuration of VLAN Ranges Using Agent-Circuit-Identifier Information Overview . . . 663
Dynamic VLAN Subinterface Creation Overview . . . 665
Overriding VLAN Base Profile Assignments Overview . . . 665
Static VLAN Subinterfaces Within VLAN Subranges Overview . . . 666
Configuring Dynamic VLAN Subinterfaces . . . 667

Creating a Base Profile to Configure Attributes for a Dynamic VLAN Subinterface . . . 669
Configuring a Base Profile for VLAN Subinterface . . . 669
Setting an Advisory Receive Speed for VLAN Subinterfaces . . . 670
Setting an Advisory Connect Speed for VLAN Subinterfaces . . . 670
Specifying the Types of Dynamic Upper-Interface Encapsulations that are Accepted or Detected by a Dynamic VLAN Subinterface . . . 670
Creating a VLAN Subinterface that is Based on the Agent-Circuit-ID Information . . . 671
Assigning Descriptions to VLAN Subinterfaces . . . 671
Assigning a VLAN Policy List to a Profile . . . 671
Adding a Nested Profile Assignment to a Base Profile for a Dynamic VLAN Subinterface . . . 672
Specifying the Available Ethertypes to Create a Dynamic VLAN Subinterface . . . 673
Specifying a Service Profile Name for a Dynamic VLAN . . . 673
Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface . . . 674
Configuring the Static VLAN Major Interface to Support Autodetection of a VLAN Dynamic Interface Type . . . 675
Configuring Dynamic VLAN Subinterfaces Based on Agent Circuit Identifier Information . . . 676
Configuring Overriding Profile Assignments for VLAN Major Interfaces . . . 678
Assigning an Overriding Profile to a VLAN Within a Bulk-Configured VLAN Subrange . . . 678
Removing an Overriding Profile Assignment from a VLAN . . . 679
Removing Overriding Profile Assignments from a VLAN Range or VLAN Subrange . . . 680
Assigning the Base Profile Configured for a Dynamic VLAN Subinterface to the Single-Tagged VLAN IDs or Double-Tagged S-VLAN IDs . . . 681
Assigning an Overriding Profile to a Single-Tagged VLAN ID or Double-Tagged S-VLAN ID . . . 682
Configuring VLAN as the Encapsulation Method for the Interface . . . 682
Disabling a Dynamic VLAN subinterface . . . 683
Changing VLAN Subranges . . . 683
Adding VLAN Subranges . . . 683
Removing VLAN Subranges . . . 684
Modifying VLAN Subranges . . . 685
Merging VLAN Subranges . . . 685
Changing the Administrative State of VLAN Subranges . . . 686
Configuring Static VLAN Subinterfaces Within VLAN Subranges . . . 688
Creating a Static VLAN Subinterface Within a VLAN Subrange . . . 688
Creating a VLAN Subrange that Includes a Static VLAN Subinterface . . . 689

Chapter 22 Monitoring Dynamic Interfaces and Profiles . . . 691

Monitoring Configuration Information of an ATM AAL5 Interface . . . 691
Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface . . . 693
Monitoring Status or Summary Information for ATM Subinterfaces . . . 696
Monitoring Summary Information for ATM VCs and Reserved VC Ranges . . . 702
Monitoring Total Static and Dynamic Interface Counts for Interface Columns . . . 704

Monitoring Summary Information About the Encapsulation Type Lockout for PPPoE Clients 705
Monitoring Detailed Information About the Current Encapsulation Type Lockout Condition for PPPoE Clients 706
Monitoring the Source MAC Address of a PPPoE Client 707
Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces 708
Monitoring Base Profile Assignments and Overriding Profile Assignments for VLAN Major Interface 716
Monitoring Dynamic VLAN Subinterfaces Created with an Overriding Profile Assignment 718
Monitoring Status and Configuration Information for VLAN Subinterfaces 719

Part 2 Index

Index 725

List of Figures

Part 1	Chapters
Chapter 1	Configuring ATM . . . 3
	Figure 1: ATM Interface Column . . . 4
	Figure 2: NBMA Interface Stack . . . 13
	Figure 3: Configuring an ATM Interface, Subinterface, and PVC . . . 22
Chapter 2	Configuring Frame Relay . . . 103
	Figure 4: Interconnection and Relationship of NNIs and Subnetworks . . . 105
Chapter 3	Configuring Multilink Frame Relay . . . 129
	Figure 5: MLFR Aggregation of T1 Lines into a Single Bundle . . . 130
	Figure 6: Terminating the Bundle at an MLFR Bridge . . . 130
	Figure 7: Structure of MLFR . . . 131
Chapter 4	Configuring Upper-Layer Protocols over Static Ethernet Interfaces . . . 149
	Figure 8: Multiplexing Multiple Protocols over a Single Physical Link . . . 150
	Figure 9: Example of IP over Ethernet Stacking Configuration Procedure . . . 152
	Figure 10: Example of PPPoE Stacking Configuration Procedure . . . 152
	Figure 11: Example of IP and MPLS Stacking Configuration Procedure . . . 153
	Figure 12: Example of IP, MPLS, and PPPoE Stacking Configuration Procedure . . . 154
Chapter 5	Configuring VLAN and S-VLAN Subinterfaces . . . 165
	Figure 13: Use of VLANs to Multiplex Different Protocols over a Single Physical Link . . . 166
	Figure 14: Example of IP/VLAN/Fast Ethernet Stacking Configuration Procedure . . . 170
	Figure 15: Example of PPPoE/VLAN/Fast Ethernet Stacking Configuration Procedure . . . 171
	Figure 16: Example of MPLS/VLAN/Fast Ethernet Stacking Configuration Procedure . . . 172
	Figure 17: Example of PPPoE over VLAN with IP over VLAN Stacking Configuration Procedure . . . 174
	Figure 18: Example of PPPoE over S-VLAN Stacking Configuration Procedure . . . 176
	Figure 19: S-VLAN Tunnels for Ethernet Layer 2 Services over MPLS . . . 178
Chapter 7	Configuring 802.3ad Link Aggregation and Link Redundancy . . . 193
	Figure 20: Interface Stack for 802.3ad Link Aggregation . . . 194
	Figure 21: Ethernet Link Redundancy Configuration Models . . . 208
	Figure 22: GE-2 Line Module Using Physical Port Redundancy . . . 208
	Figure 23: Single-Homed GE-2 Line Module Configuration . . . 209
	Figure 24: Single-Homed FE-8 Line Module Configuration (1:N) . . . 209

Figure 25: FE-8 Line Module with 4 Redundant Ethernet Links (1:1) 209
Figure 26: Single-Homed GE-4 IOA Configuration (1:4) 210
Figure 27: GE-8 IOA Configuration Across IOAs (1:N) 210
Figure 28: Dual-Homed Configuration (1:1) 211
Figure 29: Dual-Homed Heterogeneous Configuration in an RSTP Network 214

Chapter 8 Configuring IEEE 802.3ah OAM Link-Fault Management 225

Figure 30: OAM PDU Format 228
Figure 31: OAM Sublayer Interfaces 230
Figure 32: OAM Sublayer Entities 231
Figure 33: Interrelationship Between 802.3ah OAM and 802.3ad LAG 238

Chapter 9 Configuring Point-to-Point Protocol 261

Figure 34: Authentication with EAP 267

Chapter 10 Configuring Multilink PPP 315

Figure 35: MLPPP Aggregation of T1 Lines into a Single Bundle 316
Figure 36: Structure of MLPPP 317

Chapter 13 Configuring Point-to-Point Protocol over Ethernet 387

Figure 37: PPPoE over ATM 388
Figure 38: Example of PPPoE over ATM Stacking 400
Figure 39: Single DSLAM Connected to a PPPoE Access Concentrator 407
Figure 40: Multiple DSLAMs Connected to a PPPoE Access Concentrator 408
Figure 41: Example of Configuring IPv4 and IPv6 over PPPoE 409
Figure 42: Example of PPPoE Stacking 409

Chapter 16 Configuring Bridged Ethernet 451

Figure 43: Bridged Ethernet Topology, Router Terminating and Routing Traffic 452
Figure 44: Interface Stacking for VLANs over Bridged Ethernet 453

Chapter 17 Configuring Transparent Bridging 473

Figure 45: Bridge Group with Fast Ethernet and Gigabit Ethernet Bridge Interfaces 475

Chapter 19 Configuring Upper-Layer Dynamic Interfaces 519

Figure 46: Configuring an ATM 1483 Interface to Support Dynamic Interfaces . . 527
Figure 47: Single DSLAM Connected to a PPPoE Access Concentrator 536
Figure 48: Dynamic PPP Interface Columns 538
Figure 49: Dynamic PPPoE Interface Columns 539
Figure 50: Dynamic PPP Interface Columns 541
Figure 51: Dynamic PPPoE over Static PPPoE with ATM Interface Columns 543
Figure 52: Dynamic PPPoE over Static PPPoE with Non-VLAN Interface Columns 546
Figure 53: Dynamic PPPoE over Static PPPoE with VLAN Interface Columns . . 547
Figure 54: IPv4 and IPv6 Interface Columns over Static and Dynamic PPPoE . . 548
Figure 55: Dynamic PPPoE over Static PPPoE with S-VLAN Interface Columns 549
Figure 56: Dynamic IPoA over Static ATM 1483 Interface Columns 556
Figure 57: Dynamic Bridged Ethernet over Static ATM 1483 Interface Columns 559

Figure 58: Creating and Configuring a Profile 571
Figure 59: Assigning a Profile to a Static Interface 571

Chapter 21 Configuring Dynamic Interfaces Using Bulk Configuration 619

Figure 60: Dynamic Interface Columns over Dynamic ATM 1483 Subinterfaces 627
Figure 61: Dynamic Interface Columns over Dynamic VLAN Subinterfaces 657
Figure 62: Dynamic IP and PPPoE over Single Dynamic VLAN Subinterface . . . 658
Figure 63: Dynamic VLAN Subinterfaces for Subscribers 659

List of Tables

About the Documentation xxxi

Table 1: Notice Icons xxxii
Table 2: Text and Syntax Conventions xxxii

Part 1 Chapters

Chapter 1 Configuring ATM 3

Table 3: Scheduling Priorities for Traffic Classes 7
Table 4: Traffic Parameters Used to Compute Bandwidth 8
Table 5: ATM Capabilities on Line Modules and I/O Modules 12
Table 6: Handling of F4 and F5 Loopback Cells Received 19
Table 7: F5 OAM Configuration Tasks and Associated Commands 49
Table 8: Commands to Configure VC Class Attributes 56

Chapter 3 Configuring Multilink Frame Relay 129

Table 9: LIP Messages and Functions 130

Chapter 6 Monitoring VLAN and S-VLAN Subinterfaces 183

Table 10: show vlan subinterface Output Fields 185
Table 11: monitor vlan interface Output Fields 188
Table 12: show interfaces fastEthernet Output Fields 189
Table 13: show interfaces gigabitEthernet Output Fields 191

Chapter 7 Configuring 802.3ad Link Aggregation and Link Redundancy 193

Table 14: Behavior of Member Links Using Local and Remote LACP Modes 213
Table 15: show interfaces lag Output Fields 217
Table 16: show interfaces lag member Output Fields 222

Chapter 8 Configuring IEEE 802.3ah OAM Link-Fault Management 225

Table 17: show ethernet oam lfm discovery Output Fields 248
Table 18: show ethernet oam lfm statistics Output Fields 251
Table 19: show ethernet oam lfm status Output Fields 253
Table 20: show ethernet oam lfm summary Output Fields 256

Chapter 9 Configuring Point-to-Point Protocol 261

Table 21: Supported EAP Types 268
Table 22: show ppp interface Output Fields 301
Table 23: show ppp interface summary Output Fields 310
Table 24: show ppp peer-ip-address-optional Output Fields 311
Table 25: show aaa ipv4-addr-saving Output Fields 311
Table 26: show session-To-Thirteen-Years Output Fields 313

Chapter 10 Configuring Multilink PPP 315

Table 27: Supported Configurations for MLPPP Fragmentation and Reassembly . . . 333
Table 28: show ppp interface statistics Output Fields . . . 346
Table 29: show ppp interface mlppp Output Fields . . . 353
Table 30: show ppp interface summary Output Fields . . . 360

Chapter 11 Configuring Multiclass Multilink PPP . . . 361

Table 31: show ppp interface mlppp Output Fields . . . 371

Chapter 12 Configuring Packet over SONET . . . 375

Table 32: Most Common SONET/SDH Implementations . . . 376

Chapter 13 Configuring Point-to-Point Protocol over Ethernet . . . 387

Table 33: Sample PPPoE Service Name Table . . . 391
Table 34: Configuring Nondefault Formats for the PPPoE Remote Circuit ID . . . 393
Table 35: Interface Specifier Format Examples for dsl-forum-1 Keyword . . . 395
Table 36: PPPoE Duplicate Protection Scenarios for IWF and non-IWF PPPoE Sessions . . . 406
Table 37: Default PPPoE Service Name Table . . . 417
Table 38: PPPoE Service Name Table with Entries . . . 417

Chapter 14 Monitoring and Troubleshooting Point-to-Point Protocol over Ethernet . . . 425

Table 39: show aaa tunnel-parameters Output Fields . . . 426
Table 40: show pppoe interface Output Fields . . . 430
Table 41: show pppoe-service-name-table Output Fields . . . 434
Table 42: show pppoe subinterface Output Fields . . . 435
Table 43: show profile Output Fields . . . 438
Table 44: show radius override Output Fields . . . 441

Chapter 15 Configuring Bridged IP . . . 445

Table 45: Prerequisite Tasks for Configuring Bridged IP . . . 448

Chapter 17 Configuring Transparent Bridging . . . 473

Table 46: Sample Bridge Group Forwarding Table . . . 475
Table 47: Default Subscriber Policies for Bridge Group Interfaces . . . 476
Table 48: Prerequisite Tasks for Configuring Transparent Bridging . . . 480

Chapter 19 Configuring Upper-Layer Dynamic Interfaces . . . 519

Table 49: Differences in Lockout Operation for Dynamic PPPoE Configurations . . . 551

Chapter 20 Monitoring Upper-Layer Dynamic Interfaces . . . 593

Table 50: show atm aal5 interface Output Fields . . . 594
Table 51: show atm subinterface Output Fields . . . 597
Table 52: show atm vc Output Fields . . . 601
Table 53: show columns Output Fields . . . 602
Table 54: show pppoe interface Encapsulation Type Lockout Output Fields . . . 603
Table 55: show pppoe interface lockout-time Output Fields . . . 604
Table 56: show pppoe subinterface Source MAC Address Output Fields . . . 606
Table 57: show profile Output Fields . . . 608
Table 58: show vlan subinterface Output Fields . . . 616

Chapter 22 Monitoring Dynamic Interfaces and Profiles 691
Table 59: show atm aal5 interface Output Fields 692
Table 60: show atm bulk-config Output Fields 695
Table 61: show atm subinterface Output Fields 699
Table 62: show atm vc Output Fields 703
Table 63: show columns Output Fields 704
Table 64: show pppoe interface Encapsulation Type Lockout Output Fields . . . 705
Table 65: show pppoe interface lockout-time Output Fields 706
Table 66: show pppoe subinterface Source MAC Address Output Fields 707
Table 67: show profile Output Fields 710
Table 68: show vlan bulk-config Output Fields 717
Table 69: show vlan profile Output Fields 718
Table 70: show vlan subinterface Output Fields 720

About the Documentation

- E Series and JunosE Documentation and Release Notes on page xxxi
- Audience on page xxxi
- E Series and JunosE Text and Syntax Conventions on page xxxi
- Obtaining Documentation on page xxxiii
- Documentation Feedback on page xxxiii
- Requesting Technical Support on page xxxiii

E Series and JunosE Documentation and Release Notes

For a list of related JunosE documentation, see http://www.juniper.net/techpubs/software/index.html.

If the information in the latest release notes differs from the information in the documentation, follow the *JunosE Release Notes.*

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

Audience

This guide is intended for experienced system and network specialists working with Juniper Networks E Series Broadband Services Routers in an Internet access environment.

E Series and JunosE Text and Syntax Conventions

Table 1 on page xxxii defines notice icons used in this documentation.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page xxxii defines text and syntax conventions that we use throughout the E Series and JunosE documentation.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents commands and keywords in text.	• Issue the **clock source** command. • Specify the keyword **exp-msg**.
Bold text like this	Represents text that the user must type.	**host1(config)#traffic class low-loss1**

`Fixed-width text like this`	Represents information as displayed on your terminal's screen.	**host1#show ip ospf 2**
		`Routing Process OSPF 2 with Router ID 5.5.0.250`
		`Router is an Area Border Router (ABR)`

Italic text like this	• Emphasizes words. • Identifies variables. • Identifies chapter, appendix, and book names.	• There are two levels of access: *user* and *privileged*. • *clusterId, ipAddress.* • *Appendix A, System Specifications*
Plus sign (+) linking key names	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
Syntax Conventions in the Command Reference Guide		
Plain text like this	Represents keywords.	terminal length
Italic text like this	Represents variables.	*mask*, *accessListName*

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
| (pipe symbol)	Represents a choice to select one keyword or variable to the left or to the right of this symbol. (The keyword or variable can be either optional or required.)	diagnostic | line
[ ] (brackets)	Represent optional keywords or variables.	[internal | external]
[ ]* (brackets and asterisk)	Represent optional keywords or variables that can be entered more than once.	[level1 | level2 | l1]*
{ } (braces)	Represent required keywords or variables.	{ permit | deny } { in | out } { *clusterId* | *ipAddress* }

Obtaining Documentation

To obtain the most current version of all Juniper Networks technical documentation, see the Technical Documentation page on the Juniper Networks Web site at http://www.juniper.net/.

To download complete sets of technical documentation to create your own documentation CD-ROMs or DVD-ROMs, see the Portable Libraries page at

http://www.juniper.net/techpubs/resources/index.html

Copies of the Management Information Bases (MIBs) for a particular software release are available for download in the software image bundle from the Juniper Networks Web site athttp://www.juniper.net/.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation to better meet your needs. Send your comments to **techpubs-comments@juniper.net**, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf .
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/ .
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Chapters

- Configuring ATM on page 3
- Configuring Frame Relay on page 103
- Configuring Multilink Frame Relay on page 129
- Configuring Upper-Layer Protocols over Static Ethernet Interfaces on page 149
- Configuring VLAN and S-VLAN Subinterfaces on page 165
- Monitoring VLAN and S-VLAN Subinterfaces on page 183
- Configuring 802.3ad Link Aggregation and Link Redundancy on page 193
- Configuring IEEE 802.3ah OAM Link-Fault Management on page 225
- Configuring Point-to-Point Protocol on page 261
- Configuring Multilink PPP on page 315
- Configuring Multiclass Multilink PPP on page 361
- Configuring Packet over SONET on page 375
- Configuring Point-to-Point Protocol over Ethernet on page 387
- Monitoring and Troubleshooting Point-to-Point Protocol over Ethernet on page 425
- Configuring Bridged IP on page 445
- Configuring Bridged Ethernet on page 451
- Configuring Transparent Bridging on page 473
- Configuring Cisco HDLC on page 509
- Configuring Upper-Layer Dynamic Interfaces on page 519
- Monitoring Upper-Layer Dynamic Interfaces on page 593
- Configuring Dynamic Interfaces Using Bulk Configuration on page 619
- Monitoring Dynamic Interfaces and Profiles on page 691

CHAPTER 1

Configuring ATM

This chapter introduces basic Asynchronous Transfer Mode (ATM) concepts, describes features of the ATM interfaces, and provides information for configuring ATM on E Series routers.

This chapter contains the following sections:

- Overview on page 3
- Platform Considerations on page 10
- References on page 11
- Supported Features on page 11
- ATM NBMA on page 13
- Operations, Administration, and Management of ATM Interfaces on page 14
- Before You Configure ATM on page 20
- Configuration Tasks on page 21
- Creating a Basic Configuration on page 21
- Setting Optional Parameters on page 23
- Configuring OAM on page 31
- Configuring an NBMA Interface on page 37
- Creating an NBMA Static Map on page 38
- Assigning Descriptions to Interfaces on page 40
- Sending Interface Descriptions to AAA on page 41
- Configuring Individual ATM PVC Parameters on page 43
- Configuring ATM VC Classes on page 52
- Configuring Dynamic ATM 1483 Subinterfaces on page 66
- Monitoring ATM on page 66

Overview

ATM is a high-speed networking technology that handles data in fixed-size units called cells. It enables high-speed communication between edge routers and core routers in an ATM network.

ATM Interfaces

An ATM port can have a major interface and one or more subinterfaces. An ATM subinterface is a mechanism that enables a single physical ATM interface to support multiple logical interfaces. Several logical interfaces can be associated with a single physical interface.

ATM subinterfaces meet the specifications in RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999), which replaces RFC 1483. All references to ATM subinterfaces in this chapter are still to ATM 1483 subinterfaces.

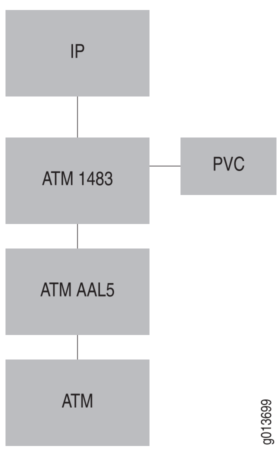
ATM 1483 subinterfaces are identified by user-defined numbers. To select a subinterface, you append a subinterface number to the port-level **interface atm** command.

When you create an ATM 1483 subinterface, you must configure a permanent virtual circuit (PVC). Protocols such as ATM require one or more virtual circuits over which data traffic is transmitted to higher layers in the protocol stack.

The ATM PVC on an ATM subinterface with an assigned IP address is reenabled after you reload the slot, or disable and reenable the slot, on which the ATM physical interface is configured.

Figure 1 on page 4 shows a typical point-to-point ATM interface column.

Figure 1: ATM Interface Column



ATM Physical Connections

ATM interfaces and subinterfaces support two types of connections—point-to-point and multipoint. The router defaults to point-to-point.

- Point-to-point—Indicates a standard connection; for example, connecting two ATM end stations
- Multipoint—Indicates a single-source end system connected to multiple destination end systems. Multipoint indicates a nonbroadcast multiaccess (NBMA) interface. See “ATM NBMA” on page 13.

Depending on the type of connection you choose, you can specify one or more PVCs on each interface. For a standard point-to-point ATM interface, you configure only one PVC. For NBMA ATM connections, you configure multiple circuits.

ATM Virtual Connections

A *virtual connection* (VC) defines a logical networking path between two endpoints in an ATM network. ATM *cells* travel from one point to the other over a virtual connection. An ATM cell is a package of information that is always 53 bytes in length, unlike a frame or packet, which has a variable length. An ATM cell has a cell header and a *payload*. The payload contains the user data.

The cell header includes an 8-bit virtual path identifier (VPI) and a 16-bit virtual channel identifier (VCI).

An ATM network can have two types of VCs, depending on the addressing used to switch the traffic:

- Virtual channel connection (VCC)
- Virtual path connection (VPC)

Virtual Channel Connection

A VCC uses all the addressing bits of the cell header to move traffic from one link to another. The VCC is formed by joining a series of virtual channels (VCs), which are logical circuits uniquely identified for each link of the network. On a VCC, switching is done based on the combined VPI and VCI values.

Virtual Path Connection

A VPC uses the higher-order addressing bits of the cell header to move traffic from one link to another. A VPC carries many VCCs within it. A VPC can be set up permanently between two points, and then switched.

VCCs can be assigned within the VPC easily and quickly. The VPC is formed by joining a series of virtual paths, which are the logical groups of circuits uniquely defined for each link of the network. On a VPC, switching is done based on the VPI value only.

ATM SVCs

JunosE Software does not support configuration and monitoring of ATM switched virtual circuits (SVCs) on the router.

ATM Adaptation Layer

The ATM Adaptation Layer (AAL) defines the conversion of user information into cells by segmenting upper-layer information into cells at the transmitter and reassembling them at the receiver. AAL1 and AAL2 handle intermittent traffic, such as voice and video, and are not relevant to the router. AAL3/4 and AAL5 support data communications by segmenting and reassembling packets.

E Series routers support the following AAL5 encapsulation types as specified in RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999), which replaces RFC 1483:

- aal5snap—LLC/SNAP
- aal5mux ip—VC-based multiplexing
- aal5autoconfig—LLC/SNAP or VC-based multiplexing. (See "Configuring Upper-Layer Dynamic Interfaces" on page 519.)
- aal5all—Martini encapsulation

NOTE: **The Juniper Networks E120 and E320 Broadband Services Routers do not support Martini encapsulation (aal5all) in the current release.**

Local ATM Passthrough

E Series routers support local ATM passthrough for ATM layer 2 services over Multiprotocol Label Switching (MPLS). Local ATM passthrough enables the router to emulate packet-based ATM switching. The ATM passthrough feature is useful for customers who run IP in most of their network but still have to carry a small amount of native ATM traffic.

Local ATM passthrough uses ATM Martini encapsulation to emulate ATM switch behavior. You can create pairs of cross-connected ATM VCs within the router. The router then passes AAL5 traffic between two VCs, regardless of the contents of the packets.

You can also use AAL0 encapsulation when you configure a local ATM passthrough connection. AAL0 encapsulation causes the router to receive raw ATM cells on this circuit and to forward the cells without performing AAL5 packet reassembly.

For more information, see *Configuring Layer 2 Services over MPLS* in *JunosE BGP and MPLS Configuration Guide*.

VCC Cell Relay Encapsulation

E Series routers support virtual channel connection (VCC) cell relay encapsulation for ATM layer 2 services over MPLS. VCC cell relay encapsulation is useful for voice-over-ATM applications that use AAL2-encapsulated voice transmission.

VCC cell relay encapsulation enables the router to emulate ATM switch behavior by forwarding individual ATM cells over an MPLS pseudowire (also referred to as an MPLS tunnel) created between two ATM VCCs, or as part of a local ATM passthrough connection between two ATM 1483 subinterfaces on the same router. The E Series implementation conforms to the required N-to-1 cell mode encapsulation method described in the Martini draft, Encapsulation Methods for Transport of ATM Over MPLS Networks—draft-ietf-pwe3-atm-encap-07.txt (April 2005 expiration), with the provision that only a single ATM virtual circuit (VC) can be mapped to an MPLS tunnel.

For more information, see *Configuring Layer 2 Services over MPLS* in *JunosE BGP and MPLS Configuration Guide*.

NOTE: The E120 and E320 routers do not support ATM over MPLS with VCC cell relay encapsulation in the current release.

Traffic Management

The scheduling priority for traffic classes depends on the type of router that you have. Table 3 on page 7 describes the scheduling priorities for each type of router.

Table 3: Scheduling Priorities for Traffic Classes

Scheduling Priority (from Highest to Lowest)	ERX7xx Models, ERX14xx Models, or the ERX310 Broadband Services Router	E120 and E320 routers
1	The following traffic classes are prioritized equally: • CBR • VBR-RT	CBR
2	The following traffic classes are prioritized equally: • VBR-NRT • UBR with a peak cell rate (PCR)	VBR-RT
3	UBR without PCR	VBR-NRT
4	–	UBR with or without PCR

The level of support for traffic management depends on the specific I/O module or IOA. See “Supported Features” on page 11.

Connection Admission Control

ATM networks use connection admission control (CAC) to determine whether to accept a connection request, based on whether allocating the connection’s requested bandwidth causes the network to violate the traffic contracts of existing connections. CAC is a set of actions that the network takes during connection setup or renegotiation.

The router supports CAC on PVCs on major ATM interfaces. This implementation of CAC determines available bandwidth based on port subscription bandwidth. The router maintains available bandwidth for each major ATM port. Bandwidth for VP tunnels is included in CAC computations.

Table 4 on page 8 lists the traffic parameter that the router uses for each service category to compute the bandwidth that the connection requires. For example, the peak cell rate is used to calculate how much bandwidth is required for CBR connections.

Table 4: Traffic Parameters Used to Compute Bandwidth

Service Category	Traffic Parameter Used to Calculate Required Bandwidth
CBR	PCR
VBR-RT	SCR
VBR-NRT	SCR
UBR	UBR bandwidth configured on the ATM major interface
UBR with PCR	UBR bandwidth configured on the ATM major interface

How CAC Works

With no connections, the available bandwidth is equal to the subscription port bandwidth. While connections are requested, the required bandwidth, which is based on the service category and traffic parameters of the connection, is compared against the available port bandwidth. If sufficient bandwidth is available, the router accepts the connection and updates the available port bandwidth accordingly.

Similarly, when a connection is deleted, the available port bandwidth is updated accordingly.

Configuring CAC

You enable and configure CAC on an ATM major interface using "atm cac" on page 27 . When you enable CAC on an ATM interface, you can optionally specify a subscription bandwidth and a UBR weight:

- The subscription bandwidth can be greater than the effective port bandwidth to allow oversubscription. The default value of the subscription bandwidth is the effective bandwidth of the ATM port.
- The UBR weight enables you to limit the number of UBR connections by assigning a bandwidth or weight to each UBR or VBR with a PCR connection

CAC and ATM Bulk Configuration

You cannot configure CAC on an ATM interface on which you have created a bulk-configured virtual circuit (VC) range for use by a dynamic ATM 1483 subinterface. Conversely, you cannot create a bulk-configured VC range on an ATM interface on which you have configured CAC. The router rejects these configurations, which causes them to fail.

If you are upgrading to the current JunosE Software release from a lower-numbered release, configurations that use CAC and bulk configuration on the same ATM interface continue to work. However, we recommend that you disable CAC on these ATM interfaces to ensure continued compatibility with future JunosE releases.

For information about how to use the **atm cac** command to configure CAC, see "Setting Optional Parameters" on page 23. For information about how to use the **atm bulk-config** command to create a bulk-configured VC range, see “Bulk Configuration of VC Ranges Overview” on page 629 in “Configuring Dynamic Interfaces Using Bulk Configuration” on page 619.

ILMI

ATM interfaces support the ATM Forum integrated local management interface (ILMI), versions 3.0, 3.1, and 4.0. An important feature of ILMI is the ability to poll or send keepalive messages across the UNI. ATM interfaces always respond to such messages, which are sent by an ATM peer device. Optionally, you can configure ATM major interfaces to generate keepalive messages, a process that enables a continuous ATM-layer connectivity verification; if the ATM peer stops responding to keepalive messages, the router disables the ATM interface.

The ATM interface is not reenabled until the keepalive message’s responses are received (or until the keepalive feature is disabled on the ATM port). To enable ILMI and control the generation of keepalive messages, use the **atm ilmi-enable** and **atm ilmi-keepalive** commands.

VPI/VCI Address Ranges

The VPI/VCI address ranges allowed on ATM interfaces are module dependent. Certain modules on ERX14xx models, ERX7xx models, or the Juniper Networks ERX310 Broadband Services Router have a fixed allocation scheme, whereas others have a configurable allocation scheme. In the configurable allocation scheme, a bit range is shared across the VPI and VCI fields.

For example, if an ATM interface has a bit range of 18, and 4 bits are allocated to the VPI space, then 14 bits are left for the VCI space. The resulting numeric range is 0 to 2n-1, where *n* is the number of bits for each space. Completing the example, if 4 bits were allocated for the VPI space and 14 for the VCI space, the configurable range would be 0 to 15 for VPI and 0 to 16,383 for the VCI space. To configure the bit range, use “atm vc-per-vp” on page 30 .

See “Supported Features” on page 11 for details on how various line module and I/O modules support configurable VPI/VCI address ranges.

NOTE: **The E120 and E320 routers support the full VPI/VCI address range; therefore, it has a fixed allocation scheme.**

VP Tunneling

Virtual path (VP) tunneling enables traffic shaping to be applied to the aggregation of all VCs within a single VP. Thus, VP tunnels can be used to ensure that the total traffic transmitted on a VP does not exceed the specified PCR. VP tunneling uses a round-robin algorithm to guarantee fairness among all of the VCs within the tunnel.

You can change the PCR associated with a tunnel even when VCs have already been configured on the tunnel. The individual VCs within a tunnel must be specified as UBR VCs. In other words, they may not have their own traffic-shaping parameters.

The level of support for VP tunneling is dependent on the specific I/O module. See “Supported Features” on page 11 for details.

Platform Considerations

You can configure ATM interfaces on the following Juniper Networks E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support ATM interfaces on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support ATM.

For information about the modules that support ATM interfaces on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support MLPPP.

Interface Specifiers

The configuration task examples in this chapter use the *slot/port*[.*subinterface*] format to specify an ATM interface. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter

1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E120 or E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

References

For more information about ATM interfaces, consult the following resources:

- ATM Forum—ATM User-Network Interface Specification, Version 3.0 (September 1993)
- ATM Forum—ATM User-Network Interface Specification, Version 3.1 (September 1994)
- ATM Forum—Integrated Local Management Interface (ILMI) Specifications, Versions 3.0, 3.1, and 4.0 (September 1996)
- ATM Forum—Traffic Management Specification, Version 4.0 (April 1996)
- ITU-T Draft Recommendation I.363 (AAL5 support) (January 1993)
- RFC 2390—Inverse Address Resolution Protocol (September 1998)
- RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999) (RFC 2684 obsoletes RFC 1483)
- ITU-T Recommendation I.610—B-ISDN Operation and Maintenance Principles and Functions (February 1999)
- Encapsulation Methods for Transport of ATM Over MPLS Networks—draft-ietf-pwe3-atm-encap-07.txt (April 2005 expiration)
- JunosE Release Notes, Appendix A, System Maximums—See the Release Notes corresponding to your software release for information about maximum values

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Web site at http://www.ietf.org for the latest drafts.

Supported Features

This section describes ATM feature support on E Series modules.

For more information about the physical layer characteristics of the modules described in this section, including the numbering schemes, see the *JunosE Physical Layer Configuration Guide*.

Module Capabilities

The level of support for certain ATM capabilities varies depending on the module. Table 5 on page 12 lists the specific differences in the capabilities of the modules.

The number of VP tunnels varies with the number of ports in the associated line module. For information about the maximum number of ATM VP tunnels supported per port for all line modules, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

NOTE: **Support for the OC3 (dual port) line module has been deprecated.**

Table 5: ATM Capabilities on Line Modules and I/O Modules

Line Module	I/O Module or IOA	Number of VP Tunnels	VPI/VCI Address Range	Configurable Bit Range	Number of VCs on Each Port	ATM Circuit Traffic Management Types	VP Tunnel Traffic Management Types
OCx/STMx ATM	OC3-4 I/O 4xDS3 ATM I/O	1024	Configurable	20	8000 active 16,000 configured	CBR, UBR, UBR with PCR, VBR-NRT, VBR-RT	CBR, VBR-NRT
OCx/STMx ATM	OC12/STM4 I/O	256	Configurable	20	8000 active 16,000 configured	CBR, UBR, UBR with PCR, VBR-NRT, VBR-RT	CBR, VBR-NRT
OC3/STM1 GE/FE	OC3-2 GE APS I/O	1024	Configurable	20	8000 active 16,000 configured	CBR, UBR, UBR with PCR, VBR-NRT, VBR-RT	CBR, VBR-NRT
ES2 4G LM	ES2-S1 OC3-8 STM1 ATM IOA	1 IOA per slot: 2048 2 IOAs per slot: 4096	Fixed VPI: 0–255 VCI: 0–65535	–	8000 active 16,000 configured	CBR, UBR, UBR with PCR, VBR-NRT, VBR-RT	CBR, VBR-NRT
ES2 4G LM	ES2-S1 OC12-2 STM4 ATM IOA	1 IOA per slot: 512 2 IOAs per slot: 1024	Fixed VPI: 0–255 VCI: 0–65535	–	8000 active 16,000 configured	CBR, UBR, UBR with PCR, VBR-NRT, VBR-RT	CBR, VBR-NRT

Virtual Channel Support

The number of virtual channels (VCs) that the router supports on each port varies depending on the E Series router and module you are using. For information about the maximum number of ATM VCs supported per chassis, per module, and per port, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

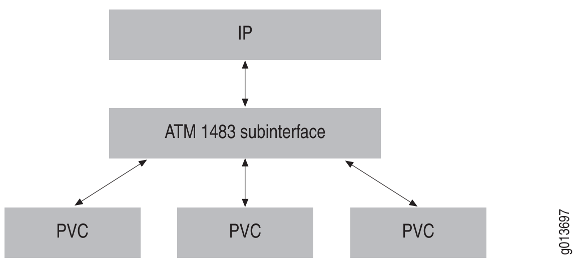
ATM NBMA

The software supports nonbroadcast multiaccess (NBMA) networks, which interconnect more than two routers and have no broadcast capabilities.

NOTE: **The E120 and E320 routers do not support ATM NBMA in the current release.**

An ATM NBMA network can be thought of as an interface stack with a single IP interface at the top, eventually fanning out to multiple independent PVCs. See Figure 2 on page 13.

Figure 2: NBMA Interface Stack



Unlike standard point-to-point ATM interfaces and broadcast-oriented Ethernet interfaces, NBMA interfaces form a point-to-multipoint connection. For example, you can use NBMA to connect a router to multiple stations.

An NBMA interface consists of a single ATM 1483 subinterface that has two or more VCs. You can add circuits to an existing ATM 1483 subinterface at any time. New circuits become usable after they have valid ARP table entries. NBMA circuits support only IP directly over ATM 1483.

The software restricts NBMA interfaces so that all circuits reside on the same physical interface. An NBMA interface can use as many PVCs as are available on a physical port.

ARP Table

To maintain the Address Resolution Protocol (ARP) table, you can use either static mapping via the CLI or Inverse ARP (InARP). InARP provides a way of determining the IP address of the device at the far end of a circuit. For NBMA interfaces, InARP enables automatic creation of ARP table entries for each circuit on the interface.

You must enable InARP when you create a PVC by using the **atm pvc** command. After you configure InARP, a protocol mapping between an ATM PVC and a network address is learned dynamically as a result of the exchange of InARP packets.

Static Map Versus Inverse ARP

If the device at the other end of a circuit does not support InARP, static mapping is required for that circuit. One of these two methods must be used to generate an ARP table entry for each circuit of the NBMA interface.

InARP and static mapping are complementary within an NBMA subinterface, but are not compatible with regard to individual circuits. If InARP is configured on a circuit, the corresponding virtual circuit descriptor (VCD) cannot be present in a static map applied to that interface.

Aging

ARP table entries, with the exception of those declared static, are aged out based on an aging interval defined on a subinterface basis. For the purposes of aging, entries produced via a static map are treated as static ARP table entries. InARP-generated entries are also treated as static; however, the InARP state machine automatically removes entries that cannot be successfully refreshed after three successive failed InARP requests.

Removing Circuits

If a circuit is removed, it is also removed from the ARP table, but not from the static map. If the circuit is reconfigured, a new ARP table entry is generated from the existing map entry. If the circuit uses InARP, the ARP table entry is immediately removed on removal of the circuit.

If a subinterface is removed, all associated circuits and their associated ARP table entries are removed.

Operations, Administration, and Management of ATM Interfaces

ATM interfaces support the OAM standards of the ITU, per recommendation I.610. OAM provides VC/VP integrity and fault and performance management. The E Series router supports F4 and F5 ATM OAM fault management, loopback, and continuity check (CC) cells. These cells perform fault detection and notification, loopback testing, and link integrity.

ATM uses F4 and F5 cell flows as follows:

- F4—Used in VPs
- F5—Used in VCs

ATM interfaces always generate and validate CRC-10 checksums on OAM cells.

For information about configuring OAM on the router, see the following sections:

- “Configuring OAM” on page 31
- “Configuring F5 OAM for Data PVCs” on page 49

End-to-End and Segment Endpoints

An ATM connection consists of a group of points. This OAM implementation provides management for the following points:

- Connection endpoint—The end of a VC/VP connection where the ATM cells are terminated
- Segment endpoint—The end of a connection segment

Fault Management

ATM uses two types of fault management cells to convey defect information to the endpoints of a VP/VC:

- Alarm indication signal (AIS) cells, which are used to indicate a fault to the downstream endpoint. AIS cells contain defect type and defect location fields, which optionally convey information about the type of defect detected and the location of the defect.
- Remote defect indication (RDI) cells, which are received from the remote endpoint of the VP/VC and indicate an interruption in the cell transfer capability of the VP/VC.

Connecting points in the VP/VC that detect a fault send AIS cells in the downstream direction to the endpoint of the VP/VC. Upon receipt of AIS cells, the downstream endpoint generates RDI cells in the upstream direction to alert all connecting points and the remote endpoint of an interruption in the cell transfer capability of the VP/VC.

If fault management detects a failure condition (because of arrival of AIS or RDI cells), the router disables the corresponding VC until the fault condition is no longer detected.

How the ATM Interface Handles AIS Cells

Nodes that detect a failure send AIS cells to the downstream endpoint. Because the ATM interface is an endpoint and there is no downstream neighbor to an ATM endpoint, the ATM interface never generates AIS cells. The ATM interface responds to the receipt of AIS cells as follows:

1. When an ATM interface receives a configurable number of F4 or F5 AIS cells, it enters the AIS state.
2. While in the AIS state, the ATM interface sends F4 or F5 RDI cells to the remote endpoint. It sends the RDI cells at the rate of one cell per second for as long as the AIS condition exists.

   For all RDI cells sent, the defect type and defect location fields contain the values from the received AIS cells.
3. RDI cell generation stops when one of the following conditions occurs:
   - The interface receives an F4 or F5 loopback cell or an F4 or F5 CC cell.
   - The interface does not receive an AIS cell for a configurable time period.
   - The OAM VC status field of "show atm vc atm" on page 90 shows that the circuit is in AIS state.

How the ATM Interface Handles RDI Cells

RDI cells received from the remote endpoint of the VP/VC indicate an interruption in the cell transfer capability of the VP/VC. For example, the remote endpoint of a VC receives an F5 AIS cell, enters the AIS state, and transmits F5 RDI cells for the duration of the AIS condition. On receipt of a configurable number of F4 or F5 RDI cells, the ATM interface declares an RDI state but does not generate OAM fault management cells in response to the condition. The ATM interface leaves the RDI condition when no RDI cells have been received for a configurable time period.

The OAM VC status field of "show atm vc atm" on page 90 shows whether the circuit is in RDI state.

Continuity Verification

CC cells provide continual monitoring of a connection on a segment or end-to-end basis. To verify the integrity of the link, you can set up a VP or VC to regularly send or receive CC cells at either the segment level or at the end-to-end level.

The CC cell source generates the CC cells, and the sink receives and processes the cells. You can set up a VP or VC as the source, the sink, or both the source and the sink. If you enable a VP or VC as a CC cell source, it generates CC cells. The VP or VC counts CC cells whether or not CC cell flow is enabled. You can enable CC cells only on data circuits, not on control circuits, such as ILMI or signaling circuits.

Activation and Deactivation Cells

To enable and disable CC cell flows, ATM OAM uses activation and deactivation cells:

- To enable a CC cell flow, the router sends activation OAM cells to the peer. The peer replies with a confirmation or denial. If the CC sink point is not activated, all received CC cells are dropped. (See "Activating CC Cell Flow" on page 16 for more details.)
- To disable a CC cell flow, the router sends deactivation OAM cells to the peer. The peer replies with a confirmation or denial.

Activating CC Cell Flow

When the router sends a CC activation cell to the peer, one of the following occurs:

- If the router receives a positive response (Activation Confirmed), the VC or VP goes to CC active state, and CC is enabled on the VC or VP.
- If the router receives a negative response (Activation Req. Denied), the VC or VP goes to CC failed state, and CC is not enabled on the VC or VP.
- If the router does not receive a response within 5 seconds, it sends another activation cell. This process is repeated three times. If the router does not receive a response, it stops the activation process.

If the VC or VP is the source point, CC cell generation starts as soon as the router sends the activation request to the peer. CC cell generation stops if the CC fails, when the maximum number of retries is reached, or when the deactivation process is complete.

Deactivating CC Cell Flow

The process of sending a deactivation request is the same as for activation cells except that deactivation cells are sent instead.

Also, the **atm oam flush** command causes the router to send a deactivation request to the peer and suspend all CC operations. Therefore, we recommend that you disable CC cell generation and transmission on all VCs before issuing **atm oam flush**.

After CC Cell Flow Is Enabled

If the VC or VP is set up as the source point, the ATM interface sends one CC cell per second. CC cell generation stops if one of the following conditions occur:

- The ATM interface goes down.
- You disable OAM CC on the circuit by using the **atm pvc** command.
- The peer deactivates the OAM CC cell flow.
- You disable OAM cell reception and transmission on the ATM interface by using the **atm oam flush** command.

If the VP is set up as a CC sink point and no CC cell is received for 4 seconds, the VP goes to AIS state and sends one RDI cell per second.

To view the current state of the activation or deactivation process, including statistics, use the **show atm oam** command for VPs and the **show atm vc atm interface** command for VCs.

Loopback

You can use loopback cells to verify connectivity between VP/VC endpoints, as well as segment endpoints within the VP/VC. You can use these tests to perform fault isolation over the VP/VC.

The ATM interface supports VC integrity, which generates F5 end-to-end loopback cells. It also supports ATM ping, which generates F4 and F5 segment and end-to-end loopback cells to test the reachability of an endpoint or a segment endpoint.

VC Integrity

VC integrity is used to monitor the operational status of an individual VC. VC integrity provides continuous ATM VC-layer connectivity verification by periodically sending F5 end-to-end loopback cells on individual PVCs to verify end-to-end connectivity. You can set the frequency with which loopback cells are transmitted for an individual VC.

If VC integrity is enabled, the peer ATM host must respond to the router's loopback cells, or the circuit will be disabled. The ATM interface does not reenable the circuit until it receives loopback responses or until local VC integrity is disabled.

You can set the following VC integrity parameters for an individual VC with the **oam retry** command. For more information, see "oam retry" on page 51.

- The retry frequency with which loopback cells are transmitted when the router verifies the down status of the circuit; that is, when the peer ATM host does not respond to a loopback cell
- The retry frequency with which loopback cells are transmitted when the router verifies the up status of the circuit; that is, when the ATM host resumes responding to a loopback cell
- The number of successive loopback cell responses missed before the router determines that the circuit is down
- The number of successive loopback responses received before the router determines that the circuit is up

VC integrity is a best-effort mechanism that tries to adhere to the loopback cell transmission frequency and retry frequency values configured for each VC without consuming excessive processing time on the line module. When you configure VC integrity for a large number of circuits on the line module, delays in transmitting OAM loopback cells might occur so new subscribers can connect and to maintain existing subscriber connections.

To set up the ATM interface to transmit F5 end-to-end loopback cells over a VC, use the **oam** keyword and an optional frequency with the **atm pvc** command. To send F5 segment loopback cells, use the ATM ping mechanism, described in "ATM Ping" on page 18.

F5 loopback receive and transmit statistics are available with "show atm vc atm" on page 90 .

F4 OAM Cells

You can generate F4 loopback cells using the **atm oam** command or the ATM ping mechanism. F4 loopback receive and transmit statistics are available with the **show atm oam** command and include statistics on incoming and outgoing F4 end-to-end and segment loopback cells.

ATM Ping

With ATM ping you can verify whether a connection endpoint or segment point can be reached on a VC or VP. ATM ping uses F4 and F5 loopback cells and is supported only for data circuits and not control circuits (ILMI, signaling circuits). To generate:

- F5 segment loopback cells or end-to-end loopback cells, issue the **ping atm** command on a VC.
- F4 segment loopback cells or end-to-end loopback cells, issue the **ping atm** command on a VP.

You can specify the number of loopback cells that are sent, the location ID, and the timer value. After the interface sends the loopback cells, the timer is started and the interface waits for a response. On receiving the loopback response (or when the timer expires) the ATM interface sends the next cell. This operation is repeated for the number of cells specified.

Because F4 and F5 are OAM cells, disabling receipt and transmission of OAM cells on the ATM interface (by using the **atm oam flush** command) stops all outstanding ping operations on the ATM interface. You need to manually restart the ping operation after you enable receipt and transmission of OAM cells for the interface.

How the ATM Interface Handles Loopback Cells Received

The ATM interface responds to received F4 and F5 loopback cells as indicated in Table 6 on page 19.

Table 6: Handling of F4 and F5 Loopback Cells Received

Loopback Cell Received	ATM Interface Response
F4 and F5 end-to-end loopback cells and segment loopback cells with the loopback location field set to all 1s (ones) and the loopback indication set.	Clears the loopback indication (sets it to all zeros) and loops back the received cell.
F4 and F5 segment loopback cells with the loopback location field set to all 0s (zeros) and the loopback indication set.	Resets the loopback indication and the location ID to all 1s (ones) and loops back the received cells.
F4 and F5 end-to-end loopback cells and segment loopback cells with the loopback location field set to the loopback location ID of the ATM interface and the loopback indication set.	Clears the loopback indication and loops back the received cell without resetting the location ID.
F5 end-to-end loopback cells with the loopback location field set to a value other than all 1s and the loopback location ID of the ATM interface.	Discards the cell.
F5 segment loopback cells with the loopback location field set to other than all 1s (ones), set to all 0s (zeros), or set to the loopback location ID of the ATM interface.	Discards the cell.

Automatic Disabling of F5 OAM Services

The router automatically disables all F5 OAM fault management and VC integrity services configured on a VC when you change the administrative status of the corresponding ATM interface, ATM AAL5 interface, or ATM 1483 subinterface from enabled to disabled.

To set the administrative status of an interface to disabled, use the **atm shutdown** command (for an ATM interface), the **atm aal5 shutdown** command (for an ATM AAL5 interface), or the **atm atm1483 shutdown** command (for an ATM 1483 subinterface). You can also use the **shutdown** command to disable the interface.

When F5 OAM is disabled, the OAM VC status field in the **show atm vc atm** command display indicates that the VC is not managed. The VC does not receive or transmit F5 OAM cells while F5 OAM is disabled. For examples of the **show atm vc atm** command display, see "show atm vc atm" on page 90.

When the corresponding ATM interface, ATM AAL5 interface, or ATM 1483 subinterface is reenabled, the router automatically restores F5 OAM services on the associated VCs.

NOTE: **If you administratively issue the shutdown command on an ATM major interface in which the ATM PVC is configured over a dynamic ATM 1483 subinterface column, or if subscribers on the ATM 1483 subinterfaces log out, the ATM PVC is deleted immediately.**

Rate Limiting for F5 OAM Cells

The router implements rate limiting for ATM F5 OAM cells to protect the corresponding ATM interface from denial-of-service (DoS) attacks. The interface discards control packets when the rate of control packets received exceeds the rate limit for ATM interfaces.

An ATM interface has a rate limit control that is non-configurable and always in effect; the rate limit is the same for all ATM interfaces. In addition, each ATM VC maintains its own state and statistics counters for tracking the rate. The rate limit for ATM OAM cells is approximately 5 packets per second.

For an ATM VC, the router increments the InOamCellDiscards statistics counter in the **show atm vc atm** command display to track the number of OAM cells received on this circuit that were discarded. The InOamCellDiscards counter operates on a per-circuit basis, not on a per-interface basis.

For examples of the **show atm vc atm** command display, see "show atm vc atm" on page 90.

Before You Configure ATM

Before you configure an ATM interface, verify that you have installed the physical module (such as an OC3 module) correctly. For more information about preconfiguration procedures, see the *ERX Hardware Guide* or the *E120 and E320 Hardware Guide*.

Also have the following information available:

- Interface specifiers for the ATM interfaces that you want to create

  For more information about specifying ATM interfaces and subinterfaces on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.
- Virtual path and channel numbers for each virtual circuit you want to create
- IP addresses and subnet mask assignments for IP interfaces

You can configure the following types of dynamic interfaces over ATM:

- IP over static ATM 1483 (IPoA)
- IP over PPP over static ATM 1483
- IP over PPPoE over static ATM 1483

- IP over bridged Ethernet over static ATM 1483
- IP over MLPPP over static ATM 1483
- ATM 1483 over static ATM AAL5 over ATM

For information about creating these dynamic configurations, see "Configuring Upper-Layer Dynamic Interfaces" on page 519.

Configuration Tasks

The following sections describe how to perform these ATM configuration tasks:

- Creating a Basic Configuration on page 21
- Setting Optional Parameters on page 23
- Configuring OAM on page 31
- Configuring an NBMA Interface on page 37
- Creating an NBMA Static Map on page 38
- Assigning Descriptions to Interfaces on page 40
- Sending Interface Descriptions to AAA on page 41
- Configuring Individual ATM PVC Parameters on page 43
- Configuring ATM VC Classes on page 52
- Configuring Dynamic ATM 1483 Subinterfaces on page 66

Creating a Basic Configuration

To configure ATM, perform the following tasks. (Figure 3 on page 22 shows the relationship of Steps 1 through 3.)

1. Configure an ATM physical interface.

   host1(config)#interface atm 0/1

2. Configure an ATM 1483 subinterface.

   host1(config-if)#interface atm 0/1.20

3. Configure a PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type.

   host1(config-subif)#atm pvc 10 15 22 aal5snap

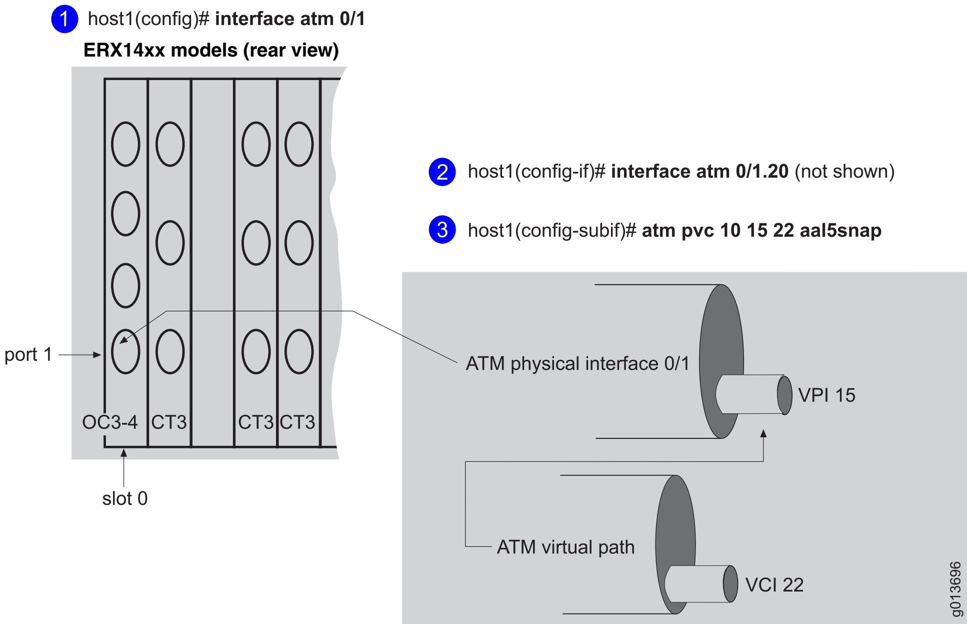
4. Assign an IP address and subnet mask to the PVC.

   host1(config-subif)#ip address 192.32.10.20 255.255.255.0

5. (Optional) Verify your configuration using the appropriate **show** commands.

   host1#show atm interface atm 0/1
   host1#show atm vc atm 0/1 10
   host1#show atm subinterface atm 0/1.20

Figure 3: Configuring an ATM Interface, Subinterface, and PVC



atm pvc

- Use to configure a PVC on an ATM interface.
- Specify one of the following encapsulation types:
  - **aal5snap**—Specifies an LLC encapsulated circuit; LLC/SNAP header precedes the protocol datagram.
  - **aal5mux ip**—Specifies a VC-based multiplexed circuit. This option is used for IP only.
  - **aal5autoconfig**—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed) for dynamic interfaces. See "Configuring Upper-Layer Dynamic Interfaces" on page 519, for more explanation.
  - **ilmi**—Defines the PVC for ILMI keepalive messages. You can set this option only on major interfaces. After the PVC is set up for ILMI, use "atm ilmi-keepalive" on page 28 to cause the router to generate ILMI keepalive messages on the interface.
- You can optionally set the *peak*, *average*, and *burst* sizes. To use VBR-RT or VBR-NRT as the service type, you must specify each of these options.
- The default service type is UBR. To set a different service type, specify one of the following keywords:
  - **rt**—Selects VBR-RT as the service type. You can select **rt** only if you set the *peak*, *average*, and *burst* parameters.
  - **cbr**—Selects CBR as the service type. You must set the CBR rate in Kbps.
- To enable VC integrity and generation of OAM F5 loopback cells on this circuit, use the **oam** keyword.

- Example

  host1(config-if)#atm pvc 6 0 11 aal5snap cbr 10000

- Use the **no** version to remove the specified PVC.
- See *atm pvc*.

interface atm

- Use to configure an ATM interface or subinterface type.
- To specify an ATM interface for ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*.[*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module; on the OC3-2 GE APS I/O module, you can specify ATM interfaces only in ports 0 and 1; port 2 is reserved for a Gigabit Ethernet interface
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM interface for the E120 or E320 router, use the *slot/adapter/port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router)
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- Specify the type of interface or subinterface: **point-to-point** or **multipoint**. Point-to-point is the default.
- Examples

  host1(config-if)#interface atm 0/1.20
  host1(config-if)#interface atm 0/0/4.20

- Use the **no** version to remove the subinterface or the logical interface.
- See *interface atm*.

Setting Optional Parameters

You can also set the following parameters:

- Set the administrative state of an ATM AAL5 interface to disabled.

  host1(config-if)#atm aal5 shutdown

- Enable CAC on the interface.

  host1(config-if)#**atm cac 3000000 ubr 3000**

- Configure the clock source.

  host1(config-if)#**atm clock internal**

- Configure framing on a T3/E3 physical interface.

  host1(config-if)#**atm framing g751adm**

- Enable ILMI on the interface.

  host1(config-if)#**atm ilmi-enable**

- Set the ILMI keepalive timer.

  host1(config-if)#**atm ilmi-keepalive 5**

- Specify the cable length (line build-out) for the ATM interface.

  host1(config-if)#**atm lbo long**

- Set the administrative state of the ATM interface to disabled.

  host1(config-if)#**atm shutdown**

- Configure SNMP link status traps on the interface.

  host1(config-if)#**atm snmp trap link-status**
  host1(config-if)#**atm aal5 snmp trap link-status**

- Set the operational mode of the physical interface to SDH STM1.

  host1(config-if)#**atm sonet stm-1**

- Configure the UNI version of ILMI using one of the following methods:

  - Enable auto configuration of ILMI.

    host1(config-if)#**atm auto-configuration**

  - Set the UNI version that the router uses when ILMI link autodetermination is unsuccessful or ILMI is disabled.

    host1(config-if)#**atm uni-version 4.0**

- Configure the number of virtual circuits for each virtual path.

  host1(config-if)#**atm vc-per-vp 128**

- Configure a virtual path tunnel and its traffic parameters.

  host1(config-if)#**atm vp-tunnel 2 128**

- Enable scrambling of the ATM cell payload on a T3 or an E3 interface.

  host1(config-if)#**ds3-scramble**

- Set the time interval at which the router records bit and packet rates.

  host1(config-if)#**load-interval 90**

- Place the interface into loopback mode for router-to-router testing.

  host1(config-if)#**loopback diagnostic**

- Disable an interface.

  host1(config-if)#**shutdown**

Optional Tasks on ATM 1483 Subinterfaces

You can perform the following optional tasks on ATM 1483 subinterfaces:

- Set the MTU.

  host1(config-subif)#**atm atm1483 mtu 7800**

- Configure SNMP link status traps.

  host1(config-subif)#**atm atm1483 snmp trap link-status**

- Set the administrative state of an ATM 1483 subinterface to disabled.

  host1(config-subif)#**atm atm1483 shutdown**

- Configure an advisory receive speed.

  host1(config-subif)#**atm atm1483 advisory-rx-speed 2000**

atm aal5 shutdown

- Use to set an ATM AAL5 interface administrative state to disabled.
- When you set the administrative state of the ATM AAL5 interface to disabled, the router automatically disables all F5 OAM services configured on the associated VC, and prevents the VC from receiving or transmitting F5 OAM cells.
- Example

  host1(config-if)#**atm aal5 shutdown**

- Use the **no** version to enable a disabled interface.
- See *atm aal5 shutdown*.

atm aal5 snmp trap link-status

- Use to enable SNMP link status traps on the AAL5 layer interface.
- Example

  host1(config-if)#**atm aal5 snmp trap link-status**

- Use the **no** version to disable the traps.
- See *atm aal5 snmp trap link-status*.

atm atm1483 advisory-rx-speed

- Use to set an advisory receive speed for an ATM 1483 subinterface. This setting has no effect on data forwarding. You can use it to indicate the speed of the client interface. When traffic is tunneled with L2TP, the advisory receive speed is sent from the LAC to the LNS. See *LAC Configuration Prerequisites* for additional information about the advisory receive speed.

NOTE: If you specify an advisory receive speed greater than 4294967 kbps, the speed is not accurately represented in the L2TP AVP, which is in bits per second (bps).

- The range is 0–2147483647 kbps.
- Example

  host1(config-subif)#**atm atm1483 advisory-rx-speed 2000**

- Use the **no** version to restore the default behavior—the RX speed is not sent to the LNS.
- See *atm atm1483 advisory-rx-speed*.

atm atm1483 mtu

- Use to set the MTU size for an ATM 1483 subinterface.
- The range is 256–9180.
- Example

  host1(config-subif)#**atm atm1483 mtu 7800**

- Use the **no** version to restore the default size of 9180.
- See *atm atm1483 mtu*.

atm atm1483 shutdown

- Use to set an ATM 1483 subinterface administrative state to disabled.
- When you set the administrative state of the ATM 1483 subinterface to disabled, the router automatically disables all F5 OAM services configured on the associated VC, and prevents the VC from receiving or transmitting F5 OAM cells.
- Example

  host1(config-subif)#**atm atm1483 shutdown**

- Use the **no** version to enable a disabled subinterface.
- See *atm atm1483 shutdown*.

atm atm1483 snmp trap link-status

- Use to enable SNMP link status traps on an ATM 1483 layer subinterface.
- Example

  host1(config-subif)#**atm atm1483 snmp trap link-status**

- Use the **no** version to disable the traps.
- See *atm atm1483 snmp trap link-status*.

atm auto-configuration

- Use to enable autoconfiguration of ILMI. Entering the **atm auto-configuration** command overrides any previous configuration of the **atm uni-version** command.
- Autoconfiguration is enabled by default.
- Example

  host1(config-if)#**atm auto-configuration**

- Use the **no** version to disable autoconfiguration and set the ILMI parameters to the UNI version configured using the **atm uni-version** command, which has a default value of UNI 4.0.
- See *atm auto-configuration*.

atm cac

- Use to enable CAC on the interface. You can set a subscription limit, so you can oversubscribe the port, and the UBR weight, so you can limit the number of UBR connections.
- You cannot configure CAC on an ATM interface on which you have created a bulk-configured VC range for use by a dynamic ATM 1483 subinterface. Conversely, you cannot create a bulk-configured VC range on an ATM interface on which you have configured CAC. For information about creating bulk-configured VC ranges, see "Bulk Configuration of VC Ranges Overview" on page 629 in "Configuring Dynamic Interfaces Using Bulk Configuration" on page 619.
- Example

  host1(config-if)#**atm cac 3000000 ubr 3000**

- Use the **no** version to disable CAC on the interface.
- See *atm cac*.

atm clock internal

- Use to cause the ATM interface to generate the transmit clock internally.
- You must specify one of the following:
  - **module**—Internal clock is from the line module (the default)
  - **chassis**—Internal clock is from the configured system clock
- Example

  host1(config-if)#**atm clock internal**

- Use the **no** version to cause ATM interfaces to recover the clock from the received signal.
- See *atm clock internal*.

atm framing

- Use to configure T3 or E3 framing on an ATM interface.
- Specify one of the following framing types for a T3 (DS3) interface:
  - **cbitadm**—c-bit with ATM direct mapping
  - **cbitplcp**—c-bit with PLCP framing (default)
  - **m23adm**—M23 ATM direct mapping
  - **m23plcp**—M23 with PLCP framing
- Specify one of the following framing types for an E3 interface:
  - **g832adm**—G.832 ATM direct mapping
  - **g751adm**—G.751 ATM direct mapping
  - **g751plcp**—G.751 PLCP mapping (default)
- Example

  host1(config-if)#**atm framing g751adm**
- Use the **no** version to return framing to the default:
  - For a T3 interface, the default is **cbitplcp**
  - For an E3 interface, the default is **g751plcp**
- See *atm framing*.

atm ilmi-enable

- Use to enable ILMI on the interface.
- Example

  host1(config-if)#**atm ilmi-enable**
- Use the **no** version to disable ILMI on the interface.
- See *atm ilmi-enable*.

atm ilmi-keepalive

- Use to generate ILMI keepalive messages. This value sets the time interval in seconds between poll PDU transmissions if no sequence data PDUs are pending.
- Example

  host1(config-if)#**atm ilmi-keepalive 5**
- Use the **no** version to disable the generation of keepalive messages.
- See *atm ilmi-keepalive*.

atm lbo

- Use to specify the cable length (line build-out) for the ATM T3 or E3 interface. The length of cable determines power requirements.
- Specify one of the following keywords:
  - long—A cable length in the range 0–225 feet
  - short—A cable length in the range 226–450 feet (the default)
- Example

  host1(config-if)#**atm lbo long**
- Use the **no** version to restore the default value, **short**.
- See *atm lbo*.

atm shutdown

- Use to set an ATM interface administrative state to disabled.
- When you set the administrative state of the ATM interface to disabled, the router automatically disables all F5 OAM services configured on the associated VC, and prevents the VC from receiving or transmitting F5 OAM cells.
- Example

  host1(config-if)#**atm shutdown**
- Use the **no** version to enable a disabled interface.
- See *atm shutdown*.

atm snmp trap link-status

- Use to enable SNMP link status traps on the ATM layer interface.
- Example

  host1(config-if)#**atm snmp trap link-status**
- Use the **no** version to disable the traps.
- See *atm snmp trap link-status*.

atm sonet stm-1

- Use to set the mode of operation on the physical interface to Synchronous Digital Hierarchy (SDH) Synchronous Transport Mode (STM).

  host1(config-if)#**atm sonet stm-1**
- Use the **no** version to restore the default value, SONET STS-3c operation.
- See *atm sonet stm-1*.

atm uni-version

- Use to specify the UNI version for the interface to use.
- Valid values are 3.0, 3.1, or 4.0.
- Example

host1(config-if)#**atm uni-version 4.0**

- There is no **no** version.
- See *atm uni-version*.

atm vc-per-vp

- Use to configure the number of VCs for each VP. The router does not execute this command when any VCs are open on the interface.
- VCs and VP tunnels must not exist when you issue this command. If they do, you must delete the VC and VP tunnel configuration before you issue this command.
- The specified value must be a power of 2, or an error message is returned.
- The minimum number of VCs per VP is 4096 for OCx/STMx ATM line modules. If you enter a value that is below the minimum, the router uses the minimum value.
- The E120 and the E320 routers support the entire VPI/VCI range; therefore, it does not support this command.
- Example

  host1(config-if)#**atm vc-per-vp 128**

- Use the **no** version to restore the default value.
- See *atm vc-per-vp*.

atm vp-tunnel

- Use to define a VP tunnel and configure the rate of traffic flow within the tunnel.
- You specify a tunnel rate in Kbps. All circuits in the VP are restricted to the rate that you set.
- The tunnel rate can be a value in the range 0–4294967295, when you specify the rate of traffic flow without the constant bit rate (CBR) service category, and can be a value in the range 1–4294967295, when you specify the rate of traffic flow with the CBR service class. Because the CBR service category guarantees a fixed amount of bandwidth to be allotted to the client, an error message is displayed if you configure a value of 0 for the tunnel rate for CBR traffic flows.
- If any virtual circuits are open within the VPI before the tunnel is created, the router does not execute this command.
- For more information about configuring a shapeless VP tunnel for QoS, see *ATM Integrated Scheduler Overview*.
- Example

  host1(config-if)#**atm vp-tunnel 2 128**

- Use the **no** version to remove the VP tunnel. When circuits are open within the tunnel, the router does not remove the tunnel.
- See *atm vp-tunnel*.

ds3-scramble

e3-scramble

- Use to scramble the ATM cell payload on a T3 or an E3 interface. DS3 (T3) and E3 scrambling assists clock recovery on the receiving end of the interface.
- Example

  host1(config-if)#**ds3-scramble**
- Use the **no** version to disable scrambling.
- See *ds3-scramble*.
- See *e3-scramble*.

load-interval

- Use to set the time interval at which the router calculates bit and packet rate counters for the ATM interface.
- You can choose a multiple of 30 seconds, in the range 30–300 seconds.
- Example

  host1(config-if)#**load-interval 90**
- Use the no version to return to the default setting, 300 seconds.
- See *load-interval*.

loopback

- Use to place the interface into loopback mode.
- Specify either:
  - **diagnostic**—Places the interface into internal loopback.
  - **line** —Places the interface into external loopback.
- Example

  host1(config-if)#**loopback diagnostic**
- Use the **no** version to remove any loopback.
- See *loopback*.

Configuring OAM

This section explains:

- Configuring F4 OAM on page 32
- Configuring F5 OAM on page 33
- Setting a Loopback Location ID on page 35
- Enabling OAM Flush on page 35
- Running ATM Ping on page 36

Configuring F4 OAM

The ATM interface does not support sending F4 segment loopback cells, but it does respond to F4 segment loopback cells that it receives.

F4 OAM flows need their own channel, and they are identified by the VCI on which they are sent or received. The following VCIs are reserved for F4 OAM flows for each virtual path, and you cannot open PVCs on them:

- VCI 3—For segment F4 flows
- VCI 4—For end-to-end F4 flows

NOTE: **You cannot enable both loopback cells and CC cells at the same time.**

To set up F4 OAM:

1. Enable F4 OAM on an interface or VP. The router enables F4 OAM at the interface level unless you specify a VPI. This example opens both segment and end-to-end F4 OAM circuits on VPI 10.

   host1(config-if)#**atm oam 10**

2. (Optional) Enable only segment or end-to-end loopback.

   host1(config-if)#**atm oam 10 seg-loopback**
   host1(config-if)#**atm oam 10 end-loopback**

3. (Optional) To cause the interface to generate end-to-end loopback cells in addition to receiving and responding to them, set the loopback timer.

   host1(config-if)#**atm oam 10 end-loopback loopback-timer 20**

4. (Optional) Enable CC cell flows.

   host1(config-if)#**atm oam 10 seg-loopback cc source**

atm oam

- Use to configure F4 OAM on an interface or circuit. F4 OAM is configured at the interface level unless you specify a VPI.
- To open F4 OAM on either a segment or end-to-end basis, use the following keywords:
  - **seg-loopback**—Enables F4 segment OAM
  - **end-loopback**—Enables F4 end-to-end OAM

  NOTE: **If you do not specify either segment or end-to-end loopback, the command applies to both end-to-end and segment F4 OAM circuits.**

- To configure CC cell flow on the PVC, use the following keywords:

- **both**—Enables the PVC as both the source and the sink endpoints.
- **sink**—Enables the PVC as the sink endpoint.
- **source**—Enables the PVC as the source endpoint.
- **loopback-timer**—When F4 OAM is enabled, the interface or circuit accepts and responds to F4 OAM cells. However, to generate F4 loopback cells, you must configure the loopback timer in the range 1–600 seconds. This timer represents the frequency with which F4 loopback cells are transmitted. You can set the loopback timer only for end-to-end loopback.

- Example 1—Opens both F4 end-to-end and segment OAM circuits for VPI 8

  host1(config-if)#**atm oam 8**

- Example 2—Opens the F4 end-to-end OAM circuit for VPI 10 and enables sending F4 end-to-end loopback cells on the circuit at a frequency of 20 seconds

  host1(config-if)#**atm oam 10 end-loopback loopback-timer 20**

- Example 3—Opens both F4 end-to-end and segment OAM circuits on all VPs on this interface

  host1(config-if)#**atm oam**

- Example 4—Opens F4 segment OAM circuits on all VPs on this interface

  host1(config-if)#**atm oam seg-loopback**

- Example 5—Opens F4 end-to-end loopback on VPI 12

  host1(config-if)#**atm oam 12 end-loopback**

- Example 6—Opens an F4 segment OAM circuit for VPI 8 and enables CC cell generation on the segment

  host1(config-if)#**atm oam 8 seg-loopback cc source**

- Use the **no** version to delete F4 OAM circuits. Using the options, you can delete all F4 OAM circuits on the interface, segment or end-to-end F4 OAM circuits, or F4 OAM circuits on a specific VPI.
  - Example 1—Deletes all F4 OAM circuits on the interface

    host1(config-if)#**no atm oam**

  - Example 2—Deletes all F4 segment OAM circuits on the interface

    host1(config-if)#**no atm oam segment**

  - Example 3—Deletes the F4 end-to-end OAM circuit on VPI 8

    host1(config-if)#**no atm oam 8 end-loopback**

- See *atm oam*.

Configuring F5 OAM

F5 OAM flows run over existing PVCs. The ATM interface does not support sending F5 segment loopback cells, but it does respond to F5 segment loopback cells that it receives.

NOTE: You cannot enable both loopback cells and CC cells at the same time.

To set up F5 OAM:

1. To enable VC integrity, which causes the ATM interface to periodically send F5 end-to-end loopback cells over a VC, use the **oam** keyword with the **atm pvc** command.

   You can include the frequency (in seconds) with which the router sends F5 end-to-end loopback cells.

   ```
   host1(config-if)#atm pvc 98 38 22 aal5snap oam 300
   ```

2. (Optional) To enable CC cell flows on a circuit, use the **cc** keyword with the **atm pvc** command. You can enable cell flows on a segment or end-to-end basis, and you can enable the PVC as a sink, source, or both a sink and a source.

   ```
   host1(config-if)#atm pvc 50 0 50 aal5snap oam cc end-to-end sink
   ```

When you issue the appropriate **shutdown** command to change the administrative status of the corresponding ATM interface, ATM AAL5 interface, or ATM 1483 subinterface from enabled to disabled, the router automatically disables all F5 OAM services configured on the associated VC. For more information, see "Automatic Disabling of F5 OAM Services" on page 19.

atm pvc

- Use the **atm pvc** command with the **oam** keyword to set up the PVC to periodically transmit F5 end-to-end loopback cells over a VC.
- You can use the **oam** keyword only if you specify one of the following encapsulation types:
  - **aal5snap**
  - **aal5mux ip**
  - **aal5autoconfig**
- The **oam** keyword is not available with the **aal5all**, **aal0**, or **ilmi**
- Optionally, you can configure the time interval in the range 1–600 seconds between transmissions of OAM F5 end-to-end loopback cells.
- Use the following keywords to enable and configure CC cell flows:
  - **end-to-end**—Opens an end-to-end CC cell flow
  - **segment**—Opens a segment CC cell flow
  - **sink**—Enables this VC as a sink point (cell receiver)
  - **source**—Enables this VC as the source point (cell generator)
  - **both**—Enables this VC as both a sink point and a source point
- Example 1—Enables F5 end-to-end loopback cells

host1(config-if)#atm pvc 20 20 20 aal5snap oam

- Example 2—Enables end-to-end CC cell flow and enables the PVC as the sink

  host1(config-if)#atm pvc 5 0 5 aal5autoconfig oam cc end-to-end sink

- Use the **no** version of the **atm pvc** command *without* the **oam** keyword to disable F5 OAM on the PVC and *without* the **cc** keyword to disable CC cell flows on the PVC. For example, the following command disables CC cell flow configured in Example 2.

  host1(config-if)#no atm pvc 5 0 5 aal5autoconfig

- See *atm pvc*.

Setting a Loopback Location ID

To enable other nodes to specifically send OAM loopback cells to the ATM interface, set the location ID of the ATM interface or circuit.

host1(config-if)#atm oam loopback-location 01090708

NOTE: Because the router is a connection endpoint, the default loopback location ID is all 1s (ones). This command enables you to specify a nondefault value.

atm oam loopback-location

- Use to set the location ID of the ATM interface. The location ID is a 4-octet field, and the default value is all 1s (ones).
  - You can set a specific value to identify this ATM interface as the intended recipient of OAM loopback cells.
  - You can also set the location ID to all 0s (zeros).

  For information about how the router handles loopback cells based on location ID, see Table 6 on page 19.

- Example

  host1(config-if)#atm oam loopback-location 01090708

- Use the **no** version to return the loopback location ID to the default value, all 1s (ones).
- See *atm oam loopback-location*.

Enabling OAM Flush

You can use the **atm oam flush** command to enable the OAM flush feature for an ATM interface. When OAM flush is enabled, the router ignores all OAM cells received on the interface, and stops sending OAM cells on this interface.

You can also issue the **atm oam flush** command with the optional **alarm-cells** keyword to cause the router to ignore only AIS and RDI cells and to accept all other OAM cells. This is useful in diagnostic situations when you might want to exclude alarm conditions.

NOTE: The OAM flush feature is supported on all E Series ATM module combinations.

atm oam flush

- Use to configure the router to ignore all OAM cells received on an ATM interface, and to stop sending OAM cells on this interface.
- To cause the router to ignore only AIS and RDI cells and to accept all other OAM cells, use the **alarm-cells** keyword.
- Example

  host1(config-if)#**atm oam flush**
- Use the **no** version to disable OAM flush on the interface.
- See *atm oam flush*.

Running ATM Ping

Keep in mind the following when you use ATM ping:

- Before you can run ATM ping, you need to add a PVC for the VPI and VCI over which you run the ping.
- Because ATM ping requires the receipt of OAM cells, make sure that the receipt and transmission of OAM cells is not disabled (using "atm oam flush" on page 36). To reenable the receipt and transmission of OAM cells, enter **no atm oam flush**.
- Disabling receipt of OAM cells during a ping operation stops all outstanding ping operations. You need to manually restart the ping operation after receipt of OAM cells for the interface is enabled.
- Because ATM ping is a dynamic (on-demand) operation, none of the configuration related to ATM ping is saved. To avoid acquiring excessive bandwidth for OAM, the number of outstanding ping operations on each interface is limited to 12.

ping atm interface atm

- Use to send loopback cells from an ATM interface or circuit.
- The VPI and VCI fields determine the type of loopback cells used for the ping operation. By default F5 end-to-end loopback OAM cells are used.
  - To send F4 segment loopback cells, set the VCI to 3.
  - To send F4 end-to-end loopback cells, set the VCI to 4.
- Use the **end-loopback** keyword to send the ping to the connection endpoint.
- Use the **seg-loopback** keyword to send the ping to the first segment point (for example, the next neighbor switch).

- Use the *destination* option to specify the value of the location ID included in the loopback cell. The location ID is a 16-octet field, and the destination portion is 4 octets. You can set the location ID to a specific destination or to 0s (zeros) or 1s (ones).
  - If you set the destination to 0, the loopback location ID in the loopback cell is initialized to all 0s, and each segment point in the network responds to the ping.
  - If you set the destination to 1s, the loopback location ID in the loopback cell is initialized to all 1s, and only the connection endpoint responds to the ping.
  - If you use the default value of 0xFFFFFFFF, the loopback location ID in the loopback cell is initialized to all 1s.

  For information about how the router handles loopback cells based on location ID, see Table 6 on page 19.
- The **count** keyword sets the number of OAM loopback cells to send to the destination. The default value is 5. The maximum is 32.
- The **timeout** keyword sets the amount of time to wait for a response to the sent OAM loopback cell. The default value is 5 seconds.
- The following characters can appear in the display after the **ping** command has been issued:
  - !—Each exclamation point indicates that a reply was received
  - .—Each period indicates that the ping timed out while waiting for a reply
- Example 1—This example generates end-to-end loopback cells for VPI=0 and VCI=105 on ATM interface 2/0. The count value is 5 OAM loopback cells, and the timeout value is 2 seconds.

```
host1#ping atm interface atm 2/0 0 105 end-loopback count 5 timeout 2
Sending 5 53-byte OAM end-to-end loopback Echoes timeout is 2 secs
Press Ctrl+c to stop
!!!!!
Success rate = 100% (5/5), round-trip min/avg/max = 0/4/10 ms
```

- Example 2—This example generates segment loopback cells for VPI=0 and VCI=105 on ATM interface 2/0. The destination is set to 0xFFFFFFFF, the count value is 3 OAM loopback cells, and the timeout value is 1 second.

```
host1#ping atm interface atm 2/0 0 105 seg-loopback 0xFFFFFFFF count 3  timeout 1
Sending 3 53-byte OAM segment loopback Echoes timeout is 1 secs
Press Ctrl+c to stop
!!!
Success rate = 100% (3/3), round-trip min/avg/max = 0/3/10 ms
```

- There is no **no** version.
- See *ping atm interface atm*.

Configuring an NBMA Interface

You configure an ATM NBMA 1483 subinterface in a manner similar to configuring a standard ATM 1483 subinterface. When you specify a subinterface, however, you must select the multipoint option if you plan to add multiple circuits to form an NBMA interface. If you do not select multipoint, the subinterface defaults to point-to-point, and only a single circuit can be affiliated with that subinterface.

You can configure one or more PVCs and associate them with the subinterface you create. Also, you can enable InARP and identify a refresh rate on each specific circuit. For each NMBA interface, either InARP must be enabled, or a static map entry must be provided for each circuit owned by the interface; otherwise, transmitting over that circuit is impossible.

NOTE: NBMA interfaces support only the aal5snap encapsulation.

To configure an NBMA interface:

1. Configure a physical interface.

   host1(config)#**interface atm 2/0**

2. Configure an ATM 1483 subinterface.

   host1(config-if)#**interface atm 2/0.2 multipoint**

3. Configure PVCs by specifying the VCD, VPI, VCI, and encapsulation type.

   host1(config-subif)#**atm pvc 1 1 1 aal5snap inarp 10**
   host1(config-subif)#**atm pvc 2 2 2 aal5snap**

4. (Optional) Specify InARP and a refresh rate (also optional).

   host1(config-subif)#**atm pvc 3 3 3 aal5snap inarp 5**
   host1(config-subif)#**atm pvc 4 4 4 aal5snap inarp**

5. Assign an IP address and subnet mask to the PVC.

   host1(config-subif)#**ip address 192.32.10.20 255.255.255.0**

6. (Optional) Use the appropriate show commands to verify your configuration.

   host1#**show atm interface atm 2/0**
   host1#**show atm map**
   host1#**show nbma arp atm 2/0**
   host1#**show atm vc atm 2/0 2**
   host1#**show atm subinterface atm 2/0.2**

Creating an NBMA Static Map

Static mapping creates an association between IP address–ATM PVC pairs for one or more member circuits of an ATM 1483 NBMA interface. Not every circuit necessarily gets the required association from a static map.

In the following procedure, you can repeat Step 2 for each circuit you want to map. You can associate with an interface a map group name that you have not already established. When you define the map list, the name is associated with that interface. You can perform Steps 3 and 4 before Steps 1 and 2 without affecting the results.

To set up a static map:

1. Create a map list by naming it.

   host1(config)#**map-list charlie**

2. Associate a protocol and an address with a specific virtual circuit.

   host1(config-map-list)#**ip 192.168.13.13 atm-vc 1 broadcast**

3. Specify an ATM interface.

   host1(config-if)#**interface atm 2/0**

4. Associate the map list with the interface.

   host1(config-if)#**map-group charlie**

atm pvc

- Use to configure a PVC on an ATM interface.
- InARP and refresh rate are optional parameters.
- InARP determines whether InARP requests are used and is specified on a per-circuit basis. If you disable InARP, you must use a static map table entry. Transmission over the circuit cannot occur unless you use either InARP or static map table entries.
- The default refresh rate is 15 minutes.
- You can configure InARP only if you specify the **aal5snap** encapsulation type.
- Example

  host1(config-if)#**atm pvc 6 0 11 aal5snap inarp 10**

- Use the **no** version to remove the specified PVC.
- See *atm pvc*.

interface atm

- Use to configure an ATM interface or subinterface type.
- For information about specifying the ATM interface or subinterface, see “interface atm” on page 23.
- Specify **multipoint** to identify the subinterface as NBMA.
- Examples

  host1(config-if)#**interface atm 0/1.20**
  host1(config-if)#**interface atm 0/0/4.20**

- Use the **no** version to remove the subinterface or the logical interface.
- See *interface atm*.

ip atm-vc

- Use to associate a protocol and address with a specific virtual circuit.
- Use this command repeatedly for each circuit to be mapped.

- This command is available in Map List Configuration mode only.
- Example

  host1(config-map-list)#ip 192.168.13.13 atm-vc 1 broadcast
- Use the **no** version to remove the association.
- See *ip atm-vc*.

map-group

- Use to associate the map list with an NBMA interface when configuring static mapping.
- You can issue this command before or after the **map-list** command without changing anything.
- This command is available in Interface Configuration mode only.
- See the **map-list** command.
- Example

  host1(config-if)#map-group charlie
- Use the **no** version to remove the association.
- See *map-group*.

map-list

- Use to create a map list when configuring static mapped NBMA interfaces.
- Limit the name of the map list to no more than 31 characters.
- You can create multiple map lists; however, you can associate only one map list with each physical interface.
- If a map list contains an entry for a VCD that was previously configured to run InARP, the map-group command fails. If this is the case, either reconfigure the circuit with InARP disabled, or remove the entry for that circuit from the map list.
- Example

  host1(config)#map-list charlie
- Use the **no** version to remove the map list.
- See *map-list*.

Assigning Descriptions to Interfaces

You can use the **description** commands to assign a text description or an alias to an interface, so that other **show** commands can display that information.

atm aal5 description

- Use to assign a text description or alias to an ATM AAL5 interface.
- Use the **show atm aal5 interface** command to display the text description.

- Example

  host1(config-if)#**atm aal5 description boston01**
- Use the **no** version to remove the text description or alias.
- See *atm aal5 description*.

atm atm1483 description

- Use to assign a text description or alias to an ATM 1483 subinterface.
- The description can be a maximum of 255 characters.
- Use the **show atm subinterface** command to display the text description.
- Example

  host1(config-subif)#**atm atm1483 description nyc33**
- Use the **no** version to remove the text description or alias.
- See *atm atm1483 description*.

atm description

- Use to assign a text description or alias to the ATM interface.
- The description can be a maximum of 255 characters and can include the # (pound sign) character.
- The first 32 characters of the ATM description are pushed out to RADIUS during authentication and accounting.
- Use the **show atm interface** command to display the description.
- Example

  host1(config-if)#**atm description myAtm**
- Use the **no** version to remove the description or alias.
- See *atm description*.

Sending Interface Descriptions to AAA

During authentication the router sends ATM interface descriptions to AAA. AAA passes the descriptions to RADIUS, and they can appear in the Calling-Station-Id attribute [31]. (For information about RADIUS and the Calling-Station-ID attribute, see *JunosE Broadband Access Configuration Guide*.)

By default, the router sends the major interface descriptions to AAA on the SRP. You can configure the router to send VP interface descriptions in place of the major interface descriptions, or to send ATM 1483 subinterface descriptions to AAA on the line module. As a result, the VP or ATM 1483 subinterface descriptions can provide a convenient way to identify or group broadband access subscribers.

If you set up multiple interface descriptions, they have the following precedence:

1. ATM 1483 subinterface description
2. VP interface description
3. Major interface description

Assigning Descriptions to Virtual Paths

To assign a description to an individual VP on an ATM interface, use the **atm vp-description** command. The VP description does not affect existing descriptions configured for the ATM interface or ATM 1483 subinterface on which the VP resides. However, if you delete the ATM interface, the descriptions of all VPs residing on that interface are also deleted. In addition, if you decrease the VPI range by issuing the **atm vc-per-vp** command, the router deletes the descriptions of any VPs that are removed.

To display the VP description, use the **show atm vp-description** command, as described in "Using ATM show Commands" on page 71. Although you need not configure a VP tunnel to specify a VP description, the router also displays the VP description in the output of the **show atm vp-tunnel** command.

Exporting ATM 1483 Subinterface Descriptions

To assign a description to an ATM 1483 subinterface and configure the router to send the ATM 1483 VC interface descriptions to the line module:

1. Configure a text description for ATM 1483 subinterfaces with the **atm atm1483 description** command. This description is included in the interface identifier that is sent to AAA.

   To configure this feature for ATM 1483 subinterfaces, enter this command in Profile Configuration mode. See "Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview" on page 626 in "Configuring Dynamic Interfaces Using Bulk Configuration" on page 619.

   host1(config-subif)#atm atm1483 description VC_atm1

2. Set up the router to export ATM 1483 VC interface descriptions to the line module.

   host1(config)#atm atm1483 export-subinterface-description

3. (Optional) Display the configuration of the export ATM 1483 VC interface descriptions feature with the **show atm atm1483** command.

   ```
   host1#show atm atm1483
   ATM1483 IF Descriptions exported
   ```

4. (Optional) Display the interface descriptions with the **show atm subinterface atm** command.

atm atm1483 description

- Use to assign a text description or alias to an ATM 1483 subinterface.
- The description can be a maximum of 255 characters.
- Example

host1(config-subif)#**atm atm1483 description nyc33**

- Use the **no** version to remove the text description or alias.
- See *atm atm1483 description*.

atm atm1483 export-subinterface-description

- Use to export ATM 1483 VC interface descriptions to the line module. Descriptions for ATM 1483 subinterfaces are configured with the **atm atm1483 description** command.
- The description can have up to 255 characters; however, when the description is sent to the line module, it is truncated to 32 characters.
- Example

  host1(config)#**atm atm1483 export-subinterface-description**
- Use the **no** version to restore the default behavior, in which ATM 1483 interface descriptions are not exported to the line module.
- See *atm atm1483 export-subinterface-description*.

atm vp-description

- Use to assign a text description to an individual VP on an ATM interface or subinterface.
- You must specify the VPI of the VP to which you want to assign the description.
- The description string can be a maximum of 32 characters.
- The VP description is stored in NVS and persists after a reboot.
- Use the **show atm vp-description** command to display the text description.
- Example

  host1(config-if)#**atm vp-description 2 vpi2Subscribers**
- Use the **no** version to restore the default value, a null string.
- See *atm vp-description*.

Configuring Individual ATM PVC Parameters

As an alternative to using the **atm pvc** command to configure ATM PVC parameters with a single command, you can access ATM VC Configuration mode to configure individual ATM PVC parameters with separate commands, one parameter at a time. You can configure parameters for the service category, encapsulation method, F5 OAM options, and Inverse ARP.

The following sections explain the benefits of using ATM VC Configuration mode and describes how to configure the ATM VC mode :

- Benefits on page 44
- Creating Control PVCs on page 44
- Creating Data PVCs on page 45

- Configuring the Service Category for Data PVCs on page 46
- Configuring Encapsulation for Data PVCs on page 48
- Configuring F5 OAM for Data PVCs on page 49
- Configuring Inverse ARP for Data PVCs on page 52

Benefits

Using commands in ATM VC Configuration mode to configure individual ATM PVC parameters provides the following benefits:

- Commands in ATM VC Configuration mode are less complex and easier to use.

  With the **atm pvc** command and keywords, you configure multiple PVC attributes on a single command line. In addition, configuration attributes available only for control (ILMI and signaling) PVCs or only for data PVCs are not mutually exclusive.

  By contrast, ATM VC Configuration mode provides commands to configure each parameter individually, and makes a clearer distinction between configuration of control PVCs and configuration of data PVCs.

- ATM VC Configuration mode interoperates with the **atm pvc** command.

  You can configure all of the parameters currently supported by the **atm pvc** command from within ATM VC Configuration mode. In addition, you can create a PVC with the **atm pvc** command and modify or delete the same PVC by using ATM VC Configuration mode. Conversely, you can modify (with certain restrictions) or delete a PVC created in ATM VC Configuration mode by using the **atm pvc** command.

- ATM VC Configuration mode supports additional F5 OAM alarm surveillance and VC integrity options.

  In most cases, you can use either an ATM VC Configuration mode command or the **atm pvc** command to configure ATM PVC parameters. However, to configure F5 OAM alarm surveillance parameters (by using the **oam ais-rdi** command) or VC integrity parameters (by using the **oam retry** command), you *must* use only ATM VC Configuration mode. There are no equivalent **atm pvc** commands to configure these parameters.

  You can, however, continue to use the **atm pvc** command to enable VC integrity and modify the loopback frequency of an ATM data PVC.

NOTE: If you have existing configuration scripts that use the **atm pvc** command, we recommend that you continue to use the **atm pvc** command to configure all ATM PVC parameters except those that require you to use the **oam ais-rdi** command or **oam retry** command in ATM VC Configuration mode.

Creating Control PVCs

A control PVC, also referred to as a control circuit, supports services such as ILMI to manage and control ATM networks. You must create a control PVC on an ATM major interface, and not on an ATM 1483 subinterface that is stacked above an ATM major interface.

To create a control PVC, you issue the **pvc** command from Interface Configuration mode. However, unlike the other tasks in this section, configuring a control PVC with the **pvc** command does not access ATM VC Configuration mode.

For example, the following commands create a control PVC with VCD 10, VPI 0, VCI 16, and ILMI encapsulation.

```
host1(config)#interface atm 3/0
host1(config-if)#pvc 10 0/16 ilmi
host1(config-if)#
```

Regardless of whether you use the **pvc** command or the **atm pvc** command to create a control PVC, you cannot modify the VCD, VPI, or VCI values after they have been configured.

pvc

- Use from Interface Configuration mode to create a control PVC for Integrated Local Management Interface (ILMI).
- To create a control PVC, specify the VCD, VPI and VCI (in the format *vpi/vci*), and the **ilmi** keyword.
- Example

  ```
  host1(config-if)#pvc 5 0/5 ilmi
  ```
- Use the **no** version to remove the specified control PVC from the router.
- See *pvc*.

Creating Data PVCs

A data PVC, also referred to as a data circuit, is an ATM PVC that carries data. You must create a data PVC on an ATM 1483 subinterface that is stacked above an ATM major interface, and not on the ATM major interface itself.

To create a data PVC, you issue the **pvc** command from Subinterface Configuration mode to access ATM VC Configuration mode. From ATM VC Configuration mode, you can then do either of the following:

- Issue the **exit** command, which creates a data PVC that uses default values for service category (unspecified bit rate without a peak cell rate), encapsulation type (**aal5snap**), F5 OAM (disabled), and Inverse ARP (disabled).
- Issue commands to configure or modify data PVC attributes including the service category, encapsulation type, F5 OAM, and Inverse ARP.

For example, the following commands create a data PVC with VCD 32, VPI 0, VCI 100 and default values for the other attributes. Issuing the **exit** command causes the configuration to take effect.

```
host1(config)#interface atm 3/2.2
```

```
host1(config-subif)#pvc 32 0/100
host1(config-subif-atm-vc)#exit
host1(config-subif)#
```

Regardless of whether you use the **pvc** command or the **atm pvc** command to create a data PVC, you cannot modify the VCD, VPI, or VCI values after they have been configured.

pvc

- Use from Subinterface Configuration mode to create a data PVC and access ATM VC Configuration mode, from which you can configure and modify individual PVC attributes one at a time.
- To create a basic data PVC with default values for service category, encapsulation type, F5 OAM, and Inverse ARP, specify the VCD and the VPI and VCI (in the format *vpi/vci*).
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

```
host1(config-subif)#pvc 10 15/50
host1(config-subif-atm-vc)#exit
```

- Use the **no** version to remove the specified data PVC from the router.
- See *pvc*.

Configuring the Service Category for Data PVCs

You can use individual commands in ATM VC Configuration mode to configure each supported service category on a data PVC, or to restore the default service category, unspecified bit rate (UBR) without a peak cell rate (PCR).

For example, the following commands configure a data PVC that uses the constant bit rate (CBR) service category with a nondefault PCR (10,000 Kbps). Issuing the **exit** command causes the configuration to take effect.

```
host1(config)#interface atm 3/0.3
host1(config-subif)#pvc 6 0/100
host1(config-subif-atm-vc)#cbr 10000
host1(config-subif-atm-vc)#exit
host1(config-subif)#
```

cbr

- Use to configure the CBR service category on an ATM data PVC.
- You must specify a PCR, in Kbps, in the range 1–149760 (for OC3 ATM modules) or 1–599040 (for OC12 ATM modules).
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

```
host1(config-subif-atm-vc)#cbr 15000
```


host1(config-subif-atm-vc)#exit

- Use the **no** version to restore the default service category, UBR without a PCR.
- See *cbr*.

ubr

- Use to configure the UBR service category on an ATM data PVC.
- You can optionally specify a PCR, in Kbps, in the range 0–149760 (for OC3 ATM modules) or 0–599040 (for OC12 ATM modules).
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  host1(config-subif-atm-vc)#ubr 5000
  host1(config-subif-atm-vc)#exit

- Use the **no** version to restore the default service category, UBR without a PCR.
- See *ubr*.

vbr-nrt

- Use to configure the variable bit rate, nonreal time (VBR-NRT) service category on an ATM data PVC.
- You must specify all of the following parameters:
  - PCR, in Kbps, in the range 0–149760 (for OC3 ATM modules) or 0–599040 (for OC12 ATM modules)
  - SCR, in Kbps, in the range 0–149760 (for OC3 ATM modules) or 0–599040 (for OC12 ATM modules)
  - Maximum burst size (MBS), in cells, in the range 0–16777215
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  host1(config-subif-atm-vc)#vbr-nrt 50000 10000 150
  host1(config-subif-atm-vc)#exit

- Use the **no** version to restore the default service category, UBR without a PCR.
- See *vbr-nrt*.

vbr-rt

- Use to configure the variable bit rate, real time (VBR-RT) service category on an ATM data PVC.
- You must specify all of the following parameters:
  - PCR, in Kbps, in the range 0–149760 (for OC3 ATM modules) or 0–599040 (for OC12 ATM modules)

- SCR, in Kbps, in the range 0–149760 (for OC3 ATM modules) or 0–599040 (for OC12 ATM modules)
- Maximum burst size (MBS), in cells, in the range 0–16777215

- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  host1(config-subif-atm-vc)#**vbr-rt 200000 30000 400**
  host1(config-subif-atm-vc)#**exit**

- Use the **no** version to restore the default service category, UBR without a PCR.
- See *vbr-rt*.

Configuring Encapsulation for Data PVCs

The encapsulation method on a data PVC represents the format of the data units that traverse the circuit. You can use the **encapsulation** command in ATM VC Configuration mode to configure the encapsulation method for a data PVC, or to restore the default encapsulation method, **aal5snap**.

For example, the following commands configure a data PVC that uses **aal5all** encapsulation. Issuing the **exit** command causes the configuration to take effect.

host1(config)#**interface atm 3/0.3**
host1(config-subif)#**pvc 6 0/250**
host1(config-subif-atm-vc)#**encapsulation aal5all**
host1(config-subif-atm-vc)#**exit**
host1(config-subif)#

encapsulation

- Use to configure the encapsulation method on an ATM data PVC.
- Specify one of the following encapsulation types:
  - **aal0**—Causes the router to receive raw ATM cells on this PVC and forward the cells without performing AAL5 packet reassembly
  - **aal5all**—Configures ATM over MPLS passthrough connections; the router passes through all ATM AAL5 traffic without interpreting it
  - **aal5autoconfig**—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed)
  - **aal5mux ip**—Configures a VC-based multiplexed circuit used for IP only
  - **aal5snap**—Configures an LLC encapsulated circuit; an LLC/SNAP header precedes the protocol datagram; this is the default encapsulation method
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  host1(config-subif-atm-vc)#**encapsulation aal5mux ip**

```
host1(config-subif-atm-vc)#exit
```

- Use the **no** version to restore the default encapsulation method, **aal5snap**.
- See *encapsulation*.

Configuring F5 OAM for Data PVCs

In ATM VC Configuration mode, you can use the individual commands listed in Table 7 on page 49 to configure nondefault values for F5 OAM services.

Table 7: F5 OAM Configuration Tasks and Associated Commands

To Configure	Use This Command
Surveillance parameters for alarm indication signal (AIS) and remote defect indication (RDI) fault management cells	**oam ais-rdi**
Continuity check (CC) verification	**oam cc**
Generation of F5 loopback cells and enabling of VC integrity	**oam-pvc**
Parameters for VC integrity	**oam retry**

For more information about OAM parameters, see "Operations, Administration, and Management of ATM Interfaces" on page 14.

NOTE: The **oam-ais rdi** command and the **oam retry** command are available only in ATM VC Configuration mode. There is no equivalent **atm pvc** command to configure these F5 OAM alarm surveillance and VC integrity parameters.

For example, the following commands enable VC integrity on a data PVC with a nondefault loopback frequency (30 seconds). Issuing the **exit** command causes the configuration to take effect.

```
host1(config)#interface atm 3/0.0
host1(config-subif)#pvc 32 0/32
host1(config-subif-atm-vc)#oam-pvc manage 30
host1(config-subif-atm-vc)#exit
host1(config-subif)#
```

The following commands, which are available only in ATM VC Configuration mode, configure nondefault VC integrity and alarm surveillance parameters on a data PVC. In this example, the VC integrity parameters configured with the **oam retry** command include the up retry count (4), down retry count (6), and retry frequency (2). The alarm surveillance parameters configured with the **oam ais-rdi** command include the alarm down count (2) and alarm clear timeout duration (4 seconds). Issuing the **exit** command causes the configuration to take effect.

```
host1(config)#interface atm 3/0.0
```



```
host1(config-subif)#pvc 32 0/32
host1(config-subif-atm-vc)#oam retry 4 6 2
host1(config-subif-atm-vc)#oam ais-rdi 2 4
host1(config-subif-atm-vc)#exit
host1(config-subif)#
```

oam ais-rdi

- Use to configure surveillance parameters for AIS and RDI F5 OAM fault management cells on an ATM data PVC.
- You can optionally specify the following values:
  - *alarmDownCount*—Number of successive alarm cells, in the range 1–60, for the router to receive before reporting that a PVC is down; the default value is 1
  - *alarmClearTimeout*—Number of seconds, in the range 3–60, for the router to wait before reporting that a PVC is up after the PVC has stopped receiving alarm cells; the default value is 3
- To configure these alarm surveillance parameters, you must use the **oam ais-rdi** command in ATM VC Configuration mode. There is no equivalent **atm pvc** command to configure these parameters.
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  ```
  host1(config-subif-atm-vc)#oam ais-rdi 5 10
  host1(config-subif-atm-vc)#exit
  ```

- Use the **no** version to restore the default values for the alarm down count and alarm clear timeout duration.
- See *oam ais-rdi*.

oam cc

- Use to enable F5 OAM CC verification on an ATM data PVC.
- You can optionally specify one of the following values to configure CC cell flows:
  - **segment**—Opens an F5 OAM CC segment cell flow
  - **end-to-end**—Opens an F5 OAM CC end-to-end cell flow
- You must specify one of the following values to enable CC verification:
  - **source**—Enables this VC as the source point (cell generator)
  - **sink**—Enables this VC as a sink point (cell receiver)
  - **both**—Enables this VC as both a sink point and a source point
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example 1—Enables CC verification with a source endpoint

  ```
  host1(config-subif-atm-vc)#oam cc source
  ```



```
host1(config-subif-atm-vc)#exit
```

- Example 2—Opens an F5 OAM CC segment cell flow and enables CC verification with a sink endpoint

```
host1(config-subif-atm-vc)#oam cc segment sink
host1(config-subif-atm-vc)#exit
```

- Use the **no** version to disable F5 OAM CC verification and restore the default setting for cell termination, **end-to-end**.
- See *oam cc*.

oam-pvc

- Use to enable generation of F5 OAM loopback cells on an ATM data PVC and, optionally, enable F5 OAM VC integrity features on the circuit.
- Use this command only on data PVCs configured with **aal5snap**, **aal5autoconfig**, or **aal5 mux ip** encapsulation; the command is not valid for data PVCs configured with other encapsulation types.
- To enable F5 OAM VC integrity on the PVC, use the **manage** keyword.
- You can optionally specify the number of seconds, in the range 1–600, for the router to wait between the transmission of loopback cells during normal operation; the default value is 10.
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

```
host1(config-subif-atm-vc)#oam-pvc manage 15
host1(config-subif-atm-vc)#exit
```

- Use the **no** version to restore the default behavior, which disables F5 OAM VC integrity on the router and restores the default value for loopback frequency, 10 seconds.
- See *oam-pvc*.

oam retry

- Use to configure F5 OAM VC integrity parameters on an ATM data PVC.
- You can optionally specify the following values:
  - *upRetryCount*—Number of successive loopback cell responses, in the range 1–60, for the router to receive before reporting that a PVC is up; default value is 3
  - *downRetryCount*—Number of successive loopback cell responses, in the range 1–60, for the router to miss before reporting that a PVC is down; default value is 5
  - *retryFrequency*—Number of seconds, in the range 1–600, for the router to wait between the transmission of loopback cells when it is verifying the state of the PVC; default value is 1
- To configure these VC integrity parameters, you must use the **oam retry** command in ATM VC Configuration mode. There is no equivalent **atm pvc** command to configure these parameters.

- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

```
host1(config-subif-atm-vc)#oam retry 5 6 3
host1(config-subif-atm-vc)#exit
```

- Use the **no** version to restore the default values for the up retry count, down retry count, and retry frequency parameters.
- See *oam retry*.

Configuring Inverse ARP for Data PVCs

You can use the **inarp** command in ATM VC Configuration mode to enable Inverse ARP (InARP) on a data PVC that resides on an ATM 1483 NBMA subinterface configured with the **multipoint** option. The PVC must use the default encapsulation method, **aal5snap**. For more information about InARP, see "Configuring an NBMA Interface" on page 37.

For example, the following commands enable InARP with a nondefault refresh rate (10 minutes) on a data PVC. The PVC uses **aal5snap** encapsulation by default. Issuing the **exit** command causes the configuration to take effect.

```
host1(config)#interface atm 3/2.1 multipoint
host1(config-subif)#pvc 6 0/11
host1(config-subif-atm-vc)#inarp 10
host1(config-subif-atm-vc)#exit
host1(config-subif)#
```

inarp

- Use to enable Inverse ARP on an ATM PVC that resides on an ATM 1483 NBMA subinterface and uses the default encapsulation method, **aal5snap**.
- You can optionally specify an Inverse ARP refresh rate, in the range 1–60 minutes; the default value is 15.
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

```
host1(config-subif-atm-vc)#inarp 5
host1(config-subif-atm-vc)#exit
```

- Use the **no** version to restore the default behavior, which disables Inverse ARP on the router.
- See *inarp*.

Configuring ATM VC Classes

As an alternative to configuring individual parameters for ATM data PVCs, you can access ATM VC Class Configuration mode to configure a class of attributes for an ATM data

PVC. A *VC class* is a set of attributes for a virtual circuit (VC) that can include the service category, encapsulation method, F5 OAM options, and Inverse ARP.

After you configure the VC class, you then apply the attributes in the class as a group by assigning the VC class to one of the following:

- An individual PVC
- All PVCs created on a specified static ATM major interface
- All PVCs created on a specified static ATM 1483 subinterface
- A base profile from which bulk-configured VC ranges are created on a dynamic ATM 1483 subinterface

VC class assignments are valid only for ATM data PVCs created with the **pvc** command. Assigning a VC class to a PVC created with the **atm pvc** command, or to a control (ILMI) PVC, has no effect. For information about creating a data PVC by using the **pvc** command, see "Creating Data PVCs" on page 45.

NOTE: For information about the total number of VC classes supported on the router, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

Benefits

Using VC classes to configure and assign attributes to ATM data PVCs provides the following benefits:

- VC classes enable you to classify and group ATM PVCs based on the OAM and traffic requirements of their associated subscribers.

  In a typical scenario, you might group subscribers based on their OAM and traffic requirements, and then create a VC class for each subscriber group. For example, you might create two VC classes: premium-subscriber-class and economy-subscriber-class.

  In premium-subscriber-class, you might enable F5 OAM VC integrity (with the **oam-pvc manage** command), and configure a traffic class that has a higher scheduling priority, such as CBR (with the **cbr** command). Conversely, in economy-subscriber-class, you might retain the default setting that disables F5 OAM VC integrity, and configure a traffic class that has a lower scheduling priority, such as UBR with or without a PCR (with the **ubr** command). By assigning each VC class to the appropriate interfaces or individual circuits, you can group and manage the PVCs associated with the VC class based on the network requirements of the subscribers they serve.

- VC classes facilitate modifications to PVC attributes.

  If the OAM or traffic requirements change for a particular subscriber group, you can simply reconfigure the VC class associated with the PVCs for that subscriber group. This method is easier and less time-consuming than having to modify the attributes for a large number of PVCs by using individual CLI commands.

  Modifications to the attributes in a VC class affect PVCs that are already associated with this VC class as well as PVCs subsequently created for this class.

Precedence Levels

Precedence levels play an important role in determining how the router assigns the attribute values for statically created and dynamically created PVCs that have associated VC classes.

Precedence Levels for Static PVCs

For PVCs that are statically created, the router determines the PVC attribute values according to the following precedence levels, in order from highest precedence to lowest precedence:

1. The most recent explicitly set value for a PVC attribute always has the highest precedence and overrides any settings in the VC class. Explicitly set values for PVC attributes are those values configured with the CLI (by using the **atm pvc** command or commands in ATM VC Configuration mode), SNMP, or assigned by RADIUS.
2. If an attribute value is not explicitly specified, the router takes the value for that attribute from the assigned VC class, in the following order of precedence:
   a. Attribute value specified in the VC class assigned to this PVC
   b. Attribute value specified in the VC class assigned to the ATM 1483 subinterface on which this PVC is created
   c. Attribute value specified in the VC class assigned to the ATM major interface on which this PVC is created
3. If no PVC attributes are explicitly specified and no VC class assignments exist, the router applies the default values for the commands listed in Table 8 on page 56. For information about the default value for each command, see the command descriptions in "Configuring VC Classes" on page 56.

Precedence Levels for Dynamic PVCs

For PVCs that are dynamically created, the router determines the PVC attribute values according to the following precedence levels, in order from highest precedence to lowest precedence:

1. The attribute value specified in the VC class assigned in the base profile always has the highest precedence.
2. If no VC class is assigned in the base profile, the router takes the value for that attribute from the VC class assigned to the associated ATM major interface.
3. If neither the base profile nor the ATM major interface has a VC class assigned, the router takes the value for that attribute from the individually specified attributes in the base profile.
4. If neither the base profile nor the ATM major interface has a VC class assigned, and no attributes are individually specified in the base profile, the router applies the default values for the commands listed in Table 8 on page 56. For information about the default value for each command, see the command descriptions in "Configuring VC Classes" on page 56.

Precedence Level Examples

For examples that illustrate how precedence levels affect the assignment of VC classes, see "Precedence Level Examples for Assigning VC Classes" on page 64.

To help you better understand these examples, we recommend that you first read the following sections to learn how to configure and assign VC classes:

Upgrade Considerations

The following considerations apply to using ATM VC classes when you upgrade to the current JunosE Software release from a lower-numbered JunosE Software release:

- It is possible to use VC classes for PVCs created in a lower-numbered release with the **atm pvc** command. In such cases, the router uses the following rules to determine the PVC attribute values:
  - Nondefault values explicitly specified for PVC attributes with the **atm pvc** command take precedence over the attribute values specified in the associated VC class. As a result, the router takes the values for these attributes from the **atm pvc** command settings.
  - Default values implicitly specified for PVC attributes with the **atm pvc** command have a lower precedence than the attribute values specified in the associated VC class. As a result, the router takes the values for these attributes from the assigned VC class.
- The output of the **show configuration** command uses either the **pvc** command format or the **atm pvc** command format to display ATM PVCs. The display format of configuration information for ATM PVCs created with the **atm pvc** command depends on the JunosE Software release from which you are upgrading, as follows:
  - When you upgrade to the current JunosE Software release from a JunosE release numbered lower than Release 7.3.x, the output of the **show configuration** command uses the **pvc** command format (**pvc** *vcd vpi/vci*) to display configuration information for all ATM PVCs. This occurs even if those PVCs were created in a JunosE release numbered lower than Release 7.3.x with the **atm pvc** command. For example, assume that you created a PVC in JunosE Release 7.2.x by issuing the command **atm pvc 2 0 33 aal5snap 0 0 0**. The **show configuration** command in the current JunosE Software release displays the identifier for this PVC as follows:

    ```
    pvc 2 0/33
    ```

  - When you upgrade to the current JunosE Software release from JunosE Release 7.3.x or a higher-numbered release, the output of the **show configuration** command uses the **atm pvc** command format to display configuration information for ATM PVCs created with the **atm pvc** command. For example, assume that you created a PVC in JunosE Release 7.3.x or Release 8.0.x by issuing the command **atm pvc 2 0 33 aal5snap 0 0 0**. The **show configuration** command in the current JunosE Software release displays the identifier for this PVC as follows:

    ```
    atm pvc 2 0 33 aal5snap 0 0 0
    ```

For PVCs previously created in the lower-numbered release by using the **pvc** command, the **show configuration** command displays configuration information using the **pvc** command format, as described previously.

For information about how to use the **show configuration** command, see *Managing the System* in *JunosE System Basics Configuration Guide*.

To make the most efficient use of the VC class feature when you upgrade to the current JunosE Software release, we recommend that you follow these steps:

1. Delete any PVCs created with the **atm pvc** command and recreate them by using the **pvc** command. For information about creating a data PVC by using the **pvc** command, see "Creating Data PVCs" on page 45.
2. Configure the VC class as described in "Configuring VC Classes" on page 56.
3. Assign the VC class in one of the following ways:
   - Assign the VC class to the individual PVC when you create or modify the PVC.
   - Assign the VC class to the associated ATM major interface or ATM 1483 subinterface before you create the PVC.

Configuring VC Classes

To configure a VC class, you issue the **vc-class atm** command to create and name the VC class. The **vc-class atm** command accesses ATM VC Class Configuration mode, from which you configure a set of attributes to apply to an ATM data PVC.

Table 8 on page 56 lists the commands that you can use in ATM VC Class Configuration mode to configure a set of attributes for a data PVC. These commands are identical to the commands in ATM VC Configuration mode described in "Configuring Individual ATM PVC Parameters" on page 43. For more information about the syntax of each command, see the *JunosE Command Reference Guide*.

Table 8: Commands to Configure VC Class Attributes

cbr	**oam-pvc**
encapsulation	**oam retry**
inarp	**ubr**
oam ais-rdi	**vbr-nrt**
oam cc	**vbr-rt**

For example, the following commands configure two VC classes: premium-subscriber-class and dsl-subscriber-class. You must issue the **exit** command from ATM VC Class Configuration mode for each VC class configuration to take effect.

```
! Configure VC class premium-subscriber-class.
host1(config)#vc-class atm premium-subscriber-class
```

```
host1(config-vc-class)#encapsulation aal5autoconfig
host1(config-vc-class)#cbr 200
host1(config-vc-class)#oam-pvc manage 60
host1(config-vc-class)#oam ais-rdi 5
host1(config-vc-class)#exit
! Configure VC class dsl-subscriber-class.
host1(config)#vc-class atm dsl-subscriber-class
host1(config-vc-class)#encapsulation aal5autoconfig
host1(config-vc-class)#ubr
host1(config-vc-class)#exit
host1(config)#
```

In premium-subscriber-class:

- The **encapsulation** command sets the encapsulation method to **aal5autoconfig**.
- The **cbr** command sets the service category to CBR with a PCR of 200 Kbps.
- The **oam-pvc** command enables generation of F5 OAM loopback cells and F5 OAM VC integrity.
- The **oam ais-rdi** command configures the alarm down count for successive AIS and RDI alarm cells to 5.

In dsl-subscriber-class:

- The **encapsulation** command sets the encapsulation method to **aal5autoconfig**.
- The **ubr** command configures the UBR service category without a PCR.

To configure an ATM VC class with systemwide default values, you can issue the **vc-class atm** command followed immediately by the **exit** command. For example, the following commands create a VC class named default-vc-class. Because no attribute values are explicitly specified in default-vc-class, the router applies the default values for the commands listed in Table 8 on page 56. For information about the default value for each command, see the command descriptions in this section.

```
! Configure VC class with default values.
host1(config)#vc-class atm default-subscriber-class
host1(config-vc-class)#exit
host1(config)#
```

To verify the VC class configuration, use the **show atm vc-class** command. For information about how to use this command, see "show atm vc-class" on page 95.

cbr

- Use to configure the CBR service category on an ATM data PVC.
- For detailed information about how to use this command, see "cbr" on page 46.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

```
host1(config-vc-class)#cbr 15000
host1(config-vc-class)#exit
```


- Use the **no** version to restore the default service category, UBR without a PCR.
- See *cbr*.

encapsulation

- Use to configure the encapsulation method on an ATM data PVC.
- For detailed information about how to use this command, see "encapsulation" on page 48.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**encapsulation aal5mux ip**
  host1(config-vc-class)#**exit**
- Use the **no** version to restore the default encapsulation method, **aal5snap**.
- See *encapsulation*.

inarp

- Use to enable Inverse ARP on an ATM PVC that resides on an ATM 1483 NBMA subinterface and uses the default encapsulation method, **aal5snap**.
- For detailed information about how to use this command, see "inarp" on page 52.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**inarp 5**
  host1(config-vc-class)#**exit**
- Use the **no** version to restore the default behavior, which disables Inverse ARP on the router.
- See *inarp*.

oam ais-rdi

- Use to configure surveillance parameters for AIS and RDI F5 OAM fault management cells on an ATM data PVC.
- For detailed information about how to use this command, see "oam ais-rdi" on page 50.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**oam ais-rdi 5 10**
  host1(config-vc-class)#**exit**
- Use the **no** version to restore the default values for the alarm down count (1 successive alarm cell) and alarm clear timeout duration (3 seconds).
- See *oam ais-rdi*.

oam cc

- Use to enable F5 OAM CC verification on an ATM data PVC.
- For detailed information about how to use this command, see "oam cc" on page 50.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example 1—Enables CC verification with a source endpoint

  host1(config-vc-class)#**oam cc source**
  host1(config-vc-class)#**exit**
- Example 2—Opens an F5 OAM CC segment cell flow and enables CC verification with a sink endpoint

  host1(config-vc-class)#**oam cc segment sink**
  host1(config-vc-class)#**exit**
- Use the **no** version to disable F5 OAM CC verification and restore the default setting for cell termination, **end-to-end**.
- See *oam cc*.

oam-pvc

- Use to enable generation of F5 OAM loopback cells on an ATM data PVC and, optionally, enable F5 OAM VC integrity features on the circuit.
- For detailed information about how to use this command, see "oam-pvc" on page 51.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**oam-pvc manage 15**
  host1(config-vc-class)#**exit**
- Use the **no** version to restore the default behavior, which disables F5 OAM VC integrity on the router and restores the default value for loopback frequency, 10 seconds.
- See *oam-pvc*.

oam retry

- Use to configure F5 OAM VC integrity parameters on an ATM data PVC.
- For detailed information about how to use this command, see "oam retry" on page 51.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**oam retry 5 6 3**
  host1(config-vc-class)#**exit**

- Use the **no** version to restore the default values for the up retry count (3 successive loopback cell responses), down retry count (5 successive loopback cell responses), and retry frequency (1 second).
- See *oam retry*.

ubr

- Use to configure the UBR service category on an ATM data PVC.
- For detailed information about how to use this command, see "ubr" on page 47.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#ubr 5000
  host1(config-vc-class)#exit
- Use the **no** version to restore the default service category, UBR without a PCR.
- See *ubr*.

vbr-nrt

- Use to configure the VBR-NRT service category on an ATM data PVC.
- For detailed information about how to use this command, see "vbr-nrt" on page 47.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#vbr-nrt 50000 10000 150
  host1(config-vc-class)#exit
- Use the **no** version to restore the default service category, UBR without a PCR.
- See *vbr-nrt*.

vbr-rt

- Use to configure the VBR-RT service category on an ATM data PVC.
- For detailed information about how to use this command, see "vbr-rt" on page 47.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#vbr-rt 200000 30000 400
  host1(config-vc-class)#exit
- Use the **no** version to restore the default service category, UBR without a PCR.
- See *vbr-rt*.

vc-class atm

- Use to create and name a VC class for an ATM data PVC.
- You must specific a VC class name of up to 32 alphanumeric characters.
- The **vc-class atm** command accesses ATM VC Class Configuration mode, from which you can configure a set of attributes for the PVC including the service category, encapsulation method, F5 OAM options, and Inverse ARP.
- You must issue the **exit** command from ATM VC Class Configuration mode for the VC class configuration to take effect.
- For information about the total number of VC classes supported on the router, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.
- Example

  host1(config)#**vc-class atm dsl-subscriber-class**
  host1(config-vc-class)#**exit**

- Use the **no** version to remove the named VC class from the router. You cannot remove a VC class that is currently assigned to at least one ATM PVC, ATM 1483 subinterface, or ATM major interface without first issuing the **no class-vc** command or the **no class-int** command to remove the VC class association with the PVC, interface, or subinterface.
- See *vc-class atm*.

Assigning VC Classes to Individual PVCs

To assign a previously configured VC class to an individual ATM data PVC, you use the **class-vc** command from ATM VC Configuration mode. Issuing this command applies the set of attributes configured in the specified VC class to the ATM data PVC.

NOTE: The **class-vc** command is valid only for a data PVC created with the **pvc** command. It has no effect for data PVCs created with the **atm pvc** command, or for control (ILMI) PVCs. For information about creating a data PVC by using the **pvc** command, see "Creating Data PVCs" on page 45.

For example, the following commands assign the VC class named premium-subscriber-class to the ATM data PVC with VCD 2, VPI 0, and VCI 200.

```
! Assign VC class premium-subscriber-class to PVC 2/0.200
host1(config)#interface atm 2/0.200
host1(config-subif)#pvc 200 0/200
host1(config-subif-atm-vc)#class-vc premium-subscriber-class
host1(config-subif-atm-vc)#exit
```

For those attributes that you do not explicitly specify for the ATM PVC, the router applies the values specified in the VC class. As explained in "Precedence Levels" on page 54, the values in a VC class assigned to an individual PVC take precedence over both of the following:

- Values in a VC class assigned to an ATM 1483 subinterface

- Values in a VC class assigned to an ATM major interface

For examples that illustrate how precedence levels affect the assignment of VC classes, see “Precedence Level Examples for Assigning VC Classes” on page 64.

class-vc

- Use to assign a previously configured VC class to an individual ATM data PVC.
- The **class-vc** command is valid only for data PVCs created with the **pvc** command.
- You must issue the **exit** command from ATM VC Configuration mode for the VC class association to take effect.
- Example

```
host1(config-subif-atm-vc)#class-vc dsl-subscriber-class
host1(config-subif-atm-vc)#exit
```

- Use the **no** version to remove the VC class association with the data PVC.
- See *class-vc*.

Assigning VC Classes to ATM Major Interfaces

To assign a previously configured VC class to an ATM major interface, you use the **class-int** command from Interface Configuration mode. Issuing this command applies the set of attributes in the specified VC class to the ATM data PVCs statically or dynamically created on this interface.

For example, the following commands assign the VC class named dsl-subscriber-class to an ATM major interface configured on slot 5, port 0.

```
! Assign VC class dsl-subscriber-class to ATM interface 5/0.
host1(config)#interface atm 5/0
host1(config-if)#class-int dsl-subscriber-class
host1(config-if)#exit
```

For those attributes that you do not explicitly specify for an ATM PVC, the router applies the values specified in the VC class. As explained in “Precedence Levels” on page 54, the values in a VC class assigned to an ATM major interface have a lower precedence than both of the following:

- Values in a VC class assigned to an individual ATM PVC
- Values in a VC class assigned to an ATM 1483 subinterface

This means that if a VC class is assigned to an individual PVC or ATM 1483 subinterface configured on the major interface, the attribute values configured in the VC class assigned to the PVC or subinterface override the attribute values configured in the VC class assigned to the major interface.

For examples that illustrate how precedence levels affect the assignment of VC classes, see “Precedence Level Examples for Assigning VC Classes” on page 64.

class-int

- Use from Interface Configuration mode to assign a previously configured VC class to an ATM major interface.
- You must issue the **exit** command from Interface Configuration mode for the VC class association to take effect.
- Example

  host1(config-if)#**class-int gold-subscriber-class**
  host1(config-if)#**exit**

- Use the **no** version to remove the VC class association with the interface. Issuing the **no** version causes the router to set the PVC attributes to their systemwide default values, or to the values set in the associated VC class with the next highest order of precedence.
- See *class-int*.

Assigning VC Classes to Static ATM 1483 Subinterfaces

To assign a previously configured VC class to a static ATM 1483 subinterface, you use the **class-int** command from Subinterface Configuration mode. Issuing this command applies the set of attributes in the specified VC class to the ATM data PVCs statically or dynamically created on this subinterface.

For example, the following commands assign the VC class named premium-subscriber-class to an ATM 1483 subinterface configured on slot 5, port 0, subinterface 100.

```
! Assign VC class dsl-subscriber-class to ATM 1483 subinterface 5/0.100.
host1(config)#interface atm 5/0.100
host1(config-subif)#class-int premium-subscriber-class
host1(config-subif)#exit
```

For those attributes that you do not explicitly specify for an ATM PVC, the router applies the values specified in the VC class. As explained in “Precedence Levels” on page 54, the values in a VC class assigned to an ATM 1483 subinterface take precedence over the values in a VC class assigned to an ATM major interface, but have a lower precedence than the values in a VC class assigned to an individual ATM PVC.

This means that if a VC class is assigned to a PVC configured on the subinterface, the attribute values configured in the VC class assigned to the individual PVC override the attribute values configured in the VC class assigned to the subinterface.

For examples that illustrate how precedence levels affect the assignment of VC classes, see “Precedence Level Examples for Assigning VC Classes” on page 64.

class-int

- Use from Subinterface Configuration mode to assign a previously configured VC class to a static ATM 1483 subinterface.
- You must issue the **exit** command from Subinterface Configuration mode for the VC class association to take effect.

- Example

  host1(config-subif)#class-int silver-subscriber-class
  host1(config-subif)#exit

- Use the **no** version to remove the VC class association with the subinterface. Issuing the **no** version causes the router to set the VC attributes to their systemwide default values, or to the values set in the associated VC class with the next highest order of precedence.
- See *class-int*.

Assigning VC Classes to Base Profiles for Bulk-Configured VC Ranges

To assign a VC class to a base profile for a dynamic ATM 1483 subinterface, you can use the **atm class-vc** command from Profile Configuration mode. Issuing this command applies the set of attributes in the specified VC class to all bulk-configured VC ranges that are dynamically created from this profile.

For more information, see "Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview" on page 626 in "Configuring Dynamic Interfaces Using Bulk Configuration" on page 619.

Precedence Level Examples for Assigning VC Classes

The examples in this section illustrate how the precedence level rules described in "Precedence Levels" on page 54 affect the assignment of VC classes and PVC attribute values.

For all of these examples, assume that you have issued the following commands to configure a VC class named my-premium-class:

host1(config)#vc-class atm my-premium-class
host1(config-vc-class)#encapsulation aal5autoconfig
host1(config-vc-class)#cbr 200
host1(config-vc-class)#oam-pvc manage 60
host1(config-vc-class)#oam ais-rdi 5
host1(config-vc-class)#exit

Example 1 and Example 2 illustrate the effect of precedence levels when you assign the VC class my-premium-class to an individual PVC with VCD 200, VPI 0, and VCI 200. Example 3 illustrates how using the **atm pvc** command affects VC class assignment. Finally, Example 4 illustrates how modifications to a VC class affect PVC attributes applied through RADIUS.

Example 1: Explicitly Changing the Service Category

Explicitly specified attribute values take precedence over attribute values specified in a VC class. As a result, the following commands cause the router to use the most recent explicitly specified value, UBR with a PCR of 200 Kbps, as the service category for this PVC instead of the service category specified in my-premium-class, CBR with a PCR of 200 Kbps. The router takes the values for the other attributes from the VC class my-premium-class.


```
host1(config)#interface atm 2/0.200
host1(config-subif)#pvc 200 0/200
host1(config-subif-vc)#ubr 200
host1(config-subif-vc)#class-vc my-premium-class
host1(config-subif-vc)#exit
```

The following commands change the service category for the PVC to VBR-RT because this is the most recent explicitly specified value for this attribute. The router takes the values for the other attributes from the VC class my-premium-class, which is still assigned to the PVC.

```
host1(config)#interface atm 2/0.200
host1(config-subif)#pvc 200 0/200
host1(config-subif-vc)#vbr-rt 200 150 200
host1(config-subif-vc)#exit
```

The following commands cause the router to retain the VBR-RT service category for the PVC because it is still the most recent explicitly specified value for this attribute. The router takes the values for the other attributes from the VC class my-premium-class.

```
host1(config)#interface atm 2/0.200
host1(config-subif)#pvc 200 0/200
host1(config-subif-vc)#class-vc my-premium-class
host1(config-subif-vc)#exit
```

Example 2: Changing the Encapsulation Method in the VC Class

The following commands change the value for the encapsulation method in the VC class my-premium-class from **aal5autoconfig** to **aal5snap**. As a result, the router now uses **aal5snap** instead of **aal5autoconfig** as the encapsulation method for the PVCs to which this VC class is assigned.

```
host1(config)#vc-class atm my-premium-class
host1(config-vc-class)#encapsulation aal5snap
host1(config-vc-class)#exit
```

Example 3: Effect of Using the atm pvc Command

The following commands, which attempt to assign the my-premium-class VC class to a PVC originally created with the **atm pvc** command, have no effect. The router interprets all attribute values specified with the **atm pvc** command as explicitly specified values, and therefore takes the values for these attributes from the **atm pvc** command instead of from the VC class. As a result, the router continues to use **aal5mux ip** as the encapsulation method for this PVC instead of the encapsulation method specified in the VC class my-premium-class.

```
host1(config)#interface atm 2/0.300
host1(config-subif)#atm pvc 300 0 300 aal5mux ip
host1(config-subif)#pvc 300 0/300
host1(config-subif-vc)#class-vc my-premium-class
host1(config-subif-vc)#exit
```

Example 4: Overriding RADIUS Values

If RADIUS is configured to provide traffic parameters for PVCs, a more recent, explicitly specified change in the VC class associated with that PVC overrides the PVC values applied through RADIUS.

In the following example, assume that RADIUS has been configured to apply a service category of CBR with a PCR of 400 Kbps to the PVC. Initially, the PVC uses the service category configured in my-premium-class, CBR with a PCR of 200 Kbps. However, when the subscriber logs in through RADIUS, the router applies the RADIUS-configured service category, CBR with a PCR of 400 Kbps.

While the subscriber is still logged in, my-premium-class is modified to change the service category to CBR with a PCR of 600 Kbps. Because this VC class modification results in the most recent, explicitly specified value for the service category, the router now uses CBR with a PCR of 600 Kbps as the service category for the PVC instead of the service category configured through RADIUS.

```
host1(config)#interface atm 2/0.200
host1(config-subif)#pvc 200 0/200
host1(config-subif-vc)#class-vc my-premium-class
host1(config-subif-vc)#exit
! Subscriber logs in through RADIUS, which applies service category of CBR
! with a PCR of 400 Kbps to PVC.
host1(config)#vc-class atm my-premium-class
host1(config-vc-class)#cbr 600
host1(config-vc-class)#exit
! Router now applies service category of CBR with a PCR of 600 Kbps to PVC.
```

Configuring Dynamic ATM 1483 Subinterfaces

As an alternative to the static ATM interface configurations described in this chapter, you can also configure dynamic ATM 1483 subinterfaces over static ATM AAL5 interfaces over ATM. Dynamic ATM 1483 subinterfaces can perform autodetection and dynamic creation of the following upper-layer encapsulation types:

- Bridged Ethernet
- IP
- PPP
- PPPoE

For details, see "Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview" on page 626 in
"Configuring Dynamic Interfaces Using Bulk Configuration" on page 619.

Monitoring ATM

This section explains how to set a statistics baseline, display bit rate and packet rate statistics for ATM virtual circuits (VCs), and use the **show** commands to view your ATM configuration and monitor ATM VCs and VPs.

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

Setting Statistics Baselines

You can set a statistics baseline for ATM interfaces, ATM virtual circuits, and ATM virtual paths configured on the router.

baseline atm vp interface

- Use to set a statistics baseline for an ATM virtual path (VP) interface.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- To set the baseline for an ATM VP, specify the VPI. The numeric range of the VPI depends on the line module capabilities and current configuration.
- To display baseline statistics, use the **delta** keyword with ATM **show** commands.
- Examples

  host1#**baseline atm vp interface atm 12/0 0**
  host1#**baseline atm vp interface atm 5/0/0 1**

- There is no **no** version.
- See *baseline atm vp interface*.

baseline interface atm

- Use to set a statistics baseline for ATM interfaces or a specific virtual circuit.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- To set the baseline for a circuit, specify a VCD in the range 1–2147483647.
- To set the baseline on an interface, omit the VCD.
- To display baseline statistics, use the **delta** keyword with ATM **show** commands.
- Examples

  host1#**baseline interface atm 9/1 123**
  host1#**baseline interface atm 5/0/0 123**

- There is no **no** version.
- See *baseline interface*.

Displaying Interface Rate Statistics for ATM VCs and ATM VPs

You can use the following commands to display bit rate and packet rate statistics over a specified time interval for one or more ATM virtual circuits (VCs) or virtual paths (VPs) configured on the router.

- To monitor the data rate for ATM VCs, use the **monitor atm vc** command.
- To monitor the data rate for ATM VPs, use the **monitor atm vp** command.

To monitor the data rate for ATM VCs and ATM VPs:

1. Log in to the router by using a local console session or a virtual terminal (vty) session (such as a Telnet session).

   While you use the **monitor atm vc** command or the **monitor atm vp** command, you must keep the console or terminal session open. You cannot issue any other commands during the session.

   For information about logging in to the router, see *Accessing the CLI* in *JunosE System Basics Configuration Guide*.

2. Access User Exec mode or Privileged Exec mode.

   For information, see *Accessing Command Modes* in *JunosE System Basics Configuration Guide*.

3. Specify the interface identifier and VCD (for each ATM VC) or VPI (for each ATM VP) that you want to monitor. For information about specifying an ATM interface, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

   host1#monitor atm vc atm 12/0 1 atm 8/0 1 display-time-of-day
   host1#monitor atm vp atm 12/0 0 atm 12/0 1 load-interval 15 display-time-of-day

   By default, the router uses a 5-second time interval between polls to calculate bit rates and packet rates for each specified VC or VP. Optionally, you can use the **load-interval** keyword to specify a nondefault time interval in the range 5–30 seconds (for ATM VCs) or 5–300 seconds (for ATM VPs).

   You can also include the optional **display-time-of-day** keyword to show the time of day at which the router gathers statistics for each interval. Displaying the time of day enables you to monitor when a particular VC or VP is underutilized or overutilized.

4. Review the command output.

```
host1#monitor atm vc atm 12/0 1 atm 8/0 1 display-time-of-day
                       Seconds
                       between                                       Time
  Interface       VCD  polls    Input bps/pps   Output bps/pps   (UTC)
--------------- ---- -------- --------------- --------------- --------
ATM 12/0         1      0        --/--          --/--         10:43:11
ATM 8/0          1      0        --/--          --/--         10:43:11
ATM 12/0         1      5     121840/100     121840/100       10:43:16
ATM 8/0                 5     121600/100     121600/100       10:43:16
ATM 12/0         1      5       98008/80     98008/80         10:43:21
ATM 8/0          1      5       98008/80       98008/80       10:43:21
...
host1#monitor atm vp atm 12/0 0 atm 12/0 1 load-interval 15 display-time-of-day
```



```
                              Seconds
                              between                                     Time
   Interface         VPI      polls     Input bps/pps   Output bps/pps   (UTC)
---------------- ----- -------- --------------- --------------- --------
ATM 12/0             0        0           --/--            --/--    09:47:18
ATM 12/0             1        0           --/--            --/--    09:47:18
ATM 12/0             0       15     6635792/6480    6635792/6480    09:47:33
ATM 12/0             1       15     6635312/6479    6635312/6479    09:47:33
ATM 12/0             0       15     6635176/6479    6635176/6479    09:47:48
ATM 12/0             1       15     6634424/6478    6634424/6478    09:47:48
ATM 12/0             0       15     6635448/6479    6635448/6479    09:48:03
```

The **monitor atm vc** command and **monitor atm vp** command display similar information, except that the **monitor atm vc** command displays the VCD for each interface and the **monitor atm vp** command displays the VPI for each interface.

The router polls the statistics of each VC or VP identified in the command at the specified load interval to calculate and display bit rate and packet rate statistics. The first line of output for each interface always displays 0 (zero) for the number of seconds between polls, and dashes (--/--) in the Input bps/pps and Output bps/pps columns. These values indicate that the router initially takes a baseline for each interface against which to measure subsequent statistics. The router continues to display subsequent lines of output for each interface at the specified load interval until you press Ctrl+c to stop the command.

For a description of the fields in the command display, see "monitor atm vc" on page 69 and "monitor atm vp" on page 69.

5. If you remove an ATM interface or (for VCs) an ATM 1483 subinterface while you are monitoring one or more VCs or VPs that reside on the deleted interface, press Ctrl+c to stop the **monitor atm vc** command or **monitor atm vp** command, and then restart the command to ensure accurate interface rate statistics are displayed.

   If you leave the **monitor atm vc** command or **monitor atm vp** command running after you remove the interface, the command displays undefined or inaccurate statistics for the VC or VP on the interface that has been removed. The display of undefined or inaccurate statistics can result when you remove the interface by issuing either the **no interface atm** command or **slot erase** command, and can continue even after you recreate the interface with the same VC or VP configuration.

6. When you are finished monitoring, press Ctrl+c to stop the **monitor atm vc** command or **monitor atm vp** command.

   ```
   host1#^C
   ```

monitor atm vc

monitor atm vp

- Use the **monitor atm vc** command to display bit rate and packet rate statistics over a specified time interval for one or more ATM VCs.
- Use the **monitor atm vp** command to display bit rate and packet rate statistics over a specified time interval for one or more ATM VPs.
- You must use either command in a dedicated console or terminal session for the duration of the monitoring session.

- Specify the interface identifier and VCD (for each ATM VC) or VPI (for each ATM VP) that you want to monitor.
- To specify a nondefault time interval in the range 5–30 seconds (for ATM VCs) or 5–300 seconds (for ATM VPs) at which the router calculates bit rate and packet rate statistics, use the optional **load-interval** keyword. The default time interval for either command is 5 seconds.
- To display the time at which the router calculates bit rate and packet rate statistics for the current interval, use the optional **display-time-of-day** keyword.
- To stop either command, press Ctrl+c.
- Field descriptions
  - Interface—Interface identifier for the ATM interface on which the VC or VP resides
  - VCD—Virtual circuit descriptor that identifies the VC (**monitor atm vc** command only)
  - VPI—Virtual path identifier of the PVC (**monitor atm vp** command only)
  - Seconds between polls—Number of seconds at which the router calculates bit rate and packet rate statistics
  - Input bps/pps—Number of bits per second (bps) and packets per second (pps) received on this interface during the specified load interval
  - Output bps/pps—Number of bits per second (bps) and packets per second (pps) transmitted on this interface during the specified load interval
  - Time—Time of day, in hh:mm:ss format, at which the router calculates the bit rate and packet rate statistics for the current interval
- Example 1—Displays bit rate and packet rate statistics over the default (5-second) load interval for a single ATM VC

```
host1#monitor atm vc atm 12/0 100
                         Seconds
                         between
    Interface       VCD   polls   Input bps/pps  Output bps/pps
------------------ ----- -------- --------------- ---------------
ATM 12/0             100    0           --/--           --/--
ATM 12/0             100    5     6631624/6476    6631624/6476
ATM 12/0             100    5     6630808/6475    6631008/6475
ATM 12/0             100    5     6632448/6477    6632240/6476
ATM 12/0             100    5     6629168/6473    6629168/6473
ATM 12/0             100    5     6631008/6475    6631216/6475
host1#^C
```

- Example 2—Displays bit rate and packet rate statistics over the default (5-second) load interval for two ATM VCs, with the time of day that the statistics were calculated

```
host1#monitor atm vc atm 12/0 100 atm 12/0 200 display-time-of-day
                          Seconds
                          between                                   Time
    Interface       VCD   polls   Input bps/pps  Output bps/pps   (UTC)
------------------ ----- -------- --------------- --------------- --------
ATM 12/0             100    0          --/--           --/-- 17:33:06
ATM 12/0             200    0          --/--           --/-- 17:33:06
```



```
ATM 12/0             100     5     6635104/6479     6635104/6479 17:33:11
ATM 12/0             200     5     6633264/6477     6633472/6478 17:33:11
ATM 12/0             100     5     6632856/6477     6632856/6477 17:33:16
ATM 12/0             200     5     6633264/6477     6633056/6477 17:33:16
host1# ^C
```

- Example 3—Displays bit rate and packet rate statistics over a 10-second load interval for two ATM VPs

```
host1#monitor atm vp atm 12/0 0 atm 12/0 1 load-interval 10
                         Seconds
                         between
    Interface       VPI  polls    Input bps/pps  Output bps/pps
----------------- ----- -------- --------------- ---------------
ATM 12/0            0      0          --/--           --/--
ATM 12/0            1      0          --/--           --/--
ATM 12/0            0      10     6635312/6479    6635312/6479
ATM 12/0            1      10     6634288/6478    6634288/6478
ATM 12/0            0      10     6637664/6482    6637664/6482
ATM 12/0            1      10     6637872/6482    6637872/6482
host1#^C
```

- Example 4—Displays bit rate and packet rate statistics over a 15-second load interval for two ATM VPs, with the time of day that the statistics were calculated

```
host1#monitor atm vp atm 12/0 0 atm 12/0 1 load-interval 15 display-time-of-day
                         Seconds
                         between                                       Time
    Interface      VPI   polls    Input bps/pps  Output bps/pps    (UTC)
----------------- ----- -------- --------------- --------------- --------
ATM 12/0           0      0           --/--           --/--      17:36:19
ATM 12/0           1      0           --/--           --/--      17:36:19
ATM 12/0           0      15     6634352/6478    6634352/6478    17:36:34
ATM 12/0           1      15     6633608/6478    6633608/6478    17:36:34
ATM 12/0           0      15     6635176/6479    6635176/6479    17:36:49
ATM 12/0           1      15     6635040/6479    6635040/6479    17:36:49
host1# ^C
```

- There is no **no** version.
- See *monitor atm vc*.
- See *monitor atm vp*.

Using ATM show Commands

Use the **show** commands described in this section to display information about your ATM configuration and monitor ATM interfaces.

You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *show Commands* in *JunosE System Basics Configuration Guide*.

show atm aal5 interface

- Use to display information about a configured ATM AAL5 interface.
- Field descriptions

  - AAL5 Interface operational status—Operational status of the AAL5 interface: up, down, lowerlayerDown
  - time since last status change—Time since last reported change to the AAL5 interface operational status
  - SNMP trap link-status—Whether SNMP link status traps are enabled or disabled on the ATM AAL5 interface
  - Auto configure ATM 1483 status—Setting of the autoconfiguration feature for a dynamic ATM 1483 subinterface configured over the ATM AAL5 interface:
    - enabled—Autodetection of the ATM 1483 dynamic encapsulation type is enabled on the ATM AAL5 interface
    - disabled—Autodetection of the ATM 1483 dynamic encapsulation type is not currently enabled on the ATM AAL5 interface
  - InPackets—Number of packets received on this interface
  - InBytes—Number of bytes received on this interface
  - OutPackets—Number of packets transmitted on this interface
  - OutBytes—Number of bytes transmitted on this interface
  - InErrors—Number of incoming errors received on this interface
  - OutErrors—Number of outgoing errors on this interface
  - InPacketDiscards—Number of incoming packets discarded on this interface
  - OutDiscards—Number of outgoing packets discarded on this interface

- Example

```
host1#show atm aal5 interface atm 3/0
AAL5 Interface ATM 3/0 operational status:    lowerLayerDown
      time since last status change: 00:08:46

SNMP trap link-status: disabled
Auto configure ATM 1483 status: disabled

InPackets:        0
InBytes:          0
OutPackets:       0
OutBytes:         0
InErrors:         0
OutErrors:        0
InPacketDiscards: 0
OutDiscards:      0
```

- See *show atm aal5 interface*.

show atm atm1483

- Use to display whether or not the router is set up to export ATM 1483 subinterface descriptions to the line module.
- Example

```
host1#show atm atm1483
ATM1483 IF Descriptions exported
```

- See *show atm atm1483*.

show atm interface

show interfaces atm

- Use to display configuration and state information and statistics about a specific ATM interface, or to display a brief description of all ATM interfaces configured in the router.
- To specify an ATM interface for ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port* format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module; on the OC3-2 GE APS I/O module, you can specify ATM interfaces only in ports 0 and 1; port 2 is reserved for a Gigabit Ethernet interface
- To specify an ATM interface for the E120 and E320 routers, use the *slot/adapter/port* format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
- To display the status and number of configured VCs for all ATM interfaces configured in the router, use the **brief** keyword.
- Field descriptions
  - ATM Interface status—State of the physical interface: up, down
  - line protocol—State of the ILMI protocol: disabled, up, down
  - AAL5 operational status—Operational status of the ATM AAL5 interface: up, down, lowerLayerDown
  - time since last status change—Time since last reported change to the AAL5 operational status
  - ATM operational status—Operational status of the ATM interface: up, down, lowerLayerDown
  - time since last status change—Time since last reported change to the ATM operational status
  - UNI version—UNI version: 3.0, 3.1, 4.0
  - Maximum VCs—Maximum number of virtual circuits supported on this interface

  - Current VCs—Current number of virtual circuits configured
  - ILMI VPI/VCI—Number of VPI and VCI configured for ILMI (displayed only when ILMI is configured on the interface)
  - VCD—Number of VCD (displayed only when ILMI is configured on the interface)
  - ILMI keepalive—State and status of the ILMI (displayed only when ILMI is configured on the interface)
  - Max VCI per VPI—Maximum number of virtual circuits on each virtual path
  - CAC admin state—Enabled, disabled
  - Subscription bandwidth—Maximum allowable bandwidth on the port (displayed only when CAC is enabled)
  - UBR weight—Configured bandwidth for UBR and UBR-PCR connections (displayed only when CAC is enabled)
  - Available bandwidth—Bandwidth currently available on the port (displayed only when CAC is enabled)
  - SNMP trap link-status—Enabled, disabled
  - OAM cell receive status—Whether the ATM interface processes or flushes OAM cells: enabled, disabled
  - OAM cell filter—Whether the interface flushes all OAM cells or flushes only AIS and RDI alarm cells (displayed only when OAM cell receive status is enabled)
  - atm oam loopback-location—Loopback location ID for this interface
  - Interface Alias—Text description or alias if configured for the interface
  - Assigned VC Class—Name of the VC class assigned to this interface, if configured
  - InPackets—Number of packets received on this interface
  - InBytes—Number of bytes received on this interface
  - InCells—Number of cells received on this interface
  - OutPackets—Number of packets transmitted on this interface
  - OutBytes—Number of bytes transmitted on this interface
  - OutCells—Number of cells transmitted on this interface
  - InErrors—Number of incoming errors received on this interface
  - OutErrors—Number of outgoing errors on this interface
  - InPacketDiscards—Number of incoming packets discarded on this interface
  - InByteDiscards—Number of incoming bytes discarded on this interface
  - InCellErrors—Increments when a T3 or an E3 ATM interface receives cells for a VPI or VCI that is not configured on that interface

- Field descriptions specific to the applicable physical interface. In Example 1, the output contains the following physical interface fields:

- SONET path operational status—State of the SONET path interface: up, down, lowerLayerDown
- time since last status change—Time since last reported change to the SONET path operational status
- SONET operational status—State of SONET interface: up, down, lowerLayerDown
- time since last status change—Time since last reported change to the SONET operational status
- PHY Type—Physical port type on which this interface is running
- Framing—Framing type of the physical interface
- TX clocking—Clocking type for the physical interface
- Loopback—Loopback status for the physical interface: enabled, disabled
- Receive FIFO Overruns—Number of times received FIFO was overrun
- qos-mode-port—Status of SAR backpressure: enabled, disabled
- queue—Hardware packet queue associated with the specified traffic class and interface
- Forwarded packets, Bytes—Number of packets and bytes forwarded on this queue
- Dropped committed packets, Bytes—Number of committed packets and bytes that were dropped
- Dropped conformed packets 0, Bytes 0—Number of conformed packets and bytes that were dropped
- Dropped exceeded packets 0, Bytes 0—Number of exceeded packets and bytes that were dropped
- Interface—ATM interface identifier
- Status—Status of the ATM interface: up, down, lowerLayerDown
- Configured VCs—Number of VCs configured on the interface

- Example 1—Displays information about a specific interface

```
host1#show atm interface atm 2/0
ATM Interface 2/0 is down, line protocol is down

AAL5 operational status:       lowerLayerDown
      time since last status change: 22:08:21
ATM operational status:        down
      time since last status change: 22:02:11
SONET path operational status: lowerLayerDown
      time since last status change: 1 day, 0 hours
SONET operational status:      down
      time since last status change: 1 day, 0 hours
UNI version: 3.0, Maximum VCs: 4096
Current VCs: 1
ILMI VPI/VCI: 17/23, VCD 26, ILMI keepalive: disabled
Max VCI per VPI: 32768
CAC admin state: enabled
Subscription bandwidth: 3000000 kbps
```

```
UBR weight: 3000 kbps
Available bandwidth: 2992000 kbps
SNMP trap link-status: enabled
OAM cell receive status: enabled
OAM cell filter : all cells
atm oam loopback-location 0XFFFFFFFF
Interface Alias: ATM interface slot #2 port 0
Assigned VC class       : dsl-subscriber-class

PHY Type: oc3, Framing: sonet, TX clocking: line
Loopback: none, Receive FIFO Overruns: 0

5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec

InPackets:        0
InBytes:          0
InCells:          0
OutPackets:       0
OutBytes:         0
OutCells:         0
InErrors:         0
OutErrors:        0
InPacketDiscards: 0
InByteDiscards:   0
InCellErrors:     0
qos-mode-port disabled

queue 0: traffic class control, bound to ATM2/0
  Forwarded packets 643, Bytes 36008
  Dropped committed packets 0, Bytes 0
  Dropped conformed packets 0, Bytes 0
  Dropped exceeded packets 0, Bytes 0
```

- Example 2—Shows a summary of all ATM interfaces

  host1#**show atm interface brief**

```
                                 Configured
Interface       Status              VCs
---------   --------------   ----------
ATM 2/0     up                    2
ATM 2/1     up                    3
ATM 2/2     lowerLayerDown        4
ATM 2/3     down                  5
ATM 4/0     up                    2
ATM 6/0     lowerLayerDown        2
```

- See *show atm interface*.
- See *show interfaces*.

show atm map

- Use to display the list of all configured ATM static maps to remote hosts on an ATM network.
- Field descriptions
  - Map list—Name of map list and method used to enter the map list. PERMANENT indicates that the map entry was configured; it was not entered automatically by a process.

- protocol address maps to VCx—Name of protocol, the protocol address, and the VCD that the address is mapped to (for ATM VCs configured with the atm pvc command).
  - VC—Number of the virtual circuit
  - broadcast—Indicates pseudo-broadcasting
  - connection up—Indicates a point-to-point virtual circuit

- Example 1

```
host1#show atm map
Map list my-map : PERMANENT
 ip 192.168.2.10 maps to VC 10 atm 2/0
 ip 192.168.2.20 maps to VC 11 atm 2/0    broadcast
 ip 192.168.2.30 maps to VC 12 atm 2/0
Map list other-map : PERMANENT
 ip 192.10.2.10 maps to VC 100 atm 2/1
 ip 192.10.2.20 maps to VC 101 atm 2/1
 ip 192.10.2.30 maps to VC 102 atm 2/1    broadcast
```

- Example 2

```
host1#show atm map brief
Map list my-map : PERMANENT
Map list other-map : PERMANENT
```

- Example 3

```
host1#show atm map my-map
Map list my-map : PERMANENT
 ip 192.168.2.10 maps to VC 10 atm 2/0
 ip 192.168.2.20 maps to VC 11 atm 2/0    broadcast
 ip 192.168.2.30 maps to VC 12 atm 2/0
```

- See *show atm map*.

show atm oam

- Use to display F4 OAM statistics for an ATM interface.
- You must specify a VPI value in addition to the required ATM interface specifier.
- You can use the following keywords.
  - **segment**—Displays information about segment loopbacks
  - **end-to-end**—Displays information about end-to-end loopbacks
- To see F4 OAM circuits that are open, use the **show atm vc** command.
- Field descriptions
  - Sending End To End Loopback Cells—Enabled, disabled
  - Frequency—Frequency configured on this circuit
  - End To End OAM CC verification—Whether end-to-end CC verification is enabled or disabled
  - OAM CC Type—Whether the circuit is a sink or a source, or both a sink and a source

- OAM Current CC state
  - Ready—OAM CC is not enabled
  - Active—OAM CC cell flow is running
  - Activation Failed—OAM CC activation failed
  - Wait Activate—Waiting for interface to come up before the software sends the activation request
  - Wait Activation Confirmation—Waiting for activation confirmation from the peer
  - Wait DeActivate—Waiting for interface to come up before the software sends the deactivation request
  - Wait DeActivation Confirmation—Waiting for deactivation confirmation from the peer
- Segment OAM CC verification—Whether segment CC verification is enabled or disabled
- VP State—State of the VP: up, down
- VP End To End Oam State
  - not managed—Circuit is in normal OAM state; no OAM fault conditions
  - AIS—Circuit is in AIS state
  - RDI—Circuit is in RDI state
- VP Segment Oam State
  - not managed—Circuit is in normal OAM state; no OAM fault conditions
  - AIS—Circuit is in AIS state
  - RDI—Circuit is in RDI state
- InOamF4Cells—Number of F4 OAM cells received
- InOamF4CellsDropped—Number of incoming F4 OAM cells that were dropped
- InOamF4EndLoopbackCells—Total number of F4 end-to-end loopback cells received on this interface, which is the sum of the following counts:
  - InOamF4EndLoopbackCommands—Number of F4 end-to-end loopback commands received
  - InOamF4EndLoopbackResponses—Number of F4 end-to-end loopback responses received
- InOamF4SegLoopbackCells—Total number of F4 segment loopback cells received on this interface, which is the sum of the following counts:
  - InOamF4SegLoopbackCommands—Number of F4 segment loopback commands received
  - InOamF4SegLoopbackResponses—Number of F4 segment loopback responses received

- InOamF4EndAisCells—Number of F4 end-to-end AIS cells received
- InOamF4SegAisCells—Number of F4 segment AIS cells received
- InOamF4EndRdiCells—Number of F4 end-to-end RDI cells received
- InOamF4SegRdiCells—Number of F4 segment RDI cells received
- InOamF4EndCCActDeActCells—Number of F4 end-to-end activation or deactivation CC cells received
- InOamF4SegCCActDeActCells—Number of F4 segment activation or deactivation CC cells received
- InOamF4EndCCCells—Number of F4 end-to-end CC cells received
- InOamF4SegCCCells—Number of F4 segment CC cells received
- OutOamF4Cells—Number of F4 OAM cells sent
- OutOamF4EndLoopbackCells—Total number of F4 end-to-end loopback cells sent on this interface, which is the sum of the following counts:
  - OutOamF4EndLoopbackCommands—Number of F4 end-to-end loopback commands sent
  - OutOamF4EndLoopbackResponses—Number of F4 end-to-end loopback responses sent
- OutOamF4SegLoopbackCells—Total number of F4 segment loopback cells sent on this interface, which is the sum of the following counts:
  - OutOamF4SegLoopbackCommands—Number of F4 segment loopback commands sent
  - OutOamF4SegLoopbackResponses—Number of F4 segment loopback responses sent
- OutOamF4EndRdiCells—Number of end-to-end RDI cells sent
- OutOAM F4SegRdiCells—Number of segment RDI cells sent
- OutOamF4EndCCActDeActCells—Number of F4 end-to-end activation or deactivation CC cells sent
- OutOamF4SegCCActDeActCells—Number of F4 segment activation or deactivation CC cells sent
- OutOamF4EndCCCells—Number of F4 end-to-end CC cells sent
- OutOamF4SegCCCells—Number of F4 segment CC cells sent

- Example 1

```
host1#show atm oam 2/1 0
Sending End To End Loopback Cells is Enabled: Frequency = 20 secs
End To End OAM CC verification enabled
OAM CC Type : CC Sink End Point
OAM Current CC state : Ready
Segment OAM CC verification enabled
OAM CC Type : CC Sink End Point
```

```
OAM Current CC state : Ready
VP State                         :down
VP End To End Oam State          :not managed
VP Segment Oam State             :not managed
InOamF4Cells                     :0
InOamF4CellsDropped              :0
InOamF4EndLoopbackCells          :0
  InOamF4EndLoopbackCommands         :0
  InOamF4EndLoopbackResponses        :0
InOamF4SegLoopbackCells          :0
  InOamF4SegLoopbackCommands         :0
  InOamF4SegLoopbackResponses        :0
InOamF4EndAisCells               :0
InOamF4SegAisCells               :0
InOamF4EndRdiCells               :0
InOamF4SegRdiCells               :0
InOamF4EndCCActDeActCells        :0
InOamF4SegCCActDeActCells        :0
InOamF4EndCCCells                :0
InOamF4SegCCCells                :0
OutOamF4Cells                    :0
OutOamF4EndLoopbackCells         :0
  OutOamF4EndLoopbackCommands        :0
  OutOamF4EndLoopbackResponses       :0
OutOamF4SegLoopbackCells         :0
  OutOamF4SegLoopbackCommands        :0
  OutOamF4SegLoopbackResponses       :0
OutOamF4EndRdiCells              :0
OutOamF4SegRdiCells              :0
OutOamF4EndCCActDeActCells       :0
OutOamF4SegCCActDeActCells       :0
OutOamF4EndCCCells               :0
OutOamF4SegCCCells               :0
Time since last status change    :00:00:33
```

- Example 2

```
host1#show atm oam 2/1 0 segment
Segment OAM CC verification enabled
OAM CC Type : CC Sink End Point
OAM Current CC state: Ready
VP State                        :down
VP Oam State                    :not managed
InOamF4SegmentCells              :0
InOamF4SegmentCellsDropped       :0
InOamF4SegLoopbackCells          :0
  InOamF4SegLoopbackCommands             :0
  InOamF4SegLoopbackResponses            :0
InOamF4SegCCActDeActCells        :0
InOamF4SegCCCells                :0
OutOamF4SegmentCells             :0
OutOamF4SegLoopbackCells         :0
  OutOamF4SegLoopbackCommands             :0
  OutOamF4SegLoopbackResponses            :0
OutOamF4SegRdiCells              :0
OutOamF4SegCCActDeActCells       :0
OutOamF4SegCCCells               :0
Time since last status change    :00:00:53
```

- Example 3


```
host1#show atm oam 2/1 0 end-to-end
Sending End To End Loopback Cells Disabled:
End To End OAM CC verification enabled
OAM CC Type : CC Sink End Point
OAM Current CC state: Ready
VP State                         :down
VP Oam State                     :not managed
InOamF4EndCells                 :0
InOamF4EndCellsDropped          :0
InOamF4EndLoopbackCells         :0
  InOamF4EndLoopbackCommands              :0
  InOamF4EndLoopbackResponses             :0
InOamF4EndAisCells :0
InOamF4EndRdiCells :0
InOamF4EndCCActDeActCells       :0
InOamF4EndCCCells               :0
OutOamF4EndCells                :0
OutOamF4EndLoopbackCells        :0
  OutOamF4EndLoopbackCommands             :0
  OutOamF4EndLoopbackResponses            :0
OutOamF4EndRdiCells             :0
OutOamF4EndCCActDeActCells      :0
OutOamF4EndCCCells              :0
Time since last status change   :00:00:53
```

- See *show atm oam*.

show atm ping

- Use to show all existing ping entries, both completed and outstanding. Remember that ping statistics are overwritten when a new ping is issued on the circuit.
- You can specify the following options to show ping for entries for a specific interface, VPI, or VCI.
  - *interfaceSpecifier*—Shows ping entries for this interface
  - *vpi*—Shows details of the last **ping atm** command on this VPI
  - *vci*—Shows details of the last **ping atm** command on this VCI
- Field descriptions
  - Interface—Interface number
  - VPI—Virtual path identifier
  - VCI—Virtual channel identifier
  - CellCount—OAM loopback cell count configured on the interface
  - TimeOut—Timeout configured on the interface
  - SentCellCount—Number of loopback cells sent
  - RespCount—Number of loopback response cells received
  - Status—Status of the ping
  - Ping Cell Count—Cell count configured on the circuit
  - Ping Time Out—Timeout, in seconds, configured on the circuit

  - No Of Cells Sent—Number of ping cells sent on this circuit
  - No Of Response Received—Number of ping responses received on this circuit
  - Success Rate—Percentage of successful responses received for pings sent
  - round-trip min/max/avg—Minimum, maximum, and average time in milliseconds that it took to receive responses to ping messages sent
  - Ping Status—Results of the ping operation
    - Ping Completed—Number of ping requests in the cell count were sent
    - Ping in Progress—Ping is in operation
    - Ping Not Started—Ping operation is not started; you may see this via SNMP
    - Ping Stopped—Ping operation was manually stopped
    - Ping Stopped OAM Down—**atm oam flush** command was issued when ping was enabled
    - ATM Interface Down—Ping operation is stopped as a result of interface down operational status
  - OAM Flow Type—Segment, End-to-end

- Example 1—Displays all entries in the router

```
host1#show atm ping
Interface VPI VCI CellCount TimeOut SentCellCount RespCount  Status
--------- --- --- --------- -------  ------------- --------- -------
ATM 2/1   0   100     5       5            5           5   Ping Completed
ATM 2/1   0   200     5       5            5           5   Ping Completed
ATM 2/2   0   100     5       5            5           5   Ping Completed
ATM 2/2   0   200     5       5            5           5   Ping Completed
% Found 4  Entries in the system
```

- Example 2—Displays entries on an interface

```
host1#show atm ping 2/1
Interface VPI VCI  CellCount TimeOut  SentCellCount RespCount  Status
--------- --- ---  --------- -------  ------------- ---------- -------
ATM 2/1   0   100      5       5            5           5   Ping Completed
ATM 2/1   0   200      5       5            5           5   Ping Completed
% Found 2  Entries in this Interface
```

- Example 3—Displays entries on a circuit

```
host1#show atm ping atm 2/1 0 100
Ping Cell Count           :5
Ping Time Out             :5secs
No Of Cells Sent          :5
No Of Response Received   :5
Success Rate              :100%
round-trip min/max/avg    :0/10/2 ms
Ping Status               :Completed
Oam Flow Type             :Segment
```

- See *show atm ping*.

show atm subinterface

- Use to display the current state of all ATM subinterfaces, all ATM subinterfaces configured on a specified ATM physical interface, or a specific ATM subinterface.
- To specify an ATM subinterface for ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port.subinterface* format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM subinterface for the E120 and E320 routers, use the *slot/adapter/port.subinterface* format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To display brief summary information for all ATM subinterfaces, or for ATM subinterfaces configured on a specified ATM physical interface, use the **summary** keyword.
- To display status information only for ATM subinterfaces with a specific operating status, use the **status** keyword with one of the following status values. (See the Status field description for an explanation of these values.)
  - dormant
  - dormantLockout
  - down
  - lowerLayerDown
  - notPresent
  - up
- To display the current state of an ATM subinterface created on the PVC with the specified VPI and VCI values, use the **atm** *slot/port/vpi/vci* format (for ERX7xx models, ERX14xx models, and ERX310 router) or the *slot/adapter/port/vpi/vci* format (for E120 and E320 routers) to identify the ATM subinterface (Example 5).

NOTE: You can use the **atm** *slot/port/vpi/vci* format as an alternative to the **atm** *slot/port.subinterface* format with the specific **show interface** and **show subinterface** commands to monitor all ATM 1483 subinterfaces (except NBMA interfaces) as well as the upper-layer interfaces configured over an ATM 1483 subinterface. You cannot, however, use the **atm** *slot/port/vpi/vci* format to create or modify an ATM 1483 subinterface.

These guidelines also apply to E120 and E320 routers when you use the **atm** *slot/adapter/port/vpi/vci* format as an alternative to the **atm** *slot/adapter/port.subinterface* format.

- For more information, see "Creating a Basic Configuration" on page 21.
- Field descriptions
  - Interface—Interface identifier
  - ATM-Prot—One of the following ATM protocol types:
    - RFC-1483—Multiprotocol encapsulation over AAL5
    - NBMA—Nonbroadcast multiaccess interface
    - ATM/MPLS—Local ATM passthrough interface
  - VCD—Virtual circuit descriptor
  - VPI—Virtual path identifier
  - VCI—Virtual circuit (or channel) identifier
  - Circuit Type—Type of circuit: PVC
  - Encap—Administered encapsulation method based on what was configured with the **atm pvc** command
  - MTU—Maximum transmission unit size for the interface
  - Status—One of the following ATM 1483 subinterface states:
    - absent—Represents the notPresent state and indicates that, although the SRP detects the ATM 1483 subinterface, the module on which the subinterface resides has not completed booting up, has failed, or is disabled.
    - dormant—Indicates that the ATM 1483 subinterface is performing autodetection of one or more upper-layer encapsulation types and is waiting to receive a packet of that type on a lower-layer interface. An ATM 1483 subinterface transitions from the dormant state to the up state when the router receives a valid packet of the specified encapsulation type on the interface.
    - dormantLockout—Indicates that a dormant ATM 1483 subinterface has one or more upper-layer encapsulation types currently undergoing encapsulation type lockout. An ATM 1483 subinterface transitions from the dormantLockout state to the dormant state when the lockout time expires for all upper-layer encapsulation types undergoing lockout. An ATM 1483 subinterface transitions from the

dormantLockout state to the up state when the router receives a valid packet for an encapsulation type that is configured for autodetection but is not undergoing lockout.

  - down—Indicates that the ATM 1483 subinterface is administratively disabled or has a circuit that is down or not configured.
  - lowerLayerDown—Indicates that a lower-layer interface below the ATM 1483 subinterface is down.
  - up—Indicates that the ATM 1483 subinterface is up and able to transfer data. For an ATM 1483 subinterface that supports one or more dynamic upper-layer interfaces, indicates that the router has created the dynamic upper-layer interface or is in the process of creating it.

- Interface Type—Type of ATM 1483 subinterface: dynamic or static
- Auto configure status—Setting of the autoconfiguration feature
  - dynamic—Autodetection is on; the router automatically detects the next upper interface
  - static—Autodetection is off
- Auto configure interface(s)—Types of dynamic upper interfaces configured with the **auto-configure** command: bridged Ethernet, IP, PPP, or PPPoE
- Detected 1483 encapsulation—If the encapsulation type is set to **aal5autoconfig**, displays the 1483 encapsulation type detected on the subinterface (displays AUTO until a packet is detected)
- Detected dynamic interface—Type of dynamic upper interface detected during autoconfiguration: bridged Ethernet, IP, PPP, PPPoE, or (if no packet has been received) none
- Interface types in lockout—Encapsulation types currently experiencing lockout: bridged Ethernet, IP, PPP, PPPoE, or none
- Lockout state (seconds)—Settings of encapsulation type lockout for the upper-layer encapsulation type indicated
  - Min—Minimum lockout time, in seconds
  - Max—Maximum lockout time, in seconds
  - Current—Current lockout time, in seconds; displays 0 (zero) if lockout is not occurring
  - Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if lockout is not occurring
  - Next—Lockout time for the router to use for the next lockout event, in seconds
- Assigned profile—For each dynamic interface type, indicates whether or not a profile is assigned and, if assigned, displays the profile name
- Interface Alias—Text description or alias if configured for the subinterface

- Subscriber info—Subscriber login information for the specified dynamic interface type (bridged Ethernet or IP)
- Assigned VC Class—Name of the VC class assigned to this subinterface, if configure
- SNMP trap link-status—Trap link status: enabled or disabled
- Advisory receive speed—Configured receive speed, in Kbps, for the dynamic ATM 1483 subinterface. The E Series LAC sends this value to the LNS in the RX Connect-Speed AVP [38].
- InPackets—Number of packets received on this interface
- InBytes—Number of bytes received on this interface
- OutPackets—Number of packets transmitted on this interface
- OutBytes—Number of bytes transmitted on this interface
- InErrors—Number of errors received on this interface
- OutErrors—Number of outgoing errors on this interface
- InPacketDiscards—Number of incoming packets discarded on this interface
- InPacketsUnknownProtocol—Number of incoming packets with an unknown protocol type
- OutDiscards—Number of outgoing packets discarded on this interface

- Example 1—Displays the current state of all ATM subinterfaces

```
host1#show atm subinterface
                                Circuit                     Interface
 Interface   ATM-Prot VCD VPI VCI Type  Encap MTU    Status      Type
----------- -------- --- --- --- ----- ----- ---- -------------- ---------
ATM 2/0.101 RFC-1483 101  0 101 PVC    AUTO  9180 dormantLockout Static
ATM 2/0.102 RFC-1483 102  0 102 PVC    AUTO  9180 up             Dynamic
ATM 2/0.103 RFC-1483 103  0 103 PVC    AUTO  9180 dormant        Static
3 interface(s) found
```

- Example 2—Displays summary information for all ATM subinterfaces shown in Example 1

```
host1#show atm subinterface summary
3 subinterfaces: 1 up, 0 down,
1 dormant, 1 dormantLockout,
0 lowerLayerDown, 0 not present
```

- Example 3—Displays status information about the current state of all ATM subinterfaces in the dormantLockout state

```
host1#show atm subinterface status dormantLockout
                                 Circuit                      Interface
 Interface   ATM-Prot VCD VPI VCI Type  Encap MTU   Status       Type
----------- -------- --- --- --- ----- ----- ---- ------------- ---------
ATM 2/0.101 RFC-1483 101  0 101 PVC    AUTO  9180 dormantLockout Static
1 interface(s) found
```

- Example 4—Displays the current state of a specific ATM subinterface

```
host1#show atm subinterface atm 2/0.101
                                 Circuit                      Interface
```



```
 Interface  ATM-Prot VCD VPI VCI Type  Encap MTU   Status       Type
----------- -------- --- --- --- ----- ----- ---- ------------ ---------
ATM 2/0.101 RFC-1483 101  0 101 PVC    AUTO  9180 dormantLockout Static

Auto configure status         : dynamic
Auto configure interface(s)   : IP  bridgedEthernet  PPP  PPPoE
Detected 1483 encapsulation   : AUTO
Detected dynamic interface    : none
Interface types in lockout    : IP
Lockout state (seconds)       : Min Max  Current Elapsed Next
------------------------------ --- ---- ------- ------- ----
IP                               1   30      16       7   30
BridgedEnet                    900 3600       0       0  900
PPP                              1  300       0       0    1
PPPoE                            1  300       0       0    1

Assigned profile (IP)         : ipoa
Assigned profile (BridgedEnet): beth
Assigned profile (PPP)        : ppptest
Assigned profile (PPPoE)      : pppoetest
Assigned profile (any)        : none assigned

Interface Alias: atm20101

BridgedEnet subscriber info   :
Username: elaine@jpeterman.com
Password: putty
Authenticate: enabled

Assigned VC class             : premium-subscriber-class
SNMP trap link-status: disabled

InPackets:                0
InBytes:                  1904
OutPackets:               0
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         14
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found
```

- Example 5—Displays the current state of a specific ATM subinterface created on the PVC with the specified VPI and VCI values

  host1#**show atm subinterface atm 0/0/0/101**

```
                                 Circuit                  Interface
Interface   ATM-Prot VCD VPI VCI  Type   Encap MTU  Status   Type
----------- -------- --- --- --- ------- ----- ---- ------ ---------
ATM 0/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 up     Static

Auto configure status         : dynamic
Auto configure interface(s)   : PPPoE
Detected 1483 encapsulation   : SNAP
Detected dynamic interface    : PPPoE
Interface types in lockout    : none

Lockout state (seconds)       : Min Max Current Elapsed Next
------------------------------ --- --- ------- ------- ----
PPPoE                            1 300       0       0    1

Assigned profile (IP)         : none assigned
Assigned profile (BridgedEnet): none assigned
Assigned profile (PPP)        : none assigned
```

```
Assigned profile (PPPoE)      : pppoeprofile
Assigned profile (any)        : none assigned

Assigned VC class             : dsl-subscriber-class
SNMP trap link-status: disabled

Advisory receive speed: 2000

InPackets:                 5119
InBytes:                   358672
OutPackets:                5107
OutBytes:                  357510
InErrors:                  0
OutErrors:                 0
InPacketDiscards:          3
InPacketsUnknownProtocol: 0
OutDiscards:               0
1 interface(s) found
```

- See *show atm subinterface*.

show atm vc

- Use to display a summary of all configured ATM virtual circuits (VCs) and reserved VC ranges.
- You can specify one or more of the following keywords individually or in combination:
  - **vpi**—Displays VCs on a specific VPI
  - **category**—Displays VCs that have a specific service category
  - **status**—Displays VCs with a certain status
- You can also specify the **reserved** keyword with no other keywords to display only a summary of all reserved VC ranges on the router. This includes VPI/VCI ranges reserved for use by dynamic ATM 1483 subinterfaces and by MPLS.
- Field descriptions
  - Interface—Interface number
  - VPI—Virtual path identifier
  - VCI—Virtual channel identifier
  - VCD—Virtual circuit descriptor
  - Type—Type of circuit: PVC
  - Encap—Encapsulation method: AUTO, AAL5, AAL0, MUX, SNAP, ILMI, F4-OAM
  - Category—Service type configured on the VC: UBR, UBR-PCR, NRT-VBR, RT-VBR, or CBR
  - Rx/Tx Peak—Peak rate, in Kbps
  - Rx/Tx Avg—Average rate, in Kbps
  - Rx/Tx Burst—Maximum number of cells that can be burst at the peak cell rate
  - Status—State of the virtual circuit: Up or Down
  - Start VPI—Starting virtual path identifier (inclusive) of the reserved VC range

  - Start VCI—Starting virtual circuit identifier (inclusive) of the reserved VC range
  - End VPI—Ending virtual path identifier (inclusive) of the reserved VC range
  - End VCI—Ending virtual circuit identifier (inclusive) of the reserved VC range

- Example 1—Displays all VCs and reserved VC ranges on the router

  host1#**show atm vc**

```
                                      Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
ATM 3/0.3      0 102  4376 PVC  AUTO  CBR   1000     0     0 DOWN
...
ATM 3/0.8099   1 8099 8099 PVC  SNAP  UBR      0     0     0 UP
ATM 3/0.8100   1 8100 8100 PVC  SNAP  UBR      0     0     0 DOWN
8000 circuit(s) found
Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

- Example 2—Displays VCs with a VPI of zero (0)

  host1#**show atm vc vpi 0**

```
                                      Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
ATM 3/0.3      0 102  4376 PVC  AUTO  CBR   1000     0     0 DOWN
2 circuit(s) found that match filter criteria
```

- Example 3—Displays VCs with a VPI of 1 (one) that are assigned the service category UBR

  host1#**show atm vc vpi 1 category ubr**

```
                                      Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.8099   1 8099 8099 PVC  SNAP  UBR      0     0     0 UP
ATM 3/0.8100   1 8100 8100 PVC  SNAP  UBR      0     0     0 DOWN
2 circuit(s) found that match filter criteria
```

- Example 4—Displays VCs with a VPI of 0 (zero) and a service category of CBR that have a status of up

  host1#**show atm vc vpi 0 category cbr status up**

```
                                      Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
1 circuit(s) found that match the filter criteria
```

- Example 5—Displays all reserved VC ranges on the router

  host1#**show atm vc reserved**

```
Reserved VCC ranges:
```

```
          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

- See *show atm vc*.

show atm vc atm

- Use to display information about a specific VC.
- To specify the circuit to display, do one of the following:
  - Enter the VCD.
  - Use the **vpi-vci** keyword and enter the VPI and VCI.
  - Enter the description configured for the ATM 1483 subinterface (with the **atm atm1483 description** command) on which the VC resides.
- Field descriptions
  - VCD—Virtual circuit descriptor
  - VPI—Virtual path identifier
  - VCI—Virtual channel identifier
  - Encap—Encapsulation method
  - Service Type—Service type configured on the VC: UBR, UBR-PCR, NRT-VBR, RT-VBR, CBR
  - Inverse ARP enable—Whether or not Inverse ARP is enabled: yes, no
  - Assigned VC class—Name of the VC class assigned to this VC, if configured
  - InPackets—Number of packets received on this circuit
  - InBytes—Number of bytes received on this circuit
  - InCells—Number of ATM cells received on this circuit
  - OutPackets—Number of packets transmitted on this circuit
  - OutBytes—Number of bytes transmitted on this circuit
  - OutCells—Number of ATM cells transmitted on this circuit
  - InErrors—Number of errors received on this circuit
  - OutErrors—Number of outgoing errors on this circuit
  - InPacketDiscards—Number of incoming packets discarded on this circuit
  - InPacketUnknownProtocol—Number of incoming packets with an unknown protocol type
  - InByteDiscards—Number of incoming bytes discarded on this circuit
  - CrcErrors—Number of CRC errors detected on this circuit

- SAR time-outs—Number of segmentation and reassembly (SAR) timeouts reached on this circuit
- Over-sized SDUs—Number of oversized service data units (SDUs) received on this circuit
- Alarm drop count—Number of successive alarm cells that the router receives before reporting that the PVC is down
- Alarm clear timeout—Number of seconds that the router waits before reporting that the PVC is up after the PVC stops receiving alarm cells
- OAM VC verification—Whether OAM verification is enabled or disabled
- OAM loopback cell status:
  - disabled—VC integrity disabled for VC
  - sent—OAM loopback cell sent; waiting for response
  - received—OAM loopback cell response received
  - failed—OAM loopback reply not received within frequency period, or reply contained a bad correlation tag
- OAM VC status:
  - AIS—VC is in AIS state
  - RDI—VC is in RDI state
  - Down Retry—OAM loopback failed; using retry frequency to verify that the VC is really down
  - Down—OAM loopback failed after Down Retry verification
  - Up Retry—OAM loopback successful; using retry frequency to verify that the VC is really up
  - Up—OAM loopback successful after Up Retry verification
  - Not Managed—VC integrity is not enabled; for more information about this status value, see "Automatic Disabling of F5 OAM Services" on page 19
- OAM loopback frequency—Frequency with which OAM loopback cells are transmitted (when enabled), in seconds
- OAM up retry count—Number of consecutive successfully looped OAM cells required to mark the VC as Up
- OAM down retry count—Number of consecutive unsuccessfully looped OAM cells required to mark the VC as Down
- OAM loopback retry frequency—Frequency with which OAM cells are transmitted in verification mode, in seconds
- OAM CC verification—Whether CC verification is enabled or disabled
- OAM CC Type—Whether the VC is a sink or a source, or both sink and source end point

- OAM CC Flow Type—End-to-end or segment
- OAM Current CC state
  - Ready—OAM CC is not enabled
  - Active—OAM CC cell flow is running
  - Activation Failed—OAM CC activation failed
  - Wait Activate—Waiting for interface to come up before the software sends the activation request
  - Wait Activation Confirmation—Waiting for activation confirmation from the peer
  - Wait DeActivate—Waiting for interface to come up before the software sends the deactivation request
  - Wait DeActivation Confirmation—Waiting for deactivation confirmation from the peer
- InOamF5Cells—Number of F5 OAM cells received on this circuit
- InOamCellDiscards—Number of received OAM cells that were dropped; dropped cells include unsupported and invalid F5 cells. The InOamCellDiscards counter is not incremented after an OAM flush is performed with the **atm oam flush** command. For more information about the InOamCellDiscards counter, see "Rate Limiting for F5 OAM Cells" on page 20.
- InF5EndLoopCells—Total number of F5 end-to-end loopback cells received on this circuit, which is the sum of the following counts:
  - InF5EndLoopCommands—Number of F5 end-to-end loopback commands received
  - InF5EndLoopResponses—Number of F5 end-to-end loopback responses received
- InF5SegLoopCells—Total number of F5 segment loopback cells received on this circuit, which is the sum of the following counts:
  - InF5SegLoopCommands—Number of F5 segment loopback commands received
  - InF5SegLoopResponses—Number of F5 segment loopback responses received
- InF5EndAisCells—Number of F5 end-to-end AIS cells received on this circuit
- InF5SegAisCells—Number of F5 segment AIS cells received on this circuit
- InF5EndRdiCells—Number of F5 end-to-end RDI cells received on this circuit
- InF5SegRdiCells—Number of F5 segment RDI cells received on this circuit
- InF5EndCCActDeActCells—Number of F5 end-to-end activation and deactivation CC cells received on this circuit
- InF5SegCCActDeActCells—Number of F5 segment activation and deactivation CC cells received on this circuit
- InF5EndCCCells—Number of F5 end-to-end CC cells received on this circuit
- InF5SegCCCells—Number of F5 segment CC cells received on this circuit

- OutOamF5Cells—Number of F5 OAM cells transmitted on this circuit
- OutF5EndLoopCells—Total number of F5 end-to-end loopback cells transmitted on this circuit, which is the sum of the following counts:
  - OutF5EndLoopCommands—Number of F5 end-to-end loopback commands transmitted
  - OutF5EndLoopResponses—Number of F5 end-to-end loopback responses transmitted
- OutF5SegLoopCells—Total number of F5 segment loopback cells transmitted on this circuit, which is the sum of the following counts:
  - OutF5SegLoopCommands—Number of F5 segment loopback commands transmitted
  - OutF5SegLoopResponses—Number of F5 segment loopback responses transmitted
- OutF5EndRdiCells—Number of F5 end-to-end RDI cells transmitted on this circuit
- OutF5SegRdiCells—Number of F5 segment RDI cells transmitted on this circuit
- OutF5EndCCActDeActCells—Number of F5 end-to-end activation and deactivation CC cells transmitted on this circuit
- OutF5SegCCActDeActCells—Number of F5 segment activation and deactivation CC cells transmitted on this circuit
- OutF5EndCCCells—Number of F5 end-to-end CC cells transmitted on this circuit
- OutF5SegCCCells—Number of F5 segment CC cells transmitted on this circuit
- Circuit is Up/Down—Status of the circuit and time since the status of the circuit last changed

- Example 1—Displays statistics for the VC with a VPI of 46 and a VCI of 47

```
host1#show atm vc atm 2/0 vpi-vci 46 47
ATM 2/0.1.1: VCD: 45, VPI: 46, VCI: 47, Encap: AAL5-AUTO
Service Type: Ubr
Inverse ARP enable:No
Assigned VC class :premium-subscriber-class
InPackets:         0
InBytes:           0
InCells:           0
OutPackets:        0
OutBytes:          0
OutCells:          0
InErrors:          0
OutErrors:         0
InPacketDiscards:  0
InPacketUnknownProtocol: 0
InByteDiscards:    0
CrcErrors:         0
SAR time-outs:     0
Over-sized SDUs:   0
Alarm drop count:  1
Alarm clear timeout:3
OAM VC verification: enabled
OAM loopback cell status: sent
```

```
OAM VC status: up
OAM loopback frequency: 10 second interval
OAM up retry count: 3, OAM down retry count: 5
OAM loopback retry frequency: 1 second interval
OAM CC verification: disabled
InOamF5Cells:      258
InOamCellDiscards: 12598
InF5EndLoopCells:  258
  InF5EndLoopCommands:    50
  InF5EndLoopResponses:   208
InF5SegLoopCells:  46
  InF5SegLoopCommands:    17
  InF5SegLoopResponses:   29
InF5EndAisCells:        49
InF5SegAisCells:        0
InF5EndRdiCells:        0
InF5SegRdiCells:        0
InF5EndCCActDeActCells: 0
InF5SegCCActDeActCells: 0
InF5EndCCCells:         0
InF5SegCCCells:         0
OutOamF5Cells:     258
OutF5EndLoopCells: 258
  OutF5EndLoopCommands:    208
  OutFEndLoopResponses:    50
OutF5SegLoopCells: 48
  OutF5SegLoopCommands:    19
  OutF5SegLoopResponses:   29
OutF5EndRdiCells:       50
OutF5SegRdiCells:       0
OutF5EndCCActDeActCells:1
OutF5SegCCActDeActCells:0
OutF5EndCCCells:        1
OutF5SegCCCells:        0

Circuit is Up, time since last change: 5 days, 23 hours
```

- Example 2—Displays statistics for the VC that resides on the ATM 1483 subinterface configured with the specified description (myAtm301)

```
host1#show atm vc myAtm301
ATM3/0.1: VCD: 10, VPI: 5, VCI: 100, Encap: SNAP
Service Type: Ubr
Assigned VC class         :dsl-subscriber-class
InPackets:                 0
InBytes:                   0
InCells:                   0
OutPackets:                0
OutBytes:                  0
OutCells:                  0
InErrors:                  0
OutErrors:                 0
InPacketDiscards:          0
InPacketUnknownProtocol:   0
InByteDiscards:            0
CrcErrors:                 0
SAR time-outs:             0
Over-sized SDUs:           0
Alarm drop count:          1
Alarm clear timeout:       3
OAM VC verification:       disabled
OAM VC status:             not managed
```



```
OAM CC verification:       disabled
InOamF5Cells:              0
InOamCellDiscards:         384723
InF5EndLoopCells:          0
  InF5EndLoopCommands:     0
  InF5EndLoopResponses:    0
InF5SegLoopCells:          0
  InF5SegLoopCommands:     0
  InF5SegLoopResponses:    0
InF5EndAisCells:           0
InF5SegAisCells:           0
InF5EndRdiCells:           0
InF5SegRdiCells:           0
InF5EndCCActDeActCells:    0
InF5SegCCActDeActCells:    0
InF5EndCCCells:            0
InF5SegCCCells:            0
OutOamF5Cells:             0
OutF5EndLoopCells:         0
  OutF5EndLoopCommands:    0
  OutFEndLoopResponses:    0
OutF5SegLoopCells:         0
  OutF5SegLoopCommands:    0
  OutF5SegLoopResponses:   0
OutF5EndRdiCells:          0
OutF5SegRdiCells:          0
OutF5EndCCActDeActCells:   0
OutF5SegCCActDeActCells:   0
OutF5EndCCCells:           0
OutF5SegCCCells:           0

Circuit is DOWN, time since last change: 02:25:52
```

- See *show atm vc atm*.

show atm vc-class

- Use to display information about the VC classes configured on the router.
- To display only the names of all VC classes configured on the router, use the command with no keywords.
- To display detailed configuration information about a particular VC class, specify the name of the VC class.
- To display the settings for parameters in the VC class that are configured with default values, use the **include-defaults** keyword.
- Field descriptions
  - Encapsulation Type—Encapsulation method configured in the VC class: AUTO, AAL5, AAL0, MUX, SNAP
  - Service Category—Service category configured in the VC class: UBR, UBR-PCR, NRT-VBR, RT-VBR, CBR
  - Peak Cell Rate—Peak cell rate (PCR), in Kbps, configured for the service category
  - OAM VC Integrity—Status of F5 OAM VC integrity features on the PVC: enabled or disabled

- OAM VC Integrity loop-back timer—Number of seconds the router waits between the transmission of loopback cells during normal operation
- OAM VC Integrity Up Retry Count—Number of successive loopback cell responses that the router receives before reporting that a PVC is up
- OAM VC Integrity Down Retry Count—Number of successive loopback cell responses that the router misses before reporting that a PVC is down
- OAM VC Integrity Retry Frequency—Number of seconds that the router waits between the transmission of loopback cells when it is verifying the state of a PVC
- OAM alarm down count—Number of successive alarm cells that the router receives before reporting that a PVC is down
- OAM alarm clear time out—Number of seconds that the router waits before reporting that a PVC is up after the PVC has stopped receiving alarm cells
- OAM continuity check—Status of F5 OAM continuity check verification on the PVC: enabled or disabled
- Inverse ARP—Status of Inverse ARP (InARP) on the PVC: enabled or disabled

- Example 1

```
host1#show atm vc-class
premium-subscriber-class
dsl-subscriber-class
found 2 VC class entries in the system
```

- Example 2

```
host1#show atm vc-class premium-subscriber-class
Encapsulation Type                 :AUTO
Service Category                   :CBR
Peak Cell Rate                     :200 kbps
OAM VC Integrity                   :enabled
OAM VC Integrity loop-back timer    :60 seconds
OAM alarm down count               :5
```

- Example 3

```
host1#show atm vc-class premium-subscriber-class include-defaults
Encapsulation Type                 :AUTO
Service Category                   :CBR
Peak Cell Rate                     :200 kbps
OAM VC Integrity                   :enabled
OAM VC Integrity loop-back timer    :60 seconds
OAM VC Integrity Up Retry Count     :3
OAM VC Integrity Down Retry Count   :5
OAM VC Integrity Retry Frequency    :1
OAM alarm down count               :5
OAM alarm clear time out           :3 seconds
OAM continuity check               :disabled
Inverse ARP                        :disabled
```

- See *show atm vc-class*.

show atm vp

- Use to display detailed statistics for a specific ATM VP configured on the router.
- Field descriptions
  - ServiceCategory—Service type on the VP tunnel, if configured: UBR, UBR-PCR, VBR-NRT, VBR-RT, or CBR
  - Peak Cell Rate—Peak cell rate in kilobits per second, if a VP tunnel is configured
  - Maximum VCI per VPI—Maximum number of virtual circuits on each virtual path
  - Current VCs—Number of VCs currently configured on the router
  - InPackets—Number of packets received
  - InBytes—Number of bytes received
  - InCells—Number of ATM cells received
  - OutPackets—Number of packets transmitted
  - OutBytes—Number of bytes transmitted
  - OutCells—Number of ATM cells transmitted
  - InErrors—Number of packets with errors received
  - OutErrors—Number of packets not transmitted on this VP due to errors
  - InPacketDiscards—Number of incoming packets discarded
  - InPacketUnknownProtocol—Number of incoming packets with an unknown protocol type
  - InByteDiscards—Number of incoming bytes discarded
  - CrcErrors—Number of CRC errors detected
  - SAR time-outs—Number of segmentation and reassembly (SAR) timeouts reached
  - Over-sized SDUs—Number of oversized service data units (SDUs) received
  - The following fields appear only if F4 OAM is enabled on the VP:
    - Sending End to End Loopback Cells—Enabled, Disabled
    - End to End OAM CC verification—Enabled, Disabled
    - VP State—State of the VP: up, down
    - VP Oam State—OAM state of the VP: not managed (normal OAM state with no OAM fault conditions), AIS, RDI
    - InOamF4EndCells—Number of F4 end-to-end cells received
    - InOamF4EndCellsDropped—Number of incoming F4 end-to-end cells that were dropped
    - InOamF4EndLoopbackCells—Number of F4 end-to-end loopback cells received
    - InOamF4EndLoopbackCommands—Number of F4 end-to-end loopback commands received

- InOamF4EndLoopbackResponses—Number of F4 end-to-end loopback responses received
- InOamF4EndAisCells—Number of F4 end-to-end AIS cells received
- InOamF4EndRdiCells—Number of F4 end-to-end RDI cells received
- InOamF4EndCCActDeActCells—Number of F4 end-to-end activation or deactivation CC cells received
- InOamF4EndCCCells—Number of F4 end-to-end CC cells received
- OutOamF4EndCells—Number of F4 end-to-end CC cells transmitted
- OutOamF4EndLoopbackCells—Number of F4 end-to-end loopback cells transmitted
- OutOamF4EndLoopbackCommands—Number of F4 end-to-end loopback commands transmitted
- OutOamF4EndLoopbackResponses—Number of F4 end-to-end loopback responses transmitted
- OutOamF4EndRdiCells—Number of F4 end-to-end RDI cells transmitted
- OutOamF4EndCCActDeActCells—Number of F4 end-to-end activation or deactivation CC cells transmitted
- OutOamF4EndCCCells—Number of F4 end-to-end CC cells transmitted
- Time since last status change—Time since last reported change to the end-to-end OAM circuit status
- Segment OAM CC verification—Enabled or Disabled
- VP State—State of the VP: up, down
- VP Oam State—OAM state of the VP: not managed (normal OAM state with no OAM fault conditions), AIS, RDI
- InOamF4SegmentCells—Number of F4 segment cells received
- InOamF4SegmentCellsDropped—Number of incoming F4 segment cells that were dropped
- InOamF4SegmentLoopbackCells—Number of F4 segment loopback cells received
- InOamF4SegmentLoopbackCommands—Number of F4 segment loopback commands received
- InOamF4SegmentLoopbackResponses—Number of F4 segment loopback responses received
- InOamF4SegCCActDeActCells—Number of F4 segment activation or deactivation CC cells received
- InOamF4SegCCCells—Number of F4 segment CC cells received
- OutOamF4SegmentCells—Number of F4 segment cells transmitted

- OutOamF4SegmentLoopbackCells—Number of F4 segment loopback cells transmitted
- OutOamF4SegmentLoopbackCommands—Number of F4 segment loopback commands transmitted
- OutOamF4SegmentLoopbackResponses—Number of F4 segment loopback responses transmitted
- OutOamF4SegRdiCells—Number of F4 segment RDI cells transmitted
- OutOamF4SegCCActDeActCells—Number of F4 segment activation or deactivation CC cells transmitted
- OutOamF4SegCCCells—Number of F4 segment CC cells transmitted
- Time since last status change—Time since last reported change to the segment OAM circuit status

- VP Description—Text description for this VP, if configured

- Example

```
host1#show atm vp atm 12/0 1
Maximum VCI per VPI: 65535     Current VCs: 3
InPackets                        :1604710953
InBytes                          :205403001984
InCells                          :519165564
OutPackets                       :1604632002
OutBytes                         :205392896256
OutCells                         :4813896006
InErrors                         :0
OutErrors                        :0
InPacketDiscards                 :0
InPacketUnknownProtocol          :0
InByteDiscards                   :0
CrcErrors                        :0
SAR time-outs                    :0
Over-sized SDUs                  :0
Sending End To End Loopback Cells Disabled:
End To End OAM CC verification Disabled
VP State                         :up
VP Oam State                     :not managed
InOamF4EndCells                  :0
InOamF4EndCellsDropped           :0
InOamF4EndLoopbackCells          :0
  InOamF4EndLoopbackCommands       :0
  InOamF4EndLoopbackResponses      :0
InOamF4EndAisCells               :0
InOamF4EndRdiCells               :0
InOamF4EndCCActDeActCells        :0
InOamF4EndCCCells                :0
OutOamF4EndCells                 :0
OutOamF4EndLoopbackCells         :0
  OutOamF4EndLoopbackCommands      :0
  OutOamF4EndLoopbackResponses     :0
OutOamF4EndRdiCells              :0
OutOamF4EndCCActDeActCells       :0
OutOamF4EndCCCells               :0
Time since last status change    :08:48:43
Segment OAM CC verification Disabled
```



```
VP State                          :up
VP Oam State                       :not managed
InOamF4SegmentCells               :0
InOamF4SegmentCellsDropped        :0
InOamF4SegmentLoopbackCells       :0
  InOamF4SegmentLoopbackCommands    :0
  InOamF4SegmentLoopbackResponses   :0
InOamF4SegCCActDeActCells         :0
InOamF4SegCCCells                 :0
OutOamF4SegmentCells              :0
OutOamF4SegmentLoopbackCells      :0
  OutOamF4SegmentLoopbackCommands   :0
  OutOamF4SegmentLoopbackResponses :0
OutOamF4SegRdiCells               :0
OutOamF4SegCCActDeActCells        :0
OutOamF4SegCCCells                :0
Time since last status change     :08:48:44
VP Description: ATM-12/0-VPI-1
```

- See *show atm vp*.

show atm vp-description

- Use to display VP descriptions configured using the **atm vp-description** command.
- To display all VP descriptions configured on the router, issue the command without an ATM identifier or VPI number (Example 1).
- To display all VP descriptions for a particular ATM interface, specify the ATM interface identifier without the VPI number (Example 2).
- To display the VP description for a particular VPI, specify both the ATM interface identifier and the VPI number (Example 3).
- Field descriptions
  - Interface—ATM interface identifier
  - VPI—Virtual path identifier
  - Description—Text description configured for the VP
- Example 1—Displays all VP descriptions configured on the router

  ```
  host1#show atm vp-description
  Interface    VPI    Description
  ATM 2/0      0      atm20Vpi0Subscribers
  ATM 2/0      1      atm20Vpi1Subscribers
  ATM 2/1      0      atm21Vpi0Subscribers
  ```

- Example 2—Displays all VP descriptions for the specified ATM interface

  ```
  host1#show atm vp-description atm 2/0
  Interface    VPI    Description
  ATM 2/0      0      atm20Vpi0Subscribers
  ATM 2/0      1      atm20Vpi1Subscribers
  ```

- Example 3—Displays the VP description for the specified VPI

```
host1#show atm vp-description atm 2/0 1
Interface    VPI    Description
ATM 2/0      1      atm20Vpi1Subscribers
```

- See *show atm vp-description*.

show atm vp-tunnel

- Use to display a summary of all configured ATM virtual path tunnels.
- Field descriptions
  - Intfc—Interface number
  - VPI—Virtual path identifier
  - Type—VP tunnel traffic management type
  - Kbps—Rate, in Kbps
  - Description—Text description for the VP, if configured
- Example

```
host1#show atm vp-tunnel 9/1
Intfc    VPI  Type  Kbps    Description
ATM 9/1  2    Cbr   4096    atm91Vpi2Subscribers
```

- See *show atm vp-tunnel*.

show mpls cross-connects atm

- Use to display all ATM cross-connects (passthrough connections between local subinterfaces).
- See *Monitoring ATM Cross-Connects for Layer 2 Services over MPLS* in *JunosE BGP and MPLS Configuration Guide* for information about using the **show mpls cross-connects atm** command.
- See *show mpls cross-connects atm*.

show nbma arp

- Use to display ARP table entries for ATM NBMA interfaces.
- Field descriptions
  - IP Address—IP address of the entry
  - VPI/VCI—VPI and VCI of the entry
  - Interface—Interface specifier of the entry
- Example

```
host1#show nbma arp
                NBMA ARP Table Entries
IP Address              VPI/VCI         Interface
1.1.1.2                 0/100           4/1
2.2.2.2                 0/101           4/1
```

- See *show nbma arp*.

CHAPTER 2

Configuring Frame Relay

This chapter describes how to configure a Frame Relay interface on E Series routers.

This chapter contains the following sections:

- Overview on page 103
- Platform Considerations on page 105
- References on page 106
- Before You Configure Frame Relay on page 106
- Configuring Frame Relay on page 106
- Configuring IPv6 over Frame Relay Interfaces on page 114
- End-to-End Fragmentation and Reassembly on page 116
- Monitoring Frame Relay on page 120

Overview

Frame Relay is a public, connection-oriented packet service based on the core aspects of the Integrated Services Digital Network (ISDN). Frame Relay eliminates all processing at the network layer and greatly restricts data-link layer processing. Such simplified processing is possible because of the availability of virtually error-free physical connections and the presence of intelligent protocols at the end-user site, which can detect and retransmit faulty or discarded packets.

Frame Relay shifts responsibility for error recovery and flow control to the end user, thereby reducing protocol complexity and allowing high-speed packet delivery with low transit delay.

For a list of the modules on which you can configure Frame Relay, see *ERX Module Guide, Appendix A, Module Protocol Support*.

Framing

E Series routers support the following framing features:

- HDLC for data-link framing
- 2-byte addresses only

- 8188-byte information field size (8192 minus 2 bytes for the address and a 16-bit CRC) or 8186-byte information field size (8192 minus 2 bytes for the address and a 32-bit CRC)

The router does not support:

- Protocol-dependent fragmentation
- Autodetection of the Local Management Interface (LMI) protocol type

Error Frames

The router relies on higher-layer protocols to detect and recover from Frame Relay data loss. All Frame Relay error frames are discarded.

Unicast and Multicast Addressing

Most Frame Relay services support both unicast (individual) and multicast (group) addressing. Under the most common implementation of multicasting, the Frame Relay network maps multiple individual addresses to a single multicast data-link connection identifier (DLCI) and delivers copies of a single Frame Relay packet to each member of the group.

NOTE: The E Series router supports only unicast addressing.

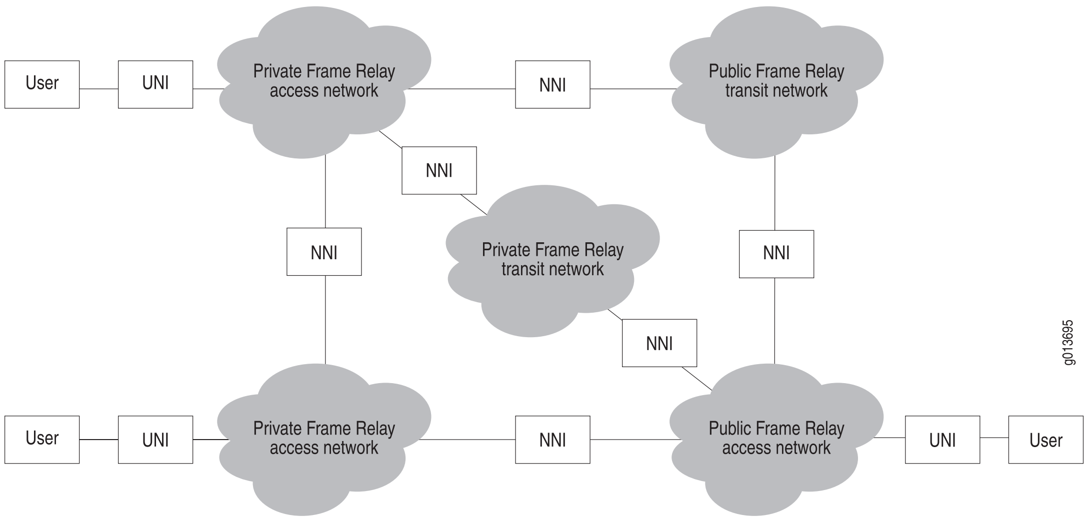
User-to-Network and Network-to-Network Interfaces

The Frame Relay User-to-Network Interface (UNI) is a protocol that permits users to access private or public Frame Relay networks and to establish a communications path to another user within the same network.

The Network-to-Network Interface (NNI) makes connections possible between users connected to different Frame Relay networks. These separate Frame Relay networks can be considered as subnetworks within a complete network service.

Figure 4 on page 105 shows the interconnection of these types of subnetworks and the location of NNI between them.

Figure 4: Interconnection and Relationship of NNIs and Subnetworks



Platform Considerations

You can configure Frame Relay interfaces on the following E Series Broadband Services Routers:

- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

NOTE: The E120 and E320 Broadband Services Routers do not support configuration of Frame Relay interfaces.

Module Requirements

For information about the modules that support Frame Relay interfaces on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support Frame Relay.

Interface Specifiers

The interface specifier format that you use depends on the type of physical interface on which you want to configure Frame Relay.

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

References

For more information about Frame Relay, consult the following resources:

- RFC 2115—Management Information Base for Frame Relay DTEs Using SMIv2 (September 1997)
- RFC 2863—The Interfaces Group MIB (June 2000)
- RFC 2427—Multiprotocol Interconnect over Frame Relay (September 1998)
- Frame Relay Forum—User-to-Network Implementation Agreement (UNI), FRF 1.1 (January 1996)
- Frame Relay Forum—Frame Relay Fragmentation Implementation Agreement, FRF.12 (December 1997)
- ANSI T1.617 Annex D
- ITU-T Recommendation Q.922, Integrated Services Digital Network (ISDN) Data Link Layer Specification for Frame Mode Bearer Services; Annex A (February 1992)
- ITU-T Q.933 Annex A

Before You Configure Frame Relay

Before you attempt to configure a Frame Relay interface, configure the physical line interface over which Frame Relay traffic flows.

This process is described in the following chapters:

- *Configuring Channelized T3 Interfaces* in *JunosE Physical Layer Configuration Guide*
- *Configuring T3 and E3 Interfaces* in *JunosE Physical Layer Configuration Guide*

The procedures described in this chapter assume that a physical interface has been configured.

Configuring Frame Relay

Configure a Frame Relay interface by entering Interface Configuration mode. The procedure that follows is an example of a Frame Relay configuration on a serial interface. All tasks are mandatory unless otherwise noted.

To configure a Frame Relay interface:

1. From Configuration mode, enter the physical interface on which you want to configure Frame Relay.

   host1(config)#**interface serial 3/1:2/1**

2. Select Frame Relay as the encapsulation method for the interface.

host1(config-if)#**encapsulation frame-relay ietf**

3. (Optional) Assign a text description or an alias to the major interface.

   host1(config-if)#**frame-relay description boston01**

4. (Optional) Enable SNMP link status processing on the major interface.

   host1(config-if)#**snmp trap frame-relay link-status**

5. Configure the interface as a DTE, DCE, or NNI.

   host1(config-if)#**frame-relay intf-type dte**

6. Configure the LMI type.

   host1(config-if)#**frame-relay lmi-type ansi**

7. (Optional) Configure Frame Relay counters and timers.

   host1(config-if)#**frame-relay lmi-n391dte 20**

8. Configure the cyclic redundancy check (CRC).

   host1(config-if)#**crc 32**

9. Create a subinterface.

   host1(config)#**interface serial 3/1:2/1.1**

10. (Optional) Assign a text description or an alias to the subinterface.

    host1(config-subif)#**frame-relay description westford011**

11. (Optional) Enable SNMP link status processing on the subinterface.

    host1(config-subif)#**snmp trap frame-relay link-status**

12. Add a circuit to a subinterface.

    host1(config-subif)#**frame-relay interface-dlci 17 ietf**

13. Assign a local IP address to the circuit.

    host1(config-subif)#**ip address 192.32.10.2 255.255.255.0**

14. (Optional) Use **show** commands to verify that your configuration changes are correct by checking the state of the interfaces.

    host1#**show frame-relay lmi**
    host1#**show frame-relay map**
    host1#**show frame-relay pvc**

15. (Optional) Disable the local management interface.

    host1#**no** frame-relay keepalive

16. (Optional) Disable the interface.

    host1(config-if)#**shutdown**

crc

- Use to set the number of bits used for CRC.
- The CRC is an error-checking technique that uses a calculated numeric value to detect errors in transmitted data.

- 16 and 32 indicate the number of bits per frame that are used to calculate the frame check sequence (FCS).
- A 32-bit CRC transmits longer streams at faster rates and therefore provides better ongoing error detection, such as for an OC12/STM4 POS module.
- The default is 16. You must configure CRC (CRC16 or CRC32) to match the configuration on the other side of the Frame Relay connection.
- Example

  host1(config-if)#**crc 32**
- Use the **no** version to set the CRC to the default value.
- See *crc*.

encapsulation frame-relay ietf

- Use to specify Frame Relay as the encapsulation method for the interface.
- The router uses IETF format (RFC 2427 encapsulation).
- Example

  host1(config-if)#**encapsulation frame-relay ietf**
- Use the **no** version to remove Frame Relay configuration from an interface.
- See *encapsulation frame-relay ietf*.

frame-relay description

- Use to assign a text description or an alias to a Frame Relay interface or subinterface.
- You can use this command to help you identify the interface and keep track of interface connections.
- The description or alias can be a maximum of 80 characters.
- Use "show frame-relay interface" on page 121 or "show frame-relay subinterface" on page 127 to display the text description.
- Examples

  host1(config-if)#**frame-relay description boston01**
  host1(config-subif)#**frame-relay description toronto011**
- Use the **no** version to remove the text description or alias.
- See *frame-relay description*.

frame-relay interface-dlci ietf

- Use to configure a Frame Relay permanent virtual circuit (PVC) over a subinterface.
- The **ietf** keyword is mandatory and indicates RFC 2427 encapsulation.
- Define a DLCI in the range 16–1007.
- To configure a Frame Relay PVC, you must specify a DLCI.

- Frame Relay service is offered in the form of PVCs. A PVC is a data-link connection that is predefined on both ends of the connection. A network operator assigns the endpoints of the circuit. Although the actual path taken through the network may vary from time to time, the beginning and end of the circuit do not change. This type of circuit behaves like a dedicated point-to-point circuit.
- PVCs are identified by DLCIs. A DLCI is a 10-bit channel number that is attached to data frames to tell a Frame Relay network how to route the data. Frame Relay is statistically multiplexed, which means that only one frame can be transmitted at a time, but many logical connections can coexist on a single physical line. The DLCI allows the data to be logically tied to one of the connections, so that when the data gets to the network, the network knows where to send it.
- DLCIs on the same physical line must match. However, DLCIs have local significance; that is, if the DLCIs are not on the same physical line, the end devices at two different ends of a connection may use a different DLCI to refer to the same connection.
- The router does not support switched virtual circuits (SVCs). An SVC is an any-to-any connection that can be established or removed as needed. With SVCs, you initiate calls using Frame Relay by requesting a destination address and assigning a DLCI, which is established for the duration of the call.
- Example

  host1(config-subif)#**frame-relay interface-dlci 17 ietf**
- Use the no version to remove DLCI/PVC assignment.
- See *frame-relay interface-dlci ietf*.

frame-relay intf-type

- Use to configure a Frame Relay interface circuit to operate as data communications equipment (DCE), data terminal equipment (DTE), or NNI.
- Frame Relay provides packet-switching data communications between user devices and network equipment across the interface. User devices are referred to as DTE.
- Network equipment that interfaces with a DTE is referred to as a DCE.
- NNI provides a connection between two Frame Relay subnetworks.
- If your router is connected to a Frame Relay switch, configure the interface as a DTE. If your router is connected by a point-to-point line, configure one end as the DTE and the other as the DCE.
- Example

  host1(config-if)#**frame-relay intf-type dte**
- Use the **no** version to set the default of DTE.
- See *frame-relay intf-type*.

frame-relay keepalive

- Use to enable the LMI on the interface.
- You can specify the keepalive interval in seconds.

- Make sure the value on the DTE is less than the value set on the DCE.
- The default is 10 seconds.
- Example

  host1#**no** frame-relay keepalive
- Use the **no** version to disable LMI on the interface.
- See *frame-relay keepalive*.

frame-relay lmi-n391dte

frame-relay lmi-n392dce

frame-relay lmi-n392dte

frame-relay lmi-n393dce

frame-relay lmi-n393dte

frame-relay lmi-t391dte

frame-relay lmi-t392dce

- Use to configure LMI counters and timers.
- LMI counters and timers have configurable ranges that allow you to control the state of the Frame Relay interface. In general, accept the default values for the timers and counters, unless you need to modify them according to a special arrangement with your customers.
- Some commands have DTE and DCE versions. Use the dte version of the command if the interface is operating as a DTE. Use the **dce** version of the command if the interface is operating as a DCE. Use both versions of the command if the interface is operating as an NNI.
- Use the **frame-relay lmi-n391dte** command to set the N391 full-status polling counter. When you set this counter to a number, *n*, the *n*th request is a full-status request. The range is 1–255 event messages. The default is 6 event messages.
- Use the **frame-relay lmi-n392dte** or **frame-relay lmi-n392dce** command to set the N392 error threshold counter, which specifies the number of errors within N393 events that will place the interface in an operationally down state. The range is 1–10. The default for the DTE version is 3. The default for the DCE version is 2.
- Use the **frame-relay lmi-n393dte** or **frame-relay lmi-n393dce** command to set the N393 monitored events counter to specify the diagnostic window used to verify link integrity. Detection of N392 errors within the window of N393 samples places the interface in an operationally down state. The range is 1–10 events. The default for the DTE version of the command is 4 events. The default for the DCE version is 2 events.
- Use the **frame-relay lmi-t391dte** command to set the T391 link integrity polling timer interval between status inquiries issued by the DTE. The network checks that the DTE polls within the verification interval, T392. The range is 5–30 seconds. The default is 10 seconds.

- Use the **frame-relay lmi-t392dce** command to set the T392 polling verification timer that specifies the maximum interval (in seconds) between the receipt of status inquiries from the DTE equipment before it considers it as an error event. The range is 5–30 seconds. The default is 15 seconds.
- Example

  host1(config-if)#**frame-relay lmi-n391dte 20**
- Use the **no** version to remove the current setting and set the default.
- See *frame-relay lmi-n391dte*.
- See *frame-relay lmi-n392dce*.
- See *frame-relay lmi-n392dte*.
- See *frame-relay lmi-n393dce*.
- See *frame-relay lmi-n393dte*.
- See *frame-relay lmi-t391dte*
- See *frame-relay lmi-t392dce*

frame-relay lmi-type

- Use to configure one of the local management interface types.
- LMI provides configuration and status information relating to the virtual circuits operating over Frame Relay.
- LMI specifies polling mechanisms to receive incremental and full-status updates from the network.
- E Series routers conform to the following LMI specifications:
  - **ansi**—ANSI T1.617 Annex D
  - **q933a**—ITU-T Q.933 Annex A
  - **cisco**—Original *Group of Four* specification developed by DEC, Northern Telecom, Stratacom, and Cisco
  - **none**—Suppresses LMI
- The default is **cisco**.
- Example

  host1(config-if)#**frame-relay lmi-type ansi**
- Use the **no** version to return to the default LMI type.
- See *frame-relay lmi-type*.

interface pos

- Use to configure a POS interface in slot/port format:
  - *slot*—Router chassis slot

  - *port*—Line module port
- Example

  host1(config)#**interface pos 0/1**
- Use the **no** version to remove the POS interface.
- See *interface pos*.

interface serial

- Use to configure a serial interface in the appropriate format by selecting a previously configured physical interface on which you want to configure Frame Relay. For example, for a channelized T3 interface use *slot/port:channel/subchannel*.
- Use to configure a Frame Relay subinterface in the appropriate format by selecting a previously configured physical interface. For example, for a T3-Frame interface use *slot/port.subinterface* ; for a channelized T1/channelized E1 interface use *slot/port.channel-group.subinterface*.

NOTE: **Before you configure Frame Relay, see the appropriate chapter in this guide for details on configuring physical interfaces.**

  - *slot*—Router chassis slot
  - *port*—CT3, T3, or E3 module I/O port
  - *channel*—T1 (DS1) channel
  - *subchannel*—Set of DS0 timeslots. See *Fractional T1* in *JunosE Physical Layer Configuration Guide*
  - *subinterface*—User-assigned nonnegative number that identifies a Frame Relay subinterface
- Example

  host1(config-if)#**interface serial 3/1:2/1.1**
- Use the **no** version to remove the subinterface or the serial interface.
- See *interface serial*.

ip address

- Use to assign an IP address and subnet mask to a subinterface.
- Example

  host1(config-subif)#**ip address 192.32.10.2 255.255.255.0**
- Use the **no** version to remove an IP address or to disable IP processing.
- See *ip address*.

pos description

- Use to assign a text description or an alias to a POS HDLC interface.
- You can use this command to help you identify the interface and keep track of interface connections.
- The description or alias can be a maximum of 80 characters.
- Use "show interfaces pos" on page 383 to display the text description. For details, see "Monitoring POS" on page 382 in "Configuring Packet over SONET" on page 375.
- Example

  host1(config-if)#**pos description austin01 pos interface**
- Use the **no** version to remove the text description or alias.
- See *pos description*.

serial description

- Use to assign a text description or an alias to a serial HDLC interface.
- You can use this command to help you identify the interface and keep track of interface connections.
- The description or alias can be a maximum of 80 characters.
- Use the **show interfaces serial** command to display information about the serial interfaces you configured. For more information about the descriptions of the fields displayed in the output of this command, see the *show interfaces serial* section in *JunosE Physical Layer Configuration Guide*. For example, for a channelized T3 interface, see the *Monitoring Interfaces* section in *JunosE Physical Layer Configuration Guide*.
- Example

  host1(config-if)#**serial description ottawa012 hdlc channel**
- Use the **no** version to remove the text description or alias.
- See *serial description*.

shutdown

- Use to disable a Frame Relay interface.
- Example

  host1(config-if)#**shutdown**
- Use the **no** version to restart a disabled interface.
- See *shutdown*.

snmp trap frame-relay link-status

- Use to enable SNMP link status processing for a Frame Relay major interface or subinterface.
- To enable SNMP link status processing for a Frame Relay major interface, you must issue the command from Interface Configuration mode.

- To enable SNMP link status processing for a Frame Relay subinterface, you must issue the command from Subinterface Configuration mode.
- Examples

  host1(config-if)#snmp trap frame-relay link-status
  host1(config-subif)#snmp trap frame-relay link-status
- Use the **no** version to disable SNMP link status processing for a Frame Relay major interface or subinterface.
- See *snmp trap frame-relay link-status*.

Configuring IPv6 over Frame Relay Interfaces

You can configure IPv6 prefix addresses on Frame Relay interfaces on a packet over SONET (POS) physical interface. IPv6 traffic is forwarded over Frame Relay circuits and the database of the forwarding controller is updated with IPv6 as an upper-layer interface to the Frame Relay layer.

You can specify IPv6 prefixes only on Frame Relay interfaces over POS major interfaces. You cannot configure IPv6 addresses on E3 and T3 interfaces. Because IPv6 neighbor discovery is not supported on POS interfaces, you cannot use a provider edge router that contains Frame Relay or MLFR interfaces configured with IPv6 addresses to respond to route solicitation packets it receives from clients in environments in which the subscriber is either an IPv6 subscriber or a combined IPv4 and IPv6 subscriber in a dual stack.

The following sections describe how to configure IPv6 addresses for Frame Relay interface circuits on POS interfaces that operate as DCE, DTE, or NNI and for Frame Relay PVCs over a POS subinterface:

- Configuring an IPv6 Address for a Frame Relay That Operates as a DCE, DTE, or NNI on page 114
- Configuring an IPv6 Address for a Frame Relay PVC on page 115

Configuring an IPv6 Address for a Frame Relay That Operates as a DCE, DTE, or NNI

To configure an IPv6 prefix address for a Frame Relay interface over a POS interface that operates as a DCE, DTE, or NNI:

1. From Configuration mode, enter a POS interface on which you want to configure Frame Relay.

   host1(config)#interface pos 0/1
2. Select Frame Relay as the encapsulation method for the interface.

   host1(config-if)#encapsulation frame-relay ietf
3. (Optional) Assign a text description or an alias to the major interface.

   host1(config-if)#frame-relay description boston01
4. Configure the interface as a DCE, DTE, or NNI.

host1(config-if)#**frame-relay intf-type dte**

5. Assign a local IP address to the circuit.

   host1(config-subif)#**ipv6 address 1::1/64**

Configuring an IPv6 Address for a Frame Relay PVC

To configure an IPv6 prefix address for a Frame Relay PVC over a POS subinterface:

1. From Configuration mode, enter a POS interface on which you want to configure Frame Relay.

   host1(config)#**interface pos 0/1**

2. Select Frame Relay as the encapsulation method for the interface.

   host1(config-if)#**encapsulation frame-relay ietf**

3. (Optional) Assign a text description or an alias to the major interface.

   host1(config-if)#**frame-relay description boston01**

4. Configure the LMI type.

   host1(config-if)#**frame-relay lmi-type ansi**

5. (Optional) Configure Frame Relay counters and timers.

   host1(config-if)#**frame-relay lmi-n391dte 20**

6. Configure the cyclic redundancy check (CRC).

   host1(config-if)#**crc 32**

7. Create a subinterface.

   host1(config)#**interface pos 0/1.1**

8. Add a circuit to a subinterface.

   host1(config-subif)#**frame-relay interface-dlci 17 ietf**

9. Assign a local IP address to the circuit.

   host1(config-subif)#**ipv6 address 1::1/64**

interface pos

- Use to configure a POS interface in slot/port format:
  - *slot*—Router chassis slot
  - *port*—Line module port
- Example

  host1(config)#**interface pos 0/1**
- Use the **no** version to remove the POS interface.
- See *interface pos*.

ipv6 address

- Use to add an IPv6 address to an interface or a subinterface.
- Example

```
host1(config)#interface gigabitEthernet 1/0.25
host1(config-if)#ipv6 address 1::1/64
```

- Use the **no** version to remove an IPv6 address.
- See *ipv6 address*.

End-to-End Fragmentation and Reassembly

The fragmentation and reassembly feature reduces excessive delays of Frame Relay packets by breaking them up into smaller fragments and interleaving them with real-time frames. By doing this, real-time and non-real-time data frames can be carried together on lower-speed links without causing excessive delays to the real-time traffic. On receiving the smaller fragments by the peer interface, the fragments are reassembled into their original packet. For example, short delay-sensitive packets, such as packetized voice, can race ahead of larger delay-insensitive packets, such as common data packets.

E Series routers support end-to-end fragmentation according to the FRF.12 Implementation Agreement standard. Unlike UNI and NNI fragmentation, end-to-end supports fragmentation only at the endpoints. End-to-end fragmentation and reassembly are supported only on non-multilink Frame Relay interfaces on cOC12/STM4 and CT3 12 FO modules.

You configure end-to-end fragmentation at the Frame Relay subinterface level. Fragmentation is applied to all PVCs associated with the subinterface. In most cases, fragmentation and reassembly are used together. Fragmentation and reassembly, however, can be configured separately for each map class.

For additional information, see Frame Relay Forum—Frame Relay Fragmentation Implementation Agreement, FRF.12 (December 1997).

Frame Fragmentation

When you enable fragmentation, you can specify a maximum payload size of the resulting fragments. If the maximum payload size is not specified, the default value of 52 bytes is used. When enabled, fragmentation begins when the portion of the packet that has not been transmitted in previous fragments exceeds the configured maximum payload size. The fragmentation process continues until the entire packet has been transmitted. Frames that do not exceed the configured maximum payload size are not fragmented.

If you disable fragmentation, all packets transmitted by the Frame Relay subinterface are transmitted intact.

Frame Reassembly

When reassembly is disabled and a data frame is received, a few scenarios may occur:

- If the frame is not a fragment, it is forwarded normally.

- If the frame is a fragment and the upper interface is IP (that is, the interface above the Frame Relay subinterface), then the fragment is immediately discarded.

If you enable reassembly, then received fragments undergo the reassembly process. Packets that are not fragments are forwarded as normal.

Map Class

Within Frame Relay, a map class acts as a container or context for fragmentation and reassembly parameters. Within the map class context, you can explicitly enable fragmentation and reassembly.

After you define a map class, you can apply it to an unlimited number of subinterfaces. This allows you to change fragmentation and reassembly parameters one time and have the changes immediately reflected in all subinterfaces configured to use that map class.

Configuring End-to-End Fragmentation

You configure end-to-end fragmentation and reassembly on a subinterface in much the same way you configure a standard Frame Relay interface. In this example, end-to-end fragmentation and reassembly is configured on a single subinterface with a 100-byte fragment size (maximum payload size). All tasks are mandatory unless otherwise noted.

NOTE: **The procedure described in this section assumes that a physical interface has been configured. See** "Before You Configure Frame Relay" on page 106.

To configure end-to-end fragmentation and reassembly:

1. Create a map class that you can apply to subinterfaces.

   host1(config)#**map-class frame-relay testmap**

2. Specify fragmentation and reassembly for the map class. Optionally, you can specify the maximum payload size of a fragment.

   host1(config-map-class)#**frame-relay fragment 100**

3. Enter the physical interface on which you want to configure Frame Relay end-to-end fragmentation and reassembly.

   host1(config-map-class)#**interface serial 5/0:4/1**

4. Select Frame Relay as the encapsulation method for the interface.

   host1(config-if)#**encapsulation frame-relay ietf**

5. Create a subinterface.

   host1(config-if)#**interface serial 5/0:4/1.1**

6. Add a circuit to a subinterface.

   host1(config-subif)#**frame-relay interface-dlci 16 ietf**

7. Assign a local IP address to the circuit.

host1((config-subif)#**ip address 42.42.42.41 255.255.255.0**

8. Associate a map class with a subinterface.

   host1(config-subif)#**frame-relay class testmap**

encapsulation frame-relay ietf

- Use to specify Frame Relay as the encapsulation method for the interface.
- The router uses IETF format (RFC 2427 encapsulation).
- Example

  host1(config-if)#**encapsulation frame-relay ietf**
- Use the **no** version to remove Frame Relay configuration from an interface.
- See *encapsulation frame-relay ietf*.

frame-relay class

- Use to associate a map class with a subinterface.
- Example

  host1(config-subif)#**frame-relay class testmap**
- Use the **no** version to remove the association between the subinterface and the specified map class from the subinterface.
- See *frame-relay class*.

frame-relay fragment

- Use to configure fragmentation and reassembly for the map class.
- Specify the keyword **fragmentation-only** to specify only fragmentation, so that reassembly is not performed.
- Specify the keyword **reassembly-only** to specify only reassembly, so that fragmentation is not performed.
- Specify the maximum payload size of a fragment by using a value from 16–8188 bytes. If a value is not specified, the default value of 52 bytes is used.
- Make sure the value for the maximum payload size of a fragment is less than or equal to the MTU size, otherwise fragmentation never occurs.
- Make sure the maximum payload size is larger than any voice packet so that voice frames are not fragmented.
- Examples

  host1(config-map-class)#**frame-relay fragment 100**
  host1(config-map-class)#**frame-relay fragment fragmentation-only**
- Use the **no** version to stop fragmentation and reassembly on the subinterface.
- See *frame-relay fragment*.

frame-relay interface-dlci ietf

- Use to configure a Frame Relay PVC over a subinterface.
- The **ietf** keyword is mandatory and indicates RFC 2427 encapsulation.
- Define a DLCI in the range 16–1007.
- To configure a Frame Relay PVC, you must specify a DLCI.
- Frame Relay service is offered in the form of PVCs. A PVC is a data-link connection that is predefined on both ends of the connection. A network operator assigns the endpoints of the circuit. Although the actual path taken through the network may vary from time to time, the beginning and end of the circuit do not change. This type of circuit behaves like a dedicated point-to-point circuit.
- PVCs are identified by DLCIs. A DLCI is a 10-bit channel number that is attached to data frames to tell a Frame Relay network how to route the data. Frame Relay is *statistically multiplexed*, which means that only one frame can be transmitted at a time, but many logical connections can coexist on a single physical line. The DLCI allows the data to be logically tied to one of the connections, so that when the data gets to the network, the network knows where to send it.
- DLCIs on the same physical line must match. However, DLCIs have local significance; that is, if the DLCIs are not on the same physical line, the end devices at two different ends of a connection may use a different DLCI to refer to the same connection.
- The router does not support SVCs. An SVC is an any-to-any connection that can be established or removed as needed. With SVCs, you initiate calls using Frame Relay by requesting a destination address and assigning a DLCI, which is established for the duration of the call.
- Example

  host1(config-subif)#**frame-relay interface-dlci 16 ietf**
- Use the **no** version to remove DLCI/PVC assignment.
- See *frame-relay interface-dlci ietf*.

interface serial

- Use to configure a serial interface in the appropriate format by selecting a previously configured physical interface on which you want to configure Frame Relay. For example, for a channelized T3 interface use *slot/port:channel/subchannel*.
- Use to configure a Frame Relay subinterface in the appropriate format by selecting a previously configured physical interface. For example, for a T3-Frame interface use *slot/port.subinterface*; for a channelized T1/channelized E1 interface use *slot/port.channel-group.subinterface*.

NOTE: See “Before You Configure Frame Relay” on page 106 for more information about configuring the underlying physical interfaces.

  - *slot*—Router chassis slot
  - *port*—CT3, T3, or E3 module I/O port

- *channel*—T1 (DS1) channel
- *subchannel*—Set of DS0 timeslots; for information, see *Fractional T1* in *JunosE Physical Layer Configuration Guide*
- *subinterface*—User-assigned nonnegative number that identifies a Frame Relay subinterface

- Example

  host1(config-if)#**interface serial 5/0:4/1.1**

- Use the **no** version to remove the subinterface or the serial interface.
- See *interface serial*.

ip address

- Use to assign an IP address and subnet mask to a subinterface.
- Example

  host1((config-subif)#**ip address 42.42.42.41 255.255.255.0**

- Use the **no** version to remove an IP address or to disable IP processing.
- See *ip address*.

map-class frame-relay

- Use to create a map class.
- Example

  host1(config)#**map-class frame-relay testmap**

- Use the **no** version to remove a map class.
- See *map-class frame-relay*.

Monitoring Frame Relay

Use the **show frame-relay** commands described in this section to monitor Frame Relay interfaces.

You can set a statistics baseline for Frame Relay interfaces, subinterfaces, or circuits using the **baseline frame-relay** command.

You can use the output-filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *show Commands* in *JunosE System Basics Configuration Guide* for details.

If you do not specify an interface type for the appropriate show command, the output indicates whether a serial or POS interface is being displayed.

baseline frame-relay interface

- Use to set a statistics baseline at the Frame Relay layer for multilink Frame Relay, POS, serial or GRE tunnel interfaces, subinterfaces, or circuits.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Specify an interface or subinterface using the interface type and specifier. For more information, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.
- Specify a circuit using the interface type and specifier and the **dlci** keyword and the *dlci* number.
- You cannot set a baseline for groups of interfaces, subinterfaces, or circuits. You must set baselines individually.
- When baselining is requested, the time since the last baseline was set is displayed in *hours:minutes:seconds* or *days/hours* format. If a baseline has not been set, the message " No baseline has been set" is displayed instead.
- The regular interface statistics and LMI statistics for interfaces are subject to the same baseline timestamp. You cannot set separate baselines.
- Use the optional **delta** keyword with Frame Relay **show** commands to specify that baselined statistics are to be shown.
- Example

  ```
  host1#baseline frame-relay  interface serial 2/0:1/1
  ```

  ```
  host1#show frame-relay interface delta
  Frame relay interface 2/0:1/1, status is lowerLayerDown
  Number of interface down transitions is 0
  Time since last status change 21:06:34
  Number of configured circuits: 0
  Time since last baseline 00:00:05
    In bytes: 0               Out bytes: 0
    In frames: 0              Out frames: 0
    In errors: 0              Out errors: 0
    In discards: 0            Out discards: 0
    In unknown protos: 0
  ```
- There is no **no** version.
- See *baseline frame-relay interface*.

show frame-relay interface

- Use to display statistics for the Frame Relay layer in a multilink Frame Relay, POS, serial, or GRE tunnel interface.
- Optionally, you can specify an interface using the interface type and specifier. For more information, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.
- Use the **brief** keyword to display the operational status of all configured interfaces.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - status—One of the following states:

- Up—Traffic can flow on the interface
- Offline—Traffic cannot flow because hardware is unavailable
- Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
- LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
- AdministrativelyDown—Traffic cannot flow because of manual administrative intervention

- Description—Text description or alias if configured for the interface
- In bytes—Number of inbound bytes received on the interface
- Out bytes—Number of outbound bytes transmitted on the interface
- In frames—Number of inbound frames received on the interface
- Out frames—Number of outbound frames transmitted on the interface
- In errors—Number of inbound errors received on the interface
- Out errors—Number of outbound errors transmitted on the interface
- In discards—Number of inbound packets discarded
- Out discards—Number of outbound packets discarded
- In unknown protos—Number of packets received on the interface with unknown protocols
- Time since last status change—Time since the last status change on the interface

- Example

```
host1#show frame-relay interface
Frame relay interface 3/2:1/1, status is up
Description: boston01
Time since last status change 01:21:10
  In bytes: 19712           Out bytes: 60918
  In frames: 1232           Out frames: 1232
  In errors: 0              Out errors: 0
  In discards: 0            Out discards: 0
  In unknown protos: 0
Frame relay interface 3/2:2/1, status is up
Description: newyork02
Time since last status change 03:06:18
  In bytes: 19728           Out bytes: 60702
  In frames: 1233           Out frames: 1233
  In errors: 0              Out errors: 0
  In discards: 0            Out discards: 0
  In unknown protos: 0
Frame relay interface 3/2:3/1, status is up
Description: chicago03
Time since last status change 01:20:38
  In bytes: 19696           Out bytes: 60744
```



```
In frames: 1231           Out frames: 1231
In errors: 0              Out errors: 0
```

- See *show frame-relay interface*.

show frame-relay lmi

- Use to display configuration and state information and statistics about the LMI for a multilink Frame Relay, POS, serial, or GRE tunnel interface.
- Optionally, you can specify an interface using the interface type and specifier. For more information, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - This command displays both DTE and DCE fields for NNI.
  - For the DTE:
    - Enquiries sent—Total number of LMI status enquiries sent by the DTE on this interface
    - Full enquiries sent—Total number of LMI full-status enquiries sent by the DTE on this interface
    - Enquiry responses received—Total number of LMI full- and regular-status responses received by the DTE on this interface
    - Full enquiry responses received—Total number of LMI full-status responses received by the DTE on this interface
    - Async updates received—Total number of LMI asynchronous updates received by the DTE on this interface
    - Unknown messages received—Total number of unknown LMI messages received on this interface
    - Loss of sequencing detected—Total number of times a loss of sequencing in received LMI messages was detected by the DTE on this interface
    - No response timeouts—Total number of times a timeout occurred without receiving a response to an LMI request by the DTE on this interface
    - Last sequence number sent—Last sequence number sent on this interface
    - Last sequence number received—Last sequence number received on this interface
  - For the DCE:
    - Enquiries received—Total number of LMI status enquiries received by the DCE on this interface
    - Enquiry responses sent—Total number of LMI status responses sent by the DCE on this interface
    - Full enquiry responses sent—Total number of LMI full-status responses sent by the DCE on this interface

- Async updates sent—Total number of LMI asynchronous updates sent by the DCE on this interface
- Unknown messages received—Total number of unknown LMI messages received on this interface
- Loss of sequencing detected—Total number of times a loss of sequencing in received LMI messages was detected by the DCE on this interface
- No response timeouts—Total number of times a timeout occurred without receiving a status inquiry from the DTE on this interface
- Last sequence number sent—Last sequence number sent on this interface
- Last sequence number received—Last sequence number received on this interface

- Example

```
host1#show frame-relay lmi
LMI information for frame relay NNI interface 3/2:1/1
DTE Parameter N391 is 6, N392 is 3, N393 is 4, T391 is 10
DCE Parameter N392 is 2, N393 is 2, T392 is 15
Configured LMI type is ANSI, status is up
Time since last status change 01:21:14
  Enquiries received: 1232
  Full enquiries received: 207
  Enquiry responses sent: 1232
  Full enquiry responses sent: 207
  Async updates sent: 0
  Unknown messages received: 0
  Loss of sequencing detected: 2
  No response timeouts: 0
  Last sequence number sent: 0
  Last sequence number received: 0
  Unknown messages received: 0
  Loss of sequencing detected: 2
LMI information for frame relay DCE interface 3/2:2/1
Parameter N392 is 2, N393 is 2, T392 is 15
Configured LMI type is ANSI, status is up
Time since last status change 03:06:22
  Enquiries received: 1233
  Full enquiries received: 207
  Enquiry responses sent: 1233
  Full enquiry responses sent: 207
  Async updates sent: 0
  Last sequence number sent: 0
  Last sequence number received: 0
```

- See *show frame-relay lmi*.

show frame-relay map

- Use to display the current Frame Relay map entries and information about Frame Relay connections.
- Field descriptions
  - Frame relay sub-interface—Interface number and one of the following states:

- Up—Traffic can flow on the interface
- Offline—Traffic cannot flow because hardware is unavailable
- Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
- LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
- AdministrativelyDown—Traffic cannot flow because of manual administrative intervention

- DLCI—Provides decimal value, hexadecimal value, and its value as it appears on the wire

- Example

```
host1#show frame-relay map
Frame relay sub-interface 3/2:1/1.1 (up): DLCI 101(0x65,0x58)
Frame relay sub-interface 3/2:1/1.2 (up): DLCI 102(0x66,0x78)
Frame relay sub-interface 3/2:1/1.3 (up): DLCI 103(0x67,0x78)
Frame relay sub-interface 3/2:1/1.4 (up): DLCI 104(0x68,0x98)
Frame relay sub-interface 3/2:1/1.5 (up): DLCI 105(0x69,0x98)
Frame relay sub-interface 3/2:1/1.6 (up): DLCI 106(0x6a,0xb8)
Frame relay sub-interface 3/2:1/1.7 (up): DLCI 107(0x6b,0xb8)
Frame relay sub-interface 3/2:1/1.8 (up): DLCI 108(0x6c,0xd8)
Frame relay sub-interface 3/2:1/1.9 (up): DLCI 109(0x6d,0xd8)
Frame relay sub-interface 3/2:1/1.10 (up): DLCI 110(0x6e,0xf8)
Frame relay sub-interface 3/2:1/1.11 (up): DLCI 111(0x6f,0xf8)
Frame relay sub-interface 3/2:1/1.12 (up): DLCI 112(0x70,0x1c)
Frame relay sub-interface 3/2:1/1.17 (up): DLCI 117(0x75,0x5c)
Frame relay sub-interface 3/2:1/1.18 (up): DLCI 118(0x76,0x7c)
Frame relay sub-interface 3/2:1/1.19 (up): DLCI 119(0x77,0x7c)
Frame relay sub-interface 3/2:1/1.20 (up): DLCI 120(0x78,0x9c)
Frame relay sub-interface 3/2:1/1.21 (up): DLCI 121(0x79,0x9c)
Frame relay sub-interface 3/2:1/1.22 (up): DLCI 122(0x7a,0xbc)
Frame relay sub-interface 3/2:1/1.23 (up): DLCI 123(0x7b,0xbc)
Frame relay sub-interface 3/2:1/1.24 (up): DLCI 124(0x7c,0xdc)
```

- See *show frame-relay map*.

show frame-relay pvc

- Use to display statistics about PVCs for Frame Relay layer on a multilink Frame Relay, POS, serial, or GRE tunnel interface or a specific PVC.
- Optionally, you can specify an interface using the interface type and specifier. For more information, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.
- Specify a virtual circuit using the DLCI number.
- Use the **brief** keyword to display abbreviated PVC information.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions

- DLCI—DLCI number
- interface—Identifies an interface in *slot/port:channel/subchannel* format or a subinterface in *slot/port:channel/subchannel.subinterface* format
- PVC status—Status of the circuit; valid states are active and inactive.
- Number of circuit status inactive transitions—number of times a circuit came down because of error conditions
- Time since creation—Time since the PVC was created
- last status change—Time since the PVC status last changed
- In pkts—Number of incoming packets received on the circuit
- Out pkts—Number of outgoing packets transmitted on the circuit
- In bytes—Number of input bytes received on the circuit
- Out bytes—Number of output bytes received on the circuit
- In FECN pkts—Number of packets received with the forward explicit congestion notification (FECN) bit set. The FECN bit is set by a network to notify the user that congestions may be experienced by data traffic in the direction of the frame carrying the FECN bit. The FECN bit is set by the network (not by the transmitting user), and there is no obligation for end systems to take any action regarding the FECN bit.
- Out FECN pkts—Number of packets transmitted with the FECN bit set
- In BECN pkts—Number of packets received with the backward explicit congestion notification (BECN) bit set. The BECN bit is set by a network to notify the user that congestions may be experienced by data traffic in the opposite direction of the frame carrying the BECN bit. The BECN bit is set by the network, and there is no obligation for end systems to take any action regarding the BECN bit.
- Out BECN pkts—Number of packets transmitted with the BECN bit set
- In DE pkts—Number of packets received with the discard eligibility (DE) bit set. When the DE bit is set, it indicates that the frame is discarded in preference to other frames without the DE bit set. The DE bit may be set by the network or the user. Once it is set, it cannot be reset by the user.
- Out DE pkts—Number of packets transmitted with the DE bit set
- Dropped packets—Number of dropped packets

- Example

```
host1#show frame-relay pvc
PVC information for frame relay NNI interface 3/2:1/1

DLCI 101 in sub-interface 3/2:1/1.1, status is active
Number of circuit status inactive transitions is 0
Time since creation 03:27:29, last status change 01:21:29
  In pkts: 0                Out pkts: 0
  In bytes: 0               Out bytes: 0
  In FECN pkts: 0           Out FECN pkts: 0
  In BECN pkts: 0           Out BECN pkts: 0
  In DE pkts: 0             Out DE pkts: 0
```



```
  Dropped pkts: 0
DLCI 102 in sub-interface 3/2:1/1.2, status is active
Number of circuit status inactive transitions is 0
Time since creation 03:27:28, last status change 01:21:29
  In pkts: 0                Out pkts: 0
  In bytes: 0               Out bytes: 0
  In FECN pkts: 0           Out FECN pkts: 0
  In BECN pkts: 0           Out BECN pkts: 0
  In DE pkts: 0             Out DE pkts: 0
  Dropped pkts: 0
DLCI 103 in sub-interface 3/2:1/1.3, status is active
Number of circuit status inactive transitions is 0
Time since creation 03:27:28, last status change 01:21:29
  In pkts: 0                Out pkts: 0
  In bytes: 0               Out bytes: 0
  In FECN pkts: 0           Out FECN pkts: 0
  In BECN pkts: 0           Out BECN pkts: 0
  In DE pkts: 0             Out DE pkts: 0
  Dropped pkts: 0
```

- See *show frame-relay pvc*.

show frame-relay subinterface

- Use to display the state of the Frame Relay subinterface.
- The subinterface can be in one of the following states:
  - Up—Traffic can flow on the interface
  - Offline—Traffic cannot flow because hardware is unavailable
  - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
  - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
  - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- The **brief** keyword displays only the operational status of all configured subinterfaces.
- Field descriptions
  - sub-interface—Identifies the subinterface in *slot/port:channel/subchannel.subinterface* format
  - status—Status of the subinterface
  - Description—Text description or alias if configured for the subinterface
  - Time since last status change—Time since the last status change on the subinterface
  - In bytes—Number of inbound bytes received on the subinterface
  - Out bytes—Number of outbound bytes transmitted on the subinterface
  - In frames—Number of inbound frames received on the interface
  - Out frames—Number of outbound frames transmitted on the interface

- In errors—Number of inbound errors received on the subinterface
- Out errors—Number of outbound errors transmitted on the subinterface
- In discards—Number of inbound packets discarded
- Out discards—Number of outbound packets discarded
- In unknown protos—Number of packets received on the subinterface with unknown protocols

- Example

```
host1#show frame-relay subinterface
Frame relay sub-interface 3/2:1/1.1, status is up
Description: toronto011
Time since last status change 01:21:26
  In bytes: 0                Out bytes: 0
  In frames: 0               Out frames: 0
  In errors: 0               Out errors: 0
  In discards: 0             Out discards: 0
  In unknown protos: 0

Frame relay sub-interface 3/2:1/1.2, status is up
Description: ottawa012
Time since last status change 01:21:26
  In bytes: 0                Out bytes: 0
  In frames: 0               Out frames: 0
  In errors: 0               Out errors: 0
  In discards: 0             Out discards: 0
  In unknown protos: 0
Frame relay sub-interface 3/2:1/1.3, status is up
Description: montreal013
Time since last status change 01:21:26
  In bytes: 0                Out bytes: 0
  In frames: 0               Out frames: 0
  In errors: 0               Out errors: 0
  In discards: 0             Out discards: 0
  In unknown protos: 0
```

- See *show frame-relay subinterface*.

show frame-relay summary

- Use to scan all defined Frame Relay interfaces and circuits; reports aggregate status as one of the following:
  - Up—Traffic can flow on the interface
  - Down—Traffic cannot flow because of a problem in the network
  - Unavailable—Traffic cannot flow because hardware is unavailable

- Example

```
host1#show frame-relay summary
28 interface(s) defined, 28 up, 0 down
840 sub-interface(s) defined, 840 up, 0 down
840 circuit(s) defined, 840 up, 0 down
```

- See *show frame-relay summary*.

CHAPTER 3

Configuring Multilink Frame Relay

This chapter describes how to configure Multilink Frame Relay (MLFR) interfaces on E Series routers.

This chapter contains the following sections:

- Overview on page 129
- Platform Considerations on page 131
- References on page 132
- Supported MLFR Features on page 132
- Unsupported MLFR Features on page 133
- Before You Configure MLFR on page 134
- Configuration Tasks on page 134
- Monitoring MLFR on page 136

Overview

MLFR aggregates multiple physical links into a single logical bundle. More specifically, MLFR bundles multiple link-layer channels into a single network layer channel.

The routers joined by the multilink each assign the same unique name to the bundle. A bundle can consist of multiple physical links of the same type—such as multiple asynchronous lines—or can consist of physical links of different types—such as leased synchronous lines and dial-up asynchronous lines.

The router treats MLFR like nonmultilink Frame Relay. Packets received with an MLFR header are subject to sequencing. Packets received without the MLFR header cannot be sequenced and can be delivered only on a first-come, first-served basis.

T1/E1 Connections

Some users need more bandwidth than a T1 or an E1 channel can provide, but cannot afford the expense or do not need the bandwidth of T3 or E3. Equal-cost multipath (ECMP) is one way to achieve a bandwidth greater than DS1 service without going to the expense and infrastructure required for DS3 service. MLFR is commonly used as an alternative to ECMP to deliver NxT1 service. Cost-analysis of NxT1 versus DS3 service typically imposes a practical limit of 8xT1 service; that is, aggregation of no more than eight T1 or E1 connections into an MLFR bundle.

This implementation of MLFR logically aggregates up to eight T1 or E1 connections into a single virtual connection, known as a bundle, to a given customer site. The connections can terminate at a CPE (Figure 5 on page 130) or a Multilink Frame Relay bridge (Figure 6 on page 130).

Figure 5: MLFR Aggregation of T1 Lines into a Single Bundle

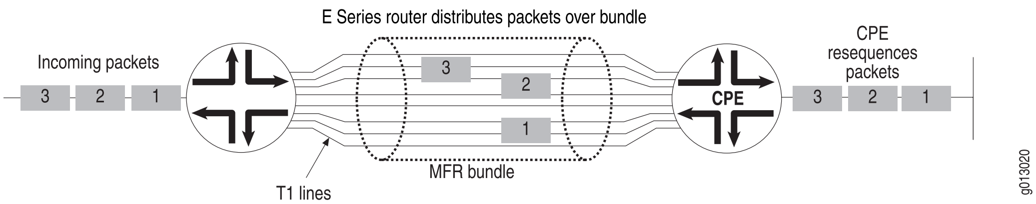
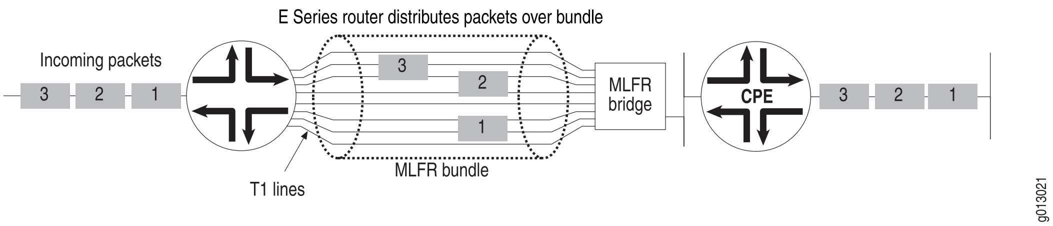


Figure 6: Terminating the Bundle at an MLFR Bridge



MLFR Link Integrity Protocol

Member links in an MLFR bundle support the MLFR Link Integrity Protocol (LIP). LIP offers several types of messages, which allow member links to join and leave a bundle. See Table 9 on page 130.

Table 9: LIP Messages and Functions

LIP Message	Function
Add-Link	Member link sends this message to request to join a bundle.
Add-Link-Ack	Member link sends this message when it receives an Add-Link message.
Add-Link-Rej	Member link sends this message to reject a request to join a bundle.
Hello	Member link sends this message to check the status of another member.
Hello-Ack	Member link sends this message when it receives a Hello message.
Remove-Link	Member link sends this message to request to leave a bundle.

Table 9: LIP Messages and Functions *(continued)*

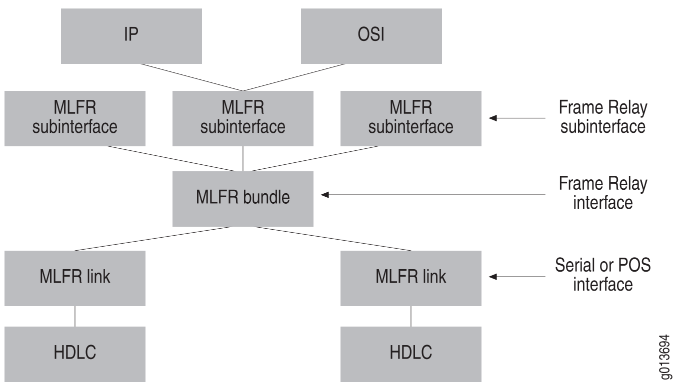
LIP Message	Function
Remove-Link-Ack	Member link sends this message when it receives a Remove-Link message.

The DTE creates a link management interface (LMI) with the network by encapsulating the Frame Relay frame within an MLFR frame. You assign one or more data link control identifiers (DLCIs) to a bundle.

Interface Stacking

Because MLFR aggregates multiple link-layer channels onto a single network layer IP interface, protocol layering within the router is different than it is for nonmultilink Frame Relay. See Figure 7 on page 131.

Figure 7: Structure of MLFR



The MLFR Link Integrity Protocol runs on each link in a bundle. However, from the major Frame Relay interface (the bundle) upward, the interface stacking is the same as for nonmultilink Frame Relay. For example, LMI runs only on the bundle. The bundle sends and receives all MLFR packets.

Platform Considerations

You can configure MLFR interfaces on the following E Series Broadband Services Routers:

- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

NOTE: The E120 and E320 Broadband Services Routers do not support configuration of MLFR interfaces.

Module Requirements

For information about the modules that support MLFR interfaces on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support MLFR.

Interface Specifiers

The interface specifier format that you use depends on the type of physical interface on which you want to configure MLFR.

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

References

For more information about the MLFR protocol, consult the following resources:

- Multilink Frame Relay UNI/NNI Implementation Agreement, FRF.16 (April 2000)
- Frame Relay Forum—Frame Relay Fragmentation Implementation Agreement, FRF.12 (December 1997)
- ANSI T1.617 Annex D
- ITU-T Q.933 Annex A

Supported MLFR Features

E Series routers support the following MLFR features on the cOCx/STMx and CT3 line modules:

- Logical aggregation of up to eight T1 or E1 links in a bundle
- Monotonically increasing sequence numbers for each circuit

  All packets distributed across the member links have monotonically increasing sequence numbers for each circuit. This feature enables the remote router on the customer premises to perform resequencing (if it is configured to do so).
- Static configuration of the links participating in a multilink bundle
- Round-robin packet distribution
  - On CT3 line modules, packet distribution across the member links in a bundle is handled only in a round-robin fashion. The round-robin approach is used even when the member links have different line rates.
  - On cOCx/STMx and COCX-F3 line modules, the router keeps track of the link with the least traffic. If this link cannot forward a packet, the router attempts to forward

the traffic on a different link. If this attempt also fails, the router uses a round-robin approach.

You can configure bundles as follows:

- On a cOCx/STMx line module and its corresponding I/O modules, you can configure:
  - Member links from different OC3/STM1 ports in the same bundle
  - The 336 available T1 channels combined in any manner that does not exceed 8 links per bundle (for example, 336 single-link T1 bundles, 42 eight-link bundles, or 41 eight-link bundles and 8 single-link bundles)
  - The 252 available E1 channels combined in any manner that does not exceed 8 links per bundle (for example, 252 single-link E1 bundles, 34 eight-link bundles, or 33 eight-link bundles and 8 single-link bundles)
- On a COCX-F3 line module and its corresponding I/O modules, you can configure:
  - Up to 8 member links from different ports in the same bundle
  - Up to 12 bundles
- On a CT3 or CT3/T3-F0 line module and its corresponding I/O module, you can configure:
  - Only member links from the same T3 interfaces into the same bundle. You cannot configure member links from different T3 ports in the same bundle.
  - The 28 available T1 channels on each port combined in any manner that does not exceed 8 links per bundle (for example, 28 single-link T1 bundles or 3 eight-link bundles and 4 single-link bundles per port)
- You can configure IPv6 addresses on Frame Relay subinterfaces in an MLFR bundle that contains serial interfaces as member links. MLFR bundles with IPv6 addresses enable effective usage of bandwidth and reduced administrative costs.

Unsupported MLFR Features

E Series routers do not support the following MLFR features:

- Fragmentation

  The router does not support MLFR fragmentation or reassembly. When using MLFR on the router, configure all peer devices so that they do not fragment MLFR frames. The router drops all fragmented frames that it receives.

- Resequencing of out-of-order packets in the absence of fragmentation

  Given the location in the network where the router resides, the NxT1 links to a customer site represent one of many places across the IP network where packets might be received out of order. For example, if the router has multiple uplinks to a core router, packets might be received out of order across these links. Packet resequencing is therefore left as an exercise for the end station rather than the aggregation router.

Before You Configure MLFR

Before you begin configuring MLFR, you must configure the physical layer interfaces that will be aggregated by MLFR.

The procedures described in this chapter assume that a physical layer interface, such as a T1 or T3 interface, has been configured. For details about configuring physical layer interfaces, see the *JunosE Physical Layer Configuration Guide*.

Configuration Tasks

MLFR configuration consists of three major tasks, each with several steps:

1. Create the member links to be aggregated into a multilink bundle.
   a. Specify the interface on which you want to configure MLFR.

      host1(config)#**interface serial 2/0:1**

   b. Specify MLFR as the encapsulation method on the interface.

      host1(config-if)#**encapsulation mlframe-relay ietf**

2. Add member links to a multilink bundle.
   a. Define the MLFR bundle.

      host1(config)#**interface mlframe-relay boston**

   b. Add each member link.

      host1(config-if)#**member-interface serial 2/0:1**

   c. (Optional) Add a description to the major interface.

      host1(config-if)#**frame-relay description bostonBundleDescription**

   d. (Optional) Configure Frame Relay parameters.

      host1(config-if)#**frame-relay intf-type dce**
      host1(config-if)#**frame-relay lmi-type cisco**

3. Configure the Frame Relay subinterface.
   a. Define the subinterface for the MLFR bundle.

      host1(config)#**interface mlframe-relay boston.1**

   b. Assign a DLCI for the subinterface.

      host1(config-subif)#**frame-relay interface-dlci 16 ietf**

   c. (Optional) Add a description to the subinterface.

      host1(config-subif)#**frame-relay description bostonBundleSubOneDescription**

   d. Assign an IP address to the subinterface.

      host1(config-subif)#**ip address 10.10.100.1 255.255.255.0**

   e. Assign an IPv6 address to the subinterface.

host1(config-subif)#**ipv6 address 1::1/64**

Configuration Example

The following commands configure three T1 lines and aggregate them into a multilink bundle named boston.

```
host1(config)#interface serial 2/0:1
host1(config-if)#encapsulation mlframe-relay ietf
host1(config-if)#exit
host1(config)#interface serial 2/0:2
host1(config-if)#encapsulation mlframe-relay ietf
host1(config-if)#exit
host1(config)#interface serial 2/0:3
host1(config-if)#encapsulation mlframe-relay ietf
host1(config-if)#exit
host1(config)#interface mlframe-relay boston
host1(config-if)#member-interface serial 2/0:1
host1(config-if)#frame-relay description bostonBundleDescription
host1(config-if)#frame-relay intf-type dce
host1(config-if)#frame-relay lmi-type cisco
host1(config-if)#member-interface serial 2/0:2
host1(config-if)#member-interface serial 2/0:3
host1(config-if)#exit
host1(config)#interface mlframe-relay boston.1
host1(config-subif)frame-relay description bostonBundleSubOneDescription
host1(config-subif)#frame-relay interface-dlci 16 ietf
host1(config-subif)#ip address 10.10.100.1 255.255.255.0
```

Configuring Frame Relay Versus MLFR

All the configuration commands that apply to Frame Relay also apply to MLFR. The following listing describes commands specific to configuring MLFR; for other Frame Relay commands, see "Configuring Frame Relay" on page 103.

encapsulation mlframe-relay ietf

- Use to configure MLFR as the encapsulation method on an individual interface.
- Use this command only within the context of an individual interface. Issuing this command creates an MLFR link, also referred to as an MLFR bundle member.
- Example

  ```
  host1(config)#interface serial 2/0:1
  host1(config-if)#encapsulation mlframe-relay ietf
  ```
- Use the **no** version to disable MLFR on an interface.
- See *encapsulation mlframe-relay ietf*.

interface mlframe-relay

- Use to create a Frame Relay major interface, also known as the MLFR bundle.
- Example

  ```
  host1(config-if)#interface mlframe-relay group2
  ```

- Use the **no** version to delete the MLFR bundle.
- See *interface mlframe-relay*.

member-interface

- Use to add an MLFR interface—also known as an MLFR bundle member—to an MLFR bundle.
- Example

  host1(config-if)#**member-interface serial 2/0:1**

- Use the **no** version to remove the specified interface from the MLFR bundle.
- See *member-interface*.

Monitoring MLFR

Use the commands in this section to display information about MLFR interfaces.

You can set a statistics baseline for an MLFR bundle or subinterface using the **baseline frame-relay interface mlframe-relay** command. Similarly, you can set a statistics baseline for an MLFR link with the **baseline frame-relay multilink interface** command. Use the **delta** keyword with the **show** commands described below to display statistics with the baseline values subtracted.

After you configure multilink Frame Relay, you can use the **show frame-relay** commands to view information about the multilink. For information about these commands, see "Configuring Frame Relay" on page 103.

You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. Refer to the section show Commands in *JunosE System Basics Configuration Guide*, for details.

baseline frame-relay interface

- Use to set a statistics baseline for the Frame Relay layer on MLFR bundles, Frame Relay interfaces, subinterfaces, and circuits.
- Specify the keyword **mlframe-relay** and the name of the MLFR bundle to set a baseline for the Frame Relay statistics on an MLFR bundle.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- You cannot set a baseline for groups of interfaces, subinterfaces, or circuits. You must set baselines one at a time.
- When baselining is requested, the time since the last baseline was set is displayed in *hours:minutes:seconds* or *days/hours* format. If a baseline has not been set, the message " No baseline has been set" is displayed instead.
- The regular interface statistics and LMI statistics for interfaces are subject to the same baseline timestamp. You cannot set separate baselines for these statistics.

- Use the optional **delta** keyword with Frame Relay **show** commands to specify that baselined statistics are to be shown.
- Example

  host1#**baseline frame-relay interface mlframe-relay boston**
- There is no **no** version.
- See *baseline frame-relay interface*.

baseline frame-relay multilinkinterface

- Use to set a statistics baseline for MLFR links.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- When baselining is requested, the time since the last baseline was set is displayed in *hours:minutes:seconds* or *days/hours* format. If a baseline has not been set, the message " No baseline has been set" is displayed instead.
- The regular interface statistics and LIP statistics for interfaces are subject to the same baseline timestamp. You cannot set separate baselines for these statistics.
- Use the optional **delta** keyword with Frame Relay **show** commands to specify that baselined statistics are to be shown.
- Example

  host1#**baseline frame-relay multilinkinterface serial 3/2**
- There is no **no** version.
- See *baseline frame-relay multilinkinterface*.

show frame-relay interface

- Use to display the information about the Frame Relay layer of the interface.
- Use the **brief** keyword to display the operational status of all configured interfaces.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - Frame relay interface mlframe-relay—Name of the MLFR bundle
  - Status of the major Frame Relay interface—One of the following states:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer

    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
  - Number of interface down transitions—Number of interfaces that have changed to a down state
  - Time since last status change—Time since the interface last changed its state
  - In bytes—Number of inbound bytes received on the interface
  - In frames—Number of inbound frames received on the interface
  - In errors—Number of inbound errors received on the interface
  - In discards—Number of inbound packets discarded
  - In unknown protos—Number of packets received on the interface with unknown protocols
  - Out bytes—Number of outbound bytes transmitted on the interface
  - Out frames—Number of outbound frames transmitted on the interface
  - Out errors—Number of outbound errors transmitted on the interface
  - Out discards—Number of outbound packets discarded

- Example 1

```
host1#show frame-relay interface brief
Frame relay interface mlframe-relayTEST, status is up
```

- Example 2

```
host1#show frame-relay interface mlframe-relay TEST
Frame relay interface mlframe-relayTEST, status is up
Number of interface down transitions is 0
Time since last status change 00:01:47
Number of configured circuits: 2
  In bytes: 452             Out bytes: 198
  In frames: 19             Out frames: 11
  In errors: 0              Out errors: 0
  In discards: 8            Out discards: 0
  In unknown protos: 0
```

- Example 3

```
host1#show frame-relay interface mlframe-relay members
Frame relay interface mlframe-relay TEST is up
   Frame relay multilink member-interface 4/0:1 is up
   Frame relay multilink member-interface 4/1:1 is up
```

- See *show frame-relay interface*.

show frame-relay lip

- Use to display the state of MLFR Link Integrity Protocol (LIP) on an MLFR link.
- Use the **brief** keyword to display the operational status of all configured interfaces.

- Use the **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - Frame relay interface—Specifier for the Frame Relay interface
  - Status of the major Frame Relay interface—One of the following states:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
  - Number of interface down transitions—Number of interfaces that have changed to a down state
  - Time since last status change—Time since the interface last changed its state
  - Add Links sent—Number of Add Link messages sent from this interface
  - Add Links received—Number of Add Link messages received on this interface
  - Add Link Acknowledgments sent—Number of Add Link acknowledgments sent from this interface
  - Add Link Acknowledgments received—Number of Add Link acknowledgments received on this interface
  - Add Link Rejects sent—Number of Add Link Reject messages sent from this interface
  - Add Link Rejects received—Number of Add Link Reject messages received on this interface
  - Hellos sent—Number of Hello messages sent from this interface
  - Hellos received—Number of Hello messages received on this interface
  - Hello Acknowledgments sent—Number of Hello messages sent from this interface
  - Hello Acknowledgments received—Number of Hello messages received on this interface
  - Remove Links sent—Number of Remove Link messages sent from this interface
  - Remove Links received—Number of Remove Link messages received on this interface
  - Remove Link Acknowledgments sent—Number of Remove Link acknowledgments sent from this interface
  - Remove Link Acknowledgments received—Number of Remove Link acknowledgments received on this interface
- Example 1

```
host1#show frame-relay lip brief
LIP information for frame relay interface 4/0:1, status is up
Number of interface down transitions is 0
Time since last status change 00:03:16

LIP information for frame relay interface 4/1:1, status is up
Number of interface down transitions is 0
Time since last status change 00:03:20
```

- Example 2

```
host1#show frame-relay lip interface serial 4/0:1
LIP information for frame relay interface 4/0:1, status is up
Number of interface down transitions is 0
Time since last status change 00:05:19
  Add Links sent: 1
  Add Links received: 1
  Add Link Acknowledgements sent: 1
  Add Link Acknowledgements received: 1
  Add Link Rejects sent: 0
  Add Link Rejects received: 0
  Hellos sent: 32
  Hellos received: 31
  Hello Acknowledgements sent: 31
  Hello Acknowledgements received: 32
  Remove Links sent: 0
  Remove Links received: 0
  Remove Link Acknowledgements sent: 0
  Remove Link Acknowledgements received: 0
```

- See *show frame-relay lip*.

show frame-relay lmi

- Use to display configuration and state information and statistics about the LMI.
- You can specify an interface type and location.
- Use the **brief** keyword to display abbreviated PVC information.
- Use the **delta** keyword to specify that baselined statistics are to be shown.
- DTE field descriptions
  - Frame relay DTE interface mlframe-relay—Name of the MLFR bundle
  - N391—Value of the N391 full-status polling counter
  - N392—Value of the N392 error threshold counter
  - N393—Value of the N393 monitored events counter
  - T391—Value of the T391 link integrity polling timer interval
  - Configured LMI type—One of the following options:
    - ANSI—ANSI T1.617 Annex D
    - Q933A—ITU-T Q.933 Annex A
    - Cisco—Original *Group of Four* specification developed by DEC, Northern Telecom, Stratacom, and Cisco
    - None—Suppresses LMI

  - status is up—Availability of the MLFR bundle: up or down
  - Number of interface down transitions—Number of times the interface has become unavailable
  - Time since last status change—elapsed time since LMI information changed
    - Enquiries sent—Total number of LMI status inquiries sent by the DTE on this interface
    - Full enquiries sent—Total number of LMI full status inquiries sent by the DTE on this interface
    - Enquiry responses received—Total number of LMI full and regular status responses received by the DTE on this interface
    - Full enquiry responses received—Total number of LMI full status responses received by the DTE on this interface
    - Async updates received—Total number of asynchronous LMI updates received by the DTE on this interface
    - Unknown messages received—Total number of unknown LMI messages received on this interface
    - Loss of sequencing detected—Total number of times a loss of sequencing in received LMI messages was detected by the DTE on this interface
    - No response timeouts—Total number of times a timeout occurred without receiving a response to an LMI request by the DTE on this interface
    - Last sequence number sent—Last sequence number sent on this interface
    - Last sequence number received—Last sequence number received on this interface

- DCE field descriptions:
  - Frame relay DCE interface mlframe-relay—Name of the MLFR bundle
  - N391—Value of the N391 full-status polling counter
  - N392—Value of the N392 error threshold counter
  - T392—Value of the T392 polling verification timer
  - Configured LMI type: one of the following options:
    - ANSI—ANSI T1.617 Annex D
    - Q933A—ITU-T Q.933 Annex A
    - Cisco—Original *Group of Four* specification developed by DEC, Northern Telecom, Stratacom, and Cisco
    - None—Suppresses LMI
  - status is up—Availability of the MLFR bundle: up or down

- Number of interface down transitions—Number of times the interface has become unavailable
- Time since last status change—Elapsed time since LMI information changed
  - Enquiries received—Total number of LMI status inquiries received by the DCE on this interface
  - Enquiry responses sent—Total number of LMI status responses sent by the DCE on this interface
  - Full enquiry responses sent—Total number of LMI full status responses sent by the DCE on this interface
  - Async updates sent—Total number of LMI ASYNC updates sent by the DCE on this interface
  - Unknown messages received—Total number of unknown LMI messages received on this interface
  - Loss of sequencing detected—Total number of times a loss of sequencing in received LMI messages was detected by the DCE on this interface
  - No response timeouts—Total number of times a timeout occurred without receiving a status inquiry from the DTE on this interface
  - Last sequence number sent—Last sequence number sent on this interface
  - Last sequence number received—last sequence number received on this interface

- Example 1

```
host1#show frame-relay lmi brief
LMI information for frame relay DTE interface mlframe-relayTEST
DTE parameter N391 is 6, N392 is 3, N393 is 4, T391 is 10
Configured LMI type is ANSI, status is up
Number of interface down transitions is 0
Time since last status change 00:05:39
```

- Example 2

```
host1#show frame-relay lmi interface mlframe-relay TEST
LMI information for frame relay DTE interface mlframe-relayTEST
DTE parameter N391 is 6, N392 is 3, N393 is 4, T391 is 10
Configured LMI type is ANSI, status is up
Number of interface down transitions is 0
Time since last status change 00:06:20
  Enquiries sent: 39
  Full enquiries sent: 7
  Enquiry responses received: 39
  Full enquiry responses received: 7
  Async updates received: 0
  Unknown messages received: 0
  Loss of sequencing detected: 0
  No response timeouts: 0
  Last sequence number sent: 39
  Last sequence number received: 39
```

- See *show frame-relay lmi*.

show frame-relay map

- Use to display the current Frame Relay and MLFR map entries.
- Field descriptions
  - subinterface—Name and subinterface number of the MLFR bundle in the format *bundle-name.subinterface-number*
  - State of the subinterface—One of the following states:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
  - DLCI number—Decimal value, hexadecimal value, and value as it appears on the wire of the DLCI
- Example

```
host1#show frame-relay map
Frame relay sub-interface mlframe-relayTEST.1 (up): DLCI 16(0x10,0x4)
Frame relay sub-interface mlframe-relayTEST.2 (up): DLCI 17(0x11,0x14)
```

- See *show frame-relay map*

show frame-relay multilinkInterface

- Use to display the statistics about all MLFR interfaces or the specified MLFR interfaces.
- Field descriptions
  - Multilink Frame relay interface—Specifier for the Frame Relay interface
  - State of the MLFR interface—One of the following states:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
  - Number of multilink interface down transitions—Number of interfaces that have changed to a down state
  - Time since last status change—Time since the interface last changed its state

  - In bytes—Number of inbound bytes received on the interface
  - In frames—Number of inbound frames received on the interface
  - In errors—Number of inbound errors received on the interface
  - In discards—Number of inbound packets discarded
  - In unknown protos—Number of packets received on the interface with unknown protocols
  - Out bytes—Number of outbound bytes transmitted on the interface
  - Out frames—Number of outbound frames transmitted on the interface
  - Out errors—Number of outbound errors transmitted on the interface
  - Out discards—Number of outbound packets discarded

- Example

```
host1#show frame-relay multilinkInterface
Multilink Frame relay interface 6/2:2, status is down
Number of multilink interface down transitions is 0
Time since last status change 2 days, 23 hours
  In bytes: 0              Out bytes: 0
  In frames: 0             Out frames: 0
  In errors: 0             Out errors: 0
  In discards: 0           Out discards: 0
  In unknown protos: 0
```

- See *show frame-relay multilinkInterface*.

show frame-relay pvc

- Use to display statistics about PVCs for Frame Relay interfaces.
- Specify a DLCI number or an interface type and location.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- The **brief** keyword displays abbreviated PVC information.
- Field descriptions
  - DLCI—DLCI number
  - subinterface—Name and subinterface number of the MLFR bundle in the format *bundle-name.subinterface-number*
  - status—Status of the PVC
  - Number of circuit status inactive transitions—Number of times a circuit came down because of error conditions
  - Time since creation—Time since the PVC was created
  - last status change—Time since the PVC status last changed
  - In pkts—Number of incoming packets received on the circuit
  - Out pkts—Number of outgoing packets transmitted on the circuit
  - In bytes—Number of input bytes received on the circuit

- Out bytes—Number of output bytes received on the circuit
- In FECN pkts—Number of packets received with the forward explicit congestion notification (FECN) bit set. The FECN bit is set by a network to notify the user that data traffic may experience congestion in the direction of the frame carrying the FECN bit. The FECN bit is set by the network (not by the transmitting user), and there is no obligation for end systems to take any action regarding the FECN bit.
- Out FECN pkts—Number of packets transmitted with the FECN bit set
- In BECN pkts—Number of packets received with the backward explicit congestion notification (BECN) bit set. The BECN bit is set by a network to notify the user that data traffic may experience congestion in the opposite direction of the frame carrying the BECN bit. The BECN bit is set by the network, and there is no obligation for end systems to take any action regarding the BECN bit.
- Out BECN pkts—Number of packets transmitted with the BECN bit set
- In DE pkts—Number of packets received with the discard eligibility (DE) bit set. When the DE bit is set, it indicates that the frame is discarded in preference to other frames without the DE bit set. The DE bit may be set by the network or the user. Once it is set, it cannot be reset by the user.
- Out DE pkts—Number of packets transmitted with the DE bit set
- Dropped packets—Number of dropped packets

- Example 1

```
host1#show frame-relay pvc brief
PVC information for frame relay DTE interface mlframe-relayTEST

DLCI 16 in sub-interface mlframe-relayTEST.1, status is active
DLCI 17 in sub-interface mlframe-relayTEST.2, status is active
```

- Example 2

```
host1#show frame-relay pvc interface mlframe-relay TEST
PVC information for frame relay DTE interface mlframe-relayTEST

DLCI 16 in sub-interface mlframe-relayTEST.1, status is active
Number of circuit status inactive transitions is 0
Time since creation 00:07:20, last status change 00:07:11
  In pkts: 14              Out pkts: 0
  In bytes: 420            Out bytes: 0
  In FECN pkts: 0          Out FECN pkts: 0
  In BECN pkts: 0          Out BECN pkts: 0
  In DE pkts: 0            Out DE pkts: 0
  Dropped pkts: 14

DLCI 17 in sub-interface mlframe-relayTEST.2, status is active
Number of circuit status inactive transitions is 0
Time since creation 00:07:20, last status change 00:07:11
  In pkts: 14              Out pkts: 0
  In bytes: 420            Out bytes: 0
  In FECN pkts: 0          Out FECN pkts: 0
  In BECN pkts: 0          Out BECN pkts: 0
```



```
In DE pkts: 0            Out DE pkts: 0
Dropped pkts: 14
```

- See *show frame-relay pvc*.

show frame-relay subinterface

- Use to display the state of the subinterface.
- The subinterface can be in one of the following states:
  - Up—Traffic can flow on the interface
  - Offline—Traffic cannot flow because hardware is unavailable
  - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
  - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
  - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
- Use the **brief** keyword to display only the operational status of all configured subinterfaces.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - Frame relay sub-interface mlframe-relay—Name and subinterface number of the MLFR bundle in the format *bundle-name.subinterface-number*
  - status—State of the subinterface, as follows:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
  - Number of sub-interface down transitions—Number of times a subinterface came down because of error conditions
  - Time since last status change—Time since the last status change on the subinterface
  - In bytes—Number of inbound bytes received on the subinterface
  - Out bytes—Number of outbound bytes transmitted on the subinterface
  - In frames—Number of inbound frames received on the interface
  - Out frames—Number of outbound frames transmitted on the interface

  - In errors—Number of inbound errors received on the subinterface
  - Out errors—Number of outbound errors transmitted on the subinterface
  - In discards—Number of inbound packets discarded
  - Out discards—Number of outbound packets discarded
  - In unknown protos—Number of packets received on the subinterface with unknown protocols

- Example 1

```
host1#show frame-relay subinterface brief
Frame relay sub-interface mlframe-relayTEST.1, status is up
Frame relay sub-interface mlframe-relayTEST.2, status is up
```

- Example 2

```
host1#show frame-relay subinterface mlframe-relay TEST
Frame relay sub-interface mlframe-relayTEST.1, status is up
Number of sub-interface down transitions is 0
Time since last status change 00:07:49
  In bytes: 512             Out bytes: 0
  In frames: 16             Out frames: 0
  In errors: 0              Out errors: 0
  In discards: 16           Out discards: 0
  In unknown protos: 0

Frame relay sub-interface mlframe-relayTEST.2, status is up
Number of sub-interface down transitions is 0
Time since last status change 00:07:50
  In bytes: 512             Out bytes: 0
  In frames: 16             Out frames: 0
  In errors: 0              Out errors: 0
  In discards: 16           Out discards: 0
  In unknown protos: 0
```

- See *show frame-relay subinterface*.

show frame-relay summary

- Use to scan all defined Frame Relay interfaces and circuits and to report the status for each discovered interface and circuit as follows:
  - Up—Traffic can flow on the interface
  - Down—Traffic cannot flow because of a problem in the network
  - Unavailable—Traffic cannot flow because hardware is unavailable

- Example

```
host1#show frame-relay summary
2 multilink interface(s) defined, 2 up, 0 down
1 interface(s) defined, 1 up, 0 down
2 sub-interface(s) defined, 2 up, 0 down
2 circuit(s) defined, 2 up, 0 down
```

- See *show frame-relay summary*.

CHAPTER 4

Configuring Upper-Layer Protocols over Static Ethernet Interfaces

This chapter describes how to configure upper-layer protocols over static Ethernet interfaces on E Series routers.

This chapter contains the following sections:

- Upper-Layer Protocols over Static Ethernet Overview on page 149
- Upper-Layer Protocols over Static Ethernet Platform Considerations on page 150
- Upper-Layer Protocols over Static Ethernet References on page 151
- Configuring IP over a Static Ethernet Interface on page 151
- Configuring PPPoE over a Static Ethernet Interface on page 152
- Configuring IP and MPLS over a Static Ethernet Interface on page 153
- Configuring IP, MPLS, and PPPoE over Ethernet on page 153
- L2TP and Ethernet on page 154
- Multinetting and Ethernet on page 155
- Monitoring Upper-Level Protocols over Ethernet on page 155

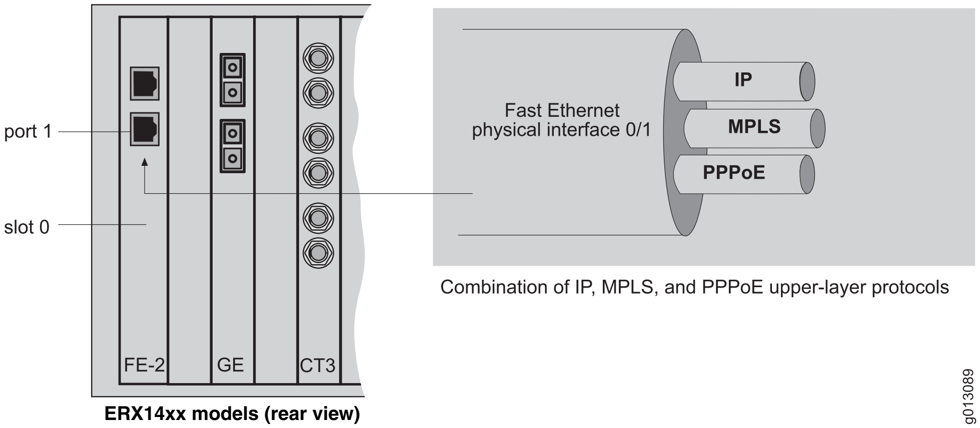
Upper-Layer Protocols over Static Ethernet Overview

You can configure one or more protocols over Ethernet with or without VLANs. This section focuses on non-VLAN configurations only. You can configure the following upper-layer protocols on Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces:

- IP
- Point-to-Point Protocol over Ethernet (PPPoE)
- Multiprotocol Label Switching (MPLS)

The Ethernet configuration examples in this section use combinations of these protocols. Figure 8 on page 150 illustrates how different protocols can be multiplexed over a single physical link without the use of VLANs.

Figure 8: Multiplexing Multiple Protocols over a Single Physical Link



The following sections describe how to create the following common non-VLAN configurations, which you can configure on Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces:

- IP over Ethernet
- PPPoE over Ethernet
- IP over Ethernet and MPLS over Ethernet
- IP over Ethernet, MPLS over Ethernet, and PPPoE over Ethernet

NOTE: You can also configure upper-layer protocols over dynamic interfaces. See "Configuring Upper-Layer Dynamic Interfaces" on page 519 and "Configuring Dynamic Interfaces Using Bulk Configuration" on page 619.

Upper-Layer Protocols over Static Ethernet Platform Considerations

You can configure upper-layer protocols over Ethernet on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules supported on E Series routers:

- See the *ERX Module Guide* for modules supported on ERX7xx models, ERX14xx models, and the ERX310 router.
- See the *E120 and E320 Module Guide* for modules supported on the E120 and E320 routers.

Interface Specifiers

The configuration task examples in this chapter use the format for ERX7xx models, ERX14xx models, and the ERX310 router to specify a VLAN or S-VLAN subinterface.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format. For example, the following command specifies a VLAN subinterface configured on port 0 of an I/O module in slot 4.

host1(config)#**interface fastEthernet 4/0.1**

For E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. For example, the following command specifies a VLAN subinterface configured on port 0 of the IOA installed in the upper adapter bay of slot 3.

host1(config)#**interface gigabitEthernet 3/0/0.1**

For more information about interface types and specifiers on E Series models, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Upper-Layer Protocols over Static Ethernet References

For more information about upper-layer protocol implementations over Ethernet, consult the following resources:

- RFC 894—A Standard for the Transmission of IP Datagrams over Ethernet Networks (April 1984)
- RFC 1042—A Standard for the Transmission of IP Datagrams over IEEE 802 Networks (February 1988)
- RFC 1112—Host Extensions for IP Multicasting (August 1989)
- RFC 2516—Method for Transmitting PPP over Ethernet (PPPoE) (February 1998)

Configuring IP over a Static Ethernet Interface

To configure IP over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

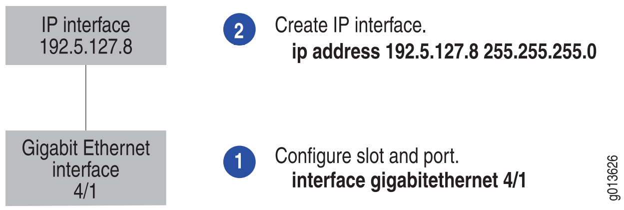
   host1(config)#**interface fastEthernet 4/1**

2. Create an IP interface.

   host1(config-if)#**ip address 192.5.127.8 255.255.255.0**

Figure 9 on page 152 illustrates this configuration.

Figure 9: Example of IP over Ethernet Stacking Configuration Procedure



Configuring PPPoE over a Static Ethernet Interface

To configure PPPoE over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/1**

2. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe**

3. Create a PPPoE subinterface.

   host1(config-if)#**pppoe subinterface** fastEthernet 4/1.1

4. Specify PPP as the encapsulation method on the interface.

   host1(config-if)#**encapsulation ppp**

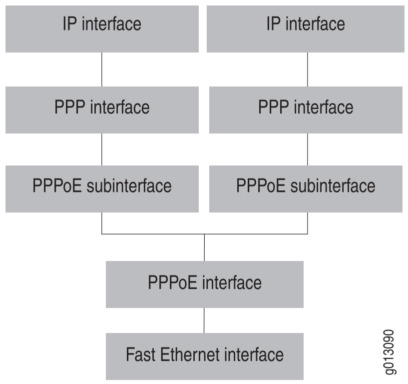
5. Assign an IP address and mask.

   host1(config-if)#**ip address 164.10.6.51 255.255.255.0**

6. (Optional) Configure additional PPPoE subinterfaces by completing Steps 3 through 5 using unique numbering.

Figure 10 on page 152 illustrates this configuration.

Figure 10: Example of PPPoE Stacking Configuration Procedure



Configuring IP and MPLS over a Static Ethernet Interface

To configure both IP and MPLS over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Create an IP interface.

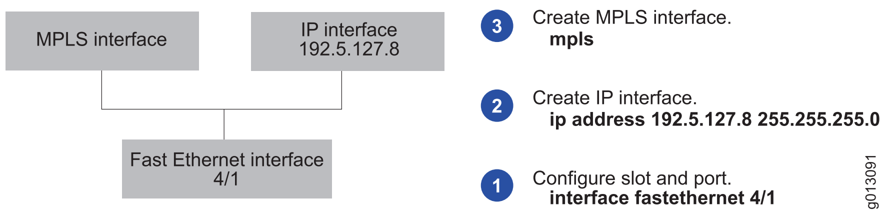
   host1(config-if)#**ip address 192.5.127.8 255.255.255.0**

3. Create an MPLS interface.

   host1(config-if)#**mpls**

Figure 11 on page 153 illustrates this configuration.

Figure 11: Example of IP and MPLS Stacking Configuration Procedure



Configuring IP, MPLS, and PPPoE over Ethernet

To configure IP, MPLS, and PPPoE over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Create an IP interface.

   host1(config-if)#**ip address 192.5.127.8 255.255.255.0**

3. Create an MPLS interface.

   host1(config-if)#**mpls**

4. Create a PPPoE interface by specifying PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe**

5. Create a PPPoE subinterface.

   host1(config-if)#**pppoe subinterface fastEthernet 4/1.1**

6. Specify PPP as the encapsulation method on the interface.

   host1(config-if)#**encapsulation ppp**

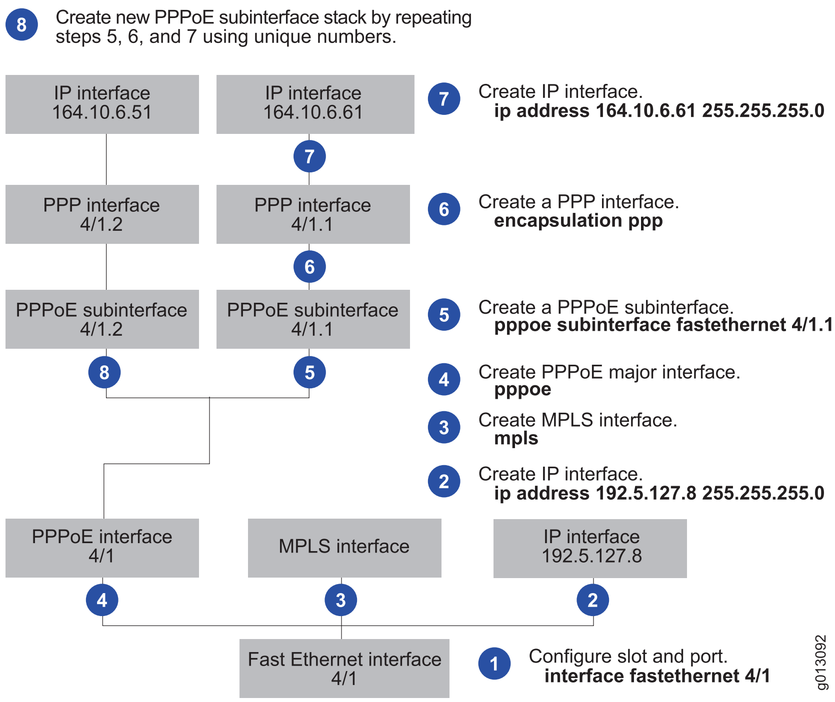
7. Assign an IP address and mask.

host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

8. (Optional) Configure additional PPPoE subinterfaces by completing Steps 5 through 7 using unique numbering.

Figure 12 on page 154 illustrates this configuration.

Figure 12: Example of IP, MPLS, and PPPoE Stacking Configuration Procedure



mpls

- Use to enable, disable, or delete MPLS on an interface. MPLS is disabled by default.
- Example

  host1(config)#**mpls**
- Use the **no** version to halt MPLS on the interface and delete the MPLS interface configuration.
- See *mpls*.

L2TP and Ethernet

Most Ethernet interfaces support L2TP. To use L2TP, you must first create a PPP interface. See *L2TP Overview* for information about configuring L2TP.

Multinetting and Ethernet

Ethernet interfaces, except for bridged Ethernet interfaces, support multinetting; that is, adding more than one IP address to an IP interface. If you want to add multiple IP addresses to a single IP interface, use the **ip address** command with the **secondary** keyword, which is described in *Configuring IP* in *JunosE IP, IPv6, and IGP Configuration Guide*.

Monitoring Upper-Level Protocols over Ethernet

This section explains how to use the **show** commands to display the physical characteristics and the configured settings for Ethernet interfaces.

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

You can use various **show** commands to monitor upper-layer protocols. For more information, see:

- "Configuring Point-to-Point Protocol over Ethernet" on page 387
- *Configuring IP* in *JunosE IP, IPv6, and IGP Configuration Guide*
- *Configuring MPLS* in *JunosE BGP and MPLS Configuration Guide*

show interfaces fastEthernet

- Use to display the status of Fast Ethernet interfaces, VLAN subinterfaces, or S-VLAN subinterfaces.
- You can specify the following keywords:
  - **delta**—Specifies that baselined statistics are to be shown
  - **brief**—Displays the operational status of all configured interfaces
- Field descriptions
  - FastEthernet *interfaceSpecifier*—Status of the hardware on this interface
    - up—Hardware is operational
    - down—Hardware is not operational
  - Administrative status—Operational state that you configured for this interface
    - up—Interface is enabled
    - down—Interface is disabled
  - Hardware—Type of MAC device on this interface

- Address—MAC address of the processor on this interface
- MAU—Type of medium attachment unit (MAU) on the physical port:
  - 10BASE-T (10 Mbps)
  - 100BASE-TX (100 Mbps)
  - 100BASE-FX-MM (100 Mbps) with the distance appearing after the type
  - 100BASE-LX-SM (100 Mbps)
  - SFP (Empty)—SFPs that are empty
  - SFP (Non-compliant Juniper Part)—SFPs that are installed in the FE-8 I/O module and do not have a Juniper Networks part number programmed
- MTU—Size of the MTU for this interface
  - Operational—Size of the largest packet processed
  - Administrative—Setting for MTU size that you specified
- Duplex Mode—Duplex option for this interface
  - Operational—Duplex option currently used
  - Administrative—Setting for duplex that you specified
- Speed—Line speed for this interface
  - Operational—Current rate at which packets are processed
  - Administrative—Setting for line speed
  - 5 minute input rate—Data rates based on traffic received in the last 5 minutes
  - 5 minute output rate—Data rates based on traffic sent in the last 5 minutes
- In—Analysis of inbound traffic on this interface
  - Bytes—Number of bytes received in error-free packets
  - Unicast—Number of unicast packets received
  - Multicast—Number of multicast packets received
  - Broadcast—Number of broadcast packets received
  - Errors—Total number of errors in all received packets; some packets might contain more than one error
  - Discards—Total number of discarded incoming packets
  - Mac Errors—Number of incoming packets discarded because of MAC sublayer failures
  - Alignment—Number of incomplete octets received
  - CRC—Number of packets discarded because the checksum the router computed from the data does not match the checksum generated by the originating devices

  - Too Longs—Number of packets discarded because the size exceeded the MTU
  - Symbol Errors—Number of symbols received that the router did not correctly decode
- Out—Analysis of outbound traffic on this interface
  - Bytes—Number of bytes sent
  - Unicast—Number of unicast packets sent
  - Multicast—Number of multicast packets sent
  - Broadcast—Number of broadcast packets sent
  - Errors—Total number of errors in all transmitted packets; some packets might contain more than one error
  - Discards—Total number of discarded outgoing packets
  - Mac Errors—Number of outgoing packets discarded because of MAC sublayer failures
  - Deferred—Number of packets that the router delayed sending because the line was busy. In half duplex mode, a high number of deferrals means the link is very busy with traffic from other stations. In full duplex mode, when the link is always available for transmission, this number is zero.
  - No Carrier—Number of packets sent when carrier sense was unavailable
- Collisions—Analysis of the collisions that occurred
  - Single—Number of packets sent after one collision
  - Multiple—Number of packets sent after multiple collisions
  - Late—Number of packets aborted during sending because of collisions after 64 bytes
  - Excessive—Number of packets not sent because of too many collisions
- ARP Statistics—Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface
  - ARP requests—Number of ARP requests
  - ARP responses—Number of ARP responses
  - Errors—Total number of errors in all ARP packets
  - Discards—Total number of discarded ARP packets
- queue—Hardware packet queue associated with the specified traffic class and interface
  - Queue length—Length of the queue, in bytes
  - Forwarded packets, bytes—Number of packets and bytes that were forwarded on this queue

- Dropped committed packets, bytes—Number of committed packets and bytes that were dropped
- Dropped conformed packets, bytes—Number of conformed packets and bytes that were dropped
- Dropped exceeded packets, bytes—Number of exceeded packets and bytes that were dropped

- Example—Displays the status of a Fast Ethernet interface

```
host1:vr2#show interfaces fastEthernet 2/0
FastEthernet2/0 is Up, Administrative status is Up
  Hardware is Intel 21440, address is 0090.1a10.0552  MAU is 10BASE-T
  MTU: Operational 1518, Administrative 1518
  Duplex Mode: Operational Full Duplex, Administrative Auto Negotiate
  Speed: Operational 100 Mbps, Administrative Auto Negotiate

  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec

  In: Bytes 39256, Unicast 612
   Multicast 0, Broadcast 0
   Errors 0, Discards 0, Mac Errors 0, Alignment 0
   CRC 0, Too Longs 0, Symbol Errors 0
  Out: Bytes 4579036, Unicast 610
   Multicast 0, Broadcast 70932
   Errors 0, Discards 0, Mac Errors 0, Deferred 0, No Carrier 3
   Collisions: Single 0, Multiple 0, Late 0, Excessive 0
 ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
Administrative qos-shaping-mode: none
Operational qos-shaping-mode: none

queue 0: traffic class control, bound to FastEthernet2/0
  Queue length 0 bytes
  Forwarded packets 1, bytes 46
  Dropped committed packets 0, bytes 0
  Dropped conformed packets 0, bytes 0
  Dropped exceeded packets 0, bytes 0
```

- See *show interfaces*.

show interfaces gigabitEthernet

show interfaces tenGigabitEthernet

- Use to display the status of Gigabit Ethernet interfaces, 10-Gigabit Ethernet interfaces, VLAN subinterfaces, or S-VLAN subinterfaces.
- You can specify the following keywords:
  - **delta**—Specifies that baselined statistics are to be shown
  - **brief**—Displays the operational status of all configured interfaces
- Field descriptions

- GigabitEthernet or tenGigabitEthernet *interfaceSpecifier*—Status of the hardware on this interface
  - up—Hardware is operational
  - down—Hardware is not operational
- Administrative status—Operational state that you configured for this interface
  - up—Interface is enabled
  - down—Interface is disabled
- Hardware—Type of MAC device on this interface
- Address—MAC address of the processor on this interface
- MAU—Type of medium attachment unit (MAU) on the primary and secondary physical ports:
  - SFP—1000BASE-LH, 1000BASE-SX, 1000BASE-ZX; for SFPs that are empty, SFP (Empty) appears in this field; for SFPs that are installed in the OC3-2 GE APS I/O module and do not have a Juniper Networks part number programmed, SFP (GE Compliant) appears in this field
  - XFP—10GBASE-SR (10 Gbps), 10GBASE-LR (10 Gbps), 10GBASE-ER (10 Gbps); for XFPs that are empty, XFP (Empty) appears in this field
- TX Output Power—Transmitted output optical power
- RX Input Power—Received input optical power
- MTU—Size of the MTU for this interface
  - Operational—Size of the largest packet processed
  - Administrative—Setting for MTU size that you specified
- Duplex Mode—Duplex option for this interface
  - Operational—Duplex option currently used
  - Administrative—Setting for duplex that you specified
- Speed—Line speed for this interface
  - Operational—Current rate at which packets are processed
  - Administrative—Setting for line speed that you specified
- Debounce—Debounce configuration for this interface
  - State is—Enabled, Disabled
  - Interval is—Number of seconds that this interface maintains a given state before the state change is reported to the upper-layer links
- Clear arp—State of the removal of the ARP entries on an interface with redundant ports
  - Enabled—Clears ARP entries on the interface when the primary link fails

  - Disabled—Maintains ARP entries on the interface until the specified timeout elapses
- Link—Link information for this interface
  - Operational Link Selected—Port that the I/O module is currently using: primary or secondary
  - Administrative link selected—Port that the I/O module is configured to use:
    - primary—Only primary port is configured to operate
    - secondary—Only redundant port is configured to operate
    - automatically—Software controls port redundancy automatically
- Link Failover Timeout — Time to wait for a failed link to be active before the router uses a different active link
- Primary link selected x times—Number of times that the I/O has used the primary port since the module was last rebooted
- Secondary link selected x times—Number of times that the I/O has used the secondary port since the module was last rebooted
- Primary/Secondary link signal detected, Primary/Secondary link signal not detected—Specifies the port (primary or secondary) on which the router detects a signal
- 5 minute input rate—Data rates based on the traffic received in the last 5 minutes
- 5 minute output rate—Data rates based on the traffic sent in the last 5 minutes
- In—Analysis of inbound traffic on this interface
  - Bytes—Number of bytes received in error-free packets
  - Unicast—Number of unicast packets received
  - Multicast—Number of multicast packets received
  - Broadcast—Number of broadcast packets received
  - Errors—Total number of errors in all received packets; some packets might contain more than one error
  - Discards—Total number of discarded incoming packets
  - Mac Errors—Number of incoming packets discarded because of MAC sublayer failures
  - Alignment—Number of incomplete octets received
  - CRC—Number of packets discarded because the checksum that the router computed from the data does not match the checksum generated by the originating devices
  - Too Longs—Number of packets discarded because the size exceeded the MTU
  - Symbol Errors—Number of symbols received that the router did not correctly decode

- Out—Analysis of outbound traffic on this interface
  - Bytes—Number of bytes sent
  - Unicast—Number of unicast packets sent
  - Multicast—Number of multicast packets sent
  - Broadcast—Number of broadcast packets sent
  - Errors—Total number of errors in all transmitted packets; note that some packets might contain more than one error
  - Discards—Total number of discarded outgoing packets
  - Mac Errors—Number of outgoing packets discarded because of MAC sublayer failures
  - Deferred—Number of packets that the router delayed sending because the line was busy. In half duplex mode, a high number of deferrals means the link is very busy with traffic from other stations. In full duplex mode, when the link is always available for transmission, this number is zero.
  - No Carrier—Number of packets sent when carrier sense was unavailable
- Collisions—Analysis of the collisions that occurred
  - Single—Number of packets sent after one collision
  - Multiple—Number of packets sent after multiple collisions
  - Late—Number of packets aborted during sending because of collisions after 64 bytes
  - Excessive—Number of packets not sent because of too many collisions
- Policed Statistics—Number of packets that exceeded the number allowed and were policed (or dropped)
- ARP Statistics—Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface
  - ARP requests—Number of ARP requests
  - ARP responses—Number of ARP responses
  - Errors—Total number of errors in all ARP packets
  - Discards—Total number of discarded ARP packets
- Administrative qos-shaping-mode—QoS shaping mode:
  - disabled—Shaping mode is configured but not operational
  - frame—Statistics are reported about bytes in frames, such as transmitted bytes and dropped bytes.
  - cell—Shaping mode for shaping and policing rates is cell-based; resulting traffic stream conforms exactly to the policing rates configured in downstream devices.

Reports statistics in bytes in cells and accounts for cell encapsulation and padding overhead.

- none—Shaping mode is not configured

- Operational qos-shaping-mode—Actual shaping mode for the interface:
  - disabled
  - frame
  - cell
  - none
- queue—Hardware packet queue associated with the specified traffic class and interface
  - traffic class—Name of traffic class
  - bound to—Interface to which queue is bound
  - Queue length—Length of the queue, in bytes
  - Forwarded packets, bytes—Number of packets and bytes that were forwarded on this queue
  - Dropped committed packets, bytes—Number of committed packets and bytes that were dropped
  - Dropped conformed packets, bytes—Number of conformed packets and bytes that were dropped
  - Dropped exceeded packets, bytes—Number of exceeded packets and bytes that were dropped

- Example—Displays the status of a Gigabit Ethernet interface

```
host1#show interfaces gigabitEthernet 14/0/0
 GigabitEthernet14/0/0 is Up, Administrative status is Up
 Hardware is Intel IXF1104, address is 0090.1a42.0b87
 MAU is 1000BASE-SX
 TX Output Power: 469.6 uW RX Input Power: 0.5 uW
 MTU: Operational 1518, Administrative 1518
 Duplex Mode: Operational Full Duplex, Administrative Auto Negotiate
 Speed: Operational 1000 Mbps, Administrative Auto Negotiate
 Debounce: State is Disabled
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
 In: Bytes 0, Unicast 0
 Multicast 0, Broadcast 0
 Errors 0, Discards 0, Mac Errors 0, Alignment 0
 CRC 0, Too Longs 0, Symbol Errors 0
 Out: Bytes 0, Unicast 0
 Multicast 0, Broadcast 0
 Errors 0, Discards 0, Mac Errors 0, Deferred 0, No Carrier 0
 Collisions: Single 0, Multiple 0, Late 0, Excessive 0
 Policed Statistics:
 In: 0, Out: 0
 ARP Statistics:
 In: ARP requests 0, ARP responses 0
```



```
Errors 0, Discards 0
Out: ARP requests 0, ARP responses 0
```

Example—Displays the status of a 10 Gigabit Ethernet interface

```
host1#show interfaces tenGigabitEthernet 4/0/0
  TenGigabitEthernet4/0/0 is Up, Administrative status is Up
  Hardware is Marvell GT64260, address is 0090.1a42.14b5
  Primary MAU is 10000BASE-LR 10km, secondary MAU is XFP (Empty)
  TX Output Power: 480.8 uW  RX Input Power: 1 uW
  MTU: Operational 1522, Administrative 1522
  Duplex Mode: Operational Full Duplex, Administrative Full Duplex
  Speed: Operational 10000 Mbps, Administrative 10000 Mbps
  Debounce: State is Disabled
  Link: Operational Primary Link Selected,
  Administrative Link Selected Automatically
  Link Failover Timeout: Operational 336 ms, Administrative default
  Primary link selected 1 time, Secondary link selected 0 times
  Primary link signal not detected, Secondary link signal not detected
  Cleararp: State is Disabled
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  In: Bytes 0, Unicast 0
  Multicast 0, Broadcast 0
  Errors 0, Discards 0, Mac Errors 0, Alignment 0
  CRC 0, Too Longs 0, Symbol Errors 0
  Out: Bytes 768, Unicast 0
  Multicast 0, Broadcast 12
  Errors 0, Discards 0, Mac Errors 0, Deferred 0, No Carrier 0
  Collisions: Single 0, Multiple 0, Late 0, Excessive 0
  Policed Statistics:
  In: 0, Out: 0
  ARP Statistics:
  In: ARP requests 0, ARP responses 0
  Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
  Errors 0, Discards 0
```

- See *show interfaces*.

CHAPTER 5

Configuring VLAN and S-VLAN Subinterfaces

This chapter describes how to configure VLAN and S-VLAN subinterfaces on E Series routers.

This chapter contains the following sections:

- VLAN Overview on page 165
- S-VLAN Overview on page 166
- VLAN and S-VLAN Platform Considerations on page 167
- VLAN and S-VLAN References on page 168
- Configuring VLAN Subinterfaces on page 168
- Configuring S-VLAN Subinterfaces on page 175
- Example: Configuring S-VLAN Tunnels for Layer 2 Services over MPLS on page 177
- S-VLAN Oversubscription on page 179
- ACI-based VLAN Subinterfaces per S-VLAN Overview on page 180
- Configuring the Number of ACI-based VLAN Subinterfaces per S-VLAN on page 181

VLAN Overview

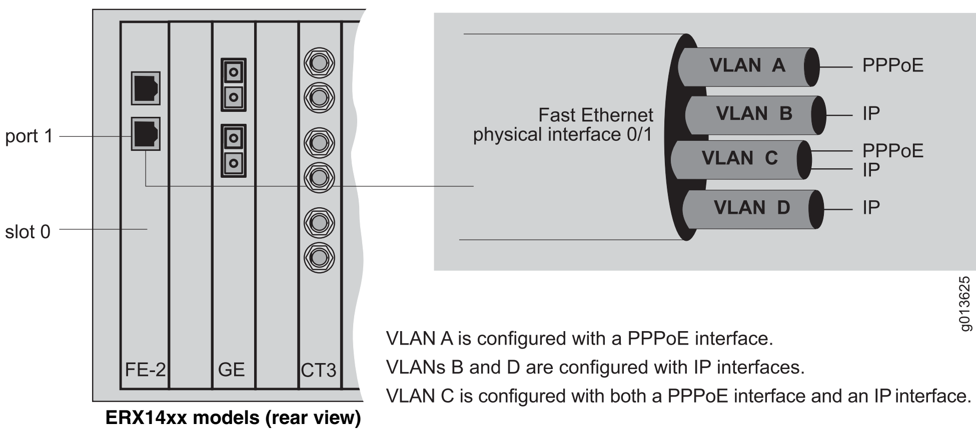
A virtual LAN (VLAN) enables multiplexing multiple IP and PPPoE interfaces and MPLS interfaces over a single physical Ethernet port. This multiplexing is accomplished through VLAN subinterfaces. Ethernet interfaces support the 802.1q-1998 IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks, which the router uses as its standardized format for frame tagging.

The Ethernet V2 frame format enables multiplexing of different protocols over a single physical link. IEEE 802.1q compatibility extends the frame format by adding a tag that contains a VLAN ID. This feature enables multiplexing of different channels (VLANs) over the physical link; each channel is able to multiplex different protocols.

This capability works very much like ATM encapsulation as described in RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999). This encapsulation type enables multiplexing of multiple protocols over a single ATM virtual circuit (VC).

As shown in Figure 13 on page 166, VLANs are similar to ATM VCs, with the VLAN ID serving the same function as the virtual path identifier (VPI) and virtual channel identifier (VCI) to multiplex the different channels over the physical link. The Ethernet protocol type serves the same function within a VLAN as the logical link control (LLC) subnetwork attachment point (SNAP) within a VC, to multiplex the different protocols over the channel.

Figure 13: Use of VLANs to Multiplex Different Protocols over a Single Physical Link



In a VLAN configuration, the router can send VLAN 0 *tagged* or *untagged frames*.

All VLAN subinterfaces use the MAC address of the Ethernet interface over which they are configured. However, some configurations, such as multiple IP over VLAN subinterfaces, require that you connect many VLAN subinterfaces to a single device. In these cases, the device uses the MAC address to identify and select the correct VLAN to use. When the MAC address is the same for all VLANs, uneven load balancing of traffic occurs. To ensure proper load balancing, you must assign unique MAC addresses to the individual VLAN subinterfaces that are connected to the device. Any ARP requests and responses generated for the IP address assigned to a VLAN subinterface use this MAC address.

You must assign the MAC address when you configure the VLAN ID. If you change the MAC address of the VLAN subinterface after you configure it, system errors can occur. To change the MAC address, you must first remove the VLAN subinterface and then reconfigure it.

Related Documentation

- Configuring VLAN Subinterfaces on page 168
- Configuring Point-to-Point Protocol over Ethernet on page 387
- *Configuring IP* in *JunosE IP, IPv6, and IGP Configuration Guide*

S-VLAN Overview

As described in "VLAN Overview" on page 165, VLANs permit multiplexing multiple IP interfaces and PPPoE interfaces over a single physical Ethernet port by creating VLAN subinterfaces. As specified in IEEE Standard 802.1q, the 12-bit VLAN identifier's tagged frames enables the construction of a maximum of 4096 distinct VLANs. In an Ethernet B-RAS application environment, however, this VLAN limit is inadequate. A stacked VLAN (S-VLAN) provides a two-level VLAN tag structure, which extends the VLAN ID space to more than 16 million VLANs.

Creating an S-VLAN requires the use of a second encapsulation tag. The router performs decapsulation twice, once to get the S-VLAN tag and once to get the VLAN tag. This *double tagging* approach enables more than 16 million address possibilities, which more than satisfies the scaling requirement for Ethernet B-RAS applications.

VLAN and S-VLAN subinterfaces can coexist over the same VLAN major interface. You configure S-VLANs in the same way that you configure VLANs, with the addition of certain commands.

NOTE: See *JunosE Release Notes*, *Appendix A*, *System Maximums* for S-VLAN limitations.

Like VLANs, all S-VLAN subinterfaces use the MAC address of the Ethernet interface over which they are configured. For more information about assigning unique MAC address to the S-VLAN subinterface when assigning VLAN or S-VLAN IDs, see "VLAN Overview" on page 165.

Related Documentation

- Configuring S-VLAN Subinterfaces on page 175
- Example: Configuring S-VLAN Tunnels for Layer 2 Services over MPLS on page 177
- S-VLAN Oversubscription on page 179

VLAN and S-VLAN Platform Considerations

You can configure VLAN and S-VLAN subinterfaces on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules supported on E Series routers:

- See the *ERX Module Guide* for modules supported on ERX7xx models, ERX14xx models, and the ERX310 router.
- See the *E120 and E320 Module Guide* for modules supported on the E120 and E320 routers.

Interface Specifiers

The configuration task examples in this chapter use the format for ERX7xx models, ERX14xx models, and the ERX310 router to specify a VLAN or S-VLAN subinterface.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format. For example, the following command specifies a VLAN subinterface configured on port 0 of an I/O module in slot 4.

```
host1(config)#interface fastEthernet 4/0.1
```

For E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. For example, the following command specifies a VLAN subinterface configured on port 0 of the IOA installed in the upper adapter bay of slot 3.

```
host1(config)#interface gigabitEthernet 3/0/0.1
```

For more information about interface types and specifiers on E Series models, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Related Documentation

- VLAN and S-VLAN References on page 168

VLAN and S-VLAN References

For more information about VLAN and S-VLAN implementations, consult the following resources:

- IEEE 802.1q (Virtual LANs)

Related Documentation

- VLAN and S-VLAN Platform Considerations on page 167

Configuring VLAN Subinterfaces

Ethernet interfaces support IP, PPPoE, MPLS, or both IP and PPPoE on each VLAN. In addition to a VLAN major interface level, a VLAN subinterface level distinguishes the VLAN.

NOTE: You cannot configure VLANs on the Fast Ethernet port of the SRP module.

Tasks to configure VLAN subinterfaces are:

- Creating a VLAN Major Interface on page 169
- Configuring IP over VLAN on page 169
- Configuring PPPoE over VLAN on page 170
- Configuring MPLS over VLAN on page 171
- Configuring IP over VLAN and PPPoE over VLAN on page 172

Creating a VLAN Major Interface

To use VLANs, you must first configure the Ethernet interface for VLAN encapsulation. This creates the VLAN major interface.

To configure the Ethernet interface for VLAN encapsulation:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The router creates the VLAN major interface.

You can now create multiple VLAN subinterfaces to carry higher-level protocols. For examples, see "Configuring VLAN Subinterfaces" on page 168, next.

Configuring IP over VLAN

To configure IP over VLAN over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/0.3**

4. Do one of the following:

   a. Assign a VLAN ID for the subinterface.

      host1(config-if)#**vlan id 201**

   b. Assign a VLAN ID and the optional unique MAC address for the subinterface.

      host1(config-if)#**vlan id 201 mac-address 0090.1a01.1234**

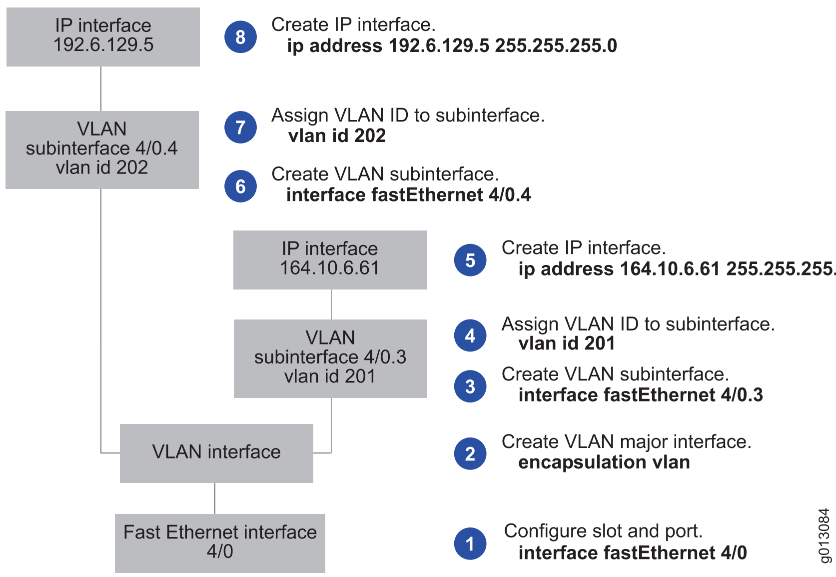
5. Assign an IP address and mask.

host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

6. (Optional) Configure additional VLAN subinterfaces by completing Steps 3 through 5.

Figure 14 on page 170 illustrates the IP/VLAN/Fast Ethernet stacking, showing two separate VLAN subinterfaces. Configure one VLAN subinterface entirely; then configure the next VLAN subinterface.

Figure 14: Example of IP/VLAN/Fast Ethernet Stacking Configuration Procedure



Configuring PPPoE over VLAN

To configure PPPoE over VLAN over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/1**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Do one of the following:

   - Assign a VLAN ID for the subinterface.

     host1(config-if)#**vlan id 201**

   - Assign a VLAN ID and the optional unique MAC address for the subinterface.

     host1(config-if)#**vlan id 201 mac-address 0090.1a01.1234**

5. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe**

6. Create a PPPoE subinterface.

   host1(config-if)#**pppoe subinterface fastEthernet 4/1.1.1**

7. Specify PPP as the encapsulation method on the interface.

   host1(config-if)#**encapsulation ppp**

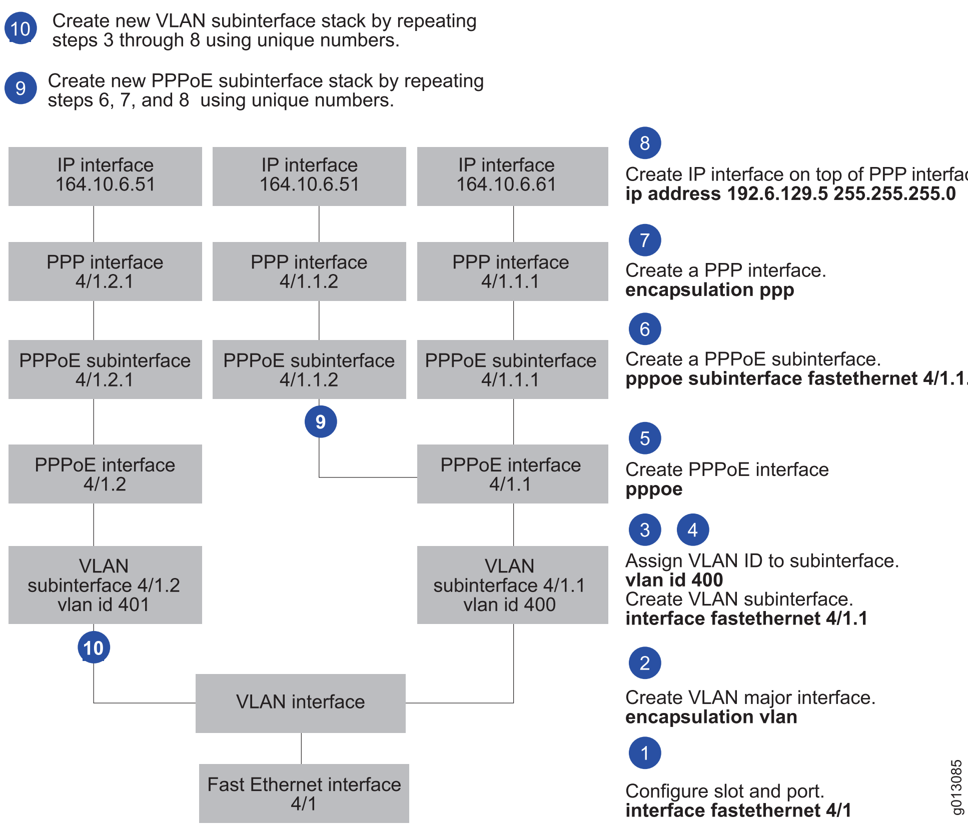
8. Assign an IP address and mask.

   host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

9. (Optional) Configure additional VLAN subinterfaces by completing Steps 3 through 8.

Figure 15 on page 171 illustrates the PPPoE/VLAN/Fast Ethernet stacking, showing two separate VLAN subinterfaces. One VLAN subinterface has two PPPoE subinterfaces, and one VLAN subinterface has one PPPoE subinterface.

Figure 15: Example of PPPoE/VLAN/Fast Ethernet Stacking Configuration Procedure



Configuring MPLS over VLAN

To configure MPLS over VLAN over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

host1(config)#**interface fastEthernet 4/0**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Do one of the following:

   - Assign a VLAN ID for the subinterface.

     host1(config-if)#**vlan id 400**

   - Assign a VLAN ID and the optional unique MAC address for the subinterface.

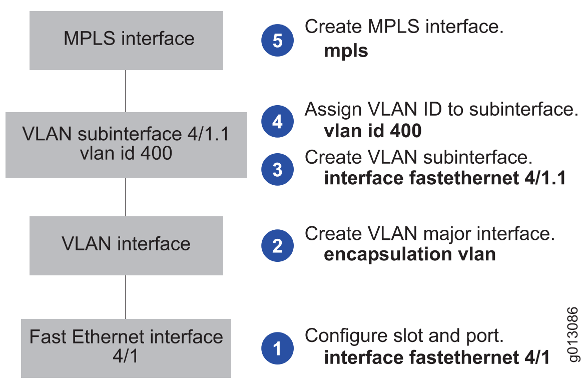
     host1(config-if)#**vlan id 400 mac-address 0090.1a01.1234**

5. Enable MPLS on the interface.

   host1(config-if)#**mpls**

Figure 16 on page 172 illustrates the MPLS/VLAN/Fast Ethernet stacking, showing one VLAN subinterface.

Figure 16: Example of MPLS/VLAN/Fast Ethernet Stacking Configuration Procedure



Configuring IP over VLAN and PPPoE over VLAN

To configure IP over VLAN with PPPoE over the same VLAN over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/1**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Do one of the following:

   - Assign a VLAN ID for the subinterface.

     host1(config-if)#**vlan id 400**

   - Assign a VLAN ID and the optional unique MAC address for the subinterface.

     host1(config-if)#**vlan id 400 mac-address 0090.1a01.1234**

5. Create an IP interface on the same VLAN as the PPPoE interface.

   host1(config-if)#**ip address 164.10.6.71 255.255.255.0**

6. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe**

7. Create a PPPoE subinterface.

   host1(config-if)#**pppoe subinterface fastEthernet 4/1.1.1**

8. Specify PPP as the encapsulation method on the interface.

   host1(config-if)#**encapsulation ppp**

9. Assign an IP address and mask.

   host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

10. (Optional) Configure additional PPPoE subinterfaces by completing Steps 7 through 9 using unique numbering.

To configure additional IP interfaces over the VLAN major interface:

1. Create a new VLAN subinterface by adding a unique subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.2**

2. Assign a VLAN ID for the subinterface.

   host1(config-if)#**vlan id 401**

3. Assign an IP address and mask.

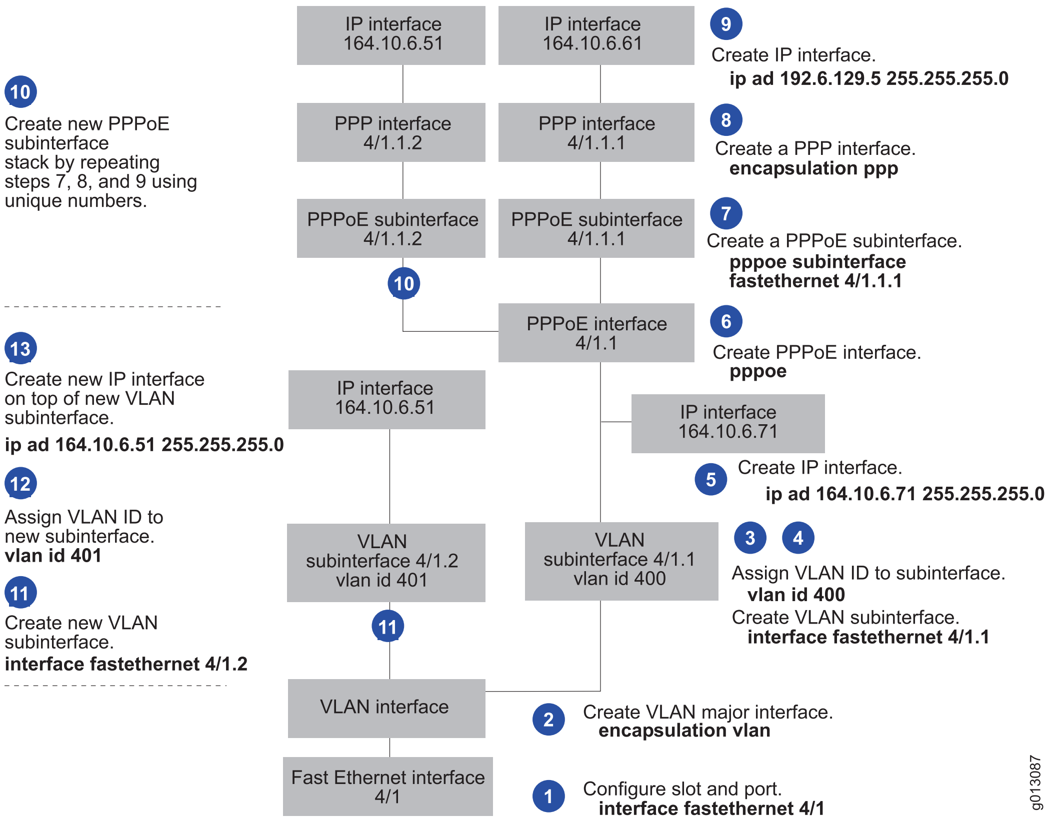
   host1(config-if)#**ip address 164.10.6.51 255.255.255.0**

Figure 17 on page 174 illustrates the configuration steps for two VLAN subinterfaces. In this example:

- VLAN subinterface 4/1.1 has an IP interface, a PPPoE interface, and multiple PPPoE subinterface stacks.
- VLAN subinterface 4/1.2 has only an IP interface.

NOTE: Before you can remove a VLAN subinterface, you must remove the upper-layer interface stack.

Figure 17: Example of PPPoE over VLAN with IP over VLAN Stacking Configuration Procedure



Related Documentation

- VLAN Overview on page 165
- *encapsulation ppp*
- *encapsulation vlan*
- *interface fastEthernet*
- *ip address*
- *mpls*
- *pppoe*
- *pppoe subinterface*
- *vlan id*

Configuring S-VLAN Subinterfaces

Tasks to configure an S-VLAN subinterface include:

- Configuring an S-VLAN Subinterface on page 175
- Configuring PPPoE over an S-VLAN on page 175

Configuring an S-VLAN Subinterface

To configure an S-VLAN subinterface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Assign an S-VLAN ID and a VLAN ID for the subinterface.

   host1(config-if)#**svlan id 4 255**

5. Assign an S-VLAN Ethertype.

   host1(config-if)#**svlan ethertype 88a8**

Configuring PPPoE over an S-VLAN

To configure PPPoE over an S-VLAN over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Assign an S-VLAN ID and a VLAN ID for the subinterface.

   host1(config-if)#**svlan id 4 255**

5. Assign an S-VLAN Ethertype.

   host1(config-if)#**svlan ethertype 88a8**

6. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#pppoe

7. Create a PPPoE subinterface.

   host1(config-if)#pppoe subinterface fastEthernet 4/1.1.1

8. Specify PPP as the encapsulation method on the interface.

   host1(config-if)#encapsulation ppp

9. Assign an IP address and mask.

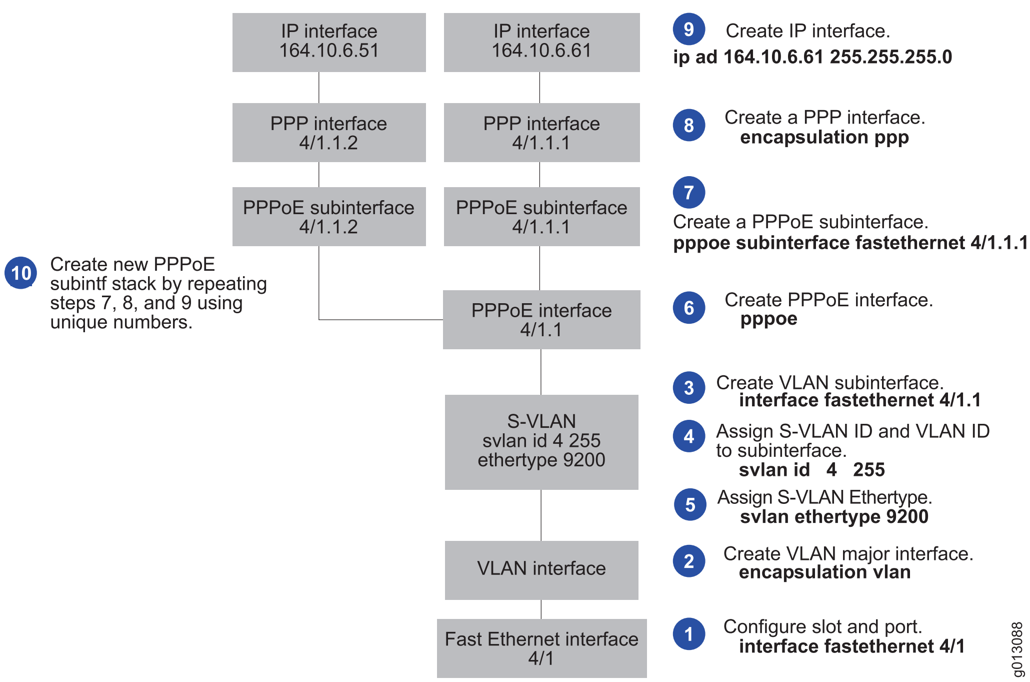
   host1(config-if)#ip address 164.10.6.61 255.255.255.0

10. (Optional) Configure additional PPPoE subinterfaces by completing Steps 7 through 9 using unique numbering.

Figure 18 on page 176 shows one S-VLAN subinterface with multiple PPPoE subinterface stacks.

NOTE: **Before you can remove an S-VLAN/VLAN subinterface, you must remove the upper-layer interface stack.**

Figure 18: Example of PPPoE over S-VLAN Stacking Configuration Procedure



Related Documentation

- S-VLAN Overview on page 166
- *encapsulation ppp*
- *encapsulation vlan*
- *interface fastEthernet*

- *ip address*
- *pppoe*
- *pppoe subinterface*
- *svlan ethertype*
- *svlan id*

Example: Configuring S-VLAN Tunnels for Layer 2 Services over MPLS

The example in this section illustrates how to configure S-VLAN tunnels for Ethernet layer 2 services over MPLS.

- Requirements on page 177
- Overview of Configuring S-VLAN Tunnels for Ethernet Layer 2 Services over MPLS on page 177
- Configuring S-VLAN Tunnels for Ethernet Layer 2 Services over MPLS on page 178

Requirements

This example uses the following software and hardware components:

- JunosE Release 7.1.0 or higher-numbered releases
- E Series router (ERX7xx models, ERX14xx models, the ERX310 router, the E120 router, or the E320 router)
- ASIC-based line modules that support Fast Ethernet or Gigabit Ethernet

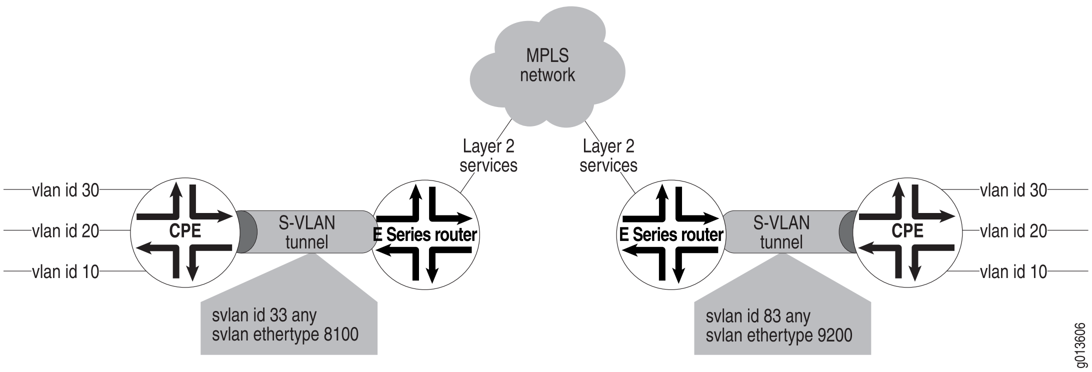
Overview of Configuring S-VLAN Tunnels for Ethernet Layer 2 Services over MPLS

When you configure Ethernet layer 2 services over MPLS, you can create a special type of S-VLAN called an S-VLAN tunnel that uses a single interface to tunnel traffic from multiple VLANs across an MPLS network. The S-VLAN tunnel enables multiple VLANs, each configured with a unique VLAN ID tag, to share a common S-VLAN ID tag when they traverse an MPLS network.

Advantages of using S-VLAN Tunnels

Using S-VLAN tunnels provides an easier and faster way to configure Ethernet layer 2 services over MPLS than using standard S-VLANs. For example, consider the network configuration shown in Figure 19 on page 178.

Figure 19: S-VLAN Tunnels for Ethernet Layer 2 Services over MPLS



In this example, traffic from three VLAN subinterfaces must traverse the MPLS network. To accomplish this using standard S-VLANs, you issue the following commands to configure three separate S-VLANs with the same S-VLAN ID value and different VLAN IDs, as follows:

```
host1(config-if)#svlan id 33 10
host1(config-if)#svlan id 33 20
host1(config-if)#svlan id 33 30
```

By contrast, using an S-VLAN tunnel achieves the same result, but requires you to issue only a single **svlan id** command with the keyword **any** in place of the VLAN ID value. For example, the following command creates a single interface that tunnels traffic from VLANs configured with an S-VLAN ID of 33 and *any* VLAN ID to the same destination across the MPLS network. In effect, this command tunnels traffic from all three VLANs shown in Figure 19 on page 178.

```
host1(config-if)#svlan id 33 any
```

Interface Stacking

When you configure Ethernet layer 2 services over MPLS using S-VLAN tunnels, the only interface that you can stack over an S-VLAN tunnel is an MPLS tunnel, which you configure using the MPLS tunneling command (**mpls-relay** or **route interface**) that is appropriate for your configuration. Attempting to configure any other interface type—such as IP, MPLS (nontunnel), or PPPoE—over the S-VLAN tunnel causes the router to generate an error and reject the configuration as invalid.

For details about configuring MPLS and layer 2 services over MPLS, see:

- *Configuring MPLS* in *JunosE BGP and MPLS Configuration Guide*
- *Configuring Layer 2 Services over MPLS* in *JunosE BGP and MPLS Configuration Guide*

Configuring S-VLAN Tunnels for Ethernet Layer 2 Services over MPLS

This section uses the sample network topology shown in Figure 19 on page 178 to illustrate the steps for configuring S-VLAN tunnels for Ethernet layer 2 services over MPLS.

To configure S-VLAN tunnels for Ethernet layer 2 services over MPLS:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.


```
host1(config)#interface fastEthernet 4/0
```

2. Specify VLAN as the encapsulation method to create the VLAN major interface.

```
host1(config-if)#encapsulation vlan
```

3. Create a VLAN subinterface.

```
host1(config-if)#interface fastEthernet 8/1.1
```

4. Create the S-VLAN tunnel. This interface tunnels traffic from VLANs configured with an S-VLAN ID of 33 and any VLAN ID to the same destination across the MPLS network.

```
host1(config-if)#svlan id 33 any
```

5. Assign an S-VLAN Ethertype.

```
host1(config-if)#svlan ethertype 8100
```

6. Create the MPLS tunnel interface using the appropriate MPLS tunneling command for your configuration. For example:

```
host1(config-if)#route interface tunnel mpls:tunnel3 45
```

   For complete instructions on configuring the MPLS tunnel, see *Configuring Layer 2 Services over MPLS* in *JunosE BGP and MPLS Configuration Guide*.

7. Repeat Steps 1 through 6 using unique values to configure the S-VLAN tunnel and MPLS tunnel interfaces on the remote E Series router. For example:

```
host2(config)#interface fastEthernet 3/1
host2(config-if)#encapsulation vlan
host2(config-if)#interface fastEthernet 3/1.1
host2(config-if)#svlan id 83 any
host2(config-if)#svlan ethertype 88a8
host2(config-if)#route interface tunnel mpls:tunnel2 45
```

Related Documentation

- Configuring S-VLAN Subinterfaces on page 175
- *encapsulation vlan*
- *interface fastEthernet*
- *route interface*
- *svlan ethertype*
- *svlan id*

S-VLAN Oversubscription

When you configure S-VLAN subinterfaces over Ethernet interfaces to support dynamic PPPoE subinterfaces, you can take advantage of S-VLAN oversubscription.

The following module combinations support S-VLAN oversubscription:

- GE/FE line module and all of its associated I/O modules
- GE-2 line module and the GE-2 SFP I/O module

- GE-HDE line module and its associated I/O modules
- OC3/STM1 GE/FE line module and the OC3-2 GE APS I/O module
- ES2 4G LM and its associated Gigabit Ethernet and 10-Gigabit Ethernet IOAs
- ES2 10G LM and its associated Gigabit Ethernet and 10-Gigabit Ethernet IOAs

The maximum number of S-VLANs that you can create per I/O module with PPPoE major interfaces stacked over them is greater than the maximum number of dynamic PPPoE subinterfaces. The maximum number of PPP interfaces supported per line module is directly proportional to the maximum number of PPPoE subinterfaces.

As a result, you can oversubscribe S-VLANs by configuring up to the maximum number of S-VLANs supported on these I/O modules, knowing that no more than the maximum number of supported PPP sessions can be connected to the router at any one time.

For configuration instructions, see "Configuring Dynamic PPPoE over Static PPPoE with Ethernet and S-VLAN Interface Columns" on page 549.

For specific information about the maximum number of S-VLANs supported per I/O module and the maximum number of PPP interfaces and PPPoE subinterfaces supported per line module, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

NOTE: The E120 and E320 routers can support up to two IOAs per line module. This maximum number of S-VLANs per line module does not change if one or two IOAs are installed.

Related Documentation

- S-VLAN Overview on page 166

ACI-based VLAN Subinterfaces per S-VLAN Overview

JunosE Software supports Agent Circuit Identifier-based (ACI) bulk configuration to build VLAN subinterfaces per S-VLAN on a VLAN or an S-VLAN subinterface. You can configure the number of ACI-based VLAN subinterfaces per S-VLAN at the interface level. The configuration settings are saved on the VLAN major interface. However, configuring the number of subinterfaces when the system is already operational or changing its value does not impact the existing VLAN subinterfaces, even if that number exceeds the newly configured limit value. Limiting the number of VLAN subinterfaces helps meet increased per subscriber bandwidth usage.

Related Documentation

- S-VLAN Overview on page 166
- Configuring S-VLAN Subinterfaces on page 175
- Configuring the Number of ACI-based VLAN Subinterfaces per S-VLAN on page 181

Configuring the Number of ACI-based VLAN Subinterfaces per S-VLAN

You can limit the number of ACI-based VLAN subinterfaces per S-VLAN at the interface level to a configurable value.

To configure the number of ACI-based VLAN subinterfaces per S-VLAN:

- From Interface Configuration mode, configure the number of VLAN subinterfaces.

  host1(config-if)#**max aci-svs per-pvs limitvalue**

The ACI-based VLAN subinterface limit is visible using the **show configuration** command.

NOTE: Even if you configure the maximum number of ACI-based VLAN subinterfaces per S-VLAN on a VLAN or an S-VLAN subinterface using the **max aci-svs per-pvs limitvalue** command, the setting is saved on the VLAN major interface. This behavior is expected and you can verify the configured value from the output of the **show configuration interface** command.

Related Documentation

- ACI-based VLAN Subinterfaces per S-VLAN Overview on page 180
- *max aci-svs per-pvs*

CHAPTER 6

Monitoring VLAN and S-VLAN Subinterfaces

This chapter describes how to monitor VLAN and S-VLAN subinterfaces on E Series routers. You can use the **show** commands described in this chapter to display information about your Ethernet configuration and to monitor Ethernet interfaces.

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

This chapter contains the following sections:

- Monitoring VLAN Subinterfaces on page 183
- Monitoring Interface Rate Statistics for VLAN Subinterfaces on page 186
- Monitoring Fast Ethernet VLAN or S-VLAN Subinterfaces on page 188
- Monitoring Gigabit Ethernet or 10-Gigabit Ethernet VLAN or S-VLAN Subinterfaces on page 190

Monitoring VLAN Subinterfaces

Purpose Display configuration and status information for a specified VLAN subinterface or for all VLAN subinterfaces configured on the router. You can use the **summary** keyword to display only the counts of all VLAN subinterfaces and VLAN major interfaces configured on the router. You can use the **mac-address** keyword to display information about the VLAN subinterfaces that were configured with unique MAC addresses. You can use the **vlan** or **svlan** keywords to display information about specific S-VLAN IDs or VLAN IDs.

Action To display full status and configuration information for all VLAN subinterfaces configured on the router:

```
host1#show vlan subinterface
    Interface       Status   MTU  Svlan Id Vlan Id Ethertype Type
----------------- -------- ---- -------- ------ --------- -------
ATM 3/0.1.2         Up      1522 ----     11      ----      Static
ATM 3/0.1.3         Up      1522 ----     12      ----      Static
ATM 3/1.1.1         Up      1522 ----     13      ----      Static
```

```
ATM 3/1.1.2         Up     1522 ----     14      ----      Static
ATM 3/2.1.1         Down   1526 4        255     0x9100    Static
FastEthernet 4/5.1  Up      1522 ----     1       ----      Dynamic
6 vlan subinterfaces found
```

To display full status and configuration information for the specified VLAN subinterface:

```
host1#show vlan subinterface fastEthernet 0/0.1
    Interface       Status  MTU  Svlan Id Vlan Id Ethertype Type
------------------- ------ ---- -------- ------ --------- -------
FastEthernet 0/0.1    Up    1526    1      0    0x9100    Static

In: Bytes 39256, Packets 612
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
  Out: Bytes 4538652, Packets 70911
   Multicast 0, Broadcast 70296
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

To display only brief summary information for all VLAN subinterfaces configured on the router:

```
host1#show vlan subinterface summary
Total VLAN interfaces: 6 subinterfaces, 3 major interfaces
```

To display full status and configuration information for all VLAN subinterfaces configured with a unique MAC address:

```
host1#show vlan subinterface mac-address
     Interface          Svlan Id Vlan Id   MAC Address
----------------------  -------- --------  --------------
FastEthernet 4/0.25       ----     25    0090.dfad.2abd
FastEthernet 4/0.10050     1       50    0090.adad.0abd
2 vlan subinterfaces found
```

To display full status and configuration information for a VLAN subinterface on a LAG bundle:

```
host1#show vlan subinterface lag boston.1
    Interface     Status  MTU  Svlan Id Vlan Id Ethertype  Type
---------------- -------- ---- -------- -------- --------- ------
lag boston.1       Up     1522 ----     1        ----      Static
```

To display full status and configuration information for the specified S-VLAN ID:

```
host1#show vlan subinterface svlan 100 53
      Interface         Status  MTU  Svlan Id Vlan Id Ethertype  Type
------------------------ ------ ---- -------- -------- --------- -------
FastEthernet 0/0.1        Up    1526 100      53      0x9100    Static
FastEthernet 4/6.1        Up    1526 100      53      0x9100    Dynamic
2 vlan subinterfaces found
```

Meaning Table 10 on page 185 lists the **show vlan subinterface** command output fields.

Table 10: show vlan subinterface Output Fields

Field Name	Field Description
Interface	Type and specifier of the VLAN subinterface
Status	Status of the VLAN subinterface: up, down, dormant, lowerLayerDown, absent
MTU	Maximum allowable size (in bytes) of the maximum transmission unit (MTU) for the VLAN subinterface
Svlan Id	S-VLAN ID value, if configured
Vlan Id	VLAN ID value for the VLAN subinterface
Ethertype	S-VLAN Ethertype value, if configured
Type	Type of VLAN subinterface: • Static—VLAN or S-VLAN subinterface was configured statically • Dynamic—VLAN or S-VLAN subinterface was configured dynamically
In	Analysis of inbound traffic on this interface: • Bytes—Number of bytes received on the VLAN or S-VLAN subinterface • Packets—Sum of all unicast, broadcast, and multicast packets received on the VLAN or S-VLAN subinterface • Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface • Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface • Errors—Total number of errors in all received packets; some packets might contain more than one error • Discards—Total number of discarded incoming packets
Out	Analysis of outbound traffic on this interface: • Bytes—Number of bytes sent on the VLAN or S-VLAN subinterface • Packets—Number of packets sent on the VLAN or S-VLAN subinterface • Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface • Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface • Errors—Total number of errors in all transmitted packets; some packets might contain more than one error • Discards—Total number of discarded outgoing packets

Table 10: show vlan subinterface Output Fields *(continued)*

Field Name	Field Description
ARP Statistics	Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface • ARP requests—Number of ARP requests • ARP responses—Number of ARP responses • Errors—Total number of errors in all ARP packets • Discards—Total number of discarded ARP packets
Total VLAN interfaces	Total numbers of VLAN subinterfaces and VLAN major interfaces configured on the router; this is the only field that appears when you specify the **summary** keyword

Related Documentation

- Monitoring Fast Ethernet VLAN or S-VLAN Subinterfaces on page 188
- Monitoring Gigabit Ethernet or 10-Gigabit Ethernet VLAN or S-VLAN Subinterfaces on page 190
- Monitoring Interface Rate Statistics for VLAN Subinterfaces on page 186
- *show vlan subinterface*

Monitoring Interface Rate Statistics for VLAN Subinterfaces

Purpose Display bit rate and packet rate statistics over a specified time interval for one or more VLAN subinterfaces configured on the router.

Action To display interface rate statistics for VLAN subinterfaces:

1. Log in to the router by using a local console session or a virtual terminal (vty) session (such as a Telnet session).

   While you are using the **monitor vlan interface** command, you must keep the console or terminal session open and you cannot issue any other commands at the session during this time.

   For information about logging in to the router, see *Accessing the CLI* in *JunosE System Basics Configuration Guide*.

2. Access User Exec mode or Privileged Exec mode.

   For information, see *Accessing Command Modes* in *JunosE System Basics Configuration Guide*.

3. Specify the interface identifier for each VLAN subinterface that you want to monitor.

   ```
   host1#monitor vlan interface fastEthernet 0/0.1 fastEthernet 4/0.1 display-time-of-day
   ```

   For information about specifying interface identifiers for VLAN subinterfaces configured over Ethernet interfaces, see "VLAN Overview" on page 165. For information about specifying interface identifiers for VLAN subinterfaces configured over LAG bundles, see "Configuring a VLAN Subinterface for a LAG Bundle" on page 199.

By default, the router uses a 5-second time interval between polls to calculate bit rates and packet rates for each specified VLAN subinterface. Optionally, you can use the **load-interval** keyword to specify a nondefault time interval in the range 5–30 seconds.

You can also include the optional **display-time-of-day** keyword to show the time of day at which the router gathers bit rate and packet rate statistics for each interval. Displaying the time of day enables you to monitor when a particular VLAN subinterface is underutilized or overutilized.

4. Review the command output.

```
host1#monitor vlan interface fastEthernet 0/0.1 fastEthernet 4/0.1
display-time-of-day
                         Seconds
                         between                                         Time
       Interface          polls    Input bps/pps    Output bps/pps    (UTC)
----------------------- -------- ---------------- ---------------- --------
FastEthernet 0/0.1             0            --/--            --/-- 10:50:07
FastEthernet 4/0.1             0            --/--            --/-- 10:50:07
FastEthernet 0/0.1             5       120240/100       120240/100 10:50:12
FastEthernet 4/0.1             5       120000/100       120000/100 10:50:12
FastEthernet 0/0.1             5       120240/100       120240/100 10:50:17
FastEthernet 4/0.1             5       120000/100       120000/100 10:50:17
```

The router polls each VLAN subinterface at the specified load interval (the default 5-second interval in this example) to calculate and display bit rate and packet rate statistics. The first line of output for each interface always displays 0 (zero) for the number of seconds between polls, and dashes (--/--) in the Input bps/pps and Output bps/pps columns. These values indicate that the router initially takes a baseline for each interface against which to measure subsequent statistics. The router continues to display subsequent lines of output for each interface at the specified load interval until you press Ctrl+c to stop the command.

For a description of each field in the **monitor vlan interface** command output, see Table 11 on page 188.

5. When you are finished, press Ctrl+c to stop the **monitor vlan interface** command.

   host1#^C

To display bit rate and packet rate statistics over the default (5-second) load interval for a single VLAN subinterface:

```
host1#monitor vlan interface fastEthernet 0/0.1
                         Seconds
                         between
       Interface          polls    Input bps/pps    Output bps/pps
----------------------- -------- ---------------- ----------------
FastEthernet 0/0.1             0            --/--            --/--
FastEthernet 0/0.1             5       120240/100       120240/100
FastEthernet 0/0.1             5       120000/100       120000/100
FastEthernet 0/0.1             5         92400/77         92400/77
FastEthernet 0/0.1             5         88800/74         88800/74
```



```
FastEthernet 0/0.1              5       120000/100       120000/100
host1#^C
```

To display bit rate and packet rate statistics over a 10-second load interval for two VLAN subinterfaces, with the time of day that the statistics were calculated:

```
host1#monitor vlan interface fastEthernet 0/0.1 fastEthernet 4/0.1
load-interval 10 display-time-of-day
                        Seconds
                        between                                          Time
       Interface         polls    Input bps/pps    Output bps/pps    (UTC)
----------------------- -------- --------------- --------------- --------
FastEthernet 0/0.1             0          --/--            --/-- 10:50:33
FastEthernet 4/0.1             0          --/--            --/-- 10:50:33
FastEthernet 0/0.1            10     120120/100       120120/100 10:50:43
FastEthernet 4/0.1            10     120000/100       120000/100 10:50:43
FastEthernet 0/0.1            10     120000/100       120000/100 10:50:53
FastEthernet 4/0.1            10     120000/100       120000/100 10:50:53
host1#^C
```

Meaning Table 11 on page 188 lists the **monitor vlan interface** command output fields.

Table 11: monitor vlan interface Output Fields

Field Name	Field Description
Interface	Interface identifier for the Ethernet or LAG interface on which the VLAN subinterface resides
Seconds between polls	Number of seconds at which the router calculates bit rate and packet rate statistics
Input bps/pps	Number of bits per second (bps) and packets per second (pps) received on this interface during the specified load interval
Output bps/pps	Number of bits per second (bps) and packets per second (pps) transmitted on this interface during the specified load interval
Time	Time of day, in hh:mm:ss format, at which the router calculates the bit rate and packet rate statistics for the current intervalS-VLAN Ethertype value, if configured

Related Documentation

- Monitoring VLAN Subinterfaces on page 183
- *monitor vlan interface*

Monitoring Fast Ethernet VLAN or S-VLAN Subinterfaces

Purpose Display the status of Fast Ethernet interfaces, VLAN subinterfaces, or S-VLAN subinterfaces. You can use the **delta** keyword to specify that baselined statistics are to be shown. You can use the **brief** keyword to display the operational status of all configured interfaces.

Action To display the status of a Fast Ethernet VLAN subinterface:

```
host1:vr2#show interfaces fastEthernet 8/3.1
FastEthernet8/3.1 is Up, Administrative status is Up
  VLAN ID: 10, address 0090.5e00.0001

In: Bytes 39256, Packets 612
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
Out: Bytes 4536220, Packets 70873
   Multicast 0, Broadcast 70258
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 1, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 1, ARP responses 0
   Errors 0, Discards 0
```

To display the status of a Fast Ethernet S-VLAN subinterface:

```
host1:vr2#show interfaces fastEthernet 0/0.1
FastEthernet0/0.1 is Up, Administrative status is Up
  SVLAN ID: 1, VLAN ID: 0, Ethertype 0x9100

  In: Bytes 39256, Packets 612
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
  Out: Bytes 4536220, Packets 70873
   Multicast 0, Broadcast 70258
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

Meaning Table 12 on page 189 lists the **show interfaces fastEthernet** command output fields.

Table 12: show interfaces fastEthernet Output Fields

Field Name	Field Description
Subinterface number	Location of the subinterface that carries the VLAN or S-VLAN traffic
Administrative status	Operational state that you configured for this interface; up or down
VLAN ID	Domain number of the VLAN
SVLAN ID	Domain number of the stacked VLAN
Ethertype	Ethertype assignment for the S-VLAN subinterface, 0x8100, 0x88a8, or 0x9100; 0x9100 is the default

Table 12: show interfaces fastEthernet Output Fields *(continued)*

Field Name	Field Description
In	Analysis of inbound traffic on this interface: • Bytes—Number of bytes received on the VLAN or S-VLAN subinterface • Packets—Sum of all unicast, broadcast, and multicast packets received on the VLAN or S-VLAN subinterface • Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface • Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface • Errors—Total number of errors in all received packets; some packets might contain more than one error • Discards—Total number of discarded incoming packets
Out	Analysis of outbound traffic on this interface: • Bytes—Number of bytes sent on the VLAN or S-VLAN subinterface • Packets—Number of packets sent on the VLAN or S-VLAN subinterface • Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface • Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface • Errors—Total number of errors in all transmitted packets; some packets might contain more than one error • Discards—Total number of discarded outgoing packets

Related Documentation

- Monitoring VLAN Subinterfaces on page 183
- *show interfaces*

Monitoring Gigabit Ethernet or 10-Gigabit Ethernet VLAN or S-VLAN Subinterfaces

Purpose

Display the status of Gigabit Ethernet interfaces, 10-Gigabit Ethernet interfaces, VLAN subinterfaces, or S-VLAN subinterfaces. You can use the **delta** keyword to specify that baselined statistics are to be shown. You can use the **brief** keyword to display the operational status of all configured interfaces.

Action

To display the status of a Gigabit Ethernet VLAN subinterface:

```
host1:vr2#show interfaces gigabitEthernet 2/0.1
GigabitEthernet2/0.1 is Up, Administrative status is Up
  VLAN ID: 10, address 0090.5e00.0001

  In: Bytes 2357, Packets 23
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
  Out: Bytes 4872, Packets 57
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
```

```
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

To display the status of a Gigabit Ethernet S-VLAN subinterface:

```
host1:vr2#show interfaces gigabitEthernet 2/0.2
GigabitEthernet2/0.2 is Up, Administrative status is Up
  SVLAN ID: 10, VLAN ID: 100, Ethertype 0x9100

  In: Bytes 2357, Packets 23
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
  Out: Bytes 4872, Packets 57
   Multicast 0, Broadcast 57
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

Meaning Table 13 on page 191 lists the **show interfaces gigabitEthernet** command output fields.

Table 13: show interfaces gigabitEthernet Output Fields

Field Name	Field Description
Subinterface number	Location of the subinterface that carries the VLAN or S-VLAN traffic
Administrative status	Operational state that you configured for this interface; up or down
VLAN ID	Domain number of the VLAN
SVLAN ID	Domain number of the stacked VLAN
Ethertype	Ethertype assignment for the S-VLAN subinterface, 0x8100, 0x88a8, or 0x9100; 0x9100 is the default
In	Analysis of inbound traffic on this interface: • Bytes—Number of bytes received on the VLAN or S-VLAN subinterface • Packets—Sum of all unicast, broadcast, and multicast packets received on the VLAN or S-VLAN subinterface • Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface • Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface • Errors—Total number of errors in all received packets; some packets might contain more than one error • Discards—Total number of discarded incoming packets

Table 13: show interfaces gigabitEthernet Output Fields *(continued)*

Field Name	Field Description
Out	Analysis of outbound traffic on this interface: • Bytes—Number of bytes sent on the VLAN or S-VLAN subinterface • Packets—Number of packets sent on the VLAN or S-VLAN subinterface • Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface • Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface • Errors—Total number of errors in all transmitted packets; some packets might contain more than one error • Discards—Total number of discarded outgoing packets

Related Documentation

- Monitoring VLAN Subinterfaces on page 183
- *show interfaces*

CHAPTER 7

Configuring 802.3ad Link Aggregation and Link Redundancy

This chapter describes how to configure 802.3ad link aggregation and link redundancy on E Series routers.

This chapter contains the following sections:

- Understanding IEEE 802.3ad Link Aggregation on page 193
- 802.3ad Link Aggregation Platform Considerations on page 196
- 802.3ad Link Aggregation References on page 197
- Configuring 802.3ad Link Aggregation on page 197
- Example: Configuring 802.3ad Link Aggregation on page 200
- Ethernet Link Redundancy Overview on page 206
- Ethernet Link Redundancy Behavior Overview on page 211
- Configuring Ethernet Link Redundancy on page 215
- Monitoring a Specified Ethernet Member Link in an IEEE 802.3ad LAG Bundle on page 216
- Monitoring Ethernet Member Links in all IEEE 802.3ad LAG Bundles on page 221

Understanding IEEE 802.3ad Link Aggregation

IEEE 802.3ad link aggregation enables you to group Ethernet interfaces at the physical layer to form a single link layer interface, also known as a link aggregation group (LAG) or bundle. For more information, see IEEE Standard 802.3ad, Link Aggregation.

Some users require more bandwidth in their network than a single Fast Ethernet link can provide, but cannot afford the expense or do not need the bandwidth of a higher-speed Gigabit Ethernet link. Using IEEE 802.3ad link aggregation in this situation provides increased port density and bandwidth at a lower cost. For example, if you need 450 Mbps of bandwidth to transmit data and have only a 100-Mbps Fast Ethernet link, creating a LAG bundle containing five 100-Mbps Fast Ethernet links is more cost-effective than purchasing a single Gigabit Ethernet link.

For information about the modules that support link aggregation, see *ERX Module Guide, Appendix A, Module Protocol Support* and *E120 and E320 Module Guide, Appendix A, IOA Protocol Support*.

LACP

The Link Aggregation Control Protocol (LACP) is a mechanism for exchanging port and system information to create and maintain LAG bundles. The LAG bundle distributes MAC clients across the link layer interface and collects traffic from the links to present to the MAC clients of the LAG bundle.

To create the links in the LAG bundles, you can add one or more Ethernet physical interfaces to it. The LACP detects Ethernet interfaces as links if they are configured on the same line module and have the same physical layer characteristics. The LACP also assigns to the LAG bundle the same MAC address of the Ethernet link with the highest port priority, which is the lowest value.

The LACP also controls the exchange of LACP protocol data units (PDUs) between the Ethernet links in the LAG bundle. The PDUs contain information about each link and enable the LAG bundle to maintain them.

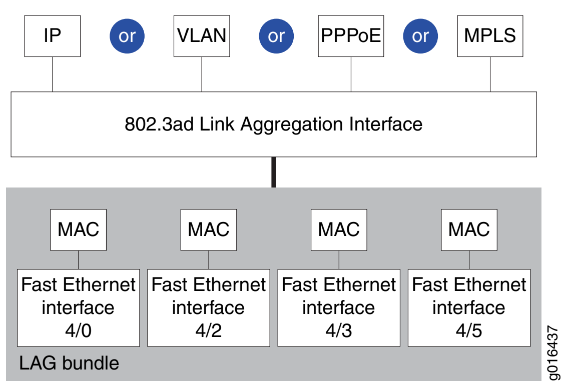
By default, Ethernet links do not exchange PDUs, which contain information about the state of the link. You can configure an Ethernet link to actively or passively transmit PDUs by sending out LACP PDUs only when it receives the PDUs from another link. The transmitting link is known as the *Actor* and the receiving link is known as the *Partner*.

Higher-Level Protocols

After you configure the LAG bundle, you can route IP traffic over it, create a VLAN over it, route PPPoE traffic over it, or route MPLS traffic over it.

Figure 20 on page 194 displays the interface stack for 802.3ad link aggregation.

Figure 20: Interface Stack for 802.3ad Link Aggregation



For information about configuring higher-level protocols over VLANs, see "Configuring VLAN and S-VLAN Subinterfaces" on page 165.

NOTE: On the ES2 10G LM and ES2-S1 GE-8 IOA combination, you can configure only IP or VLAN over a LAG bundle.

Load Balancing and QoS

You can configure load balancing across 802.3ad links to provide quality of service (QoS). To ensure that QoS is symmetrically applied to all the links, the router periodically rebalances the traffic on the LAG. When you attach a QoS profile to the LAG, the load balancing properties that are configured are applied to the LAG, which determines how traffic is distributed.

For example, if VLANs are configured, IP queues are provisioned over the VLANs. In this case, the default behavior is per-VLAN load balancing.

For more information, see *Providing QoS for Ethernet Overview*.

Ethernet Link Aggregation and MPLS

CE-side load balancing in a Martini layer 2 transport environment enables an E Series router to interoperate with an 802.3ad switch in a topology designed for Ethernet link aggregation. See *Configuring Layer 2 Services over MPLS* in *JunosE BGP and MPLS Configuration Guide* for more information.

IPv6 Packets and LAG

You can configure IPv6 prefix addresses for a LAG bundle and for VLAN and PPPoE subinterfaces that are members of a LAG bundle. In environments in which an independent or a dual-stack IPv6 subscriber exists, the PPP link between the customer premises equipment (CPE) and the provider edge (PE) router might require both IPv4 and IPv6 addresses for transmission of data. In such networks, you can use LAG bundles configured with IPv6 addresses for effective usage of bandwidth and reduced administrative costs.

The selection of the member interface in a LAG bundle is performed on the egress side of the module. For IPv6 traffic, the hash value is calculated using the IPv6 source and destination addresses. One of the links, based on the hashing of the IPv6 source and destination addresses, is used to send the packets out of the router. Hashed mode is the default equal-cost multipath (ECMP) mode of operation.

The egress side of the module contains eight segments. Each segment maps to one interface in the LAG bundle. If the member interfaces in a LAG bundle are fewer than eight, the segment is filled by repeating the member interface for that bundle.

You cannot aggregate links with IPv6 traffic from multiple line modules, bridge IPv6 packets over a LAG bundle, or use Link Aggregation Control Protocol (LACP) to dynamically configure LAG bundles with IPv6 addresses.

Related Documentation

- Configuring 802.3ad Link Aggregation on page 197
- Example: Configuring 802.3ad Link Aggregation on page 200
- Monitoring a Specified Ethernet Member Link in an IEEE 802.3ad LAG Bundle on page 216
- Monitoring Ethernet Member Links in all IEEE 802.3ad LAG Bundles on page 221
- See *Configuring Layer 2 Services over MPLS* in the *JunosE BGP and MPLS Configuration Guide*

- See *Appendix A, Module Protocol Support* in the *ERX Module Guide*
- See *Appendix A, Module Protocol Support* in the *E120 and E320 Module Guide*

802.3ad Link Aggregation Platform Considerations

You can configure 802.3ad link aggregation on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support 802.3ad link aggregation on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support 802.3ad link aggregation.

For information about the modules that support 802.3ad link aggregation on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support 802.3ad link aggregation.

Interface Specifiers

The configuration task examples in this chapter use the format for ERX7xx models, ERX14xx models, and the ERX310 router to specify 802.3ad link aggregation.

For example, the following command specifies a Gigabit Ethernet interface on port 0 of an I/O module in slot 4.

host1(config)#**interface gigabitEthernet 4/0**

When you configure a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface on E120 or E320 routers, you must include the adapter identifier as part of the interface specifier. For example, the following command specifies a Gigabit Ethernet interface on port 0 of the IOA installed in the upper adapter bay of slot 3.

```
host1(config)#interface gigabitEthernet 3/0/0
```

For more information about interface types and specifiers on E Series models, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Related Documentation

- Understanding IEEE 802.3ad Link Aggregation on page 193
- Configuring 802.3ad Link Aggregation on page 197

802.3ad Link Aggregation References

For more information about 802.3ad link aggregation implementations, consult the following resources:

- IEEE 802.1w (Rapid Reconfiguration of Spanning Tree)
- IEEE 802.3ad (Link Aggregation)

Related Documentation

- Understanding IEEE 802.3ad Link Aggregation on page 193
- Configuring 802.3ad Link Aggregation on page 197

Configuring 802.3ad Link Aggregation

To configure link aggregation on Ethernet interfaces, you must configure the Ethernet interface, create the LAG bundle, and add the Ethernet interface as a member link in the LAG bundle. Optionally, you can then configure IP, a VLAN subinterface, a PPPoE subinterface, or MPLS for the LAG bundle.

For more information about specifying LAG interfaces and subinterfaces on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Tasks to configure 802.3ad link aggregation interfaces are:

1. Configuring an Ethernet Physical Interface on page 197
2. Configuring a LAG Bundle on page 198
3. Configuring IP for a LAG Bundle on page 198
4. Configuring IPv6 for a LAG Bundle on page 198
5. Configuring a VLAN Subinterface for a LAG Bundle on page 199
6. Configuring a PPPoE Subinterface for a LAG Bundle on page 199
7. Configuring MPLS for a LAG Bundle on page 200

Configuring an Ethernet Physical Interface

To configure a member link, perform the following steps:

1. Specify a Fast Ethernet or Gigabit Ethernet interface for which you want to create a member link.

   host1(config)#**interface gigabitEthernet 2/0**

2. Configure LACP in passive or active mode.

   host1(config-if)#**lacp active**

3. Specify the speed and the duplex mode for the Ethernet interface.

   host1(config-if)#**speed 100**
   host1(config-if)#**duplex full**

4. Specify the MTU.

   host1(config-if)#**mtu 9000**

5. To configure additional member links, repeat steps 1 through 4.

NOTE: All of the member links that you configure must be on the same line module and have the same physical layer characteristics, such as speed, duplex mode, and MTU.

Configuring a LAG Bundle

To configure a LAG bundle and add member links, perform the following steps:

1. Create the LAG bundle.

   host1(config)#**interface lag bundleBoston**

2. Add a member link to the LAG bundle.

   host1(config-if)#**member–interface gigabitEthernet 2/0**

3. (Optional) Configure the minimum number of member links required in the LAG bundle for the LAG interface to be considered up.

   host1(config-if)#**minimum-links 2**

Configuring IP for a LAG Bundle

To configure IP for a LAG bundle, perform the following steps:

1. Specify the LAG bundle.

   host1(config)#**interface lag bundleBoston**

2. Assign an IP address and mask.

   host1(config-if)#**ip address 192.5.127.8 255.255.255.0**

Configuring IPv6 for a LAG Bundle

To configure an IPv6 address for a LAG bundle, perform the following steps:

1. Specify the LAG bundle.

host1(config)#**interface lag bundleBoston**

2. Assign an IPv6 prefix address to the LAG bundle.

   host1(config-if)#**ipv6 address 1::1/64**

Configuring a VLAN Subinterface for a LAG Bundle

To configure a VLAN subinterface for the LAG bundle, perform the following steps:

1. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

2. Specify the VLAN subinterface for the LAG bundle by adding a unique subinterface number to the LAG interface identification command.

   host1(config)#**interface lag bundleBoston.1**

3. Assign a VLAN ID for the subinterface.

   host1(config-if)#**vlan id 203**

4. Assign an IP address and mask.

   host1(config-if)#**ip address 192.168.1.1 255.255.0.0**

5. Assign an IPv6 prefix address.

   host1(config-if)#**ipv6 address 1::1/64**

Configuring a PPPoE Subinterface for a LAG Bundle

To configure a PPPoE subinterface for the LAG bundle, perform the following steps:

1. Specify PPPoE as the encapsulation method.

   host1(config-if)#**encapsulation pppoe**

2. Specify the PPPoE subinterface for the LAG bundle in either of the following ways:

   - Use the **interface lag** command to add a unique subinterface number to the LAG bundle name.

     host1(config)#**interface lag bundleBoston.2**

   - Use the **pppoe subinterface lag** command to add a unique subinterface number to the LAG bundle name.

     host1(config)#**pppoe subinterface lag bundleBoston.2**

3. Specify PPP as the encapsulation method on the PPPoE subinterface.

   host1(config-if)#**encapsulation ppp**

4. Assign an IP address and mask.

   host1(config-if)#**ip address 192.168.1.2 255.255.0.0**

5. Assign an IPv6 prefix address.

   host1(config-if)#**ipv6 address 1::1/64**

You can also configure a PPPoE subinterface over a VLAN subinterface over a LAG bundle. For an example of this configuration, see "Example: Configuring 802.3ad Link Aggregation" on page 200.

Configuring MPLS for a LAG Bundle

To configure MPLS for a LAG bundle, perform the following steps:

1. Specify the LAG bundle.

   host1(config)#**interface lag bundleBoston**

2. Create an MPLS interface.

   host1(config-if)#**mpls**

Related Documentation

- Understanding IEEE 802.3ad Link Aggregation on page 193
- Monitoring a Specified Ethernet Member Link in an IEEE 802.3ad LAG Bundle on page 216
- Monitoring Ethernet Member Links in all IEEE 802.3ad LAG Bundles on page 221

- ***encapsulation***
- ***interface fastEthernet***
- ***interface gigabitEthernet***
- ***interface lag***
- ***lacp***
- ***lacp port-priority***
- ***member-interface-type***
- ***minimum-links***
- ***mpls***
- ***mtu***
- ***pppoe subinterface***
- ***virtual-router***
- ***vlan description***
- ***vlan id***

Example: Configuring 802.3ad Link Aggregation

The examples in this topic illustrate how to configure 802.3ad Link Aggregation.

- Requirements on page 201
- Overview of Configuring 802.3ad Link Aggregation on page 201
- Example: Configuring an IP Interface for a LAG Bundle on page 201
- Example: Configuring a PPPoE Subinterface for a LAG Bundle on page 202

- Example: Configuring a PPPoE Subinterface over a VLAN for a LAG Bundle on page 203
- Example: Configuring MPLS for a LAG Bundle on page 204
- Example: Configuring MPLS over a VLAN for a LAG Bundle on page 205

Requirements

This example uses the following software and hardware components:

- JunosE Release 7.1.0 or higher-numbered releases
- E Series router (ERX7xx models, ERX14xx models, the ERX310 router, the E120 router, or the E320 router)
- ASIC-based line modules that support Fast Ethernet or Gigabit Ethernet

Overview of Configuring 802.3ad Link Aggregation

IEEE 802.3ad link aggregation enables you to group Ethernet interfaces at the physical layer to form a single link layer interface, also known as a link aggregation group (LAG) or bundle. For more information, see IEEE Standard 802.3ad, Link Aggregation.

To configure link aggregation on Ethernet interfaces, you must configure the Ethernet interface, create the LAG bundle, and add the Ethernet interface as a member link in the LAG bundle. Optionally, you can then configure IP, a VLAN subinterface, a PPPoE subinterface, or MPLS for the LAG bundle.

NOTE: After you configure member interfaces in a LAG bundle, if you attempt to modify the physical layer characteristics, such as speed, duplex mode, and MTU, for the interfaces part of the LAG bundle, an error message is displayed stating that you cannot edit the values for these settings because the interfaces are contained in the LAG bundle.

Example: Configuring an IP Interface for a LAG Bundle

Step-by-Step Procedure

To configure an IP Interface for a LAG bundle:

1. Configure a Fast Ethernet interface in slot 0.

   host1(config)#**interface fastEthernet 0/0**
   host1(config-if)#**speed 100**
   host1(config-if)#**duplex full**
   host1(config-if)#**lacp active**

   NOTE: In a LAG group, member interfaces can be added only from a single slot. For ERX platforms, member interfaces can be added from both of the I/O adapters from a slot.

2. Similarly, configure another Fast Ethernet interface in slot 0.

```
host1(config)#interface fastEthernet 0/5
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp active
```

NOTE: The interfaces are enabled for active LACP. The speed and duplex characteristics are the same for both the interfaces.

3. Create a virtual router.

```
host1(config-if)#virtual-router boston
```

NOTE: LAG is not specific to any virtual router. A LAG interface is visible and can be used in any virtual router.

4. Create a LAG bundle named bundleBoston and add the Ethernet physical interfaces to it.

```
host1:boston(config)#interface lag boston
host1:boston(config-if)#member-interface fastEthernet 0/0
host1:boston(config-if)#member-interface fastEthernet 0/5
```

5. Assign an IP address and mask.

```
host1:boston(config-if)#ip address 1.1.1.1 255.255.255.0
```

6. Assign an IPv6 address.

```
host1:boston(config-if)#ipv6 address 1::1/64
```

Example: Configuring a PPPoE Subinterface for a LAG Bundle

Step-by-Step Procedure

To configure a PPPoE Subinterface for a LAG bundle:

1. Configure a Fast Ethernet interface in slot 4.

```
host1(config)#interface fastEthernet 4/0
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp passive
```

2. Similarly, configure another Fast Ethernet interface in slot 4.

```
host1(config)#interface fastEthernet 4/3
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp passive
```

NOTE: The interfaces are enabled for passive LACP. The speed and duplex characteristics are the same for both the interfaces.

3. Add the Ethernet physical interfaces to a LAG bundle named chicago.

   host1(config)#**interface lag chicago**
   host1(config-if)#**member-interface fastEthernet 4/0**
   host1(config-if)#**member-interface fastEthernet 4/3**

4. Specify PPPoE as the encapsulation method.

   host1(config-if)#**encapsulation pppoe**

5. Add a unique subinterface number to the LAG bundle name. In the LAG interface identification command (**interface lag chicago.1**), the number 1 represents the subinterface number for the PPPoE subinterface.

   host1(config)#**interface lag chicago.1**

   NOTE: As an alternative to using the command **interface lag chicago.1** to configure the PPPoE subinterface in this example, you can also use the command **pppoe subinterface lag chicago.1** to achieve the same result.

6. Specify PPP as the encapsulation method on the PPPoE subinterface.

   host1(config-if)#**encapsulation ppp**

7. Assign an IP address and mask.

   host1(config-if)#**ip address 10.10.1.1 255.255.0.0**

8. Assign an IPv6 prefix address.

   host1(config-if)#**ipv6 address 1::1/64**

Example: Configuring a PPPoE Subinterface over a VLAN for a LAG Bundle

Step-by-Step Procedure

To configure a PPPoE subinterface over a VLAN for a LAG bundle:

1. Configure a Fast Ethernet interface in slot 3.

   host1(config)#**interface fastEthernet 3/0**
   host1(config-if)#**speed 100**
   host1(config-if)#**duplex full**
   host1(config-if)#**lacp active**

2. Similarly, configure another Fast Ethernet interface in slot 3.

   host1(config)#**interface fastEthernet 3/1**
   host1(config-if)#**speed 100**
   host1(config-if)#**duplex full**
   host1(config-if)#**lacp active**

   NOTE: The interfaces are enabled for active LACP. The speed and duplex characteristics are the same for both the interfaces.

3. Add the Ethernet physical interfaces to a LAG bundle named sunnyvale.

   host1(config)#**interface lag sunnyvale**
   host1(config-if)#**member-interface fastEthernet 3/0**
   host1(config-if)#**member-interface fastEthernet 3/1**

4. Configure a VLAN subinterface for the LAG bundle named sunnyvale. In the LAG interface identification command (**interface lag sunnyvale.1**), the number 1 represents the subinterface number for the VLAN subinterface.

   host1(config-if)#**encapsulation vlan**
   host1(config)#**interface lag sunnyvale.1**
   host1(config-if)#**vlan id 100**

5. Configure a PPPoE subinterface over the VLAN subinterface for the LAG bundle named sunnyvale. In the LAG interface identification command (**interface lag sunnyvale.1.2**), the number 2 represents the subinterface number for the PPPoE subinterface.

   host1(config-if)#**encapsulation pppoe**
   host1(config)#**interface lag sunnyvale.1.2**

   NOTE: As an alternative to using the command **interface lag sunnyvale.1.2** to configure the PPPoE subinterface in this example, you can also use the command **pppoe subinterface lag sunnyvale.1.2** to achieve the same result.

6. Specify PPP as the encapsulation method on the PPPoE subinterface.

   host1(config-if)#**encapsulation ppp**

7. Assign an IP address and mask.

   host1(config-if)#**ip address 10.10.2.2 255.255.0.0**

8. Assign an IPv6 prefix address.

   host1(config-if)#**ipv6 address 1::1/64**

Example: Configuring MPLS for a LAG Bundle

Step-by-Step Procedure

To configure MPLS for a LAG bundle:

1. Configure a Fast Ethernet interface in slot 5.

   host1(config)#**interface fastEthernet 5/0**
   host1(config-if)#**speed 100**
   host1(config-if)#**duplex full**
   host1(config-if)#**lacp active**

2. Similarly, configure another Fast Ethernet interface in slot 5.

   host1(config)#**interface fastEthernet 5/1**
   host1(config-if)#**speed 100**
   host1(config-if)#**duplex full**
   host1(config-if)#**lacp active**

NOTE: The interfaces are enabled for active LACP. The speed and duplex characteristics are the same for both the interfaces.

3. Create a virtual router.

   host1(config-if)#**virtual-router kanata**

4. Create a LAG bundle named kanata and add the Ethernet physical interfaces to it.

   host1:kanata(config)#**interface lag kanata**
   host1:kanata(config-if)#**member-interface fastEthernet 0/0**
   host1:kanata(config-if)#**member-interface fastEthernet 0/5**

5. Assign an IP address and mask.

   host1:kanata(config-if)#**ip address 1.1.1.1 255.255.255.0**

6. Configure an MPLS interface.

   host1(config-if)#**mpls**

Example: Configuring MPLS over a VLAN for a LAG Bundle

Step-by-Step Procedure

To configure MPLS over a VLAN for a LAG bundle:

1. Configure a Fast Ethernet interface in slot 5.

   host1(config)#**interface fastEthernet 5/0**
   host1(config-if)#**speed 100**
   host1(config-if)#**duplex full**
   host1(config-if)#**lacp active**

2. Similarly, configure another Fast Ethernet interface in slot 5.

   host1(config)#**interface fastEthernet 5/1**
   host1(config-if)#**speed 100**
   host1(config-if)#**duplex full**
   host1(config-if)#**lacp active**

NOTE: The interfaces are enabled for active LACP. The speed and duplex characteristics are the same for both the interfaces.

3. Create a virtual router.

   host1(config)#**virtual-router kanata**

4. Create a LAG bundle named kanata and add the Ethernet physical interfaces to it.

   host1:kanata(config)#**interface lag kanata**
   host1:kanata(config-if)#**member-interface fastEthernet 5/0**
   host1:kanata(config-if)#**member-interface fastEthernet 5/1**

5. Configure a VLAN subinterface for the LAG bundle named kanata. In the LAG interface identification command (**interface lag kanata.1**), the number 1 represents the subinterface number for the VLAN subinterface.

   host1:kanata(config-if)#**encapsulation vlan**
   host1:kanata(config)#**interface lag sunnyvale.1**
   host1:kanata(config-if)#**vlan id 100**

6. Configure an MPLS interface.

   host1(config-if)#**mpls**

Related Documentation

- Understanding IEEE 802.3ad Link Aggregation on page 193
- Configuring 802.3ad Link Aggregation on page 197
- ***encapsulation***
- ***interface fastEthernet***
- ***interface gigabitEthernet***
- ***interface lag***
- ***lacp***
- ***lacp port-priority***
- ***member-interface-type***
- ***minimum-links***
- ***mpls***
- ***mtu***
- ***pppoe subinterface***
- ***virtual-router***
- ***vlan id***

Ethernet Link Redundancy Overview

You can use 802.3ad Link Aggregation (LAG) to configure Ethernet link redundancy for Fast Ethernet and Gigabit Ethernet interfaces. Ethernet link redundancy enables you to protect against physical link failure and account for network topology changes that redirect network traffic to redundant ports.

The following configurations are available:

- LAG to LAG—Provides redundancy capabilities for two or more ports that are assigned to a LAG. One member link is configured as the backup interface for all other ports in the LAG bundle (1:N). Traffic is not forwarded over the backup member interface; it is disabled until it takes over for an active member interface.

- LAG to non-LAG—Provides redundancy capabilities when redundant ports are connected to a bridged network that has Rapid Spanning Tree Protocol (RSTP) controlling the topology. This configuration supports only two links in the LAG.

For information about the modules that support link aggregation, see *ERX Module Guide, Appendix A, Module Protocol Support* and *E120 and E320 Module Guide, Appendix A, IOA Protocol Support*.

Ethernet Link Redundancy Configuration Models

The link connections determine the configuration model for link redundancy. The following connection types are available:

- Single-homed—Connections are between the local Ethernet interface and a single remote device. When the peer is also configured with LAG, LACP can be used to control link access.
- Dual-homed—Connections are between two separate, uncoordinated remote devices. The remote interfaces can be on the same module or on separate hardware. If LAG is not configured on the peers, LACP cannot be used to select ports; other protocols such as RSTP can be used.

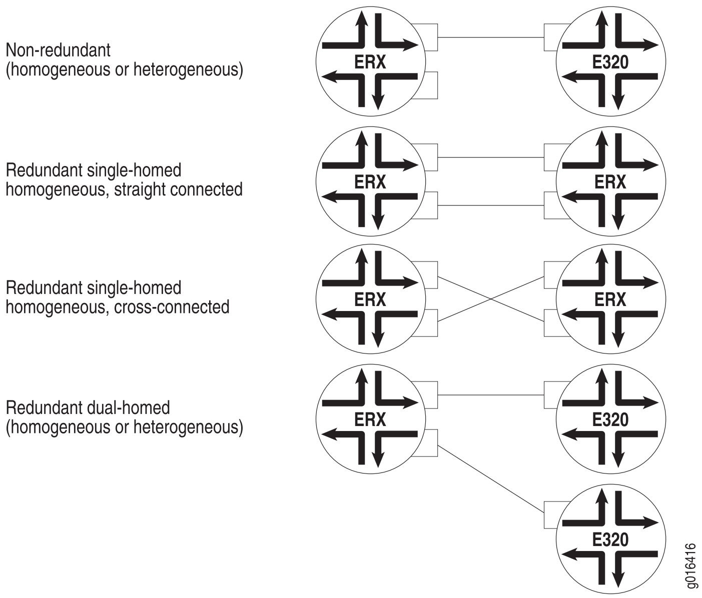
The type of hardware used for connections further characterizes the single-homed and dual-homed configuration models. The following hardware types are available:

- Homogeneous—Remote interface is on another Fast Ethernet or Gigabit Ethernet port in a back-to-back router configuration of identical hardware and JunosE Software versions. Both interfaces support the same redundant cabling and algorithm. The interfaces can be cabled on the same ports (port 0–port 0, port 1–port 1) or cross-cabled (port 0–port 1, port 1–port 0).
- Heterogeneous—Remote interface is on a different type of hardware that might or might not support redundant cabling, or on the same type of equipment with different software versions. For example, a heterogeneous configuration can include an ES2-S1 GE-4 IOA and an ES2-S1 GE-8 IOA on the E320 router, or an E Series router operating JunosE Software connected to another vendor's router and software.

NOTE: **You cannot configure link redundancy across different types of line modules in a router. You also cannot configure link redundancy across two GE-4 IOAs on the E120 or the E320 routers.**

Figure 21 on page 208 illustrates the configuration models for Ethernet link redundancy.

Figure 21: Ethernet Link Redundancy Configuration Models



Ethernet Link Redundancy Configuration Diagrams

The diagrams in this section illustrate examples of Ethernet link redundancy configurations. The diagrams display adjacent ports bundled in a LAG.

GE-2 Line Module Configurations

These diagrams compare physical port redundancy and link redundancy on a GE-2 line module.

Figure 22 on page 208 displays a GE-2 line module with physical port redundancy on both ports.

Figure 22: GE-2 Line Module Using Physical Port Redundancy

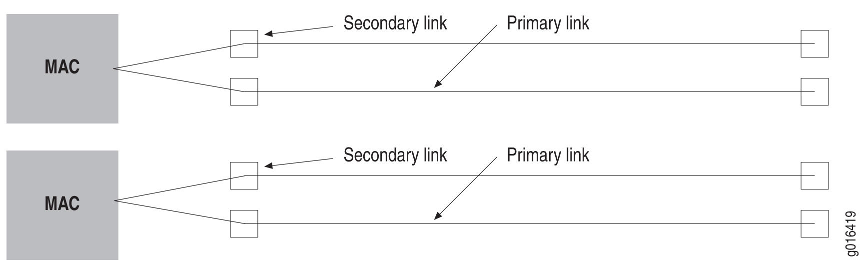
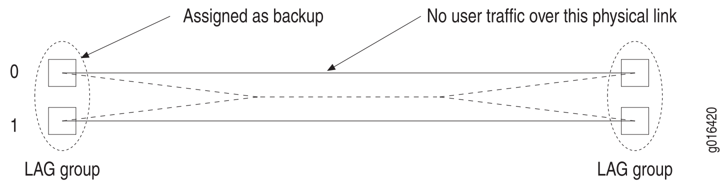


Figure 23 on page 209 displays a single-homed configuration with port 0 backing up port 1 on a GE-2 line module.

Figure 23: Single-Homed GE-2 Line Module Configuration



FE-8 Line Module Configurations

Figure 24 on page 209 displays an FE-8 line module with a link failure in a 1:N single-homed configuration.

Figure 24: Single-Homed FE-8 Line Module Configuration (1:N)

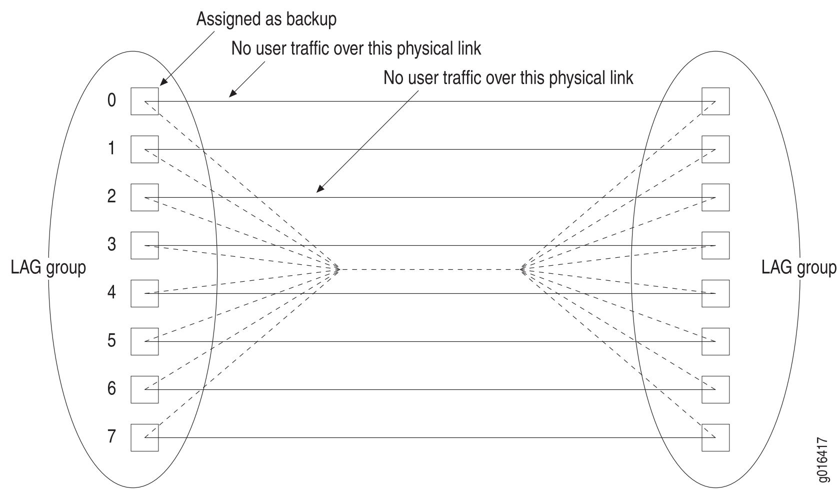
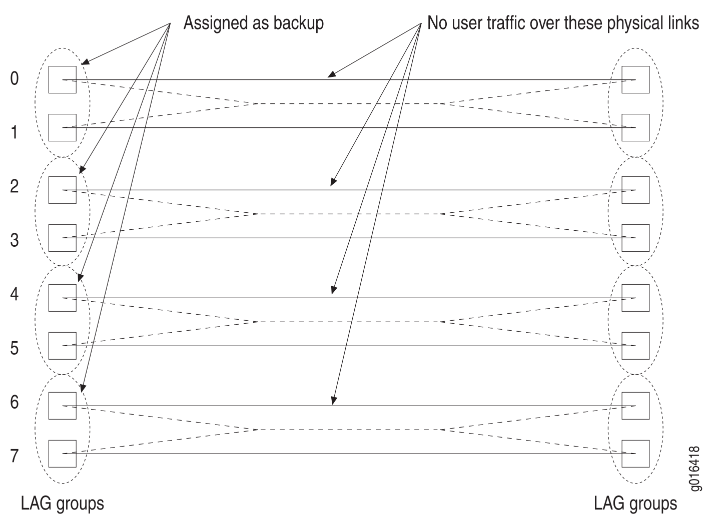


Figure 25 on page 209 displays an FE-8 line module with four redundant Ethernet links in a 1:1 configuration.

Figure 25: FE-8 Line Module with 4 Redundant Ethernet Links (1:1)



E120 and E320 Routers Configurations

Figure 26 on page 210 and Figure 27 on page 210 display link redundancy configurations on the E120 and E320 routers.

Figure 26 on page 210 displays a single-homed 1:4 configuration on an E120 router.

Figure 26: Single-Homed GE-4 IOA Configuration (1:4)

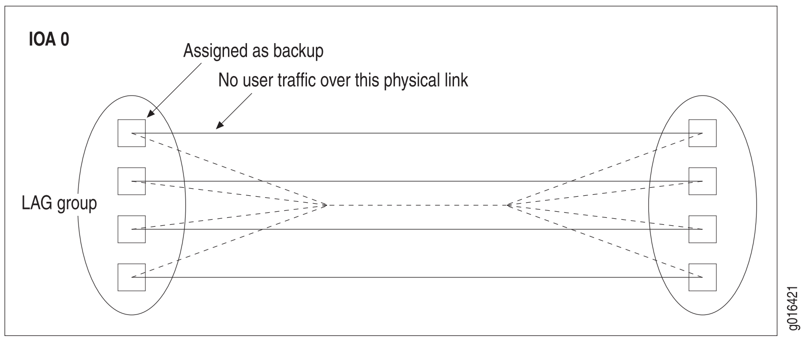
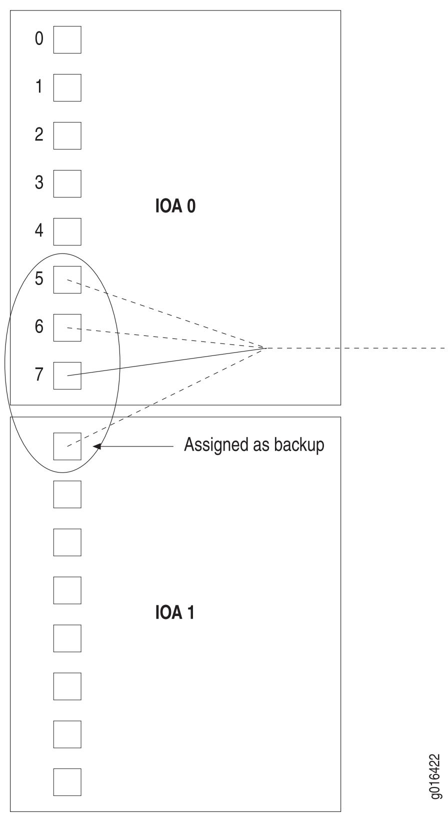


Figure 27 on page 210 displays an E320 router with 1:N configuration across IOAs.

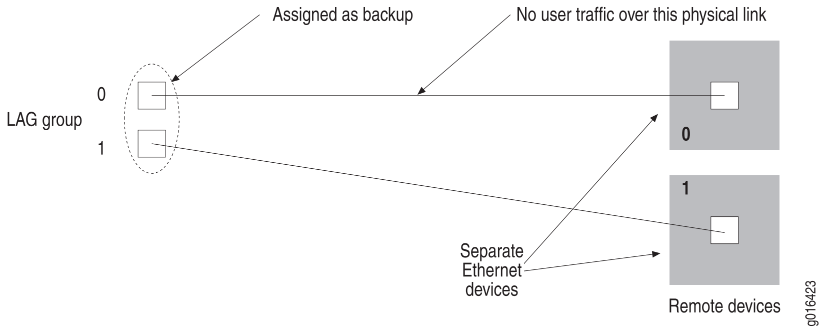
Figure 27: GE-8 IOA Configuration Across IOAs (1:N)



Dual-Homed Configurations with LAG Disabled

Figure 28 on page 211 displays how you can configure Ethernet link redundancy with LACP disabled locally using a dual-homed configuration. LACP is disabled because there is no LAG at the peer.

Figure 28: Dual-Homed Configuration (1:1)



Related Documentation

- Ethernet Link Redundancy Behavior Overview on page 211
- Configuring Ethernet Link Redundancy on page 215

Ethernet Link Redundancy Behavior Overview

When you create a LAG bundle, you can configure LACP with the Disabled, Passive, or Active states. For more information about these states, see “LACP” on page 194.

The following sections describe link redundancy behavior in various scenarios:

- Link Failure and Acquisition on page 211
- LACP Configuration and Member Link Behavior on page 213
- Member Link with Non-LAG Partner on page 213

Link Failure and Acquisition

Link failure on the local system occurs when the active link is no longer active. Failures can be characterized as physical link failure or virtual link failure.

Each type of link failure has different requirements for detection, failover, and link acquisition. In all cases, you configure the link to fail over when it fails by issuing the **redundant-port** command . Optionally, you can force the failover automatically by issuing the **redundant-port force-failover** command

Protecting Against Physical Link Failure

Physical link failures can occur when a cable is cut.

To protect against physical link failure, issue the **transmitter** keyword with the **redundant-port** command to enable or disable the local redundant link. When the redundant link needs to be down, the link behavior in failure detection and failover follows a similar port redundancy scheme available with line modules such as the GE-2 line module. Disabling the transmitter also enables the remote end of the redundant link to be in the operational Down state, which might be a requirement for third-party equipment when supporting redundancy over LAG.

Enabling the transmitter provides for a quick LAG failover in the event one of the non-redundant links in the LAG fail. This is particularly true when LACP has been enabled on the LAG, because it can take several seconds for LACP to converge on a link. When the transmitter on the remote end is enabled on the redundant link before it fails over, the local system considers the redundant link to be viable and enables the transmitter if it is disabled. If the remote end is disabled, the local end must enable the transmitter and wait for the remote end to enable.

Protecting Against Virtual Link Failure

A virtual link failure can occur when the active link is no longer used by the network because of topology changes caused by physical failure in the network. Topology changes can occur when, for example, a link is blocked because of network protocols such as RSTP blocking the port leading to selection of the redundant port connected to the receiver.

To protect against virtual link failure in conjunction with network protocols, use the **packet-sampling** keyword with the **redundant-port** command to detect link the viability. For example, when there is a network protocol decision that changes the topology and blocks a link to compensate for failures in the network, the system monitors the traffic to detect the change in network topology and fails over to the redundant port if necessary. It also determines whether the failover is successful. For more information, see “Member Link with Non-LAG Partner” on page 213.

Reverting After a Failover

When you specify the **auto-revert** keyword with the **redundant-port** command, the redundant link reverts back to redundant mode when the failed link becomes active again.

The system uses the following processes when the auto-revert feature is enabled (by specifying the **auto-revert** keyword) or disabled.

When **auto-revert** is enabled:

1. An active link fails and a redundant link becomes active.
2. The original active link becomes active.
3. The original redundant link fails over to the original active link.
4. The redundant link can fail over to any other active link again.

When **auto-revert** is disabled:

1. An active link fails and a redundant link becomes active.
2. The original active link becomes active.

3. The original redundant link remains the active link.
4. You can force the link to fail over by issuing the **redundant-port force-failover** command.

LACP Configuration and Member Link Behavior

By default, when a redundant member link is configured, the system disables LACP and the transmitter on that link.

When a member link is administratively down, the link state is operationally down at the local and remote ends, which means it does not transmit or receive PDUs.

The active link does not fail over when:

- An active link goes down and you set the redundant member link to administratively down.
- An active link is set to administratively down.

LACP configurations affect member link behavior based on the local or remote endpoint. For a remote end to include a member link in link aggregation, the two member links that are connected must have LACP configured.

Table 14 on page 213 lists the acceptable configurations that enable redundant behavior for LACP modes at local and remote endpoints.

Table 14: Behavior of Member Links Using Local and Remote LACP Modes

	Remote LACP Mode			
		Disabled	Passive	Active
Local LACP Mode	Disabled	✓	✓	–
	Passive	✓	✓	✓
	Active	–	✓	✓

Member Link with Non-LAG Partner

When a member link has a non-LAG partner, there are two separate links in a 1:1 configuration. To successfully configure this, you must disable LACP.

When a failover occurs and LACP is active, the partner might receive a new LAG ID and the LACP PDUs receive a new MAC address; therefore, the member links are not aggregated or the bundle is disabled, terminating the sessions above it.

The partner that is connected to the redundant link must not be forwarding network traffic; that is, it is either blocked through a protocol such as RSTP, or MAC address learning has selected the active port. The redundant link must not transmit over the redundant link to that MAC. The behavior of the redundant link depends on the failure detection method that is controlled by the network protocol that is blocking the port.

Ethernet Link Redundancy and RSTP

In a LAG to non-LAG configuration, you can configure redundancy capabilities when redundant ports are connected to a bridged network that has RSTP controlling the topology.

On external devices, we recommend that you configure RSTP-enabled bridged ports that are connected to the LAG interfaces as edge ports to enable the ports to transition quickly to forwarding state upon reconfiguration, and to avoid the listening and learning states required by the spanning tree protocol. The edge port designation instructs the local bridge that bridge loops do not exist through the interface, enabling it to skip the listening and learning states.

Figure 29: Dual-Homed Heterogeneous Configuration in an RSTP Network

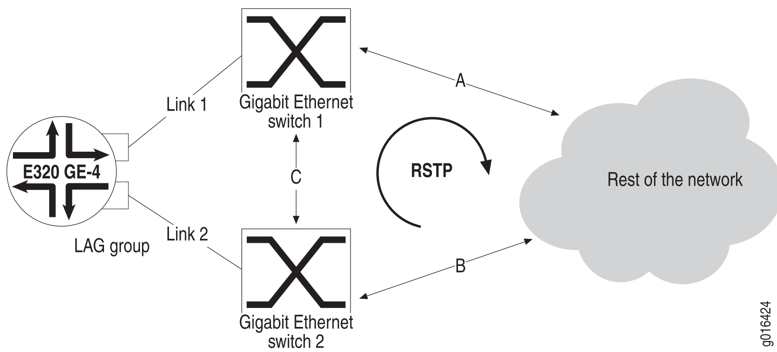


Figure 29 on page 214 displays a network with RSTP enabled on Gigabit Ethernet switches 1 and 2. The local port receives bridge PDUs (BPDU), Ethernet broadcasts, and flooded unicast packets. If Link 1 is initially active and Link 2 is the backup, initial traffic destined for the LAG can be Ethernet broadcasts, PPPoE PDUs, or flooded Ethernet unicasts. The responses are only sent on the active link; in this case, Link 1.

The Ethernet network topology that is managed by RSTP learns that the MAC for the LAG group is through Link 1. Broadcasts and flooded packets are still sent on Link 2. If Link 1 is no longer viable, but has not suffered a physical failure, then that address ages out of the bridge databases and any packets directed to the LAG are flooded. The LAG detects traffic on Link 2 after the minimum delay time and then fails over.

Acquiring Initial Links

In an RSTP network, the system uses the following process for acquiring new links:

1. Based on the configuration, the system selects a link as active and the other as redundant.
2. The spanning tree converges on a topology.
3. When convergence occurs and the status of the spanning tree ports change to forwarding, network traffic appears on the links.
4. The local port detects the traffic and confirms the active member as active and the other as the redundant port. Because the initial traffic is broadcast or flooded, both

ports receive the packets. However, because of the timing difference, the selected active port remains active.

Detecting Failures

In an RSTP network, the system uses the following process for detecting when the link has switched over due to topology changes:

1. BPDUs are ignored on the redundant port and system time is not retrieved. Because MAC learning forces non-flooded unicast packets to the active link, traffic to the redundant link does not receive non-flooded packets. The most recent system time is always retrieved when a network packet is received.
2. When the network cannot reach the active link because of topology changes, traffic appears on the redundant link. The redundant port detects the traffic and captures the latest timestamp. When the difference between the timestamp of the first non-bridged PDU and the time the last packet that was received on the active port is sufficiently large to account for the minimum spanning tree convergence time and latency for flooded and broadcast packets, then the port fails over.

Failing Over

In an RSTP network, the system uses the following process to fail over:

1. When the link has failed over, the system monitors the previously active port.
2. When a network packet is received on the redundant port, the system retrieves the timestamp. If the difference in timestamps between that one and the most recent on the current active port is more than the configured failover delay time, then the link fails over. If the difference is less than the delay time, the system ignores it but counts the event. If many of these transitions occur in a time period, then the system administratively brings the ports down. If no network traffic is received on either port, then failover does not occur.

Related Documentation

- Ethernet Link Redundancy Overview on page 206
- Configuring Ethernet Link Redundancy on page 215

Configuring Ethernet Link Redundancy

To configure Ethernet link redundancy:

1. Specify the Fast Ethernet or Gigabit Ethernet interface on which to configure a redundant link.

   host1(config)#**interface gigabitEthernet 1/1**

2. For LAG to non-LAG configurations only, specify that LACP is disabled on the port.

   host1(config-if)#**no lacp**

3. Configure a backup interface and disable LACP on it.

   host1(config)#**interface gigabitEthernet 1/0** host1(config-if)#**no lacp**

4. Configure a LAG interface and assign a member link to the backup interface.

   host1(config)#**interface lag myBundle**
   host1(config-if)#**member-interface gigabitEthernet 1/0**
   host1(config-if)#**member-interface gigabitEthernet 1/1**

5. Do one of the following:

   - Configure link redundancy on the port you specified in step 1.

     host1(config-if)#**redundant-port gigabitEthernet 1/1**

   - Force the port you specified in step 1 to fail over.

     host1(config-if)#**redundant-port gigabitEthernet 1/1 force-failover**

6. (Optional) Configure the redundant link to revert back to redundant mode when the failed link becomes active again.

   host1(config-if)#**redundant-port gigabitEthernet 1/1 auto-revert**

NOTE: In JunosE Release 12.1.x and lower-numbered releases, if you shut down the member interface in a LAG bundle and clear the ARP entry associated with that member link, ARP does not work correctly when the active link encounters a fault and fails over to the redundant link in the LAG bundle. This problem occurs because the forwarding controller incorrectly sends an exception for the channel ID to the interface controller for member interfaces in the LAG bundle configured with the **redundant-port** command that fail over to other links in the LAG bundle.

Beginning with JunosE Release 12.2.x, ARP works correctly for member links in the LAG bundle that are configured as redundant interfaces when they encounter a failure and fails over to other links in the LAG bundle.

Related Documentation

- Ethernet Link Redundancy Overview on page 206
- Ethernet Link Redundancy Behavior Overview on page 211

- ***redundant-port***

Monitoring a Specified Ethernet Member Link in an IEEE 802.3ad LAG Bundle

Purpose Display information about a specified Ethernet member link in an IEEE 802.3ad link aggregation group (LAG) bundle.

Specify either the Fast Ethernet or Gigabit Ethernet interface type when issuing this command:

host1#**show interfaces** ***interfaceType interfaceSpecifier*** **lag**

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

Action To display information about a specified Ethernet member link in an IEEE 802.3ad LAG bundle:

```
host1#show interfaces fastEthernet 4/0 lag
FastEthernet4/0 is Up, Administrative status is Up
 Member of Lag boston
 LACP passive, mux state collecting/distributing
 LACP state (0x3c) passive, long timeout, aggregatable, in-sync, collecting,
distributing
 port 0 priority 32768 key 8
 System Priority 32768 System MAC Address is 0090.1a40.2043
 Partner: state (0x3d) active, short timeout, aggregatable, in-sync,
collecting, distributing
  port 0 priority 32768 key 8 age 25
  System Priority 32768 System MAC Address is 0090.1a40.2043

 LACP packets: received 8, transmitted 7
 Marker Protocol request packets: received 0, transmitted 0
 Marker Protocol response packets: received 0, transmitted 0
 Discarded 0, unknown protocol received 0
```

Meaning Table 15 on page 217 lists the **show interfaces lag** command output fields.

Table 15: show interfaces lag Output Fields

Field Name	Field Description
interfaceSpecifier	Status of the hardware on this interface: • Up—Hardware is operational • Down—Hardware is not operational
Administrative status	Operational state that you configured for this interface
Member	Membership status of the Ethernet link
LACP	Status of LACP configuration for the Ethernet link: • active—Ethernet link participates in the protocol regardless of whether its Partner member link is set to active or passive LACP PDU participation • passive—Ethernet link transmits LACP PDUs only when it receives LACP PDUs from its Partner member link

Table 15: show interfaces lag Output Fields *(continued)*

Field Name	Field Description
mux state	Status of collecting and distributing at the Mux state machine: • collecting/distributing—Ethernet link is actively collecting incoming frames and distributing outgoing frames • detached—Ethernet link is detached from the LAG bundle due to protocol changes or system constraints • waiting—Ethernet link is waiting to attach to a LAG bundle

Table 15: show interfaces lag Output Fields *(continued)*

Field Name	Field Description
LACP state	State of the LACP: • active—Actor link actively particulates in LACP • passive—Actor link transmits LACP PDUs • timeout—Timeout control value; this value is not configurable and is set to long timeout (30 seconds) • aggregatable—Actor link can be aggregated • individual—Actor link cannot be aggregated; must operate as an individual link • in-sync—Actor link has joined the correct LAG bundle • out-of-sync—Actor link is unable to join the correct LAG bundle • collecting—Actor link is actively collecting incoming frames; if this field does not appear, the Actor link is not actively collecting incoming frames • distributing—Actor link is actively distributing outgoing frames; if this field does not appear, the Actor link is not actively distributing outgoing frames • defaulted—Actor link is using defaulted operational information about the Partner link that was administratively configured for Partner; if this field does not appear, the operational information about the Partner link has been received by the Actor link in an LACP PDU • expired—Actor link's receive machine is expired; if this field does not appear, the Actor link's receive machine is active • port—Port number assigned to the Ethernet link by the Actor link • priority—Priority assigned to this Ethernet link by the Actor link • Key—Operational key value assigned to the Ethernet link by the Actor link • System Priority—Priority assigned to the Ethernet link by the system • System MAC Address—MAC address assigned to the Actor link

Table 15: show interfaces lag Output Fields *(continued)*

Field Name	Field Description
Partner	Status of the Partner link • active— Partner link participates in the LACP • passive—Partner link transmits LACP PDUs • timeout—Timeout control value; short timeout or long timeout • aggregatable—Partner link can be aggregated • individual—Partner link cannot be aggregated • in-sync—Partner link has joined the correct LAG bundle • out-of-sync—Partner link has joined the incorrect LAG bundle • collecting—Partner link is actively collecting incoming frames; if this field does not appear, the Partner link is not actively collecting incoming frames • distributing—Partner link is actively distributing outgoing frames; if this field does not appear, the Partner link is not actively distributing outgoing frames • defaulted—Partner link is using defaulted operational information about the Partner link that was administratively configured for Partner; if this field does not appear, the operational information about the Partner link has been received by the Actor link in an LACP PDU • expired—Partner link's receive machine is expired; if this field does not appear, the Partner link's receive machine is active • port—Port number assigned to the Ethernet link by the Actor link • priority—Priority assigned to this Ethernet link by the Actor link • Key—Operational key value assigned to the Ethernet link by the Actor link • System Priority—Priority assigned to the Ethernet link by the system • System MAC Address—MAC address assigned to the Partner link by the system
LACP packets	Number of transmitted and received LACP packets
Marker Protocol request packets	Number of Marker Protocol packets requested to verify transmissions
Marker Protocol response packets	Number of Marker Protocol response packets that verified transmissions
Discarded	Number of invalid LACP packets

Related Documentation

- Understanding IEEE 802.3ad Link Aggregation on page 193
- Configuring 802.3ad Link Aggregation on page 197
- Monitoring Ethernet Member Links in all IEEE 802.3ad LAG Bundles on page 221

- ***show interfaces lag***
- ***show interfaces lag members***

Monitoring Ethernet Member Links in all IEEE 802.3ad LAG Bundles

Purpose Display information about the Ethernet member links in all IEEE 802.3ad link aggregation group (LAG) bundles configured on the router, or about the member links in a specified IEEE 802.3ad LAG bundle.

Specify either the Fast Ethernet or Gigabit Ethernet interface type when issuing this command:

host1#**show interfaces** *interfaceType interfaceSpecifier* **lag members**

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

Action To display information about the Ethernet member link in an IEEE 802.3ad LAG bundle:

```
host1#show interfaces lag members
Lag lag1 is Up, Administrative status is Up
 MAC Address is 0090.1a42.7568
 MTU: Operational 1518
 Duplex Mode: Operational Full Duplex
 Speed: Operational 4000 Mbps
 System Priority 32768 System MAC Address is 0090.1a42.6273 key 9
 Partner System Priority 32768 System MAC Address is 0090.1a42.6273 key 21

 Configured Redundant Port: GigabitEthernet14/0/3
  Failover Timeout: 1000 ms Transmitter off Auto revert enabled
  Active Port: GigabitEthernet14/0/3, Failed Port: GigabitEthernet14/0/2
  Last failover at TUE AUG 26 2008 03:17:49.230 UTC Failover count 3
  Last Failed Port: GigabitEthernet14/0/3

   Member-interface GigabitEthernet14/0/0 is Up
      (LACP active, state collecting/distributing)
   Member-interface GigabitEthernet14/0/1 is Up
      (LACP active, state collecting/distributing)
   Member-interface GigabitEthernet14/0/2 is Down
      (LACP active, state detached)
   Member-interface GigabitEthernet14/0/3 is Up
      (LACP active, state collecting/distributing)
```

Meaning Table 16 on page 222 lists the **show interfaces lag member** command output fields.

Table 16: show interfaces lag member Output Fields

Field Name	Field Description
Lag	Name of the LAG bundle
Administrative status	Operational state that you configured for this interface
MAC Address	MAC address of the processor on this interface
System Priority	Priority assigned to the Ethernet link by the system
MTU	Status of LACP configuration for the Ethernet link: • Operational—MTU size of the current packet being processed • Administrative—Setting for MTU size that you specified
Duplex Mode	Duplex option for this interface: • Operational—Duplex option currently used • Administrative—Setting for duplex that you specified
Speed	Line speed for this interface: • Operational—Current rate at which packets are processed • Administrative—Setting for line speed that you specified
System MAC Address	MAC address assigned to the Actor link
Partner System Priority	Priority assigned to the Ethernet link by the Partner link's system
System MAC Address	MAC address assigned to the Partner link by the system
Failover Timeout	The amount of time between the current link event leading to failover or reversion and the previous link failover or reversion, in the range 100-10000 milliseconds (ms)
Active Port	Port number of the currently active port in the LAG bundle
Failed Port	Port number of the port that encountered a link failure in the LAG bundle
Last failover at	The time at which the last failover from the primary link in the LAG bundle to the secondary link occurred

Table 16: show interfaces lag member Output Fields *(continued)*

Field Name	Field Description
Last Failed Port	Port number of the last failed interface in the LAG bundle
Member-interface	Status of the member interface in the bundle: • *Interface Specifier*—Status of the hardware on this interface (up or down) • LACP active—Ethernet link participates in the protocol regardless of whether its Partner member link is set to active or passive LACP PDU participation • LACP passive—Ethernet link transmits LACP PDUs only when it receives LACP PDUs from its Partner link • collecting/distributing—Ethernet link is actively collecting incoming frames and distributing outgoing frames • detached—Ethernet link is detached from the LAG bundle due to protocol changes or system constraints • waiting—Ethernet link is waiting to attach to a LAG bundle

Related Documentation

- Understanding IEEE 802.3ad Link Aggregation on page 193
- Configuring 802.3ad Link Aggregation on page 197
- Monitoring a Specified Ethernet Member Link in an IEEE 802.3ad LAG Bundle on page 216
- ***show interfaces lag***
- ***show interfaces lag members***

CHAPTER 8

Configuring IEEE 802.3ah OAM Link-Fault Management

This chapter describes how to configure IEEE 802.3ah Operation, Administration, and Maintenance (OAM) link-fault management on your E Series router, and contains the following sections:

- Ethernet OAM Link-Fault Management Overview on page 226
- Ethernet OAM Link-Fault Management Platform Considerations on page 227
- Ethernet OAM Link-Fault Management References on page 228
- OAM Messages on page 228
- OAM Elements Overview on page 229
- OAM Client on page 230
- OAM Sublayer on page 230
- OAM Feature Overview on page 232
- OAM Discovery Feature on page 232
- OAM Link Monitoring Feature on page 234
- OAM Remote Fault Detection Feature on page 236
- OAM Remote and Local Loopback Feature on page 237
- Interrelationship of OAM Link-Fault Management with Ethernet Subsystems on page 238
- Guidelines for Configuring 802.3ah OAM Link-Fault Management on page 239
- Configuring 802.3ah OAM Link-Fault Management on page 240
- Example: Configuring 802.3ah OAM Link-Fault Management and Enabling Remote Failure Monitoring on an Interface on page 246
- Example: Enabling Remote Loopback Support on the Local Interface on page 247
- Monitoring OAM Link-Fault Management Discovery Settings for an Interface on page 247
- Monitoring OAM Link-Fault Management Statistics for an Interface on page 250
- Monitoring OAM Link-Fault Management Configuration for an Interface on page 252
- Monitoring OAM Link-Fault Management Sessions on All Configured Interfaces on page 255

Ethernet OAM Link-Fault Management Overview

The growth of Ethernet as a large-scale networking technology has propelled the necessity for a new set of Operation, Administration, and Maintenance (OAM) protocols. Service provider networks are large and expansive, with the need for different communication operators to work in a combined way to deliver end-to-end services to enterprise users. With the constant growth in the demands of enterprise end users, the need to enhance the features and reliability of service provider Ethernet networks, especially in the areas of availability and mean time to repair (MTTR), is steadily increasing. Ethernet OAM is an enhancement that caters to tracking and resolving connectivity problems in circuits, thereby improving the competitiveness of the service provider.

As DSL access infrastructure in networks worldwide migrates from ATM to Ethernet-based connections, a requirement has evolved to enable Ethernet systems to offer the same set of capabilities as their ATM counterparts in the fields of scalability, provisioning, security, reliability, and manageability. A large difference exists between the ATM and Ethernet networks in the field of OAM. Currently, a comprehensive set of OAM mechanisms exist for ATM topologies to enable proactive monitoring of network health and troubleshooting of network errors. In the recent past, both the Metro Ethernet Forum (MEF) and the IEEE groups have developed OAM standards for Ethernet at both the MAC (802.3) and High Level Interface (802.1) layers.

Ethernet MAC-layer OAM defined in IEEE Standard 802.3ah describes link-based OAM mechanisms. These mechanisms improve the ability of a connected network element to monitor the health of the link and the peer system. This improved ability enables the connected network element to more quickly, proactively, and decisively react to deteriorating or failing conditions of the link. A primary advantage of 802.3ah OAM is to improve the member-link failover time of 802.3ad link aggregation groups (LAGs) that are supported on all E Series router models.

NOTE: Ethernet running on top of a layer 2 protocol, such as Ethernet over ATM, is not supported in OAM configurations.

The Ethernet OAM link fault management feature on routers running JunosE Software interoperates with Junos platforms that implement 802.3ah, such as M Series and MX Series routers (except M5 and M10 routers). Also, the OAM functionality integrates with physical-level redundancy hardware available on certain IOAs, and with 802.3ad link aggregation and link redundancy policies. The Ethernet OAM link fault management functionality enables internal signaling about OAM link fault mechanisms to other internal entities, such as the Ethernet application or the LAG bundle. The 802.3ah OAM capability enables any failure in the member links of a LAG bundle to be detected and notified. SNMP link up/down traps are generated for link up/down events that are triggered by OAM. OAM PDUs are assigned a higher priority than regular data packets.

Related Documentation

- OAM Feature Overview on page 232
- Ethernet OAM Link-Fault Management Platform Considerations on page 227

- Ethernet OAM Link-Fault Management References on page 228

Ethernet OAM Link-Fault Management Platform Considerations

You can configure 802.3ah link-fault management on the following E Series routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support 802.3ah link-fault management on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support 802.3ah link-fault management.

For information about the modules that support 802.3ah link-fault management on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support 802.3ah link-fault management.

Interface Specifiers

The configuration task examples in this chapter use the format for ERX7xx models, ERX14xx models, and the ERX310 router to specify 802.3ah link-fault management

For example, the following command specifies a Gigabit Ethernet interface on port 0 of an I/O module in slot 4.

host1(config)#**interface gigabitEthernet 4/0**

When you configure a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface on E120 or E320 routers, you must include the adapter identifier as part of the interface specifier. For example, the following command specifies a Gigabit Ethernet interface on port 0 of the IOA installed in the upper adapter bay of slot 3.

host1(config)#**interface gigabitEthernet 3/0/0**

For more information about interface types and specifiers on E Series models, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Related Documentation

- *Interface Types and Specifiers*

Ethernet OAM Link-Fault Management References

For more information about 802.3ah link-fault management implementations, consult the following resources:

- IEEE 802.3ah-2004 (Clause 57, Operations, Administration, and Maintenance [OAM])—Media Access Control Parameters, Physical Layers, and Management Parameters for Subscriber Access Networks
- IEEE 802.3ah-2000—Part 3: Carrier Sense multiple access with collision detection (CSMA/CD) access methods and physical layer specifications

Related Documentation

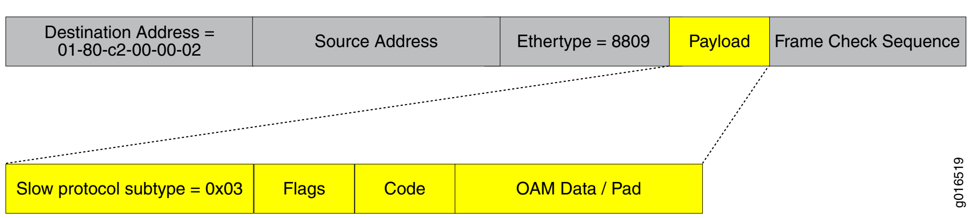
- Ethernet OAM Link-Fault Management Overview on page 226
- Ethernet OAM Link-Fault Management Platform Considerations on page 227

OAM Messages

Because 802.3ah is an optional component and not all functionalities of OAM might be supported on a particular system, discovery mechanisms are used to ascertain the presence and capabilities of the remote peer.

Transmitted Ethernet OAM messages or OAM PDUs are of standard length, untagged Ethernet frames within the normal frame length limits in the range 64–1518 bytes. The maximum OAM PDU frame size exchanged between two peers is determined during the discovery phase. OAM PDUs contain the destination MAC address of the slow protocols multicast (0180.c200.0002) and an Ethertype of 8809. In a slow protocol environment, the bandwidth required is minimal and the frame transmission rate is limited to a maximum of 10 frames per second. The first octet of the frame payload is the slow protocol subtype field and is set to 0x03. OAM PDUs do not travel beyond a single hop and are transmitted at a rate limited to a maximum of 10 OAM messages per second. Certain OAM PDU types might be transmitted multiple times to improve the probability of their successful receipt on degrading, lossy links. Figure 30 on page 228 shows the OAM PDU format.

Figure 30: OAM PDU Format



The Flags field is used to inform the local state to the peer. This state is used in discovery and in remote failure detection. The Code field denotes the type of OAM packet. The format of the OAM Data/Pad field consists of TLV elements.

Four types of OAM messages are supported:

- Information OAM PDU—A variable-length OAM PDU that is used for the discovery process. This OAM PDU contains local, remote, and organization-specific information.
- Event notification OAM PDU—A variable-length OAM PDU that is used for link monitoring. This type of OAM PDU might be transmitted multiple times to improve the probability of a successful receipt, such as in environments that result in high-bit errors. Event notification OAM PDUs also include a timestamp to signify the time at which they are triggered.
- Loopback control OAM PDU—An OAM PDU predefined with a length of 64 bytes to enable or disable the remote loopback command.
- Vendor-specific OAM PDU—A variable-length OAM PDU that enables the addition of vendor-specific extensions to OAM.

Related Documentation

- OAM Feature Overview on page 232
- Configuring 802.3ah OAM Link-Fault Management on page 240

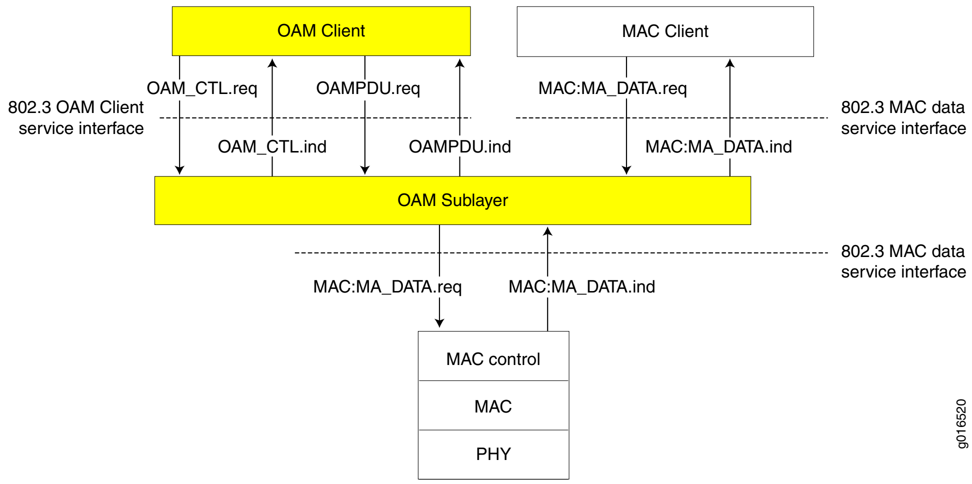
- *ethernet oam lfm*
- *ethernet oam lfm remote-loopback*
- *ethernet oam lfm remote-loopback supported*

OAM Elements Overview

IEEE 802.3ah defines OAM procedures for a single point-to-point Ethernet link. Ethernet OAM is a slow protocol with limited bandwidth requirements. The frame transmission rate is limited to a maximum of 10 frames per second. As a result, the impact of OAM on normal operations is negligible. However, when link monitoring is enabled, the CPU must poll error counters frequently. In this case, the required processor memory and usage are proportional to the number of interfaces that have to be polled.

Two major elements, the OAM client and the OAM sublayer, make up the Ethernet OAM. The OAM sublayer resides above the MAC layer and below the logical link control (LLC) layer. The OAM sublayer presents a MAC data interface to MAC clients and an OAM client interface to OAM clients. Figure 31 on page 230 shows the OAM sublayer interfaces. For effective interoperation and enhanced collaboration with 802.3ad link aggregation, the OAM sublayer exists below the LAG bundle. The LAG bundle is present between the OAM sublayer and the MAC client.

Figure 31: OAM Sublayer Interfaces



The following sections describe the OAM elements:

- OAM Client on page 230
- OAM Sublayer on page 230

Related Documentation

- OAM Feature Overview on page 232
- OAM Messages on page 228

OAM Client

The OAM client establishes and manages Ethernet OAM on a link. The OAM client also enables and configures the OAM sublayer. During the OAM discovery stage, the OAM client monitors OAM PDUs received from the remote peer and enables OAM functionality on the link, depending on the local and remote states, and configuration settings. Outside of the discovery stage, the OAM client manages the rules of response to OAM PDUs and the OAM remote loopback mode.

Related Documentation

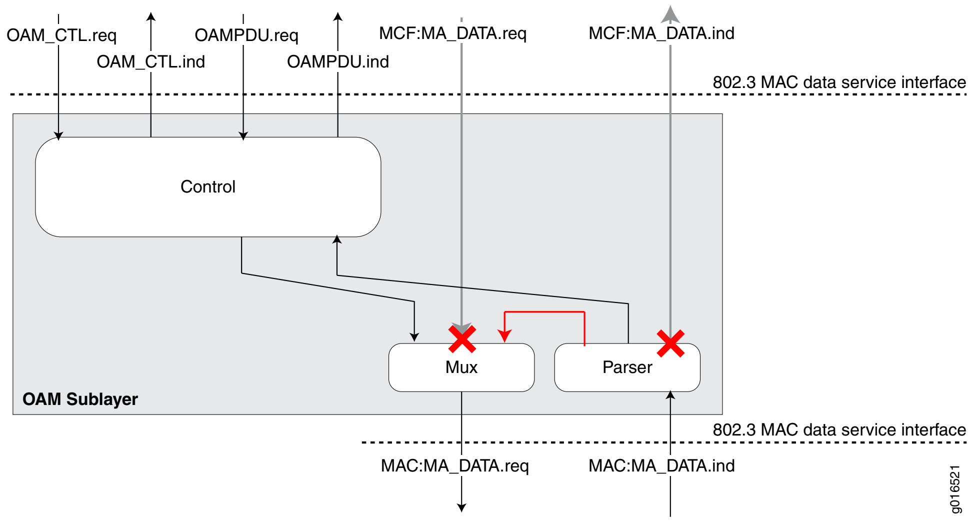
- OAM Sublayer on page 230
- OAM Elements Overview on page 229

OAM Sublayer

The OAM sublayer presents two standard IEEE 802.3 MAC service interfaces: one pointed toward the superior sublayers, which include the MAC client (or link aggregation), and the other interface pointed toward the subordinate MAC control sublayer. The OAM sublayer provides a dedicated interface for transmission of OAM control information and OAM PDUs to and from a client.

The OAM sublayer is made up of three entities: control block, multiplexer, and packet parser. The following sections describe each of the entities. Figure 32 on page 231 shows the entities of the OAM sublayer and the traversal of OAM PDUs among the different entities.

Figure 32: OAM Sublayer Entities



Control Block

The control block functions as the interface between the OAM client and other blocks internal to the OAM sublayer. The control block implements the discovery process, which detects the existence and capabilities of remote OAM peers. It also includes the transmit process that administers the transmission of OAM PDUs to the multiplexer and a set of rules that manage the receipt of OAM PDUs from the packet parser.

Multiplexer

The multiplexer manages frames generated or forwarded from the MAC client, control block, and packet parser. The multiplexer passes through frames generated by the MAC client untouched. It sends OAM PDUs generated by the control block to the subordinate sublayer; for example, the MAC sublayer. The multiplexer also forwards loopback frames from the packet parser to the same subordinate sublayer when the interface is in OAM remote loopback mode.

Parser

The parser categorizes frames as OAM PDUs, MAC client frames, or loopback frames and then transfers each class to the appropriate entity. OAM PDUs are delivered to the control block. MAC client frames are transmitted to the superior sublayer. Loopback frames are distributed to the multiplexer.

Related Documentation

- OAM Client on page 230
- OAM Elements Overview on page 229

OAM Feature Overview

An important behavior of a carrier-class network element is the implementation of OAM capabilities. These capabilities relate to operational fields, such as link monitoring, fault signaling, and remote loopback across multi-vendor equipment, administrative boundaries, and diversified physical locations. OAM capabilities currently exist in ATM and SONET infrastructures. With the continued upgrade to Ethernet-based infrastructures, comparable OAM functions at the MAC (802.3) level are essential. The IEEE 802.3 CSMA/CD Working Group, using its Ethernet in First Mile (802.3ah) task force, defined a set of OAM enhancements for Ethernet links in the 802.3ah standard. These enhancements gracefully ensure backward compatibility with existing Ethernet implementations, while also providing advanced monitoring functionality required in public networks.

JunosE Software supports the following OAM features:

- OAM Discovery Feature on page 232
- OAM Link Monitoring Feature on page 234
- OAM Remote Fault Detection Feature on page 236
- OAM Remote and Local Loopback Feature on page 237

Related Documentation

- Ethernet OAM Link-Fault Management Overview on page 226
- Configuring 802.3ah OAM Link-Fault Management on page 240
- Guidelines for Configuring 802.3ah OAM Link-Fault Management on page 239

OAM Discovery Feature

Discovery is the first phase of Ethernet OAM and it is used to detect the devices in the network and their OAM capabilities. The discovery process is triggered automatically when OAM is enabled on the interface. The discovery phase uses Information OAM PDUs. The discovery process permits Ethernet interfaces to discover and monitor the peer on the link if it also supports the IEEE 802.3ah standard. You can specify the discovery mode used for IEEE 802.3ah OAM support as active or passive.

- In active mode, the interface discovers and monitors the peer on the link if the peer also supports IEEE 802.3ah OAM functionality. In active mode, the peer initiates the discovery process. After the discovery process has been initiated, both sides participate in discovery.
- In passive mode, an OAM entity does not initiate the discovery process. Therefore, the OAM exchange cannot be performed if you configure both the endpoints, the local and the peer entities, for passive mode operation.

The discovery mode that you set up for an OAM entity also determines certain other attributes that can be initiated by an OAM entity. For example, a passive mode OAM entity cannot initiate a variable request or a loopback procedure. In a carrier environment, the customer edge (CE) devices are normally configured for passive mode operation, whereas the provider edge (PE) equipment is configured for active mode operation.

Information OAM PDU Components

An OAM entity in active mode initiates the discovery process by sending an Information OAM PDU to the slow protocols multicast address (destination MAC address of the remote entity) at a configured rate. The transmitted Information OAM PDU contains a local-information type-length-value (TLV). This TLV contains the following fields:

- State—Transmission or receiving state for forwarded packets. The mode can be either active or passive and can be used to determine device functionality.
- Capabilities of the OAM sublayer—Advertises the capabilities of the local OAM entity. With this information a peer can determine what functions are supported and accessible, such as loopback or unidirectional operation. This field also specifies the maximum OAM PDU size for receipt and delivery. This information, together with the rate limiting value of 10 frames per second, can be used to specify the bandwidth allocated to OAM traffic.
- Vendor OUI—Organization unique identifier (OUI), which is controlled by the IEEE and is typically the first three bytes of a MAC address.
- Vendor-specific information—A 32-bit identifier, which is used to distinguish the type of platform in conjunction with the vendor OUI field.

After a local entity sends an Information OAM PDU, the remote OAM entity waits to receive the local information of the peer. After receipt of the Information OAM PDU, the OAM entity applies a policy to determine whether an OAM relationship can be established. For example, loopback mode might be required for the OAM association to be completed. If the remote entity does not support loopback, the local entity might disable the OAM association.

Transmission Settings for Information OAM PDUs

After an OAM association is established, Information OAM PDUs are sent at a configured rate. If no OAM PDUs other than Information OAM PDUs are available to be sent from the local peer to the remote peer, the local entity sends Information OAM PDUs that contain both the local and remote information TLVs. This constant bidirectional transfer of Information OAM PDUs serves as a keepalive mechanism for the OAM association. If no Information OAM PDUs are received within 5 seconds, the discovery process restarts and a link-fault event is generated that might cause a transition in the operational status of the Ethernet interface to the down state. If the OAM association with the peer is reestablished, OAM clears the link-fault event to cause a transition of the interface to the operational up state. The operational status of an interface does not depend only on the OAM status. Other factors, such as the administrative state of the interface, also impact the operational state.

You can configure the OAM discovery function in JunosE Software per Ethernet major interface as either active or passive mode. The OAM state machine labels a port to be in the operational down state until the discovery process is completed successfully. You can configure the PDU timer, which is the rate at which Information OAM PDUs are sent to the remote peer to keep the OAM association active, in the range of milliseconds with a maximum value of 1000 (the default value) and a minimum value of 100. Also, you can configure the local OAM function with a packet loss threshold, which specifies the number of Information OAM PDUs that an interface can miss before the link between peers is considered down. When the PDU loss threshold is exceeded, a link fault event is triggered. The product of the PDU timer and the PDU loss threshold equals the lost-link timer value, which is used to reset the discovery state diagram that maintains the states of the OAM entities and determines the condition of the link based on various stored values.

When the PDU loss threshold is exceeded, the OAM function signals a problem with the link and the link is immediately transitioned to the down state. When the OAM association with the peer is rediscovered after a successful discovery operation, the link transitions to the up state.

NOTE: The operational status displayed in the output of the **show** commands related to interface settings will be down if the OAM session is down owing to loss of association.

Related Documentation

- OAM Feature Overview on page 232
- OAM Messages on page 228
- Configuring 802.3ah OAM Link-Fault Management on page 240
- Monitoring OAM Link-Fault Management Discovery Settings for an Interface on page 247

- *ethernet oam lfm*
- *ethernet oam lfm mode*
- *ethernet oam lfm pdu-lost-threshold*
- *ethernet oam lfm pdu-transmit-interval*
- *show ethernet oam lfm discovery*

OAM Link Monitoring Feature

The router performs link monitoring by sending periodic Information OAM PDUs to advertise OAM mode, configuration, and capabilities. Link monitoring uses the Event Notification OAM PDU and sends events to the remote OAM entity when problems are observed on the link. You can configure the OAM application to track frame and symbol errors on the link to analyze the overall health and quality of the link. Frame errors include frameTooLong, lengthError (runts), alignmentError, and frameCheckSequence errors.

You can monitor frame and symbol errors by setting up a monitoring period or window and a threshold value. If the number of errors observed during the window meets or exceeds the configured low threshold, an Event Notification PDU (in the appropriate TLVs) is generated and sent to the peer. Alternatively, if you configure a high threshold value on the local OAM peer, the OAM function attempts to alter the operational state of the link whenever the high threshold value is exceeded. The monitoring of the link continues with a new window or period as long as the operational state of the link is up. When the number of errors observed during the window equals or goes below the configured low threshold value, the OAM application attempts to reverse the operational state of the link to up.

Supported Error Events for Tracking Link Faults

The OAM application maintains an updated cumulative count of frame and symbol errors and also an updated summation of events generated as a result of a threshold exception. Both these sums are displayed in the appropriate link event TLVs and in the output of the **show ethernet oam** commands.

Because certain MAC devices on the IOAs might not support a symbol error statistic, enabling the monitoring of symbol errors is benign and no events are raised for that link.

The following error events are supported for configuration on Ethernet interfaces:

- Error Symbol Period (error symbols per second)—The number of symbol errors that occurred during a specified period exceeded a threshold. These errors are coding symbol errors.
- Error Frame (error frames per second)—The number of frame errors observed during a specified period exceeded a threshold.
- Error Frame Seconds Summary (error seconds per *n* seconds)—The number of error seconds (1-second intervals with at least one frame error) within the last *n* seconds has exceeded a threshold.

Because IEEE 802.3ah OAM does not provide a guaranteed delivery of any OAM PDU, the event notification OAM PDU might be sent multiple times to reduce the probability of a lost notification. A sequence number is used to distinguish among duplicate events.

Actions Performed on Exceeding Threshold Values

You can configure the OAM application to influence the operational state of the link, when a link quality threshold is exceeded or a critical event PDU is received from the peer, or both. You can configure either of the following actions to be taken when a high threshold value is exceeded or when a failure condition is communicated by the remote peer:

- Disable—OAM unconditionally attempts to influence the operational state of the interface to down. If the interface is a member link of a LAG bundle and at least one other viable link (redundant member or another active/up link) is present, OAM attempts to influence the operational state of the link to down. Otherwise, no action is taken.
- Failover—On GE-2 and GE-HDE line modules that are paired with GE-2 SFP I/O modules with physical link redundancy, this action attempts to transition the link from active to redundant.

By default, no action is performed on the link. The operational status displayed in the output of the **show** commands for interfaces is down if the OAM session is marked as down/nonfunctional after the configured action is taken on the link.

Related Documentation

- OAM Feature Overview on page 232
- Guidelines for Configuring 802.3ah OAM Link-Fault Management on page 239
- OAM Messages on page 228
- Monitoring OAM Link-Fault Management Configuration for an Interface on page 252
- Monitoring OAM Link-Fault Management Statistics for an Interface on page 250
- *ethernet oam lfm high-threshold*
- *ethernet oam lfm link-monitor frame-seconds*
- *ethernet oam lfm link-monitor frame-seconds-summary*
- *ethernet oam lfm link-monitor symbol-period*
- *show ethernet oam lfm status*
- *show ethernet oam lfm statistics*

OAM Remote Fault Detection Feature

Failures in Ethernet connectivity that are caused by slowly degrading quality are difficult to notice. JunosE Software enables you to configure the OAM application to monitor the receive path of the link for quality and generate Event Notification PDUs to the remote peer. The following failure conditions can be communicated by the remote entity to its peer using the Flags field of an Information OAM PDU (Figure 30 on page 228):

Link Fault

This event type signifies that a loss of signal is detected by the receiver; for example, the peer's laser is not operating correctly. A link fault is sent once per second in the Information OAM PDU. Link fault is applicable only when the physical sublayer is capable of independent transmit and receive operations. If you configure this option, the local OAM entity detects loss-of-signal conditions that occurred in the receive path of the link. The local entity influences the state of the link based on an Information OAM PDU with the Link Fault bit set in the Flags field that it receives from the remote peer.

Dying Gasp

This event type denotes that an unrecoverable condition, such as a power failure, has taken place. This type of condition is vendor specific. A notification about the condition might be sent immediately and continuously. If you configure this option, the local OAM entity detects unrecoverable error conditions that occurred in the receive path of the link. The local entity influences the state of the link based on an Information OAM PDU with the Dying Gasp bit set in the Flags field that it receives from the remote peer.

Critical Event

This event type indicates that an unspecified vendor-specific critical event has occurred. A critical event might be sent immediately and continuously. If you configure this option, the local OAM entity detects critical event error conditions that occurred in the receive path of the link. The local entity influences the state of the link based on an Information

OAM PDU with the Critical Event bit set in the Flags field that it receives from the remote peer.

You can specify the action to be taken by the system when the configured link-fault event occurs, such as disabling the interface or failing over to the secondary port on GE-2 and GE-HDE line modules that are paired with GE-2 SFP I/O modules with physical link redundancy. You can also configure the OAM application to react to event notifications received from the peer.

If a link fault is detected, any Information OAM PDUs sent with the Link Fault bit set do not contain any TLV data. Any OAM PDU received with these flags set are processed with priority by the router. Other link events, such as Errored Symbol Period Event and Errored Frame Event, which result in threshold values being exceeded are notified using TLVs in Event Notification PDUs.

In JunosE Software, you can configure the OAM application to monitor the receive path of the link for quality and generate Event Notification PDUs to the remote peer. You can also configure the OAM application to respond to event notifications received from the peer.

Related Documentation

- OAM Feature Overview on page 232
- OAM Messages on page 228
- Configuring 802.3ah OAM Link-Fault Management on page 240
- Monitoring OAM Link-Fault Management Sessions on All Configured Interfaces on page 255

- *ethernet oam lfm remote-failure*
- *show ethernet oam lfm summary*

OAM Remote and Local Loopback Feature

Remote loopback mode ensures link quality between the router and a remote peer during installation or troubleshooting. JunosE Software can place a remote entity into loopback mode (if remote loopback mode is supported by the remote entity). When you place a remote entity into loopback mode, the interface receives the remote loopback request and puts the interface into remote loopback mode. When the interface is in remote loopback mode, all frames except OAM PDUs are looped back without any changes made to the frames. OAM PDUs continue to be sent and processed.

A local OAM entity in active mode can start a remote loopback of its peer through a Loopback Control OAM PDU that contains the option to enable loopback. During the initiation phase, the local OAM entity discards any locally sourced non-OAM PDUs. When the peer receives a loopback request, and assuming that it supports the service, it sets the forwarding state to loop any received non-OAM PDUs; any locally generated non-OAM PDUs are discarded while in loopback (Figure 32 on page 231). This forwarding state is conveyed to the peer using an Information OAM PDU. When the Information OAM PDU is received after loopback is disabled, the local OAM entity resumes transmission of locally sourced non-OAM PDUs, in addition to OAM PDUs. You can prevent two active mode OAM entities from simultaneously placing each other into loopback mode by making sure that the lower valued source address is the entity that is placed in loopback mode (Figure 32 on page 231).

Because OAM PDUs are processed during remote loopback, variables can be retrieved to measure the link performance. The initiating OAM entity stops the remote loopback process by sending another Loopback Control OAM PDU with the option to disable the looping of any non-OAM PDUs. When the loopback feature is enabled, the forwarding process counts the number of packets and bytes transmitted to the peer, and the number of packets and bytes received from the peer.

NOTE: The peer in loopback mode might intentionally discard data frames to accommodate OAM traffic. OAM PDUs are assigned a higher priority than regular data packets when oversubscription of the allocated bandwidth occurs.

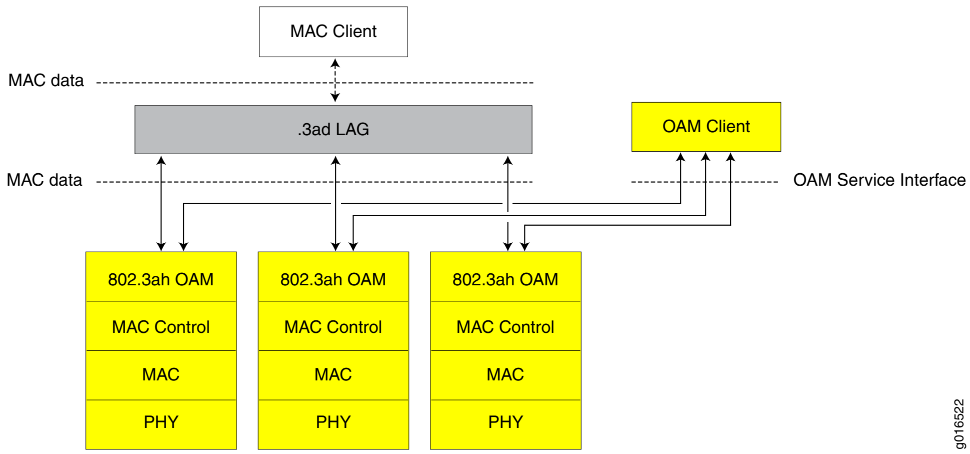
Related Documentation

- OAM Messages on page 228
- Configuring 802.3ah OAM Link-Fault Management on page 240
- Example: Enabling Remote Loopback Support on the Local Interface on page 247
- *ethernet oam lfm remote-loopback supported*
- *ethernet oam lfm remote-loopback*

Interrelationship of OAM Link-Fault Management with Ethernet Subsystems

802.3ah OAM assists the monitoring of individual Ethernet links. Link aggregation mechanisms, such as 802.3ad, operate above the 802.3ah sublayer. Figure 33 on page 238 shows the interconnection between OAM and LAG bundles. You can use the results of OAM link monitoring to configure the LAG sublayer accordingly to improve failover and recovery times.

Figure 33: Interrelationship Between 802.3ah OAM and 802.3ad LAG



Link events can be generated locally using the health monitoring of an Ethernet link. These link events can be supplied to the link aggregation multiplexing logic, in addition to the logic generated by the LAC protocol. For example, if the configured symbol-period threshold is exceeded, you can configure the link aggregator to remove the member link from the bundle and rebalance the bundle. Where a LAG member link is designated as a redundant member, you can use the link monitor functionality to trigger the failover and the reversal of the link.

Certain line module and IOA combinations support physical level redundancy. The redundancy feature enables a primary Gigabit Ethernet link to fail over to the secondary link without signaling higher-layer protocols and by maintaining the same MAC address on the link. The preservation of the same MAC address on the link also retains bindings to the MAC address (for example, ARP entries). When the OAM entity signals that a health monitoring threshold is exceeded, the event can trigger the failover to the secondary link.

JunosE Software implements the Marker Responder segment of the Marker protocol. If the OAM entity signals a link event, such as the exceeding of a high threshold value, using the health monitoring system, then Marker Response PDUs are not sent in such circumstances.

Related Documentation

- OAM Feature Overview on page 232
- OAM Elements Overview on page 229
- Configuring 802.3ah OAM Link-Fault Management on page 240
- Guidelines for Configuring 802.3ah OAM Link-Fault Management on page 239

Guidelines for Configuring 802.3ah OAM Link-Fault Management

Keep the following points in mind when you configure OAM link-fault management on Ethernet interfaces:

- The OAM application transmits and receives an OAM PDU at a maximum frequency of every 100 milliseconds (10 OAM PDUs/sec) on every port on the set of IOAs for a line module. The most impact on performance might occur on ES2 4G LMs with two GE-8 I/O modules (16 ports) that cause 160 OAM PDUs to be transmitted per second. Also, the OAM application must query MAC-layer statistics (up to 6-8 for frame errors) every 10 seconds and errored symbol statistics per second.
- Because the outage time of the forwarding controller can last up to about 15 seconds or longer, the 802.3ah OAM feature does not support unified ISSU. The link-fault management of Ethernet interfaces is halted during a unified ISSU operation. Monitoring resumes immediately after the unified ISSU operation is completed.
- Although OAM configurations on an interface are preserved during a stateful SRP switchover procedure, the retention of such OAM settings depends on the time that the stateful SRP switchover process takes to complete. If the stateful SRP switchover operation causes a traffic disruption of more than two seconds, the previously configured OAM settings are affected. We recommend that you configure the number of PDUs that are failed to be received by an OAM entity before it generates a link

fault/down event to be greater than two, and the high and low thresholds for an error to be exceeded on an Ethernet OAM interface to be more than two seconds.

- 802.3ah OAM functionality is not supported on SRP Ethernet interfaces. Also, JunosE Software does not support unidirectional operation of Ethernet OAM links, which enable the OAM entities to send Link Fault Information OAM PDUs to the peer whenever a receive path failure is detected. In addition, an active mode OAM entity can retrieve and respond to the performance variables that it receives from its peer entity. However, the local OAM entity does not send a list of such performance variables that it can process from the peer.

Related Documentation

- OAM Feature Overview on page 232
- Interrelationship of OAM Link-Fault Management with Ethernet Subsystems on page 238
- Configuring 802.3ah OAM Link-Fault Management on page 240

Configuring 802.3ah OAM Link-Fault Management

Ethernet OAM link-fault management can be used for physical link-level fault detection and management. It uses a new, optional sublayer in the data link layer of the OSI model. Ethernet OAM can be implemented on any full-duplex point-to-point or emulated point-to-point Ethernet link. A system-wide implementation is not required; OAM can be deployed on particular interfaces of a router. Transmitted Ethernet OAM messages or OAM PDUs are of standard length, untagged Ethernet frames within the normal frame length limits in the range 64–1518 bytes.

To configure OAM link-fault management on an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface, and enable IEEE 802.3ah OAM support on the interface. When the IEEE 802.3ah OAM protocol is enabled on a physical interface, the discovery process is automatically triggered. The default discovery mode of a local interface is active.

   host1(config)#**interface GigabitEthernet 6/0**
   host1(config-if)#**ethernet oam lfm**

   NOTE: You must enable the OAM link-fault management feature to be able to configure parameters that govern the link monitoring and management process.

   All of the following steps are optional. You can choose which of the OAM configurations you want to set up on the interface to enable link-fault administration. If you enable OAM support on the interface without specifying any of the other parameters, such as discovery mode and threshold settings, default values are assumed for those attributes.

2. Specify whether the interface or the peer initiates the discovery process by configuring the link discovery mode to active or passive.

   host1(config-if)#**ethernet oam lfm mode active**

In this case, the discovery mode of the interface is set as active. In active mode, the interface discovers and monitors the peer on the link if the peer also supports IEEE 802.3ah OAM functionality. An OAM entity in active mode initiates the discovery process by sending an Information OAM PDU to the multicast address of the slow protocol (0180.c200.0002) at a configured rate. In a carrier environment, the customer edge (CE) devices are normally configured for passive mode operation, whereas the provider edge (PE) equipment is configured for active mode operation.

3. Specify the frequency, in milliseconds, at which OAM PDUs are sent to the peer to keep the OAM association alive.

   host1(config-if)#**ethernet oam lfm pdu-transmit-interval 200**

   In this example, the local interface is configured to send Information OAM PDUs every 200 milliseconds to the remote peer. This OAM PDU includes local, remote, and organization-specific information, and contains a local-information TLV.

   The rate of transmission of OAM PDUs can be a number in the range 100–1000 milliseconds; the default value is 1000 milliseconds.

4. Specify the number of OAM PDUs that a local OAM client can fail to receive from a remote peer before a link-fault event is triggered. The product of the PDU timer and the PDU loss threshold equals the lost-link timer value, which is used to reset the Discovery state diagram that maintains the states of the OAM entities and determines the condition of the link based on various stored values.

   host1(config-if)#**ethernet oam lfm pdu-lost-threshold 4**

   In this example, the local interface is set to wait for 4 OAM PDUs to be missed from the remote peer before it generates a link-fault event. You can configure the local interface to wait for a larger number of OAM PDUs to be missed from the remote peer in networks that are prone to high losses and fluctuating performances, such as jitter, higher latency, and poor transmission of packets.

   The number of OAM PDUs that can fail to be received from a remote peer before the local OAM entity triggers a link-fault event can be in the range 3–10; the default period is 5 PDUs.

5. Configure the interface to detect local link faults and send events to the remote OAM entity when problems are noticed on the link. When link monitoring is enabled, the interface sends event OAM protocol data units (PDUs) when errors occur and interprets event OAM PDUs from the remote peer. Link monitoring can be effective only if both the local client and remote peer agree to support it.

   You can specify the event threshold values on an interface for the local errors that occur or a period of time during which such local errors are detected. The following are the error events that you can track using the OAM functionality:

   - Error frame events
   - Symbol error events
   - Error frame second events

   a. Configure a low threshold value for sending frame error events, which when exceeded causes an Errored Frame Event TLV to be sent to the remote OAM entity.

The Errored Frame Event TLV counts the number of errored frames detected during the specified period.

host1(config-if)#**ethernet oam lfm link-monitor frame-seconds threshold high 600**

In this case, a low error frame threshold of 600 frames is set. When this threshold is exceeded, an Errored Frame Event TLV is sent to the remote peer.

b. Configure a high threshold value for frame errors, which when exceeded triggers an action.

host1(config-if)#**ethernet oam lfm link-monitor frame-seconds threshold high 60**

In this case, a high error frame threshold of 60 frames is set. When this threshold is exceeded, the action configured on the interface using the **ethernet oam lfm high-threshold action** command is taken.

c. Configure a period of time during which error frames are counted for both high and low threshold settings. The time duration is specified in hundred millisecond units.

host1(config-if)#**ethernet oam lfm link-monitor frame-seconds window 10**

In this case, the window during which error frames are counted is set as 10 hundred millisecond units. The configured window is valid for both high and low threshold settings. The high and low threshold settings are reset whenever a new window, during which errors are counted, commences.

d. Configure a low threshold value for errored frame seconds, which causes an Errored Frame Seconds Summary Event TLV to be sent to the OAM entity when the threshold is exceeded. The Errored Frame Seconds Summary Event TLV counts the number of errored frame seconds that occurred during the specified period. An errored frame second is any 1-second period that has at least one errored frame.

host1(config-if)#**ethernet oam lfm link-monitor frame-seconds-summary threshold low 60**

In this case, a low errored frame seconds threshold of 60 frame seconds is set. When this threshold is exceeded, an Errored Frame Seconds Summary Event TLV in an Event Notification OAM PDU is sent from the local OAM entity to the remote peer.

e. Configure a high threshold value for errored frame seconds, which when exceeded triggers an action.

host1(config-if)#**ethernet oam lfm link-monitor frame-seconds-summary threshold high 6**

In this case, a high threshold of 6 errored frame seconds is set. When this threshold is exceeded, the action configured on the interface using the **ethernet oam lfm high-threshold action** command is taken.

f. Specify a period of time in which frame seconds summary error events are counted for both high and low threshold settings. The time period used for counting events is specified in seconds.

host1(config-if)#**ethernet oam lfm link-monitor frame-seconds-summary window 10**

In this case, frame seconds summary events are detected during a period of 10 seconds. The configured window is valid for both high and low threshold settings. The high and low threshold settings are reset whenever a new window, during which errors are counted, commences.

g. Configure a low threshold value for symbol error events that causes an Errored Symbol Period Event TLV to be sent to the OAM entity when it is exceeded. The Errored Symbol Period Event TLV counts the number of symbol error events that occurred during the specified period.

   host1(config-if)#**ethernet oam lfm link-monitor symbol-period threshold low 60**

   In this case, a low symbol errors threshold of 60 symbols is set. When this threshold is exceeded, an Errored Symbol Period Event TLV in an Event Notification OAM PDU is sent from the local OAM entity to the remote peer. This event is generated if the symbol error count is equal to or greater than the specified threshold for that period.

h. Configure a high threshold value for symbol errors, which when exceeded, triggers an action.

   host1(config-if)#**ethernet oam lfm link-monitor symbol-period threshold low 10**

   In this case, a low symbol errors threshold of 10 symbols is set. When this threshold is exceeded, the action configured on the interface using the **ethernet oam lfm high-threshold action** command is taken.

i. Specify a period of time in which symbol error events are counted for both high and low threshold settings. The time period is specified in seconds.

   host1(config-if)#**ethernet oam lfm link-monitor symbol-period window 10**

   In this case, symbol error events are counted over a period of 10 seconds. The configured window is valid for both high and low threshold settings. The high and low threshold settings are reset whenever a new window, during which errors are counted, commences.

   NOTE: We recommend that you do not use a multiple of the number of symbols because the window size varies greatly, depending on the speed of the link. For example, a 10 Gigabit Ethernet link generates 10.3x10M symbols per second. If the window has a lower bound of 1M symbols, sampling the symbol error statistic occurs every 97 microseconds.

   Some of the interfaces do not support statistics for errored symbol events. If you configure a monitor for symbol errors on such interfaces, the setting does not have any effect.

6. Configure a specific action to occur when a high threshold for an error is exceeded on an Ethernet OAM interface. You can configure the OAM application to influence the operational state of the link, when a link quality threshold is exceeded or a critical event PDU is received from the peer, or both. If you configured the action for an OAM

event to disable an Ethernet OAM interface when a high threshold for an error is exceeded, the link moves to the operational down status.

```
host1(config-if)#ethernet oam lfm high-threshold action failover
```

In this case, when the high threshold is exceeded for a local link error, a failover occurs to the secondary link of the redundant port on GE-2 and GE-HDE line modules that are paired with GE-2 SFP I/O modules.

7. Configure the Ethernet OAM link-fault management functionality to detect failure conditions that occurred at a remote peer and influence the state of the link based on an Event Notification PDU received from the remote peer. You can also set the action to be performed when a failure condition is observed in the link. If you enable detection of faults that occurred at the remote peer, the local OAM entity monitors unspecified critical event, unrecoverable error, and loss-of-signal conditions that the remote peer notifies it using an Information OAM PDU with the Critical Event, Dying Gasp, and Link Fault bits set in the Flags field.

   ```
   host1(config-if)#ethernet oam lfm remote-failure critical-event action disable-interface
   host1(config-if)#ethernet oam lfm remote-failure dying-gasp action disable-interface
   host1(config-if)#ethernet oam lfm remote-failure link-fault action disable-interface
   ```

   The operational status of the interface is set to down when an OAM PDU is received from the remote peer by the local OAM entity to signal fault conditions at the remote entity.

8. Set an interface into loopback mode to enable the current Ethernet OAM configuration for the interface of the local OAM entity to allow initiation of remote loopback operation or to respond to a remote loopback request from a peer.

   ```
   host1(config-if)#ethernet oam lfm remote-loopback supported
   ```

   NOTE: You must configure the interface of the local OAM entity to be placed in remote loopback mode and respond to loopback requests from the remote peer by using the **ethernet oam lfm remote-loopback supported** command before you enable the remote peer to loop back PDUs by using the **start** or **stop** keywords with the **ethernet oam lfm remote-loopback** command in Privileged Exec mode. Otherwise, a warning message is displayed prompting you to configure the interface of the local OAM entity to be placed in remote loopback mode.

   Also, the remote peer can place the local OAM entity in loopback mode only if you configured the **ethernet oam lfm remote-loopback supported** command on the local entity to enable remote loopback functionality on the local entity.

9. Configure the local OAM entity to instruct the remote peer at the specified interface to start looping back the non-OAM PDUs that it receives from the local OAM entity or to stop the resending of such received non-OAM PDUs from the local entity.

   a. Enable the remote loopback operation on a remote OAM entity, which causes the remote entity at the specified Gigabit Ethernet interface to loop any received non-OAM PDUs back to the local entity.

host1#ethernet oam lfm remote-loopback start interface gigabitEthernet 1/0

This configuration setting is not preserved across a reboot. The setting that you configured on the local OAM entity to start or stop the loopback operation on the remote peer is not available after a warm or cold restart of the router, because the router does not store the secure logs in NVS.

NOTE: If you attempt to enable the loopback operation on a remote OAM entity by entering the **ethernet oam lfm remote-loopback start** command, an error message is displayed if the remote entity is not configured for loopback behavior and if the interface of the local entity is not placed into loopback mode (to send and receive loopback PDUs).

b. Alternatively, you can disable the remote loopback operation on the remote OAM entity by instructing it to not loop back any received non-OAM PDUs from the local OAM entity.

host1#ethernet oam lfm remote-loopback stop interface gigabitEthernet 1/0

When you halt the remote loopback operation to cause the remote peer to not loop back any PDUs that it receives from the local entity by using the **ethernet oam lfm remote-loopback stop** command, the number of frames and bytes that are transmitted from the local entity to the remote peer when the local interface is in loopback mode, and the number of frames and bytes that are received from the remote peer when the remote peer is in loopback mode are displayed using appropriate field labels at the CLI prompt. You can also view the calculated loopback parameter values later from the Remote Loopback section in the output of the **show ethernet oam lfm status** command.

Related Documentation

- Guidelines for Configuring 802.3ah OAM Link-Fault Management on page 239
- Monitoring OAM Link-Fault Management Configuration for an Interface on page 252
- Monitoring OAM Link-Fault Management Discovery Settings for an Interface on page 247
- Monitoring OAM Link-Fault Management Statistics for an Interface on page 250
- Monitoring OAM Link-Fault Management Sessions on All Configured Interfaces on page 255

- *ethernet oam lfm*
- *ethernet oam lfm mode*
- *ethernet oam lfm high-threshold*
- *ethernet oam lfm link-monitor frame-seconds*
- *ethernet oam lfm link-monitor frame-seconds-summary*
- *ethernet oam lfm link-monitor symbol-period*
- *ethernet oam lfm pdu-lost-threshold*
- *ethernet oam lfm pdu-transmit-interval*

- *ethernet oam lfm remote-failure*
- *ethernet oam lfm remote-loopback*
- *ethernet oam lfm remote-loopback supported*

Example: Configuring 802.3ah OAM Link-Fault Management and Enabling Remote Failure Monitoring on an Interface

The following example shows how to enable the OAM link-fault management feature on an interface and configure a link monitoring operation for frame seconds events on the interface.

1. Specify the Gigabit Ethernet interface on which OAM link-fault management needs to be enabled, and configure the link discovery mode as active. When OAM functionality is enabled on the interface, the discovery mode of the local OAM entity is set to active by default. An OAM entity in active mode initiates the discovery process by sending an Information OAM PDU to the multicast address of the slow protocol (0180.c200.0002) at a configured rate.

   host1(config)#**interface gigabitEthernet 4/1**
   host1(config-if)#**ethernet oam lfm mode active**

2. Configure the Ethernet OAM link-fault management functionality to detect link-fault and dying-gasp conditions that occurred in the receive path of the link and influence the state of the link based on an Event Notification PDU received from the remote peer. Link Fault means a loss of signal, Dying Gasp means an unrecoverable condition such as a power failure.

   host1(config-if)#**ethernet oam lfm remote-failure dying-gasp action disable-interface**
   host1(config-if)#**ethernet oam lfm remote-failure link-fault action disable-interface**

3. Configure the local interface to be disabled when the high threshold for an error condition is exceeded. The OAM functionality unconditionally attempts to influence the operational state of the interface to down.

   host1(config-if)#**ethernet oam lfm high-threshold action disable-interface**

4. Configure link monitoring operations for frame error events on the interface. Specify the high threshold in number of frames for frame error events as 200, which when exceeded causes an action to be triggered. Specify a low threshold for frame error events, which when exceeded causes an Errored Frame Seconds Summary Event TLV to be sent to the peer, as 20 frames. Also, set the window during which frame error events are counted as 300 hundred millisecond units or 30 seconds.

   host1(config-if)#**ethernet oam lfm link-monitor frame-seconds threshold high 200**
   host1(config-if)#**ethernet oam lfm link-monitor frame-seconds threshold low 20**
   host1(config-if)#**ethernet oam lfm link-monitor frame-seconds window 300**

Related Documentation

- Configuring 802.3ah OAM Link-Fault Management on page 240
- *ethernet oam lfm mode*
- *ethernet oam lfm high-threshold*

- *ethernet oam lfm link-monitor frame-seconds*
- *ethernet oam lfm link-monitor frame-seconds-summary*
- *ethernet oam lfm link-monitor symbol-period*
- *ethernet oam lfm remote-failure*

Example: Enabling Remote Loopback Support on the Local Interface

You can allow a remote entity to set a local interface into remote loopback mode on all Ethernet interfaces on the E Series routers in which OAM link-fault management support is present. When a remote-loopback request is sent by a remote entity, the JunosE Software places the local interface into loopback mode. When an interface is in loopback mode, all frames except OAM PDUs are looped back without any changes to the frames. OAM PDUs continue to be sent to the management plane and processed. By default, the remote loopback feature is not enabled.

The following example configures the Gigabit Ethernet interface for OAM link-fault management and enables it to be placed in remote loopback mode to respond to loopback requests from the peer.

```
host1(config)#interface gigabitEthernet 4/1
host1(config-if)#ethernet oam lfm
host1(config-if)#ethernet oam lfm remote-loopback supported
```

Related Documentation

- OAM Remote and Local Loopback Feature on page 237
- *ethernet oam lfm*
- *ethernet oam lfm remote-loopback supported*

Monitoring OAM Link-Fault Management Discovery Settings for an Interface

Purpose Display the results of the Ethernet OAM link-fault management discovery process for a particular interface, such as the hardware status of the interface, operational status of the interface, status of the established link discovery mode of the local and remote OAM entities, states of the multiplier and parser functions of the OAM sublayer for the local and remote OAM entities, and functionalities and configuration of the local and remote OAM entities.

Action To display information regarding the OAM discovery process for a specific interface:

```
host1#show ethernet oam lfm discovery GigabitEthernet 4/1
GigabitEthernet 4/1 is Up, Administrative status is Up
OAM status is  Down
Local(0090.1A03.0063):
 Mode          : active
 Capabilities  : link events
 Mux Action    : Forwarding
 Parser Action  : Forwarding
Remote(0090.690a.0202):
 Mode          : passive
 Capabilities  : link events, loopback, variable retrieval
```



```
Mux Action    : Forwarding
Parser Action  : Forwarding
```

Meaning Table 17 on page 248 lists the **show ethernet oam lfm discovery** command output fields.

Table 17: show ethernet oam lfm discovery Output Fields

Field Name	Field Description
Interface type/name *interfaceSpecifier*	Status of the hardware on this interface: • Up—Hardware is operational • Down—Hardware is not operational
Administrative status	Operational state that you configured for this interface: • Up—Interface is enabled • Down—Interface is disabled
OAM status	Status of link-fault management functionality on the interface. The operational status of an interface does not depend only on the OAM status. Other factors, such as the administrative state of the interface, also impact the operational state • Up—Link-fault management feature is activated on the interface • Down—Link-fault management feature is disabled on the interface, either because it was not activated or because it was turned off as a result of error conditions
Local (*address*)	MAC address of the interface of the local entity
Mode	Discovery mode of the interface of the local OAM entity: • active—The interface discovers and monitors the peer on the link if the peer also supports IEEE 802.3ah OAM functionality. An OAM entity in active mode initiates the discovery process by sending an Information OAM PDU to the multicast address of the slow protocol (0180.c200.0002) at a configured rate. The default discovery mode of the OAM client is active • passive—An OAM entity does not initiate the discovery process. You cannot perform link-fault management if you configure both the local client and the remote peer for passive mode operation

Table 17: show ethernet oam lfm discovery Output Fields *(continued)*

Field Name	Field Description
Capabilities	Functions that the interface of the local entity can perform, such as link monitoring or responding to remote loopback requests. With this information a peer can determine what functions are supported and accessible; for example, loopback capability. All the configured settings on the local interface for OAM tasks are displayed in this field Displays one or more of the following values: • unidirectional—Indicates the ability to operate a link in a unidirectional mode for diagnostic purposes • loopback—Indicates whether remote loopback is supported or unsupported • link events—Indicates whether interpreting link events is supported or unsupported on the remote peer • variable retrieval—Indicates whether the local OAM entity supports receipt of performance MIB variables from its peer entity variable requests using Variable Request OAM PDUs
Mux Action	State of the multiplexer functions of the OAM sublayer for the local OAM entity. Device is forwarding non-OAM PDUs to the lower sublayer or discarding non-OAM PDUs
Parser Action	State of the parser functions of the OAM sublayer for the local OAM entity. Device is forwarding non-OAM PDUs to higher sublayer, looping back non-OAM PDUs to the lower sublayer, or discarding non-OAM PDUs
Remote (*address*)	MAC address of the interface of the remote peer
Mode	Discovery mode of the interface of the remote peer: • active—The interface discovers and monitors the entity on the other side of the link if that entity also supports IEEE 802.3ah OAM functionality. • passive—An OAM entity does not initiate the discovery process. You cannot perform link-fault management if you configure both the local client and the remote peer for passive mode operation

Table 17: show ethernet oam lfm discovery Output Fields *(continued)*

Field Name	Field Description
Capabilities	Functions that the interface of the remote peer can perform, such as link monitoring or responding to remote loopback requests. With this information a peer can determine what functions are supported and accessible; for example, loopback capability. All the configured settings on the local interface for OAM tasks are displayed in this field Displays one or more of the following values: • unidirectional—Indicates the ability to operate a link in a unidirectional mode for diagnostic purposes • loopback—Indicates whether remote loopback is supported or unsupported • link events—Indicates whether interpreting link events is supported or unsupported on the remote peer • variable retrieval—Indicates whether the local OAM entity supports receipt of performance MIB variables from its peer entity variable requests using Variable Request OAM PDUs
Mux Action	State of the multiplexer functions of the OAM sublayer for the remote peer. Device is forwarding non-OAM PDUs to the lower sublayer or discarding non-OAM PDUs
Parser Action	State of the parser functions of the OAM sublayer for the remote peer. Device is forwarding non-OAM PDUs to higher sublayer, looping back non-OAM PDUs to the lower sublayer, or discarding non-OAM PDUs

Related Documentation

- Configuring 802.3ah OAM Link-Fault Management on page 240
- *show ethernet oam lfm discovery*

Monitoring OAM Link-Fault Management Statistics for an Interface

Purpose Display detailed information about the Ethernet OAM link-fault management packets that are processed by a particular interface, such as the number of Information OAM PDUs transmitted and received, the number of Event Notification PDUs transmitted and received, Duplicate Event Notification PDUs transmitted and received, and Loopback Control PDUs transmitted and received.

Action To display details about the Ethernet OAM link-fault management packets that are analyzed by a specific interface:

```
host1#show ethernet oam lfm statistics GigabitEthernet 4/1
GigabitEthernet 4/1
In:
Information OAMPDUs                     :291
Event Notification OAMPDUs              :1
 Errored Frame                          :1
 Errored Symbol                         :0
Duplicate Event Notification OAMPDUs    :0
```

```
Loopback Control OAMPDUs               :0
Unsupported OAMPDUs                    :0

Out:
Information OAMPDUs                    :291
Event Notification OAMPDUs             :0
 Errored Frame                          :0
 Errored Symbol                        :0
Loopback Control OAMPDUs               :0
Duplicate Event Notification OAMPDUs   :0
```

Meaning Table 18 on page 251 lists the **show ethernet oam lfm statistics** command output fields.

Table 18: show ethernet oam lfm statistics Output Fields

Field Name	Field Description
Interface type/name *interfaceSpecifier*	Name and type of the Ethernet interface, in the interface specifier format, for which link-fault management packet details are displayed
In	Details about the types of PDUs that are sent from the interface
Information OAMPDUs	Number of Information OAM PDUs transmitted from the interface to the remote peer
Event Notification OAMPDUs	Number of unique Event Notification PDUs transmitted from the interface, when the number of errors equals or exceeds the configured low threshold for a specified time period. A breakdown of the types of errors that resulted in the generation of an Event Notification PDU is also displayed
Errored Frame	Number of errored frame event TLVs that have been transmitted since the OAM sublayer was reset
Errored Symbol	Number of symbols error event TLVs that have been transmitted since the OAM sublayer was reset
Errored Frame Seconds Summary	Number of framed seconds error event TLVs that have been transmitted since the OAM sublayer was reset
Loopback Control OAMPDUs	Total number of loopback control PDUs transmitted
Duplicate Event Notification OAMPDUs	Number of duplicate event notification OAM PDUs transmitted
Unsupported OAMPDUs	Number of unsupported OAM PDUs sent
Out	Details about the types of PDUs that are received on the interface
Information OAMPDUs	Number of Information OAM PDUs received on the interface from the remote peer

Table 18: show ethernet oam lfm statistics Output Fields *(continued)*

Field Name	Field Description
Event Notification OAMPDUs	Number of unique Event Notification PDUs received on the interface, when the number of errors equals or exceeds the configured low threshold for a specified time period. A breakdown of the types of errors that resulted in the generation of an Event Notification PDU is also displayed
Errored Frame	Number of errored frame event TLVs that have been received since the OAM sublayer was reset
Errored Symbol	Number of symbols error event TLVs that have been received since the OAM sublayer was reset
Errored Frame Seconds Summary	Number of framed seconds error event TLVs that have been received since the OAM sublayer was reset
Loopback Control OAMPDUs	Total number of loopback control PDUs received
Duplicate Event Notification OAMPDUs	Number of duplicate event notification OAM PDUs received
Unsupported OAMPDUs	Number of unsupported OAM PDUs received Based on the counts displayed for the supported OAM PDUs in the corresponding fields, you can determine additional details, such as the number of Variable Request PDUs, by viewing the value displayed in the Unsupported OAM PDUs field and comparing it against the total number of all supported OAM PDUs

Related Documentation

- Configuring 802.3ah OAM Link-Fault Management on page 240
- *show ethernet oam lfm statistics*

Monitoring OAM Link-Fault Management Configuration for an Interface

Purpose Display the current Ethernet OAM link-fault management configuration for a particular interface, such as the discovery mode of the interface, the interval at which OAM PDUs are transmitted to the remote peer, the number of OAM PDUs that can be missed from the remote peer before a link-fault event is generated, actions that are taken when the high threshold for an error is exceeded or when an OAM PDU from a remote peer signifies a fault condition, link monitoring attributes, and remote loopback settings.

Action To display the runtime settings of link-monitoring and general OAM operations for a particular interface:

```
host1#show ethernet oam lfm status GigabitEthernet 4/0
GigabitEthernet 4/0
  Mode:                   Passive
  Transmit-interval:      1000 ms
  Loss-threshold:         5 packets
```

```
  Event Action:             disable
  High Threshold Action:    disable
  Remote-loopback:          supported

Frame-seconds Error Monitor
  Window:                   30 (100 millisecond units)
  Low threshold:            20 errored frames
  High threshold:           none

Remote-loopback:
  Frames sent:              104437
  Bytes sent:               16167885

  Frames received:          104437
  Bytes received:            16167885
```

Meaning Table 19 on page 253 lists the **show ethernet oam lfm status** command output fields.

Table 19: show ethernet oam lfm status Output Fields

Field Name	Field Description
Interface type/name *interfaceSpecifier*	Name and type of the Ethernet interface, in the interface specifier format, for which link-fault management packet details are displayed
Mode	Discovery mode of the interface: • Active—The interface discovers and monitors the peer on the link if the peer also supports IEEE 802.3ah OAM functionality. An OAM entity in active mode initiates the discovery process by sending an Information OAM PDU to the multicast address of the slow protocol (0180.c200.0002) at a configured rate. The default discovery mode of the OAM client is active • Passive—An OAM entity does not initiate the discovery process. You cannot perform link-fault management if you configure both the local client and the remote peer for passive mode operation
Transmit-interval	Number of milliseconds, after which Information OAM PDUs are sent from the local OAM entity to the remote peer to maintain the OAM association in an active state
Loss-threshold	Number of Information OAM PDUs that can be missed from the remote peer before a link fault event is triggered
Event Action	Action to be performed on an interface when an Information OAM PDU is received from the remote peer by the local OAM entity to signal a fault condition at the remote entity. Possible values are: • disable—Sets the OAM functionality to unconditionally attempt to influence the operational state of the interface to down • failover—On GE-2 and GE-HDE line modules that are paired with GE-2 SFP I/O modules with physical link redundancy, causes the transition of the link from active to redundant

Table 19: show ethernet oam lfm status Output Fields *(continued)*

Field Name	Field Description
High Threshold Action	Action that occurs when the high threshold for an error is exceeded: • disable—Sets the OAM functionality to unconditionally attempt to influence the operational state of the interface to down. If the interface is a member link of a LAG bundle and at least one other viable link (redundant member or another active/up link) is present, OAM attempts to influence the operational state of the link to down. Otherwise, no action is taken • failover—On GE-2 and GE-HDE line modules that are paired with GE-2 SFP I/O modules with physical link redundancy, causes the transition of the link from active to redundant
Remote-loopback	Indicates whether the local interface is enabled for remote loopback functionality and whether it can respond to remote loopback requests from peers: supported or unsupported. An OAM entity can put its remote peer into loopback mode using the Loopback control OAM PDU. In loopback mode, every frame received is transmitted back on the same port (except for OAM PDUs, which are needed to maintain the OAM session) to the local entity
Frame-seconds Error Monitor	Displays a detailed classification of frame seconds errors event TLVs since the OAM sublayer was reset
Window	Specified amount of time in milliseconds during which frame seconds error events are counted
Low threshold	Lowest value for frame second error events in number of frames, which when exceeded causes an Errored Frame Seconds Summary Event TLV to be sent to the peer
High threshold	Highest value for frame second error events in number of frames, which when exceeded causes an action to be triggered
Frame-errors Error Monitor	Displays a detailed classification of frame errors event TLVs since the OAM sublayer was reset
Window	Specified amount of time in hundred-millisecond units during which frame error events are counted
Low threshold	Lowest value for frame error events in number of frames, which when exceeded causes an Errored Frame Event TLV to be sent to the peer
High threshold	Highest value for frame error events in number of frames, which when exceeded causes an action to be triggered
Symbol-errors Error Monitor	Displays a detailed classification of symbol errors event TLVs since the OAM sublayer was reset

Table 19: show ethernet oam lfm status Output Fields *(continued)*

Field Name	Field Description
Window	Specified amount of time in seconds during which symbol error events are counted
Low threshold	Lowest value for frame second error events in number of error symbols, which when exceeded causes an Error Symbol Period TLV to be sent to the peer
High threshold	Highest value for symbol error events in number of error symbols, which when exceeded causes an action to be triggered
Remote-loopback	Displays details on the non-OAM PDUs that are sent and received when the interface is in remote loopback mode
Frames sent	Number of frames sent to the remote peer that are transmitted back or looped to the local interface
Bytes sent	Number of bytes sent to the remote peer that are transmitted back or looped to the local interface
Frames received	Number of frames received from the remote peer when the local entity is in loopback mode
Bytes received	Number of bytes received from the remote peer when the local entity is in loopback mode

Related Documentation

- Configuring 802.3ah OAM Link-Fault Management on page 240
- *show ethernet oam lfm status*

Monitoring OAM Link-Fault Management Sessions on All Configured Interfaces

Purpose

Display a summary of the MAC-layer OAM status of all Ethernet links on which OAM link-fault management is enabled. This command displays the state of each of the links, with a brief synopsis about the OAM configurations of each of the links, such as the discovery mode of the OAM entity, state of the discovery mechanism, MAC address of the remote peer, the Link Flag details that contain information about the interface, and loopback configuration.

Action

To display Ethernet OAM link-fault management settings for all interfaces on which OAM is enabled:

`host1#`**show ethernet oam lfm summary**

```
FastEthernet4/0 is Up, Administrative status is Up
        Ethernet OAM (ver 1)
  Mode: Active, Discovery State: Send any
  Remote address: 0090.0a38.0208
  Flags: Remote-Stable Remote-State-Valid Local-Stable
  Loopback: Supported, Remote enabled
```

```
FastEthernet4/1 is Down, Administrative status is Up
        Ethernet OAM (ver 1)
  Mode: Active, Discovery State: Fault
  Remote address: 0090.0b92.032a
  Flags: Local Evaluating
  Loopback: Supported, Local enabled
```

Meaning Table 20 on page 256 lists the **show ethernet oam lfm summary** command output fields.

Table 20: show ethernet oam lfm summary Output Fields

Field Name	Field Description
Interface type/name *interfaceSpecifier*	Status of the hardware on this interface: • Up—Hardware is operational • Down—Hardware is not operational
Administrative status	Operational state that you configured for this interface: • Up—Interface is enabled • Down—Interface is disabled
Ethernet OAM (ver 1)	Revision of the OAM configuration. A new revision results from each change to the configuration.
Mode	Discovery mode of the interface: • Active—The interface discovers and monitors the peer on the link if the peer also supports IEEE 802.3ah OAM functionality. An OAM entity in active mode initiates the discovery process by sending an Information OAM PDU to the multicast address of the slow protocol (0180.c200.0002) at a configured rate. The default discovery mode of the OAM client is active • Passive—An OAM entity does not initiate the discovery process. You cannot perform link-fault management if you configure both the local client and the remote peer for passive mode operation
Remote address	MAC Address of the remote peer

Table 20: show ethernet oam lfm summary Output Fields *(continued)*

Field Name	Field Description
Discovery State	State of the discovery mechanism: • Fault—When the discovery process enters the Fault state, the local PDU value is set based on the value of local link status field. While the local link status is set to Fail, the local OAM entity remains in this state indicating to the remote peer there is link fault. This condition is accomplished by sending Information OAM PDUs once per second with the Link Fault bit of the Flags field set and no Information TLVs in the Data field • Active send local—A local entity configured in Active mode sends Information OAM PDUs that only contain the Local Information TLV. This state is referred to as Active Send Local. While in this state, the local entity waits for Information OAM PDUs received from the remote entity • Passive wait—An entity configured in passive mode waits until receiving Information OAM PDUs with Local Information TLVs before sending any Information OAM PDUs with Local Information TLVs. This state is called Passive Wait. By waiting until first receiving an Information OAM PDU with the Local Information TLV, a passive entity cannot complete the OAM Discovery process when connected to another entity in passive mode • Send any—After an OAM PDU has been received indicating the remote device is satisfied with the respective settings, the local device enters the SEND_ANY state. This is the expected normal operating state for OAM on fully operational links • Send local remote—After the local entity has received an Information OAM PDU with the Local Information TLV from the remote entity, the local entity begins sending Information OAM PDUs that contain both the Local and Remote Information TLVs. This state is called Send Local Remote. If at any time the settings on either the local or remote entity change resulting in the local OAM client becoming unsatisfied with the settings, the discovery process returns to the Send Local Remote state • Send local remote ok—If the local OAM client deems the settings on both the local and remote entities are appropriate, it enters the Send Local Remote Ok state. If at any time the settings on the local OAM client change resulting in the remote OAM client becoming unsatisfied with the settings, the OAM discovery process returns to the Send Local Remote Ok state

Table 20: show ethernet oam lfm summary Output Fields *(continued)*

Field Name	Field Description
Flags	Provides information about the physical link; displays one or more of the following values: • Remote-Stable—Indicates remote OAM client acknowledgment and acceptance of local OAM state information. False indicates that remote entity either has not received or remote state settings do no match local state information. True indicates that remote entity has received and remote state settings match local state information • Local-Stable—Indicates local OAM client acknowledgment and acceptance of remote OAM state information. False indicates that local entity either has not received or local state settings do not match remote state information. True indicates that local entity has received and local state settings match remote state information • Local-Evaluating—The Local Stable and Local Evaluating bits of the Flags field communicate the status of the local discovery process to the peer. When the OAM discovery process is started, the local entity sets the Local Stable to 0 and Local Evaluating bits to 1 indicating OAM discovery has not completed. When Local Stable is set to 1 and Local Evaluating is set to 0 and Remote Stable is set to 1 and Remote Evaluating is set to 0 indicating that the settings of both the local and remote OAM clients match, the OAM Discovery process has successfully completed • Remote-State-Valid—Indicates the OAM client has received remote state information found within Local Information TLVs of received Information OAM PDUs. False indicates that OAM client has not seen remote state information. True indicates that the OAM client has seen remote state information
Loopback	State of the loopback functionality of the local and remote OAM entities; displays one or more of the following values: • Supported—Indicates that the Ethernet OAM configuration on the interface is configured to initiate remote loopback or respond to a remote loopback request it receives from a peer. When you place a remote entity into loopback mode, the interface receives the remote-loopback request and puts the interface into remote-loopback mode. When a remote-loopback request is sent by a remote entity, the local interface is placed into loopback mode • Local enabled—Indicates that the loopback operation is enabled on the specified interface of the local OAM entity, which causes the local entity to loop back the received frames other than OAM PDUs to the remote peer • Remote enabled—Indicates that the loopback operation is enabled on the specified interface of the remote peer, which causes the remote peer to loop back all received frames other than OAM PDUs to the local OAM entity

Related Documentation

- Configuring 802.3ah OAM Link-Fault Management on page 240

- *show ethernet oam lfm summary*

CHAPTER 9

Configuring Point-to-Point Protocol

This chapter describes how to configure a Point-to-Point Protocol (PPP) interface on E Series routers.

This chapter contains the following topics:

- Understanding PPP on page 262
- Understanding PPP Link Control Protocol on page 263
- Broadband Remote Access Support for PPP Overview on page 265
- Understanding Extensible Authentication Protocol on page 267
- Remote Peer Scenarios During Negotiation of PPP Options on page 271
- IPCP Lockout and Local IP Address Pool Restoration Overview on page 272
- IPCP Negotiation with Optional Peer IP Address Overview on page 273
- Processing NCP Negotiations in a Dual-Stack Environment Overview on page 273
- Overview of Processing IPCP Negotiations for Dual-Stack Subscribers on page 275
- Releasing IPv4 Addresses During Termination of PPP Sessions on page 277
- IPCP Renegotiation of IPv4 Addresses for Dual-Stack Subscribers on page 277
- PPP Platform Considerations on page 279
- PPP References on page 281
- Configuring PPP over a Serial Interface on page 281
- Overview of Sequencing NCP Packets for POS Interfaces with PPP Encapsulation on page 283
- Sequencing NCP Packets for POS Interfaces with PPP Encapsulation on page 283
- Configuring Optional PPP Configuration Tasks on page 284
- Adding a Description or Alias for a Static PPP Interface on page 285
- Configuring the IPCP Lockout Option for Each PPP Interface on page 286
- Configuring the IPCP Netmask Option for Each PPP Interface on page 286
- Configuring the KeepAlive Timeout for an Interface on page 287
- Disabling the Negotiation of the Local Magic Number on page 287
- Configuring the Router to Ignore a Mismatch of the Peer Magic Number on page 288

- Configuring the Maximum Number of Renegotiation Attempts from a PPP Client on page 288
- Configuring the Maximum Receive Unit for PPP on page 289
- Forcing the PPP Interface into a Passive Mode on page 290
- Configuring the PPP Peer to Take Precedence of DNS and WINS Addresses on page 290
- Configuring the PPP IP Address as Optional in an IPCP Request on page 291
- Terminating the PPP Session on page 291
- Configuring PPP Authentication on page 291
- Requesting Authentication from a PPP Peer on page 292
- Configuring IPCP Renegotiations in a Dual-Stack Network for Optimal Utilization of Released IPv4 Addresses on page 294
- Modifying the Challenge Length for CHAP Authentication on page 295
- Configuring the Maximum Retries for PAP and CHAP Authentication on page 295
- PPP Accounting Statistics Overview on page 296
- Setting a Baseline for PPP Interface Statistics on page 297
- Monitoring PPP Interfaces on page 297
- Monitoring Multilinked and Nonmultilinked PPP Interfaces on page 309
- Monitoring the Status of an IP Address in IPCP Configuration on page 311
- Monitoring AAA IPv4 Address Saving on page 311
- Troubleshooting PPP Interfaces on page 312
- Monitoring the Maximum Timeout of PPP Sessions on page 313

Understanding PPP

PPP provides a standard method for transporting multiprotocol datagrams over a point-to-point link. PPP uses the High-Speed Data Link Control (HDLC) protocol for its physical interface and provides a packet-oriented interface for the network-layer protocols.

Internet Protocol Control Protocol (IPCP) (which negotiates for transport of IP version 4 datagrams), IPv6CP (which negotiates for transport of IP version 6 datagrams), the OSI Network Layer Control Protocols (OSINLCPs), and Multiprotocol Label Switching (MPLS) run within PPP.

The router supports dynamic PPP interfaces. For details, see “Configuring Upper-Layer Dynamic Interfaces” on page 519.

Framing

The software restricts the use of the general HDLC protocol (RFC 1662) to unnumbered mode:

- HDLC address field is 0xFF (all stations)
- HDLC control field is 0x03 (to indicate unnumbered mode)

The router does not support the following framing features:

- Numbered mode (RFC 1663)
- Autodetection of encapsulation

Error Frames

The router relies on higher-layer protocols to recover from PPP data loss. All unrecognized protocol data units (PDUs) are discarded; however, statistics are maintained for packets dropped.

Related Documentation

- Understanding PPP Link Control Protocol on page 263
- Broadband Remote Access Support for PPP Overview on page 265
- Configuring PPP over a Serial Interface on page 281

Understanding PPP Link Control Protocol

PPP's Link Control Protocol (LCP) establishes a PPP link by negotiating with the PPP peer at the other end of a proposed connection. When two routers initialize a PPP dialogue, each router sends control packets to the peer. The control packets contain a list of LCP options and corresponding values that the sending peer uses to define its end of the link, such as the maximum receive unit (MRU).

LCP negotiations continue until the peers either converge (that is, reach an agreement about values for connection parameters) or abandon attempts to establish a connection.

If you configure a PPP interface without an IP interface or profile, the router negotiates LCP, but then terminates LCP after 2 to 3 minutes. Previously, the behavior in such a circumstance was to negotiate LCP and then leave LCP open.

For static PPP interfaces, whenever LCP achieves a stopped state because of termination, negotiation failure, or some other cause, it goes into passive mode and waits for the other side of the connection to restart the negotiation process. Once in passive mode, the router periodically attempts to negotiate with the other side according to an exponential timeout algorithm.

For static PPP interfaces, the router waits 15 seconds, attempts negotiation, waits 30 seconds if it fails, attempts negotiation, waits 60 seconds if it fails, and so on. The timeout periods are 15 seconds, 30 seconds, 60 seconds, 2 minutes, 4 minutes, 8 minutes, and 15 minutes. Once it reaches the 15-minute timeout, the router attempts negotiation every 15 minutes until successful. When LCP reaches the open state, the timer resets to 15 seconds.

Dynamic PPP interfaces are always torn down when LCP achieves a stopped state. For more information, see "Configuring Upper-Layer Dynamic Interfaces" on page 519.

LCP Negotiation Parameters

LCP can negotiate many PPP options, as follows:

- MRU size—Maximum receive unit size (always accepted).
- Magic number—Randomly generated number used to identify one end of a point-to-point connection. Each side negotiates its magic number, taking note of each other's magic number. If both sides discover that the magic numbers they are negotiating are the same, each side attempts to change its magic number. If they are not successful, and the magic numbers remain the same, the session terminates because of the loopback that is detected. Magic numbers are always accepted.

  By default, the router always attempts to negotiate a local magic number. The peer can also determine whether to negotiate its magic number—the peer magic number. The router always accepts a peer's attempt to negotiate its magic number.

  If the peer does not attempt to negotiate its magic number, you can configure the router to ignore a mismatch of the peer magic number and retain the PPP connection. For details, see "Validation of LCP Peer Magic Number" on page 264.
- Authentication—Requested if configured.
- Protocol-Field-Compression (PFC) and Address-and-Control-Field-Compression (ACFC)—Accepted, but never requested.
- Multilink PPP—Additional options can be negotiated when Multilink PPP is configured. See "MLPPP Overview" on page 316.
- Async-Control-Character-Map (ACCM—Supported by PPP when used with an L2TP Network Server (LNS). ACCM allows PPP to indirectly support asynchronous PPP connections tunneled via a third-party L2TP Access Concentrator (LAC). PPP on the router uses the ACCM configuration data as supplied by the LAC via proxy LCP. The router does not directly support asynchronous PPP connections and will not negotiate an ACCM option unless directed to do so by a third-party LAC.

PPP can also detect a loopback that occurs after LCP is negotiated, provided that:

- No loopback occurs during LCP negotiations.
- A loopback is introduced after LCP negotiation without forcing LCP renegotiation. (LCP is renegotiated if the lower layer goes down or if an LCP confReq is received from the other end.)

Validation of LCP Peer Magic Number

If the peer has not negotiated an LCP magic number, you can configure the router to ignore a mismatch of the LCP peer magic number and retain the PPP connection.

Previously, the router terminated a PPP connection with a non-conforming peer when it received LCP echo request packets or LCP echo reply packets from the peer with a magic number that did not match the LCP peer magic number on the router. This is still the current default behavior if you do not explicitly configure the router to ignore the LCP peer magic number mismatch if the peer has not negotiated the magic number and retain the PPP connection.

Configuring the router to ignore the peer magic number mismatch and retain the PPP connection is useful if your network includes peers that send a non-null or invalid magic number in the LCP echo request and reply packets despite having not negotiated the magic number. In this situation, the router expects to receive a null magic number from the peer, and terminates the PPP connection unless you configure it to ignore the peer magic number mismatch and retain the connection.

To configure the router to ignore the LCP peer magic number mismatch and retain the PPP connection, use the **ppp magic-number ignore-mismatch** command from Interface Configuration mode or Subinterface Configuration mode. For more information, see *ppp magic-number ignore-mismatch*.

To verify configuration of LCP peer magic number validation on the router, you can use the **show ppp interface** command. For more information, see "Monitoring PPP Interfaces" on page 297.

Keep the following points in mind when configuring the router to ignore the peer magic number mismatch and retain the PPP connection:

- If the peer negotiates the magic number but sends the router an LCP echo request or reply packet that contains a null or invalid magic number, the router strictly terminates the PPP connection. The router can ignore a mismatch of the LCP peer magic number only when the peer has not negotiated the magic number.
- Using the **ppp magic-number disable** command to disable negotiation of the magic number on the router does not affect validation of the peer magic number. When you issue the **ppp magic-number disable** command, the router sets only the local magic number to null, but does not change or validate the peer magic number. (For more information, see *ppp magic-number disable*.)

You can also configure validation of the LCP peer magic number for static MLPPP interfaces, dynamic PPP interfaces, and dynamic MLPPP interfaces. For more information about configuring static MLPPP interfaces, see "Configuring Multilink PPP" on page 315. For more information about using profiles to configure dynamic PPP and dynamic MLPPP interfaces, see "Dynamic Interface Configuration Using a Profile" on page 565.

Related Documentation

- Understanding PPP on page 262
- Disabling the Negotiation of the Local Magic Number on page 287
- Configuring the Router to Ignore a Mismatch of the Peer Magic Number on page 288
- Monitoring Multilinked and Nonmultilinked PPP Interfaces on page 309

Broadband Remote Access Support for PPP Overview

Broadband Remote Access Server (B-RAS) is an application that aggregates the output from digital subscriber line access multiplexers (DSLAMs). B-RAS provides user Point-to-Point Protocol (PPP) sessions and PPP session termination and routes traffic onto the backbone. See *JunosE Broadband Access Configuration Guide* for details on B-RAS.

Broadcast accounting can be enabled in a PPP profile by assigning a broadcast virtual router group to the PPP profile using the **ppp aaa-accounting-broadcast** command. If the virtual router group exists in the router, authentication, authorization and accounting (AAA) reads the configurations of the assigned virtual router group and sends the broadcast accounting messages on the basis of virtual router group configurations to the broadcast accounting servers.

The router provides an enhanced version of PPP to accommodate B-RAS with the following features:

- Internet Protocol Control Protocol (IPCP) extensions for Windows Internet Name Service (WINS) and Domain Name System (DNS) name server addresses
- Password Authentication Protocol (PAP)
- Challenge Handshake Authentication Protocol (CHAP)
- Keepalive timeout
- Session timeout
- Inactivity timeout
- Accounting

Authentication

The router acts as an authenticator. It demands authentication from a remote PPP peer but refuses to authenticate itself.

Rate Limiting for PPP Control Packets

The router implements rate limiting for PPP control packets to protect the corresponding PPP interface from denial-of-service (DoS) attacks. The interface discards control packets when the rate of control packets received exceeds the rate limit for PPP interfaces.

A PPP interface has a rate limit control that is non-configurable and always in effect; the rate limit is the same for all PPP interfaces. In addition, each interface instance maintains its own state and statistics counters for tracking the rate. The rate limit for PPP control packets is approximately 10 packets per second.

For a PPP interface, the router increments the discards counter in the **show ppp interface** command display to track the number of PPP control packets discarded on receipt (in) or discarded before they were transmitted (out) on this interface.

For examples of the **show ppp interface** command display, see "Monitoring PPP Interfaces" on page 297.

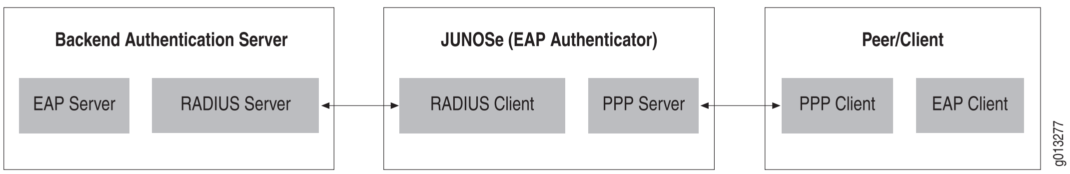
Related Documentation

- IPCP Lockout and Local IP Address Pool Restoration Overview on page 272
- Configuring the IPCP Lockout Option for Each PPP Interface on page 286
- Configuring the KeepAlive Timeout for an Interface on page 287
- Configuring PPP Authentication on page 291

Understanding Extensible Authentication Protocol

The JunosE Software supports Extensible Authentication Protocol (EAP) for authenticating a peer before allowing network layer protocols to transmit over the link. EAP supports multiple authentication methods, including EAP-TLS and EAP-MD5-Challenge. The EAP server and the peer negotiate the specific authentication method to be used. Figure 34 on page 267 illustrates the three components required for EAP: an EAP authenticator, an EAP server, and an EAP client.

Figure 34: Authentication with EAP



After LCP negotiation, JunosE starts the EAP negotiation process by initiating an identity exchange with the EAP client on the peer. The router sends an EAP identity request packet to the peer, which replies with an EAP identity response packet. After this exchange, the E Series router acts only as a pass-through device, enabling the EAP server residing on the backend authentication server to select and negotiate the particular EAP authentication method directly with the EAP client on the peer.

The JunosE Software forwards or discards packets received from the backend authentication router and the peer depending on the identifying code contained in the packet.

The E Series router forwards:

- Packets received from the peer with a Response code
- Packets received from the backend authentication server with a Request, Success, or Failure code

The E Series router discards:

- Packets received from the peer with a Request, Success, or Failure code
- Packets received from the backend authentication server with a Response code

The JunosE Software determines the outcome of the authentication based only on the Accept or Reject indication sent by the RADIUS server

EAP Types

The JunosE Software has been qualified to work with the EAP authentication methods—known as EAP types—described in Table 21 on page 268. Other EAP authentication methods have not been qualified with the JunosE Software

Table 21: Supported EAP Types

EAP Type	Behavior
1—Identity	When LCP negotiation completes, PPP sends an initial EAP identity request packet to the peer. The EAP identity response packet received from the peer is forwarded to AAA. AAA forwards the response as an Access-Request to the RADIUS server hosted on the backend authentication server.
2—Notification	The JunosE Software forwards Notification requests from the backend authentication server to the peer and Notification responses from the peer to the server. The JunosE Software does not initiate any Notification requests or responses.
3—NAK	The JunosE Software forwards the NAKs received from the peer to the backend authentication server.
4—MD5-Challenge	The JunosE Software acts as a pass-through for the EAP-MD5-Challenge negotiated between the peer and backend authentication server.
13—TLS	The JunosE Software acts as a pass-through for the EAP-TLS negotiated between the peer and backend authentication server.

EAP Packet Retransmission

PPP retransmits the EAP request packets to the peer. The RADIUS client retransmits the EAP response packets to the RADIUS Server. The request packets to the peer are governed by nonconfigurable values for retransmission attempts and interval. The configuration of the RADIUS client determines retransmission values for response packets to the RADIUS server. The retransmission values are as follows:

- PPP makes five attempts to retransmit an EAP request before the authentication attempt is terminated. You cannot configure the number of retransmission attempts.
- When an EAP request is transmitted, a timer is started with a nonconfigurable retransmission interval value of 3 seconds. When the timer expires, the EAP request is retransmitted.

In some cases, you might want a longer retransmission interval. For example, you might need to accommodate the additional time required by a user to enter information or scan a fingerprint or retina. RADIUS can instruct the JunosE Software to wait longer by passing an appropriate Session-Timeout attribute in the RADIUS Access-Challenge packet. This retransmission interval value applies only to the EAP request packet present in the RADIUS Access-Challenge packet.

The Session-Timeout attribute value overrides the default retransmission interval value, up to a maximum of 30 seconds. If RADIUS recommends a greater value, then PPP resets it back to 30 seconds in order to avoid longer or infinite delays.

EAP Behavior in an L2TP Environment

EAP behavior in an L2TP environment varies depending on whether the router acts as a LAC or an LNS,

When the E Series Router Acts as a LAC

When PPP forwards an EAP identity response packet to AAA, AAA might be configured to return a tunnel response upon successful validation of the packet. You can use AAA domain maps, a AAA profile, or both to force such tunneling.

On an LAC, PPP forwards the PPP EAP authentication information to the LNS during the establishment of the L2TP session. This authentication information consists of the EAP type, the data appropriate to the type (such as a username) contained in the EAP identity response packet, and the identifier of the EAP identity response packet. If the LNS trusts the LAC, then the LNS uses this authentication information to resume the EAP negotiation where the LAC left off.

L2TP on an LAC forwards the PPP EAP authentication information in the Proxy Authen AVPs as described in L2TP Proxy Authenticate Extensions for EAP—draft-ietf-l2tpext-proxy-authen-ext-eap-01.txt (December 2006 expiration).

When the E Series Router Acts as an LNS

PPP on an LNS resumes the EAP negotiation operation by detecting the presence of EAP information in the proxy authentication data supplied by L2TP. PPP reconstructs the EAP identity response packet from the proxy authentication data and forwards it to AAA.

L2TP on an LNS processes the received Proxy Authen AVPs as described in L2TP Proxy Authenticate Extensions for EAP—draft-ietf-l2tpext-proxy-authen-ext-eap-01.txt (December 2006 expiration).

Limitations

EAP is subject to internal limits. When the E Series router acts as a pass-through between the backend authentication server and the peer, EAP packets traverse the controllers within the router. The size of EAP packets and fragments tends to be larger than the buffer exchange limit—1450 bytes—between the controllers. This intercontroller buffer exchange limit is tuned for the optimal system performance and scalability; also, when stacked over L2TP on LNS, it prevents PPP control packets from causing IP fragmentation and reassembly on the Ethernet downlink. Hence, if EAP is configured as a PPP authentication protocol, then EAP packet or fragment size is affected by the intercontroller buffer exchange limit as follows:

- The MRU value advertised by JunosE in the LCP request packet takes the lowest of the following values:
  - the lower layer MRU minus the PPP overhead
  - the configured MRU
  - 1450 bytes

- The MTU value is initialized by JunosE to the lowest of the following values:
  - the lower layer MTU minus the PPP overhead
  - the peer MRU
  - 1450 bytes

The MTU value is passed to RADIUS in an Access-Request packet by means of the Framed-Mtu attribute.

Performance

When EAP is configured on the router, it affects the performance and scalability of PPP in terms of round-trip packet exchanges, negotiations, EAP server requirements, and EAP client requirements. For information on the number of PPP interfaces supported with EAP, see the *Link Layer Maximums* tables in *Appendix A, System Maximums*, of the current *JunosE Release Notes*.

- Performance depends on the number of packets exchanged during the negotiation. When the number of packets exchanged increases—that is, when the number of round-trips increases—it takes longer to finish the interface negotiation. System resources are locked for a longer duration. As a result it takes longer to bring up all the interfaces.

  The number of round-trip message exchanges varies with the EAP authentication method. When no retransmission of packets takes place and there is no fragmentation, PAP and CHAP require one round-trip, EAP-MD5-Challenge requires two round-trips, and EAP-TLS requires four round-trips.

  Retransmission increases the number of round-trips. When the negotiated EAP authentication method requires fragmentation, such as for the exchange of large certificate chains, then the number of round-trips increases.
- The number of simultaneous EAP negotiations is limited to 50 because of resource limitations. Consequently, the time required to bring up interfaces when you configure EAP authentication is longer than when you specify PAP or CHAP authentication.
- EAP authentication methods fragment packets when the EAP packet size is greater than the link MTU. The EAP server must fragment the EAP packet to the size of the Framed-Mtu attribute contained in the RADIUS Access-Request packet.

  If the server fragments the packet to a larger size than specified by the attribute, then JunosE drops the packet, because the E Series router acts as a pass-through device and is not involved in the authentication method's fragmentation and reassembly mechanisms.

  On the other hand, if the EAP server fragments the EAP packet to a smaller size than specified by the attribute, then performance decreases because of the increased number of smaller packets that must be exchanged.
- The EAP client on the peer must fragment the EAP packets to the size of the link MTU on the E Series router. When it does not do so, performance can be affected.

Related Documentation

- Remote Peer Scenarios During Negotiation of PPP Options on page 271
- Configuring PPP Authentication on page 291

Remote Peer Scenarios During Negotiation of PPP Options

During a PPP configuration request, if any of the primary or secondary DNS option is rejected, or if they are unacceptable, the customer premises equipment (CPE) is prompted to negotiate the Internet Protocol Control Protocol (IPCP) primary and secondary DNS options that are locally available with the Broadband Remote Access Server (B-RAS). This provision is controlled by CLI and SNMP configuration options.

The following describes the peer negotiations in different scenarios:

- The CPE does not send the prompted options in the subsequent configure request
  - If B-RAS sends another NAK, it prompts the options again, until max configure-nak is exceeded
  - If B-RAS sends an ACK, it ignores the options and brings up the link
- The CPE negotiates a different option from the prompted options
  - If B-RAS sends another NAK, it prompts the options again, until max configure-nak is exceeded
  - If B-RAS sends an ACK, it ignores the options and brings up the link
- The CPE negotiates the prompted options but the option values are not acceptable
  - B-RAS sends another NAK with the prompted options, until max configure-nak is exceeded
- The CPE negotiates the prompted options but some option values are not acceptable
  - B-RAS sends a NAK for unacceptable options, until max configure-nak is exceeded
- The CPE stops responding on receiving the prompted options
  - B-RAS negotiation timer expires and the link is terminated
- The CPE negotiates the prompted option after the link comes up
  - This is treated as a renegotiation request and B-RAS sends an ACK/NAK until max renegotiation and max configure-nak counters are exceeded, respectively
- The CPE starts renegotiation without the prompted options
  - B-RAS renegotiates, until max renegotiation is exceeded
  - If B-RAS sends a NAK, it prompts the options, until max configure-nak is exceeded
  - If B-RAS sends an ACK, it ignores the options and brings up this link
- The CPE NAKs or rejects the prompted options

- This behavior is non-compliant with RFC because a configure-rej or a configure-nak must be sent only in response to a configure-req

The Link Control Protocol (LCP) state machine stores the various phases through which an IPCP negotiation and link establishment occurs. When an IPCP configuration request is received from a CPE in the closed state (such as an IPCP request with an IP address of 0.0.0.0 received from the CPE) at the provider edge (PE) device, the LCP negotiation moves to the stopped state and the PE device does not send a NAK packet to the CPE.

After an IP interface is created, which occurs when the PPP link is in the opened state and the PE device receives an open event, a Receive-Configure-Request (Bad) or RCR– event is sent to the CPE. This event occurs only if the previous event from the CPE was a close event or a bad IPCP configuration request was received. The RCR– event causes the configuration request and configuration NAK packets to be sent from the PE to the CPE device.

Related Documentation

- Understanding Extensible Authentication Protocol on page 267
- Broadband Remote Access Support for PPP Overview on page 265

IPCP Lockout and Local IP Address Pool Restoration Overview

You can configure the router to terminate invalid IPv4 subscribers and return the unused IPv4 addresses to the local address pool. When Internet Protocol version 6 Control Protocol (IPv6CP) is negotiated, the router waits for 10 seconds for Internet Protocol Control Protocol (IPCP) negotiation. If IPCP is not negotiated in 10 seconds, the interface blocks IPv4 over Network Control Protocol (NCP) packets and the IP address is returned to the local address pool. The subscribers must then reconnect to negotiate IPCP again.

The router assigns IPv4 and IPv6 addresses for a PPP subscriber after authentication in the following ways:

- RADIUS returns a valid IP address or a IPv6 prefix
- The configured local address pool returns a valid IP address

The subscriber can negotiate IPv4 addresses, IPv6 addresses, or both. After an IPv6 address is negotiated for an IPCP service, the PPP application waits for the negotiation of IPv4 address and then returns the assigned unused addresses to the local pool. By default, this feature is disabled. To enable the feature, issue the **ppp ipcp lockout** command from Interface Configuration, Subinterface Configuration, or Profile Configuration modes. This command terminates the invalid subscriber entry and prevents additional IPCP negotiations. When IPv6CP is active and if the IPCP must close, the router does not terminate PPP and Link Control Protocol (LCP) and does not return the address to the pool. In this case, AAA uses the assigned IP address and reassigns the same address when IPCP is negotiated again.

Related Documentation

- Broadband Remote Access Support for PPP Overview on page 265
- IPCP Negotiation with Optional Peer IP Address Overview on page 273
- Configuring the IPCP Lockout Option for Each PPP Interface on page 286

- Configuring the PPP IP Address as Optional in an IPCP Request on page 291
- Configuring the IPCP Netmask Option for Each PPP Interface on page 286

IPCP Negotiation with Optional Peer IP Address Overview

During normal operation for an IPCP negotiation, if the client does not request a specific IP address, the server sends an IP address obtained from RADIUS or from the local address pool.

If the client seeks a specific IP address, on receiving a NAK from the server, it sends a confReq message without specifying the IP address option. In this case, even though the server sends an IPCP confAck message , the server terminates the client because the server requires an IP address from the client.

You can use the **ppp peer-ip-address-optional** command in Global Configuration mode to specify that the peer IP address is optional. By default, this command is disabled. This feature also supports high availability (HA) and unified in-service software upgrade (Unified ISSU).

NOTE: Even though the **ppp peer-ip-address-optional** command is configured, on receiving a NAK from the server, if the client sends an IPCP confReq message with specific IP address, IPCP is terminated after a maximum of five attempts.

When the IPCP negotiation succeeds by configuring the **ppp peer-ip-address-optional** command, the server does not have the client IP address. An IP address from RADIUS or from the local pool is allocated to the client and the route towards this is added on the server even though the client is not assigned with this address. If you want the server to have the route to the client-requested IP address, use the framed-route RADIUS attribute or configure statics routes. The client adds or configures static routes towards the server for proper forwarding.

Related Documentation

- Configuring the PPP IP Address as Optional in an IPCP Request on page 291
- *show ppp peer-ip-address-optional*
- *ppp peer-ip-address-optional*

Processing NCP Negotiations in a Dual-Stack Environment Overview

JunosE Software enables interoperability of clients running Windows Vista platforms with ICMPv6 router advertisements that are sent from an E Series router configured for IPv6. This interoperability with Windows Vista clients is available only in environments where the PPP or PPPoE link is established over the Gigabit Ethernet interface on the provider edge (PE) router from the customer edge (CE) device. This functionality is not supported on connections established over other types of interfaces on the PE router. Support for Windows Vista clients is available only when ERX1440 and E320 routers are used as the requesting routers at the service provider edge of the network. The PPP or PPPoE link between the CE and PE devices can be configured for both IPv4 and IPv6 protocols for transmission of data. Such networks require that PE devices run a dual stack of IPv4 and IPv6 services. Neighbor Discovery and Prefix Delegation are supported in environments in which the subscriber is either an IPv6 subscriber or a combined IPv4 and IPv6 subscriber in a dual stack.

PPP includes a family of Network Control Protocols (NCPs) to establish and configure different network layer protocols. Internet Protocol version 6 Control Protocol (IPv6CP) is negotiated during the NCP phase and the interface ID is also negotiated during this phase. PPP's Link Control Protocol (LCP) establishes a PPP link by negotiating with the PPP peer at the other end of a proposed connection.

ERX routers save the order of the NCP negotiation process as soon as the process begins. When the router sends NCP configuration request packets for each NCP peer, it checks the recorded order and avoids sending the configuration request packet until the first NCP negotiation process is completed. The completion or failure of the first NCP negotiation process triggers the remaining NCP negotiation attempts.

For dynamic interfaces, NCP configuration request packets are sent after the corresponding upper interface (IPv4 or IPv6) is created and stacked. Therefore, the dynamic creation of interfaces is still permitted in any order until the upper interface is stacked. However, after the upper interface is stacked, the corresponding NCP configuration request packet is not sent if the first NCP negotiation process is not completed successfully.

In the case of static interface columns, because the upper interfaces are previously stacked, ERX routers initiate the NCP configuration requests as soon as LCP changes to the up state, instead of waiting for the client. In such a case, ERX routers do not send the NCP negotiation packets until they receive NCP configuration requests from the client and process the configuration requests based on the order of those received packets.

During Internet Protocol Control Protocol (IPCP) negotiations between a customer premises equipment (CPE) and the PE router, which can function as an LNS device in an L2TP tunnel, if RADIUS is used to authenticate the PPP clients, the Access-Accept message sent from the RADIUS server to the PE router or the RADIUS client might contain the MS-Primary-DNS-Server [311-28] and the MS-Secondary-DNS-Server [311-29] RADIUS VSA attributes. In such a scenario, the PE router or the B-RAS server processes these Microsoft VSAs and uses the primary and secondary DNS server addresses in the IPCP negotiation with the CPE for allocation of IP addresses. For more information about the handling of Microsoft VSAs during IPCP negotiations, see *Processing DNS Addresses from Microsoft RADIUS VSAs for PPP Clients During IPCP*.

Enabling IPCP and IPv6CP Negotiations for IPv4 and IPv6 Clients Based on RADIUS-Returned Attributes

Point-to-Point Protocol (PPP) uses Internet Protocol Control Protocol (IPCP) and Internet Protocol version 6 Control Protocol (IPv6CP) negotiations for assigning IP version 4 (IPv4) and IP version 6 (IPv6) addresses to authenticated PPP Broadband Remote Access Server (B-RAS) subscribers by using RADIUS-returned attributes or the local address pool configured on the router.

You can now enable IPCP and IPv6CP negotiations for IPv4 and IPv6 clients based only on RADIUS-returned attributes by issuing the **aaa radius-override-ncp-negotiation** command with the **enable** keyword.

IPCP negotiation is initiated for IPv4 clients only when the Framed-Ip-Address [8] attribute or Framed-Pool [88] attribute is returned by the RADIUS server. IPv6CP negotiation is initiated for IPv6 clients only when the Framed-Interface-Id [96] attribute, IPv6 prefix attributes, or IPv6 pool name attributes are returned by the RADIUS server.

Sending the Acct-Start and Acct-Update Messages After NCP Negotiation Is Completed

When authentication, authorization, and accounting (AAA) receives authentication-grant permission from the RADIUS server before NCP negotiation is started, an Acct-Start message is sent immediately to the RADIUS server. An Acct-Update message is sent from the B-RAS to the RADIUS server immediately after IP notifies AAA to set a user IP address or user interface ID.

However, you can use the **aaa accounting delay-start** command to delay the sending of the Acct-Start message from the B-RAS to the RADIUS server until Network Control Protocol (NCP) negotiation is completed.

You can use the **aaa accounting immediate-update-framed-ipv6-interfaceId-negotiation** command to send the Acct-Update message with the Framed-Interface-Id [96] RADIUS attributes from the B-RAS to the RADIUS server for PPPv6 users immediately after NCP negotiation is completed.

Related Documentation

- *Sending the Acct-Start and Acct-Update Messages After NCP Negotiation Is Completed Overview*
- Understanding PPP on page 262
- *Understanding IPCP and IPv6CP Negotiations for IPv4 and IPv6 Clients Based on RADIUS-Returned Attributes*

Overview of Processing IPCP Negotiations for Dual-Stack Subscribers

In a dual-stack network topology, which contains both IPv4 and IPv6 subscriber sessions, either as independent or combined sessions, you can enable the PPP application running on the router to send a message to the AAA server immediately after an IPv4 address is released by a client. The PPP link between the customer premises edge equipment (CPE) and the provider edge (PE) devices can be configured for both IPv4 and IPv6 protocols for transmission of data. The PPP application uses Link Control Protocol (LCP) negotiations to establish the connection with the subscriber.

After the PPP application authenticates the subscriber or CPE using the configured authentication protocol, the AAA server sends both IPv4 and IPv6 addresses in dual-stack networks to the PPP application. These addresses can be assigned from either the local address pool or the AAA server, which can be a server configured for RADIUS authentication. PPP enables the creation of a dynamic IPv4 or IPv6 interface, or both these types of interfaces based on the user environment. A dynamic interface is created, after the receipt of incoming requests from the subscriber or the CPE. The dynamic IPv4 or IPv6 interface created and stacked over the underlying physical PPP interface obtains its settings from the attached interface profile and the subscriber details returned by the AAA or RADIUS server. After a link has been established and optional facilities have been negotiated as needed by the LCP, PPP sends Network Control Protocol (NCP) packets.

When the CPE sends Internet Protocol Control Protocol (IPCP) messages during the NCP negotiation process, the AAA server sends IPv4 or IPv6 addresses to the CPE. Based on a successful negotiation, PPP notifies the IP application that the IP service is activated. The IP application installs an access route for the relevant IP address of the client to transmit data packets to it. When a dual-stack subscriber terminates the IPCP negotiation for only the IPv4 address, the IPv4 address is released to the PPP application. You can enable PPP to send a message to the AAA server immediately after the address is freed up by the client. This immediate, periodic notification enables optimal, effective usage of IPv4 addresses from the AAA server. This behavior is applicable only if the IPv4 address is allocated previously from the AAA server and not from the local address pool or the DHCP local server.

In such a scenario, the IPv6 address client of the dual-stack user is retained with the user. The released IPv4 address cannot be renegotiated. The client must send a new IPCP configuration request to obtain a fresh IPv4 address.

Guidelines for Configuring the AAA Server for Release of IPv4 Addresses for Dual-Stack Subscribers

Keep the following points in mind when you enable PPP to send a notification to the AAA server to release IPv4 addresses of dual-stack subscriber sessions for terminated IPv4 connections:

- The policies configured on the subscriber are retained even after the release of the IPv4 address and are reapplied for the new IPv4 address obtained after an IPCP renegotiation.
- Independent IPv4 or combined IPv4 and IPv6 sessions that contain hierarchical policies and external parent groups with rate-limit profiles configured for PPP sessions are maintained when PPP notifies AAA regarding the released IPv4 addresses. These policy management profiles are reactivated when the same subscriber renegotiates a session with a new IPv4 address.
- The combined IPv4 and IPv6 services statistics and user session statistics are also preserved for the entire lifetime of the PPP session.

Communicating to the AAA server about released IPv4 addresses is not effective in the following situations:

- If the IPv6 prefixes of dual-stack clients are released or renewed during the lifetime of the PPP session or if the IPv6 prefixes undergo a change during the time the PPP session is active.
- If the network contains only IPv4 subscribers.
- If the IPv4 addresses are allocated from the local address pool or the DHCP local server.

Related Documentation

- Configuring IPCP Renegotiations in a Dual-Stack Network for Optimal Utilization of Released IPv4 Addresses on page 294
- IPCP Renegotiation of IPv4 Addresses for Dual-Stack Subscribers on page 277
- Releasing IPv4 Addresses During Termination of PPP Sessions on page 277

Releasing IPv4 Addresses During Termination of PPP Sessions

When a request for termination of the IPCP negotiation is received from the CPE, the B-RAS application running on the router acknowledges this terminate-request for closing the IPv4 session by sending a terminate-acknowledgment packet. If the IPCP termination is successful, the IPv4 address that the RADIUS server allocated to the client is released. The RADIUS client, which is the router, sends an immediate accounting message that contains the Ipv4-release-control RADIUS VSA attribute [26-164] to the RADIUS server. In this immediate accounting message, the Framed-Ip-Address RADIUS IETF attribute [8] is not included. This Acct-Update message is sent immediately on release of the IPv4 address to the RADIUS server and the configured interval for Interim-Acct messages is not considered in such a case. The Ascend-Data-Filter attribute [242] is modified to delete the Framed-IP-Address attribute for the released IPv4 address.

After the immediate accounting message is sent, the Interim-Acct message is sent to the RADIUS server at the configured interval. The Interim-Acct message contains all the supported RADIUS attributes except the Framed-Ip-Address attribute. Because the IPv6 addresses are not released in this scenario, the Interim-Acct message contains the IPv6 prefixes.

Related Documentation

- IPCP Renegotiation of IPv4 Addresses for Dual-Stack Subscribers on page 277
- Overview of Processing IPCP Negotiations for Dual-Stack Subscribers on page 275
- *AAA Access Messages During IPCP Negotiations for Dual-Stack Subscribers*
- *AAA Accounting Messages During IPCP Negotiations for Dual-Stack Subscribers*

IPCP Renegotiation of IPv4 Addresses for Dual-Stack Subscribers

After the PE device or the router receives an IPCP configuration request from the CPE, which starts the IPCP negotiation process, the B-RAS application running on the router requests a new IPv4 address from the RADIUS server. The router or the RADIUS client sends an Access-Request message with the Ipv4-release-control RADIUS VSA attribute [26-164] included in it, along with all the other attributes that are supported for inclusion in the Access-Request message. After successful authentication, the RADIUS server sends the Access-Accept message with all of the supported attributes for all established sessions. For IPCP negotiations, the following attributes are not contained in the Access-Accept message if the Access-Request message contains the Ipv4-release-control attribute:

- [26-58] LI-Action
- [27] Session-Timeout

- [28] Idle-Timeout
- Service Manager attributes

The Access-Accept message received from the RADIUS server contains the updated IPv4 settings in the Ascend-Data-Filter attribute for IPv4 sessions. This modified policy defined in the Ascend-Data-Filter attribute for IPv4 clients is applied on the RADIUS client. The IPv4 policy associated with the Ascend-Data-Filter attribute is removed when the IPv4 address is released. If the Access-Accept message during an IPCP negotiation contains the IPv6 policy in the Ascend-Data-Filter attribute, it is disregarded. IPv6 policy parameters are already configured on the router when the authentication of the dual-stack subscriber occurred previously during initial establishment of the session. After an IPv4 address is received from the RADIUS server, a validation for a duplicate address is performed. If the received IPv4 address is determined to be a duplicate address, the subscriber session is completely terminated. If the duplicate address validation completes successfully, PPP performs IPCP negotiation with the CPE.

If the IPCP negotiation is successful, the RADIUS client sends an immediate accounting message with the Ipv4-release-control [26-164] and Framed-Ip-Address [8] attributes included in it without waiting for the configured interim accounting interval. If the IPCP negotiation is not successful, the Interim-Acct message is sent with the Ipv4-release-control [26-164] attribute added and the Framed-Ip-Address [8] attribute excluded. This type of interim accounting message indicates to the RADIUS server to release the IPv4 address to enable optimal utilization of the addresses in the pool.

After the IPCP renegotiation for the IPv4 address, the interim accounting message is sent with all of the supported attributes, including the negotiated address, the Framed-Ip-Address attribute, and IPv6 prefixes. If the RADIUS server sends only IPv6 prefixes and does not return an IPv4 address when the CPE sends an IPCP renegotiation for an IPv4 address, PPP sends a Protocol Reject message to the CPE. In this case, the interim accounting interval for communicating to the RADIUS server about released IPv4 addresses does not impact the configured interval for Acct-Start and Acct-Stop messages. If the RADIUS server sends only IPv4 addresses and does not return an IPv4 address when the CPE sends an IPCP renegotiation for an IPv6 address, PPP sends a Protocol Reject message to the CPE.

If the RADIUS server returns both the IPv4 and IPv6 addresses in the initial Access-Accept message, and if the CPE does not negotiate the IPv4 address, the IPv4 address is not released and the client can perform IPCP negotiations for the IPv4 address. The router initiates the IPCP negotiation only when the Framed-Ip-Address attribute is contained in the Access-Accept message. If the CPE sends a Protocol Reject packet to the router, the IPCP negotiation is not conducted for the current user session.

When the IPCP negotiations for both IPv4 and IPv6 addresses are terminated in a particular established subscriber session because no service is available, the complete session is closed. In this scenario, the router does not attempt to initiate an IPCP negotiation and sends an LCP terminate request packet and a PPPoE Active Discovery Termination (PADT) packet when NCP is not active in that session.

The attributes contained in the RADIUS access and accounting messages during an IPCP renegotiation for IPv4 addresses for clients in a dual-stack network is slightly different from the attributes included in these messages during the initial session establishment. See *AAA Access Messages During IPCP Negotiations for Dual-Stack Subscribers* and *AAA Accounting Messages During IPCP Negotiations for Dual-Stack Subscribers* for detailed information on the attributes transmitted in these messages.

Rate Limit on IPCP Negotiations

With the capability to enable PPP to notify the AAA server regarding released IPv4 addresses, the CPE can dynamically negotiate IPv4 addresses and exchange IPCP packets at any point during the PPP session. To prevent the B-RAS application on the router from being flooded with a large burst of IPCP packets from the CPE, a rate-limit is pre-configured for the IPCP negotiations for IPv4 addresses per subscriber and for the number of IPv4 address requests received per line module. These rate-limit settings are configured in the interface controller and not the forwarding controller. The time interval during which the number of renegotiation requests must be restricted is set to 60 seconds by default. Within this period, the maximum number of renegotiations permitted per client is six. For each PPP subscriber, a counter is maintained in the PPP application to record the number of IPCP negotiations performed during this time period to avoid a flood of IPCP packets. This counter is incremented after a successful IPCP negotiation. You can configure the maximum number of successful IPCP renegotiations for IPv4 addresses that the router can receive per subscriber by using the **ppp ipcp-max-negotiation *maxRenegotiationsCount*** command. These settings are effective when you configure the router with the capability to send a notification to the AAA server regarding released IPv4 addresses in a dual-stack environment by using the **aaa ipv4-addr-saving** command. You can also specify the interval during which the maximum number of IPCP renegotiations per subscriber needs to be restricted by using the **ppp ipcp-nego-duration *RenegotiationsInterval*** command.

If the CPE attempts to send an IPCP negotiation after the maximum permissible value is exceeded, a system logging message is generated and the statistics counter for rejected PPP sessions is incremented. IPCP negotiation is gracefully terminated by the PE device or the router. After the maximum number of configured IPCP negotiations per subscriber is exceeded, the router blocks that particular subscriber from further IPCP negotiations for a pre-configured time period.

Related Documentation

- Configuring IPCP Renegotiations in a Dual-Stack Network for Optimal Utilization of Released IPv4 Addresses on page 294
- Overview of Processing IPCP Negotiations for Dual-Stack Subscribers on page 275
- Releasing IPv4 Addresses During Termination of PPP Sessions on page 277
- *AAA Access Messages During IPCP Negotiations for Dual-Stack Subscribers*
- *AAA Accounting Messages During IPCP Negotiations for Dual-Stack Subscribers*

PPP Platform Considerations

You can configure PPP interfaces on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support PPP interfaces on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support PPP.

For information about the modules that support PPP interfaces on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support PPP.

Interface Specifiers

Some of the configuration task examples in this chapter use the *slot/port*[.*subinterface*] format to specify the physical interface on which you want to configure PPP. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Related Documentation

- Understanding PPP on page 262
- Broadband Remote Access Support for PPP Overview on page 265

PPP References

For more information about the PPP protocol, consult the following resources:

- L2TP Proxy Authenticate Extensions for EAP—draft-ietf-l2tpext-proxy-authen-ext-eap-01.txt (December 2006 expiration)
- RFC 1332—The PPP Internet Protocol Control Protocol (IPCP) (May 1992)
- RFC 1661—The Point-to-Point Protocol (PPP) (July 1994)
- RFC 1662—PPP in HDLC-like Framing (July 1994)
- RFC 1877—PPP Internet Protocol Control Protocol Extensions for Name Server Addresses (December 1995)
- RFC 1994—PPP Challenge Handshake Authentication Protocol (CHAP) (August 1996)
- RFC 2153—PPP Vendor Extensions (May 1997)
- RFC 2246—The TLS Protocol Version 1.0 (January 1999)
- RFC 2615—PPP over SONET/SDH (June 1999)
- RFC 2716—PPP EAP TLS Authentication Protocol (October 1999)
- RFC 3032—MPLS Label Stack Encoding (January 2001)
- RFC 3579—RADIUS EAP (September 2003)
- RFC 3748—Extensible Authentication Protocol (EAP) (June 2004)

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Web site at http://www.ietf.org for the latest drafts.

Related Documentation

- Understanding PPP on page 262
- Broadband Remote Access Support for PPP Overview on page 265

Configuring PPP over a Serial Interface

Before you configure a PPP interface, configure the interface or tunnel over which PPP traffic will flow. The procedures described in this topic assume that a physical interface has been configured. See the following chapters for additional information:

- “Configuring ATM” on page 3
- “Configuring Packet over SONET” on page 375

- “Configuring Point-to-Point Protocol over Ethernet” on page 387
- Configuring Channelized T3 Interfaces in *JunosE Physical Layer Configuration Guide*
- Configuring T3 and E3 Interfaces in *JunosE Physical Layer Configuration Guide*
- Configuring Unchannelized OCx/STMx Interfaces in *JunosE Physical Layer Configuration Guide*
- Managing Tunnel-Service and IPsec-Service Interfaces in *JunosE Physical Layer Configuration Guide*

To configure PPP over a serial interface:

1. From Global Configuration mode, specify the physical interface on which you want to configure PPP.

   host1(config)#**interface serial 3/0:2/5**

2. Specify PPP as the encapsulation method (data-link protocol) on the interface.

   host1(config-if)#**encapsulation ppp**

3. Assign an IP address and subnet mask for the interface.

   host1(config-if)#**ip address 192.168.22.10 255.255.255.0**

4. (Optional) Configure the packet over SONET (POS) interface on which PPP encapsulation is configured to transition to the up state only when the NCP packets are received in a sequence. If you configure this setting, the PPP sessions are not established if the NCP packets arrive in an out-of-order format. By default, the capability to establish PPP sessions only when the NCP packets arrive in a sequence is configured.

   NOTE: This step applies only if you configure the physical interface in Step 1 of this procedure as a POS interface by using the **interface pos** command. Otherwise, it is not required to set the POS interfaces with PPP encapsulation with this option on sequencing of NCP packets.

   host1(config)#**ppp ncp-Ordering-Required**

5. Verify that your configuration changes are correct.

   host1#**show ppp interface serial 3/0:2/5 config**

Related Documentation

- Understanding PPP on page 262
- PPP Platform Considerations on page 279
- *encapsulation ppp*
- *interface atm*
- *interface pos*
- *interface serial*
- *ip address*

- *ppp ncp-Ordering-Required*
- *show ppp interface*

Overview of Sequencing NCP Packets for POS Interfaces with PPP Encapsulation

For packet over SONET (POS) physical interfaces, you can configure PPP as the encapsulation method on the interface to enable a PPP dialogue to be initiated between the source and the destination points. After the completion of the link-establishment and authentication phases in a PPP session between two routers are completed, the network phase begins and the PPP connection is fully established. At this point, any higher level protocols (for example, IP protocols) can initialize and perform their own negotiations and authentication.

After a link has been established and optional facilities have been negotiated as needed by the LCP, PPP must send Network Control Protocol (NCP) packets to choose and configure one or more network-layer protocols, such as IP, IPX, or AppleTalk. Once each of the chosen network-layer protocols has been configured, datagrams from each network-layer protocol can be sent over the link. NCP initializes the PPP protocol stack to handle multiple network layer protocols, such as IPv4, IPv6, and Connectionless Network Protocol (CLNP).

To enable the PPP session to be established only when the NCP packets arrive in an ordered format on POS interfaces that are configured with PPP encapsulation, you can enter the **ppp ncp-Ordering-Required** command in Global Configuration mode. Otherwise, the POS interfaces with PPP encapsulation configured remain in the down state and are disabled unless the NCP packets arrive in a sequenced way. If you want the PPP session to be established and the POS interfaces to move to the up state even when the NCP packets arrive in an out-of-order format, you must enter the **no ppp ncp-Ordering-Required** command for the POS interfaces to be activated.

Related Documentation

- Sequencing NCP Packets for POS Interfaces with PPP Encapsulation on page 283
- *ppp ncp-Ordering-Required*

Sequencing NCP Packets for POS Interfaces with PPP Encapsulation

For packet over SONET (POS) physical interfaces with PPP encapsulation configured, the interfaces are not transitioned to the administratively up state if the arrival of the NCP packets is out of sequence. By default, the NCP packets need to be received in sequence to enable the POS interfaces to be in the enabled state.

To enable the PPP sessions to be established even when the NCP packets are not received in sequence:

- From Global Configuration mode, enter the **no ppp ncp-Ordering-Required** command.

  host1(config)#**no ppp ncp-Ordering-Required**

The POS interfaces with the PPP encapsulation method configured remain in the down state until this command is entered. By default, the NCP packets are received in a sequence format.

To cause the PPP sessions to be established only when the NCP packets are received in sequence:

- From Global Configuration mode, enter the **ppp ncp-Ordering-Required** command.

  host1(config)#**ppp ncp-Ordering-Required**

Related Documentation

- Overview of Sequencing NCP Packets for POS Interfaces with PPP Encapsulation on page 283
- *ppp ncp-Ordering-Required*

Configuring Optional PPP Configuration Tasks

You can optionally configure tasks on a PPP interface, such as specifying the alias or text description for a PPP interface, stop and restart a PPP session, as well as other tasks mentioned in this topic.

To optionally configure tasks on a PPP interface, such as:

- Assign a text description to a PPP interface.

  See "Adding a Description or Alias for a Static PPP Interface" on page 285.
- Configure the IPCP lockout option.

  See "Configuring the IPCP Lockout Option for Each PPP Interface" on page 286
- Configure the IPCP netmask option.

  See "Configuring the IPCP Netmask Option for Each PPP Interface" on page 286
- Specify the keepalive timeout value.

  See "Configuring the KeepAlive Timeout for an Interface" on page 287
- Disable negotiation of the local magic number.

  See "Disabling the Negotiation of the Local Magic Number" on page 287
- Configure the router to ignore a mismatch of the peer magic number.

  See "Configuring the Router to Ignore a Mismatch of the Peer Magic Number" on page 288
- Configure the maximum renegotiation attempts that a router can receive from a PPP client.

  See "Configuring the Maximum Number of Renegotiation Attempts from a PPP Client" on page 288
- Configure the maximum receive unit (mru) for PPP.

  See "Configuring the Maximum Receive Unit for PPP" on page 289

- Force the PPP interface into a passive mode.

  See "Forcing the PPP Interface into a Passive Mode" on page 290

- Resolve conflicts when the router and the PPP peer have different primary and secondary DNS and WINS name server addresses.

  See "Configuring the PPP Peer to Take Precedence of DNS and WINS Addresses" on page 290

- Configure the peer IP address as optional in an IPCP request.

  See "Configuring the PPP IP Address as Optional in an IPCP Request" on page 291

- Terminate the PPP session.

  See "Terminating the PPP Session" on page 291

- Configure PPP Authentication.

  See "Configuring PPP Authentication" on page 291

Related Documentation

- *ppp authentication*
- *ppp description*
- *ppp ipcp lockout*
- *ppp ipcp netmask*
- *ppp keepalive*
- *ppp magic-number disable*
- *ppp magic-number ignore-mismatch*
- *ppp max-negotiations*
- *ppp mru*
- *ppp passive-mode*
- *ppp peer*
- *ppp peer-ip-address-optional*
- *ppp shutdown*

Adding a Description or Alias for a Static PPP Interface

You can optionally assign a text description or alias to the static PPP interface from the Interface Configuration Mode. You can also assign the text description from the Subinterface Configuration Mode.

To assign a description to the static PPP interface:

- (Optional) From the Interface Configuration mode, assign the description:

  host1(config-if)#**ppp description pah8999**

Use the **no** version to remove the description.

Related Documentation

- *ppp description*

Configuring the IPCP Lockout Option for Each PPP Interface

You can optionally terminate invalid IPv4 subscribers and prevent additional IPCP negotiations. The subscriber can negotiate IPv4 addresses, IPv6 addresses, or both. When Internet Protocol version 6 Control Protocol (IPv6CP) is active, this command enables unused IPv4 addresses, which are allocated for the IPv6 subscribers, to be available for the IPCP services for an internally defined time interval (10 seconds). When the time interval elapses, the subscriber must connect again to negotiate IPCP. You can terminate the invalid IPv4 subscribers from the Interface Configuration mode, the Subinterface Configuration Mode, and the Profile Configuration Mode.

To terminate invalid IPv4 subscribers and configure the IPCP lockout option for each PPP interface:

- (Optional) From the Subinterface Configuration mode, enable the IPCP lockout option:

  host1(config-subif)#**ppp ipcp lockout**

  Use the **no** version to disable the IPCP lockout option on the interface.

Related Documentation

- IPCP Lockout and Local IP Address Pool Restoration Overview on page 272
- *ppp ipcp lockout*

Configuring the IPCP Netmask Option for Each PPP Interface

You can optionally enable the IPCP netmask option (option 0x90) on a per-PPP interface basis explicitly, either in a profile or on a static interface. By default, the IPCP option 0x90 is disabled on the interface. The IPCP netmask option is a nonstandard option that enables a peer to request the netmask associated with the assigned IP address. The netmask can be specified via RADIUS attribute 9, Framed-Ip-Netmask. If the netmask is 255.255.255.255, the option is not negotiated. You can enable the IPCP netmask option from the Interface Configuration Mode, Subinterface Configuration mode, and Profile Configuration Mode.

To enable the IPCP netmask option for each PPP interface:

- (Optional) From the SubInterface Configuration mode, enable the IPCP lockout option:

  host1(config-subif)#**ppp ipcp netmask**

  Use the **no** version to disable the IPCP netmask option on the interface.

Related Documentation

- *ppp ipcp netmask*
- *radius ignore*

Configuring the KeepAlive Timeout for an Interface

You can optionally specify a keepalive timeout value. The keepalive mechanism tracks the status of the connection. There are two keepalive modes of operation: high-density mode and low-density mode. High-density mode is automatically selected when PPP is layered over ATM, tunnel, or PPPoE. Low-density mode is selected when PPP is layered over HDLC. The keepalive mode selection is made per interface.

In high-density mode, also known as smart keepalive, when the keepalive timer expires, the interface first verifies whether any frames were received from the peer in the prior keepalive timeout interval. If so, the interface does not send an LCP echo request (keepalive). Keepalive packets are sent only if the peer is silent (that is, no traffic was received from the peer during the previous keepalive timeout interval). If both sides are configured with keepalive, receipt of an LCP echo request by one end suppresses the transmission of an LCP echo request by that end. Smart keepalive is disabled when the keepalive timeout value is at least 60 seconds, even when in high-density mode. Smart keepalive is always disabled when in low-density mode. This mode suppresses transmission of unnecessary LCP echo requests. For high-density keepalive mode, the range is 30–64800 seconds. The default value is 30 seconds.

In low-density mode, when the keepalive timer expires, the interface *always* sends an LCP echo request, regardless of whether the peer is silent. For low-density keepalive mode, the range is 1–64800 seconds for POS uplink interfaces, and 10–64800 seconds for all other HDLC interfaces. The default value for all interfaces is 30 seconds.

If the keepalive interval is 30 seconds, a failed link is detected between 90 and 120 seconds after failure. To restore the default of 30 seconds, enter **ppp keepalive** without a value.

To specify the keepalive value for each interface:

- (Optional) From the Interface Configuration mode, specify the keepalive value:

  host1(config-if)#**ppp keepalive 50**

  Use the **no** version to disable keepalive.

Related Documentation

- *ppp keepalive*

Disabling the Negotiation of the Local Magic Number

You can optionally disable negotiation of the local magic number. By doing so, you can prevent the router from detecting loopback configurations. You can disable negotiation from the Interface Configuration Mode, the Subinterface Configuration mode, and the Profile Configuration mode.

To disable the negotiation of the local magic number:

- (Optional) From the Interface Configuration mode, specify the following command:

host1(config-if)#**ppp magic-number disable**

Use the **no** version to restore negotiation of the local magic number.

Related Documentation

- *ppp magic-number disable*

Configuring the Router to Ignore a Mismatch of the Peer Magic Number

You can optionally configure the router to ignore a mismatch of the Link Control Protocol (LCP) peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number. You can configure the router to ignore a mismatch from the Interface Configuration Mode, the Subinterface Configuration mode, and the Profile Configuration mode.

To configure the router to ignore a mismatch of the peer magic number:

- (Optional) From the Interface Configuration mode, specify the following command:

  host1(config-if)#**ppp magic-number ignore-mismatch**

  Use the **no** version to restore the default behavior, in which the router terminates the PPP connection if it detects an LCP peer magic number mismatch.

Related Documentation

- Understanding PPP Link Control Protocol on page 263
- *ppp magic-number ignore-mismatch*

Configuring the Maximum Number of Renegotiation Attempts from a PPP Client

You can optionally configure the maximum number of renegotiation attempts from LCP, IPCP, and IPv6CP before terminating a session. Configuring the maximum number of renegotiation attempts helps prevent massive renegotiation loops between the router and a noncompliant PPP client. Such renegotiation loops can cause excessive CPU utilization and can prevent the PPP client from coming up properly.

When a PPP client exceeds the configured maximum number of renegotiation attempts, the router sends a termination request to end the PPP session. When the PPP session is terminated and LCP goes into a stopped (closed) state, static PPP or MLPPP interfaces go into passive mode and wait for the other side of the connection to start the LCP negotiation process.

When both IPv4 interface columns and IPv6 interface columns are configured over a PPP link-layer interface, the router terminates the PPP session only when the PPP client exceeds the configured maximum number of renegotiation attempts for both the IPv4 interface and the IPv6 interface.

You can configure the maximum number of renegotiation attempts from the Interface Configuration Mode, the Subinterface Configuration mode, and the Profile Configuration mode.

NOTE: If you do not specify the optional **lcp**, **ipcp**, or **ipv6cp** keyword, the **ppp max-negotiations** command sets the maximum number of renegotiation attempts for each of LCP, IPCP, and IPv6CP to the value you specify, or to the default value (30) if you omit the optional value for maximum renegotiation attempts.

To configure the maximum number of renegotiation attempts for LCP, IPCP, and IPv6CP:

- (Optional) From the Interface Configuration mode, specify the following command:

  host1(config-if)#**ppp max-negotiations 15**

  Use the **no** version to restore the default value, 30 renegotiation attempts for LCP, IPCP, and IPv6CP.

Related Documentation

- *ppp max-negotiations*

Configuring the Maximum Receive Unit for PPP

You can optionally configure the maximum receive unit (MRU) for PPP or MLPPP interfaces. You can configure the maximum number of renegotiation attempts from the Interface Configuration Mode, the Subinterface Configuration mode, and the Profile Configuration mode.

NOTE: We recommend you coordinate this value with the network administrator on the other end of the line.

If the value configured for the PPP MRU is greater than the value of the lower-layer MRU minus the PPP header length, the router logs a warning message and uses the lesser of the configured MRU value or the lower-layer MRU value minus the PPP header length to negotiate the local MRU. If the value configured for the PPP MRU conflicts with a similar value configured for another protocol, such as the MTU value for PPPoE, the router uses the lesser of the two values.

To configure the maximum receive unit for PPP or MLPPP:

- (Optional) From the Interface Configuration mode, specify the following command:

  host1(config-if)#**ppp mru 576**

  Use the **no** version to restore the default value, which causes PPP to use the lower-layer MRU minus the PPP header length as the MRU value.

Related Documentation

- *ppp mru*

Forcing the PPP Interface into a Passive Mode

You can optionally force a static or dynamic PPP interface into passive mode before LCP negotiation begins, for a period of one second. This delay enables slow clients to start up and initiate the LCP negotiation. You can force the PPP interface into a passive mode from the Interface Configuration Mode, the Subinterface Configuration mode, and the Profile Configuration mode.

To force the PPP interface into a passive mode:

- (Optional) From the Interface Configuration mode, specify the following command:

  host1(config-if)#**ppp passive-mode**

  Use the **no** version to disable passive mode.

Related Documentation

- *ppp passive-mode*

Configuring the PPP Peer to Take Precedence of DNS and WINS Addresses

You can optionally configure the PPP peer to take precedence of DNS and WINS addresses. By default, the DNS and WINS addresses configured on the router take precedence. By configuring the peer to take precedence, you can resolve conflicts when the router and the PPP peer have the primary and secondary DNS and WINS name server addresses configured with different values. You can configure the PPP peer to take precedence from the Interface Configuration Mode, the Subinterface Configuration mode, and the Profile Configuration mode.

NOTE: Use the **dns** keyword or the **wins** keyword to configure which PPP peer address takes precedence. This command has no effect unless both routers have the address configured and the address is in conflict. If the PPP peer has the address and the router does not, the peer always supplies the address regardless of how you have configured the PPP peer.

To configure the PPP peer to take precedence of DNS and WINS addresses:

- (Optional) From the Interface Configuration mode, specify the following command:

  host1(config-if)#**ppp peer dns**

  Use the **no** version when you want the router to take precedence during setup negotiations between the router and the peer. If the IP addresses that the peer sends to the router differ from the ones configured on your router, the router returns the values that you configured as the correct values to the peer.

Related Documentation

- *ppp peer*

Configuring the PPP IP Address as Optional in an IPCP Request

You can optionally allow the IPCP negotiation to succeed even though the peer does not include the IP address option in an IPCP configuration request. By default, the command is disabled. You can configure the PPP IP address as optional from the Global Configuration mode.

To configure the PPP IP address as optional in an IPCP Configuration request:

- (Optional) From the Global Configuration mode, specify the following command:

  host1(config)#**ppp peer-ip-address-optional**

  Use the **no** version to restore the default behavior.

Related Documentation

- *ppp peer-ip-address-optional*

Terminating the PPP Session

You can optionally terminate the PPP session. By default, all PPP sessions are enabled. You can stop the PPP session from the Interface Configuration Mode and the Subinterface Configuration Mode.

NOTE: You can specify the optional **ip** keyword with the **ppp shutdown** command, to administratively disable the IPCP service. Also, you can use the optional **ipv6cp** keyword, or **mpls** keyword, or **osi** keyword, with the **ppp shutdown** command to administratively disable the IPv6CP service, or the MPLS service, or the OSINLCP service, respectively.

To stop the PPP session:

- (Optional) From the Interface Configuration mode, specify the following command:

  host1(config-if)#**ppp shutdown**

  Use the **no** version to restart a disabled session.

Related Documentation

- *ppp shutdown*

Configuring PPP Authentication

You can optionally configure authentication tasks on a PPP interface such as requesting for authentication from a peer, specifying the authentication type, or the challenge length for CHAP authentication.

NOTE: The JunosE Software's PPP application accepts null usernames during PAP and CHAP authentication. When the PPP application receives an authentication request that includes a null username, PPP passes the request to AAA. To take advantage of this feature, configure your authentication server to support the use of null usernames.

To optionally configure authentication on a PPP peer, you can:

- Request authentication from a peer and set the authentication method.

  See "Requesting Authentication from a PPP Peer" on page 292
- Modify the CHAP challenge length.

  See "Modifying the Challenge Length for CHAP Authentication" on page 295
- Specify the maximum number of retries for PAP and CHAP.

  See "Configuring the Maximum Retries for PAP and CHAP Authentication" on page 295

Related Documentation

- *ppp authentication*
- *ppp chap-challenge-length*
- *ppp max-bad-auth*

Requesting Authentication from a PPP Peer

You can optionally request for authentication from a PPP peer and set the authentication method. The order of preference of the authentication protocol depends on the order in which you specify the authentication protocol in the command line. If the peer refuses the first authentication protocol, the router requests the second authentication protocol. If the peer refuses to negotiate authentication, the router terminates the PPP session.

You can also specify the authentication virtual router context. You can request for authentication from a PPP peer from the Interface Configuration Mode, the Subinterface Configuration Mode and the Profile Configuration Mode.

NOTE: When you specify a VR in the **ppp authentication** command, AAA does not query the domain map for the assigned VR context. Instead, AAA uses the VR specified in the **ppp authentication** command as the authentication VR context and issues the authentication request to the authentication server in the assigned VR context. If you specify the default VR as the authentication VR context, AAA loosely binds the user to the default VR. This means that RADIUS *can override* the default VR context with a new VR context during the authentication process. When the **ppp authentication virtual-router** command specifies the default VR, AAA returns either the default VR or the VR specified by RADIUS. If you specify a VR other than the default VR as the authentication VR, AAA tightly binds the user to the specified VR. This means that RADIUS *cannot override* the specified VR context with a new VR context during the authentication process. When the **ppp authentication virtual-router** command specifies a nondefault VR, AAA returns the specified VR.

To specify the order of preference for the primary authentication protocol:

- (Optional) From the Interface Configuration mode, specify the following command:

  host1(config-if)#**ppp authentication pap chap eap**

  Use the **no** version to specify that the router does not require authentication.

  The router requests the use of PAP as the authentication protocol (because it appears first in the command line). If the peer refuses to use PAP, the router requests the CHAP protocol. If the peer refuses to use CHAP, the router requests the EAP protocol. If the peer refuses to negotiate authentication, the router terminates the PPP session.

To specify a virtual router for the authentication virtual router context:

- (Optional) From the Interface Configuration mode, specify the following command:

  host1(config-if)#**ppp authentication virtual-router boston pap chap**

  Use the **no** version to specify that the router does not require authentication.

  This command is available in static configurations and in profiles.

To configure EAP as the only authentication protocol on a static PPP interface:

- (Optional) From the Global Configuration mode, specify the following command:

  host1(config)#**interface atm 3/2.100**

- (Optional) From the Subinterface Configuration mode, specify the following command:

  host1(config-subif)#**ppp authentication eap**

  Use the **no** version to specify that the router does not require authentication.

To configure EAP as the only authentication protocol on a dynamic PPP interface:

- (Optional) From the Global Configuration mode, specify the following command:

host1(config)#**profile ppptest**

- (Optional) From the Profile Configuration mode, specify the following command:

  host1(config-profile)#**ppp authentication eap**

  Use the **no** version to specify that the router does not require authentication.

Related Documentation

- *interface*
- *ppp authentication*
- *profile*

Configuring IPCP Renegotiations in a Dual-Stack Network for Optimal Utilization of Released IPv4 Addresses

You can enable the PPP application to inform the RADIUS server about released IPv4 addresses for dual-stack subscribers immediately after the address is released. You can also specify a time period during which IPCP renegotiations for IPv4 addresses that the router or the PE device receives from a subscriber is restricted. You can specify the maximum number of requests for IPv4 addresses that can be received per subscriber during the time interval configured for IPCP renegotiations to be received.

To configure IPCP renegotiations in a dual-stack network:

1. From Global Configuration mode, enable the PPP application to inform the RADIUS server about released IPv4 addresses.

   host1(config)#**aaa ipv4-addr-saving** ***address***

2. From Interface Configuration mode, configure the maximum number of successful IPCP renegotiations for IPv4 addresses that the router can receive per subscriber. You can specify this value in a PPP profile or a PPP interface.

   host1(config-if)#**ppp ipcp-max-negotiation 3**

3. Configure the interval during which the maximum number of IPCP renegotiations for IPv4 addresses that the router receives from a subscriber must be restricted. You can configure this setting in a PPP profile or a PPP interface.

   host1(config-if)#**ppp ipcp-nego-duration 400**

4. Configure the interval during which additional IPCP negotiations are prevented from being received. When the time interval elapses, the subscriber must connect again to negotiate IPCP. You can configure the interval from Interface Configuration mode, Subinterface Configuration Mode, and Profile Configuration Mode.

   host1(config-if)#**ppp ipcp-lockout-duration 400**

Related Documentation

- IPCP Renegotiation of IPv4 Addresses for Dual-Stack Subscribers on page 277
- Overview of Processing IPCP Negotiations for Dual-Stack Subscribers on page 275
- Releasing IPv4 Addresses During Termination of PPP Sessions on page 277
- Monitoring AAA IPv4 Address Saving on page 311

- *aaa ipv4-addr-saving*
- *ppp ipcp-lockout-duration*
- *ppp ipcp-max-negotiation*
- *ppp ipcp-nego-duration*

Modifying the Challenge Length for CHAP Authentication

You can optionally modify the length of the CHAP challenge by specifying the allowable minimum length and maximum length in bytes in the range 8-63. By default, all PPP sessions are enabled. You can stop the PPP session from the Interface Configuration Mode and the Subinterface Configuration Mode.

CAUTION: **Do *not* decrease the range. Increasing the range is acceptable, provided that you do not lower the minimum to do so. The recommended minimum is 16. A longer challenge and a more unpredictable challenge length provide a higher level of security.**

To modify the maximum and minimum challenge length for CHAP authentication:

- (Optional) From the Interface Configuration mode, specify the following command:

  host1(config-if)#**ppp chap-challenge-length 24 28**

  Use the **no** version to restore the default minimum (16 bytes) and default maximum (32 bytes).

Related Documentation

- *ppp chap-challenge-length*

Configuring the Maximum Retries for PAP and CHAP Authentication

You can optionally specify the maximum number of authentication retries the router allows before terminating the PPP session. By default, no retries are allowed for PAP and CHAP authentication. You can specify a value between 0 to 7. You can specify the value for maximum number of authentication retries from the Interface Configuration Mode and the Subinterface Configuration Mode.

To specify the maximum number of authentication retries:

- (Optional) From the Interface Configuration mode, specify the following command:

  host1(config-if)#**ppp max-bad-auth 3**

  Use the **no** version to return the number of retries to the default, 0.

Related Documentation

- *ppp max-bad-auth*

PPP Accounting Statistics Overview

The JunosE Software begins the collection of accounting statistics for terminated PPP sessions following, but not including, authentication acknowledgement from the E Series router. The acknowledgment is either a CHAP success or PAP acknowledgement packet. All subsequent traffic is counted up the point that PPP at the router terminates the subscriber's session. The statistics are reported in the following RADIUS attributes:

Attribute Number	Attribute Name
42	Acct-Input-Octets
43	Acct-Output-Octets
47	Acct-Input-Packets
48	Acct-Output-Packets

PPP session termination can be initiated through a number of mechanisms: PPP shutdown at the client or router interface, subscriber logout at the router (by means of the **logout subscriber** command), lower layer down events, and silent client termination.

The following rules apply to all termination scenarios:

- Accounting statistics reported in RADIUS octet counts (Acct-Input-Octets and Acct-Output-Octets) for terminated PPP customers include the following data:
  - All upper layer control traffic, including IPCP, IPCPv6, OSICP, and MPLSNCP
  - All data traffic, including IP, IPv6, MPLS, and OSI
  - All PPP LCP echo requests and responses following authentication
  - Other PPP LCP packets following the PAP or CHAP acknowledgment
  - Retransmits of the PAP or CHAP traffic
- PPP accounting statistics reported in RADIUS octet counts (Acct-Input-Octets and Acct-Output-Octets) exclude the following data:
  - PPP traffic prior to completion of authentication
  - PPP LCP terminate-request or terminate-acknowledgement packets
  - PPPoE padding for PPP control and data packets
- Accounting statistics reported in RADIUS packet counts (Acct-Input-Packets and Acct-Output-Packets) for terminated PPP customers are based on packets delivered to or received from the upper transport layer: IP, IPv6, MPLS, and OSI.

For information on accounting statistics for tunneled PPP sessions, see *PPP Accounting Statistics*.

Related Documentation

- *PPP Accounting Statistics*
- *logout subscribers*

Setting a Baseline for PPP Interface Statistics

You can set a statistics baseline for PPP interfaces. To view baseline statistics, use the **delta** keyword with the PPP **show** commands.

NOTE: The router implements the baseline by reading and storing the statistics at the time the baseline is set, then subtracting this baseline whenever baseline-related statistics are retrieved.

To set a baseline for PPP statistics:

- Issue the **baseline ppp interface** command for an interface:

  host1#**baseline ppp interface atm 3/0/3.20**

There is no **no** version.

Related Documentation

- *baseline ppp interface*

Monitoring PPP Interfaces

Purpose Display information about the PPP interfaces, selectively. You can also filter the display for specific information such as interface type, data type, configured protocol, or interface state.

Action To display detailed information for a specific IP interface:

```
host1#show ppp interface gig 2/1/7.1.1 full
PPP interface GigabitEthernet 2/1/7.1.1 is up
Interface administrative status is open
Configured network protocol is IPCP
IPCP protocol configuration
   configured                     true
   administrative-status          open
   ip-address                     6.6.6.6
   dns-precedence                 local
   wins-precedence                local
   ipcp-netmask-option            disabled
   ipcp-lockout            disabled
   max-negotiations            30
   ipcp prompt-option dns         enabled
IPCP protocol status
   operational-status             up
IPCP negotiated options           local               peer
   ip-address                     11.1.1.1           11.1.1.2
   ip-address-mask                none                none
   primary-dns-address            none                none
   secondary-dns-address          none                none
   primary-wins-address           none                none
```



```
   secondary-wins-address         none              none
IPV6CP Protocol Configuration
   configured                     false
   administrative-status          open
   ipv6-interfaceId                     0.0.0.0
   max-negotiations            30
IPV6CP protocol status
   operational-status             not present
OSINLCP protocol configuration
   configured                     false
   administrative-status          open
OSINLCP protocol status
   operational-status             not present
  Interface statistics              in                 out
   packets                        0                    0
   octets                         14515                11296
   errors                         0                    0
   discards                       0                    0
   policed                        0                    0
LCP protocol configuration
   max-receive-unit               use lower layer
   authentication
   magic-number                   enabled
   magic-number-mismatch          reject
   keepalive-timer                30 seconds
   restart-timer                  3 seconds
   max-terminate                  2
   max-configure                  10
   max-failure                    5
   passive-mode                   disabled
LCP protocol status
   link-status                    network
LCP negotiated options            local             peer
   max-receive-unit               1492              1492
   authentication                 none              chap
   magic-number                   0x3e51ca08        0x740bbf81
   accm                           none              none
   pfc                            none              none
   acfc                           none              none
LCP protocol statistics
   in-keepalive-requests          11
   out-keepalive-requests         11
   in-keepalive-replies           11
   out-keepalive-replies          11
   keepalive-failures             0
   max-renegotiation terminates   0
IP protocol statistics
   max-renegotiation-terminates   0
IPv6 protocol statistics
   max-renegotiation-terminates   0
Authentication configuration
   authenticate-retry             0
   authentication-router          ''
   aaa-profile          ''
Authentication status
   grant                          true
   session-timeout                none
   inactivity-timeout             none
   monitor—ingress-only           false
   accounting-timeout             none
   peer-ip-address                none
```



```
   peer-ip-address-mask           none
   peer-primary-dns-address       none
   peer-secondary-dns-address     none
   peer-primary-wins-address      none
   peer-secondary-wins-address    none
   peer-ipv6-interface-id         none
Authentication statistics
   up-time                        0 seconds
   in-octets                      0
   out-octets                     0
   in-packets                     0
   out-packets                    0
PAP protocol configuration
   request-timeout                20 seconds
CHAP protocol configuration
   name                           ''
   challenge-retry                10
   challenge-timeout              4 seconds
   minimum-challenge-length       16
   maximum-challenge-length       32
   minimum-rechallenge-timeout    0 seconds
   maximum-rechallenge-timeout    0 seconds
EAP Protocol Configuration
   request-retry                5
   request-timeout              3 seconds
```

To display detailed information for a specific IPv6 interface:

```
host1#show ppp interface fastEthernet 12/0.1.1 full
PPP interface FastEthernet 12/0.1.1 is lowerDown

Interface administrative status is open
Configured network protocol is IPV6CP
IPCP protocol configuration
   configured                     false
   administrative-status          open
   ip-address                     0.0.0.0
   dns-precedence                 local
   wins-precedence                local
   ipcp-netmask-option            disabled
   ipcp-lockout-option            enabled
   max-negotiations               10
IPCP protocol status
   operational-status             not present
IPV6CP protocol configuration
   configured                     true
   administrative-status          open
   ipv6-interfaceId               90:1a00:140:4b39
IPV6CP protocol status
   operational-status             down
   terminate-reason               link down

OSINLCP protocol configuration
   configured                     false
   administrative-status          open
OSINLCP protocol status
   operational-status             not present
Interface statistics              in                    out
   packets                        0                     0
   octets                         1163                  706
   errors                         0                     0
   discards                       153482                0
```

```
LCP protocol configuration
   max-receive-unit               use lower layer
   authentication                 none
   magic-number                   enabled
   magic-number-mismatch          reject
   keepalive-timer                30 seconds
   restart-timer                  3 seconds
   max-terminate                  2
   max-configure                  10
   max-failure                    5
   passive-mode                   disabled
LCP protocol status
   link-status                    initial
LCP protocol statistics
   in-keepalive-requests          11
   out-keepalive-requests         11
   in-keepalive-replies           11
   out-keepalive-replies          11
   keepalive-failures             0
Authentication configuration
   authenticate-retry             0
   authentication-router          ''
   aaa-profile                    ''
Authentication status
   grant                          false
   terminate-reason               lower layer down
PAP protocol configuration
   request-timeout                20 seconds
CHAP protocol configuration
   name                           ''
   challenge-retry                10
   challenge-timeout              4 seconds
   minimum-challenge-length       16
   maximum-challenge-length       32
   minimum-rechallenge-timeout    0 seconds
   maximum-rechallenge-timeout    0 seconds
```

To display the reason for termination when the operational status of the interface is down:

```
host1#show ppp interface
PPP interface pos 0/1:1 is lowerDown
PPP interface pos 4/0:1 is lowerDown
PPP interface pos 12/1:1 is lowerDown
3 ppp interfaces found
PPP interface serial 0/0:1/1 is Up
PPP interface serial 0/0:1/2 is Down (administrative disable)
```

To display information about the PPP interface statistics for IPCP:

```
host1#show ppp interface fastEthernet 2/0.1 statistics ip
PPP interface FastEthernet 2/0.1 is up
LCP protocol statistics
   in-keepalive-requests          833
   out-keepalive-requests         830
   in-keepalive-replies           830
   out-keepalive-replies          833
   keepalive-failures             0
   max-renegotiation-terminates   10
```

Meaning Table 22 on page 301 lists the **show ppp interface** command output fields.

Table 22: show ppp interface Output Fields

Field Name	Field Description
inactivity-timeout	Inactivity timeout, in seconds; session is terminated if it is not active for specified timeout.
monitor-ingress-only	Indicates whether the router monitors only ingress traffic for the configured idle (inactivity) timeout period; true (router monitors only ingress traffic) or false (router monitors both ingress traffic and egress traffic).
PPP interface	Interface type, interface specifier, and status (up, down, lowerDown, not present, passive, or tunnel). For more information about specifying the physical interface on which you want to configure PPP, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.
accounting-timeout	Accounting timeout in seconds; frequency of accounting updates to the authentication server
Interface alias	Alias or description of the PPP interface
Interface administrative status	Indicates whether the interface is administratively enabled (open), which means that the **no ppp shutdown** command is operational; or administratively disabled (closed), which means that the **ppp shutdown** command is operational.
Configured network protocol	Indicates the network protocol configured on the interface
Baseline status	Indicates whether a statistics baseline is set.
Interface statistics	PPP interface statistics: • packets—Number of packets received (in) or transmitted (out) on the interface • octets—Number of octets received (in) or transmitted (out) on the interface • errors—Number of errors received (in) or transmitted (out) on the interface • discards—Number of packets discarded on receipt (in) or discarded before they were transmitted (out); for more information about the discards counter, see “Rate Limiting for PPP Control Packets” on page 266 • policed—Number of packets received (in) and marked as per the rates configured for the interface; for more information, see “Rate Limiting for PPP Control Packets” on page 266

Table 22: show ppp interface Output Fields *(continued)*

Field Name	Field Description
IPCP protocol configuration	PPP IPCP configuration information: • configured—IPCP is configured on this interface (true or false) • administrative-status—IPCP administrative status (open or closed) • ip-address—Address to be used for negotiation of the local IP address option • dns-precedence—Used to resolve conflicts during negotiation of DNS addresses; “local” indicates that the local side takes precedence and the **no ppp peer dns** command is operative; “peer” indicates that the remote side takes precedence and the **ppp peer dns** command is operative • wins-precedence—Used to resolve conflicts during negotiation of WINS addresses; “local” indicates that the local side takes precedence and the **no ppp peer wins** command is operative; “peer” indicates that the remote side takes precedence and the **ppp peer wins** command is operative • ipcp-netmask-option—Controls negotiation of the IPCP netmask option; disabled = do not negotiate, enabled = negotiate • ipcp-lockout-option—Terminates an invalid subscriber entry and prevents additional IPCP negotiations (enabled or disabled) • max-negotiations—Maximum number of renegotiation attempts that the router accepts before terminating a PPP session • ipcp prompt-option dns—Prompts the CPE (Customer Premises Equipment) to negotiate the IPCP primary and secondary DNS options that are locally available with the broadband remote access server (enabled or disabled)
IPv6CP protocol configuration	PPP IPv6 configuration information: • configured—IPv6CP is configured on this interface (true or false) • administrative-status—IPv6CP administrative status (open or closed) • ipv6-interfaceId—Address to be used for negotiation of local IPv6 address option
IPCP protocol status	IPCP status: • operational-status—IPCP operational status (up, down, not present, or not present no resources) • terminate-reason—Reason for termination of IPCP service

Table 22: show ppp interface Output Fields *(continued)*

Field Name	Field Description
IPv6CP protocol status	IPCPv6 status: • operational-status—IPv6CP operational status (up, down, not present, or not present no resources) • terminate-reason—Reason for termination of IPv6CP service
IPCP negotiated options	Shows the following negotiated addresses for the local and remote (peer) side of the link: • ip-address—IP address • ip-address-mask—IP address mask • primary-dns-address—Primary DNS address • secondary-dns-address—Secondary DNS address • primary-wins-address—Primary WINS address • secondary-wins-address—Secondary WINS address NOTE: The command displays a value of “none” for any negotiated option parameters if the option was not negotiated.
OSINLCP protocol configuration	OSINLCP configuration information: • configured—OSINLCP is configured on this interface (true or false) • administrative-status—OSINLCP administrative status (open or closed)
OSINLCP protocol status	OSINLCP status: • operational-status—OSINLCP operational status (up, down, not present, or not present no resources) • terminate-reason—Reason for termination of OSINLCP service
OSINLCP negotiated options	OSINLCP negotiated options: • npdu-alignment—Negotiated NPDU alignment for the local and remote (peer) side of the link NOTE: The command displays a value of “none” for any negotiated option parameters if the option was not negotiated.
MPLSNLCP protocol configuration	MPLSNLCP configuration information: • configured—MPLSNLCP is configured on this interface (true or false) • administrative-status—MPLSNLCP administrative status (open or closed)

Table 22: show ppp interface Output Fields *(continued)*

Field Name	Field Description
MPLSNLCP protocol status	MPLSNLCP protocol status • operational-status—MPLSNLCP operational status (up, down, not present, or not present no resources) • terminate-reason—Reason for termination of MPLSNLCP service
MPLSNLCP negotiated options	MPLSNLCP negotiated options: • npdu-alignment—Negotiated NPDU alignment for the local and remote (peer) side of the link NOTE: The command displays a value of “none” for any negotiated option parameters if the option was not negotiated.
LCP protocol configuration	• max-receive-unit—Controls negotiation of the local MRU option; “use lower layer” indicates that the MRU of the layer below PPP defines the MRU to be negotiated; “disabled” indicates that the MRU option is not to be negotiated. A numeric value indicates the MRU value to be negotiated • authentication—Controls the negotiation of the local authentication option; “none” indicates do not negotiate; “chap” indicates negotiate chap; “pap” indicates negotiate pap; “chap/pap” indicates negotiate chap and, if it is rejected, negotiate pap; “pap/chap” indicates negotiate pap and, if it is rejected, negotiate chap. • magic-number—Controls whether the local magic number is negotiated: enabled (negotiate), or disabled (do not negotiate) • magic-number-mismatch—Indicates whether the router is configured to ignore the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number: ignore (ignore the peer magic number mismatch and retain the PPP connection), or reject (router terminates the PPP connection if it detects a peer magic number mismatch) • keepalive-timer—Rate of LCP echo requests • restart-timer—Retry frequency during LCP, IPCP, OSINLCP, and MPLS negotiations • max-terminate—Maximum number of terminate requests • max-configure—Maximum number of configure requests • max-failure—Maximum number of configure NAKs • passive-mode—Forces a PPP interface into a passive mode before LCP negotiation begins; “disabled” means do not wait for peer; “enabled” means wait for peer to initiate negotiation

Table 22: show ppp interface Output Fields *(continued)*

Field Name	Field Description
LCP protocol status	• link-status—Overall status of LCP negotiations, including the following states: Initial (idle), Starting (ready to negotiate), Authenticate (authenticating), and Network (LCP is up)
LCP negotiated options	Shows the following negotiated values for the local and remote (peer) side of the link: • max-receive unit—Maximum receive unit, in octets • authentication—Authentication method (none, pap, or chap) • magic-number—Magic number • pfc—PFC (none or enabled) • acfc—ACFC (none or enabled) NOTE: The command displays a value of “none” for any negotiated option parameters if the option was not negotiated.
LCP Endpoint Discriminator options	• local discriminator class—Endpoint discriminator type, format, and address space for the local and remote (peer) router • local endpoint discriminator—Endpoint discriminator value for the local router within the specified class • peer discriminator class—Endpoint discriminator type, format, and address space for the remote router • peer endpoint discriminator—Endpoint discriminator value for the remote router within the specified class
LCP protocol statistics	Shows the following statistics for the life of the interface (since system boot or interface creation, whichever is later) : • in-keepalive-requests—Number of received keepalive requests (LCP Echo Requests) • out-keepalive-requests—Number of transmitted keepalive requests • in-keepalive-replies—Number of received keepalive replies • out-keepalive-replies—Number of transmitted keepalive replies • keepalive-failures—Number of keepalive failures reported on the interface • max-renegotiation-terminates—Number of renegotiation terminations for the PPP session

Table 22: show ppp interface Output Fields *(continued)*

Field Name	Field Description
IP protocol statistics	• max-renegotiation-terminates—Number of times the link has been terminated since it was created, due to the peer exceeding the maximum renegotiation attempts.
IPv6 protocol statistics	• max-renegotiation-terminates—Number of times the link has been terminated since it was created, due to the peer exceeding the maximum renegotiation attempts.
Authentication configuration	• authenticate-retry—Maximum number of authentication retries configured using the **ppp max-bad-auth** command • authentication-router—Virtual router for the authentication virtual router context
Authentication status	• grant—Authentication status (true = access granted, false = access not granted) • session-timeout—Session timeout, in seconds; session is terminated at expiration • inactivity-timeout—Inactivity timeout, in seconds; session is terminated if it is not active for specified timeout • accounting-timeout—Accounting timeout in seconds; frequency of accounting updates to the authentication server • peer-ip-address—IP address to be used in negotiation of peer IP address • peer-ip-address-mask—IP address mask to be used in negotiation of the peer IP address mask • peer-primary-dns-address—IP address to be used in negotiation of the peer primary DNS address • peer-secondary-dns-address—IP address to be used in negotiation of the peer secondary DNS address • peer-primary-wins-address—IP address to be used in negotiation of the peer primary WINS address • peer-secondary-wins-address—IP address to be used in negotiation of the peer secondary WINS address NOTE: The command displays the authentication status as “none” for any parameters not provided by the authentication server. • peer-ipv6–interface-id—IPv6 interface Identifier of the peer returned in an authentication grant from the RADIUS server

Table 22: show ppp interface Output Fields *(continued)*

Field Name	Field Description
Authentication statistics	Shows statistics accumulated since the session was established • up-time—Time the session has been up, in seconds • in-octets—Number of octets received on the interface • out-octets—Number of octets transmitted out the interface • in-packets—Number of packets received on the interface • out-packets—Number of packets transmitted out the interface
PAP protocol configuration	• request-timeout—Maximum time, in seconds, to wait for an authentication request packet
CHAP protocol configuration	• name—Name to be used in challenge packets • challenge-retry—Maximum number of challenge packets to be transmitted • challenge-timeout—Frequency, in seconds, of challenge packet retransmission • minimum-challenge-length—Minimum length of challenge packet • maximum-challenge-length—Maximum length of challenge packet; the size of the challenge used for each challenge packet is a random number between minimum-challenge-length and maximum-challenge-length • minimum-rechallenge-timeout—Minimum time, in seconds, before initiating a rechallenge to peer • maximum-rechallenge-timeout—Maximum time, in seconds, before initiating a rechallenge to peer; the actual time before a rechallenge is a random number between minimum-rechallenge-timeout and maximum-rechallenge-timeout
EAP protocol configuration	• request-retry—Maximum number of authentication requests retried before returning a deny • request-timeout—Maximum time, in seconds, to wait for an authentication request packet

Table 22: show ppp interface Output Fields *(continued)*

Field Name	Field Description
Operational Status	If the operational status is down for a specific interface, one of the following termination reasons may appear within parentheses: • administrative disable—Interface has been administratively disabled, which means that the **ppp shutdown** command is in effect. This applies to an interface, IPCP, IPv6CP, OSINLCP, and MPLS. • administrative logout—Interface has been administratively logged out, which means that the **logout subscriber** command has been issued. This applies to an interface only. • no upper interface—No upper layer is configured. This applies to an interface only. • authentication failure—Authentication is required and has failed. This applies to an interface only. • no local xxx—local option, *xxx*, is required and could not be negotiated (for example, IP address). This applies to an interface, IPCP, IPv6CP, OSINLCP, and MPLS. • no peer xxx—Remote peer option, xxx, is required and could not be negotiated (for example, authentication). This applies to an interface, IPCP, IPv6CP, OSINLCP, and MPLS. • keepalive drop count exceeded—Keepalive drop count has been exceeded. This applies to an interface only. • session timeout—Session timeout period has expired. This applies to an interface only. • inactivity timeout—Inactivity timeout period has expired. This applies to an interface only. • address lease expired—Address lease period has expired. This applies to an interface only. • not configured—Protocol is not configured on the interface. This applies to IPCP, IPv6CP, OSINLCP, and MPLS. • link down—Link is down and the protocol is not operationally up. This applies to IPCP, IPv6CP, OSINLCP, and MPLS. • lower layer down—Lower protocol layer is down. This applies to an interface only. • max configure exceeded—Maximum number of configure requests was exceeded while negotiations were in progress. This means that there was no response from the peer, or the peer refused to negotiate. This applies to an interface, IPCP, IPv6CP, OSINLCP, and MPLS. • peer requested termination—Remote peer requested termination of the connection, which means that a terminate request was received while the session was in an open state. This applies to an interface, IPCP, IPv6CP, OSINLCP, and MPLS.

Related Documentation

- *show ppp interface*

Monitoring Multilinked and Nonmultilinked PPP Interfaces

Purpose Display a summary of all the multilinked and nonmultilinked PPP interfaces configured on the router.

Action To display a summary of all the PPP interfaces configured on the router:

```
host1#show ppp interface summary
PPP Status
Configuration status      configured notConfigured
   Interface              4000       n/a
   Ip                     4000       0
   Ipv6                   0          4000
   Osi                    0          4000
   Mpls                   0          4000
Administrative status     open       closed
   Interface              4000       0
   Ip                     4000       0
   Ipv6                   4000       0
   Osi                    4000       0
   Mpls                   4000       0
Operational status        up         down       notPresent noResources
   Interface              4000       0          0          n/a
   Ip                     4000       0          0          0
   Ipv6                   0          0          4000       0
   Osi                    0          0          4000       0
   Mpls                   0          0          4000       0
Operational status        lowerDown  passive    tunnel
   Interface              0          0          0

PPP Multilink Status
Configuration status      configured notConfigured
   Link Interface         8000       n/a
   Network Interface      2000       n/a
   Ip                     2000       0
   Ipv6                   0          2000
   Osi                    0          2000
   Mpls                   0          2000
Administrative status     open       closed
   Link Interface         8000       0
   Network Interface      2000       0
   Ip                     2000       0
   Ipv6                   2000       0
   Osi                    2000       0
   Mpls                   2000       0
Operational status        up         down       notPresent noResources
   Link Interface         8000       0          0          n/a
   Network Interface      2000       0          0          n/a
   Ip                     2000       0          0          0
   Ipv6                   0          0          2000       0
   Osi                    0          0          2000       0
   Mpls                   0          0          2000       0
Operational status        lowerDown  passive    tunnel
   Link Interface         0          0          0
   Network Interface      0          0          0
```

Meaning Table 23 on page 310 lists the **show ppp interface summary** command output fields.

Table 23: show ppp interface summary Output Fields

Field Name	Field Description
PPP Status	Status of the nonmultilinked PPP interfaces.
Configuration Status	Indicates the configuration state of the PPP interface, IPCP, IPv6CP, OSINLCP, or MPLS: • configured—Interface or protocol is configured • notConfigured—Interface or protocol is not configured
Administrative status	Indicates the administrative state of the PPP interface, IPCP, IPv6CP, OSINLCP, or MPLS • open—Interface or protocol is administratively enabled • closed—Interface or protocol is administratively disabled
Operational status (Interface)	Indicates the operational state of the PPP interface • up—Interface is operational • down—Interface is not operational because of a problem in the PPP layer • lowerDown—Interface is not operational because a lower layer in the protocol stack is down • notPresent—Interface is not operational because the hardware is unavailable • passive—Interface is waiting for the peer to send an LCP confReq message • tunnel—Interface is being redirected through a tunnel
Operational status (Ip, Ipv6, Osi. Mpls)	Indicates the operational state of the IPCP, IPv6CP, OSINLCP, or MPLS protocol • up—Protocol is operational • down—Protocol is not operational because of a problem in the PPP layer • notPresent—Protocol is not operational because it does not exist • noResources—Protocol is not operational because it does not exist due to a lack of resources
PPP Multilink Status	Status of Multilinked PPP interfaces.

Related Documentation

- *show ppp interface summary*

Monitoring the Status of an IP Address in IPCP Configuration

Purpose Displays if the peer IP address is required for an IPCP negotiation.

Action To display if the peer IP address is required for the IPCP negotiation:

```
host1#show ppp peer-ip-address-optional
Ppp peer ip address optional: enabled
```

Meaning Table 24 on page 311 lists the **show ppp peer-ip-address-optional** command output fields.

Table 24: show ppp peer-ip-address-optional Output Fields

Field Name	Field Description
Ppp peer-ip-address-optional	Indicates whether the peer IP address is optional in the IPCP configuration request. Possible values: enabled or disabled.

Related Documentation

- *show ppp peer-ip-address-optional*

Monitoring AAA IPv4 Address Saving

Purpose Display the 32-byte string that the router communicates to the AAA server for released IPv4 addresses in a dual-stack.

Action To display the configured string in the VSA:

```
host1#show aaa ipv4-addr-saving
Address-Saving-String Configured :address
```

Meaning Table 25 on page 311 lists the **show aaa ipv4-addr-saving** command output field.

Table 25: show aaa ipv4-addr-saving Output Fields

Field Name	Field Description
Address-Saving-String Configured	32-byte string sent in the VSA

Related Documentation

- Configuring IPCP Renegotiations in a Dual-Stack Network for Optimal Utilization of Released IPv4 Addresses on page 294
- Overview of Processing IPCP Negotiations for Dual-Stack Subscribers on page 275
- *aaa ipv4-addr-saving*
- *show aaa ipv4-addr-saving*

Troubleshooting PPP Interfaces

Problem To analyze and diagnose issues with the PPP interface, you must have access to detailed information on the most recent PPP operations. By default, nothing is traced.

Solution For static PPP interfaces, you can configure a packet trace log for a PPP interface to diagnose issues on the PPP interface. First, specify a PPP interface which can be any one of the following interface types:

- serial—Serial interface
- atm—ATM interface
- pos—Packet over SONET interface

NOTE: For more information, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

```
host1(config-if)#log severity debug pppPacket serial 0/0:1/1
DEBUG 01/01/1970 00:16:58 pppPacket (1000001,*): interface:0/0:1/11/0:1,
time: 0.00, tx lcp confReq, id = 226, length = 19, mru = 32759,
authentication = chap MD5,magicNumber = 0x5387f9a2

DEBUG 01/01/1970 00:16:58 pppPacket (1000001,*): interface: 0/0:1/11/0:1,
time: 0.01, rx lcp confReq, id = 156, length = 18, mru = 32759,
magicNumber = 0x2d8eac91, pfc, acfc

DEBUG 01/01/1970 00:16:58 pppPacket (1000001,*): interface: 0/0:1/11/0:1,
time: 0.01, tx lcp confAck, id = 156, length = 18, mru = 32759,
magicNumber = 0x2d8eac91, pfc, acfc
```

You can also configure logging to direct the output to a specific destination. For information, see *Overview of System Logging*.

On dynamic PPP interfaces, you can use the **ppp log** command within the profile, as described in "Configuring Upper-Layer Dynamic Interfaces" on page 519. To enable PPP packet or state machine logging on any dynamic interface that uses the profile being configured, use one of the following keywords:

- **pppPacket**—Enables PPP packet logging
- **pppStateMachine**—Enables PPP state machine logging

```
host1(config-profile)#ppp log pppPacket
```

NOTE: This command is equivalent to the **log severity debug pppPacket** and **log severity debug pppStateMachine** commands.

Related Documentation

- *log severity*
- *ppp log*

Monitoring the Maximum Timeout of PPP Sessions

Purpose Displays whether the capability to preserve PPP sessions as active for the maximum timeout period of 13 years is enabled.

Action To display whether the capability to preserve PPP sessions as active for the maximum timeout period of 13 years is enabled using the **tech-support encoded-string** command:

```
host1#show ppp session-To-Thirteen-Years
Ppp Session To Thirteen Years: enabled
```

Meaning Table 26 on page 313 lists the **show session-To-Thirteen-Years** command output fields.

Table 26: show session-To-Thirteen-Years Output Fields

Field Name	Field Description
Ppp Session To Thirteen Years	Indicates whether the ability to maintain PPP sessions as active for the maximum timeout duration of 13 years is enabled or disabled

Related Documentation

- *show ppp session-To-Thirteen-Years*

CHAPTER 10

Configuring Multilink PPP

This chapter describes how to configure a Multilink Point-to-Point Protocol (PPP) interface on E Series routers.

This chapter contains the following sections:

- MLPPP Overview on page 316
- MLPPP Link Selection Overview on page 318
- MLPPP Platform Considerations on page 318
- MLPPP Interface Specifiers on page 319
- MLPPP Module Requirements on page 320
- MLPPP References on page 320
- Supported MLPPP Features on page 320
- MLPPP Unsupported Features on page 325
- Configuring Static MLPPP on page 325
- Example: Configuring Fragmentation and Reassembly for Static MLPPP on page 327
- MLPPP Contextual Command Differences on page 328
- Configuring MLPPP Authentication and Other Attributes on page 329
- Configuring Hash-Based MLPPP Link Selection on page 329
- Example: Configuring Hash-Based MLPPP Link Selection on page 330
- Configuring Dynamic MLPPP on page 332
- MLPPP Fragmentation and Reassembly on page 332
- Configuring Fragmentation and Reassembly for Dynamic MLPPP on page 336
- Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over PPPOE on page 338
- Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over L2TP on page 339
- Example: Configuring MLPPP Fragmentation and Reassembly for an MLPPP Bundle on page 341
- Multiclass MLPPP Overview on page 342
- Monitoring MLPPP on page 342

MLPPP Overview

Multilink PPP (MLPPP; also referred to as PPP Multilink, MLP, and MP) aggregates multiple physical links into a single logical bundle. More specifically, MLPPP bundles multiple link-layer channels into a single network-layer channel. Peers negotiate MLPPP during the initial phase of Link Control Protocol (LCP) option negotiation. Each router indicates that it is multilink capable by sending the multilink option as part of its initial LCP configuration request.

An MLPPP bundle can consist of multiple physical links of the same type—such as multiple asynchronous lines—or can consist of physical links of different types—such as leased synchronous lines and dial-up asynchronous lines.

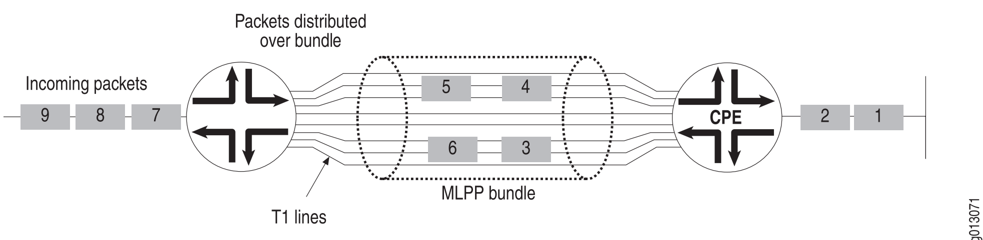
The router acts on MLPPP like another PPP Network Control Protocol (NCP). Packets received with an MLPPP header are subject to fragmentation, reassembly, and sequencing. Packets received without the MLPPP header cannot be sequenced and can be delivered only on a first-come, first-served basis.

Application

Some users need more bandwidth than a T1 or an E1 channel can provide, but cannot afford the expense or do not need the bandwidth of T3 or E3. Equal-cost multipath (ECMP) is one way to achieve the desired bandwidth. MLPPP is commonly used as an alternative to ECMP to deliver *NxT1* service. NxT1 service provides bandwidth greater than DS1 service without going up to the expense and infrastructure required for DS3 service. Cost-analysis of NxT1 versus DS3 service typically imposes a practical limit of 8xT1 service; that is, aggregation of no more than eight T1 or E1 connections into an MLPPP bundle.

The NxT1 implementation of MLPPP logically aggregates up to eight T1 or E1 connections into a single virtual connection, or bundle, to a given customer site, as shown in Figure 35 on page 316.

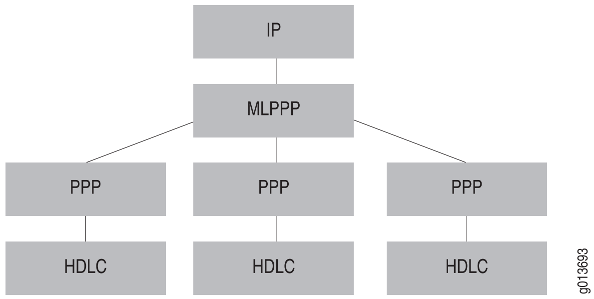
Figure 35: MLPPP Aggregation of T1 Lines into a Single Bundle



Because MLPPP aggregates multiple link-layer channels onto a single network-layer IP interface, protocol layering within the router is different than for non-multilink PPP.

Figure 36 on page 317 illustrates interface stacking with MLPPP.

Figure 36: Structure of MLPPP



MLPPP LCP Extensions

Multilink PPP adds the following LCP negotiation options:

- Multilink maximum received reconstructed unit (MRRU) option—The MRRU option has two functions. First, it informs the other end of the link the maximum size of the PPP packet payload that the router can receive. Second, it informs the other end that the router supports MLPPP. When you enable multilink on your router, the router includes the MRRU option in LCP negotiation with the value set to the maximum received unit (MRU) value for PPP. If the remote system rejects this option, the local system determines that the remote system does not support multilink PPP and it terminates the link without negotiation.

  NOTE: **The router does not bring up a link if the MRU value received from a peer device differs from the MRRU value received from the peer.**

- Short sequence number (SSN) header format option (not currently supported)—The SSN option indicates that the transmitting router wants to use a short sequence number (12 bits) in the MLPPP header rather than a long sequence number (24 bits). The router currently supports only long sequence numbers.
- Endpoint discriminator option—The endpoint discriminator option identifies the router transmitting the packet. If the receiving router determines that packets on another link have the same endpoint discriminator option, this link must be joined to that bundle. If the receiving router determines that no packets on other links have the same option, the receiving router must create a new bundle from this link.

  The endpoint discriminator is generated internally; you cannot configure it. The endpoint discriminator option is the same for all links on one end of the bundle; at the other end, all links also share a common endpoint discriminator. The two endpoint discriminators are different if the MLPPP bundle is set up between two E Series routers.

Related Documentation

- Configuring Static MLPPP on page 325
- Configuring Dynamic MLPPP on page 332

MLPPP Link Selection Overview

By default, E Series routers use a round-robin algorithm to select the link on which to transmit data on an MLPPP interface. The round-robin link selection method applies to both best-effort packets, such as data, and non-best-effort (high-priority) packets, such as voice and video. Best-effort packets are encapsulated with an MLPPP header that contains a sequence number, whereas non-best-effort packets are encapsulated with a PPP header that does *not* contain a sequence number.

The member links in an MLPPP bundle can experience different queuing delays due to the volume of traffic transmitted on the MLPPP interface. These delays can cause packets to arrive out of order at the remote router. The effect of such delays differs for best-effort packets and non-best effort packets, as follows:

- For best-effort packets that arrive out of order from the E Series router, the remote router can use the sequence number to reorder and forward the packets in the correct order, regardless of the order in which the packets were received.
- For non-best-effort packets that arrive out of order from the E Series router, the lack of a sequence number prevents the remote router from being able to determine the correct order in which to forward the packets. This can cause problems with applications that require high-priority voice and video traffic transmitted on MLPPP interfaces to be received in the same order transmitted by the peer applications.

To ensure that the E Series router maintains the proper packet order when transmitting non-best-effort traffic, you can use the **ppp hash-link-selection** command to enable use of a hash-based algorithm to select the link on which the router transmits high-priority packets on an MLPPP interface.

When you use hash-based link selection instead of the default round-robin link selection for non-best-effort traffic, the router uses the IP source address (SA) and IP destination address (DA) of the packet as a hash to select the MLPPP member link on which to transmit the packet. Specifically, the router uses the hash algorithm to bind the transmission of all traffic between this IP SA and IP DA to the same member link in the MLPPP bundle.

If the member link selected to transmit high-priority packets becomes inoperable or is removed from the MLPPP bundle, the router must select a different link on which to transmit the packets. As a result, packets transmitted on this new link might sometimes arrive at the remote destination before the traffic sent on the previously selected member link.

Related Documentation

- Multiclass MLPPP Overview on page 342

MLPPP Platform Considerations

You can configure MLPPP interfaces on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Related Documentation

- MLPPP Overview on page 316
- MLPPP Interface Specifiers on page 319
- MLPPP Module Requirements on page 320
- MLPPP References on page 320

MLPPP Interface Specifiers

Some of the configuration task examples in this chapter use the *slot/port* format to specify the physical interface on which you want to configure MLPPP. However, the interface specifier format that you use depends on the type of physical interface on which you want to configure MLPPP and on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port* format. For example, the following command specifies an ATM interface on slot 5, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

```
host1(config)#interface atm 5/1
```

For E120 and E320 routers, use the *slot/adapter/port* format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies a tunnel-server port on slot 3, adapter 0, port 0 of an E320 router.

```
host1(config)#tunnel-server 3/0/0
```

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Related Documentation

- MLPPP Overview on page 316
- MLPPP Platform Considerations on page 318
- MLPPP Module Requirements on page 320
- MLPPP References on page 320

MLPPP Module Requirements

For information about the modules that support MLPPP interfaces on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support MLPPP.

For information about the modules that support MLPPP interfaces on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support MLPPP.

Related Documentation

- MLPPP Overview on page 316
- MLPPP Platform Considerations on page 318
- MLPPP Interface Specifiers on page 319
- MLPPP References on page 320

MLPPP References

For more information about the MLPPP protocol and MLPPP fragmentation and reassembly, consult the following resources:

- RFC 1661—The Point-to-Point Protocol (PPP) (July 1994)
- RFC 1990—The PPP Multilink Protocol (MP) (August 1996)
- RFC 2233—The Interfaces Group MIB using SMIv2 (November 1997)

Related Documentation

- MLPPP Overview on page 316
- MLPPP Platform Considerations on page 318
- MLPPP Interface Specifiers on page 319
- MLPPP Module Requirements on page 320

Supported MLPPP Features

The router currently supports both the static configuration of the links participating in a multilink bundle and the dynamic creation of MLPPP bundles over L2TP (on the LAC and LNS) when the LAC or LNS detects multilink LCP option negotiation in LCP proxy data. You can also configure MLPPP bundles for locally terminated PPP sessions.

The following MLPPP features are available for both static and dynamic MLPPP:

- Logical aggregation of up to eight links in a bundle
- Long sequence numbers
- Authentication for interfaces with MLPPP encapsulation or for MLPPP bundles
- Monotonically increasing sequence numbers

  All packets distributed across the member links have monotonically increasing sequence numbers. This feature enables the remote system on the customer premises to perform resequencing (if the system is configured to do so).

- Round-robin packet distribution or hash-based packet distribution

  By default, E Series routers use a round-robin algorithm to handle packet distribution across the member links in a bundle for both best-effort traffic and non-best-effort traffic. The round-robin approach is used even when the member links have different line rates.

  As an alternative to round-robin packet distribution for non-best-effort traffic, you can enable use of a hash-based algorithm for distribution of non-best-effort (high-priority) packets, such as voice or video. Using a hash-based packet distribution mechanism instead of the default round-robin packet distribution mechanism for non-best-effort traffic ensures that the router maintains the proper packet order when transmitting high-priority packets. For details, see "MLPPP Link Selection Overview" on page 318.

- Forwarding of multilink traffic to L2TP tunnels

  E Series routers support dynamic MLPPP over L2TP configurations (on the L2TP access concentrator or the L2TP network server).

- Fragmentation and reassembly

  For details, see "MLPPP Fragmentation and Reassembly" on page 332.

- Packet resequencing for best-effort traffic, for non-best-effort traffic, and when MLPPP reassembly is enabled

  For details on how the router supports packet resequencing for best-effort traffic and non-best-effort traffic, see "MLPPP Link Selection Overview" on page 318.

  For details on enabling MLPPP reassembly, see "MLPPP Fragmentation and Reassembly" on page 332.

- Multiclass MLPPP

  For information about multiclass MLPPP, see "Configuring Multiclass Multilink PPP" on page 361.

You can configure bundles as follows:

- On a COCX-F3 line module and its corresponding I/O modules, you can configure:
  - Up to 8 member links from different ports in the same bundle, with the following restriction for MLPPP reassembly:

    - For a COCX-F3 line module with either a 12-port E3-12 FRAME I/O module or a 12-port CT3/T3 12 I/O module, the restriction is based on the ports on which member links in the same bundle are configured.

      A 12-port E3-12 FRAME I/O module and a 12-port CT3/T3 12 I/O module each contain 12 ports numbered 0 through 11. When MLPPP reassembly is enabled, you can configure a bundle with member links on the same port; on ports 0, 1, and 2; on ports 3, 4, and 5; on ports 6, 7, and 8; or on ports 9, 10, and 11. However, the router *cannot* properly reassemble fragments if you configure a bundle with member links that span ports in different bundles; for example, on ports 0, 1, and 4.

      When MLPPP reassembly is disabled, this restriction is not in effect; that is, member links can span ports in different bundles.

  - Up to 12 bundles

- On a cOCx/STMx line module and its corresponding I/O module, you can configure:
  - Member links from different OC3/STM1 ports in the same bundle, with the following restrictions for MLPPP reassembly:
    - For a cOCx/STMx line module with a 4-port cOC3/STM1 I/O module, the restriction is based on the ports on which member links in the same bundle are configured.

      A 4-port cOC3/STM1 I/O module contains four ports numbered 0 through 3. When MLPPP reassembly is enabled, you can configure a bundle with member links on the same port, on ports 0 and 1, or on ports 2 and 3. However, the router *cannot* properly reassemble fragments if you configure a bundle with member links that span ports in different bundles; for example, on ports 1 and 2.

      When MLPPP reassembly is disabled, this restriction is not in effect; that is, member links can span ports in different bundles.

    - For a cOCx/STMx line module with a 1-port cOC12/STM4 I/O module, the restriction is based on the STM1 (OC3) paths on which member links in the same bundle are configured.

      A 1-port cOC12/STM4 I/O module has four logical paths numbered 1 through 4. When MLPPP reassembly is enabled, you can configure a bundle with member links on the same path, on paths 1 and 2, or on paths 3 and 4. However, the router *cannot* properly reassemble fragments if you configure a bundle with member links that span paths in different bundles; that is, on paths 2 and 3.

      When MLPPP reassembly is disabled, this restriction is not in effect; for example, member links can span paths in different bundles.

  - Any combination of bundles that does not exceed the 336 available T1 channels (for example, 336 single-link T1 bundles, 42 eight-link bundles, or 41 eight-link bundles and 8 single-link bundles)
  - Any combination of bundles that does not exceed the 252 available E1 channels (for example, 252 single-link T1 bundles, 34 eight-link bundles, or 33 eight-link bundles and 8 single-link bundles)

- On a CT3/T3-F0 line module with a CT3/T3 12 I/O module, you can configure:

  - Member links from different T3 ports in the same bundle
  - Any combination of bundles that does not exceed the 336 available T1 channels (for example, 336 single-link T1 bundles, 42 eight-link bundles, or 41 eight-link bundles and 8 single-link bundles)

- On an ES2-S1 Service IOA, you can configure:
  - Up to 16,000 member links per line module, not to exceed a total of 12,000 MLPPP bundles per chassis
  - Any combination of bundles that does not exceed either of these maximums (for example, 4000 single-link bundles, 4000 two-link bundles, 4000 four-link bundles, and 2000 eight-link bundles)

- On an OCx/STMx ATM line module and its corresponding line modules, you can configure:
  - Up to 8000 member links per line module, not to exceed a total of 8000 MLPPP bundles per chassis
  - Any combination of bundles that does not exceed either of these maximums (for example, 4000 single-link bundles, 4000 two-link bundles, 2000 four-link bundles, and 1000 eight-link bundles)

- On a Service line module (SM), you can configure:
  - Up to 16,000 member links per line module, not to exceed a total of 12,000 MLPPP bundles per chassis
  - Any combination of bundles that does not exceed either of these maximums (for example, 4000 single-link bundles, 4000 two-link bundles, 4000 four-link bundles, and 2000 eight-link bundles)

- On a shared tunnel-server port configured on a GE-2 or GE-HDE line module and corresponding line modules, you can configure:
  - Up to 8000 member links per line module, not to exceed a total of 8000 MLPPP bundles per chassis
  - Any combination of bundles that does not exceed either of these maximums (for example, 4000 single-link bundles, 4000 two-link bundles, 2000 four-link bundles, and 1000 eight-link bundles)

- On an ES2-S1 GE-4 IOA, ES2-S1 GE-8 IOA, ES2-S1 OC3-8 STM1 ATM IOA, and ES2-S1 OC12-2 STM4 ATM IOA modules that pair with an ES2 4G LM on E120 and E320 routers, you can configure:
  - MLPPP bundles with one or more links per bundle for dynamic MLPPP-over-PPPoE-over-Ethernet configurations.
  - MLPPP bundles with only one link per bundle when configuring static MLPPP-over-PPPoE-over-Ethernet. When you create multilink bundles in a static MLPPP-over-PPPoE-over-Ethernet configuration, PPPoE is unable to direct the PPPoE Active Discovery Initiation (PADI) packets received from the MLPPP bundle links on the client to the appropriate (matching) links in the MLPPP bundle on the

server. As a result, the connections between bundle links become crossed, and the bundle does not come up as expected. Creating MLPPP bundles with only a single link for this configuration ensures a one-to-one correspondence between a PPPoE subscriber and its associated link, and guarantees that the MLPPP bundle comes up properly.

  - MLPPP bundles with only a single link per bundle are *not* required for static MLPPP-over-PPPoE-over-Ethernet with VLAN configurations if all of the links in a bundle have the same VLAN ID that is unique across all MLPPP bundles configured on the line module.

- On all E Series ATM module combinations that support MLPPP, you can configure:
  - MLPPP bundles with one or more links per bundle for dynamic MLPPP-over-multiple PPPoE subinterfaces-over-one PPPoE major interface-over-ATM 1483 subinterface configurations.
  - MLPPP bundles with only one link per bundle when configuring static MLPPP-over-multiple PPPoE subinterfaces-over-one PPPoE major interface-over-an ATM 1483 subinterface. In this configuration, you can stack multiple PPPoE subinterfaces over a single PPPoE major interface.
  - Typically when you create ATM PVCs on an ATM module, there is a one-to-one correspondence between a PPPoE subscriber and the ATM PVC with which the subscriber is associated. However, in configurations with multiple PPPoE subinterfaces stacked over a single PPPoE major interface, crossed MLPPP bundle link connections can occur, as is the case with the ES2-S1 GE-4 IOA, and the bundle does not come up as expected. Creating MLPPP bundles with only a single link for this configuration ensures a one-to-one correspondence between a PPPoE subscriber and its associated link, and guarantees that the MLPPP bundle comes up properly.
  - MLPPP bundles with only a single link per bundle are *not* required for static MLPPP-over-multiple PPPoE subinterfaces-over-one PPPoE major interface-over-ATM 1483 subinterface configurations if all PPPoE subinterfaces stacked over the same PPPoE major interface belong to the same bundle.

NOTE: For information about the modules that support MLPPP on ERX14xx models, ERX7xx models, and the ERX310 router, see *ERX Module Guide, Appendix A, Module Protocol Support*. For information about the modules that support MLPPP on the E120 and E320 routers, see *E120 and E320 Module Guide, Appendix A, IOA Protocol Support*.

- IPv6 Neighbor Discovery router advertisements are supported for multilink PPP interfaces when a multilink PPP bundle between the subscriber and the PPP server running on the router uses IPv6 for data transmission. When an IPv6 interface is stacked on a multilink PPP bundle, the delegating router allocates IPv6 prefixes to the requesting routers. The router uses ICMPv6 Neighbor Discovery router advertisements to respond to route solicitation packets it receives from the subscriber.

Related Documentation

- MLPPP Unsupported Features on page 325
- MLPPP Overview on page 316

MLPPP Unsupported Features

The router does not support the following MLPPP features:

- Short sequence numbers
- Resequencing out-of-order packets in the absence of fragmentation

  Given the location in the network where the router resides, the *N*xT1 links to a customer site represent one of many places across the IP network where packets might be received out of order. For example, if the router has multiple uplinks to a core router, packets might be received out of order across these links.

  You can lose packets if you transmit layer 2 traffic on an MPLS LSP that passes over an MLPPP link bundle.

  Packets are passed along to the next protocol layer in the order that they are processed. Packet resequencing is therefore performed at the end station rather than the aggregation router. IP datagrams can be resequenced by the end station using the IP identification field.

  Layer 2 packets such as Ethernet/MPLS and ATM-AAL5/MPLS have no sequence number information and are sent in the order received. The packets are dropped if their out-of-order condition is detected by a downstream device.

  Frame Relay/MPLS packets do have a native sequence number in the header and are rejected at the end of the LSP if the MLPPP sequence number order is violated.

  To ensure that the router maintains the proper packet order when transmitting high-priority (non-best-effort) packets such as voice and video, you can use the **ppp hash-link-selection** command to enable use of a hash-based algorithm to select the link on which the router transmits high-priority packets on an MLPPP interface. For details, see “MLPPP Link Selection Overview” on page 318.

Related Documentation

- MLPPP Overview on page 316
- Supported MLPPP Features on page 320

Configuring Static MLPPP

The following tasks show you how to configure static MLPPP:

- Before You Configure Static MLPPP on page 326
- Configuring Static MLPPP and Adding Member Links to a Multilink Bundle on page 326

Before You Configure Static MLPPP

Before you begin configuring static MLPPP, you must configure the physical line interfaces to be aggregated by MLPPP. See the following chapters:

- Configuring Channelized T3 Interfaces in *JunosE Physical Layer Configuration Guide*
- Configuring T3 and E3 Interfaces in *JunosE Physical Layer Configuration Guide*
- Configuring Channelized OCx/STMx Interfaces in *JunosE Physical Layer Configuration Guide*

The procedures described in "Configuring Static MLPPP and Adding Member Links to a Multilink Bundle" on page 326 assume that a physical line interface has been configured.

Configuring Static MLPPP and Adding Member Links to a Multilink Bundle

Static MLPPP configuration consists of two general tasks, each with several subtasks.

To configure static MLPPP:

1. Create the member links to be aggregated into a multilink bundle.
   a. From Global Configuration mode, specify the individual interface on which you want to configure MLPPP.

      host1(config)#**interface serial 2/0:1/1**
   b. Specify MLPPP as the encapsulation method on the interface.

      host1(config-if)#**encapsulation mlppp**
   c. (Optional) Specify the keepalive timeout value for the member link interface.

      host1(config-if)#**ppp keepalive 50**
   d. (Optional) Specify the authentication method for the member link interface.

      host1(config-if)#**ppp authentication pap chap**
   e. (Optional) Enable hash-based link selection instead of the default round-robin link selection for the member link interface.

      host1(config-if)#**ppp hash-link-selection**
2. Add member links to a multilink bundle.
   a. Define the MLPPP bundle.

      host1(config)#**interface mlppp group1**
   b. Add each member link.

      host1(config-if)#**member-interface serial 2/0:1/1**
   c. Assign an IP address to the MLPPP bundle.

      host1(config-if)#**ip address 10.10.100.1 255.255.255.0**
   d. (Optional) Specify the keepalive timeout value for the MLPPP network interface (the entire MLPPP bundle).

host1(config-if)#**ppp keepalive 50**

e. (Optional) Specify the authentication method for the MLPPP network interface (the entire MLPPP bundle).

host1(config-if)#**ppp authentication pap chap**

f. (Optional) Enable hash-based link selection instead of the default round-robin link selection for the MLPPP network interface (the entire MLPPP bundle).

host1(config-if)#**ppp hash-link-selection**

Example: Configuring Fragmentation and Reassembly for Static MLPPP

This example shows you how to configure fragmentation and reassembly for static MLPPP.

- Requirements on page 327
- Overview on page 327
- Configuring Fragmentation and Reassembly for Static MLPPP on page 327

Requirements

Before you begin configuring static MLPPP, you must configure the individual interfaces on which you want to enable MLPPP. See the following:

- Configuring Channelized T3 Interfaces in *JunosE Physical Layer Configuration Guide*
- Configuring T3 and E3 Interfaces in *JunosE Physical Layer Configuration Guide*
- Configuring Channelized OCx/STMx Interfaces in *JunosE Physical Layer Configuration Guide*

Overview

To configure fragmentation and reassembly for a static MLPPP interface you must specify the individual link interfaces on which you want to configure fragmentation and reassembly. The **ppp fragmentation** command and **ppp reassembly** command then enable fragmentation and reassembly for the static MLPPP interface. Once the link interfaces are created, you must add them to the MLPPP bundle using the **member-interface** command.

Configuring Fragmentation and Reassembly for Static MLPPP

Step-by-Step Procedure

The following example configures MLPPP fragmentation and reassembly for two member links in an MLPPP bundle over an ATM 1483 subinterface.

1. From Global Configuration mode, specify the individual link interface on which you want to configure fragmentation and reassembly.

   host1(config)#**interface atm 2/0**
   host1(config-if)#**interface atm 2/0.2**
   host1(config-subif)#**atm pvc 42 0 42 aal5snap**

2. Specify MLPPP as the encapsulation method on the link interface.


```
host1(config-subif)#encapsulation mlppp
```

3. Enable fragmentation, reassembly, and authentication on the link interface.

```
host1(config-subif)#ppp fragmentation
host1(config-subif)#ppp reassembly 1400
host1(config-subif)#ppp authentication pap chap
host1(config-subif)#exit
```

4. You must similarly configure all additional link interfaces on which you want to configure fragmentation and reassembly.

```
host1(config)#interface atm 2/0.3
host1(config-subif)#atm pvc 43 0 43 aal5snap
host1(config-subif)#encapsulation mlppp
host1(config-subif)#ppp fragmentation
host1(config-subif)#ppp reassembly 1600
host1(config-subif)#ppp authentication pap chap
host1(config-subif)#exit
```

5. Define the MLPPP bundle.

```
host1(config)#interface mlppp client1
```

6. Add the link interfaces as member links to the bundle.

```
host1(config-if)#member-interface atm 2/0.2
host1(config-if)#member-interface atm 2/0.3
```

7. Assign an IP address to the MLPPP bundle.

```
host1(config-if)#ip address 10.10.200.1 255.255.255.0
```

Related Documentation

- Configuring Fragmentation and Reassembly for Dynamic MLPPP on page 336
- MLPPP Fragmentation and Reassembly Overview on page 333

MLPPP Contextual Command Differences

The MLPPP configuration commands have different effects depending on the interface context. If you issue an MLPPP configuration command in the context of an individual interface, the command affects only the MLPPP link interface associated with that individual interface.

For example, the following commands disable negotiation of the local magic number only for serial interface 2/0:1/1.

```
host1(config-if)#member-interface serial 2/0:1/1
host1(config-if)#encapsulation mlppp
host1(config-if)#ppp magic-number disable
```

If you issue an MLPPP configuration command in the context of an MLPPP bundle—the MLPPP network interface—the command affects all the member links of the bundle. This feature prevents you from having to issue MLPPP configuration commands for each member link interface.

For example, the following commands disable negotiation of the local magic number for the entire bundle, *group1*.

```
host1(config)#interface mlppp group1
host1(config-if)#member-interface serial 2/0:1/1
host1(config-if)#ip address 10.10.100.1 255.255.255.0
host1(config-if)#ppp magic-number disable
```

Any member links added to the bundle after issuing an MLPPP configuration command are not affected by the command. For example, if you add serial interface 2/0:4/1 to the *group1* bundle after you issue the **ppp magic-number disable** command, negotiation of the local magic number for this link and any member links subsequently added to the bundle is not disabled.

Related Documentation

- Configuring Hash-Based MLPPP Link Selection on page 329

Configuring MLPPP Authentication and Other Attributes

Perform the following optional tasks to configure authentication on interfaces with MLPPP encapsulation or MLPPP bundles.

- Specify one or more PPP authentication types.
- Modify the length of the CHAP challenge.
- Specify the maximum number of retries.

NOTE: The JunosE Software PPP application accepts null usernames during PAP and CHAP authentication. When the PPP application receives an authentication request that includes a null username, PPP passes the request to AAA. To take advantage of this feature, configure your authentication server to support the use of null usernames.

The available **ppp** command options are the same for interfaces whether they are configured with PPP or MLPPP.

Related Documentation

- MLPPP Overview on page 316
- *ppp authentication*

Configuring Hash-Based MLPPP Link Selection

You can configure hash-based MLPPP link selection in any of the following ways:

- To configure hash-based link selection for a individual MLPPP member link, issue the **ppp hash-link-selection** command from Interface Configuration mode or Subinterface Configuration mode in the context of the individual link interface.
- To configure hash-based link selection for all current member links in an MLPPP bundle, issue the **ppp hash-link-selection** command from Interface Configuration mode in the

context of the MLPPP bundle. Doing this has the same effect as issuing the **ppp hash-link-selection** command separately for each member link in the bundle.

- To configure hash-based link selection for all dynamic MLPPP link interfaces created by a profile, issue the **ppp hash-link-selection** command from Profile Configuration mode.

For a detailed description and examples of using the **ppp hash-link-selection** command.

Related Documentation

- Configuring Static MLPPP and Adding Member Links to a Multilink Bundle on page 326
- MLPPP Contextual Command Differences on page 328
- Configuring Dynamic MLPPP on page 332
- Example: Configuring Hash-Based MLPPP Link Selection on page 330

- *ppp hash-link-selection*

Example: Configuring Hash-Based MLPPP Link Selection

This example shows you how to configure hash-based MLPPP link selection.

- Requirements on page 330
- Overview on page 330
- Configuring Hash-Based MLPPP Link Selection on page 331

Requirements

This example uses the following software and hardware components:.

- E Series router (ERX7xx models, ERX14xx models, the ERX310 router, the E120 router, or the E320 router)
- ASIC-based line modules that support Fast Ethernet or Gigabit Ethernet
- JunosE Release 7.1.0 or higher-numbered releases

Overview

A hash-based algorithm is used to select the link on which the router transmits non-best-effort (high-priority) packets, such as voice or video, on the dynamic MLPPP interfaces created by the profile. Using hash-based link selection instead of the default round-robin link selection for non-best-effort traffic ensures that the router maintains the proper packet order when transmitting high-priority packets. When you configure hash-based link selection, the router uses the IP source address and IP destination address of the packet as a hash to select the MLPPP member link on which to transmit the packet.

NOTE: Hash-based MLPPP link selection is available only for non-best-effort traffic.

Configuring Hash-Based MLPPP Link Selection

Step-by-Step Procedure

The following example creates a hash-based MLPPP link selection for an individual MLPPP member link interface.

1. Configure the link interface.

   host1(config)#**interface atm 2/0**

2. Configure the virtual circuit parameters such as the virtual circuit ID and encapsulation for the link interface.

   host1(config-subif)#**atm pvc 42 0 42 aal5snap**

3. Configure MLPPP as the encapsulation method on the interface and enables use of a hash-based algorithm on the MLPPP interface.

   host1(config-subif)#**encapsulation mlppp**
   host1(config-subif)#**ppp hash-link-selection**

For All Current Member Links In An MLPPP Bundle

Step-by-Step Procedure

The following commands configure hash-based MLPPP link selection for all current member links in the MLPPP bundle (group1). Doing this has the same effect as issuing the ppp hash-link-selection command separately for each member link in the bundle.

1. Configure the MLPPP bundle.

   host1(config)#**interface mlppp group1**

2. Enable use of a hash-based algorithm to select the link on which the router transmits high-priority (non-best-effort) packets, such as voice or video, on the MLPPP bundle interface.

   host1(config-if)#**ppp hash-link-selection**

For All Dynamic MLPPP Links

Step-by-Step Procedure

The following commands configure hash-based MLPPP link selection for all dynamic MLPPP interfaces created by the profile named dynamicMlppp

1. From Global Configuration mode, create a profile by assigning it a name, and access Profile Configuration mode.

   host1(config)#**profile dynamicMlppp**

2. Enable the creation of dynamic MLPPP interfaces.

   host1(config-profile)#**ppp multilink enable**

3. Enable use of hash-based algorithm on a dynamic MLPPP interface.

   host1(config-subif)#**ppp hash-link-selection**

Related Documentation

- Configuring Hash-Based MLPPP Link Selection on page 329

Configuring Dynamic MLPPP

You can define a profile to dynamically create MLPPP bundles over L2TP on the LNS. The profile consists of commands to define the bundle attributes, just as you would for static configuration. For more information about profiles for dynamic interfaces, see "Dynamic Interface Configuration Using a Profile" on page 565 and "Configuring MLPPP and PPP Characteristics for a Profile" on page 582.

To configure a profile for dynamic MLPPP:

1. Create a profile by assigning it a name.

   host1(config)#**profile dynmlppp**

2. Enable creation of dynamic MLPPP interfaces.

   host1(config-profile)#**ppp multilink enable**

3. Specify a virtual router to which dynamic IP interfaces created using this profile will be assigned.

   host1(config-profile)#**ip virtual-router egypt**

4. Specify an IP loopback interface with which dynamic IP interfaces created using this profile will be associated.

   host1(config-profile)#**ip unnumbered loopback 0**

5. (Optional) Set other desired PPP characteristics by using the **ppp** commands described in "Configuring MLPPP Authentication and Other Attributes" on page 329.

Related Documentation

- Configuring Static MLPPP on page 325

- *ppp multilink enable*
- *ip virtual-router*

MLPPP Fragmentation and Reassembly

You can configure MLPPP fragmentation and reassembly on a static link interface before adding the link to a bundle, or in a profile assigned to a dynamic MLPPP interface. You can also configure fragmentation and reassembly for all current member links in an MLPPP bundle.

- MLPPP Fragmentation and Reassembly Overview on page 333
- MLPPP Fragmentation and Reassembly Supported Parameters on page 333
- MLPPP Fragmentation and Reassembly Module Requirements on page 334
- MLPPP Fragmentation and Reassembly Configuration Parameters on page 334
- MLPPP Bundle Validation and Configuration Guidelines on page 334
- MLPPP Fragmentation Guidelines on page 335
- MLPPP Reassembly Guidelines on page 335

- MLPPP Bundle Validation Failure on page 336
- Recovering from MLPPP Bundle Validation Failure on page 336

MLPPP Fragmentation and Reassembly Overview

E Series routers support fragmentation and reassembly as part of their MLPPP implementation. *Fragmentation* is the process by which a large packet is broken up into multiple smaller fragments for simultaneous transmission across multiple links of an MLPPP bundle. *Reassembly* is the process by which the destination router reassembles the fragments into the original packets.

Application

You can use MLPPP fragmentation and reassembly to reduce transmission latency. You can also use the feature to implement a packet-prioritization scheme that allows smaller, delay-sensitive packets (such as high-priority voice packets) to be interleaved with or race ahead of larger, delay-insensitive packets (such as low-priority data packets) when they are transmitted in the network.

Related Documentation

- Configuring Fragmentation and Reassembly for Dynamic MLPPP on page 336
- Example: Configuring Fragmentation and Reassembly for Static MLPPP on page 327

MLPPP Fragmentation and Reassembly Supported Parameters

Table 27 on page 333 lists the static and dynamic MLPPP configurations on E Series routers that support fragmentation and reassembly.

Table 27: Supported Configurations for MLPPP Fragmentation and Reassembly

Static MLPPP Configurations	Dynamic MLPPP Configurations
Static MLPPP over ATM 1483 subinterfaces	Dynamic MLPPP over ATM 1483 subinterfaces
Static MLPPP over PPPoE over ATM 1483 subinterfaces	Dynamic MLPPP over PPPoE over ATM 1483 subinterfaces
Static MLPPP over serial (HDLC) interfaces	Dynamic MLPPP over serial (HDLC) interfaces
–	Dynamic MLPPP over L2TP (on the L2TP network server)
Static MLPPP over PPPoE over Ethernet	Dynamic MLPPP over PPPoE over Ethernet

Related Documentation

- MLPPP Fragmentation and Reassembly Overview on page 333
- Example: Configuring Fragmentation and Reassembly for Static MLPPP on page 327
- Configuring Fragmentation and Reassembly for Dynamic MLPPP on page 336
- Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over PPPOE on page 338

• Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over L2TP on page 339

MLPPP Fragmentation and Reassembly Module Requirements

For a list of the line modules and corresponding I/O modules that support MLPPP fragmentation and reassembly on ERX7xx models, ERX14xx models, and the ERX310 router, see *ERX Module Guide*, *Appendix A*, *Module Protocol Support*.

For a list of the line modules and corresponding IOAs that support MLPPP fragmentation and reassembly on the E120 and E320 routers, see *E120 and E320 Module Guide, Appendix A, IOA Protocol Support*.

Related Documentation

• MLPPP Fragmentation and Reassembly Overview on page 333

MLPPP Fragmentation and Reassembly Configuration Parameters

The parameters for MLPPP fragmentation and reassembly are configured on a per-link basis for each link interface (also known as a member link) in an MLPPP bundle.

By default, fragmentation and reassembly are disabled for MLPPP links. You can enable or disable fragmentation and reassembly for an individual link, or for all member links in a bundle, by using the **ppp fragmentation** and **ppp reassembly** commands. However, you must configure the same fragmentation setting and the same reassembly setting—enabled or disabled—for all member links in a bundle.

When you use the **ppp fragmentation** command to enable fragmentation on a link, you can optionally specify the maximum fragment size to be used on the link interface. When you use the **ppp reassembly** command to enable reassembly on a link, you can optionally specify the administrative multilink maximum received reconstructed unit (MRRU) value for the link.

Related Documentation

• MLPPP Fragmentation and Reassembly Overview on page 333
• MLPPP Fragmentation and Reassembly Supported Parameters on page 333
• MLPPP Fragmentation Guidelines on page 335
• MLPPP Reassembly Guidelines on page 335

MLPPP Bundle Validation and Configuration Guidelines

When you configure MLPPP, the router validates that each link interface attempting to join a statically or dynamically created bundle has Link Control Protocol (LCP) parameters that are compatible with the other member links already in the bundle. This validation includes examining the parameters configured for fragmentation and reassembly on a particular link interface and verifying that these parameters are compatible with the other member links in the bundle.

To ensure that the bundle validation succeeds, make sure you observe the following configuration guidelines for MLPPP fragmentation and reassembly.

Related Documentation

- MLPPP Bundle Validation Failure on page 336
- Recovering from MLPPP Bundle Validation Failure on page 336
- Example: Configuring MLPPP Fragmentation and Reassembly for an MLPPP Bundle on page 341

MLPPP Fragmentation Guidelines

Use the following guidelines when you configure MLPPP fragmentation on a link interface:

- Configure the same fragmentation setting—enabled or disabled—for all member links in a bundle.
- When fragmentation is enabled, configure the same fragment size for all member links in a bundle.
- Make sure a link's fragment size does not exceed its maximum transmission unit (MTU) size.
- Do not configure both MLPPP fragmentation (with the **ppp fragmentation** command) and IP fragmentation of L2TP packets (with the **ip mtu** command) on the same interface. Instead, you must choose only one of the fragmentation configurations by setting it to the necessary value and set the other fragmentation configuration to the maximum allowable value.

Related Documentation

- MLPPP Fragmentation and Reassembly Overview on page 333
- MLPPP Fragmentation and Reassembly Supported Parameters on page 333
- MLPPP Fragmentation and Reassembly Configuration Parameters on page 334
- Example: Configuring Fragmentation and Reassembly for Static MLPPP on page 327
- Configuring Fragmentation and Reassembly for Dynamic MLPPP on page 336

MLPPP Reassembly Guidelines

Use the following guidelines when you configure MLPPP reassembly on a link interface:

- Configure the same reassembly setting—enabled or disabled—for all member links in a bundle.
- Make sure a link's administrative MRRU is greater than or equal to the local maximum receive unit (MRU) negotiated both on that link and on other member links in the bundle.
- The local MRRU negotiated on a link must be the same as the local MRRU negotiated on the other member links in the bundle.
- The peer MRRU negotiated on a link must be the same as the peer MRRU negotiated on the other member links in the bundle.

- When reassembly is enabled, member links belonging to the same bundle can have different local MRU values.
- When reassembly is disabled, member links belonging to the same bundle must negotiate the same local MRU value.

Related Documentation

- MLPPP Fragmentation and Reassembly Overview on page 333
- MLPPP Fragmentation and Reassembly Supported Parameters on page 333
- MLPPP Fragmentation and Reassembly Configuration Parameters on page 334
- Example: Configuring Fragmentation and Reassembly for Static MLPPP on page 327
- Configuring Fragmentation and Reassembly for Dynamic MLPPP on page 336

MLPPP Bundle Validation Failure

If an MLPPP link interface fails bundle validation because one or more of the preceding configuration guidelines are not met, the router's actions differ depending on whether you are using a static MLPPP configuration or a dynamic MLPPP configuration, as follows:

- For static MLPPP configurations, the router permits the failed link to join the bundle, but forces the link into a down state.
- For dynamic MLPPP configurations, the router prohibits the failed link from joining the bundle, and subsequently tears down the link.

Related Documentation

- Recovering from MLPPP Bundle Validation Failure on page 336
- MLPPP Bundle Validation and Configuration Guidelines on page 334

Recovering from MLPPP Bundle Validation Failure

To recover from a bundle validation failure, you must reconfigure the link interface (for static MLPPP configurations) or reconfigure the profile (for dynamic MLPPP configurations) according to the guidelines described in "MLPPP Bundle Validation and Configuration Guidelines" on page 334.

Related Documentation

- MLPPP Bundle Validation Failure on page 336
- MLPPP Bundle Validation and Configuration Guidelines on page 334

Configuring Fragmentation and Reassembly for Dynamic MLPPP

To configure fragmentation and reassembly for dynamic MLPPP, you must create a profile that includes commands to define the link and bundle attributes, just as you do for a static MLPPP configuration.

For more information, see:

- "Configuring Upper-Layer Dynamic Interfaces" on page 519

- *LAC Configuration Prerequisites*
- *LNS Configuration Prerequisites*

To define a profile that configures MLPPP fragmentation and reassembly for a dynamic MLPPP interface:

1. From Global Configuration mode, create a profile by assigning it a name, and access Profile Configuration mode.

   host1(config)#**profile dynmlppp1**
   host1(config-profile)#

2. Enable the creation of dynamic MLPPP interfaces.

   host1(config-profile)#**ppp multilink enable**

3. Enable fragmentation on the link interface, and optionally specify the maximum allowable fragment size to use.

   host1(config-profile)#**ppp fragmentation 128**

   NOTE: You can specify the maximum fragment size for a link only when you use the **ppp fragmentation** command to enable fragmentation on that link. You cannot specify the maximum fragment size for a link when fragmentation is disabled.

4. Enable reassembly on the link interface, and optionally specify the administrative MRRU value to use.

   host1(config-profile)#**ppp reassembly 1800**

   NOTE: You can specify the administrative MRRU value for a link only when you use the **ppp reassembly** command to enable reassembly on that link. You cannot specify the administrative MRRU for a link when reassembly is disabled.

5. (Optional) Specify a virtual router to which dynamic IP interfaces created with this profile will be assigned.

   host1(config-profile)#**ip virtual-router boston**

6. (Optional) Specify an IP loopback interface with which dynamic IP interfaces created with this profile will be associated.

   host1(config-profile)#**ip unnumbered loopback 0**

7. (Optional) Set other PPP characteristics as needed by using the **ppp** commands described in "Configuring Multilink PPP" on page 315.

Related Documentation

- Configuring Dynamic MLPPP on page 332
- Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over L2TP on page 339

- Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over PPPOE on page 338

- *ppp multilink enable*
- *ppp fragmentation*
- *ppp reassembly*

Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over PPPOE

This example shows you how to configure fragmentation and reassembly for dynamic MLPPP over PPPoE.

- Requirements on page 338
- Overview on page 338
- Configuring Fragmentation and Reassembly for Dynamic MLPPP Interface over PPPoE on page 338

Requirements

Before you begin configuring static MLPPP, you must configure the individual interfaces on which you want to enable MLPPP. See the following:

- Configuring Channelized T3 Interfaces in *JunosE Physical Layer Configuration Guide*
- Configuring T3 and E3 Interfaces in *JunosE Physical Layer Configuration Guide*
- Configuring Channelized OCx/STMx Interfaces in *JunosE Physical Layer Configuration Guide*

Overview

To configure fragmentation and reassembly for dynamic MLPPP interface you must first enable the creation of a dynamic profile using the **ppp dynamic enable** command. The **ppp fragmentation** and **ppp reassembly** commands then enable fragmentation and reassembly for the dynamic MLPPPP interface.

Configuring Fragmentation and Reassembly for Dynamic MLPPP Interface over PPPoE

Step-by-Step Procedure

The following example configures MLPPP fragmentation and reassembly for a dynamic MLPPP interface over dynamic PPPoE over an ATM 1483 subinterface.

1. From Global Configuration mode, create a profile by assigning it a name, and access Profile Configuration mode.

   ```
   host1(config)#profile dynmlppp2
   ```

2. Enable creation of dynamic MLPPP interfaces and enable fragmentation and reassembly on the link interface.

   ```
   host1(config-profile)#ppp multilink enable
   host1(config-profile)#ppp fragmentation 128
   host1(config-profile)#ppp reassembly 1800
   ```


3. (Optional) Specify a virtual router to which dynamic IP interfaces created with this profile will be assigned.

   host1(config-profile)#**ip virtual-router westford**

4. (Optional) Specify an IP loopback interface with which dynamic IP interfaces created with this profile will be associated.

   host1(config-profile)#**ip unnumbered loopback 1**

5. (Optional) Set other PPP and PPPoE characteristics.

   host1(config-profile)#**pppoe sessions 9**
   host1(config-profile)#**ppp authentication chap**
   host1(config-profile)#**exit**

6. Configure dynamic MLPPP link interface.

   host1(config)#**interface atm 4/0**
   host1(config-if)#**interface atm 4/0.1**
   host1(config-subif)#**atm pvc 52 0 52 aal5autoconfig 0 0 0**
   host1(config-subif)#**profile pppoe dynmlppp2**
   host1(config-subif)#**auto-configure pppoe**

Related Documentation

- Configuring Dynamic MLPPP on page 332
- Configuring Fragmentation and Reassembly for Dynamic MLPPP on page 336
- Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over L2TP on page 339

Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over L2TP

This example shows you how to configure fragmentation and reassembly for dynamic MLPPP over L2TP.

- Requirements on page 339
- Overview on page 340
- Configuring Fragmentation and Reassembly for Dynamic MLPPP over L2TP on page 340

Requirements

Before you begin configuring static MLPPP, you must configure the individual interfaces on which you want to enable MLPPP. See the following:

- Configuring Channelized T3 Interfaces in *JunosE Physical Layer Configuration Guide*
- Configuring T3 and E3 Interfaces in *JunosE Physical Layer Configuration Guide*
- Configuring Channelized OCx/STMx Interfaces in *JunosE Physical Layer Configuration Guide*

Overview

To configure fragmentation and reassembly for dynamic MLPPP over L2TP, create an L2TP profile using the **profile l2tp-profile** command. You can then define the router address as the destination LAC address using the **l2tp destination profile** command.

Configuring Fragmentation and Reassembly for Dynamic MLPPP over L2TP

Step-by-Step Procedure

The following example configures MLPPP fragmentation and reassembly for a dynamic MLPPP interface over L2TP over a Gigabit Ethernet interface.

1. Configure the IP address of the router and define a loopback interface.

```
host1(config)#ip router-id 193.1.1.1
host1(config)#interface loopback 0
```

2. Configure the Gigabit Ethernet interface.

```
host1(config-if)#ip address 193.1.1.1 255.255.255.0
host1(config-if)#interface gigabitEthernet 1/1
host1(config-if)#ip unnumbered loopback 0
host1(config-if)#exit
host1(config)#ip route 193.1.1.2 255.255.255.255 gigabitEthernet 1/1
```

3. Configure the L2TP profile.

```
host1(config)#profile l2tp-profile
host1(config-profile)#ip virtual-router default
host1(config-profile)#ip unnumbered loopback 0
host1(config-profile)#ip access-routes
host1(config-profile)#ppp authentication pap
host1(config-profile)#ppp keepalive
host1(config-profile)#ppp multilink enable
host1(config-profile)#ppp mru 1590
host1(config-profile)#ppp reassembly 1590
host1(config-profile)#ppp fragmentation 128
host1(config-profile)#pppoe session 8000
host1(config-profile)#exit
```

4. Configure the L2TP destination profile that defines the router IP address as the LAC address.

```
host1(config)#l2tp destination profile lac ip address 193.1.1.2
host1(config-l2tp-dest-profile)#remote host xxx.com
host1(config-l2tp-dest-profile-host)#enable proxy authenticate
host1(config-l2tp-dest-profile-host)#tunnel password welcome
host1(config-l2tp-dest-profile-host)#profile l2tp-profile
```

Related Documentation

- Configuring Dynamic MLPPP on page 332
- Configuring Fragmentation and Reassembly for Dynamic MLPPP on page 336

Example: Configuring MLPPP Fragmentation and Reassembly for an MLPPP Bundle

This example shows you how to configure MLPPP fragmentation and reassembly for an MLPPP bundle.

- Requirements on page 341
- Overview on page 341
- Configuring MLPPP Fragmentation and Reassembly for All Member Links on page 341

Requirements

This example uses the following software and hardware components:.

- E Series router (ERX7xx models, ERX14xx models, the ERX310 router, the E120 router, or the E320 router)
- ASIC-based line modules that support Fast Ethernet or Gigabit Ethernet
- JunosE Release 7.1.0 or higher-numbered releases

Overview

If you issue the **ppp fragmentation** command or the **ppp reassembly** command in the context of an MLPPP bundle, the command affects all the current member links in the bundle. This enables you to issue a single command for the entire bundle instead of having to issue individual commands for each member link in the bundle. Any member links added to the bundle after you issue an MLPPP configuration command in the bundle context are not affected by the command.

Configuring MLPPP Fragmentation and Reassembly for All Member Links

The following commands configure MLPPP fragmentation and reassembly for all member links in the bundle group1. If you add a member link to the group1 bundle after you issue the **ppp fragmentation** or **ppp reassembly** command, MLPPP fragmentation and reassembly for this link and any member links subsequently added to the bundle is not enabled.

1. Configure the MLPPP bundle.

   ```
   host1(config)#interface mlppp group1
   ```

2. Enable fragmentation and reassembly and optionally specify the maximum allowable fragment size to use and the administrative MRRU value.

   ```
   host1(config-if)#ppp fragmentation 128
   host1(config-if)#ppp reassembly 1590
   host1(config-if)#exit
   ```

Related Documentation

- Example: Configuring Fragmentation and Reassembly for Static MLPPP on page 327
- Example: Configuring Fragmentation and Reassembly for Dynamic MLPPP Over PPPOE on page 338

Multiclass MLPPP Overview

Multiclass MLPPP enables you to fragment data packets of different priorities into multiple classes in an MLPPP bundle. It also enables you to send high-priority packets between fragments of larger, lower priority packets. Multiclass MLPPP ensures that high-priority, delay-sensitive traffic, such as voice and video, are received in the proper sequence.

For information about multiclass MLPPP, see "Configuring Multiclass Multilink PPP" on page 361.

Related Documentation

- Multiclass MLPPP Overview on page 361

Monitoring MLPPP

Use the commands in this section to display information about MLPPP interfaces.

You can set a statistics baseline for MLPPP serial (member link) or bundle (multilink) interfaces using the **baseline ppp** command. Use the **delta** keyword with the **show** commands described below to display statistics with the baseline values subtracted.

After you configure multilink PPP, you can use the **show ppp interface** commands to display configuration and statistics information about MLPPP and MLPPP fragmentation and reassembly.

You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. For details, see *show Commands* in *JunosE System Basics Configuration Guide*.

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

The following tasks display information about MLPPP interfaces:

- Monitoring Baseline Statistics for MLPPP Interfaces on page 342
- Monitoring MLPPP Information Configured on PPP Interfaces on page 348
- Monitoring MLPPP Summary Information on page 359

Monitoring Baseline Statistics for MLPPP Interfaces

Purpose Display baseline statistics for PPP interfaces—including MLPPP interfaces, either individual serial (member link) interfaces or multilink (bundle) interfaces.

Action To display PPP interface (including MLPPP interface) statistics without baselining:

- Use only the **serial** or **mlppp** keywords.
- For serial interfaces, specify the interface location in the format *slot/port:channel/subchannel* for CT3 modules.
- For MLPPP interfaces, specify the interface location as the name of the MLPPP bundle.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- When baselining is requested, the time since the last baseline was set is displayed in *hours:minutes:seconds* or *days/hours* format. If a baseline was not set, the following message is displayed instead:

  ```
  No baseline has been set
  ```

host1#**show ppp interface statistics**

```
PPP interface serial 2/0:4/1 is up
No baseline has been set
Interface statistics            in                  out
   packets                      0                   0
   octets                       572                 684
   errors                       0                   0
   discards                     0                   0
PPP interface serial 2/0:5/1 is up
No baseline has been set
Interface statistics            in                  out
   packets                      0                   0
   octets                       572                 684
   errors                       0                   0
   discards                     0                   0
PPP interface serial 2/1:4/1 is up
No baseline has been set
Interface statistics            in                  out
   packets                      0                   0
   octets                       572                 684
   errors                       0                   0
   discards                     0                   0
PPP interface serial 2/1:5/1 is up
No baseline has been set
Interface statistics            in                  out
   packets                      0                   0
   octets                       572                 684
   errors                       0                   0
   discards                     0                   0
4 ppp interfaces found

PPP interface mlppp group1 is up
PPP multilink member-interface serial 2/0:1/1 is up
No baseline has been set
Interface statistics            in                  out
   packets                      0                   0
   octets                       608                 716
   errors                       0                   0
   discards                     0                   0
PPP multilink member-interface serial 2/0:2/1 is up
No baseline has been set
Interface statistics            in                  out
   packets                      0                   0
```

```
   octets                     608               716
   errors                     0                 0
   discards                   0                 0
PPP multilink member-interface serial 2/0:3/1 is up
No baseline has been set
Interface statistics          in                out
   packets                    0                 0
   octets                     596               704
   errors                     0                 0
   discards                   0                 0

PPP interface mlppp group2 is up
PPP multilink member-interface serial 2/1:1/1 is up
No baseline has been set
Interface statistics          in                out
   packets                    0                 0
   octets                     628               740
   errors                     0                 0
   discards                   0                 0
PPP multilink member-interface serial 2/1:2/1 is up
No baseline has been set
Interface statistics          in                out
   packets                    0                 0
   octets                     628               740
   errors                     0                 0
   discards                   0                 0
PPP multilink member-interface serial 2/1:3/1 is up
No baseline has been set
Interface statistics          in                out
   packets                    0                 0
   octets                     616               728
   errors                     0                 0
   discards                   0                 0
2 mlppp interfaces found
```

To display PPP interface (including MLPPP interface) statistics with baselining:

- Use the optional **delta** keyword with MLPPP **show** commands to specify that baselined statistics are to be shown.

host1#**show ppp interface statistics delta**

```
PPP interface serial 2/0:4/1 is up
Time since last baseline 00:00:35
Interface statistics             in                 out
   packets                       0                  0
   octets                        75                 82
   errors                        0                  0
   discards                      0                  0
PPP interface serial 2/0:5/1 is up
Time since last baseline 00:00:37
Interface statistics             in                 out
   packets                       0                  0
   octets                        87                 90
   errors                        0                  0
   discards                      0                  0
PPP interface serial 2/1:4/1 is up
Time since last baseline 00:00:39
Interface statistics             in                 out
   packets                       0                  0
   octets                        101                112
   errors                        0                  0
   discards                      0                  0
PPP interface serial 2/1:5/1 is up
Time since last baseline 00:00:43
Interface statistics             in                 out
   packets                       0                  0
   octets                        94                 99
   errors                        0                  0
   discards                      0                  0
4 ppp interfaces found
PPP interface mlppp group1 is up
PPP multilink member-interface serial 2/0:1/1 is up
Time since last baseline 00:00:17
Interface statistics             in                 out
   packets                       0                  0
   octets                        28                 26
   errors                        0                  0
   discards                      0                  0
PPP multilink member-interface serial 2/0:2/1 is up
Time since last baseline 00:10:22

Interface statistics             in                 out
   packets                       0                  0
   octets                        102                104
   errors                        0                  0
   discards                      0                  0
PPP multilink member-interface serial 2/0:3/1 is up
Time since last baseline 00:00:19
Interface statistics             in                 out
   packets                       0                  0
   octets                        112                126
   errors                        0                  0
   discards                      0                  0

PPP interface mlppp group2 is up
PPP multilink member-interface serial 2/1:1/1 is up
Time since last baseline 00:00:23
Interface statistics             in                 out
```

```
   packets                       0                   0
   octets                        125                 132
   errors                        0                   0
   discards                      0                   0
PPP multilink member-interface serial 2/1:2/1 is up
Time since last baseline 00:00:25
Interface statistics             in                  out
   packets                       0                   0
   octets                        135                 138
   errors                        0                   0
   discards                      0                   0
PPP multilink member-interface serial 2/1:3/1 is up
Time since last baseline 00:00:30
Interface statistics             in                  out
   packets                       0                   0
   octets                        125                 132
   errors                        0                   0
   discards                      0                   0
2 mlppp interfaces found
```

Meaning Table 28 on page 346 lists the **show ppp interface statistics** command output fields.

Table 28: show ppp interface statistics Output Fields

Field Name	Field Description
PPP interface mlppp *bundleName*	Name and state (up or down) of the MLPPP interface
PPP multilink member-interface *interfaceName*	Name and state (up or down) of the MLPPP member link interface
Link interface administrative status	Administrative state of the member link interface: open (enabled) or closed (disabled)
PPP multilink multiclass	Configuration of multiclass MLPPP on the MLPPP interface: enabled or disabled
PPP multilink multiclass classes	Number of multilink classes created on the MLPPP interface: 1 through 8
PPP multilink multiclass fragmentation	Configuration of fragmentation of the multiclass MLPPP interface: enabled or disabled. If fragmentation is enabled then the command also lists the QoS traffic classes on which fragmentation is enabled.
PPP multilink multiclass reassembly	Configuration of reassembly of the multiclass MLPPP interface: enabled or disabled. If reassembly is enabled then it also lists the QoS traffic classes on which reassembly is enabled.
mlppp interfaces found	Number of MLPPP interfaces configured
Time since last baseline	When baselining is configured, the time since the last baseline was set is displayed in *hours:minutes:seconds* or *days/hours* format

Table 28: show ppp interface statistics Output Fields *(continued)*

Field Name	Field Description
receive class, transmit class	Number of multilink classes configured on the MLPPP bundle
Interface statistics	Statistics for data received by (in) and transmitted (out) on the MLPPP interface: • packets—Number of packets received and transmitted on the interface • octets—Number of octets received and transmitted on the interface • errors—Number of errors received and transmitted on the interface • discards—Number of packets discarded on receipt or discarded before they were transmitted • fragments—Number of fragments received and transmitted on the interface
Receive Class *number*	Information about the packets of the specified multilink class received by the router
fragments in reassembly list	Number of buffered fragments reassembled
reordered	Number of fragments reordered
discarded fragments/bytes	Number of fragments and bytes that have been discarded.
last received sequence number	Sequence number of the last multiclass MLPPP packet received
Transmit Class *number*	Information about the packets of the specified multilink class transmitted by the router
fragments	Number of fragments that are sent by the router
last sent sequence number	Sequence number of the last multiclass MLPPP packet sent
Bundle name	Name of the MLPPP bundle
MLPPP interface *interfaceName*	Name and state (up or down) of the configured MLPPP interface

Table 28: show ppp interface statistics Output Fields *(continued)*

Field Name	Field Description
LCP negotiated options	Negotiated LCP options for the local and peer systems: • max-receive-unit—Negotiated maximum receive unit, in octets, for the local and remote (peer) side of the link • max-receive-reconstructed-unit—Negotiated maximum receive reconstructed unit, in octets, for the local and remote (peer) side of the link • authentication—Negotiated authentication method (none, pap, or chap) for the local and remote (peer) side of the link • magic-number—Negotiated magic number for the local and remote (peer) side of the link • pfc—Negotiated protocol field compression (none or enabled) for the local and remote (peer) side of the link • acfc—Negotiated address and control field compression (none or enabled) for the local and remote (peer) side of the link • multiclass-classes—Number of multilink classes negotiated • multiclass-sequence-format—Format of the negotiated sequence number: long or short

Monitoring MLPPP Information Configured on PPP Interfaces

Purpose Display statistics for MLPPP information configured on PPP interfaces.

Action To display information about the MLPPP member links configured in bundle group1;

```
host1#show ppp interface mlppp group1 members
PPP interface mlppp group1 is up
 PPP multilink member-interface serial 2/0:1/1 is up
 PPP multilink member-interface serial 2/0:2/1 is up
 PPP multilink member-interface serial 2/0:3/1 is up
```

To display information about all MLPPP member links configured for all bundles:

```
host1#show ppp interface mlppp members
PPP interface mlppp group1 is up
 PPP multilink member-interface serial 2/0:1/1 is up
 PPP multilink member-interface serial 2/0:2/1 is up
 PPP multilink member-interface serial 2/0:3/1 is up
PPP interface mlppp group2 is up
 PPP multilink member-interface serial 2/1:1/1 is up
 PPP multilink member-interface serial 2/1:2/1 is up
 PPP multilink member-interface serial 2/1:3/1 is up
PPP interface mlppp group3
 No member-interfaces found
```


To display information about all MLPPP encapsulated links, regardless of whether the links are members of an MLPPP bundle:

```
host1#show ppp interface mlppp links
PPP multilink interface serial 2/0:1/1 is up
PPP multilink interface serial 2/0:2/1 is up
PPP multilink interface serial 2/0:3/1 is up
PPP multilink interface serial 2/1:1/1 is up
PPP multilink interface serial 2/1:2/1 is up
PPP multilink interface serial 2/1:3/1 is up
```

To display configuration information about MLPPP member links configured in bundle group1:

```
host1#show ppp interface mlppp group1 config
PPP interface mlppp group1 is up
Network interface administrative status is open
Configured network protocol is IPCP
PPP multilink member-interface ATM 10/0.10 is up
Link interface administrative status is open
Link interface fragmentation is enabled
Link interface fragment size is 128
Link interface reassembly is enabled
Link interface administrative MRRU is 2000
PPP multilink member-interface ATM 10/0.11 is down (lower layer down)
Link interface administrative status is closed
Link interface fragmentation is enabled
Link interface fragment size is 128
Link interface reassembly is enabled
Link interface administrative MRRU is 2000
PPP multilink member-interface ATM 10/0.12 is down (lower layer down)
Link interface administrative status is closed
Link interface fragmentation is enabled
Link interface fragment size is 128
Link interface reassembly is enabled
Link interface administrative MRRU is 2000
PPP multilink member-interface ATM 10/0.13 is down (lower layer down)
Link interface administrative status is closed
Link interface fragmentation is enabled
Link interface fragment size is 128
Link interface reassembly is enabled
Link interface administrative MRRU is 2000
1 mlppp interfaces found
```

To display statistics about all configured MLPPP member links configured in bundle group1:

```
host1#show ppp interface mlppp group1 statistics
PPP interface mlppp group1 is up
PPP multilink member-interface ATM 10/0.10 is up
No baseline has been set
Interface statistics            in                   out
   packets                      0                    0
   octets                       170                  690
   errors                       0                    0
   discards                     0                    0
PPP multilink member-interface ATM 10/0.11 is down (lower layer down)
No baseline has been set
Interface statistics            in                   out
   packets                      0                    0
   octets                       50                   0
```

```
   errors                        0                   0
   discards                      0                   0
PPP multilink member-interface ATM 10/0.12 is down (lower layer down)
No baseline has been set
Interface statistics             in                  out
   packets                       0                   0
   octets                        50                  0
   errors                        0                   0
   discards                      0                   0
PPP multilink member-interface ATM 10/0.13 is down (lower layer down)
No baseline has been set
Interface statistics             in                  out
   packets                       0                   0
   octets                        50                  0
   errors                        0                   0
   discards                      0                   0
1 mlppp interfaces found
```

To display status information about the specified MLPPP bundle:

```
host1#show ppp interface mlppp group1 status
PPP interface mlppp group1 is up
1 mlppp interfaces found
```

To display complete configuration, statistics, and status information about the specified MLPPP bundle:

```
host1#show ppp interface mlppp group1 full
PPP interface mlppp group1 is up
Network interface administrative status is open
Configured network protocol is IPCP
IPCP protocol configuration
   configured                    true
   administrative-status         open
   ip-address                    1.2.3.4
   dns-precedence                local
   wins-precedence               local
   max-negotiations              10
IPCP protocol status
   operational-status            up
IPCP negotiated options          local              peer
   ip-address                    1.2.3.4            6.7.8.9
   primary-dns-address           none               none
   secondary-dns-address         none               none
   primary-wins-address          none               none
   secondary-wins-address        none               none
OSINLCP protocol configuration
   configured                    false
   administrative-status         open
OSINLCP protocol status
   operational-status            not present

   terminate-reason              not configured
PPP multilink member-interface serial 2/0:1/1 is up
Link interface administrative status is open
No baseline has been set
Interface statistics             in                 out
   packets                       0                  0
   octets                        1488               1972
   errors                        0                  0
   discards                      0                  0
LCP protocol configuration
```

```
   max-receive-unit               use lower layer
   authentication                 none
   magic-number                   enabled
   magic-number-mismatch          ignore
   keepalive-timer                30 seconds
   restart-timer                  3 seconds
   max-terminate                  2
   max-configure                  10
   max-failure                    5
LCP protocol status
   link-status                    network

LCP negotiated options            local              peer
   max-receive-unit               1590               1590
   max-receive-reconstructed-unit 1590               1590
   authentication                 none               none
   magic-number                   0x6c079eb0         0x2c5a5798
   pfc                            none               none
   acfc                           none               none
LCP Endpoint Discriminator options
   local discriminator class      Locally Assigned Address
   local endpoint discriminator   0x31393933313030303800001b000001
   peer discriminator class       Locally Assigned Address
   peer endpoint discriminator    0x31393933313030303800001b000002
LCP protocol statistics
   in-keepalive-requests          70
   out-keepalive-requests         70
   in-keepalive-replies           70
   out-keepalive-replies          70
   keepalive-failures             0
   max-renegotiation-terminates   10
PPP multilink member-interface serial 2/0:2/1 is up
Link interface administrative status is open
No baseline has been set
Interface statistics              in                 out
   packets                        0                  0
   octets                         1508               1996
   errors                         0                  0
   discards                       0                  0
LCP protocol configuration
   max-receive-unit               use lower layer
   authentication                 none
   magic-number                   enabled
   magic-number-mismatch          ignore
   keepalive-timer                30 seconds
   restart-timer                  3 seconds
   max-terminate                  2
   max-configure                  10
   max-failure                    5
LCP protocol status
   link-status                    network
LCP negotiated options            local              peer
   max-receive-unit               1590               1590
   max-receive-reconstructed-unit 1590               1590
   authentication                 none               none
   magic-number                   0x7ada4a05         0x1bb178cd
   pfc                            none               none
   acfc                           none               none

LCP Endpoint Discriminator options
   local discriminator class      Locally Assigned Address
   local endpoint discriminator   0x31393933313030303800001b000001
```

```
   peer discriminator class       Locally Assigned Address
   peer endpoint discriminator    0x31393933313030303800001b000002
LCP protocol statistics
   in-keepalive-requests          71
   out-keepalive-requests         71
   in-keepalive-replies           71
   out-keepalive-replies          71
   keepalive-failures             0
PPP multilink member-interface serial 2/0:3/1 is up
Link interface administrative status is open
No baseline has been set
Interface statistics              in                  out
   packets                        0                   0
   octets                         1568                2068
   errors                         0                   0
   discards                       0                   0
LCP protocol configuration
   max-receive-unit               use lower layer
   authentication                 none
   magic-number                   enabled
   magic-number-mismatch          ignore
   keepalive-timer                30 seconds
   restart-timer                  3 seconds
   max-terminate                  2
   max-configure                  10
   max-failure                    5
LCP protocol status
   link-status                    network
LCP negotiated options            local               peer
   max-receive-unit               1590                1590
   max-receive-reconstructed-unit 1590                1590
   authentication                 none                none
   magic-number                   0x31cc52e0          0x32ebdec6
   pfc                            none                none
   acfc                           none                none
LCP Endpoint Discriminator options
   local discriminator class      Locally Assigned Address
   local endpoint discriminator   0x31393933313030303800001b000001
   peer discriminator class       Locally Assigned Address
   peer endpoint discriminator    0x31393933313030303800001b000002
LCP protocol statistics
   in-keepalive-requests          74
   out-keepalive-requests         74
   in-keepalive-replies           74
   out-keepalive-replies          74
   keepalive-failures             0
1 mlppp interfaces found
```

To display information about the MLPPP interface:

```
host1#show ppp interface mlppp 0/0-1 full
PPP interface mlppp 0/0-1 is up
Network interface administrative status is open
Configured network protocols are IPCP, IPV6CP
IPCP protocol configuration
   configured                     true
   administrative-status          open
   ip-address                     192.168.1.1
   dns-precedence                 local
   wins-precedence                local
   ipcp-netmask-option            disabled
   ipcp-lockout                   enabled
```



```
IPCP protocol status
   operational-status              up
IPCP negotiated options            local               peer
   ip-address                      192.168.1.1         100.100.100.17
   ip-address-mask                 none                none
   primary-dns-address             none                none
   secondary-dns-address           none                none
   primary-wins-address            none                none
   secondary-wins-address          none                none
IPV6CP protocol configuration
   configured                      true
...

PPP multilink member-interface ATM 0/0.10 is up
Link interface administrative status is open
Link interface fragmentation is enabled
Link interface fragment size is (use MTU)
Link interface reassembly is enabled
Link interface administrative MRRU is (use MRU)
Link interface hash-link-selection is disabled
No baseline has been set
...
Authentication status
   grant                           true
   session-timeout                 31622400 seconds
   inactivity-timeout              none
   monitor-ingress-only            false
   accounting-timeout              none
   peer-ip-address                 100.100.100.17
   peer-ip-address-mask            none
   peer-primary-dns-address        none
   peer-secondary-dns-address      none
   peer-primary-wins-address       none
   peer-secondary-wins-address     none
   peer-ipv6-interface-id          none
```

Meaning Table 29 on page 353 lists the **show ppp interface mlppp** command output fields

Table 29: show ppp interface mlppp Output Fields

Field Name	Field Description
PPP interface mlppp	Name and administrative status (up or down) for an MLPPP bundle
PPP multilink interface	Interface type, interface specifier, and administrative status (up or down) for an MLPPP member link
Network interface administrative status	Indicates whether the interface for the MLPPP bundle is administratively enabled (open), meaning that the **no ppp shutdown**command is operational, or administratively disabled (closed), meaning that the **ppp shutdown** command is operational
Configured network protocol	Network protocol configured on the interface

Table 29: show ppp interface mlppp Output Fields *(continued)*

Field Name	Field Description
Link interface administrative status	Indicates whether the interface for the member link is administratively enabled (open), meaning that the **no ppp shutdown** command is operational, or administratively disabled (closed), meaning that the **ppp shutdown** command is operational
Link interface fragmentation	Indicates whether MLPPP fragmentation is enabled or disabled on the link interface
Link interface fragmentation size	MLPPP fragment size, in octets, currently in use on the link interface
Link interface reassembly	Indicates whether MLPPP reassembly is enabled or disabled on the link interface
Link interface administrative MRRU	Administrative MRRU value, in octets, currently in use on the link interface
Interface statistics	
packets	Number of packets received (in) and sent (out) on the interface
octets	Number of octets received (in) and sent (out) on the interface
errors	Number of errors received (in) and sent (out) on the interface
discards	Number of packets discarded on receipt (in) or discarded before they were transmitted (out) NOTE: For the LCP, IPCP, and OSINLCP negotiated options, the command displays a value of “none” if the option was not negotiated.
IPCP protocol configuration	
configured	IPCP is configured on this interface (true or false)
administrative- status	IPCP administrative status (open or closed)
ip-address	Address to be used for negotiation of local IP address option
dns-precedence	Used to resolve conflicts during DNS address negotiation
wins-precedence	Used to resolve conflicts during WINS address negotiation

Table 29: show ppp interface mlppp Output Fields *(continued)*

Field Name	Field Description
max-negotiations	Maximum number of renegotiation attempts that the router accepts before terminating a PPP session
IPCP protocol status	
operational- status	IPCP operational status (up, down, not present, or not present no resources)
IPCP negotiated options	
ip-address	Negotiated IP address for the local and remote (peer) side of the link
primary-dns-address	Negotiated primary DNS address for the local and remote (peer) side of the link
secondary-dns-address	Negotiated secondary DNS address for the local and remote (peer) side of the link
primary-wins-address	Negotiated primary WINS address for the local and remote (peer) side of the link
secondary-wins-address	Negotiated secondary WINS address for the local and remote (peer) side of the link NOTE: The command displays a value of “none” for any negotiated option parameters if the option was not negotiated.
OSINLCP protocol configurations	
configured	OSINLCPis configured on this interface (true or false)
administrative -status	OSINLCP administrative status (open or closed)
OSINLCP protocol status	
operational-status	OSINLCP operational status (up, down, not present, or not present no resources)
terminate-reason	Reason for termination of OSINLCP service
LCP protocol configuration	

Table 29: show ppp interface mlppp Output Fields *(continued)*

Field Name	Field Description
max-receive-unit	Controls negotiation of the local MRU option; value can be one of the following: • use lower layer–MRU of the layer below PPP defines the MRU to be negotiated • disabled–MRU option is not to be negotiated • a numeric value–MRU value to be negotiated
authentication	Controls negotiation of the local authentication option; value can be one of the following: • none–Do not negotiate • chap–Negotiate CHAP • pap–Negotiate PAP • chap/pap–Negotiate CHAP, and if it is rejected, negotiate PAP • pap/chap–Negotiate PAP, and if it is rejected, negotiate CHAP
magic-number	Controls whether the local magic number is negotiated: enabled (negotiate), or disabled (do not negotiate)
magic-number-mismatch	Indicates whether the router is configured to ignore the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number: ignore (ignore the peer magic number mismatch and retain the PPP connection), or reject (router terminates the PPP connection if it detects a peer magic number mismatch)
keepalive-timer	Rate of LCP echo requests, in seconds
restart-timer	Retry frequency during LCP, IPCP, and OSINLCP negotiations, in seconds
max-terminate	Maximum number of terminate requests
max-configure	Maximum number of configure requests
max-failure	Maximum number of configure NAKs
LCP protocol status	
link-status	Indicates the overall status of LCP negotiations, including the following states: initial (idle), starting (ready to negotiate), authenticate (authenticating), and network (LCP is up)
LCP negotiated options	

Table 29: show ppp interface mlppp Output Fields *(continued)*

Field Name	Field Description
max-receive-unit	Negotiated maximum receive unit, in octets, for the local and remote (peer) side of the link
max-receive-reconstruct-unit	Negotiated maximum receive reconstructed unit, in octets, for the local and remote (peer) side of the link
authentication	Negotiated authentication method (none, pap, or chap) for the local and remote (peer) side of the link
magic-number	Negotiated magic number for the local and remote (peer) side of the link
pfc	Negotiated pfc (none or enabled) for the local and remote (peer) side of the link
acfc	Negotiated acfc (none or enabled) for the local and remote (peer) side of the link NOTE: The command displays a value of “none” for any negotiated option parameters if the option was not negotiated.
LCP Endpoint Discriminator options	
local discriminator class	Endpoint discriminator type, format, and address space for the local system
local endpoint discriminator	Endpoint discriminator value for the local system within the specified class
peer discriminator class	Endpoint discriminator type, format, and address space for the remote system
peer endpoint discriminator	Endpoint discriminator value for the remote system within the specified class
LCP protocol statistics	
in-keepalive-requests	Number of received keepalive requests (LCP Echo Request) for the life of the interface (since either system boot or interface creation, whichever is later)
out-keepalive-requests	Number of transmitted keepalive requests for the life of interface
in-keepalive-replies	Number of received keepalive replies for the life of the interface
out-keepalive-replies	Number of transmitted keepalive replies for the life of the interface

Table 29: show ppp interface mlppp Output Fields *(continued)*

Field Name	Field Description
keepalive-failures	Number of keepalive failures reported on the interface
max-renegotiations-terminates	Number of renegotiation terminations for the PPP session
IPV6CP protocol configuration	IPV6CP is configured on this interface (true or false)
Authentication status	
grant	Authentication status. Possible values are: • true–access granted • false–access not granted
session-timeout	Session timeout, in seconds; session is terminated at expiration
inactivity-timeout	Inactivity timeout, in seconds; session is terminated if it is not active for specified timeout
monitor-ingress-only	Indicates to monitor the ingress interface only
accounting-timeout	Accounting timeout in seconds; frequency of accounting updates to the authentication server
peer-ip-address	IPaddress to be used in negotiation of peer IPaddress
peer-ip-address-mask	IP address mask to be used in negotiation of the peer IP address mask
peer-primary-dns-address	IP address to be used in negotiation of the peer primary DNS address
peer-secondary-dns-address	IP address to be used in negotiation of the peer secondary DNS address
peer-primary-wins-address	IP address to be used in negotiation of the peer primary WINS address
peer-secondary-wins-address	IP address to be used in negotiation of the peer secondary WINS address
peer-ipv6-interface-id	IPv6 interface Identifier of the peer returned in an authentication grant from the RADIUS server

Monitoring MLPPP Summary Information

Purpose Display a summary of all the multilinked and non-multilinked PPP interfaces configured on the router.

Action To display a summary of all MLPPP and non-MLPPP interfaces:

```
host1#show ppp interface summary
PPP Status
Configuration status      configured notConfigured
   Interface              4000       n/a
   Ip                     4000       0
   Ipv6                   0          4000
   Osi                    0          4000
   Mpls                   0          4000
Administrative status     open       closed
   Interface              4000       0
   Ip                     4000       0
   Ipv6                   4000       0
   Osi                    4000       0
   Mpls                   4000       0
Operational status        up         down       notPresent noResources
   Interface              4000       0          0          n/a
   Ip                     4000       0          0          0
   Ipv6                   0          0          4000       0
   Osi                    0          0          4000       0
   Mpls                   0          0          4000       0
Operational status        lowerDown  passive    tunnel
   Interface              0          0          0

PPP Multilink Status
Configuration status      configured notConfigured
   Link Interface         8000       n/a
   Network Interface      2000       n/a
   Ip                     2000       0
   Ipv6                   0          2000
   Osi                    0          2000
   Mpls                   0          2000
Administrative status     open       closed
   Link Interface         8000       0
   Network Interface      2000       0
   Ip                     2000       0
   Ipv6                   2000       0
   Osi                    2000       0
   Mpls                   2000       0
Operational status        up         down       notPresent noResources
   Link Interface         8000       0          0          n/a
   Network Interface      2000       0          0          n/a
   Ip                     2000       0          0          0
   Ipv6                   0          0          2000       0
   Osi                    0          0          2000       0
   Mpls                   0          0          2000       0
Operational status        lowerDown  passive    tunnel
   Link Interface         0          0          0
   Network Interface      0          0          0
```

Meaning Table 30 on page 360 lists the **show ppp interface summary** command output fields

Table 30: show ppp interface summary Output Fields

Field Name	Field Description
PPP status	Non-multilinked PPP interfaces
Configuration status	Indicates the configuration state of the PPP interface, IPCP, IPv6CP, OSINLCP, or MPLS • configured–Interface or protocol is configured • notConfigured–Interface or protocol is not configured
Administrative status	Indicates the administrative state of the PPP interface, IPCP, IPv6CP, OSINLCP, or MPLS • open–Interface or protocol is administratively enabled • closed–Interface or protocol is administratively disabled
Operational status (Interface)	Indicates the operational state of the PPP interface • up–Interface is operational • down–Interface is not operational because of a problem in the PPP layer • lowerDown–Interface is not operational because a lower layer in the protocol stack is down • notPresent–Interface is not operational because the hardware is unavailable • passive–Interface is waiting for the peer to send an LCP confReq message • tunnel–Interface is being redirected through a tunnel
Operational status (Ip, Ipv6, Osi, Mpls)	Indicates the operational state of the IPCP, IPv6CP, OSINLCP, or MPLS protocol • up–Protocol is operational • down–Protocol is not operational because of a problem in the PPP layer • notPresent–Protocol is not operational because it does not exist • noResources–Protocol is not operational because it does not exist due to a lack of resources
PPP multilink status	Multilinked PPP interfaces

CHAPTER 11

Configuring Multiclass Multilink PPP

- Multiclass MLPPP Overview on page 361
- Multiclass MLPPP Traffic Classes Overview on page 362
- Multiclass MLPPP LCP Extensions Overview on page 363
- Multiclass MLPPP Platform Considerations on page 363
- Multiclass MLPPP References on page 364
- Configuring Multiclass MLPPP on page 364
- Enabling Multiclass MLPPP on page 365
- Configuring Traffic Classes on Multiclass MLPPP Interfaces on page 366
- Configuring Fragmentation on Multiclass MLPPP Interfaces on page 366
- Configuring Reassembly on Multiclass MLPPP Interfaces on page 367
- Example: Configuring Multiclass MLPPP on a Dynamic Interface on page 368
- Example: Configuring Multiclass MLPPP on a Static Interface on page 369
- Monitoring Multiclass MLPPP on page 369

Multiclass MLPPP Overview

Multiclass MLPPP enables the fragmentation of data packets of different priorities into multiple classes in an MLPPP bundle. It enables you to interleave data packets of higher priority between packets of lower priority before transmission.

When the MLPPP bundle consists of more than one multilink interface, multiclass MLPPP ensures the delivery of high-priority, delay-sensitive traffic, such as voice and video, in the proper sequence. Multiclass MLPPP accomplishes this by creating separate transmit and receive contexts for every multilink class in the MLPPP bundle. The transmit and receive contexts contain the sequence numbers and all other statistics pertaining to the multilink class

Multiclass MLPPP Fragmentation and Reassembly

You can configure fragmentation and reassembly on a multiclass MLPPP interface. Fragmentation is the process by which a large packet is broken up into multiple smaller fragments for simultaneous transmission across multiple links of an MLPPP bundle. Reassembly is the process by which the destination router reassembles the fragments into the original packets.

On MLPPP bundles that consist of physical links of different types, MLPPP does not guarantee the receipt of high-priority data packets in sequence. Multiclass MLPPP enables you to fragment data packets of different priorities into multiple multilink classes. Because every multilink class has its own transmit and receive context, data packets of each class are received in the same sequence they were transmitted.

With multiclass MLPPP, data packets of each multilink class are encapsulated in an MLPPP header. The sequence numbers of each class are also embedded within the header before transmission. The receiving peer processes each class independently and uses the sequence numbers in the MLPPP header to internally reorder and reassemble packets in the desired sequence.

Multiclass MLPPP Configuration Guidelines

Use the following guidelines while configuring multiclass MLPPP:

- Configure multiclass MLPPP on each link in the MLPPP bundle. If any link is not configured, the receiving peer might prevent the mismatched link from joining the bundle.
- The first link to join a bundle determines whether multiclass MLPPP is configured on the bundle. All subsequent links must also negotiate the same multiclass MLPPP parameters as that of the first link. The configuration for each link in a bundle is identical.

Related Documentation

- Configuring Multiclass MLPPP on page 364
- Configuring Multilink PPP on page 315

Multiclass MLPPP Traffic Classes Overview

A traffic class is a system-wide collection of buffers, queues, and bandwidth that you can allocate to provide a defined level of service to packets in the traffic class. With multiclass MLPPP, high-priority and low-priority data packets are fragmented into their respective QoS traffic classes before being transmitted. The QoS traffic classes are each mapped to a separate multilink class.

The major benefits of mapping traffic classes to multilink classes are:

- The multiclass MLPPP feature supports the mapping of up to eight traffic classes. You can fragment data packets into a maximum of eight different priorities of traffic classes.
- Classes of higher-priority can be interleaved between classes of lower priority, which reduces transmission latency.
- Every multilink class has its own transmit and receive context. These contexts ensure that data packets of higher priority traffic classes are received in the order they were transmitted.

The default traffic class is the best-effort traffic class. You can configure fragmentation and reassembly on all traffic classes. Any packet without a traffic-class-to-multilink-class mapping is transmitted without a multiclass MLPPP header.

Related Documentation

- Configuring Traffic Classes on Multiclass MLPPP Interfaces on page 366
- Multiclass MLPPP Overview on page 361
- *Traffic Class and Traffic-Class Groups Overview*

Multiclass MLPPP LCP Extensions Overview

Link Control Protocol (LCP) establishes an MLPPP link by negotiating LCP options with the MLPPP peer at the receiving end of a proposed connection. Peers negotiate multiclass MLPPP during the initial phase of the LCP option negotiation.

LCP uses the following new LCP configuration options for establishing a multiclass MLPPP link:

- Multilink header format—Multilink header format has three functions:
  - It informs the receiving peer that the multiclass MLPPP feature is supported.
  - It informs the peer of the multilink header format that will be used.
  - It informs the peer of the number of multilink classes configured.

  The router currently supports only the multilink header format option. The two number formats that are negotiated for the multilink header format option are:

  - Short sequence number
  - Long sequence number—The router currently supports only long sequence numbers
- Prefix elision—Prefix elision informs the receiving peer that only packets with a certain prefix must be accepted. This prefix is not transmitted as part of the information in the fragments of a multilink class. When the prefix is negotiated, all multilink packets must be transmitted with the negotiated prefix removed from the start of the packet. The prefix is empty for all classes, by default. The router currently does not support the prefix elision option.

Related Documentation

- Multiclass MLPPP Overview on page 361
- Configuring Multiclass MLPPP on page 364
- Understanding PPP Link Control Protocol on page 263

Multiclass MLPPP Platform Considerations

You can configure multiclass MLPPP on the following E Series Broadband Services Routers:

- E120 router
- E320 router

Module Requirements

For information about modules that support multiclass MLPPP on the E120 and E320 Broadband Services Routers:

- See *E120 and E320 Module Guide, Table 1, Module and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support multiclass MLPPP.

Interface Specifiers

For E120 and E320 routers, use the *slot/adapter/port* format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies a gigabitEthernet interface on slot 3, adapter 0, port 0 of an E320 router.

host1(config)#**interface gigabitEthernet 3/0/0**

Related Documentation

- Configuring Multiclass MLPPP on page 364
- Multiclass MLPPP Overview on page 361
- *Interface Types and Specifiers* in *JunosE Command Reference Guide*

Multiclass MLPPP References

For more information about multiclass MLPPP, see the following resources:

- RFC 1990—The PPP Multilink Protocol (MP) (August 1996)
- RFC 2686—The Multi-Class Extension to Multi-Link PPP (September 1999)

Related Documentation

- Configuring Multiclass MLPPP on page 364
- Multiclass MLPPP Overview on page 361

Configuring Multiclass MLPPP

When you configure multiclass MLPPP, you enable the creation of multilink classes on the MLPPP interface and assign QoS traffic classes to the multilink classes. You can configure multiclass MLPPP on a static MLPPP interface or in a dynamic profile for a dynamic MLPPP interface.

To configure multiclass MLPPP:

1. Enable multiclass MLPPP.

   See "Enabling Multiclass MLPPP" on page 365.

2. Configure traffic classes on the multiclass MLPPP interface.

   See "Configuring Traffic Classes on Multiclass MLPPP Interfaces" on page 366.

3. (Optional) Configure fragmentation on the multiclass MLPPP interface.

   See "Configuring Fragmentation on Multiclass MLPPP Interfaces" on page 366.

4. (Optional) Configure reassembly on the multiclass MLPPP interface.

   See "Configuring Reassembly on Multiclass MLPPP Interfaces" on page 367.

Related Documentation

- Multiclass MLPPP Overview on page 361
- Example: Configuring Multiclass MLPPP on a Dynamic Interface on page 368
- Example: Configuring Multiclass MLPPP on a Static Interface on page 369

Enabling Multiclass MLPPP

You can enable multiclass MLPPP on an MLPPP interface and create a specified number of multilink classes on it. A maximum number of eight multilink classes are supported per MLPPP interface.

Before you enable multiclass MLPPP:

- Define the QoS traffic classes

  See *Configuring Traffic Classes That Define Service Levels*.

To enable multiclass MLPPP and create a specified number of multilink classes on a dynamic MLPPP interface:

1. From Global Configuration mode, create an MLPPP dynamic profile by assigning it a name.

   host1(config)#**profile ppp-multilink-profile**

2. Enable multiclass MLPPP and specify the number of multilink classes to create.

   host1(config-profile)#**ppp multilink multiclass multilink-classes 8**

To enable multiclass MLPPP and create a specified number of multilink classes on a static MLPPP interface:

1. From Global Configuration mode, specify the individual interface on which you want to configure multiclass MLPPP.

   host1(config)#**interface gigabitEthernet 3/0/0**

2. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#**encapsulation pppoe**

3. Create a PPPoE subinterface.

host1(config)#**interface gigabitEthernet 3/0/0.10**

4. Specify MLPPP as the encapsulation method on the subinterface.

   host1(config-if)#**encapsulation mlppp**

5. Enable multiclass MLPPP and specify the number of multilink classes to create.

   host1(config-if)#**ppp multilink multiclass multilink-classes 8**

Related Documentation

- Dynamic Interface Configuration Using a Profile on page 565
- *ppp multilink multiclass*

Configuring Traffic Classes on Multiclass MLPPP Interfaces

You can configure mapping of QoS traffic classes to multilink classes. The **best-effort** QoS traffic class is mapped to multilink class 0 by default.

To configure mapping of QoS traffic classes to multilink classes:

- Specify the QoS traffic classes to be mapped to the multilink classes.

  To map QoS traffic classes to multilink classes in a dynamic profile:

  host1(config-profile)#**ppp multilink multiclass traffic-class best-effort voice otherData video**

  To map QoS traffic classes to multilink classes in a static MLPPP interface:

  host1(config-if)#**ppp multilink multiclass traffic-class best-effort voice otherData video**

NOTE: You must include the **best-effort** traffic class in the **ppp multilink multiclass traffic-class** command, or the command fails.

Related Documentation

- Configuring Upper-Layer Dynamic Interfaces on page 519
- *ppp multilink multiclass traffic-class*

Configuring Fragmentation on Multiclass MLPPP Interfaces

You can configure fragmentation on a multiclass MLPPP interface with the same fragment size as that of the MLPPP interface.

Before you configure fragmentation on the multiclass MLPPP interface:

- Ensure that fragmentation is enabled on the MLPPP interface.

  See "MLPPP Fragmentation and Reassembly" on page 332.

To configure fragmentation on a multiclass MLPPP interface:

- Specify the QoS traffic classes to be fragmented.

  To configure fragmentation on multilink classes in a dynamic profile:

  host1(config-profile)#**ppp multilink multiclass fragmentation best-effort voice otherData video**

  To configure fragmentation on multilink classes in a static MLPPP interface:

  host1(config-if)#**ppp multilink multiclass fragmentation best-effort voice otherData video**

  The order of the QoS traffic classes does not affect the execution of the command.

  NOTE: You must include the **best-effort** traffic class in the **ppp multilink multiclass fragmentation** command, or the command fails.

Related Documentation

- Configuring Upper-Layer Dynamic Interfaces on page 519
- *ppp multilink multiclass fragmentation*

Configuring Reassembly on Multiclass MLPPP Interfaces

You can configure reassembly on a multiclass MLPPP interface with the same maximum received reconstructed unit (MRRU) value as that of the MLPPP interface.

Before you configure reassembly on the multiclass MLPPP interface:

- Ensure that reassembly is enabled on the MLPPP interface.

  See *Configuring MLPPP Fragmentation and Reassembly* in *Chapter 8, Configuring Multilink PPP*.

To configure reassembly on a multiclass MLPPP interface:

- Specify the QoS traffic classes to be reassembled.

  To configure reassembly on multilink classes in a dynamic profile:

  host1(config-profile)#**ppp multilink multiclass reassembly best-effort voice otherData video**

  To configure reassembly on multilink classes in a static MLPPP interface:

  host1(config-if)#**ppp multilink multiclass reassembly best-effort voice otherData video**

  The order of the QoS traffic classes does not affect the execution of the command.

NOTE: You must include the **best-effort** traffic class in the **ppp multilink multiclass reassembly** command, or the command fails.

Related Documentation

- Configuring Upper-Layer Dynamic Interfaces on page 519
- *ppp multilink multiclass reassembly*

Example: Configuring Multiclass MLPPP on a Dynamic Interface

The following example shows how to configure multiclass MLPPP on a dynamic MLPPP interface. To configure multiclass MLPPP you first define the traffic classes that need to be mapped to the multilink classes. You configure multiclass MLPPP in the dynamic profile and optionally configure fragmentation and reassembly on the multiclass MLPPP interface.

1. Define the QoS traffic classes.

   host1 (config)#**traffic-class voice**
   host1 (config-traffic-class)#**fabric-strict-priority**
   host1 (config)#**traffic-class low-loss**
   host1 (config-traffic-class)#**fabric-strict-priority**
   host1 (config)#**traffic-class low-latency**
   host1 (config-traffic-class)#**fabric-strict-priority**

   For more information about defining QoS traffic classes, see *Traffic Class and Traffic-Class Groups Overview*.

2. Create a dynamic profile.

   host1(config)#**profile ppp-multilink-dynamic-profile**

3. Configure multiclass MLPPP in the dynamic profile.

   host1 (config-profile)#**ppp multilink multiclass multilink-classes 4**
   host1(config-profile)#**ppp multilink multiclass traffic-class best-effort voice low-loss low-latency**

   For more information about configuring a dynamic profile for multiclass MLPPP, see *Configuring a Profile* in *Chapter 19, Configuring Dynamic Interfaces*.

4. Configure fragmentation and reassembly on the multiclass MLPPP interface.

   host1(config-profile)#**ppp multilink multiclass fragmentation best-effort voice low-loss low-latency**

   host1(config-profile)#**ppp multilink multiclass reassembly best-effort voice low-loss low-latency**

Related Documentation

- Configuring Multiclass MLPPP on page 364

Example: Configuring Multiclass MLPPP on a Static Interface

The following example shows how to configure multiclass MLPPP on a static MLPPP interface. To configure multiclass MLPPP you first define the traffic classes that need to be mapped to the multilink classes. You configure multiclass MLPPP on the static MLPPP interface and optionally configure fragmentation and reassembly on the interface.

1. Define the QoS traffic classes.

   host1 (config)#**traffic-class video**
   host1 (config-traffic-class)#**fabric-strict-priority**
   host1 (config)#**traffic-class low-loss**
   host1 (config-traffic-class)#**fabric-strict-priority**

   For more information about QoS traffic classes, see *Traffic Class and Traffic-Class Groups Overview*.

2. Specify the interface and the encapsulation method on which you want to configure MLPPP

   host1(config)#**interface gigabitEthernet 3/0/0**
   host1(config-if)#**encapsulation pppoe**
   host1(config)#**interface gigabitEthernet 3/0/0.10**
   host1(config-if)#**encapsulation mlppp**

3. Configure multiclass MLPPP on the static interface.

   host1(config-if)#**ppp multilink multiclass multilink-classes 3**
   host1(config-if)#**ppp multilink multiclass traffic-class best-effort video low-loss**

4. Configure fragmentation and reassembly on the multiclass MLPPP interface.

   host1(config-if)#**ppp multilink multiclass fragmentation best-effort video low-loss**
   host1(config-if)#**ppp multilink multiclass reassembly best-effort video low-loss**

Related Documentation

- Configuring Multiclass MLPPP on page 364

Monitoring Multiclass MLPPP

Purpose Display information about the status and configuration of multilink classes on an MLPPP Interface.

Action To display configuration information about multiclass MLPPP member links configured in the specified MLPPP bundle:

```
host1#show ppp interface mlppp bundle1 config
PPP interface mlppp mlppp1 is up
...

PPP multilink member-interface gigabitEthernet 3/0.1.1 is up
Link interface administrative status is open
PPP multilink multiclass is enabled
PPP multilink multiclass classes 4
PPP multilink multiclass fragmentation is enabled on “voice”, otherData”
PPP multilink multiclass reassembly is enabled on “voice”, ”otherData”
...
```



```
PPP multilink member-interface gigabitEthernet 3/0.1.2 is up
Link interface administrative status is open
PPP multilink multiclass is enabled
PPP multilink multiclass classes 4
PPP multilink multiclass fragmentation is enabled on “voice”,”otherData”
PPP multilink multiclass reassembly is enabled on “voice”, ”otherData”
...
1 mlppp interfaces found
```

To display the statistics of the multiclass MLPPP member links configured in the specified MLPPP bundle:

```
host1#show ppp interface mlppp bundle1 statistics
PPP interface mlppp bundle1 is up
Time since last baseline 2 days, 18 hours
2 receive classes, 2 transmit classes
Interface statistics            in                  out
   packets                      0                   0
   octets                       0                   0
   errors                       0                   0
   discards                     0                   0
   fragments                    0                   0

Receive Class 0:
0 fragments in reassembly list
0 reordered, 0/0 discarded fragments/bytes,
0x0 last received sequence number

Receive Class 1:
0 fragments in reassembly list
0 reordered, 0/0 discarded fragments/bytes,
0x0 last received sequence number

Transmit Class 0:
0 fragments, 0x8 last sent sequence number

Transmit Class 1:
0 fragments, 0x0 last sent sequence number
...
1 mlppp interfaces found
```

To display complete configuration, statistics, and status information of multiclass MLPPP member links configured in the specified MLPPP bundle:

```
host1#show ppp interface mlppp bundle1 full
Bundle name: bundle1
MLPPP interface gigabitEthernet 3/0.1.1 is up
Link interface administrative status is open
PPP multilink multiclass is enabled
PPP multilink multiclass classes 4
PPP multilink multiclass fragmentation is enabled on “voice”, otherData”
PPP multilink multiclass reassembly is enabled on “voice”, ”otherData”
...
LCP negotiated options          local              peer
   max-receive-unit             1590               1590
   max-receive-reconstructed-unit 1590             1590
   authentication               none               none
   magic-number                 0x78b5606f         0x5aa2de54
   pfc                          none               none
   acfc                         none               none
```



```
   multiclass-classes           4                    4
   multiclass-sequence-format   long                 long
...
1 mlppp interfaces found
```

Meaning Table 31 on page 371 lists the **show ppp interface mlppp** command output fields.

Table 31: show ppp interface mlppp Output Fields

Field Name	Field Description
PPP interface mlppp *bundleName*	Name and state (up or down) of the MLPPP interface
PPP multilink member-interface *interfaceName*	Name and state (up or down) of the MLPPP member link interface
Link interface administrative status	Administrative state of the member link interface: open (enabled) or closed (disabled)
PPP multilink multiclass	Configuration of multiclass MLPPP on the MLPPP interface: enabled or disabled
PPP multilink multiclass classes	Number of multilink classes created on the MLPPP interface: 1 through 8
PPP multilink multiclass fragmentation	Configuration of fragmentation of the multiclass MLPPP interface: enabled or disabled. If fragmentation is enabled then the command also lists the QoS traffic classes on which fragmentation is enabled.
PPP multilink multiclass reassembly	Configuration of reassembly of the multiclass MLPPP interface: enabled or disabled. If reassembly is enabled then it also lists the QoS traffic classes on which reassembly is enabled.
mlppp interfaces found	Number of MLPPP interfaces configured
Time since last baseline	When baselining is configured, the time since the last baseline was set is displayed in *hours:minutes:seconds* or *days/hours* format
receive class, transmit class	Number of multilink classes configured on the MLPPP bundle

Table 31: show ppp interface mlppp Output Fields *(continued)*

Field Name	Field Description
Interface statistics	Statistics for data received by (in) and transmitted (out) on the MLPPP interface: • packets—Number of packets received and transmitted on the interface • octets—Number of octets received and transmitted on the interface • errors—Number of errors received and transmitted on the interface • discards—Number of packets discarded on receipt or discarded before they were transmitted • fragments—Number of fragments received and transmitted on the interface
Receive Class *number*	Information about the packets of the specified multilink class received by the router
fragments in reassembly list	Number of buffered fragments reassembled
reordered	Number of fragments reordered
discarded fragments/bytes	Number of fragments and bytes that have been discarded.
last received sequence number	Sequence number of the last multiclass MLPPP packet received
Transmit Class *number*	Information about the packets of the specified multilink class transmitted by the router
fragments	Number of fragments that are sent by the router
last sent sequence number	Sequence number of the last multiclass MLPPP packet sent
Bundle name	Name of the MLPPP bundle
MLPPP interface *interfaceName*	Name and state (up or down) of the configured MLPPP interface

Table 31: show ppp interface mlppp Output Fields *(continued)*

Field Name	Field Description
LCP negotiated options	Negotiated LCP options for the local and peer systems: • max-receive-unit—Negotiated maximum receive unit, in octets, for the local and remote (peer) side of the link • max-receive-reconstructed-unit—Negotiated maximum receive reconstructed unit, in octets, for the local and remote (peer) side of the link • authentication—Negotiated authentication method (none, pap, or chap) for the local and remote (peer) side of the link • magic-number—Negotiated magic number for the local and remote (peer) side of the link • pfc—Negotiated protocol field compression (none or enabled) for the local and remote (peer) side of the link • acfc—Negotiated address and control field compression (none or enabled) for the local and remote (peer) side of the link • multiclass-classes—Number of multilink classes negotiated • multiclass-sequence-format—Format of the negotiated sequence number: long or short

Related Documentation

- *show ppp interface*

CHAPTER 12

Configuring Packet over SONET

Use the procedures described in this chapter to configure packet over SONET (POS) on E Series routers.

This chapter contains the following sections:

- Overview on page 375
- Platform Considerations on page 376
- References on page 377
- Before You Configure POS on page 378
- Configuration Tasks on page 378
- Monitoring POS on page 382

Overview

Packet over SONET (Synchronous Optical Network)/SDH (Synchronous Digital Hierarchy) is the serial transmission of data over SONET frames through the use of a protocol such as Point-to-Point Protocol (PPP).

Packet over SONET/SDH is an ideal feature for networks that are built for providing Internet or IP data. It provides superior bandwidth utilization and efficiency compared with other transport methods. For expensive WAN links, packet over SONET can provide as much as 25 to 30 percent higher throughput than networks based on Asynchronous Transfer Mode (ATM). By transporting frames directly into the SONET/SDH payload, the overhead required in an ATM cell header for IP over ATM encapsulation is eliminated.

The router supports PPP, Cisco High-Level Data Link Control (HDLC), and Frame Relay over SONET/SDH.

POS Features

POS supports the following features:

- Payload scrambling
- Clock source configuration
- Maximum transmission unit (MTU) size configuration
- Maximum receive unit (MRU) size configuration

- POS framing
- Cyclic redundancy check (CRC) checking
- Loopback configuration

SONET/SDH

SONET is an ANSI standard for transmitting bits over fiber-optic cable. SDH is the international standard defined by the International Telecommunication Union (ITU). SONET/SDH is the physical infrastructure of choice for carrier ATM networks operating at speeds above 50 Mbps.

SONET/SDH allows carriers to build high-speed international links without requiring conversion from one transmission protocol to another (for example, T1 to T3 or T1 to E3 conversion).

SONET transmission speeds start at 51.84 Mbps and are referred to as OC1. SDH transmission speeds start at 155.52 Mbps and are referred to as STM1. All other speeds are multiples of these base numbers. Table 32 on page 376 shows the speeds of the most common SONET/SDH implementations.

Table 32: Most Common SONET/SDH Implementations

SONET	SDH	Transmission Speed
OC1	—	51.84 Mbps
OC3	STM1	155.52 Mbps
OC12	STM4	622.08 Mbps
OC48	STM16	2.4 Gbps
OC96	STM32	4.876640 Gbps
OC192	STM64	9.953280 Gbps

Platform Considerations

You can configure POS interfaces on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router

- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support POS interfaces on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support POS.

For information about the modules that support POS interfaces on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support POS.

Interface Specifiers

The configuration task examples in this chapter use the *slot/port*[.*subinterface*] format to specify a POS interface. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format. For example, the following command specifies a POS interface on slot 0, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

```
host1(config)#interface pos 0/1
```

For E120 and E320 routers, use the *slot/adapter/port* format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies a POS interface on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface pos 5/0/0
```

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

References

For more information about POS interfaces, consult the following resources:

- RFC 1662—PPP in HDLC-like Framing (July 1994)
- RFC 2615—PPP over SONET/SDH (June 1999

- RFC 2427—Multiprotocol Interconnect over Frame Relay (September 1998)

Before You Configure POS

Before you configure a POS interface, verify that you have correctly installed the required module. For information about installing modules in ERX7xx models, ERX14xx models, and ERX310 router, see *ERX Hardware Guide, Chapter 4, Installing Modules*. For information about installing modules in the E120 and E320 routers, see *E120 and E320 Hardware Guide*, *Chapter 4*, *Installing Modules*. Then verify that no ATM interfaces are defined on the physical port.

Also have the following information available:

- Interfaces specifiers for the POS interfaces that you want to create

  For more information about specifying POS interfaces on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

- IP addresses and subnet mask assignments for IP interfaces

Configuration Tasks

To configure a POS interface:

1. Configure a physical interface.

   host1(config)#**interface pos 0/1**

2. (Optional) Assign a text description or an alias to the interface.

   host1(config-if)#**pos description austin01 pos interface**

3. Configure the encapsulation method.

   host1(config-if)#**encapsulation ppp**

4. (Optional) Configure the internal clock source.

   host1(config-if)#**clock source internal module**

5. (Optional) Set the size of the CRC.

   host1(config-if)#**crc 32**

6. (Optional) Set the time interval at which the router calculates bit and packet rate counters.

   host1(config-if)#**load-interval 90**

7. (Optional) Set the type of loopback mode.

   host1(config-if)#**loopback line**

8. (Optional) Set the MRU size.

   host1(config-if)#**mru 1000**

9. (Optional) Set the MTU size.

   host1(config-if)#**mtu 1000**

10. (Optional) Set the type of framing.

    host1(config-if)#**pos framing sdh**

11. Disable payload scrambling.

    host1(config-if)#**no pos scramble-atm**

12. (Optional) Disable an interface.

    host1(config-if)#**shutdown**

clock source

- Use to set the clock source.
- You can set **internal** or **line** clocking.
- Internal clocking has two options:
  - **module**—Uses internal clock from the line module
  - **chassis**—Uses the configured router clock
- Example

  host1(config-if)#**clock source internal module**
- Use the **no** version to restore the default value, **line**.
- See *clock source*.

crc

- Use to set the number of bits used for CRC checking.
- CRC is an error-checking technique that uses a calculated numeric value to detect errors in transmitted data; 16 and 32 indicate the number of check digits per frame that are used to calculate the frame check sequence (FCS). Both the sender and receiver must use the same setting.
- Example

  host1(config-if)#**crc 32**
- Use the **no** version to restore the default value, 16.
- See *crc*.

encapsulation frame-relay ietf

- Use to specify Frame Relay as the encapsulation method for the interface.
- The router uses IETF format (RFC 2427 encapsulation).
- Example

  host1(config-if)#**encapsulation frame-relay ietf**
- Use the **no** version to remove the Frame Relay configuration from an interface.
- See *encapsulation frame-relay ietf*

encapsulation ppp

- Use to specify PPP as the encapsulation method for the interface.
- Example

  host1(config-if)#**encapsulation ppp**
- Use the **no** version to remove the PPP configuration from an interface.
- See *encapsulation ppp*.

interface pos

- Use to configure a POS interface.
- To specify a POS interface for ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface
- To specify a POS interface for E120 and E320 routers, use the *slot/adapter/port* format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
- For more information about modules that support POS interfaces, see *Configuring Unchannelized OCx/STMx Interfaces* in *JunosE Physical Layer Configuration Guide*.
- Examples

  host1(config-if)#**interface pos 0/1**
  host1(config-if)#**interface pos 5/0/0**
- Use the **no** version to remove the POS interface.
- See *interface pos*.

load-interval

- Use to set the time interval at which the router calculates bit and packet rate counters.
- You can choose a multiple of 30 seconds, in the range 30–300 seconds.
- Example

  host1(config-if)#**load-interval 90**
- Use the **no** version to restore the default value, 300.
- See *load-interval*.

loopback

- Use to specify the type of loopback for a POS interface.
  - **internal**—Connects the local transmitted signal to the local received signal.
  - **line**—Connects the received network signal directly to the transmit network signal. When configured in line loopback mode, the router never receives data from the network.
- Example

  host1(config-if)#**loopback line**
- Use the **no** version to clear the loopback.
- See *loopback*.

mru

- Use to set the maximum allowable size of the MRU.
- Specify a value in the range 1–9996 bytes.
- Example

  host1(config-if)#**mru 1000**
- Use the **no** version to restore the default value, 4470.
- See *mru*.

mtu

- Use to set the maximum allowable size of the MTU.
- Specify a value in the range 1–9996 bytes.
- Example

  host1(config-if)#**mtu 1000**
- Use the **no** version to restore the default value, 4470.
- See *mtu*.

pos description

- Use to assign a text description or an alias to a POS HDLC interface.
- You can use this command to help you identify the interface and keep track of interface connections.
- The description or alias can be a maximum of 80 characters.
- Use "show interfaces pos" on page 383 to display the text description.
- Example

  host1(config-if)#**pos description austin01 pos interface**
- Use the **no** version to remove the text description or alias.
- See *pos description*.

pos framing

- Use to set the type of framing for a POS interface.
  - **sdh**—Uses SDH framing format
  - **sonet**—Uses SONET framing format (the default)
- Example

  host1(config-if)#**pos framing sdh**
- Use the **no** version to restore the default value, **sonet**.
- See *pos framing*.

pos scramble-atm

- Use to enable payload scrambling on a POS interface.
- Payload scrambling is enabled by default. When enabled, both sides of the connection must be using the scrambling algorithm. The router uses a 43rd-order synchronous scrambler to scramble the output data.
- Example

  host1(config-if)#**pos scramble-atm**
- Use the **no** version to disable scrambling on the POS interface.
- See *pos scramble-atm*.

shutdown

- Use to disable a POS interface.
- Example

  host1(config-if)#**shutdown**
- Use the **no** version to restart a disabled interface.
- See *shutdown*.

Monitoring POS

Use the **show interfaces pos** command to display information about the POS interface. You can set a statistics baseline for POS interfaces using the **baseline interface pos** command.

You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *show Commands* in *JunosE System Basics Configuration Guide*, for details.

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

baseline interface pos

- Use to set a statistics baseline for POS interfaces. The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Example

  host1#**baseline interface pos 8/0**
- There is no **no** version.
- See *baseline interface*.

show interfaces pos

- Use to display the configuration, state, and statistics for a POS interface.
- To specify a POS interface for ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface
- To specify a POS interface for E120 and E320 routers, use the *slot/adapter/port* format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
- You can include the following keywords:
  - **delta**—Specifies that baselined statistics are to be shown
  - **brief**—Displays the operational status of all configured interfaces
- Field descriptions
  - POS interface status—State of the physical interface: up, down
  - Description—Text description or alias if configured for the interface
  - snmp trap link-status—SNMP trap status: disabled: up, down

- Encapsulation—Layer 2 encapsulation display; options: ppp, frame-relay ietf, mlppp, mlframe-relay ietf, hdlc
- SONET path operational status—State of the SONET path: up, down, lowerLayerDown
- time since last status change—Last reported change to the SONET path operational status
- SONET operational status—State of SONET operation: up, down, lowerLayerDown
- time since last status change—Last reported change to the SONET operational status
- loopback—Loopback status for the physical interface: enabled, disabled
- timing source—Clocking source for the physical interface
- framing type—Framing type for the physical interface
- Crc type checking—Number of bits used for CRC checking: crc16, crc32, none
- Hdlc mru—MRU size allowed on the interface
- Hdlc mtu—MTU size allowed on the interface
- Hdlc interface speed—Line speed of the interface
- Hdlc scrambling—Status of payload scrambling on the interface: on, off
- 5 minute input rate—Data rates based on the traffic received in the last five minutes
- 5 minute output rate—Data rates based on the traffic sent in the last five minutes
- Packets received—Number of incoming packets received on this interface
- Bytes received—Number of incoming bytes received on this interface
- Errored packets received—Number of incoming errors received on this interface
- Packets sent—Number of outgoing packets transmitted on this interface
- Bytes sent—Number of outgoing bytes transmitted on this interface
- Errored packets sent—Number of outgoing errors on this interface

- Example

```
host1#show interfaces pos 8/0
Packet over SONET interface 8/0 is ifOperUp
Description: houston80 pos interface
snmp trap link-status = disabled
Encapsulation ppp
SONET path operational status: up
      time since last status change: 00:20:37
SONET operational status:      up
      time since last status change: 00:20:37
loopback not set
timing source is loop timing
framing type is SONET
Crc type checking - CRC32
Hdlc mru = 4470
Hdlc mtu = 4470
```



```
Hdlc interface speed = 155520000
Hdlc scrambling is off
5 minute input rate 24910848 bits/sec, 1023242 packets/sec
5 minute output rate 24905728 bits/sec, 1023233 packets/sec

Interface statistics
Packets received             1066995954
Bytes received               3836558195
Errored packets received     0
Packets sent                 1055275550
Bytes send                   3039550548
Errored packets sent         0
```

- See *show interfaces*.

CHAPTER 13

Configuring Point-to-Point Protocol over Ethernet

This chapter describes how to configure the Point-to-Point Protocol (PPP) over Ethernet interfaces on E Series routers.

This chapter contains the following sections:

- Understanding PPPoE on page 388
- PPPoE Service Name Tables Overview on page 390
- Identification of Subscribers Using the PPPoE Remote Circuit Identifier on page 392
- PPPoE Platform Considerations on page 397
- PPPoE References on page 398
- Access Nodes in Ethernet Aggregation Networks Overview on page 399
- ATM-to-Ethernet Interworking Overview on page 400
- Configuring PPPoE over ATM on page 402
- Guidelines for Overriding the PPPoE Maximum Session Value on page 403
- Overview of IWF PPPoE Sessions with Duplicate MAC Addresses on page 404
- Guidelines for Configuring Duplicate Protection for IWF PPPoE Sessions on page 405
- Single DSLAM Connected to a PPPoE Access Concentrator on page 406
- Multiple DSLAMs Connected to a PPPoE Access Concentrator on page 407
- PPPoE with Ethernet Modules Configuration Overview on page 408
- Configuring IPv4 and IPV6 over static PPPoE with VLANs on page 410
- Configuring IPv4 over PPPoE Without VLANs on page 411
- PADM Messages Overview on page 412
- Configuring MOTM Messages from Privileged Exec Mode on page 413
- Configuring MOTM Messages from Interface Configuration Mode on page 413
- Configuring MOTM Messages from Profile Configuration Mode on page 414
- Configuring URL Messages from Interface Configuration Mode on page 414
- Configuring URL Messages from Profile Configuration Mode on page 415
- PADN Messages Overview on page 415

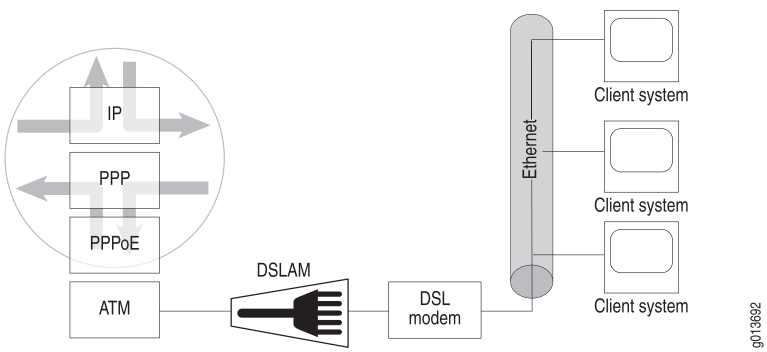
- Configuring PADN Messages on page 416
- Creating and Populating PPPoE Service Name Tables on page 416
- Enabling Usage of PPPoE Service Name Tables with Static Interfaces over ATM on page 418
- Enabling Usage of PPPoE Service Name Tables with Static Interfaces over Ethernet on page 419
- Enabling Usage of PPPoE Service Name Tables with Dynamic Interfaces on page 419
- Configuring PADS Packet Content on page 420
- Configuring PPPoE Remote Circuit ID Capture on page 421

Understanding PPPoE

E Series routers use PPP over Ethernet (PPPoE) to enable multiple hosts to open PPP sessions to the router using one or more bridging modems. When service providers want to maintain the session abstraction associated with PPP, PPPoE is used with Broadband Remote Access Server (B-RAS) technologies that provide a bridged Ethernet topology. PPPoE can be configured over ATM or on Ethernet modules with or without VLANs.

Figure 37 on page 388 shows how PPPoE allows the router to handle multiple PPP sessions originating on an Ethernet module to be multiplexed over one PVC on an ATM interface. PPP, as described in "Configuring Point-to-Point Protocol" on page 261, runs above the PPPoE layer.

Figure 37: PPPoE over ATM



The router handles the server part of PPPoE session management and never initiates a setup of a PPPoE session. The router only responds to session requests that are sent to it by the remote PPP client. After the sessions are set up, the router demultiplexes the sessions based on session identifiers assigned to a specific connection.

PPPoE Stages

PPPoE has two distinct stages: Discovery and Session.

Discovery

PPPoE includes a Discovery protocol that allows each PPP session to learn the Ethernet address of the remote peer, as well as establish a unique session identifier. When a host wants to initiate a PPPoE session, it must first perform Discovery to identify the Ethernet MAC address of the peer and establish a PPPoE session ID.

Although PPP defines a peer-to-peer relationship, Discovery is inherently a client-server relationship. In the Discovery process, a host acting as a client discovers a remote access concentrator (AC), which acts as the server.

Based on the network topology, there may be more than one remote AC with whom the host can communicate. The Discovery stage allows the host to discover all remote ACs and then select the one to which it wants to connect.

In summary, the Discovery stage consists of the following four steps:

1. The host (PPPoE client) broadcasts a PPPoE Active Discovery Initiation (PADI) packet to all remote ACs in the network.
2. One or more remote ACs respond to the PADI packet by sending a PPPoE Active Discovery Offer (PADO) packet, indicating that they can serve the client request. The PADO packet includes the name of the AC from which it was sent.
3. The host sends a unicast PPPoE Active Discovery Request (PADR) packet to the AC to which it wants to connect.
4. The selected AC sends a PPPoE Active Discovery Session (PADS) packet to confirm the session.

Session

When Discovery is successfully completed, both the host and the selected remote AC have the information they need to build their point-to-point connection over Ethernet.

The only parameter that you can configure is the number of PPPoE sessions.

NOTE: The router supports dynamic PPPoE interfaces. Also, profiles support PPPoE interfaces. See "Configuring Upper-Layer Dynamic Interfaces" on page 519 and "Configuring Dynamic Interfaces Using Bulk Configuration" on page 619, for more information.

PPPoE MTU Configuration

To avoid fragmentation and reassembly, Ethernet access networks require larger MTU sizes for PPP traffic. With JunosE PPPoE MTU, you can control the deployment of larger packet sizes. You can configure PPPoE MTU directly on the PPPoE interface or use a dynamic configuration profile. When you use the PPPoE MTU tag, each PPPoE subinterface can have a unique MTU value. Operational MTU is the lesser of the PPPoE MTU or the lower layer MTU minus the PPPoE overhead.

You can use the **pppoe mtu** command to set the MTU using a combination of lower layer restrictions and controls:

- Greater MTU than the current maximum permitted by RFC 2516, with the default equal to the current maximum setting (1494 octets)
- Optional setting for absolute maximum PPPoE MTU
- Optional use of a larger lower layer MTU
- Optional use of the PPPoE-Max-Mtu tag transmitted from the client

Related Documentation

- PPPoE Service Name Tables Overview on page 390
- Setting a Baseline for PPPoE Interface Statistics on page 425
- Monitoring the PPPoE Interface on page 427
- Monitoring the PPPoE Subinterface on page 434
- Troubleshooting PPPoE Interfaces on page 441

PPPoE Service Name Tables Overview

PPPoE clients use service name tags, as defined in RFC 2516, to request that an AC support certain services. The client includes a custom service name tag in the PADI packet that it broadcasts to remote ACs. Alternatively, the client can include an empty service name tag of zero length to indicate that any service is acceptable, or an unknown service name tag to represent a service not yet configured in the PPPoE service name table.

On receipt of a PADI packet that it can serve, the AC responds with a PADO packet. The PADO packet contains a service name tag that is identical to the one in the PADI, as well as one or more additional service name tags indicating other services that the AC offers.

A PPPoE service name table consists of one or more service name entries and their associated action. The PPPoE service name table can include three types of service name tags:

- **Custom service name tag (*serviceName*)** — A service entry that represents a specific client service that an AC can support. The length of the custom service name tag can be up to 31 alphanumeric characters; for example, myQoSClass or my ISPService. You can optionally associate an action (**drop** or **terminate**) with a custom service. The default action for a custom service is **terminate**.
- **Empty service name tag (empty-service-name)** — A service entry of zero length that is used to represent any service. The router either responds with a PADO packet to all PADI requests containing an empty service name tag, or denies (drops) all PADI requests based on the action configured for the service.
- **Unknown service name tag (unknown-service-name)** — A service entry that has not been configured in the PPPoE service name table. When a PPPoE client includes an unknown service name tag in the PPPoE service name table, the router responds based on the action (**drop** or **terminate**) associated with the unknown service name entry.

  The default action associated with the unknown service name tag depends on the PPPoE service name table configuration. If all the services in the table are configured to **drop**, the default action for the unknown service name tag is **terminate**. If all the services in the table are configured to **terminate**, the default action for the unknown service name tag is **drop**. If both **terminate** and **drop** are configured for services in the table, all unknown service name tags are dropped by default.

Features

PPPoE service name tables enable an AC, such as an E Series router, to support multiple service name tags in addition to the empty service name tag and the unknown service name tag. You can configure up to 16 PPPoE service name tables per E Series router to:

- Define the set of service name tags (empty service name, custom service name, and unknown service name) that the router advertises in the PADO packets sent to PPPoE clients.
- Control whether the router responds to (**terminate**) or denies (**drop**) PADI requests based on the action associated with the service name tags.

Table Structure

Each entry, or row, in a PPPoE service name table consists of the following components:

- Service name tag—Service name tags specify the client services that an AC supports. You can add three types of service name tags to the PPPoE service name table: empty service name, custom service name (*serviceName*), and unknown service name. Every PPPoE service name table includes one empty service name tag and one unknown service name service tag. An empty service name tag is a service tag of zero length that is used to represent any service. An unknown service name service tag is used to represent a service tag that has not been configured in the service name table. In addition to these two tags, you can configure up to 16 custom service name tags per table.
- Action—Each service name tag has an associated action: **terminate** (the default action) or **drop**. For empty service name and unknown service name entries, you can use the **action** keyword with the **service** command to modify the default action associated with the service. For custom service name (*serviceName*) entries, using the **action** keyword with the **service** command is optional. The default action for a custom service tag entry is **terminate**.

For example, Table 33 on page 391 shows a PPPoE service name table containing five entries: three custom service name tags, two associated with the **terminate** action and one associated with the **drop** action; an empty service name tag (“ ”) associated with the **drop** action; and an unknown service name tag associated with the **drop** action.

Table 33: Sample PPPoE Service Name Table

Service Name	Action
“myISPService”	Terminate
“myQOSClass1”	Terminate
“myQOSClass2”	Drop

Table 33: Sample PPPoE Service Name Table *(continued)*

Service Name	Action
" " (empty-service-name)	Drop
unknown-service-name	Drop

NOTE: You can associate the **drop** action with a maximum of eight service tags in a PPPoE service name table.

Enabling the Service Name Table for Use

After you create a PPPoE service name table and populate it with entries, you must enable it for use with a static or dynamic PPPoE interface. To enable a PPPoE service name table for use with a static interface, you assign the table to the PPPoE major interface. To enable a PPPoE service name table for use with a dynamic interface, you add the table to a profile that is dynamically assigned to a PPPoE interface column.

Related Documentation

- Creating and Populating PPPoE Service Name Tables on page 416
- Enabling Usage of PPPoE Service Name Tables with Static Interfaces over ATM on page 418
- Enabling Usage of PPPoE Service Name Tables with Static Interfaces over Ethernet on page 419
- Enabling Usage of PPPoE Service Name Tables with Dynamic Interfaces on page 419
- Monitoring the PPPoE Service Name Table on page 433

Identification of Subscribers Using the PPPoE Remote Circuit Identifier

You can enable the router to capture and format a vendor-specific tag containing a PPPoE remote circuit ID transmitted from a digital subscriber line access multiplexer (DSLAM) device. The router can then send this value to a Remote Authentication Dial-In User Service (RADIUS) server or to a Layer 2 Tunneling Protocol (L2TP) network server (LNS) to uniquely identify subscriber locations.

This feature is supported on all modules on which you can configure PPPoE interfaces. The feature is particularly useful in Ethernet-based Broadband Remote Access Server (B-RAS) configurations as a means of uniquely identifying subscribers connected to the router on a single Ethernet link.

For detailed configuration instructions, see "Configuring PPPoE Remote Circuit ID Capture" on page 421.

Application

When a connection between an E Series router and a DSLAM is on an ATM interface, subscribers are typically assigned an ATM PVC to communicate with the router. Each ATM PVC is created on a different ATM 1483 subinterface. When a RADIUS server in this configuration sends messages to the router containing the NAS-Port-Id [87] RADIUS attribute, each ATM 1483 subinterface produces a unique NAS-Port-Id that can differentiate subscribers on the ATM link.

By contrast, when the connection between the router and the DSLAM is on an Ethernet interface that does not use either virtual LANs (VLANs) or stacked VLANs (S-VLANs), the NAS-Port-Id value is the same for all subscribers on the Ethernet link. Enabling the router to capture the remote circuit ID sent from the DSLAM and use it as a RADIUS or L2TP attribute facilitates the process of identifying individual subscribers on an Ethernet link.

PPPoE Remote Circuit ID Capture

When you enable capture of the PPPoE remote circuit ID by issuing the **pppoe remote-circuit-id** command, the E Series router captures the remote circuit ID value if it is sent from the DSLAM. The PPPoE intermediate agent on the DSLAM appends a vendor-specific tag containing the remote circuit ID to the existing PPPoE PADI or PADR packet and sends this packet to the E Series router. The PPPoE remote circuit ID value can be a maximum of 64 characters. The router stores this value on the line module on which the PPPoE interface is configured.

PPPoE Remote Circuit ID Format

By default, the router formats the captured PPPoE remote circuit ID to include only the agent-circuit-id suboption (suboption 1) of the PPPoE intermediate agent tags sent from the DSLAM. To configure a nondefault format for the captured PPPoE remote circuit ID, you can use one of the **radius remote-circuit-id-format** commands listed in Table 34 on page 393.

Table 34: Configuring Nondefault Formats for the PPPoE Remote Circuit ID

To Configure This Nondefault Format	Use This Command
Include only the agent-remote-id suboption (suboption 2) of the tags supplied by the PPPoE intermediate agent	host1(config)#**radius remote-circuit-id-format agent-remote-id**
Include both the agent-circuit-id suboption (suboption 1) and the agent-remote-id suboption (suboption 2) of the tags supplied by the PPPoE intermediate agent	host1(config)#**radius remote-circuit-id-format agent-circuit-id agent-remote-id**

Table 34: Configuring Nondefault Formats for the PPPoE Remote Circuit ID *(continued)*

To Configure This Nondefault Format	Use This Command
Include the NAS-Identifier [32] RADIUS attribute with either or both of the agent-circuit-id and agent-remote-id suboptions of the tags supplied by the PPPoE intermediate agent	host1(config)#**radius remote-circuit-id-format nas-identifier agent-circuit-id** or host1(config)#**radius remote-circuit-id-format nas-identifier agent-remote-id** or host1(config)#**radius remote-circuit-id-format nas-identifier agent-circuit-id agent-remote-id**
Append the agent-circuit-id suboption to an interface specifier that is consistent with the recommended format in the DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006). For details about how the router implements this format, see "Format for dsl-forum-1 Keyword" on page 394.	host1(config)#**radius remote-circuit-id-format dsl-forum-1**

For more information about configuring the format of the PPPoE remote circuit ID value, see *radius remote-circuit-id-format*.

Remote Circuit ID Delimiter

If the format of the PPPoE remote circuit ID consists of two or more components, the router uses a # character by default to delimit the components. Optionally, you can use the **radius remote-circuit-id-delimiter** command to configure a nondefault delimiter character (for example, ! or %) to separate multiple components in the PPPoE remote circuit ID value. For information about how to use this command, see *radius remote-circuit-id-delimiter*.

Format for dsl-forum-1 Keyword

When you specify the **radius remote-circuit-id-format** command with the **dsl-forum-1** keyword, the router appends the agent-circuit-id suboption value to an interface specifier that is consistent with the recommended format in the DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006).

The format of the PPPoE remote circuit ID when you use the **dsl-forum-1** keyword is as follows:

dslForum1InterfaceSpecifier#agent-circuit-id

where:

- *dslForum1InterfaceSpecifier* is the interface specifier in **dsl-forum-1** format

- # is the default delimiter character
- *agent-circuit-id* is the agent-circuit-id suboption (suboption 1) of the PPPoE intermediate agent tags sent from the DSLAM

If the DSLAM transmits empty data for *agent-circuit-id*, the router appends the value 0/0/0/0/0/0 to *dslForum1InterfaceSpecifier*.

To obtain the value for *dslForum1InterfaceSpecifier*, the router translates an internally generated interface specifier into the format for the **dsl-forum-1** keyword, using the following conventions:

- The **dsl-forum-1** format for ATM interfaces is atm *slot/adapter/port*:*vpi*.*vci*
- The **dsl-forum-1** format for Ethernet interfaces is eth *slot/adapter/port*:*svlanId*.*vlanId*
- For the E120 or the E320 routers, the router uses the actual *adapter* value (0 or 1) in the **dsl-forum-1** format. For ERX14xx models, ERX7xx models, and the ERX310 router, which do not support an *adapter* value, the router sets the *adapter* value to 0 (zero).
- For Ethernet interfaces that use VLANs but do not use S-VLANs, the router sets the *svlanId* value to 4096 and uses the actual *vlanId* value in the **dsl-forum-1** format.
- For Ethernet interfaces that use neither S-VLANs nor VLANs, the router sets both the *svlanId* value and the *vlanId* value to 4096 in the **dsl-forum-1** format.
- The router ignores subinterface values for ATM and Ethernet interfaces in the translated **dsl-forum-1** format.

NOTE: The format of the interface specifier that the router generates internally is different from the interface specifier format that you use to configure interfaces on the router. For information about the interface types and specifiers to use when configuring interfaces on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Format Examples for dsl-forum-1 Keyword

Table 35 on page 395 provides several examples of how the router uses the conventions described in "Format for dsl-forum-1 Keyword" on page 394 to translate internally generated interface specifiers into the format of the *dslForum1InterfaceSpecifier* value. The examples in the table use adapter 1 for interfaces on an E120 or E320 router, and adapter 0 (no adapter value) for interfaces on ERX14xx models, ERX7xx models, and the ERX310 router.

Table 35: Interface Specifier Format Examples for dsl-forum-1 Keyword

Interface Example	Internal Router Format	How Router Translates	Format of dslForum1InterfaceSpecifier
ATM 1483 subinterface on slot 2, port 0, subinterface 1 with VPI 100 and VCI 101	atm 2/0.1:100.101	• Sets *adapter* to 0 • Ignores subinterface 1 • Uses other values as supplied	atm 2/0/0:100.101

Table 35: Interface Specifier Format Examples for dsl-forum-1 Keyword *(continued)*

Interface Example	Internal Router Format	How Router Translates	Format of dslForum1InterfaceSpecifier
ATM 1483 subinterface on slot 3, adapter 1, port 7, subinterface 6 with VPI 200 and VCI 201	atm 3/1/7.6:200.201	• Ignores subinterface 6 • Uses other values as supplied	atm 3/1/7:200.201
Gigabit Ethernet interface on slot 2, port 0 with no VLAN or S-VLAN subinterfaces	gigabitEthernet 2/0	• Sets *adapter* to 0 • Sets both *svlanId* and *vlanId* to 4096 • Uses other values as supplied	eth 2/0/0:4096.4096
Gigabit Ethernet interface on slot 4, adapter 1, port 1 with no VLAN or S-VLAN subinterfaces	gigabitEthernet 4/1/1	• Sets both *svlanId* and *vlanId* to 4096 • Uses other values as supplied	eth 4/1/1:4096.4096
Gigabit Ethernet interface on slot 2, port 0, subinterface 1 with VLAN ID 5	gigabitEthernet 2/0.1:5	• Sets *adapter* to 0 • Ignores subinterface 1 • Sets *svlanId* to 4096 • Uses other values as supplied	eth 2/0/0:4096.5
Gigabit Ethernet interface on slot 4, adapter 1, port 1, subinterface 3 with VLAN ID 10	gigabitEthernet 4/1/1.3:10	• Ignores subinterface 3 • Sets *svlanId* to 4096 • Uses other values as supplied	eth 4/1/1:4096.10
Gigabit Ethernet interface on slot 2, port 0, subinterface 1 with S-VLAN ID 5 and VLAN ID 6	gigabitEthernet 2/0.1:5-6	• Sets *adapter* to 0 • Ignores subinterface 1 • Replaces - (hyphen) between *svlanId* and *vlanID* with . (period) • Uses other values as supplied	eth 2/0/0:5.6
Gigabit Ethernet interface on slot 4, adapter 1, port 1, subinterface 3 with S-VLAN ID 10 and VLAN ID 20	gigabitEthernet 4/1/1.3:10-20	• Ignores subinterface 3 • Replaces - (hyphen) between *svlanId* and *vlanID* with . (period) • Uses other values as supplied	eth 4/1/1:10.20

Use by RADIUS or L2TP

Enabling the router to capture and format the PPPoE remote circuit ID sent from the DSLAM has no effect by itself. To use the PPPoE remote circuit ID value, you must send it to a RADIUS server, to an L2TP network server (LNS), or to both by doing one or more of the following:

- Issue the **radius override calling-station-id remote-circuit-id** command to substitute the remote circuit ID value for the standard Calling-Station-Id [31] RADIUS attribute.
- Issue the **radius override nas-port-id remote-circuit-id** command to substitute the remote circuit ID value for the standard NAS-Port-Id [87] RADIUS attribute.
- Issue the **aaa tunnel calling-number-format** command to generate L2TP Calling Number attribute value pair (AVP) 22 in a descriptive format that includes either or both of the agent-circuit-id (suboption 1) and agent-remote-id (suboption 2) suboptions of the PPPoE intermediate agent tags.

For more information about configuring RADIUS and L2TP on E Series routers, see the *JunosE Broadband Access Configuration Guide*.

System Event Log

You can use the radiusSendAttributes system event log category to troubleshoot applications that use PPPoE remote circuit ID capture. The radiusSendAttributes event category logs RADIUS attributes added to outbound RADIUS requests.

You can also use the **log severity debug pppoeControlPacket** command to configure a packet trace log for a PPPoE interface that includes the PPPoE remote circuit ID value captured on that interface. For information about how to use the **log severity debug pppoeControlPacket** command, see "Troubleshooting PPPoE Interfaces" on page 441.

For information about how to log system events, see *Overview of System Logging*.

Related Documentation

- Configuring PPPoE Remote Circuit ID Capture on page 421
- Monitoring the RADIUS Override Settings on page 441
- Monitoring Tunnel Parameters Configured for L2TP Tunnel Definitions on page 426
- Troubleshooting PPPoE Interfaces on page 441
- *log severity*
-

PPPoE Platform Considerations

You can configure PPPoE interfaces on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router

- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support PPPoE interfaces on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support PPPoE.

For information about the modules that support PPPoE interfaces on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support PPPoE.

Interface Specifiers

The configuration task examples in this chapter use the *slot/port*[.*subinterface*] format to specify the physical interface on which you want to configure PPPoE. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

host1(config)#**interface atm 0/1.10**

For E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

host1(config)#**interface atm 5/0/0.20**

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

PPPoE References

For more information about PPPoE, consult the following resources:

- DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006)
- Extensions to a Method for Transmitting PPP over Ethernet (PPPoE)—draft-carrel-info-pppoe-ext-00.txt (November 2000 expiration)
- IEEE 802.1q (Virtual LANs)
- RFC 2516—Method for Transmitting PPP over Ethernet (PPPoE) (February 1998)

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Refer to the IETF Web site at http://www.ietf.org for the latest drafts.

Related Documentation

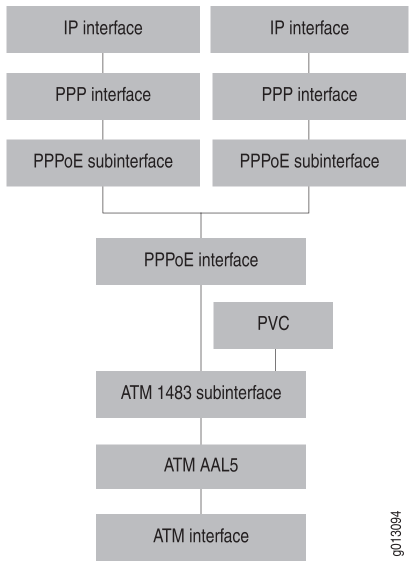
- Understanding PPPoE on page 388
- PPPoE Platform Considerations on page 397
- Configuring PPPoE over ATM on page 402
- Creating and Populating PPPoE Service Name Tables on page 416

Access Nodes in Ethernet Aggregation Networks Overview

The access node is the first aggregation point in the digital subscriber line (DSL) access network and apart from terminating the DSL physical layer signals, it also terminates the user ATM layer. The access node contains an Ethernet uplink to provide connectivity to the aggregation network. When ATM is supported on the DSL line, the access node provisions an interworking function between the ATM layer on the user side and the Ethernet layer on the network side, encompassing protocol translation, access loop identification, QoS, security, and OAM attributes. This functionality might require the access node to snoop, modify, or terminate protocols in layers above the ATM Adaptation Layer 5 (AAL5) encapsulation. The access node operates as an Ethernet switch, while it also offers enhanced functionality for protocol interworking, multicast support, and customization for support of access networks (such as ARP and IGMP processing, and user identification and isolation).

Figure 38 on page 400 illustrates the interface stack for this configuration.

Figure 38: Example of PPPoE over ATM Stacking



The subscriber's access node indicates to the B-RAS application running on the router whenever a PPPoE session carries interworked PPPoA or PPPoE over ATM traffic. This indication enables the B-RAS application to modify its behavior for interworked PPPoE sessions. To denote that a PPPoE session contains interworked traffic, the PPPoE client or the host includes the IWF-Session DSL Forum VSA (26-254) in the unicast PPPoE Active Discovery Request (PADR) packet that it transmits to the PPPoE access concentrator to which it wants to connect.

Related Documentation

- ATM-to-Ethernet Interworking Overview on page 400
- Configuring PPPoE over ATM on page 402
- Monitoring the PPPoE Interface on page 427
- Monitoring the PPPoE Subinterface on page 434

ATM-to-Ethernet Interworking Overview

In the ATM-to-Ethernet IWF topology, each ATM virtual path identifier (VPI) or virtual circuit identifier (VCI) is mapped to a corresponding stacked Ethernet Virtual Local Area Network (VLAN). When this feature is used, the ATM packets are translated into Ethernet packets through mapping between an ATM link and an Ethernet link. The VPI of ATM packets is mapped to the external stacked VLAN (S-VLAN) tag, and the VCI of ATM packets is mapped to the internal customer VLAN (C-VLAN) tag, thereby enabling the transmission of ATM packets over Ethernet links. VPI and VCI are mapped to double tags. The inner and outer SVLAN tags identify ATM digital subscriber line access multiplexer (DSLAM) device information and user information, respectively. After the PPPoE authentication packet reaches B-RAS, B-RAS can authenticate based on user account, device information (the outer tag), and user information (the inner tag). In this way, account hacking can be avoided in terms of security.

The DSL Forum defined the IWF to devise the process for conversion of PPP over ATM (PPPoA) and PPPoE over ATM sessions to PPPoE sessions at the DSLAM to the B-RAS application running on routers. This functionality was defined to enable DSL access infrastructure in networks worldwide to migrate from ATM to Ethernet-based connections.

IWF is a set of mechanisms required to interlink two networks of different technologies. IWF is used to describe the PPPoA conversion to PPPoE sessions at the DSLAM. These mechanisms include conversion of PDU framing, addressing policies, priority mapping, security mechanisms, and OAM flows. In ATM-to-Ethernet interworking circuits, the PPPoA session that arrives at the DSLAM over ATM from a customer premises equipment (CPE) or access loop is converted to a PPPoE session at the DSLAM. This PPPoE session is then continued to be transmitted to the PPPoE access concentrator to B-RAS as a PPPoE session. Every PPPoA session is associated with a corresponding PPPoE session.

A PPPoE session from the DSLAM to the B-RAS that is actually a PPPoA session from the end user to the DSLAM is referred to as an IWF PPPoE session. The B-RAS application is configured to limit PPPoE sessions that originate from the same MAC address to protect itself from a denial of service (DoS) attack. This restriction on maximum number of PPPoE client sessions poses a problem for IWF PPPoE sessions because all PPPoE sessions contain the same MAC address of the DSLAM.

To avoid this problem, the PPPoE client inserts the IWF PPPoE tag in the PADR packet to the PPPoE access concentrator to which it wants to connect. The B-RAS application uses the IWF PPPoE tag to distinguish between an IWF PPPoE session and a regular, non-IWF PPPoE session during the PPPoE discovery stage. The IWF PPPoE tag enables the B-RAS application running on E Series routers to distinguish the IWF PPPoE session from the regular PPPoE sessions to overcome the limit on the B-RAS the maximum number of PPPoE sessions per MAC address as a protection from DoS attacks sourced from the same MAC address. For more information about ATM-to-Ethernet interworking functions, see the *DSL Forum Technical Report 101: Migration to Ethernet-Based DSL Aggregation*.

These ATM-to-Ethernet interworking circuits can be mapped to individual logical interfaces configured on an ATM, Gigabit Ethernet, or 10-Gigabit Ethernet physical interface. The ATM-to-Ethernet interworking cross-connect essentially provides Layer 2 switching, and statistics are reported at the logical interface level.

During the conversion from ATM to Ethernet, the least significant 12 bits of the ATM cell VCI are copied to the Ethernet frame inner VLAN tag. Cells received on an ATM logical interface configured with the ATM-to-Ethernet interworking encapsulation type and falling within the configured VCI range are reassembled into packets. These packets are forwarded to a designated Ethernet logical interface that is configured with the ATM-to-Ethernet interworking encapsulation type.

During the conversion from Ethernet to ATM, the Ethernet frame inner VLAN tags that fall within the configured range are copied to the least significant 12 bits of the ATM cell VCI. The ATM logical interface uses its configured VPI when segmenting the Ethernet packets into cells. ATM-to-Ethernet interworking is supported on E Series routers with aggregated Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces.

Related Documentation

- Configuring PPPoE over ATM on page 402
- Overview of IWF PPPoE Sessions with Duplicate MAC Addresses on page 404
- Single DSLAM Connected to a PPPoE Access Concentrator on page 406
- Multiple DSLAMs Connected to a PPPoE Access Concentrator on page 407
- PPPoE with Ethernet Modules Configuration Overview on page 408
- Access Nodes in Ethernet Aggregation Networks Overview on page 399

Configuring PPPoE over ATM

Before you attempt to configure a PPPoE interface, configure the physical interface over which PPPoE traffic will flow. The procedure described in this topic assumes that a physical interface has been configured.

To configure PPPoE over ATM:

1. Configure a physical interface.

   host1(config)#**interface atm 0/1**

2. Configure the ATM 1483 subinterface.

   host1(config-if)#**interface atm 0/1.20**

3. Configure a PVC by specifying the vcd (virtual circuit descriptor), the vpi (virtual path identifier), the vci (virtual channel identifier), and the encapsulation type.

   host1(config-if)#**atm pvc 10 22 100 aal5snap**

4. Select PPPoE as the encapsulation method.

   host1(config-subif)#**encapsulation pppoe**

5. Do one of the following to configure the maximum number of PPPoE sessions (subinterfaces) supported on the interface:

   - Configure the maximum number of PPPoE sessions on a per-interface basis.

     host1(config-if)#**pppoe sessions 128**

   - Configure the maximum number of PPPoE sessions on a per-subscriber basis by overriding the current PPPoE maximum session value with the PPPoE maximum session value returned by the RADIUS server in the Max-Clients-Per-Interface vendor-specific attribute (VSA) [26-143] in Access-Accept messages.

     host1(config-if)#**pppoe max-session-vsa override**

6. Create a PPPoE subinterface.

   host1(config-subif)#**interface atm 0/1.20.1**

7. Select PPP as the encapsulation method.

   host1(config-subif)#**encapsulation ppp**

8. (Optional) Configure maximum transfer unit (MTU) parameters.

host1(config-if)#pppoe mtu 1380

9. (Optional) Configure an access concentrator (AC) name on the PPPoE interface.

   host1(config-subif)#pppoe acname CYM9876

10. (Optional) Set up the router to prevent a client from establishing more than one session using the same MAC address.

    When the duplicate protection feature is enabled, multiple IWF PPPoE sessions that contain the same MAC address are still processed and can access network services until the maximum number of PPPoE sessions configured per major interface (configured using the **pppoe sessions** command) is reached. For more information about how IWF PPPoE sessions with duplicate addresses are handled, see "Overview of IWF PPPoE Sessions with Duplicate MAC Addresses" on page 404.

    host1(config-subif)#pppoe duplicate-protection

11. Assign an IP address and subnet mask to the PVC.

    host1(config-subif)#ip address 192.32.10.20 255.255.255.0

12. (Optional) Configure additional PPPoE subinterfaces by completing Steps 6 through 11 using unique numbering.

    host1(config-subif)#interface atm 0/1.20.2

Related Documentation

- Configuring ATM on page 3
- Configuring Bridged Ethernet on page 451
- Configuring Upper-Layer Dynamic Interfaces on page 519
- Configuring Upper-Layer Protocols over Static Ethernet Interfaces on page 149

- *atm pvc*
- *encapsulation ppp*
- *encapsulation pppoe*
- *interface atm*
- *ip address*
- *pppoe acName*
- *pppoe duplicate-protection*
- *pppoe max-session-vsa*
- *pppoe mtu*
- *pppoe sessions*

Guidelines for Overriding the PPPoE Maximum Session Value

The following rules apply if you configure the router to override the current PPPoE maximum session value for the interface with the PPPoE maximum session value returned by RADIUS in the Max-Clients-Per-Interface VSA:

- If the current PPPoE maximum session value is *less than* the PPPoE maximum session value returned by RADIUS, the PPPoE application overrides the current maximum session value with the value returned by RADIUS and proceeds with creation of the PPPoE interface.
- If the current number of active PPPoE sessions, including the current session, is *greater than* the PPPoE maximum session value returned by RADIUS, PPPoE ignores (drops) the new session.
- If the current number of active PPPoE sessions is *less than or equal to* the PPPoE maximum session value returned by RADIUS, PPPoE proceeds with creation of the PPPoE interface.

The following table shows examples of the PPPoE maximum session value when a new subscriber logs in, and indicates the status of the current session.

PPPoE Maximum Session Value from RADIUS	Current PPPoE Maximum Session Value	Existing Number of PPPoE Sessions	New PPPoE Maximum Session Value	New Number of PPPoE Sessions	Status of Session
10	5	4	10	5	PPP session up
3	5	4	3	5	PPP/PPPoE session down
5	5	4	5	5	PPP session up

Related Documentation

- Monitoring the PPPoE Profile on page 436
- Monitoring the RADIUS Override Settings on page 441
- *pppoe max-session-vsa*

Overview of IWF PPPoE Sessions with Duplicate MAC Addresses

JunosE Software supports detection of PPPoE sessions with duplicate MAC addresses that contain interworking function (IWF) tags. The IWF feature performs a set of operations on a subscriber's session to enable the transport of PPPoE over ATM traffic on a PPPoE interface.

PPPoE supports duplicate detection based on MAC addresses to prevent spoofed MAC addresses and to avoid unauthorized users from attempting to use the MAC address of another valid user. When duplicate protection is configured for the underlying interface, a dynamic PPPoE logical interface cannot be activated when an existing active logical interface is present for the same PPPoE client. This mechanism prevents an unauthorized user to deny or disrupt service to a legitimate user.

Although duplicate protection of PPPoE sessions with the same MAC address enables prevention of unauthorized access to resources, there might be scenarios in interworked PPPoE sessions in which multiple sessions that originate from the same MAC address are required for access to network services and applications. In this release, you can enable multiple PPPoE sessions with the same MAC address that contain the IWF tag to be established. This feature is useful for IWF PPPoE sessions because of a number of such sessions contain the same MAC address of the DSLAM at which multiplexing and conversion functions are performed.

For PPPoE sessions that contain the IWF-Session DSL Forum VSA (26-254) in the PADR packets sent from the client to the PPPoE access concentrator, multiple subscriber sessions with the same MAC address can originate. This behavior occurs because the interworking functionality (IWF) causes a PPPoE over ATM or PPP over ATM (PPPoA) session to be converted by the digital subscriber line access multiplexer (DSLAM) into a PPPoE session. As a result of this conversion, the MAC addresses of all IWF PPPoE sessions contain the MAC address of the DSLAM device.

For PPPoE sessions with the IWF-Session VSA, duplication of MAC addresses is permitted by default. Regardless of whether the duplicate protection feature is enabled, multiple IWF PPPoE sessions with the same MAC address (the address of the DSLAM device) are not terminated until the limit on the maximum number of PPPoE sessions configured on the major interface is reached.

Related Documentation

- Guidelines for Configuring Duplicate Protection for IWF PPPoE Sessions on page 405
- Single DSLAM Connected to a PPPoE Access Concentrator on page 406
- Multiple DSLAMs Connected to a PPPoE Access Concentrator on page 407

Guidelines for Configuring Duplicate Protection for IWF PPPoE Sessions

Keep the following points in mind when you configure duplicate protection for IWF PPPoE sessions:

- In most environments, a 1:1 relationship between the DSLAM and PPPoE access concentrator is present. In such situations, all IWF sessions demultiplexed at any PPPoE access concentrator are required to contain the same source MAC address. In deployments where IWF sessions originate from multiple MAC addresses (because of multiple DSLAMs used to demultiplex subscriber sessions) and no VLAN grouping of VLAN IDs is configured, IWF sessions are not limited per source MAC address.
- If a user spoofs the IWF-Session VSA in a PPPoE PADR that originates from the PPPoE client or access loop for a non-IWF session, this user might be able to bypass the duplicate protection setting configured on the router. The PPPoE access concentrator cannot detect such spoofing when the interworking functionality is activated.
- Table 36 on page 406 describes the different scenarios in which duplicate MAC addresses are supported for IWF PPPoE sessions and non-IWF PPPoE sessions, when duplicate protection configuration is enabled or disabled on a router.

Table 36: PPPoE Duplicate Protection Scenarios for IWF and non-IWF PPPoE Sessions

Type of PPPoE Session	Duplicate Protection Enabled	Duplicate Protection Disabled
IWF PPPoE session (IWF-Session DSL VSA contained in the PADR packet)	Sessions with duplicate MAC addresses are processed until the maximum number of PPPoE sessions configured per major interface is reached.	Sessions with duplicate MAC addresses are processed.
Non-IWF PPPoE session (IWF-Session DSL VSA not contained in the PADR packet)	Sessions with duplicate MAC addresses are terminated and cannot access network resources	Sessions with duplicate MAC addresses are processed.

Related Documentation

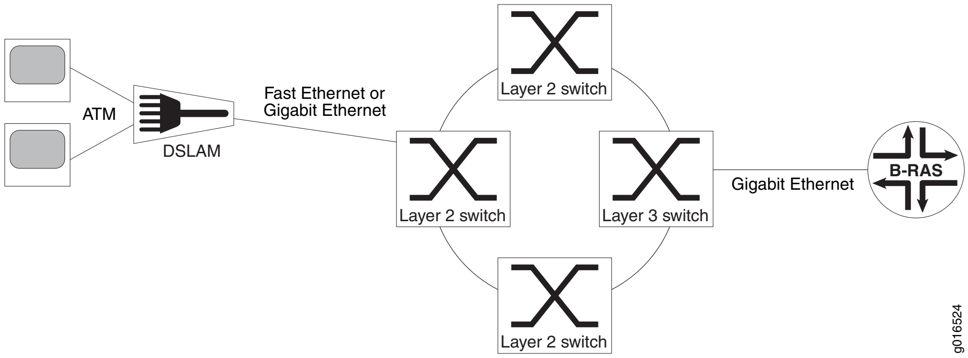
- Overview of IWF PPPoE Sessions with Duplicate MAC Addresses on page 404
- Single DSLAM Connected to a PPPoE Access Concentrator on page 406
- Multiple DSLAMs Connected to a PPPoE Access Concentrator on page 407

Single DSLAM Connected to a PPPoE Access Concentrator

This topic describes a sample configuration case to illustrate how the interworking functions are performed when a 1:1 association exists between DSLAM and the PPPoE access concentrator (B-RAS application).

Consider a topology in which a DSLAM multiplexes multiple PPPoE over ATM sessions over the Ethernet aggregate network using interworking functionality between PPPoE over ATM and PPPoE. Figure 39 on page 407 shows a sample configuration scenario in which one DSLAM is connected to one PPPoE access concentrator. The subscriber access nodes are connected using ATM links to the DSLAMs. In this circuit, a 1:1 relationship exists between the DSLAM and the PPPoE access concentrator (PPPoE major interface). Because of the presence of a 1:1 association between the DSLAM and the access concentrator, segregation of VLAN IDs is not required. A one-to-one mapping is configured between the user port and VLAN. The access node is assigned a unique VLAN identification to a port using either a unique S-VLAN ID or a unique (S-VLAN ID, C-VLAN ID) pair. The uniqueness of the mapping is maintained in the access node and across the aggregation network. The B-RAS PPPoE access concentrator demultiplexes the PPPoE sessions.

Figure 39: Single DSLAM Connected to a PPPoE Access Concentrator



In such a case, the PPPoE access concentrator uses the maximum number of subinterfaces configured on a PPPoE interface using the **pppoe sessions** command to limit the maximum number of IWF PPPoE sessions. The B-RAS tracks the IWF flag in a PPPoE PADR packet it receives from the access concentrator, which the PPPoE client forwarded to the access concentrator during the discovery stage. The presence of the IWF tag causes the PPPoE access concentrator to accept duplicate PPPoE sessions with the same MAC address of the DSLAM are permitted by default. The suboption for the IWF-Session DSL Forum VSA (26-254) contains the field code of 0xFE in PADR packet and its length field is set to 0x00.

Related Documentation

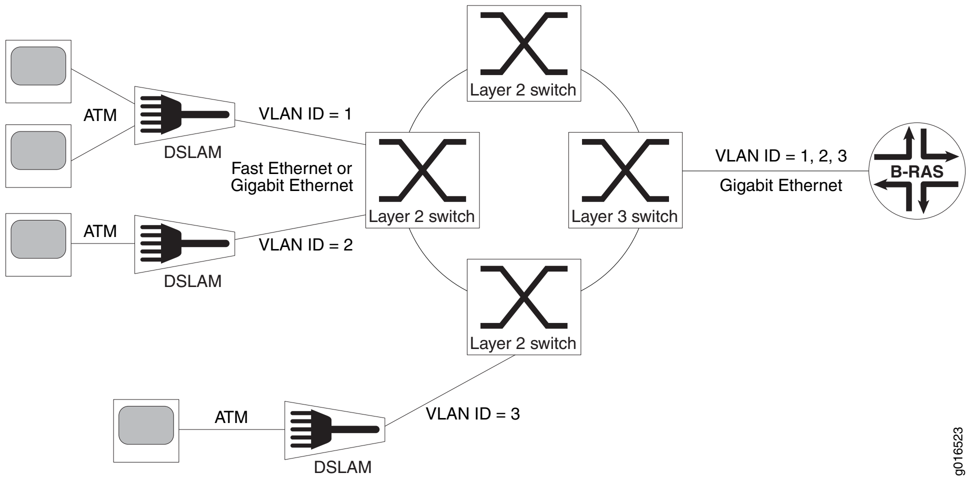
- Multiple DSLAMs Connected to a PPPoE Access Concentrator on page 407
- Guidelines for Configuring Duplicate Protection for IWF PPPoE Sessions on page 405
- Overview of IWF PPPoE Sessions with Duplicate MAC Addresses on page 404

Multiple DSLAMs Connected to a PPPoE Access Concentrator

This topic describes a sample configuration case to illustrate how the interworking functions are performed when a N:1 association exists between DSLAM and the PPPoE access concentrator (B-RAS application).

Consider a scenario in which the access node assigns the same S-VLAN ID to a group of ports. Figure 40 on page 408 shows a sample configuration scenario in which multiple DSLAMs are connected to a single PPPoE access concentrator. This configuration pattern is denoted as N:1 VLAN to indicate many-to-one mapping between ports and VLAN. Sample criteria for grouping multiple ports with the same S-VLAN ID comprise the same originating virtual path, same service, or same destination service provider. In this topology, multiple DSLAMs, which aggregate sessions from multiple PPPoE clients, are connected to one access concentrator.

Figure 40: Multiple DSLAMs Connected to a PPPoE Access Concentrator



Although in such a scenario, in which a N:1 DSLAM and PPPoE access concentrator relationship is exhibited, normally a VLAN grouping is performed to segregate the traffic from each DSLAM. Such a grouping of VLAN IDs is a regular network administration practice. If such a grouping of VLAN IDs is made, a 1:1 relationship between DSLAM and the PPPoE access concentrator is created that enables each access concentrator to serve a VLAN group.

Related Documentation

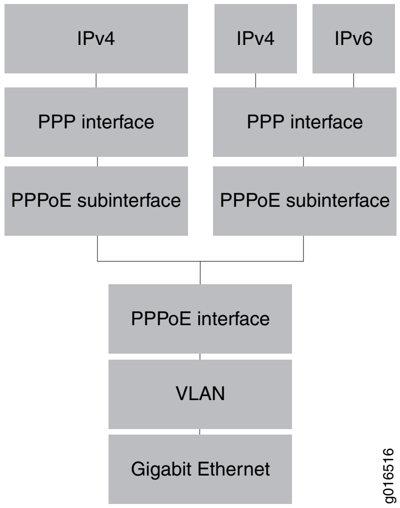
- Single DSLAM Connected to a PPPoE Access Concentrator on page 406
- Guidelines for Configuring Duplicate Protection for IWF PPPoE Sessions on page 405
- Overview of IWF PPPoE Sessions with Duplicate MAC Addresses on page 404

PPPoE with Ethernet Modules Configuration Overview

You can configure PPPoE on Fast Ethernet (FE), Gigabit Ethernet (GE), and 10-Gigabit Ethernet (10GE) modules. You can configure Ethernet interfaces with IP only, with PPPoE only, with both IP and PPPoE, and with or without VLANs.

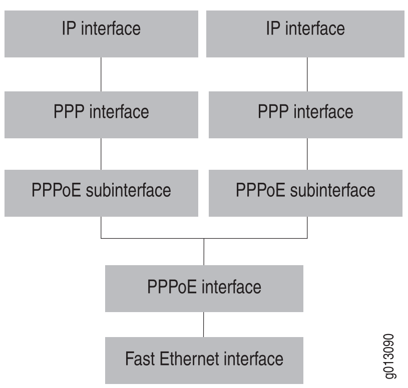
This topic provides information about configuring PPPoE without VLANs as well as configuring PPPoE with VLANs. You can configure IPv4 and IPv6 interface columns over static PPPoE with VLAN, as shown in Figure 41 on page 409.

Figure 41: Example of Configuring IPv4 and IPv6 over PPPoE



You can also configure an IP interface over PPPoE without VLAN. Figure 42 on page 409 illustrates the interface stack for this type of configuration.

Figure 42: Example of PPPoE Stacking



If you want to configure PPPoE with VLANs, see "Configuring VLAN and S-VLAN Subinterfaces" on page 165, which shows common VLAN configurations such as:

- PPPoE over VLAN
- IP over VLAN and PPPoE over VLAN

NOTE: "Configuring VLAN and S-VLAN Subinterfaces" on page 165 provides other non-VLAN configuration examples, such as configurations using MPLS.

For more information about specific Ethernet modules and the protocols and applications they support, see:

- *ERX Module Guide*, *Appendix A*, *Module Protocol Support* (for ERX7xx models, ERX14xx models, and ERX310 router)
- *E120 and E320 Module Guide*, *Appendix A*, *IOA Protocol Support* (for E120 and E320 routers)

PPPoE Interface and Subinterface Limits

PPPoE subinterfaces can be distributed in any way across I/O module ports. For example, you can configure the maximum supported number of PPPoE subinterfaces on one port of an FE-2 I/O module and no PPPoE subinterfaces on the other port.

For information about current system maximums supported for PPPoE interfaces and subinterfaces, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

Related Documentation

- Access Nodes in Ethernet Aggregation Networks Overview on page 399
- Configuring IPv4 and IPV6 over static PPPoE with VLANs on page 410
- Configuring IPv4 over PPPoE Without VLANs on page 411
- Setting a Baseline for PPPoE Interface Statistics on page 425
- Monitoring the PPPoE Interface on page 427
- Monitoring the PPPoE Subinterface on page 434

Configuring IPv4 and IPV6 over static PPPoE with VLANs

To configure IPv4 and IPv6 interface columns over static PPPoE with VLANs:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface.

   host1(config)#**interface gigabitEthernet 2/0/1**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface gigabitEthernet 2/0/1.1**

4. Assign a VLAN ID for the subinterface.

   host1(config-if)#**vlan id 1**

5. Create a PPPoE subinterface.

   host1(config-if)#**pppoe**

6. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe subinterface gigabitEthernet 2/0/1.1.1**
   host1(config-if)#**encapsulation ppp**

7. Specify the order of preference for the primary authentication protocol.

host1(config-if)#**ppp authentication pap chap eap**

8. Enable IPv6 processing on an interface without assigning an explicit IPv6 address to that interface.

   host1(config-if)#**ipv6 unnumbered loopback 0**

9. Enable the IPv6 Neighbor Discovery process on an interface.

   host1(config-if)#**ipv6 nd**

10. Specify which IPv6 prefixes the system includes in IPv6 router advertisements.

    host1(config-if)#**ipv6 nd prefix-advertisement 2002:1::/64 60000 45000 onlink autoconfig**

11. (Optional) Configure additional VLAN subinterfaces by completing Steps 3 through 10.

Related Documentation

- Access Nodes in Ethernet Aggregation Networks Overview on page 399
- PPPoE with Ethernet Modules Configuration Overview on page 408
- Monitoring the PPPoE Interface on page 427
- Monitoring the PPPoE Subinterface on page 434

- *encapsulation ppp*
- *interface fastEthernet*
- *ip address*
- *pppoe*
- *pppoe acName*
- *pppoe duplicate-protection*

Configuring IPv4 over PPPoE Without VLANs

To configure PPPoE over an Ethernet interface without VLANs:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface.

   host1(config)#**interface fastEthernet 4/1**

2. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe**

3. Create a PPPoE subinterface.

   host1(config-if)#**pppoe subinterface fastEthernet 4/1.1**

4. Specify PPP as the encapsulation method on the interface.

   host1(config-subif)#**encapsulation ppp**

5. (Optional) Configure an access concentrator (AC) name on the PPPoE interface.

   host1(config-subif)#**pppoe acname CYM9876**

6. (Optional) Set up the router to prevent a client from establishing more than one session using the same MAC address.

   When the duplicate protection feature is enabled, multiple IWF PPPoE sessions (sent from PPPoE clients to the PPPoE access concentrator) that contain the same MAC address are still processed and can access network services until the maximum number of PPPoE sessions configured per major interface (configured using the **pppoe sessions** command) is reached.

   host1(config-subif)#**pppoe duplicate-protection**

7. Assign an IP address and mask.

   host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

8. (Optional) Configure additional PPPoE subinterfaces by completing Steps 3 through 7 using unique numbering.

Related Documentation

- Access Nodes in Ethernet Aggregation Networks Overview on page 399
- PPPoE with Ethernet Modules Configuration Overview on page 408
- Monitoring the PPPoE Interface on page 427
- Monitoring the PPPoE Subinterface on page 434

- *encapsulation ppp*
- *interface fastEthernet*
- *ip address*
- *pppoe*
- *pppoe acName*
- *pppoe duplicate-protection*
- *pppoe subinterface*

PADM Messages Overview

You can configure PPPoE to issue and display a PPPoE Active Discovery Message (PADM). The PADM message is a control message that servers send to clients. The clients may act on the control message, but are not required to do so. There are two types of PADM messages:

- Message of the minute (MOTM)—Informs clients of interesting system information
- URL—Typically spawns an Internet browser with the specified URL as the initial page

You can configure the router to send PADM messages as follows:

- Send MOTM messages to all clients connected to the router.
- Send MOTM and URL messages to all clients connected to a subinterface.

- Configure profiles to send MOTM and URL messages to new clients created when the profile is dynamically attached to an IP interface.

NOTE: You can use the **pppoe motm** command at three different points in the configuration process: Privileged Exec, Interface Configuration, and Profile Configuration modes. You can use the **pppoe url** command at two different points in the configuration process: Interface Configuration and Profile Configuration modes. Note the differences described in guidelines below.

Related Documentation

- Configuring MOTM Messages from Privileged Exec Mode on page 413
- Configuring MOTM Messages from Interface Configuration Mode on page 413
- Configuring MOTM Messages from Profile Configuration Mode on page 414
- Configuring URL Messages from Interface Configuration Mode on page 414
- Configuring URL Messages from Profile Configuration Mode on page 415

- *pppoe motm*
- *pppoe url*

Configuring MOTM Messages from Privileged Exec Mode

You can configure the PPPoE application to send a PADM message to all PPPoE clients connected to the router. The MOTM string is passed with no changes. The message string is not saved in NVS.

To configure the PPPoE application to send a MOTM message in a PADM packet:

- From the Privileged Exec mode, specify the message:

  host1#**pppoe motm Router going down at 10:00 p.m.**

  Use the **no** version to disable the message.

Related Documentation

- PADM Messages Overview on page 412
- Configuring MOTM Messages from Interface Configuration Mode on page 413
- Configuring MOTM Messages from Profile Configuration Mode on page 414

- *pppoe motm*

Configuring MOTM Messages from Interface Configuration Mode

You can configure the PPPoE application to send a specific PADM message to the client as it is configured (if connected). The message is also sent whenever the subinterface transitions from down to up. The MOTM string is passed with no changes. The message string is saved in NVS.

To configure the PPPoE application to send a MOTM message in a PADM packet:

- From the Interface Configuration mode, specify the message:

  host1(config-if)#interface fastEthernet 1/0.1.1
  host1(config-if)#pppoe motm Router going down at 10:00 p.m.

  Use the **no** version to disable the message.

Related Documentation

- PADM Messages Overview on page 412
- Configuring MOTM Messages from Privileged Exec Mode on page 413
- Configuring MOTM Messages from Profile Configuration Mode on page 414
- *pppoe motm*

Configuring MOTM Messages from Profile Configuration Mode

You can configure the PPPoE application to send a specific PADM message to the new client that is created when the profile is dynamically attached to an IP interface. The MOTM string is passed with no changes. The message string is saved in NVS.

To configure the PPPoE application to send a MOTM message in a PADM packet:

- From the Profile Configuration mode, specify the message:

  host1(config-profile)#pppoe motm Router going down now

  Use the **no** version to disable the message.

Related Documentation

- PADM Messages Overview on page 412
- Configuring MOTM Messages from Privileged Exec Mode on page 413
- Configuring MOTM Messages from Interface Configuration Mode on page 413
- *pppoe motm*

Configuring URL Messages from Interface Configuration Mode

You can configure the PPPoE application to send the url to the client as it is configured (if connected). The message is also sent whenever the subinterface transitions from down to up. The message string is saved in NVS.

To configure the PPPoE application to send the message string or url:

- From the Interface Configuration mode, specify the url:

  host1(config-if)#interface fastEthernet 1/0.1.1
  host1(config-if)#pppoe url http://www.relevanturl.com

  Use the **no** version to disable the message.

Related Documentation

- PADM Messages Overview on page 412
- Configuring URL Messages from Profile Configuration Mode on page 415
- *pppoe url*

Configuring URL Messages from Profile Configuration Mode

You can configure the PPPoE application to send the url to the new client that is created when the profile is dynamically attached to an IP interface. The message string is saved in NVS. PPPoE substitutes the following characters for information in the specified URL string before transmitting:

- %U user and domain name
- %u user name
- %d domain name
- %D profile name
- %% % character

To configure the PPPoE application to send the message string or url:

- From the Profile Configuration mode, specify the url:

  host1(config-profile)#**pppoe url http://www.relevanturl.com**

  Use the **no** version to disable the message.

Related Documentation

- PADM Messages Overview on page 412
- Configuring URL Messages from Interface Configuration Mode on page 414
- *pppoe url*

PADN Messages Overview

You can configure PPPoE to receive PPPoE Active Discovery Network (PADN) messages. When a client connects to a PPPoE server, such as an E Series router, the client receives configuration information from the server via the PADN message. This PADN information associates the PPPoE sessions with a set of routes. The client can use this set of routes to determine which session to use based on the destination IP address.

The PADN packet data is relevant only when the PPP network layer is " up." To reach an up state, PPP alerts PPPoE after the Network Control Protocol (NCP) completes negotiation.

The routes of interest can be maintained on the router in domain maps.

NOTE: For information about domain mapping, see *JunosE Broadband Access Configuration Guide*.

Related Documentation

- Configuring PADN Messages on page 416
- *aaa domain-map*
- *padn*

Configuring PADN Messages

You can configure the PPPoE application to receive a specific PADN message from the server. When a client connects to a PPPoE server, the client receives configuration information from the server in the PADN message.

To configure the PPPoE application to receive a PADN message:

- From the Global Configuration mode, specify the domain name that maps to the virtual router and the client's domain name:

  host1#**aaa domain-map xyz.com**

  Use the **no** version to delete the map entry.
- From the Domain Map Configuration mode, specify the PADN parameters for the domain name.

  host1(config-domain-map)#**padn 10.2.25.6 255.255.255.0 10**
  host1(config-domain-map)#**padn 20.2.0.0 255.255.0.0 11**

  Use the **no** version to delete PADN parameters for the domain map.

Related Documentation

- PADN Messages Overview on page 415
- *aaa domain-map*
- *padn*

Creating and Populating PPPoE Service Name Tables

To create and populate a PPPoE service name table on the router:

1. From Global Configuration mode, create a PPPoE service name table by assigning it a name.

   host1(config)#**pppoe-service-name-table myServiceTable1**

   This command accesses PPPoE Service Name Table Configuration mode and builds a default PPPoE service name table named myServiceTable1. The table contains two entries: an empty service name entry associated with the default action, **terminate**; and an unknown service name entry associated with the default action, **drop** as shown in Table 37 on page 417. This table directs the router to respond to all PADI requests containing an empty service name tag; and denies requests that contain a service name tag that has not been configured in the service name table.

Table 37: Default PPPoE Service Name Table

Service Name	Action
“ ”	Terminate
unknown-service-name	Drop

2. (Optional) From PPPoE Service Name Table Configuration mode, create entries to populate the PPPoE service name table.

   You can configure up to 16 specific service name entries per table, in addition to the empty and unknown service name tags.

   ```
   host1(config-pppoe-service-name-table)#service myISPService action drop
   host1(config-pppoe-service-name-table)#service myQOSClass1 action terminate
   host1(config-pppoe-service-name-table)#service myQOSClass2 action drop
   host1(config-pppoe-service-name-table)#service myQOSClass3
   host1(config-pppoe-service-name-table)#service empty-service-name action drop
   host1(config-pppoe-service-name-table)#service unknown-service-name action
     terminate
   ```

   These commands build the PPPoE service name table shown in Table 38 on page 417. This table directs the router to send a PADO packet in response to all PADI requests containing the myQOSClass1, myQOSClass3, empty service name tags, and unknown service name tag. The router is directed to **drop** all PADI requests containing the myISPService or myQOSClass2 or the empty service name tags.

Table 38: PPPoE Service Name Table with Entries

Service Name	Action
myISPService	Drop
myQOSClass1	Terminate
myQOSClass2	Drop
myQOSClass3	Terminate
“ ”	Drop
unknown-service-name	Terminate

3. Exit PPPoE Service Name Table Configuration mode.

   ```
   host1(config-pppoe-service-name-table)#exit
   ```

4. (Optional) Use the appropriate **show** command to verify the creation of the PPPoE service name table and entries.

host1(config)#**show pppoe-service-name-table name myServiceTable1**

5. (Optional) Repeat Steps 1 through 4 to configure additional PPPoE service name tables on the router.

Related Documentation

- PPPoE Service Name Tables Overview on page 390
- Enabling Usage of PPPoE Service Name Tables with Static Interfaces over ATM on page 418
- Enabling Usage of PPPoE Service Name Tables with Static Interfaces over Ethernet on page 419
- Enabling Usage of PPPoE Service Name Tables with Dynamic Interfaces on page 419
- Monitoring the PPPoE Service Name Table on page 433
- *pppoe-service-name-table*
- *service*

Enabling Usage of PPPoE Service Name Tables with Static Interfaces over ATM

To enable a PPPoE service name table for use with a static interface, assign the service name table to the PPPoE major interface.

To enable a PPPoE service name table for use with a static interface in PPPoE over ATM configurations:

1. Configure an ATM physical interface.

   host1(config)#**interface atm 3/0**

2. Configure an ATM 1483 subinterface.

   host1(config-if)#**interface atm 3/0.1**

3. Configure an ATM PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type.

   host1(config-subif)#**atm pvc 10 100 22 aal5snap**

4. Select PPPoE as the encapsulation method on the interface. This command creates the PPPoE major interface.

   host1(config-subif)#**encapsulation pppoe**

5. Assign the PPPoE service name table to the PPPoE major interface.

   host1(config-subif)#**pppoe service-name-table myServiceTable1**

Related Documentation

- PPPoE Service Name Tables Overview on page 390
- Creating and Populating PPPoE Service Name Tables on page 416
- Enabling Usage of PPPoE Service Name Tables with Static Interfaces over Ethernet on page 419
- Enabling Usage of PPPoE Service Name Tables with Dynamic Interfaces on page 419

- Monitoring the PPPoE Service Name Table on page 433

- *atm pvc*
- *encapsulation pppoe*
- *interface atm*
- *pppoe service-name-table*
- *service*

Enabling Usage of PPPoE Service Name Tables with Static Interfaces over Ethernet

To enable a PPPoE service name table for use with a static interface in PPPoE over Ethernet configurations:

1. Configure a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet physical interface.

   host1(config)#**interface fastEthernet 4/1**

2. Select PPPoE as the encapsulation method on the interface. This command creates the PPPoE major interface.

   host1(config-if)#**pppoe**

3. Assign the PPPoE service name table to the PPPoE major interface.

   host1(config-if)#**pppoe service-name-table myServiceTable1**

Related Documentation

- PPPoE Service Name Tables Overview on page 390
- Creating and Populating PPPoE Service Name Tables on page 416
- Enabling Usage of PPPoE Service Name Tables with Static Interfaces over ATM on page 418
- Enabling Usage of PPPoE Service Name Tables with Dynamic Interfaces on page 419
- Monitoring the PPPoE Service Name Table on page 433

- *interface fastEthernet*
- *interface gigabitEthernet*
- *interface tenGigabitEthernet*
- *pppoe*
- *pppoe service-name-table*

Enabling Usage of PPPoE Service Name Tables with Dynamic Interfaces

To enable a PPPoE service name table for use with a dynamic interface, add the service name table to a profile that is dynamically assigned to the interface.

For complete details, see "Dynamic Interface Configuration Using a Profile" on page 565.

To enable a PPPoE service name table for use with a dynamic interface:

1. Create a profile by assigning it a name.

   host1(config)#**profile baseProfile**

2. Assign the PPPoE service name table to the profile as a PPPoE characteristic.

   host1(config-profile)#**pppoe service-name-table myServiceTable1**

3. Exit Profile Configuration mode.

   host1(config-profile)#**exit**

4. Configure a physical interface.

   On ERX7xx models, ERX14xx models, and the ERX310 router:

   host1(config-if)#**interface atm 3/0.1**

5. Configure an ATM PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type.

   host1(config-subif)#**atm pvc 10 100 22 aal5snap**

6. Apply the profile to the interface.

   host1(config-subif)#**profile pppoe baseProfile**

7. Enable the PPPoE dynamic encapsulation type.

   host1(config-subif)#**auto-configure pppoe**

Related Documentation

- PPPoE Service Name Tables Overview on page 390
- Creating and Populating PPPoE Service Name Tables on page 416
- Enabling Usage of PPPoE Service Name Tables with Static Interfaces over ATM on page 418
- Enabling Usage of PPPoE Service Name Tables with Static Interfaces over Ethernet on page 419
- Monitoring the PPPoE Service Name Table on page 433

- *atm pvc*
- *auto-configure*
- *interface atm*
- *pppoe service-name-table*
- *profile*

Configuring PADS Packet Content

By default, an E Series router acting as an AC sends both the AC-Name and AC-Cookie tags as part of the PADS (PPPoE Active Discovery Session) packet when it confirms a session with a PPPoE client. These tags are defined in RFC 2516 as follows:

- AC-Name—String that uniquely identifies the particular AC
- AC-Cookie—Tag used by the AC to help protect against denial of service (DoS) attacks

If necessary for compatibility with your network equipment, you can prevent the router from sending the AC-Name and AC-Cookie tags in the PADS packet.

NOTE: The **pppoe pads disable-ac-info** command affects PADS packets sent only on PPPoE interfaces configured on the router after the command is issued. It has no effect on PADS packets sent on previously created PPPoE interfaces.

To configure the packet content of the PADS packet:

- From Global Configuration mode, prevent the router from sharing the AC-Name and AC-Cookies information in the PADs packet:

  host1(config)#**pppoe pads disable-ac-info**

Related Documentation

- Understanding PPPoE on page 388
- *pppoe pads disable-ac-info*

Configuring PPPoE Remote Circuit ID Capture

To capture and use the PPPoE remote circuit ID:

1. Configure a static or dynamic PPPoE interface.

   For instructions on configuring a static PPPoE interface, see "Configuring PPPoE over ATM" on page 402 or "PPPoE with Ethernet Modules Configuration Overview" on page 408.

   For instructions on configuring a dynamic PPPoE interface, see "Configuring Upper-Layer Dynamic Interfaces" on page 519.

2. Configure capture of the PPPoE remote circuit ID on this interface.

   a. Enable the router to capture the PPPoE remote circuit ID transmitted from the DSLAM by using one of the following methods:

      - For a static PPPoE interface, issue the **pppoe remote-circuit-id** command from Interface Configuration mode or Subinterface Configuration mode.

        host1(config-if)#**pppoe remote-circuit-id**

      - For a dynamic PPPoE interface, issue the **pppoe remote-circuit-id** command from Profile Configuration mode as a characteristic of the profile assigned to the dynamic PPPoE interface column.

        host1(config)#**profile pppoeTest**
        host1(config-profile)#**pppoe remote-circuit-id**

By default, the router formats the captured PPPoE remote circuit ID to include only the agent-circuit-id suboption (suboption 1) of the PPPoE intermediate agent tags sent from the DSLAM.

b. (Optional) Use the **show pppoe interface** command (for static PPPoE interfaces) or the **show profile** command (for dynamic PPPoE interfaces) to verify that PPPoE remote circuit capture is enabled.

```
host1#show pppoe interface fastEthernet 4/1.1
host1#show profile name pppoeTest
```

3. (Optional) Configure the format of the captured PPPoE remote circuit ID value.

   a. Configure RADIUS to specify a nondefault format for the PPPoE remote circuit ID value.

   - For example, the following command formats the PPPoE remote circuit ID to include only the agent-remote-id suboption (suboption 2) of the tags supplied by the PPPoE intermediate agent.

     ```
     host1(config)#radius remote-circuit-id-format agent-remote-id
     ```

   - The following command formats the PPPoE remote circuit ID to include the NAS-Identifier [32] RADIUS attribute with both the agent-circuit-id and agent-remote-id suboptions of the tags supplied by the PPPoE intermediate agent.

     ```
     host1(config)#radius remote-circuit-id-format nas-identifier  agent-circuit-id
       agent-remote-id
     ```

   - The following command formats the PPPoE remote circuit ID to append the agent-circuit-ID suboption to an interface specifier that is consistent with the recommended format in the DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006). For details about how the router implements this format, see "Format for dsl-forum-1 Keyword" on page 394.

     ```
     host1(config)#radius remote-circuit-id-format dsl-forum-1
     ```

   b. Configure RADIUS to specify a nondefault delimiter character to separate components in the PPPoE remote circuit ID value. (The default delimiter character is #.)

   ```
   host1(config)#radius remote-circuit-id-delimiter %
   ```

   c. Use the **show radius remote-circuit-id format** command and the **show radius remote-circuit-id-delimiter** command to verify the format and delimiter settings for the PPPoE remote circuit ID value.

   ```
   host1#show radius remote-circuit-id-format
   host1#show radius remote-circuit-id-delimiter
   ```

4. Send the PPPoE remote circuit ID value to a RADIUS server, to an LNS, or to both.

   a. Configure RADIUS to use the PPPoE remote circuit ID captured from the DSLAM in place of either (or both) of the Calling-Station-Id [31] and NAS-Port-Id [87] RADIUS attributes.

   ```
   host1(config)#radius override calling-station-id remote-circuit-id
   ```


host1(config)#**radius override nas-port-id remote-circuit-id**

b. Configure the E Series L2TP access controller (LAC) to generate L2TP Calling Number AVP 22 in fixed format or one of several formats that includes either or both of the agent-circuit-id (suboption 1) and agent-remote-id (suboption 2) suboptions of the tags supplied by the PPPoE intermediate agent.

host1(config)#**aaa tunnel calling-number-format fixed**

or

host1(config)#**aaa tunnel calling-number-format descriptive include-agent-circuit-id include-agent-remote-id**

or

host1(config)#**aaa tunnel calling-number-format include-agent-circuit-id**

c. (Optional) Configure a fallback format for the L2TP Calling Number AVP 22. The fallback format is used only when you have configured the calling number format as anything other than fixed and the PPPoE agent ID is null or unavailable.

host1(config)#**aaa tunnel calling-number-format fallback fixed**

or

host1(config)#**aaa tunnel calling-number-format fallback descriptive**

d. (Optional) Use the **show radius override** command to verify the override settings configured for RADIUS, and the **show aaa tunnel-parameters** command to verify the parameters configured for L2TP tunnel definitions.

host1#**show radius override**
host1#**show aaa tunnel-parameters**

Related Documentation

- Identification of Subscribers Using the PPPoE Remote Circuit Identifier on page 392
- *aaa tunnel calling-number-format*
- *aaa tunnel calling-number-format-fallback*
- *pppoe remote-circuit-id*
- *radius override calling-station-id remote-circuit-id*
- *radius override nas-port-id remote-circuit-id*
- *radius remote-circuit-id-delimiter*
- *radius remote-circuit-id-format*

CHAPTER 14

Monitoring and Troubleshooting Point-to-Point Protocol over Ethernet

This chapter describes how to monitor as well as troubleshoot the Point-to-Point Protocol (PPP) over Ethernet interfaces on E Series routers.

NOTE: The E120 and E320 routers output for monitor and show commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (slot/adapter/port).

This chapter contains the following sections:

- Setting a Baseline for PPPoE Interface Statistics on page 425
- Monitoring Tunnel Parameters Configured for L2TP Tunnel Definitions on page 426
- Monitoring the PPPoE Interface on page 427
- Monitoring the PPPoE Service Name Table on page 433
- Monitoring the PPPoE Subinterface on page 434
- Monitoring the PPPoE Profile on page 436
- Monitoring the RADIUS Override Settings on page 441
- Troubleshooting PPPoE Interfaces on page 441

Setting a Baseline for PPPoE Interface Statistics

You can set a statistics baseline for PPPoE interfaces, subinterfaces, and circuits. To view baseline statistics, use the **delta** keyword with the PPPoE **show** commands.

NOTE: You cannot set a baseline for groups of interfaces, subinterfaces, or circuits. You must set them one at a time.

To set a baseline for PPPoE statistics:

- Issue the **baseline pppoe interface** command for an interface:

```
host1#baseline pppoe interface atm 2/0.1.1
```

There is no **no** version.

Related Documentation

- *baseline pppoe interface*

Monitoring Tunnel Parameters Configured for L2TP Tunnel Definitions

Purpose Display information about the tunnel parameters that are configured for L2TP tunnel definitions.

Action To display information about the tunnel parameters that are configured for L2TP definitions:

```
host1#show aaa tunnel-parameters
Tunnel password is 3&92k5b#q4
Tunnel client-name is host1
Tunnel nas-port-method is none
Tunnel nas-port ignore disabled
Tunnel nas-port-type ignore disabled
Tunnel assignmentId format is assignmentId
Tunnel calling number format is descriptive, includes agent-circuit-id and
agent-remote-id
```

Meaning Table 39 on page 426 lists the **show aaa tunnel-parameters** command output fields.

Table 39: show aaa tunnel-parameters Output Fields

Field Name	Field Description
Tunnel password	Default tunnel password
Tunnel client-name	Hostname that the LAC sends to the LNS when communicating about the tunnel
Tunnel nas-port-method	Default NAS port type
Tunnel nas-port ignore	Whether the router uses the tunnel peer's NAS-Port [5] attribute. Possible values: enabled or disabled.
Tunnel nas-port-type ignore	Whether the router uses the tunnel peer's NAS-Port-Type [61] attribute. Possible values: enabled or disabled
Tunnel assignmentID format	Value of the tunnel assignment ID that is passed to PPP/L2TP
Tunnel calling number format	Format configured for L2TP Calling Number AVP 22 generated by the LAC

Related Documentation

- *show aaa tunnel-parameters*

Monitoring the PPPoE Interface

Purpose Display information about the parameters on a PPPoE interface or subinterface as well as the operational and administrative status of all configured PPPoE interfaces.

NOTE: If you do not specify an interface and subinterface, the router displays the PPPoE interface and status parameters for all configured interfaces. If you specify an interface with no subinterface, the router displays the PPPoE interface and status parameters for that interface. If you specify an interface and subinterface, the router displays detailed parameters available for that subinterface.

Action To display information about the PPPoE interface over an ATM interface:

```
host1#show pppoe interface atm 1/0.1
PPPoE interface ATM 1/0.1 is operStatusUp (dynamic)
    PPPoE interface ATM 1/0.1 has max sessions = 4000
    PPPoE interface ATM 1/0.1 has acName of 111111111111111
    PPPoE interface ATM 1/0.1 will not send ac info in PADS packet

    PPPoE interface ATM 1/0.1 is in duplicate-protection
    PPPoE interface ATM 1/0.1 will capture remote circuit ID

    PPPoE interface ATM 1/0.1 has 1 active connections,
        out of 1 configured subinterfaces

Assigned profile (any)        : baseProfile
PPPoE Statistics
  Counters:
    PADI received       0
    PADI transmitted    1
    PADO received       1
    PADO transmitted    0
    PADR received       0
    PADR transmitted    1
    PADS received       1
    PADS transmitted    0
    PADT received       0
    PADT transmitted    0
    PADM received       1
    PADM transmitted    0
    PADN received       0
    PADN transmitted    0
  PAD packets received        2
  PAD packets transmitted     2
  Invalid PAD Packets:
    Invalid Version           0
    Invalid PAD Code          0
    Invalid PAD Tags          0
    Invalid PAD Tag length    0
    Invalid PAD Type          0
    Invalid PADI Session      0
    Invalid PADR Session      0
    Invalid PAD packet length 3
```



```
    Invalid PAD packets       0
   Total Invalid PAD packets  3

   Insufficient Resources 0
Lockout Configuration (seconds): Min 5, Max 60
 Total clients in active lockouts: 0
 Total clients in lockout grace period: 0
```

To display information about the PPPoE interface over a FastEthernet interface:

```
host1#show pppoe interface fastEthernet 9/4
PPPoE interface fastEthernet 9/4 is operStatusUp
    PPPoE interface fastEthernet 9/4 has max sessions = 8000
    PPPoE interface fastEthernet 9/4 has no acName set
    PPPoE interface fastEthernet 9/4 is in test mode
    PPPoE interface fastEthernet 9/4 has 16 active connections,
        out of 16 configured subinterfaces
PPPoE Statistics
  Counters:
    PADI received      0
    PADI transmitted   16
    PADO received      16
    PADO transmitted   0
    PADR received      0
    PADR transmitted   16
    PADS received      16
    PADS transmitted   0
    PADT received      0
    PADT transmitted   0
    PADM received      16
    PADM transmitted   0
    PADN received      16
    PADN transmitted   0
  PAD packets received        32
  PAD packets transmitted     32
  Invalid PAD Packets:
    Invalid Version           0
    Invalid PAD Code          0
```

To display information about the PPPoE interface with the default MTU value (1494):

```
host1#show pppoe interface full
PPPoE interface FastEthernet 2/0 is operStatusUp
    PPPoE interface FastEthernet 2/0 has max sessions = 8000
    PPPoE interface FastEthernet 2/0 mtu 1494
    PPPoE interface FastEthernet 2/0 has no acName set
    PPPoE interface FastEthernet 2/0 autoconfigured subinterfaces
    PPPoE interface FastEthernet 2/0 has 1 active connections,
        out of 1 configured subinterfaces
Assigned profile (any)        : pppoetest
PPPoE Statistics
  Counters:
    PADI received      42
    PADI transmitted   0
    PADO received      0
    PADO transmitted   8
    PADR received      8
    PADR transmitted   0
    PADS received      0
    PADS transmitted   8
    PADT received      0
    PADT transmitted   7
```



```
    PADM received      0
    PADM transmitted   0
    PADN received      0
    PADN transmitted   0
  PAD packets received       50
  PAD packets transmitted    23

  Invalid PAD Packets:
    Invalid Version          0
    Invalid PAD Code         0
    Invalid PAD Tags         0
    Invalid PAD Tag length   0
    Invalid PAD Type         0
    Invalid PADI Session     0
    Invalid PADR Session     0
    Invalid PAD packet length 0
    Invalid PAD packets      0
   Total Invalid PAD packets  0

  Insufficient Resources 0

Lockout Configuration (seconds): Min 10, Max 120
 Total clients in active lockouts: 0
 Total clients in lockout grace period: 0
```

To display information about the PPPoE interface with MTU tag:

```
host1#show pppoe interface full
PPPoE interface FastEthernet 2/0 is operStatusUp
    PPPoE interface FastEthernet 2/0 has max sessions = 8000
    PPPoE interface FastEthernet 2/0 will use tag value for mtu
    PPPoE interface FastEthernet 2/0 has no acName set
    PPPoE interface FastEthernet 2/0 autoconfigured subinterfaces
    PPPoE interface FastEthernet 2/0 has 1 active connections,
        out of 1 configured subinterfaces
Assigned profile (any)       : pppoetest
PPPoE Statistics
  Counters:
    PADI received      44
    PADI transmitted   0
    PADO received      0
    PADO transmitted   10
    PADR received      10
    PADR transmitted   0
    PADS received      0
    PADS transmitted   10
    PADT received      0
    PADT transmitted   9
    PADM received      0
    PADM transmitted   0
    PADN received      0
    PADN transmitted   0
  PAD packets received       54
  PAD packets transmitted    29

  Invalid PAD Packets:
    Invalid Version          0
    Invalid PAD Code         0
    Invalid PAD Tags         0
    Invalid PAD Tag length   0
    Invalid PAD Type         0
    Invalid PADI Session     0
```



```
    Invalid PADR Session      0
    Invalid PAD packet length 0
    Invalid PAD packets       0
   Total Invalid PAD packets  0

  Insufficient Resources 0

Lockout Configuration (seconds): Min 5, Max 60
 Total clients in active lockouts: 0
 Total clients in lockout grace period: 0
```

To display information about the operational and administrative status of all configured PPPoE interfaces:

```
host1:01#show pppoe interface summary
Total PPPoE interfaces: 16

Administrative Status:
                    Up: 15
                  Down: 1

Operational Status:
                    Up: 15
                  Down: 1
        LowerLayerDown: 1
            NotPresent: 0
```

Meaning Table 40 on page 430 lists the **show pppoe interface** command output fields.

Table 40: show pppoe interface Output Fields

Field Name	Field Description
PPPoE interface	The interface identifier. NOTE: For more information about specifying the physical interface, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.
Status	Operational status of the interface. Possible values are: • operStatusUp—Interface or subinterface is operational • Down—Interface or subinterface is not operational • LowerLayerDown—Subinterface is not operational because an underlying interface is down
full	Displays configuration, status, and statistics information
max sessions	Configured maximum number of PPPoE sessions
mtu	Maximum transfer unit (MTU) value; when derived from the PPPoE MTU tag, the value can only be determined from an active session.
acName	Name of PPPoE access concentrator

Table 40: show pppoe interface Output Fields *(continued)*

Field Name	Field Description
will not send ac info in PADS packet	When the **pppoe pads disable-ac-info** command is issued, indicates that the router does not send the AC-Name and AC-Cookie tags in the PADS packet.
duplicate-protection	Whether duplicate protection is enabled or disabled for the interface
capture remote circuit id	Whether capture of the PPPoE remote circuit ID sent from the DSLAM is enabled or disabled for the interface
active connections	Number of live PPPoE connections
configured subinterfaces	Number of PPPoE subinterfaces you configured on an interface
Assigned profile	Name of profile assigned to dynamic PPPoE interface
PPPoE Statistics Counter	Number of control packets received or transmitted. Possible values: • PADI received/PADI transmitted—Number of initiation control packets received/transmitted • PADO received/PADO transmitted—Number of offer control packets received/transmitted • PADR received/PADR transmitted—Number of request control packets received/transmitted • PADS received/PADS transmitted—Number of session confirmation control packets received/transmitted • PADT received/PADT transmitted—Number of termination control packets received/transmitted • PADM received/PADM transmitted—Number of message control packets received/transmitted • PADN received/PADN transmitted—Number of network control packets received/transmitted
PAD packets received	Total number of control packets received on the interface
PAD packets transmitted	Total number of control packets transmitted on the interface

Table 40: show pppoe interface Output Fields *(continued)*

Field Name	Field Description
Invalid PAD Packets	Number of invalid packets received. Packets are considered invalid for various reasons. Possible values: • Invalid Version—Number of control packets received with an invalid version • Invalid PAD Code—Number of control packets received with an invalid code • Invalid PAD Tags—Number of control packets received with invalid tags • Invalid PAD Tag length—Number of control packets received with an invalid tag length • Invalid PAD Type—Number of control packets received with an invalid type • Invalid PADI Session—Number of invalid PPPoE Active Discovery Initiation sessions • Invalid PADR Session—Number of invalid PPPoE Active Discovery Request sessions • Invalid PAD packet length—Number of control packets received with an invalid packet length • Invalid PAD packets—Number of invalid control packets received
Total Invalid PAD packets	Total number of invalid control packets received on the interface
Insufficient Resources	Number of requests denied because of an inadequate number of sessions; check the number of active clients
Lockout Configuration (seconds)	Encapsulation type lockout settings for the PPPoE client associated with the dynamic PPPoE subinterface column. For more information about these fields, see "Encapsulation Type Lockout for PPPoE Clients Overview" on page 550. Possible values include: • Min—Minimum lockout time, in seconds • Max—Maximum lockout time, in seconds • Total clients in active lockouts—Number of PPPoE clients currently undergoing dynamic encapsulation type lockout • Total clients in lockout grace period—Number of PPPoE clients currently in a lockout grace period
Summary Statistics	Summary of the PPPoE statistics. Possible value: • Total PPPoE interfaces—Number of configured PPPoE interfaces included in summary

Table 40: show pppoe interface Output Fields *(continued)*

Field Name	Field Description
Administrative Status	Administrative status of the PPPoE interfaces. Possible values: • Up—Number of interfaces not affected by manual administrative intervention • Down—Number of interfaces that cannot flow because of manual administrative intervention
Operational Status	Operational status of the PPPoE interfaces. Possible values: • Up—Number of interfaces that are operational • Down—Number of interfaces that are not operational • LowerLayerDown—Number of interfaces that are not operational because an underlying interface is down • NotPresent—Number of interfaces that are not operational because hardware is unavailable

Related Documentation

- *show pppoe interface*

Monitoring the PPPoE Service Name Table

Purpose Display the contents of a PPPoE service name table configured on the router.

Action To display the names of PPPoE service name tables that you can specify to complete the command:

```
host1#show pppoe-service-name-table name ?
  myDefaultTable   myDefaultTable  service-name-table
  myServiceTable1  myServiceTable1 service-name-table
  myServiceTable2  myServiceTable2 service-name-table
  myServiceTable3  myServiceTable3 service-name-table
```

To display the contents of a default PPPoE service name table with no specific service name entries:

```
host1#show pppoe-service-name-table name myDefaultTable
Service Name Table myDefaultTable
 Empty service name action: terminate
Unknown service name action: drop
```

To display the contents of a PPPoE service name table that includes: empty service name tag and associated action, **terminate**; two custom service tags and their associated actions (**terminate** and **drop**); and the unknown service name tag and its default action (**terminate**).

```
host1#show pppoe-service-name-table name myServiceTable1
Service Name Table myServiceTable1
 Empty service name action: terminate
```

```
 Service name: myISPService action: terminate
 Service name: myQOSClass1 action: drop
 Unknown service name action: terminate
```

To display the names of all PPPoE service name tables configured on the router:

```
host1#show pppoe-service-name-table brief
Service-Name Table:
myServiceTable1
myServiceTable2
```

Meaning Table 41 on page 434 lists the **show pppoe-service-name-table** command output fields.

Table 41: show pppoe-service-name-table Output Fields

Field Name	Field Description
Service Name Table	Name of the PPPoE service name table configured with the **pppoe-service-name-table** command
Empty service name action	Action associated with the empty service name. Possible values: **terminate** or **drop**.
Service name	Name of the custom service name tag configured with the **service** command
action	Action (**terminate** or **drop**) associated with the service name tags configured using the **action** keyword with the **service** command
Unknown service name action	Action associated with an unknown service. Possible values: **terminate** or **drop**.

Related Documentation

- *show pppoe-service-name-table*

Monitoring the PPPoE Subinterface

Purpose Display information about the parameters associated with the PPPoE subinterface.

Action To display configuration, status, and statistics information of the PPPoE subinterface:

```
host1:v0#show pppoe subinterface full
show pppoe subinterface fullPPPoE subinterface FastEthernet 2/0.11 is operStatusUp
 (dynamic)
    PPPoE subinterface FastEthernet 2/0.11 has a session id of 8
    PPPoE subinterface FastEthernet 2/0.11 has source MAC address 0090.1a40.280a

    PPPoE subinterface FastEthernet 2/0.11 has a MTU of 1494
PPPoE Statistics
    In Octets:  165922
    Out Octets: 108283
    In Packets:  3607
    Out Packets: 3608
```


To display detailed information about the PPPoE subinterface:

```
host1:v0#show pppoe subinterface fastEthernet 1/1.1.1
PPPoE subinterface fastEthernet 1/1.1.1 is operStatusUp
      URL String: http://www.urlofinterest.com
      MOTM String: a horse walks into a bar
PPPoE subinterface fastEthernet 1/1.1.1 has a session id of 1
PPPoE Statistics
        In Octets:  480
        Out Octets: 256
        In Packets:  8
        Out Packets: 8
```

To display the operational and administrative status of all configured PPPoE subinterfaces:

```
host1:01#show pppoe subinterface summary
Total PPPoE subinterfaces: 116

Administrative Status:
                    Up: 115
                  Down: 1

Operational Status:
                    Up: 115
                  Down: 1
        LowerLayerDown: 1
            NotPresent: 0
```

Meaning Table 42 on page 435 lists the **show pppoe subinterface** command output fields.

Table 42: show pppoe subinterface Output Fields

Field Name	Field Description
PPPoE subinterface	Interface specifier
Status	Operational status of the interface. Possible values are: • operStatusUp—Interface or subinterface is operational • Down—Interface or subinterface is not operational • LowerLayerDown—Subinterface is not operational because an underlying interface or subinterface is down
URL String	URL string sent in the PADM message to PPPoE clients
MOTM String	Message of the minute string sent in the PADM message to PPPoE clients
session id	Session ID of the subinterface
source MAC address	MAC address of PPPoE client

Table 42: show pppoe subinterface Output Fields *(continued)*

Field Name	Field Description
MTU	Maximum transfer unit (MTU) value; when derived from the PPPoE MTU tag, the value can only be determined from an active session.
In Octets	Number of octets received on the subinterface
Out Octets	Number of octets transmitted on the subinterface
In Packets	Number of packets received on the subinterface
Out Packets	Number of packets transmitted on the subinterface
Total PPPoE subinterfaces	Number of configured PPPoE subinterfaces
Administrative Status	Administrative status of the PPPoE subinterface. Possible values: • Up—Number of subinterfaces not affected by manual administrative intervention • Down—Number of subinterfaces that cannot flow because of manual administrative intervention
Operational Status	Operational status of the PPPoE subinterface. Possible values: • Up—Number of subinterfaces that are operational • Down—Number of subinterfaces that are not operational • LowerLayerDown—Number of subinterfaces that are not operational because an underlying interface is down • NotPresent—Number of subinterfaces that are not operational because hardware is unavailable

Related Documentation

- *show pppoe subinterface*

Monitoring the PPPoE Profile

Purpose Display information about the profiles configured on the router and detailed information on a specific profile associated with PPPoE interface.

NOTE: Use the **name** keyword to display information about a specific profile. Use the **brief** keyword to display a list of profiles configured on the router.

Action To display configuration information for a PPPoE profile assigned to a dynamic interface:

```
host1#show profile name pppoeProfile
Profile                      : pppoeProfile
Unnumbered interface on      : loopback 1
Router                       : default
Directed Broadcast           : Disabled
ICMP Redirects               : Disabled
Access Route Addition        : Enabled
Network Address Translation: Disabled
Source-Address Validation    : Disabled
Ignore DF Bit                : Disabled
Filter Option Packets        : Disabled
Administrative MTU           : 1500
TCP MSS value                : 0
Inactivity Timer             : Disabled
Route Map Name               : Disabled
Auto Detect                  : Disabled
Auto Configure               : Disabled
Append VR Name with DSI      : Enabled

IGMP                         : Enabled
  static-groups              :
  Input policy:  bobb statistics enabled
  Output policy: bobb statistics enabled

PPP Keepalive                 : 30
PPP Magic Number              : enabled
PPP Peer DNS Priority         : disabled
PPP Peer WINS Priority        : disabled
PPP Authentication            : pap/chap
PPP Authentication Router     :
PPP Negotiate MRU             : (use lower layer MRU)
PPP Packet Log                : disabled
PPP State Log                 : disabled
PPP Chap Challenge Length     : 16 - 32
PPP Passive Mode              : disabled
PPP Multilink                 : disabled
PPP IPCP Netmask Option       : disabled
PPP AAA Profile               :
PPP Multilink Fragmentation : disabled
PPP Multilink Fragment Size : (use MTU)
PPP Multilink Reassembly      : disabled
PPP Multilink Mrru            : (use MRU)
PPP Initiate IP                      : disabled
PPP Initiate IPv6                    : disabled
PPPoE Max Sessions            : 2
PPPoE Always-offer            : Disabled
PPPoE Remote-Circuit-Id       : Enabled
PPPoE Log PPPoeControlPacket: Disabled
PPPoE duplicate-protect       : Enabled
PPPoE ACNAME                  : CYM9876
PPPoE URL                     : http://www.urlofinterest.com
PPPoE MOTM                    : goodmorning
PPPoE Service-Name table      : myServiceTable1
```

Meaning Table 43 on page 438 lists the **show profile** command output fields.

Table 43: show profile Output Fields

Field Name	Field Description
Profile	Name of the profile
IP address	IP address and subnet mask of the interface, or none if the interface is unnumbered
Unnumbered interface	Specifier for the unnumbered interface, or none if the interface is numbered
Router	Name of the virtual router (VR) assigned to the profile; interfaces created by the profile are attached to this VR.
Directed Broadcast	Whether the router supports directed broadcast. Possible values: enabled or disabled
ICMP Redirects	Whether the router accepts ICMP redirects. Possible values: enabled or disabled
Access Route Addition	Whether the router enables access route additions. Possible values: enabled or disabled.
Network Address Translation	Whether the router supports network address translation. Possible values: enabled or disabled; domain location (inside or outside)
Source-Address Validation	Router validates the source address associated with the profile. Possible values: enabled or disabled.
Ignore DF Bit	Router ignores the DF bit. Possible values: enabled or disabled.
Filter Option Packets	Router filters out packets with IP options. Possible values: enabled or disabled.
Administrative MTU	MTU size configured on the profile
TCP MSS value	Maximum segment size for TCP SYN packets traveling through the interface
Inactivity Timer	Inactivity timer setting. Possible values: enabled or disabled.
Route Map Name	Route map applied to the IP interface subscriber. Possible values: enabled or disabled.
Auto Detect	Router automatically detects packets that do not match any entries in the demultiplexer table. Possible values: enabled or disabled.

Table 43: show profile Output Fields *(continued)*

Field Name	Field Description
Auto Configure	Dynamic creation of subscriber interfaces on a primary IP interface. Possible values: enabled or disabled.
IGMP	Whether the router supports IGMP. Possible values: enabled or disabled.
static-groups	Displays address of any static groups configured for IGMP
Input policy	Name of input policy and whether statistics are enabled or disabled
Output policy	Name of output policy and whether statistics are enabled or disabled.
PPP Keepalive	Keepalive duration, in seconds
PPP Magic Number	Determines if the PPP magic number was negotiated. Possible values: enabled or disabled.
PPP Peer DNS Priority	Determines the DNS address that prevails in the event of a negotiation conflict. Possible values: enabled or disabled.
PPP Peer WINS Priority	Determines the WINS address that prevails in the event of a negotiation conflict. Possible values: enabled or disabled.
PPP Authentication	Type of authentication configured. Possible values: PAP, CHAP, or none.
PPP Authentication Router	Name of authentication virtual router.
PPP Negotiate MRU	MRU configured for the profile.
PPP Packet Log	Whether the packet trace log for PPP is configured. Possible values: enabled or disabled.
PPP State Log	Whether the state trace log for PPP is configured. Possible values: enabled or disabled.
PPP Chap Challenge Length	Minimum and maximum value of the Chap authenticator challenge length value.
PPP Passive Mode	Forces the PPP link into passive mode. Possible values: enabled or disabled.
PPP Multilink	Whether the router supports multilink. Possible values: enabled or disabled.

Table 43: show profile Output Fields *(continued)*

Field Name	Field Description
PPP IPCP netmask option	Whether the router supports negotiation of the IPCP option netmask during IPCP negotiation. Possible values: enabled or disabled.
PPP AAA Profile	AAA profile associated with this PPP interface
PPP Multilink Fragmentation	Whether the router supports fragmentation of the PPP multilink. Possible values: enabled or disabled.
PPP Multilink Fragment Size	Multilink fragment size for this PPP interface.
PPP Multilink Reassembly	Whether the router supports reassembly of the PPP multilink. Possible values: enabled or disabled.
PPP Multilink Mrru	Multilink MRRU value for this PPP interface
PPP Initiate IP	Initiation of IPv4 over this PPP interface. Possible values: enabled or disabled.
PPP Initiate IPv6	Initiation of IPv6 over this PPP interface. Possible values: enabled or disabled.
PPPoE Max Sessions	Maximum number of PPPoE subinterfaces that can be configured on an interface
PPPoE Always-offer	Router offers to set up a session for the client, even if the router has insufficient resources to establish a session. Possible values: enabled or disabled.
PPPoE Remote-Circuit-Id	The router captures and processes a vendor-specific tag containing a remote circuit ID transmitted from a digital subscriber line access multiplexer (DSLAM). Possible values: enabled or disabled.
PPPoE Log PPPoeControlPacket	Whether the packet trace log for the PPPoE interface is configured. Possible values: enabled or disabled.
PPPoE duplicate-protect	Whether the router can prevent a client from establishing more than one session using the same MAC address. Possible values: enabled or disabled.
PPPoE ACNAME	Name of the PPPoE access concentrator
PPPoE URL	URL sent in PADM message to PPPoE clients
PPPoE MOTM	Message of the minute sent in the PADM message to PPPoE clients
PPPoE Service-Name Table	Name of the PPPoE service name table, if configured for the specified profile.

Related Documentation

- *show profile*

Monitoring the RADIUS Override Settings

Purpose Display current override settings configured on the RADIUS client (LNS) for the NAS-IP-Address [4], NAS-Port-Id [87], Calling-Station-Id [31], and NAS-Identifier [32] RADIUS attributes.

Action To display the current override settings on the RADIUS client:

```
host1#show radius override
nas-ip-addr:        nas-ip-addr
nas-port-id:        remote-circuit-id
calling-station-id: remote-circuit-id
nas-info:           from current virtual router
```

Meaning Table 44 on page 441 lists the **show radius override** command output fields.

Table 44: show radius override Output Fields

Field Name	Field Description
nas-ip-addr	Override setting for the NAS-IP-Address attribute
nas-port-id	Override setting for the NAS-Port-Id attribute. Value is remote-circuit-id if configured with **radius override nas-port-id remote-circuit-id** command.
calling-station-id	Override setting for the Calling-Station-Id attribute. Value is remote-circuit-id if configured with **radius override calling-station-id remote-circuit-id** command.
nas-info	Virtual router that generates the NAS-IP-Address and NAS-Identifier attributes for AAA broadcast accounting packets; current virtual router or authentication virtual router

Related Documentation

- *show radius override*
- *show radius remote-circuit-id-delimiter*
- *show radius remote-circuit-id-format*

Troubleshooting PPPoE Interfaces

Problem To analyze and diagnose issues with the PPPoE interface, you must have access to detailed information on the most recent PPPoE operations. However, by default, nothing is traced.

Solution You can configure a packet trace log for a PPPoE interface to diagnose issues on a PPPoE interface. First, specify a PPPoE major interface which can be any one of the following interface types:

- **fastEthernet**
- **gigabitEthernet**
- **atm**
- **tenGigabitEthernet**

NOTE: For more information, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

The packet trace log for a PPPoE interface displays only the first 256 bytes of packet data. Data in excess of 256 bytes does not appear in the packet trace log. You also configure logging to direct the output to a specific destination. For information, see *Overview of System Logging*.

```
host1(config-if)#log severity debug pppoeControlPacket atm 10/0.1
host1:v0#DEBUG 07/25/2000 15:13:19 pppoeControlPacket (interface atm 10/0.1):
PADI rx from 00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:19 pppoeControlPacket (interface atm 10/0.1): PADO tx to
00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:19 pppoeControlPacket (interface atm 10/0.1): PADR rx from
 00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:19 pppoeControlPacket (interface atm 10/0.1): PADS tx to
00-09-01-a0-00-2e, connection made  using session id 3 on sub interface 1
RX-a0-00-2e:v0#
RX-a0-00-2e:v0#
RX-a0-00-2e:v0#
RX-a0-00-2e:v0#
RX-a0-00-2e:v0#
RX-a0-00-2e:v0#
RX-a0-00-2e:v0#config t
Enter configuration commands, one per line.  End with CNTL/Z.
RX-a0-00-2e:v0(config)#interface atm 10/1.1.1
RX-a0-00-2e:v0(config-if)#ppp shut
RX-a0-00-2e:v0(config-if)#DEBUG 07/25/2000 15:13:38 pppoeControlPacket (interface
 atm 10/0.1): PADT rx from 00-09-01-a0-00-2e
RX-a0-00-2e:v0(config-if)#
RX-a0-00-2e:v0(config-if)#no ppp shut
RX-a0-00-2e:v0(config-if)#pppoe test
RX-a0-00-2e:v0(config-if)#DEBUG 07/25/2000 15:13:49 pppoeControlPacket (interface
 atm 10/0.1): PADI rx from 00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:49 pppoeControlPacket (interface atm 10/0.1): PADO tx to
00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:49 pppoeControlPacket (interface atm 10/0.1): PADR rx from
 00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:49 pppoeControlPacket (interface atm 10/0.1): PADS tx to
00-09-01-a0-00-2e, connection made  using session id 4 on sub interface 1
```



```
RX-a0-00-2e:v0(config-if)#
RX-a0-00-2e:v0(config-if)#exit
```

Related Documentation

- *log severity*
- *pppoe log pppoeControlPacket*

CHAPTER 15

Configuring Bridged IP

E Series routers support bridged IP (1483) interfaces.

This chapter contains the following sections:

- Overview on page 445
- Platform Considerations on page 446
- References on page 447
- Before You Configure Bridged IP on page 447
- Configuring Bridged IP on page 448

Overview

You can configure a bridged IP interface to manage IP packets that are encapsulated inside an Ethernet frame running over a permanent virtual circuit (PVC).

When you configure a bridged IP interface, it automatically performs proxy Address Resolution Protocol (ARP). You can also configure the router as a relay agent that forwards Dynamic Host Configuration Protocol (DHCP) broadcasts.

Proxy ARP

Proxy ARP allows your router to respond to ARP requests on behalf of an Ethernet end node.

The router performs proxy ARP for the ARP requests that come in over the bridged IP interface when both of the following conditions are met:

- The IP address in the ARP request matches an entry in the routing table.
- The route is on a different interface than the one on which the router received the ARP request.

If you specify that the bridged IP interface performs unrestricted proxy ARP, it also performs proxy ARP when the route is on the interface that received the ARP request.

In most situations, do not configure the router to perform unrestricted proxy ARP. Do so for special situations, such as when cable modems are used. When an IP client broadcasts the ARP request across the Ethernet wire, the end node with the correct IP address responds to the ARP request and provides the correct MAC address. If the unrestricted proxy ARP feature is enabled, the router response is redundant and might fool the IP client into determining that the destination MAC address within its own subnet is the same as the address of the router.

DHCP

DHCP provides a mechanism through which hosts using TCP/IP can obtain protocol configuration parameters automatically from a DHCP server on the network.

The most important configuration parameter carried by DHCP is the IP address. A host must be initially assigned a specific IP address that is appropriate to the network to which the computer is attached, and that is not assigned to any other host on that network. If you move a host to a new network, you must give it a new IP address.

DHCP also carries other important configuration parameters such as the subnet mask, default router, and Domain Name System (DNS) server.

An IP client contacts a DHCP server for configuration parameters. The DHCP server is typically centrally located and operated by the network administrator. Because a network administrator manages the server, DHCP clients can obtain reliable parameters appropriate to the current network architecture.

For information about DHCP, see *DHCP Overview Information*.

Platform Considerations

You can configure bridged IP interfaces on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support bridged IP interfaces on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support bridged IP.

For information about the modules that support bridged IP interfaces on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support bridged IP.

Interface Specifiers

The configuration task examples in this chapter use the *slot/port*[.*subinterface*] format to specify the ATM physical interface on which you want to configure bridged IP. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

References

For more information about bridged IP, consult RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999). Note that RFC 2684 obsoletes RFC 1483.

Before You Configure Bridged IP

Before you configure bridged IP on an ATM interface, verify that:

- You have correctly installed a module that supports bridged IP. For information, see *ERX Module Guide, Appendix A, Module Protocol Support* (on ERX7xx models, ERX14xx models, and the ERX310 router) or *E120 and E320 Module Guide*, *Appendix A*, *IOA Protocol Support* (on the E120 or the E320 router).
- Each configured line can transmit data to and receive data from your switch connections.

Table 45 on page 448 lists the prerequisite tasks for configuring bridged IP and the resources that you can consult to learn how to perform these tasks.

Table 45: Prerequisite Tasks for Configuring Bridged IP

To Learn About	See
Preconfiguration and hardware diagnostic procedures	*ERX Hardware Guide* *E120 and E320 Hardware Guide*
Configuring T3 and E3 line modules	*Configuring T3 and E3 Interfaces* in *JunosE Physical Layer Configuration Guide*
Configuring OC3 line modules	*Configuring Unchannelized OCx/STMx Interfaces* in *JunosE Physical Layer Configuration Guide*

Also have the following information available:

- Interface specifiers for the ATM interfaces on which you want to configure bridged IP

  For more information about specifying ATM interfaces and subinterfaces on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.
- Subinterface numbers for each logical interface that you want to create
- Virtual path and channel numbers for each virtual circuit that you want to create
- IP addresses and subnet mask assignments for IP interfaces
- IP address of the DHCP server

Configuring Bridged IP

To configure an ATM interface using bridged IP encapsulation:

1. Configure a physical interface.

   host1(config)#**interface atm 0/1**
2. Configure the subinterface.

   host1(config-if)#**interface atm 0/1.20**
3. Configure a PVC on the subinterface by specifying the virtual circuit descriptor (VCD), the virtual path identifier (VPI), the virtual channel identifier (VCI), and the encapsulation type.

   host1(config-if)#**atm pvc 10 22 100 aal5snap**
4. Configure bridged IP encapsulation.

   host1(config-if)#**encapsulation bridge1483**
5. Assign an IP address and subnet mask to the PVC.

   host1(config-subif)#**ip address 192.168.10.20 255.255.255.0**

NOTE: You can also assign an IP template to the interface or create an unnumbered interface instead of assigning an IP address. For details, see *Configuring IP* in *JunosE IP, IPv6, and IGP Configuration Guide*.

6. (Optional) Use the appropriate **show** commands to verify your configuration.

   host1#**show atm interface 0/1**
   host1#**show atm vc 0/1 10**
   host1#**show atm subinterface 0/1.20**

   For more information about using these commands, see "Monitoring ATM" on page 66 in "Configuring ATM" on page 3.

atm pvc

- Use to configure a PVC on an ATM interface.
- The following fields are mandatory:
  - *vcd*—Unique number that identifies a virtual circuit in the range 1–2147483647. The VCD value has no relationship to the VPI and VCI values and has meaning only to the E Series router.
  - *vpi*—8-bit field in the ATM cell header. The VPI value is unique on a single link, not throughout the ATM network, because it has meaning only to the E Series router. The VPI value must match the value on the ATM switch.

    NOTE: Do not set both the VPI and VCI values to zero.

  - *vci*—16-bit field in the ATM cell header. The VCI value is unique on a single link, not throughout the ATM network, because it has meaning only to the router. You cannot set both the VPI and VCI to 0.
  - *encapsulation type*:
    - **aal5snap**—Specifies a logical link control (LLC) encapsulated circuit. An LLC/Subnetwork Access Protocol (LLC/SNAP) header precedes the protocol datagram.
    - **aal5muxip**—Specifies a multiplexed circuit used for IP only.
- Example

  host1(config-if)#**atm pvc 10 22 100 aal5snap**

- Use the **no** version to remove the specified PVC.
- See *atm pvc*.

encapsulation bridge1483

- Use to configure bridged IP as the encapsulation method on an interface.
- Use the **unrestrictedProxyArp** keyword to allow the router to perform unrestricted processing of ARP requests even if the route is on the same interface on which the request is received. See "Proxy ARP" on page 445 for details.
- Example

  host1(config-if)#**encapsulation bridge1483**
- Use the **no** version to remove bridged IP as the encapsulation method on the interface.
- See *encapsulation bridge1483*.

interface atm

- Use to configure an ATM interface.
- To specify an ATM interface for ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*.[*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM interface for E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- For more information, see "Creating a Basic Configuration" on page 21 in "Configuring ATM" on page 3.
- Examples

  host1(config)#**interface atm 3/1.20**
  host1(config)#**interface atm 5/0/1.20**
- Use the **no** version to remove the ATM subinterface or the logical interface.
- See *interface atm*.

CHAPTER 16

Configuring Bridged Ethernet

This chapter describes how to configure bridged Ethernet on E Series routers.

E Series routers also support bridged Ethernet on dynamic interfaces. See "Configuring Upper-Layer Dynamic Interfaces" on page 519, for details.

This chapter contains the following sections:

- Overview on page 451
- Platform Considerations on page 454
- References on page 455
- Configuring Bridged Ethernet on page 455
- Configuring VLANs over Bridged Ethernet on page 460
- Configuring S-VLANs over Bridged Ethernet on page 465
- Configuring the MTU Size for Bridged Ethernet on page 468
- Monitoring Bridged Ethernet on page 468

Overview

Bridged Ethernet allows multiple upper-layer interface types (IP and PPPoE) to be simultaneously multiplexed over the same interface. You can set up the router to either terminate interfaces and route data or to pass data transparently through the router to another terminating device. This capability supports multiple client devices that use both IP and Point-to-Point Protocol over Ethernet (PPPoE) encapsulation over an Ethernet LAN.

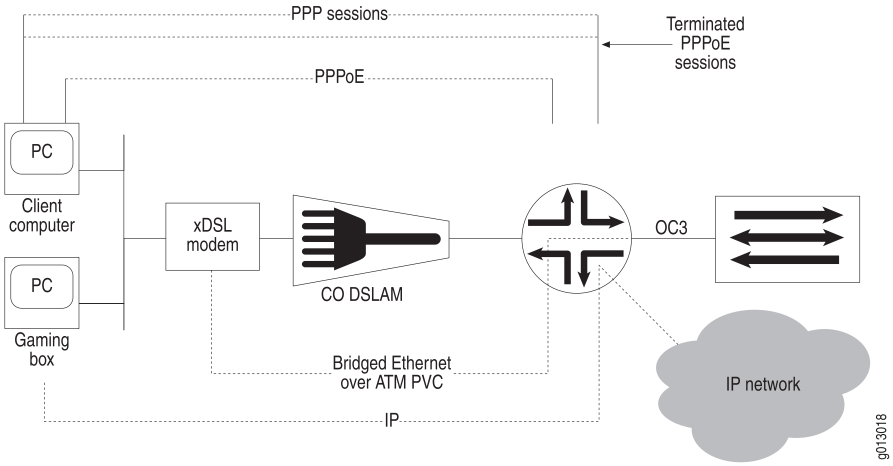
NOTE: Although connection-based forwarding is *not* supported on any E Series router, as an alternative, you can configure a local cross-connect, which uses layer 2 services over MPLS to transmit data between two layer 2 interfaces that reside on the same E Series router. Configuration of local cross-connects is supported on all E Series routers. For more information about configuring local cross-connects, see *Configuring Layer 2 Services over MPLS* in *JunosE BGP and MPLS Configuration Guide*.

Bridged Ethernet Application

Figure 43 on page 452 shows an example of a client computer using IP/PPP/PPPoE and an Internet gaming system running IP, connecting to the E Series router over the same ATM PVC. The client computer and gaming system can connect to an E Series router via an xDSL modem over a single ATM PVC, and the router can forward these two data streams independently. When the router receives the two data streams, it uses the Ethertype contained in the bridged Ethernet header to select which upper interface (IP or PPPoE) receives the frame.

In Figure 43 on page 452, IP and PPPoE interfaces are configured so that the non-PPPoE IP traffic is received by the IP interface, and the IP/PPP/PPPoE traffic is received by the PPPoE interface. Since the router receives these data streams on different IP interfaces, they may be routed independently.

Figure 43: Bridged Ethernet Topology, Router Terminating and Routing Traffic



Assigning MAC Addresses

When you create a bridged Ethernet interface, the system media access control (MAC) address is assigned to the interface by default. However, you can assign a specific MAC address to each statically configured bridged Ethernet interface. For example, if multiple statically configured bridged Ethernet interfaces are connected to the same device, using specific MAC addresses enables the connected device to select the correct ATM port or VC to use.

You configure a specific MAC address when you create the bridged Ethernet interface. If you want to modify an existing MAC address, you must remove the interface and create it again. Also, you cannot configure multicast MAC addresses on bridged Ethernet interfaces.

VLAN and S-VLAN Configurations

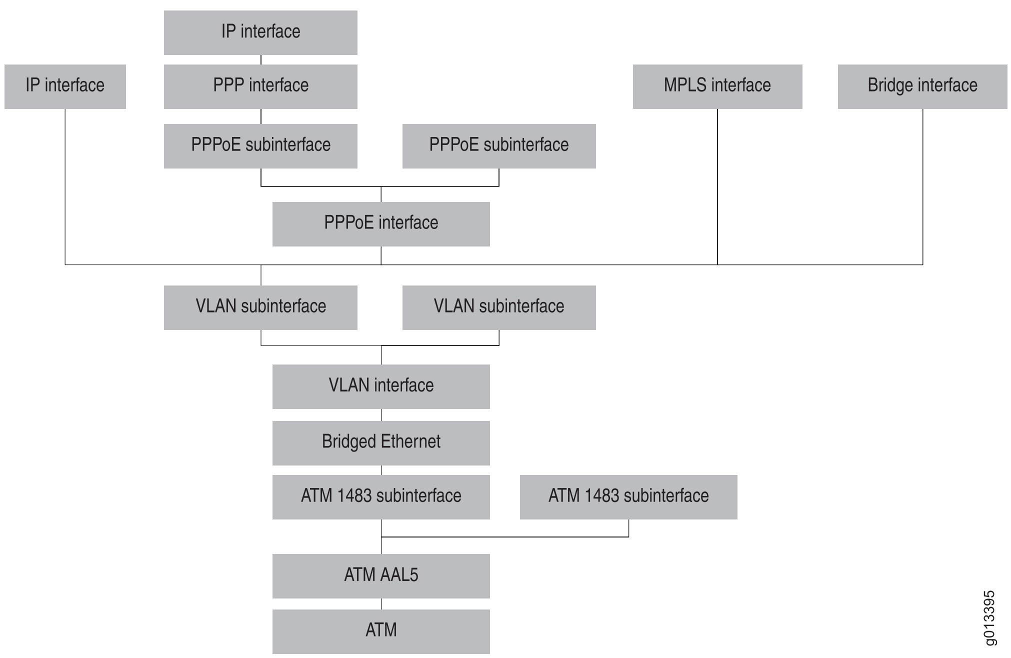
Bridged Ethernet interfaces on E Series routers support the configuration of virtual local area networks (VLANs) and stacked virtual local area networks (S-VLANs). A VLAN permits multiplexing multiple higher-level protocols over a single physical port. An S-VLAN provides a two-level VLAN tag structure, which extends the VLAN ID space to more than 16 million VLAN tags.

Specifically, you can statically configure the following higher-level protocols over a VLAN or an S-VLAN subinterface that is stacked above a bridged Ethernet interface:

- IP
- MPLS
- PPPoE
- Transparent bridging

Figure 44 on page 453 illustrates the interface stacking supported on E Series routers for VLANs over bridged Ethernet.

Figure 44: Interface Stacking for VLANs over Bridged Ethernet



VLANs and S-VLANs configured over bridged Ethernet interfaces provide the same basic capabilities as VLANs and S-VLANs configured over Ethernet interfaces, with the following exception:

- S-VLAN oversubscription is not supported on bridged Ethernet interfaces.

After you configure the bridged Ethernet interface, you configure the VLANs, S-VLANs, and the supported higher-level protocols in the same way that you configure them over Ethernet interfaces.

For more information, see:

- "Configuring VLAN and S-VLAN Subinterfaces" on page 165 for introductory information about VLANs and S-VLANs.

- "Configuring VLANs over Bridged Ethernet" on page 460 and "Configuring S-VLANs over Bridged Ethernet" on page 465 for examples that illustrate VLAN and S-VLAN configurations over bridged Ethernet.

Platform Considerations

You can configure bridged Ethernet on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support bridged Ethernet on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support bridged Ethernet.

For information about the modules that support bridged Ethernet on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support bridged Ethernet.

Interface Specifiers

The configuration task examples in this chapter use the *slot/port*[.*subinterface*] format to specify the ATM physical interface on which you want to configure bridged Ethernet. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot/adapter/port*[*.subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

host1(config)#interface atm 5/0/0.20

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

References

For more information about bridged Ethernet, consult the following resources:

- RFC 826—An Ethernet Address Resolution Protocol (November 1982)
- RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999)

Configuring Bridged Ethernet

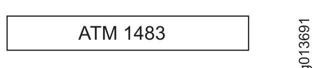
This section shows how to configure IP with PPPoE terminated at the E Series router. With each step, an illustration shows how the router is building the interface columns.

Configuring IP with PPPoE Terminated at the Router

This section shows how to create IP with PPPoE interfaces that terminate the connection and route the data received on the PVC, as shown in the example in Figure 43 on page 452. To create a terminated PVC:

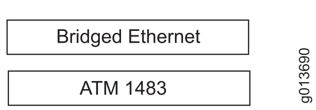
1. Create an ATM 1483 subinterface and associated PVC.

   host1(config)#interface atm 9/1.1 point-to-point
   host1(config-subif)#atm pvc 1 0 32 aal5snap 0 0 0



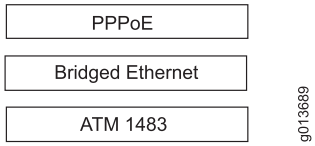
2. Encapsulate the ATM 1483 subinterface with bridged Ethernet. The use of the **encapsulation** keyword implies that the bridged Ethernet interface is the only interface stacked directly above the ATM 1483 subinterface. As a result, the bridged Ethernet interface cannot have a peer interface stacked above the same lower-layer interface.

   host1(config-subif)#encapsulation bridge1483



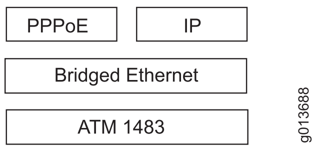
3. Create a PPPoE major interface over the bridged Ethernet interface. This command does not use the **encapsulation** keyword.

   host1(config-subif)#pppoe



4. Create an IP interface over the bridged Ethernet interface as a peer to the PPPoE interface.

   host1(config-subif)#ip address 160.1.0.1 255.255.255.0



5. (Optional) Set up the router to validate MAC addresses on the IP interface.

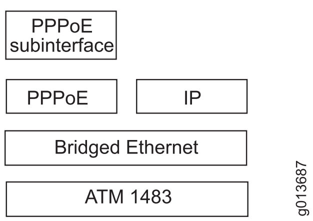
   host1(config-subif)#ip mac-validate strict

6. Exit the subinterface context.

   host1(config-subif)#exit

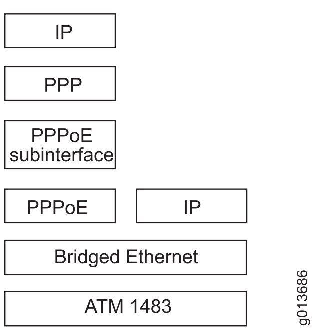
7. Create a PPPoE subinterface over the major interface.

   host1(config)#pppoe subinterface atm 9/1.1.1



8. Configure PPP encapsulation over the PPPoE subinterface, and the IP interface over the PPP interface.

   host1(config-subif)#encapsulation ppp
   host1(config-subif)#ip address 160.1.1.1 255.255.255.0



atm pvc

- Use to configure a PVC on an ATM interface. Specify one of the following encapsulation types:

- **aal5snap**—Specifies a logical link control (LLC) encapsulated circuit; LLC/Subnetwork Access Protocol (LLC/SNAP) header precedes the protocol datagram.
- **aal5mux ip**—Specifies a VC multiplexed circuit. This option is used for IP only.

- Example

  host1(config-subif)#**atm pvc 1 0 32 aal5snap 0 0 0**

- Use the **no** version to remove the specified PVC.
- See *atm pvc*.

encapsulation bridge1483

- Use to configure bridged Ethernet as the encapsulation method on an interface, and to optionally assign the MAC address to the interface.
- Use the **mac-address** keyword to configure a specific MAC address for the interface. Otherwise, the system MAC address is used. The same MAC address can be used on multiple interfaces.
- If the MAC address is configured, you must use the same MAC address whenever you reenter the **encapsulation bridge1483** command for the interface.
- The MAC address can be configured only when the interface is created. To change a MAC address, you must remove the interface and create it again.
- Example

  host1(config-subif)#**encapsulation bridge1483 mac-address 0090.1a01.1234**

- Use the **no** version, without the MAC address, to remove bridged Ethernet as the encapsulation method on the interface.
- See *encapsulation bridge1483*.

encapsulation ppp

- Use to configure PPP as the encapsulation method for an interface.
- Example

  host1(config-subif)#**encapsulation ppp**

- Use the **no** version to remove PPP as the encapsulation method on the interface.
- See *encapsulation ppp*.

interface atm

- Use to configure an ATM interface or subinterface type.
- To specify an ATM interface for ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*.[*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647

- To specify an ATM interface for E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format.
- *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- For more information, see “Creating a Basic Configuration” on page 21 in “Configuring ATM” on page 3.
- Examples

  host1(config)#**interface atm 9/1.1 point-to-point**
  host1(config)#**interface atm 5/0/1.1 point-to-point**
- Use the **no** version to remove the interface or subinterface.
- See *interface atm*.

ip address

- Use to set an IP address for the interface.
- Note that you cannot add more than one IP address to bridged Ethernet interfaces.
- Example

  host1(config-subif)#**ip address 160.1.0.1 255.255.255.0**
- Use the **no** version to remove the IP address.
- See *ip address*.

ip mac-validate

- Use to enable or disable MAC address validation on a per interface basis.
- When MAC address validation is enabled, the router checks the entry in the MAC validation table that corresponds to the IP source address of an incoming packet. The MAC source address of the packet must match the MAC source address of the table entry for the router to forward the packet.
- Use the **strict** keyword to forward packets only when both the IP source address and the MAC source address match one of the IP-MAC address pair entries in the table. When the MAC address in the table does not match the MAC source address, or when IP source address of the incoming packet does not match any of the IP addresses in the validation table, the packet is dropped.

NOTE: When a DHCP discover or a DHCP request packet arrives from a requesting client to the router that functions as the DHCP server or the delegating router on an interface, and if you configured either strict or loose mode of MAC address validation on that interface, the DHCP discover or request packets are processed correctly and are not dropped.

- Use the **loose** keyword to forward packets when both the IP source address and the MAC source address match one of the IP-MAC address pair entries in the MAC validation table. When the IP source address matches one of the IP source addresses in the table, but the MAC address of the incoming packet does not match the MAC address of the entry in the table, the packet is dropped. However, when the IP source address of the incoming packet does not match any of the IP addresses in the table, the packet is forwarded. This is the default setting.
- The default behavior is not to perform MAC address validation.
- Example

  host1(config-subif)#**ip mac-validate strict**
- Use the **no** version to disable the command.

NOTE: For more information, see *MAC Address Validation* in *JunosE IP, IPv6, and IGP Configuration Guide*.

- See *ip mac-validate*.

pppoe

- Use to create a PPPoE major interface.
- Example

  host1(config-subif)#**pppoe**
- Use the **no** version to remove the PPPoE major interface.
- See *pppoe*.

pppoe subinterface atm

- Use to create a PPPoE subinterface on an ATM interface.
- On ERX7xx models, ERX14xx models, and the ERX310 router, use the *slot/port.atmSubinterface.pppoeSubinterface* format.
- On the E120 and E320 routers, use the *slot/adapter/port.atmSubinterface.pppoeSubinterface* format. RLI-1050
- Examples

  host1(config)#**pppoe subinterface atm 9/1.1.1**
  host1(config)#**pppoe subinterface atm 5/0/1.1.1**

- Use the **no** version to remove the PPPoE subinterface.
- See *pppoe subinterface*.

Alternative Configuration

In previous releases, you could configure a PPPoE major interface directly over ATM 1483 only. The router still supports this stacking and configuration method for PPPoE. Although the older and newer interface stacks are different, they behave the same in terms of frame encapsulation and handling. As a result, an interface created using the older stacking will interoperate with an interface using the new stacking. Note, however, that the previous command syntax (**encapsulation pppoe**) cannot be used when a bridged Ethernet interface already exists, because it is intended to produce the old stacking for PPPoE over ATM 1483.

1. Create the ATM 1483 subinterface and associated PVC:

   ```
   host1(config)#interface atm 9/1.1 point-to-point
   host1(config-subif)#atm pvc 1 0 32 aal5snap 0 0 0
   ```

2. Create a PPPoE major interface over the ATM 1483 subinterface. Note that since this command uses the **encapsulation** keyword, it will fail if a bridged Ethernet interface was already created over the ATM 1483 subinterface using the new syntax.

   ```
   host1(config-subif)#encapsulation pppoe
   ```

3. Create a PPPoE subinterface over the major interface. Because PPPoE is the only top layer protocol in the stack, there is no need to use **pppoe** to identify the subinterface type (it is implied).

   ```
   host1(config)#interface atm 9/1.1.1
   ```

4. Configure the PPP encapsulation over the PPPoE subinterface, and the IP interface over the PPP interface.

   ```
   host1(config-subif)#encapsulation ppp
   host1(config-subif)#ip address 160.1.1.1 255.255.255.0
   ```

Configuring VLANs over Bridged Ethernet

This section describes how to create the following common static VLAN over bridged Ethernet configurations:

- IP over VLAN over bridged Ethernet
- PPPoE over VLAN over bridged Ethernet
- MPLS over VLAN over bridged Ethernet

You can also configure transparent bridging over VLANs over bridged Ethernet. For information about configuring transparent bridging, see "Configuring Bridged Ethernet" on page 451.

Configuring VLANs over bridged Ethernet interfaces consists of two basic tasks:

1. Configure the layers up to and including the VLAN subinterface. The steps for this task are common to all configurations.
2. Configure the higher-level protocols above the VLAN subinterface.

The following sections describe how to configure VLANs over bridged Ethernet. For more information about the commands used in these procedures, see the command descriptions listed in alphabetical order at the end of "Configuring Higher-Level Protocols over VLANs" on page 461.

NOTE: **Before you can remove a VLAN subinterface, you must remove the upper-layer interface stack.**

For more information about specifying ATM interfaces and subinterfaces, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Configuring VLAN Subinterfaces over Bridged Ethernet

To configure a VLAN subinterface over bridged Ethernet:

1. Create an ATM 1483 subinterface and associated PVC.

   ```
   host1(config)#interface atm 4/0.101
   host1(config-subif)#atm pvc 1 0 32 aal5snap 0 0 0
   ```

2. Specify bridged Ethernet as the encapsulation method for the ATM 1483 subinterface.

   ```
   host1(config-subif)#encapsulation bridge1483
   ```

3. Create a VLAN major interface by specifying VLAN as the encapsulation method for the bridged Ethernet interface.

   ```
   host1(config-subif)#encapsulation vlan
   ```

4. Create a VLAN subinterface to carry the higher-level protocols by adding a subinterface number to the interface identification command.

   ```
   host1(config-subif)#interface atm 4/0.101.1
   ```

5. Assign a VLAN ID for the subinterface.

   ```
   host1(config-subif)#vlan id 10
   ```

6. (Optional) Configure additional VLAN subinterfaces by repeating Steps 4 and 5, using unique values.

   ```
   host1(config-subif)#interface atm 4/0.101.2
   host1(config-subif)#vlan id 11
   ```

Proceed to the next section for instructions on configuring higher-level protocols over the VLAN subinterfaces.

Configuring Higher-Level Protocols over VLANs

This section provides examples for configuring IP, PPPoE, and MPLS interfaces over VLAN subinterfaces configured on bridged Ethernet. These procedures assume that you have already configured one or more VLAN subinterfaces over the bridged Ethernet interface to carry the higher-level protocols.

Configuring IP over VLAN

To configure IP over VLAN over a bridged Ethernet interface:

1. Follow the steps in "Configuring VLAN Subinterfaces over Bridged Ethernet" on page 461 to configure the VLAN subinterface.
2. Assign an IP address and mask to the VLAN subinterface.

   host1(config-subif)#**ip address 10.1.1.1 255.255.255.0**

Configuring PPPoE over VLAN

To configure PPPoE over VLAN over a bridged Ethernet interface:

1. Follow the steps in "Configuring VLAN Subinterfaces over Bridged Ethernet" on page 461 to configure the VLAN subinterface.
2. Create a PPPoE major interface on the VLAN subinterface.

   host1(config-subif)#**pppoe**
3. Exit the subinterface context.

   host1(config-subif)#**exit**
4. Create a PPPoE subinterface by adding a subinterface number to the interface identification command.

   host1(config)#**pppoe subinterface atm 4/0.101.2.1**
5. Specify PPP as the encapsulation method on the interface.

   host1(config-subif)#**encapsulation ppp**
6. Assign an IP address and mask to the interface.

   host1(config-subif)#**ip address 10.1.1.2 255.255.255.0**

Configuring MPLS over VLAN

To configure MPLS over VLAN over a bridged Ethernet interface:

1. Follow the steps in "Configuring VLAN Subinterfaces over Bridged Ethernet" on page 461 to configure the VLAN subinterface.
2. Enable MPLS on the VLAN subinterface.

   host1(config-subif)#**mpls**

atm pvc

- Use to configure a PVC on an ATM interface. Specify one of the following encapsulation types:

    - **aal5snap**—Specifies a logical link control (LLC) encapsulated circuit; LLC/Subnetwork Access Protocol (LLC/SNAP) header precedes the protocol datagram.
    - **aal5mux ip**—Specifies a VC multiplexed circuit. This option is used for IP only.
- Example

  host1(config-subif)#**atm pvc 1 5 50 aal5snap 0 0 0**
- Use the **no** version to remove the specified PVC.
- See *atm pvc*.

encapsulation bridge1483

- Use to configure bridged Ethernet as the encapsulation method on an ATM 1483 subinterface.
- Example

  host1(config-subif)#**encapsulation bridge1483**
- Use the **no** version to remove bridged Ethernet as the encapsulation method on the interface.
- See *encapsulation bridge1483*.

encapsulation ppp

- Use to configure PPP as the encapsulation method on an interface.
- Example

  host1(config-subif)#**encapsulation ppp**
- Use the **no** version to remove PPP as the encapsulation method on the interface.
- See *encapsulation ppp*.

encapsulation vlan

- Use to configure VLAN as the encapsulation method on an interface.
- Example

  host1(config-subif)#**encapsulation vlan**
- Use the **no** version to remove VLAN as the encapsulation method on the interface.
- See *encapsulation vlan*.

interface atm

- Use to configure an ATM interface, ATM 1483 subinterface, or VLAN subinterface.
- On ERX7xx models, ERX14xx models, and the ERX310 router, use the *slot/port.subinterface.vlanSubinterface* format.
- On E120 and E320 routers, use the *slot/adapter/port.subinterface.vlanSubinterface* format.

- For more information, see “Creating a Basic Configuration” on page 21 in “Configuring ATM” on page 3.
- Example 1—Configures a VLAN subinterface over bridged Ethernet on ERX7xx models, ERX14xx models, and the ERX310 router

  host1(config)#**interface atm 4/2.2 point-to-point**
  host1(config-subif)#**interface atm 4/2.2.3**
- Example 2—Configures a VLAN subinterface over bridged Ethernet on the E320 router

  host1(config)#**interface atm 4/0/2.2 point-to-point**
  host1(config-subif)#**interface atm 4/0/2.2.3**
- Use the **no** version to remove the interface or subinterface.
- See *interface atm*.

ip address

- Use to set an IP address for the interface.
- Note that you cannot add more than one IP address to bridged Ethernet interfaces.
- Example

  host1(config-subif)#**ip address 10.1.2.3 255.255.255.255**
- Use the **no** version to remove the IP address.
- See *ip address*.

mpls

- Use to enable, disable, or delete MPLS on an interface. MPLS is disabled by default.
- Example

  host1(config)#**mpls**
- Use the **no** version to halt MPLS on the interface and delete the MPLS interface configuration.
- See *mpls*.

pppoe

- Use to create a PPPoE major interface.
- Example

  host1(config-subif)#**pppoe**
- Use the **no** version to remove the PPPoE major interface.
- See *pppoe*.

pppoe subinterface atm

- Use to create a PPPoE subinterface over a VLAN subinterface configured on a bridged Ethernet interface.
- On ERX7xx models, ERX14xx models, and the ERX310 router, use the *slot/port.atmSubinterface.vlanSubinterface.pppoeSubinterface* format.
- On E120 and E320 routers, use the *slot/adapter/port.atmSubinterface.vlanSubinterface.pppoeSubinterface* format.
- Examples

  host1(config)#**pppoe subinterface atm 4/0.1.2.1**
  host1(config)#**pppoe subinterface atm 4/1/0.1.2.1**

- Use the **no** version to remove the PPPoE subinterface.
- See *pppoe subinterface*.

vlan id

- Use to specify the VLAN ID.
- Use a VLAN ID that is in the range 0–4095 and is unique within the interface.
- Issue the **vlan id** command before any upper bindings are made, such as IP or PPPoE.
- Use the optional keyword **untagged** to specify that frames be sent untagged. The keyword is valid only for VLAN ID 0. It allows tagged frames to be received, but sends out untagged frames.
- Example

  host1(config-subif)#**vlan id 400**

- There is no **no** version.
- See *vlan id*.

Configuring S-VLANs over Bridged Ethernet

S-VLANs over bridged Ethernet support the same set of higher-level protocols as VLANs over bridged Ethernet. You configure S-VLANs over bridged Ethernet in the same way that you configure VLANs over bridged Ethernet, with the addition of certain commands.

Like VLANs, configuring S-VLANs over bridged Ethernet interfaces consists of two basic tasks:

1. Configure the layers up to and including the S-VLAN subinterface.
2. Configure the higher-level protocols above the S-VLAN subinterface.

Before you can remove an S-VLAN subinterface, you must remove the upper-layer interface stack.

NOTE: S-VLAN oversubscription is not supported on bridged Ethernet interfaces.

For more information about specifying ATM interfaces and subinterfaces, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Configuring S-VLAN Subinterfaces over Bridged Ethernet

To configure an S-VLAN subinterface over bridged Ethernet:

1. Create an ATM 1483 subinterface and associated PVC.

   host1(config)#**interface atm 3/1.1**
   host1(config-subif)#**atm pvc 1 5 33 aal5snap 0 0 0**

2. Specify bridged Ethernet as the encapsulation method for the ATM 1483 subinterface.

   host1(config-subif)#**encapsulation bridge1483**

3. Create a VLAN major interface by specifying VLAN as the encapsulation method for the bridged Ethernet interface.

   host1(config-subif)#**encapsulation vlan**

4. Create a VLAN subinterface to carry the higher-level protocols by adding a subinterface number to the interface identification command.

   host1(config-subif)#**interface atm 3/1.1.1**

5. Assign an S-VLAN ID and a VLAN ID for the subinterface.

   host1(config-subif)#**svlan id 4 255**

6. Assign an S-VLAN Ethertype.

   host1(config-subif)#**svlan ethertype 9200**

Proceed to "Configuring Higher-Level Protocols over S-VLANs" on page 467 for information about configuring higher-level protocols over the S-VLAN subinterfaces.

svlan ethertype

- Use to assign an Ethertype value for the S-VLAN subinterface.
- Choose one of the following Ethertype values:
  - 8100—Specifies Ethertype value 0x8100, as defined in IEEE Standard 802.1q
  - 9100—Specifies Ethertype value 0x9100, which is the default
  - 9200—Specifies Ethertype value 0x9200
- Use an Ethertype value that matches the Ethertype value set on the customer premises equipment (CPE) to which your router connects.
- Example

  host1(config-if)#**svlan ethertype 8100**

- Use the **no** version to restore the default value, 9100.
- See *svlan ethertype*.

svlan id

- Use to assign S-VLAN IDs and VLAN IDs to VLAN subinterfaces.
- Use S-VLAN ID and VLAN ID numbers that are in the range 0–4095 and that are unique within the Ethernet interface.
- Issue the **svlan id** command before any upper bindings are made, such as IP or PPPoE.
- Example

  host1(config-if)#**svlan id 4 255**

- There is no **no** version.
- See *svlan id*.

Configuring Higher-Level Protocols over S-VLANs

The procedures for configuring IP, PPPoE, and MPLS protocols over S-VLANs on bridged Ethernet interfaces are identical to the procedures for configuring these protocols over VLANs on bridged Ethernet interfaces.

This section provides an example for configuring PPPoE interfaces over S-VLAN subinterfaces configured on bridged Ethernet. For descriptions of the commands used in this procedure, see "Configuring Higher-Level Protocols over VLANs" on page 461.

To configure PPPoE over S-VLAN over a bridged Ethernet interface:

1. Follow the steps in "Configuring S-VLAN Subinterfaces over Bridged Ethernet" on page 466 to configure the S-VLAN subinterface.
2. Create a PPPoE major interface on the S-VLAN subinterface.

   host1(config-subif)#**pppoe**

3. Exit the subinterface context.

   host1(config-subif)#**exit**

4. Create a PPPoE subinterface by adding a subinterface number to the interface identification command.

   host1(config)#**pppoe subinterface atm 3/1.1.1.1**

5. Specify PPP as the encapsulation method on the interface.

   host1(config-subif)#**encapsulation ppp**

6. Assign an IP address and mask to the interface.

   host1(config-subif)#**ip address 10.1.2.3 255.255.255.255**

Configuring the MTU Size for Bridged Ethernet

You can use the **bridge1483 mtu** command to configure a nondefault maximum transmission unit (MTU) size, in bytes, for a bridged Ethernet interface. The default MTU size for a bridged Ethernet interface is 1518 bytes.

Because you configure a bridged Ethernet interface over an ATM 1483 subinterface, the MTU size set with the **bridge1483 mtu** command is limited by the MTU set for the underlying ATM 1483 subinterface. As a result, the **bridge1483 mtu command requires you to configure an** MTU size for the bridged Ethernet interface that does not exceed the maximum allowable MTU size for the underlying ATM 1483 subinterface, 9180 bytes.

The configured MTU size for an interface is referred to as its *administrative MTU*, and the MTU size at which the interface actually operates is referred to as its *operational MTU*. For bridged Ethernet interfaces, the operational MTU is the lesser of the following two values:

- The administrative MTU of the bridged Ethernet interface
- The administrative MTU of the underlying ATM 1483 subinterface

You can also use the **bridge1483 mtu** command in a profile to configure a nondefault MTU size for a dynamic bridged Ethernet interface. For information, see "Dynamic Interface Configuration Using a Profile" on page 565.

bridge1483 mtu

- Use to set the maximum allowable size, in bytes, of the MTU for bridged Ethernet interfaces.
- Specify an MTU size in the range 64–9180 bytes.
- Example

  ```
  host1(config-subif)#bridge1483 mtu 1684
  ```
- Use the **no** version to restore the default MTU size for bridged Ethernet interfaces, 1518 bytes.
- See *bridge1483 mtu*.

Monitoring Bridged Ethernet

You can:

- Display information about bridged Ethernet interfaces by using the **show bridge1483 interface** command.
- Monitor MAC address validation by using the **show ip mac-validate interface** command.
- Display information about VLANs configured on bridged Ethernet interfaces by using the **show vlan subinterface** command.

Bridged Ethernet interfaces are not bound to a specific virtual router (VR).

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

show bridge1483 interface

- Use to display configuration and status information for all bridged Ethernet interfaces currently configured on the router.
- Use the **atm** keyword and an interface specifier to display information for a bridged Ethernet interface that is stacked over an ATM subinterface.
- Use the **summary** keyword to display only a count of all bridged Ethernet interfaces configured on the router.
- Field descriptions
  - Interface—Type and specifier of the lower-layer interface on which bridged Ethernet is configured
  - Status—Status of the bridged Ethernet interface: up, down, lowerLayerDown, notPresent
  - MAC Address—MAC address assigned to the bridged Ethernet interface, if configured
  - Type—Type of interface: static or dynamic
  - Oper/Admin MTU—Operational MTU, which is the MTU at which the interface actually operates, and administrative MTU, which is the MTU configured for the interface; the administrative MTU displays 1518 (the default value) if not configured
  - In—Analysis of inbound traffic on this interface
    - Bytes—Number of bytes received in error-free packets
    - Packets—Number of packets received
    - Multicast—Number of multicast packets received
    - Broadcast—Number of broadcast packets received
    - Errors—Total number of errors in all received packets; some packets might contain more than one error
    - Discards—Total number of discarded incoming packets
  - Out—Analysis of outbound traffic on this interface
    - Bytes—Number of bytes sent
    - Packets—Number of packets sent
    - Multicast—Number of multicast packets sent
    - Broadcast—Number of broadcast packets sent

- Errors—Total number of errors in all transmitted packets; some packets might contain more than one error
- Discards—Total number of discarded outgoing packets

- ARP Statistics—Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface
  - ARP requests—Number of ARP requests
  - ARP responses—Number of ARP responses
  - Errors—Total number of errors in all ARP packets
  - Discards—Total number of discarded ARP packets
- Total bridge1483 interfaces—Total number of bridged Ethernet interfaces configured on the router; this is the only information that appears when you specify the **summary** keyword

- Example 1—Displays full configuration and status information

```
host1#show bridge1483 interface
                                                               Oper/Admin
        Interface           Status    MAC Address    Type        MTU
------------------------- ---------- -------------- ------- ----------
ATM 5/1.1                    Up       ----.----.---- Static  1500/1684
ATM 5/1.2                    Up       ----.----.---- Static  8192/9188
2 bridge1483 interfaces found
```

- Example 2—Displays full status and configuration information for the specified bridge1483 interface

```
host1#show bridge1483 interface atm 12/0.1
                                                               Oper/Admin
        Interface           Status    MAC Address    Type        MTU
------------------------- ---------- -------------- ------- ----------
ATM 12/0.1                Up         ----.----.---- Static  1522/1522

  In: Bytes 0, Packets 0
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
  Out: Bytes 0, Packets 0
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

- Example 3—Displays only brief summary information

```
host1#show bridge1483 interface summary
Total bridge1483 interfaces: 3
```

- See *show bridge1483 interface*.

show ip mac-validate interface

- Use to display the status of the MAC address validation on the physical interface.
- Field descriptions
  - *interfaceSpecifier*—Interface type slot/port
  - Keyword assigned to interface—Strict or Loose
  - Address—IP address of the entry
  - Hardware Addr—Physical (MAC) address of the entry
- Example

```
host1:vr1#show ip mac-validate interface atm 8/0.1
ATM8/0.1:  Strict

         Address     Hardware Addr
         180.1.0.2   0000.1111.2222
```

- See *show ip mac-validate interface*.

show vlan subinterface

- Use to display configuration and status information for a specified VLAN subinterface or for all VLAN subinterfaces configured on the router.
- Use the **summary** keyword to display only the counts of all VLAN subinterfaces and VLAN major interfaces configured on the router.
- Field descriptions
  - Interface—Type and specifier of the VLAN subinterface. For more information about specifying the ATM physical interface on which you want to configure the VLAN subinterface, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.
  - Status—Status of the VLAN subinterface: up, down, lowerLayerDown, notPresent
  - MTU—Maximum allowable size (in bytes) of the maximum transmission unit for the VLAN subinterface
  - Svlan Id—S-VLAN ID value, if configured
  - Vlan Id—VLAN ID value for the VLAN subinterface
  - Ethertype—S-VLAN Ethertype value, if configured
  - Total VLAN interfaces—Total numbers of VLAN subinterfaces and VLAN major interfaces configured on the router; this is the only field that appears when you specify the **summary** keyword
- Example 1—Displays full status and configuration information for all VLAN subinterfaces configured on the router

```
host1#show vlan subinterface
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype
------------------------- ---------- ---- -------- -------- ---------
ATM 3/0.1.2               Up         1522 ----     11       ----
ATM 3/0.1.3               Up         1522 ----     12       ----
ATM 3/1.1.1               Up         1522 ----     13       ----
ATM 3/1.1.2               Up         1522 ----     14       ----
ATM 3/2.1.1               Down       1526 4        255      0x9200
```

```
FastEthernet 4/5.1        Up         1522 ----     1        ----
6 vlan subinterfaces found
```

- Example 2—Displays full status and configuration information for the specified VLAN subinterface

```
host1#show vlan subinterface atm 3/0.1.2
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype
------------------------- ---------- ---- -------- -------- ---------
ATM 3/0.1.2               Up         1522 ----     11       ----
```

- Example 3—Displays only brief summary information for all VLAN subinterfaces configured on the router

```
host1#show vlan subinterface summary
Total VLAN interfaces: 6 subinterfaces, 3 major interfaces
```

- See *show vlan subinterface*.

CHAPTER 17

Configuring Transparent Bridging

This chapter provides an introduction to transparent bridging and describes how to configure transparent bridging on E Series routers.

This chapter contains the following sections:

- Overview on page 473
- Platform Considerations on page 478
- References on page 479
- Before You Configure Transparent Bridging on page 479
- Configuration Tasks on page 480
- Configuration Examples on page 493
- Monitoring Transparent Bridging on page 495

Overview

This section introduces important concepts that you need to understand before configuring transparent bridging. These concepts include:

- How Transparent Bridging Works on page 473
- Bridge Groups and Bridge Group Interfaces on page 474
- Bridge Interface Types and Supported Configurations on page 475
- Subscriber Policies on page 476
- Concurrent Routing and Bridging on page 477
- Transparent Bridging and VPLS on page 478
- Unsupported Features on page 478

How Transparent Bridging Works

A *transparent bridge* is a data-link layer (layer 2) relay device that connects two or more networks or network systems. When a transparent bridge powers up, it automatically begins learning the network topology by examining the media access control (MAC) source address of every incoming packet. The bridge then creates an entry in the forwarding table consisting of the address and associated interface where the packet was received.

More specifically, a transparent bridge performs all of the following actions to learn the network topology:

- Learning—The bridge examines the MAC address of every incoming packet, records the MAC address and associated interface in the forwarding table, and manages the database of MAC addresses and their associated interfaces.
- Flooding—When a packet's destination address does not match any entries in the forwarding table, the bridge transmits (floods) the packet on all bridge interfaces to all network segments except the interface on which the packet was received.
- Forwarding—Once the bridge has learned a packet's destination address (that is, has a matching entry in its forwarding table), the bridge uses the associated port and interface information to send the packet toward its destination.
- Filtering—If the bridge detects that a packet's source and destination addresses are on the same network segment, it ignores (filters) that packet. *Filtering* is the process by which the bridge can screen network traffic for certain characteristics and determine whether to forward or discard (drop) that traffic based on user-defined criteria. On E Series routers, filtering criteria can include the MAC source address, MAC destination address, and protocol type.
- Aging—When a bridge adds a dynamic (learned) MAC address entry to the forwarding table, it assigns an age to the entry. The bridge updates this age each time it receives a packet. To manage MAC entries more efficiently, you can configure an entry's aging time, which is the maximum time that an entry can remain in the forwarding table before it "ages out."

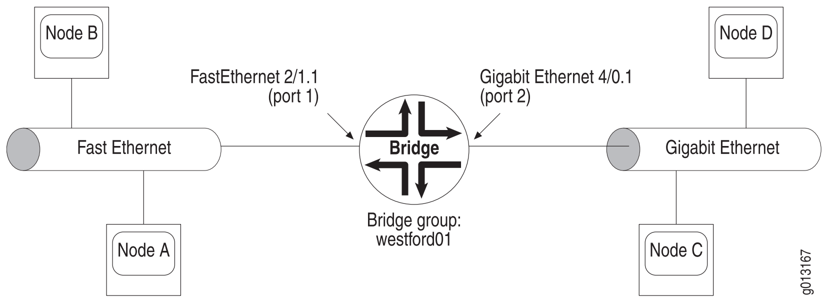
Bridge Groups and Bridge Group Interfaces

You configure transparent bridging by creating one or more bridge groups on the router. A *bridge group* is a collection of network interfaces (ports) that forms a broadcast domain. Each bridge group has its own set of forwarding tables and filters and, as such, functions as a logical transparent bridging device. For information about the maximum number of bridge groups that you can configure per E Series router, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

After you create a bridge group, you associate one or more network interfaces with the bridge group. This association is called a *bridge group interface*, or simply *bridge interface*. For information about the maximum number of bridge interfaces that you can configure per line module and per E Series router, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

Figure 45 on page 475 shows an example of a simple transparent bridging network configuration that illustrates the concepts discussed so far in this section.

Figure 45: Bridge Group with Fast Ethernet and Gigabit Ethernet Bridge Interfaces



In Figure 45 on page 475, a bridge group named westford01 is configured on the E Series router, which allows the router to function as a transparent bridge between a Fast Ethernet LAN segment and a Gigabit Ethernet LAN segment. The bridge group includes two bridge interfaces. The bridge interface associated with port 1 is stacked on a VLAN subinterface over a Fast Ethernet interface. The bridge interface associated with port 2 is stacked on a VLAN subinterface over a Gigabit Ethernet interface.

Table 46 on page 475 presents a simple representation of the forwarding table for bridge group westford01.

Table 46: Sample Bridge Group Forwarding Table

Port	Source Address	Interface
1	Node A	Fast Ethernet 2/1.1
1	Node B	Fast Ethernet 2/1.1
2	Node C	Gigabit Ethernet 4/0.1
2	Node D	Gigabit Ethernet 4/0.1

Bridge Interface Types and Supported Configurations

A bridge interface can be configured as one of the following types:

- Subscriber (client)—A subscriber (client) bridge interface is *downstream* from the traffic flow; that is, the traffic flow direction is from the server (trunk) to the client (subscriber). This is the default bridge group interface type.
- Trunk (server)—A trunk (server) bridge interface is *upstream* from the traffic flow; that is, the traffic flow direction is from the client (subscriber) to the server (trunk). To configure a trunk bridge group interface, you must specify the **subscriber-trunk** keyword as part of the **bridge-group** command.

You can configure bridge interfaces to add transparent bridging capabilities to your existing network configurations. Currently, bridge interfaces can be stacked on:

- Bridged Ethernet over ATM 1483 subinterfaces
- Fast Ethernet interfaces
- Gigabit Ethernet interfaces
- 10-Gigabit Ethernet interfaces
- VLAN subinterfaces over Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, or bridged Ethernet interfaces

For sample configurations that include bridge interfaces, see "Configuration Examples" on page 493. For information about configuring Ethernet, ATM, and bridged Ethernet interfaces, see:

- "Configuring ATM" on page 3
- "Configuring VLAN and S-VLAN Subinterfaces" on page 165
- "Configuring Bridged Ethernet" on page 451
- *Configuring Ethernet Interfaces* in *JunosE Physical Layer Configuration Guide*

Subscriber Policies

To enable intelligent flooding of packets within a bridge group's broadcast domain, each bridge group interface you create is associated with a default subscriber policy. A *subscriber policy* is a set of forwarding and filtering rules that defines how the bridge group interface handles various packet or attribute types, as follows:

- For each packet type, the subscriber policy specifies whether you want the bridge group interface to permit (forward) or deny (filter or drop) packets of that type.
- For the relearn attribute, the subscriber policy specifies whether the bridge interface can relearn a MAC address entry on a different interface from the one initially associated with this entry in the forwarding table. Permit indicates that relearning is allowed, and deny indicates that relearning is prohibited.

The router provides two default subscriber policies: default Subscriber for subscriber (client) bridge interfaces, and default Trunk for trunk (server) bridge interfaces.

Table 47 on page 476 lists the default values for each packet or attribute type defined in the default Subscriber and default Trunk policies. The only difference between the two policies is how broadcast packets and packets with unknown unicast destination addresses (DAs) are handled.

Table 47: Default Subscriber Policies for Bridge Group Interfaces

Packet/Attribute Type	Default Subscriber Policy	Default Trunk Policy
ARP	Permit	Permit
Broadcast	Deny	Permit
IP	Permit	Permit

Table 47: Default Subscriber Policies for Bridge Group Interfaces *(continued)*

Packet/Attribute Type	Default Subscriber Policy	Default Trunk Policy
MPLS	Permit	Permit
Multicast	Permit	Permit
PPPoE	Permit	Permit
Relearn	Permit	Permit
Unicast (user-to-user)	Permit	Permit
Unknown unicast DA	Deny	Permit
Unknown protocol	Permit	Permit

You cannot change the default subscriber policy values listed in Table 47 on page 476 for a trunk bridge interface. You can, however, configure a nondefault subscriber policy for a subscriber bridge interface to change the default permit or deny value for one or more packet or attribute types. For details, see “Configuring Subscriber Policies” on page 486.

Concurrent Routing and Bridging

After you create the necessary bridge groups and bridge interfaces for your network configuration, you can use the **bridge crb** command to enable concurrent routing and bridging (CRB) for all bridge groups configured on your router. When CRB is enabled, the router can route a protocol among a group of interfaces in one bridge group and concurrently bridge the same protocol among a separate group of interfaces in a different bridge group on the router.

The router does not switch the protocol between the two bridge groups. Instead, it confines routed traffic to the routed interfaces and bridged traffic to the bridged interfaces. As a result, a protocol can be either routed or bridged on a particular interface, but cannot be both routed and bridged on the same interface.

By default, CRB is disabled for all bridge groups on the router. When you use the **bridge crb** command to enable CRB, it takes effect for all bridge groups currently configured on your router; you cannot enable CRB for some bridge groups on the router but not for others.

When you first enable CRB, the router issues an implicit **bridge route** command for any IP, MPLS, or PPPoE interfaces that are currently configured in the interface stack for the bridge group. This command directs the bridge group to route traffic for these protocols. After CRB has been enabled, you must issue an explicit **bridge route** command to route any new IP, MPLS, or PPPoE interface that is the first occurrence of this protocol in the bridge group. (See “Configuring Explicit Routing” on page 491 for details about using the **bridge route** command.)

As a result, it is important that you issue the **bridge crb** command after you configure all bridge group interfaces. In this way, the router can detect all IP, MPLS, or PPPoE interfaces in your configuration and direct the bridge group to route traffic from these protocols.

Transparent Bridging and VPLS

Except for the **bridge crb** and **bridge route** commands, you can use the existing transparent bridging commands to configure one or more instances of the Virtual Private LAN Service (VPLS), referred to as *VPLS instances*, on the router. VPLS employs a layer 2 virtual private network (VPN) to connect multiple individual LANs across a service provider's MPLS core network. The geographically dispersed multiple LANs functions as a single virtual LAN.

A single VPLS instance is analogous to a bridge group, and performs similar functions. In effect, a VPLS instance is a new or existing bridge group that has additional VPLS attributes configured.

For details about configuring and using VPLS, see *Configuring VPLS* in *JunosE BGP and MPLS Configuration Guide*.

Unsupported Features

The current E Series implementation of transparent bridging does not support the spanning-tree algorithm as defined in IEEE 802.1D.

NOTE: **Because the spanning-tree algorithm is not currently supported, make sure that your topology avoids the creation of network loops.**

Platform Considerations

You can configure transparent bridging on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support transparent bridging on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.

- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support transparent bridging.

For information about the modules that support transparent bridging on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support transparent bridging.

Interface Specifiers

The configuration task examples in this chapter use the *slot/port*[.*subinterface*] format to specify the physical interface on which to configure transparent bridging. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

host1(config)#**interface atm 0/1.10**

For E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

host1(config)#**interface atm 5/0/0.20**

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

References

For more information about transparent bridging, consult the following resources:

- IEEE 802.1D—Media access control (MAC) bridges
- Draft Standard P802.1Q/D9 IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks
- RFC 1493—Definitions of Managed Objects for Bridges (July 1993)

Before You Configure Transparent Bridging

Before you configure transparent bridging on an E Series router, verify that:

- You have correctly installed a line module that supports transparent bridging. For a list of the line modules that support transparent bridging, see *ERX Module Guide,*

Appendix A, Module Protocol Support or *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support transparent bridging.

- Each configured line can transmit data to and receive data from your switch connections.

Table 48 on page 480 lists the prerequisite tasks for configuring transparent bridging and the resources that you can consult to learn how to perform these tasks.

Table 48: Prerequisite Tasks for Configuring Transparent Bridging

To Learn About	See
Preconfiguration and hardware diagnostic procedures	*ERX Hardware Guide* *E120 and E320 Hardware Guide*
Configuring T3 ATM line modules	*Configuring T3 and E3 Interfaces* in *JunosE Physical Layer Configuration Guide*
Configuring OCx/STMx ATM line modules	*Configuring Unchannelized OCx/STMx Interfaces* in *JunosE Physical Layer Configuration Guide*
Configuring Ethernet line modules	*Configuring Ethernet Interfaces* in *JunosE Physical Layer Configuration Guide*

Also have the following information available:

- A diagram of your network topology indicating the names of the bridge groups and bridge group interfaces that you need to create
- On ERX7xx models, ERX14xx models, and ERX310 router, the slot and port numbers of the line modules over which you want to configure transparent bridging
- On E120 and E320 routers, slot, adapter, and port numbers of the IOAs over which you want to configure transparent bridging
- Types and specifiers for the interfaces and subinterfaces over which you want to create bridge group interfaces

Configuration Tasks

To configure transparent bridging on an E Series router:

1. Create a bridge group.
2. (Optional) Set optional attributes for the bridge group.
3. Configure bridge group interfaces.
4. (Optional) Configure nondefault subscriber policies for bridge interfaces.
5. (Optional) Enable concurrent routing and bridging.
6. (Optional) If CRB is enabled, configure explicit routing for IP, MPLS, or PPPoE protocols.

The following sections describe how to perform each of these tasks. See "Configuration Examples" on page 493 for detailed sample configurations.

NOTE: For information about the maximum values that the router supports for transparent bridging, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

Creating Bridge Groups

To create a bridge group:

1. From Global Configuration mode, create a bridge group and give it an alphanumeric name.

   host1(config)#**bridge westford01**

   NOTE: Do not assign the bridge group the same name as an existing VR configured on your router.

2. (Optional) Repeat Step 1 to create additional bridge groups, one at a time.

   host1(config)#**bridge westford02**
   host1(config)#**bridge westford03**

3. (Optional) Use the appropriate **show** command to verify the bridge group creation.

   host1#**show bridge groups**

bridge

- Use to create a bridge group for transparent bridging.
- You must specify an alphanumeric name for the bridge group; the name can be a maximum of 32 characters and can use any combination of alphanumeric characters.
- Example

  host1(config)#**bridge westford04**

- Use the **no** version to remove the bridge group from the router.
- See *bridge*.

Configuring Optional Bridge Group Attributes

After you create a bridge group, you can configure the following optional attributes for the bridge group to manage the MAC address entries in the bridge group's forwarding table:

- Enable or disable the bridge group's ability to acquire dynamically learned MAC addresses; acquiring dynamic MAC addresses is enabled by default.

  host1(config)#**bridge westford01 acquire**

- Enable or disable the bridge group's ability to filter (forward or discard) frames with a particular MAC source or destination address.

  host1(config)#**bridge westford01 address 0090.1a40.4c7c forward atm 3/0.1**
  host1(config)#**bridge westford02 address 1011.22c2.333d discard**

- Set the aging time of a dynamic (learned) entry in the forwarding table.

  host1(config)#**bridge westford01 aging-time 200**

- Set the maximum number of dynamic MAC addresses that a bridge group can learn.

  host1(config)#**bridge westford02 learn 10000**

You can also optionally enable SNMP link status processing for the bridge group. For example:

host1(config)#**bridge westford03 snmp-trap link-status**

bridge acquire

- Use to enable or disable a specified bridge group's ability to acquire dynamically learned MAC addresses; acquiring dynamic MAC addresses is enabled by default.
- Enables the bridge group to forward any frames it receives for nodes (stations) whose address it has learned dynamically.
- Example

  host1(config)#**bridge westford01 acquire**

- Use the **no** version to prevent the bridge group from acquiring dynamically learned MAC addresses and to limit forwarding only to those nodes that have a statically configured address entry in the forwarding table.
- See *bridge acquire*.

bridge address

- Use to enable or disable a specified bridge group's ability to filter (forward or discard) frames based on their MAC address.
- Enables the bridge group to filter frames by their MAC address and add static (nonlearned) address entries to the forwarding table.
- Specify the following:
  - *bridgeGroupName*—Alphanumeric name of the bridge group specified in the **bridge** command
  - *macAddress*—Unique 48-bit (6-byte) physical address or hardware address of the LAN network interface card as a dotted triple of four-digit hexadecimal numbers
- Specify one of the following filter types:
  - **forward**—Forwards frames destined for the specified MAC address out the specified interface
  - **discard**—Discards (drops) frames sent from or destined for the specified MAC address without further processing

- If you use the **forward** keyword, you must additionally specify the following:
  - *interfaceType*—One of the following bridge interface types listed in *Interface Types and Specifiers* in *JunosE Command Reference Guide*:
    - **atm**
    - **fastEthernet**
    - **gigabitEthernet**
    - **tenGigabitEthernet**
  - *interfaceSpecifier*—Particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *JunosE Command Reference Guide* for information
- Example 1—Forwards frames destined for the node with MAC address 0090.1a40.4c7c out the specified Fast Ethernet interface

  host1(config)#**bridge westford02 address 0090.1a40.4c7c forward fastEthernet 3/0.1**
- Example 2—Drops frames sent from or destined for the node with MAC address 1011.22b2.333c

  host1(config)#**bridge westford03 address 1011.22b2.333c discard**
- Use the **no** version to remove the static MAC address entry from the forwarding table.
- See *bridge address*.

bridge aging-time

- Use to set the length of time, in seconds, that a dynamic (learned) MAC address entry can remain in a specified bridge group's forwarding table.
- When a dynamic entry reaches its configured aging time, it "ages out" of the forwarding table.
- The default aging time is 300 seconds.
- The aging-time range is 1–1000000 seconds.
- Example

  host1(config)#**bridge westford04 aging-time 1000**
- Use the **no** version to restore the default value, 300 seconds.
- See *bridge aging-time*.

bridge learn

- Use to set the maximum number of dynamic (learned) MAC address entries that a specified bridge group can learn.
- For information about the maximum number of learned MAC address entries combined for all bridge groups on an E Series router, see *JunosE Release Notes, Appendix A, System Maximums.*

- The default value is 0 (zero) learned addresses. This default implies that there is no maximum number of learned entries for an individual bridge group; that is, an individual bridge group can learn an unlimited number of MAC addresses, up to the maximum number that the router supports.
- Example

  host1(config)#**bridge westford05 learn 2000**
- Use the **no** version to restore the default value, 0 (zero) learned addresses.
- See *bridge learn*.

bridge snmp-trap link-status

- Use to enable SNMP link status processing for a specified bridge group and to enable SNMP traps for all bridge interfaces configured in the bridge group.
- Example

  host1(config)#**bridge westford06 snmp-trap link-status**
- Use the **no** version to disable SNMP link status processing for the bridge group.
- See *bridge snmp-trap link-status*.

Configuring Bridge Group Interfaces

To configure a bridge group interface:

1. From Global Configuration mode, select the ATM, Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface or subinterface that you want to assign to the bridge group.
2. Assign the interface or subinterface to an existing bridge group to create the bridge interface.
3. (Optional) Configure the bridge group interface as a trunk (server) interface.
4. (Optional) Enable SNMP link status processing for the bridge group interface.
5. (Optional) Set the maximum number of dynamic MAC addresses that the bridge group interface can learn.

For detailed sample configurations that include bridge interfaces, see “Configuration Examples” on page 493.

bridge-group

- Use to assign a bridge interface to an existing bridge group.
- To create a subscriber (client) bridge group interface, which is the default, you must supply the alphanumeric name of the bridge group (specified in the **bridge** command) to which you want to assign the interface.
- Optionally, you can also choose one of the following keywords:
  - **subscriber-trunk**—Creates a trunk (server) bridge group interface

- **snmp-trap link-status**—Enables SNMP link status processing for the specified interface in the specified bridge group; SNMP link status processing is disabled by default
- **learn** *addressCount*—Sets the maximum number of MAC addresses that the bridge group interface can learn, where *addressCount* is an integer in the range 0–64000. A value of 0 indicates that an individual bridge group interface can learn an unlimited number of MAC addresses, up to the maximum number that the router supports.

- Example 1—Creates a subscriber (client) bridge group interface for a bridge group named westford02 with SNMP link status processing enabled

  host1(config-subif)#**bridge-group westford02 snmp-trap link-status**

- Example 2—Sets the maximum number of learned MAC addresses on the westford02 bridge interface to 1000

  host1(config-subif)#**bridge-group westford02 learn 1000**

- Example 3—Creates a trunk (server) interface for a bridge group named westford03

  host1(config-subif)#**bridge-group westford03 subscriber-trunk**

- Use the **no** version to remove the interface from the bridge group and to restore the default value for the keyword you specified.
- See *bridge-group*.

interface atm

- Use to select an ATM interface or subinterface type.
- Example

  host1(config)#**interface atm 3/2.1**

- Use the **no** version to remove the interface or subinterface.
- See *interface atm*.

interface fastEthernet

- Use to select a Fast Ethernet interface.
- Example

  host1(config)#**interface fastEthernet 1/0.2**

- Use the **no** version to remove the interface or subinterface. You must issue the no version from the highest level down; you cannot remove an interface or a subinterface if the one above it still exists.
- See *interface fastEthernet*.

interface gigabitEthernet

interface tenGigabitEthernet

- Use to select a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface.
- Examples

```
host1(config)#interface gigabitEthernet 1/0
host1(config)#interface gigabitEthernet 4/0/1
host1(config)#interface tenGigabitEthernet 4/0/1
```

- Use the **no** version to remove the interface or subinterface. You must issue the no version from the highest level down; you cannot remove an interface or subinterface if the one above it still exists.
- See *interface gigabitEthernet*.
- See *interface tenGigabitEthernet*.

Configuring Subscriber Policies

To configure a nondefault client subscriber policy:

1. From Global Configuration mode, create the subscriber policy and assign it an alphanumeric name.

   ```
   host1(config)#subscriber-policy client01
   ```

   This command accesses Subscriber Policy Configuration mode.
2. From Subscriber Policy Configuration mode, define the rules for each packet or attribute type for which you want to change the default value. (All other packet or attribute types will continue to use the default values listed in Table 47 on page 476.)

   ```
   host1(config-policy)#broadcast permit
   host1(config-policy)#multicast deny
   host1(config-policy)#relearn deny
   ```
3. Exit Subscriber Policy Configuration mode.

   ```
   host1(config-policy)#exit
   ```
4. From Global Configuration mode, associate the new subscriber policy with the bridge group in which the subscriber (client) interface resides.

   ```
   host1(config)#bridge westford02 subscriber-policy client01
   ```
5. (Optional) Use the appropriate **show** commands to verify the creation of the subscriber policy and its association with the bridge group interface.

   ```
   host1#show subscriber-policy client01
   host1#show bridge westford02
   ```

arp

- Use to modify the subscriber policy for ARP to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) ARP packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- ARP packets are forwarded by default.
- Example

host1(config-policy)#**arp deny**

- Use the **no** version to restore the default value.
- See *arp*.

bridge subscriber-policy

- Use to associate a subscriber (client) bridge interface with a nondefault subscriber policy.
- Specify the following:
  - *bridgeGroupName*—Alphanumeric name of the bridge group specified in the **bridge** command
  - *subscriberPolicyName*—Alphanumeric name of the subscriber policy specified in the **subscriber-policy** command
- Example

  host1(config)#**bridge westford02 subscriber-policy client01**
- Use the **no** version to remove the association with the subscriber policy.

NOTE: You cannot change the default subscriber policy values for a trunk (server) bridge interface. As a result, you cannot use the **bridge subscriber-policy** command to associate a nondefault subscriber policy with a trunk bridge interface.

- See *bridge subscriber-policy*.

broadcast

- Use to modify the subscriber policy for the broadcast protocol to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) broadcast packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- Broadcast packets are filtered or dropped by default.
- Example

  host1(config-policy)#**broadcast permit**
- Use the **no** version to restore the default value.
- See *broadcast*.

ip

- Use to modify the subscriber policy for IP to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) IP packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- IP packets are forwarded by default.
- Example

  host1(config-policy)#**ip deny**
- Use the **no** version to restore the default value.
- See *ip*.

mpls

- Use to modify the subscriber policy for MPLS to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) MPLS packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- MPLS packets are forwarded by default.
- Example

  host1(config-policy)#**mpls deny**
- Use the **no** version to restore the default value.
- See *mpls*.

multicast

- Use to modify the subscriber policy for the multicast protocol to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) multicast packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- Multicast packets are forwarded by default.
- Example

  host1(config-policy)#**multicast deny**
- Use the **no** version to restore the default value.
- See *multicast*.

pppoe

- Use to modify the subscriber policy for PPPoE to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) PPPoE packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- PPPoE packets are forwarded by default.
- Example

  host1(config-policy)#**pppoe deny**
- Use the **no** version to restore the default value.
- See *pppoe*.

relearn

- Use to modify the relearning policy for a subscriber (client) bridge interface.
- The **relearn** command defines whether the bridge interface can relearn a MAC address entry on a different interface from the one initially associated with this entry in the forwarding table.
- Specify one of the following keywords:
  - **permit**—Enables relearning
  - **deny**—Prohibits relearning and forces the bridge interface to wait until an entry "ages out" of the forwarding table to relearn it on the new interface
- Relearning is enabled by default.
- Example

  host1(config-policy)#**relearn deny**
- Use the **no** version to restore the default value.
- See *relearn*.

subscriber-policy

- Use to create a nondefault subscriber policy for a subscriber (client) bridge interface.
- A subscriber policy is a set of forwarding and filtering rules that defines how the bridge interface handles various packet types.
- You must specify an alphanumeric name for the subscriber policy; the name can be a maximum of 32 characters and can use any combination of alphanumeric characters.
- Example

  host1(config)#**subscriber-policy client01**
- Use the **no** version to remove the nondefault subscriber policy.

NOTE: You cannot change the default subscriber policy values for a trunk (server) bridge interface. As a result, you cannot use the **subscriber-policy** command to create a nondefault subscriber policy for a trunk interface.

- See *subscriber-policy*.

unicast

- Use to modify the subscriber policy for the unicast (user-to-user) protocol to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) unicast packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- Unicast packets are forwarded by default.
- Example

  host1(config-policy)#**unicast deny**
- Use the **no** version to restore the default value.
- See *unicast*.

unknown-destination

- Use to modify the subscriber policy for packets with unknown unicast DAs to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) packets with unknown unicast DAs.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- Packets with unknown unicast DAs are filtered or dropped by default.
- Example

  host1(config-policy)#**unknown-destination permit**
- Use the **no** version to restore the default value.
- See *unknown-destination*.

unknown-protocol

- Use to modify the subscriber policy for packets containing an unknown protocol to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) these packets.
- An unknown protocol is any protocol other than ARP, IP, MPLS, or PPPoE.

- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- Packets containing an unknown protocol are forwarded by default.
- Example

  host1(config-policy)#**unknown-protocol deny**
- Use the **no** version to restore the default value.
- See *unknown-protocol*.

Enabling Concurrent Routing and Bridging

To enable concurrent routing and bridging (CRB) for all bridge groups on the router:

1. From Global Configuration mode, issue the **bridge crb** command.

   host1(config)#**bridge crb**
2. (Optional) Use the appropriate **show** command to verify that CRB is enabled for the bridge groups on your router.

   host1#**show bridge groups details**

bridge crb

- Use to enable concurrent routing and bridging (CRB) for all bridge groups configured on an E Series router.
- CRB is disabled by default.
- When CRB is enabled, the router can route a protocol among a group of interfaces in one bridge group and concurrently bridge the same protocol among a separate group of interfaces in a different bridge group.
- The command takes effect for all bridge groups on an E Series router; you cannot enable CRB for some bridge groups on the router but not for others.
- Example

  host1(config)#**bridge crb**
- Use the **no** version to disable CRB on all bridge groups and restore the default bridging capability.
- See *bridge crb*.

Configuring Explicit Routing

After you enable concurrent routing and bridging, you may need to issue the **bridge route** command to configure explicit routing for IP, MPLS, or PPPoE protocols if both of the following conditions are true:

- You configure new IP, MPLS, or PPPoE interfaces after you issue the **bridge crb** command to enable concurrent routing and bridging.
- The IP, MPLS, or PPPoE interface is the first occurrence of this protocol in the bridge group.

For example, assume that you want to route (rather than bridge) IP, MPLS, and PPPoE interfaces, but only IP and MPLS interfaces are configured when you issue the **bridge crb** command. The router detects the IP and MPLS interfaces and issues implicit **bridge route** commands to route these protocols.

If you subsequently add a new IP interface to a bridge group, you do not need to issue the **bridge route** command because the implicit **bridge route** command for IP is still in effect. However, if you subsequently add a new PPPoE interface to the bridge group, you must issue an explicit **bridge route** command for PPPoE to direct the bridge group to route PPPoE packets.

You can also use the **bridge route** command as a way to filter packets by routing. If you issue an explicit **bridge route** command for a protocol that is not currently configured in any of your bridge groups, the bridge group must route rather than bridge that protocol, but does not have the required interface stacking to do so. As a result, the bridge group discards (drops) those packets.

To configure explicit routing:

1. Ensure that you have enabled concurrent routing and bridging. (See "Enabling Concurrent Routing and Bridging" on page 491 for details.)
2. From Global Configuration mode, enable routing of IP, MPLS, or PPPoE packets in a specified bridge group.

   host1(config)#**bridge westford02 route ip**
   host1(config)#**bridge westford02 route mpls**
   host1(config)#**bridge westford03 route pppoe**
3. (Optional) Use the appropriate **show** command to verify that routing is enabled for the specified protocols in the bridge group.

   host1#**show bridge westford02**

bridge route

- Use to enable the routing of IP, MPLS, or PPPoE packets in a specified bridge group when concurrent routing and bridging (CRB) is enabled.
- If you issue this command for a protocol that is not configured in any bridge groups on your router, the bridge group discards (drops) those packets.
- You must specify the alphanumeric name of the bridge group specified in the **bridge** command.
- Choose one of the following keywords to indicate the protocol type that the bridge group routes: **ip**, **mpls**, or **pppoe**.
- Example

  host1(config)#**bridge westford02 route ip**

- Use the **no** version to disable routing of the specified protocol in the specified bridge group.
- See *bridge route*.

Configuration Examples

This section provides examples that show how to configure transparent bridging on the router. With each step, an illustration shows how the router is building the interface column.

Example 1: Bridging with Bridged Ethernet

The following example illustrates how to configure transparent bridging with bridged Ethernet.

1. Create the bridge group.

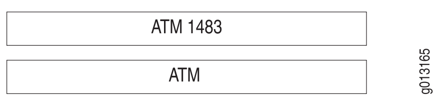
   host1(config)#**bridge westford01**

2. Create an ATM major interface.

   host1(config)#**interface atm 3/3**

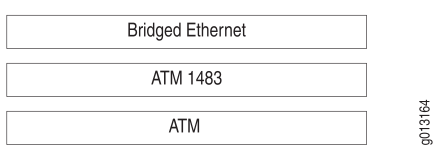
3. Create an ATM 1483 subinterface and associated PVC.

   host1(config-if)#**interface atm 3/3.1**
   host1(config-subif)#**atm pvc 1 0 10 aal5snap**



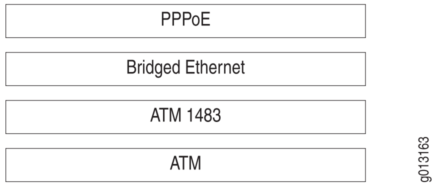
4. Specify bridged Ethernet as the encapsulation method on the subinterface. The **encapsulation** keyword implies that the bridged Ethernet interface is the only interface stacked directly above the ATM 1483 subinterface. As a result, the bridged Ethernet interface cannot have a peer interface stacked above the same lower-layer interface.

   host1(config-subif)#**encapsulation bridge1483**



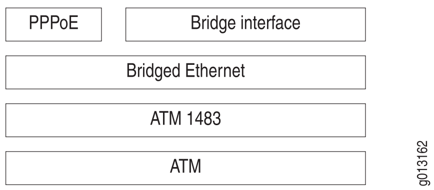
5. Create a PPPoE major interface over the bridged Ethernet interface. Because this command does not use the **encapsulation** keyword, the PPPoE interface can have one or more peer interfaces stacked above the same bridged Ethernet interface.

   host1(config-subif)#**pppoe**



6. Configure a subscriber (client) bridge group interface over the bridged Ethernet interface as a peer to the PPPoE interface. Assign the interface to the bridge group you created in Step 1.

   host1(config-subif)#**bridge-group westford01**



Example 2: Bridging with VLANs

The following example illustrates how to configure transparent bridging with VLANs over a Fast Ethernet interface.

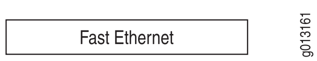
NOTE: You can also configure transparent bridging with VLANs over a bridged Ethernet interface. For information, see "Configuring VLANs over Bridged Ethernet" on page 460 in "Configuring Bridged Ethernet" on page 451.

1. Create the bridge group.

   host1(config)#**bridge westford02**

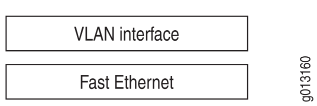
2. Create a Fast Ethernet interface.

   host1(config)#**interface fastEthernet 2/0**



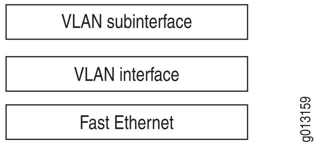
3. Create a VLAN major interface by specifying VLAN as the encapsulation method for the interface.

   host1(config-if)#**encapsulation vlan**



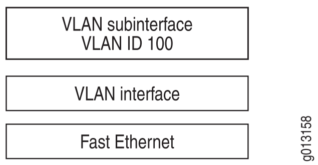
4. Create a VLAN subinterface by adding a subinterface number to the **interface fastEthernet** command.

   host1(config-if)#**interface fastEthernet 2/0.1**



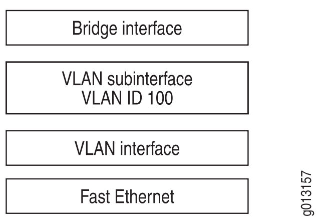
5. Assign a unique VLAN ID to the VLAN subinterface.

   host1(config-if)#**vlan id 100**



6. Configure a subscriber (client) bridge group interface over the VLAN subinterface. Assign the interface to the bridge group you created in Step 1.

   host1(config-subif)#**bridge-group westford02**



7. Exit Subinterface Configuration mode.

   host1(config-subif)#**exit**

8. (Optional) Configure additional VLAN subinterfaces and bridge group interfaces by repeating Steps 4 through 6, supplying unique values.

Monitoring Transparent Bridging

This section describes how to:

- Set a statistics baseline for bridge groups and bridge interfaces.
- Remove all dynamic MAC address entries or a specific dynamic MAC address entry from the forwarding table for bridge groups and bridge interfaces.
- Use the **show** commands to monitor bridge groups, bridge group interfaces, and subscriber policies

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

Setting Statistics Baselines

You can set a statistics baseline for a bridge group (by using the **baseline bridge** command) or for a bridge interface (by using the **baseline bridge interface** command).

baseline bridge

- Use to set a statistics baseline for a specified bridge group.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Example

  host1#**baseline bridge westford03**
- There is no **no** version.
- See *baseline bridge*.

baseline bridge interface

- Use to set a statistics baseline for a particular network interface belonging to a bridge group.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- You must specify the following:
  - *interfaceType*—One of the following bridge interface types listed in *Interface Types and Specifiers* in *JunosE Command Reference Guide*:
    - **atm**
    - **fastEthernet**
    - **gigabitEthernet**
    - **tenGigabitEthernet**
  - *interfaceSpecifier*—Particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *JunosE Command Reference Guide* for information
- Example

  host1#**baseline bridge interface atm 3/3.1**
- There is no **no** version.
- See *baseline bridge interface*.

Removing Dynamic MAC Address Entries

You can remove all dynamic (learned) MAC address entries from the forwarding table for a bridge group (using the **clear bridge** command) or for a bridge interface (using the

clear bridge interface command). You can also use the **clear bridge address** command to remove a specific dynamic MAC address entry from the forwarding table for a bridge group.

clear bridge

- Use to remove all dynamic MAC address entries from the forwarding table for the specified bridge group.
- Example

```
host1#show bridge westford01 table
  Bridge: westford01 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   -------------------   -----
0090.1a01.0205   forward      ATM3/3.1                0
1234.abcd.5678   discard      ---                    ---

host1#clear bridge westford01

host1#show bridge westford01 table
  Bridge: westford01 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   -------------------   -----
```

- There is no **no** version.
- See *clear bridge*.

clear bridge address

- Use to remove a specific dynamic MAC address entry from the forwarding table for the specified bridge group.
- Example

```
host1#show bridge westford01 table
  Bridge: westford01 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   -------------------   -----
0090.1a01.0205   forward      ATM3/3.1                0
1234.abcd.5678   discard      ---                    ---

host1#clear bridge westford01 address 1234.abcd.5678

host1#show bridge westford01 table
  Bridge: westford01 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   -------------------   -----
0090.1a01.0205   forward      ATM3/3.1                0
```

- There is no **no** version.
- See *clear bridge address*.

clear bridge interface

- Use to remove all dynamic MAC address entries for a network interface belonging to a bridge group from the forwarding table for that bridge group.
- You must specify the following:

- *interfaceType*—One of the following bridge interface types listed in *Interface Types and Specifiers* in *JunosE Command Reference Guide*:
  - **atm**
  - **fastEthernet**
  - **gigabitEthernet**
  - **tenGigabitEthernet**
- *interfaceSpecifier*—Particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *JunosE Command Reference Guide* for information

- Example

```
host1#show bridge westford02 table dynamic
  Bridge: westford02 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   --------------------   -----
0090.1a01.0205   forward      ATM3/3.1                 0
0090.1a01.0206   forward      ATM3/3.2                10
0090.1a01.0207   forward      ATM3/3.3                 5
```

```
host1#clear bridge interface atm 3/3.2
```

```
host1#show bridge westford02 table dynamic
  Bridge: westford02 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   --------------------   -----
0090.1a01.0205   forward      ATM3/3.1                 0
0090.1a01.0207   forward      ATM3/3.3                 5
```

- There is no **no** version.
- See *clear bridge interface*.

Monitoring Bridge Groups

You can use **show** commands to display information about the bridge groups configured on your router.

show bridge

- Use to display configuration and statistics information for the specified bridge group.
- To display information about the MAC address table and bridge interfaces, use the **all** keyword.
- Field descriptions
  - BridgeGroup—Name assigned to the bridge group
  - Bridge Mode—Bridging capability currently enabled, either concurrent routing and bridging (CRB) or default bridging
  - Aging Time—Length of time, in seconds, that a MAC address entry can remain in the forwarding table
  - Learning—Whether acquisition of dynamically learned MAC addresses is currently enabled or disabled

- Max Learn—Maximum number of dynamic MAC addresses that the bridge group can learn
- Link Status Snmp Traps—Whether SNMP link status processing is enabled or disabled for all bridge interfaces in the bridge group
- Subscriber Policy—Name of the subscriber policy currently in effect for the bridge group
- Protocol Actions—When CRB is enabled, displays the protocols (IP, MPLS, or PPPoE) for which explicit routing has been configured
- Port Count—Number of ports (interfaces) currently configured for the bridge group; this value typically matches the Interface Count
- Interface Count—Number of bridge group interfaces currently configured for the bridge group
- Address Table—Displays the current static and dynamic entries in the MAC address table
  - Address—MAC address of the entry
  - Action—How the bridge group handles this entry: forward or discard
  - Interface—Interface type and specifier on which the entry will be forwarded; this value does not appear for entries that are discarded
  - Age—Length of time that a dynamic entry has been in the forwarding table; this value does not appear for static entries
- Interfaces—Displays statistics information for each bridge group interface; the entries for each interface are preceded by the interface type and specifier (for example, ATM3/3.1)
  - Port Number—Bridge group port number on which this interface resides
  - Operational Status—Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent
  - Admin Status—State of the physical interface: Up, Down
  - Snmp Link Status Trap—Whether SNMP link status processing is enabled or disabled for the specified bridge interface
  - Max Learn—Maximum number of dynamic MAC addresses that the bridge group interface can learn
  - Subscriber Policy—Name of the subscriber policy currently in effect for the bridge group interface
  - In Octets—Number of octets received on this interface
  - In Frames—Number of frames received on this interface
  - In Discards—Number of incoming packets discarded on this interface
  - In Errors—Number of incoming errors received on this interface
  - Out Octets—Number of octets transmitted on this interface

    - Out Frames—Number of frames transmitted on this interface
    - Out Discards—Number of outgoing packets discarded on this interface
    - Out Errors—Number of outgoing errors on this interface
    - queue—Hardware packet queue associated with the specified traffic class and interface
    - Queue length—Length of the queue, in bytes
    - Forwarded packets, Bytes—Number of packets and bytes forwarded on this queue
    - Dropped committed packets, Bytes—Number of committed packets and bytes that were dropped
    - Dropped conformed packets, Bytes—Number of conformed packets and bytes that were dropped
    - Dropped exceeded packets, Bytes—Number of exceeded packets and bytes that were dropped

- Example 1—Displays configuration settings for the specified bridge group

```
host1#show bridge westford01
  BridgeGroup: westford01
        Bridge Mode:            CRB
        Aging Time:             300 secs
        Learning:               Enabled
        Max Learn:              Unlimited
        Link Status Snmp Traps: Disabled
        Subscriber Policy:      default Subscriber
        Protocol Actions:
           Route  IP
           Route  PPPoE
        Port Count:             1
        Interface Count:        1
```

- Example 2—Displays information about configuration settings, MAC address table entries, and bridge group interfaces for the specified bridge group

```
host1#show bridge westford01 all
  BridgeGroup: westford01
        Bridge Mode:            CRB
        Aging Time:             300 secs
        Learning:               Enabled
        Max Learn:              Unlimited
        Link Status Snmp Traps: Disabled
        Subscriber Policy:      default Subscriber
        Protocol Actions:
           Route  IP
           Route  PPPoE
        Port Count:             1
        Interface Count:        1

  Address Table:
   Address         Action        Interface          Age
--------------   ----------   -------------------   -----
1011.22b2.333c   forward      ATM3/3.1               ---
1234.abcd.5678   discard      ---                    ---
```



```
Interfaces:
  ATM3/3.1
    Port Number: 1
    Operational Status: LowerLayerDown
    Admin Status: Up
    Snmp Link Status Trap: Disabled
    Max Learn: Unlimited
    Subscriber Policy: default Subscriber
    Statistics:
       In Octets:    0
       In Frames:    0
       In Discards:  0
       In Errors:    0
       Out Octets:   0
       Out Frames:   0
       Out Discards: 0
       Out Errors:   0
    queue 0: traffic class best-effort, bound to bridge     ATM3/3.1

      Queue length 0 Bytes
      Forwarded packets 0, Bytes 0
      Dropped committed packets 0, Bytes 0
      Dropped conformed packets 0, Bytes 0
      Dropped exceeded packets 0, Bytes 0
```

- See *show bridge*.

show bridge groups

- Use to display configuration information for all bridge groups currently configured on your router.
- To display the configuration settings for all bridge groups on your router, use the **details** keyword.
- Field descriptions
  - BridgeGroup—Name assigned to the bridge group
  - Bridge Mode—Bridging capability currently enabled, either concurrent routing and bridging (CRB) or default bridging
  - Aging Time—Length of time, in seconds, that a MAC address entry can remain in the forwarding table
  - Learning—Whether acquisition of dynamically learned MAC addresses is currently enabled or disabled
  - Max Learn—Maximum number of dynamic MAC addresses that the bridge group can learn
  - Link Status Snmp Traps—Whether SNMP link status processing is enabled or disabled for all bridge interfaces in the bridge group
  - Subscriber Policy—Name of the subscriber policy currently in effect for the bridge group
  - Protocol Actions—When CRB is enabled, displays the protocols (IP, MPLS, or PPPoE) for which explicit routing has been configured

- Port Count—Number of ports (interfaces) currently configured for the bridge group; this value typically matches the Interface Count
- Interface Count—Number of bridge group interfaces currently configured for the bridge group

- Example 1—Displays the names of the bridge groups configured on your router

```
host1#show bridge groups
  BridgeGroup: westford02
  BridgeGroup: westford01
```

- Example 2—Displays the configuration settings for each bridge group on your router

```
host1#show bridge groups details
  BridgeGroup: westford02
        Bridge Mode:            CRB
        Aging Time:             300 secs
        Learning:               Enabled
        Max Learn:              Unlimited
        Link Status Snmp Traps: Disabled
        Subscriber Policy:      client01
        Protocol Actions:
           Route  IP
           Route  PPPoE
        Port Count:             0
        Interface Count:        0
  BridgeGroup: westford01
        Bridge Mode:            CRB
        Aging Time:             300 secs
        Learning:               Enabled
        Max Learn:              Unlimited
        Link Status Snmp Traps: Disabled
        Subscriber Policy:      default Subscriber
        Protocol Actions:
        Port Count:             1
        Interface Count:        1
```

- See *show bridge groups*.

show bridge port

- Use to display configuration, statistics, and status information for all ports (interfaces) or for a specified port associated with a bridge group.
- To display only the port number, interface identifier, and status for each port, use the **brief** keyword.
- Field descriptions
  - Port Number—Bridge group port number on which this interface resides
  - Operational Status—Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent
  - Admin Status—State of the physical interface: Up, Down
  - Snmp Link Status Trap—Whether SNMP link status processing is enabled or disabled for the specified bridge interface

- Max Learn—Maximum number of dynamic MAC addresses that the bridge group interface can learn
- Subscriber Policy—Name of the subscriber policy currently in effect for the bridge group interface
- Statistics—Displays statistics information for the specified port
  - In Octets—Number of octets received on this interface
  - In Frames—Number of frames received on this interface
  - In Discards—Number of incoming packets discarded on this interface
  - In Errors—Number of incoming errors received on this interface
  - Out Octets—Number of octets transmitted on this interface
  - Out Frames—Number of frames transmitted on this interface
  - Out Discards—Number of outgoing packets discarded on this interface
  - Out Errors—Number of outgoing errors on this interface
- queue—Hardware packet queue associated with the specified traffic class and interface
  - Queue length—Length of the queue, in bytes
  - Forwarded packets, Bytes—Number of packets and bytes forwarded on this queue
  - Dropped committed packets, Bytes—Number of committed packets and bytes that were dropped
  - Dropped conformed packets, Bytes—Number of conformed packets and bytes that were dropped
  - Dropped exceeded packets, Bytes—Number of exceeded packets and bytes that were dropped
- Using the **brief** keyword displays the following fields:
  - Port—Bridge group port number on which this interface resides
  - Interface—Interface type and specifier associated with the port (for example, ATM3/3.1)
  - Status—Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent

- Example 1—Displays configuration, statistics, and status information for all ports currently associated with the bridge group

```
host1#show bridge westford01 port 1
    ATM3/3.1
      Port Number: 1
      Operational Status: LowerLayerDown
      Admin Status: Up
      Snmp Link Status Trap: Disabled
      Max Learn: Unlimited
      Subscriber Policy: default Subscriber
```



```
Statistics:
   In Octets:    0
   In Frames:    0
   In Discards:  0
   In Errors:    0
   Out Octets:   0
   Out Frames:   0
   Out Discards: 0
   Out Errors:   0
queue 0: traffic class best-effort, bound to bridge     ATM3/3.1

  Queue length 0 Bytes
  Forwarded packets 0, Bytes 0
  Dropped committed packets 0, Bytes 0
  Dropped conformed packets 0, Bytes 0
  Dropped exceeded packets 0, Bytes 0
```

- Example 2—Uses the **brief** keyword to display summary information for each port

  host1#**show bridge westford01 port brief**
  ```
   Port            Interface               Status
  ------   -------------------------   ------------
  1        ATM3/3.1                    LowerLayerDown
  ```

- See *show bridge port*.

show bridge table

- Use to display information about dynamic and static entries in the MAC address table for the specified bridge group.
- To display only static address entries, use the **static** keyword.
- To display only dynamic address entries, use the **dynamic** keyword.
- Field descriptions
  - Bridge—Name of the bridge group for which the MAC address table is displayed
  - Address—MAC address of the entry
  - Action—Specifies how the bridge group handles this entry: forward or discard
  - Interface—Interface type and specifier on which the entry will be forwarded; this value does not appear for entries that are discarded
  - Age—Length of time that a dynamic entry has been in the forwarding table; this value does not appear for static entries
- Example

  host1#**show bridge westford01 table static**
  ```
    Bridge: westford01 MAC Address Table
     Address          Action         Interface          Age
  --------------   ----------   -------------------   -----
  1a11.22b2.333c   forward      ATM3/3.1               ---
  1234.abcd.5678   discard      ---                    ---
  ```

- See *show bridge table*.

Monitoring Bridge Interfaces

You can use the **show bridge interface** command to display information for a specified bridge interface or for all interfaces assigned to a bridge group.

show bridge interface

- Use to display configuration, statistics, and status information for a specified bridge interface or for all interfaces assigned to a bridge group.
- Field descriptions
  - BridgeGroup—Name of the bridge group to which the interface belongs
  - Port Number—Bridge group port number on which this interface resides
  - Operational Status—Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent
  - Admin Status—State of the physical interface: Up, Down
  - Snmp Link Status Trap—Whether SNMP link status processing is enabled or disabled for the specified bridge interface
  - Max Learn—Maximum number of dynamic MAC addresses that the bridge group interface can learn
  - Subscriber Policy—Name of the subscriber policy currently in effect for the bridge group interface
  - Statistics—Displays statistics information for the specified port
    - In Octets—Number of octets received on this interface
    - In Frames—Number of frames received on this interface
    - In Discards—Number of incoming packets discarded on this interface
    - In Errors—Number of incoming errors received on this interface
    - Out Octets—Number of octets transmitted on this interface
    - Out Frames—Number of frames transmitted on this interface
    - Out Discards—Number of outgoing packets discarded on this interface
    - Out Errors—Number of outgoing errors on this interface
  - queue—Hardware packet queue associated with the specified traffic class and interface
    - Queue length—Length of the queue, in bytes
    - Forwarded packets, Bytes—Number of packets and bytes forwarded on this queue
    - Dropped committed packets, Bytes—Number of committed packets and bytes that were dropped

    - Dropped conformed packets, Bytes—Number of conformed packets and bytes that were dropped
    - Dropped exceeded packets, Bytes—Number of exceeded packets and bytes that were dropped
  - Using the **brief** keyword displays the following fields:
    - Interface—Interface type and specifier associated with the port (for example, FastEthernet9/1.1)
    - Port—Bridge group port number on which this interface resides
    - Status—Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent

- Example 1—Displays information about a specified interface

```
host1#show bridge interface fastEthernet 9/1.1
    fastEthernet9/1.1
      BridgeGroup: 1
      Port Number: 1
      Operational Status: Up
      Admin Status: Up
      Snmp Link Status Trap: Disabled
      Max Learn: Unlimited
      Subscriber Policy: atmfe1
      Statistics:
         In Octets:    0
         In Frames:    0
         In Discards:  0
         In Errors:    0
         Out Octets:   0
         Out Frames:   0
         Out Discards: 0
         Out Errors:   0
     queue 0: traffic class best-effort, bound to bridge     FastEthernet9/1.1

        Queue length 0 Bytes
        Forwarded packets 0, Bytes 0
        Dropped committed packets 0, Bytes 0
        Dropped conformed packets 0, Bytes 0
        Dropped exceeded packets 0, Bytes 0
```

- Example 2—Uses the **brief** keyword to display a summary of all bridge interfaces configured on the router

```
host1#show bridge westford01 interface brief
Interface            Port       Status
-----------------  --------   ------------
FastEthernet9/1.1     1           Up
FastEthernet9/1.2     2           Up
FastEthernet9/3.1     3           Up
ATM11/0.5             4           Up
ATM11/3.2             5           Up
ATM11/0.7             6           Up
```

- See *show bridge interface*.

Monitoring Subscriber Policies

You can use the **show subscriber-policy** command to display the rules for all subscriber policies configured on your router or for a specified subscriber policy.

show subscriber-policy

- Use to display the set of forwarding and filtering rules for all default and nondefault subscriber policies configured on the router or for a specified subscriber policy.
- For all packet types except Relearn, the command displays **permit** to indicate that the bridge interface forwards the packets, or **deny** to indicate that the bridge interface filters the packets. (For information about the meaning of **permit** and **deny** for Relearn, see the field descriptions.)
- Field descriptions
  - Subscriber—Name of the subscriber policy
  - ARP—Specifies how the bridge interface handles ARP packets
  - Broadcast—Specifies how the bridge interface handles broadcast packets
  - Multicast—Specifies how the bridge interface handles multicast packets
  - Unknown Destination—Specifies how the bridge interface handles packets with unknown unicast DAs
  - Unicast—Specifies how the bridge interface handles unicast (user-to-user) packets
  - PPPoE—Specifies how the bridge interface handles PPPoE packets
  - Relearn—Specifies whether the bridge interface can relearn a MAC address entry on a different interface from the one initially associated with this entry in the forwarding table; **permit** indicates that relearning is allowed, and **deny** indicates that relearning is prohibited
  - Mpls—Specifies how the bridge interface handles MPLS packets
- Example 1—Displays the rules for all default and nondefault subscriber policies currently configured on the router

```
host1#show subscriber-policy
  Subscriber: default Subscriber
  ARP                 : Permit
  Broadcast           : Deny
  Multicast           : Permit
  Unknown Destination : Deny
  IP                  : Permit
  Unknown Protocol    : Permit
  Unicast             : Permit
  PPPoE               : Permit
  Relearn             : Permit
  Mpls                : Permit
  Subscriber: default Trunk
  ARP                 : Permit
  Broadcast           : Permit
  Multicast           : Permit
  Unknown Destination : Permit
```

```
IP                  : Permit
Unknown Protocol    : Permit
Unicast             : Permit
PPPoE               : Permit
Relearn             : Permit
Mpls                : Permit

Subscriber: client01
ARP                 : Permit
Broadcast           : Permit
Multicast           : Deny
Unknown Destination : Deny
IP                  : Permit
Unknown Protocol    : Permit
Unicast             : Permit
PPPoE               : Permit
Relearn             : Deny
Mpls                : Permit
```

- Example 2—Displays the rules for a specified subscriber policy

```
host1#show subscriber-policy client01
  Subscriber: client01
  ARP                 : Permit
  Broadcast           : Permit
  Multicast           : Deny
  Unknown Destination : Deny
  IP                  : Permit
  Unknown Protocol    : Permit
  Unicast             : Permit
  PPPoE               : Permit
  Relearn             : Deny
  Mpls                : Permit
```

- See *show subscriber-policy*.

CHAPTER 18

Configuring Cisco HDLC

Cisco High-Level Data Link Control (HDLC) is an encapsulation protocol that governs information transfer. This chapter describes how to configure Cisco HDLC on E Series routers.

This chapter contains the following sections:

- Overview on page 509
- Platform Considerations on page 510
- Before You Configure Cisco HDLC on page 511
- Configuration Tasks on page 511
- Configuring IPv6 over Cisco HDLC Interfaces on page 514
- Monitoring Cisco HDLC on page 515

Overview

Cisco HDLC is a bit-oriented synchronous data-link layer protocol developed by the International Organization for Standardization (ISO). It specifies a data encapsulation method on synchronous serial links using frame characters and checksums.

By default, synchronous serial lines use the HDLC serial encapsulation method, which provides the synchronous framing-detection and error-detection functions of HDLC without windowing or retransmission.

Cisco HDLC monitors line status on a serial interface by exchanging keepalive request messages with peer network devices. It also enables routers to discover IP addresses of neighbors by exchanging Serial Line Address Resolution Protocol (SLARP) address-request and address-response messages with peer network devices.

The router responds to a SLARP address-request message from a remote peer with a SLARP address-response message, which indicates that it cannot participate in a SLARP session.

Cisco HDLC is compatible with Cisco Systems Cisco-HDLC protocol, the default protocol for all Cisco serial interfaces.

Framing

The router supports the following framing features:

- HDLC for data-link framing
- 18,000-byte information field size

Error Frames

All Cisco HDLC error frames are discarded.

SLARP Keepalive

One feature of Cisco HDLC is the exchange of keepalive messages. A keepalive message is a signal from one endpoint to the other that the first endpoint is still active. Keepalives are used to identify inactive or failed connections.

Platform Considerations

You can configure Cisco HDLC on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support Cisco HDLC on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support Cisco HDLC.

For information about the modules that support Cisco HDLC on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support Cisco HDLC.

Interface Specifiers

The configuration task examples in this chapter use the *slot/port* format to specify the physical interface on which you configure Cisco HDLC. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port* format. For example, the following command specifies a packet over SONET (POS) interface on slot 0, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

host1(config)#**interface pos 0/1**

For E120 and E320 routers, use the *slot/adapter/port* format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies a POS interface on slot 5, adapter 0, port 0 of an E320 router.

host1(config)#**interface pos 5/0/0**

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Before You Configure Cisco HDLC

Before you configure a Cisco HDLC interface, you need to configure the physical interface over which Cisco HDLC traffic flows, described in the following chapters:

- Configuring Channelized T3 Interfaces in *JunosE Physical Layer Configuration Guide*
- Configuring T3 and E3 Interfaces in *JunosE Physical Layer Configuration Guide*

The procedures described in this chapter assume that a physical interface has been configured.

Configuration Tasks

To configure a Cisco HDLC interface:

1. Configure the physical interface on which you want to configure Cisco HDLC.

   host1(config)#**interface serial 3/1:2/1**

2. Select Cisco HDLC as the encapsulation method for the interface.

   host1(config-if)#**encapsulation hdlc**

3. Assign a local IP address to the interface.

   host1(config-subif)#**ip address 192.32.10.2 255.255.255.0**

4. (Optional) Use the appropriate "show hdlc interface" on page 516 to verify that the configuration changes are correct.

encapsulation hdlc

- Use to specify Cisco HDLC as the encapsulation method for the interface.
- Example

  host1(config-if)#**encapsulation hdlc**
- Use the **no** version to disable Cisco HDLC on the interface.
- See *encapsulation hdlc*.

interface serial

- Use to configure a serial interface in the appropriate format by selecting a previously configured physical interface on which you want to configure Cisco HDLC. For example, to specify a channelized T3 interface, use the format *slot/port*:*channel/subchannel.subinterface*.
  - *slot*—Router chassis slot
  - *port*—Port on CT3, T3, or E3 I/O module
  - *channel*—T1 (DS1) channel
  - *subchannel* —Set of DS0 subchannels. For information about T1 subchannels, see *Fractional T1* in *JunosE Physical Layer Configuration Guide*.
  - *subinterface*—User-assigned number that identifies a subinterface
- Example

  host1(config)#**interface serial 3/1:2/1**
- Use the **no** version to remove the interface or subinterface.
- See *interface serial*.

ip address

- Use to assign an IP address and subnet mask to the interface.
- Example

  host1(config-subif)#**ip address 192.32.10.2 255.255.255.0**
- Use the **no** version to remove the IP address of the interface.
- See *ip address*.

Optional Tasks

The following tasks are optional.

1. Configure the SLARP keepalive interval.

   host1(config-if)#**hdlc keepalive 10**
2. Enable loopback detection on an interface.

   host1(config-if)#**hdlc down-when-looped**
3. Disable an interface.

host1(config-if)#**hdlc shutdown**

hdlc down-when-looped

- Use to enable loopback detection on a Cisco HDLC interface.
- By default, loopback detection is disabled.
- Example

  host1(config-if)#**hdlc down-when-looped**
- Use the **no** version to disable loopback detection on a Cisco HDLC interface.
- See *hdlc down-when-looped*.

hdlc keepalive

- Use to specify the keepalive timeout value.
- When the keepalive timer expires, the interface increments its own counter; then it compares the value of this counter with the last value received from a peer. If the difference between the values of the two counters is greater than three, the Cisco HDLC interface is declared down. After that, the interface sends a keepalive message containing the value of its counter and the last received value of the peer's counter.
- The router stores the values received in keepalive messages from a peer interface. If the interface is down, the router compares the received value of its own counter with the value from the peer. If the difference between the values of the two counters is less than four, the router declares the interface to be up. Both sides have to configure the same value for the keepalive interval.
- If the keepalive interval is 10 seconds, then a failed link is detected between 30 and 40 seconds after failure.
- The range is 0–6553 seconds. A value of 0 turns keepalive off.
- The default is 10 seconds.
- Example

  host1(config-if)#**hdlc keepalive 10**
- Use the **no** version to turn off the keepalive feature.
- See *hdlc keepalive*.

hdlc shutdown

- Use to terminate a Cisco HDLC session.
- This command administratively disables the interface.
- Example

  host1(config-if)#**hdlc shutdown**
- Use the **no** version to restart a disabled session. The default for each **hdlc shutdown** command is the **no** version.
- See *hdlc shutdown*.

Configuration Example

This example shows how to configure Cisco HDLC over an unchannelized DS3 interface on a cOCx/STMx line module. The example shows the complete configuration procedure, from configuring the SONET interface to assigning an IP address to the Cisco HDLC interface.

1. Create or select a virtual router, vr1.

   host1(config)#**virtual-router vr1**

2. Configure SONET controller, slot 13, port 0.

   host1:vr1(config)#**controller sonet 13/0**

3. Set the SONET clock source to internal.

   host1:vr1(config-controll)#**clock source internal module**

4. Create an OC1 path.

   host1:vr1(config-controll)#**path 1 oc1 1**

5. Create an unchannelized DS3 interface.

   host1:vr1(config-controll)#**path 1 ds3 1 unchannelized**

6. Set the DS3 interface clock source to internal.

   host1:vr1(config-controll)#**path 1 ds3 1 clock source internal module**

7. Exit Controller Configuration mode.

   host1:vr1(config-controll)#**exit**

8. Create or select a serial interface over the DS3 interface.

   host1:vr1(config)#**interface serial 13/0:1/1**

9. Set the encapsulation to Cisco HDLC.

   host1:vr1(config-if)#**encapsulation hdlc**

10. Enable loopback detection on the interface.

    host1:vr1(config-if)#**hdlc down-when-looped**

11. Assign an IP address to the interface.

    host1:vr1(config-if)#**ip address 160.1.0.1 255.255.255.0**

Configuring IPv6 over Cisco HDLC Interfaces

You can configure IPv6 prefix addresses over a Cisco HDLC interface on a POS physical interface. IPv6 traffic is forwarded over HDLC circuits and the database of the forwarding controller is updated with IPv6 as an upper-layer interface to the HDLC interface.

NOTE: You can specify IPv6 prefixes only over Cisco HDLC interfaces on POS major interfaces. You cannot configure IPv6 addresses on E3 and T3 interfaces.

To configure an IPv6 prefix address over a Cisco HDLC interface on a POS interface:

1. From Configuration mode, enter a POS interface on which you want to configure an HDLC circuit.

   host1(config)#**interface pos 0/1**

2. Select Cisco HDLC as the encapsulation method for the interface.

   host1(config-if)#**encapsulation hdlc**

3. Assign a local IPv6 address to the circuit.

   host1(config-subif)#**ipv6 address 1::1/64**

interface pos

- Use to configure a POS interface in slot/port format:
  - *slot*—Router chassis slot
  - *port*—Line module port
- Example

  host1(config)#**interface pos 0/1**
- Use the **no** version to remove the POS interface.
- See *interface pos*.

ipv6 address

- Use to add an IPv6 address to an interface or a subinterface.
- Example

  host1(config)#**interface gigabitEthernet 1/0.25**
  host1(config-if)#**ipv6 address 1::1/64**
- Use the **no** version to remove an IPv6 address.
- See *ipv6 address*.

Monitoring Cisco HDLC

You can monitor Cisco HDLC interfaces using the **show hdlc interface** command.

You can set a statistics baseline for Cisco HDLC interfaces, subinterfaces, or circuits using the **baseline hdlc interface serial** command.

You can use the filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. For details, see *show Commands* in *JunosE System Basics Configuration Guide*.

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

baseline hdlc interface

- Use to set a statistics baseline for Cisco HDLC interfaces. The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Example

  host1#**baseline hdlc interface serial 2/0:1/1**
- There is no **no** version.
- See *baseline hdlc interface*.

show hdlc interface

- Use to display statistics for the specified HDLC interfaces.
- You can specify the following keywords:
  - **statistics**—Displays interface statistics
  - **delta**—Specifies that baselined statistics are to be shown
  - **status**—Displays the operational status of all configured interfaces
  - **closed**—Displays interfaces with administrative status Closed
  - **config**—Displays configuration information
  - **down**—Displays interfaces with operational status Down
  - **lower-layer-down**—Displays interfaces with operational status LowerLayerDown
  - **not-present**—Displays interfaces with operational status NotPresent
  - **open**—Displays interfaces with administrative status Open
  - **up**—Displays interfaces with operational status Up
  - **full**—Displays configuration information, status, and statistics
  - **filter**—Specifies a CLI output filter
- Field descriptions
  - interface status—State of the interface:

   - Up—Traffic can flow on the interface
   - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
   - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
   - NotPresent—Traffic cannot flow because hardware is unavailable
  - Interface administrative status—Configured state of the interface:
    - Open—**no hdlc shutdown** command is operative
    - Closed—**hdlc shutdown** command is operative
  - Interface maximum-transmission-unit—Configured MTU size
  - Interface keepalive time—Configured keepalive interval value
  - Interface loop detection—Status of loopback detection: enabled, disabled
  - Interface statistics:
    - packets in—Number of inbound packets received on the interface
    - packets out—Number of outbound packets transmitted on the interface
    - octets in—Number of inbound octets received on the interface
    - octets out—Number of outbound octets transmitted on the interface
    - errors in—Number of inbound errors received on the interface
    - errors out—Number of outbound errors transmitted on the interface
    - discards in—Number of inbound packets discarded on the interface
    - discards out—Number of outbound packets discarded on the interface

- Example 1

```
host1#show hdlc interface serial 5/1:5/1
Cisco-HDLC interface serial 5/1:5/1 is LowerLayerDown
```

- Example 2

```
host1#show hdlc interface full
Cisco-HDLC interface serial 4/0:2 is Up
Interface administrative status is open
Interface maximum-transmission-unit is 1596
Interface keepalive time is 10 seconds
Interface loop detection is disabled
Interface statistics             in                  out
   packets                       0                   0
   octets                        242                 242
   errors                        0                   0
   discards                      0                   0
Cisco-HDLC interface serial 5/0:1/1 is NotPresent
2 Cisco-HDLC interfaces found
```

- See *show hdlc interface*.

CHAPTER 19

Configuring Upper-Layer Dynamic Interfaces

This chapter explains upper-layer dynamic interfaces and describes the procedures for configuring them on E Series routers.

This chapter contains the following sections:

- Dynamic Interfaces Overview on page 519
- Upper-Layer Dynamic Interfaces Platform Considerations on page 525
- RADIUS References for Upper-Layer Dynamic Interfaces on page 526
- Upper-Layer Dynamic Interfaces over Static ATM Overview on page 527
- How to Configure Upper-Layer Dynamic Interfaces Using the RADIUS Server on page 527
- Configuring a Dynamic Interface over an ATM 1483 Subinterface on page 530
- Dynamic Encapsulation Type Lockout on page 532
- Creating a PVC on an ATM 1483 Subinterface on page 537
- Dynamic PPP and PPPoE Interfaces over Static ATM on page 538
- Dynamic PPPoE Interfaces over PPPoE Static Interfaces on page 542
- Dynamic IPoA Interfaces Overview on page 556
- Configuring a Dynamic IPoA Interface on page 556
- Dynamic Bridged Ethernet Interfaces on page 558
- Dynamic Interface Configuration Using a Profile on page 565
- Configuring Profile Characteristics on page 576
- Scripts and Macros for Dynamic Interfaces Overview on page 589
- Reassigning a Debug Profile Before Troubleshooting PPP and PPPoE Dynamic Interfaces on page 590

Dynamic Interfaces Overview

A *dynamic interface* is created automatically and transparently through some external event, typically through the receipt of data over a lower-layer link, such as an Asynchronous Transfer Mode (ATM) virtual circuit (VC) or a virtual LAN (VLAN) through a process known as *autodetection*.

The layers of a dynamic interface are created based on the packets received on the link and can be configured through any one of the following:

- RADIUS authentication
- Profiles
- A combination of RADIUS authentication and profiles

You create and configure each layer of a *static interface* manually through an existing configuration mechanism such as the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

Unlike static interfaces, dynamic interfaces are not restored through nonvolatile storage (NVS) after a reboot.

This topic describes the following:

- Autodetection on page 520
- Types of Dynamic Interfaces on page 520
- Upper-Layer Dynamic Interface Configurations on page 521
- Profiles on page 522
- RADIUS Authentication on page 523
- ATM Oversubscription for Dynamic Interfaces on page 523
- Ethernet Oversubscription for Dynamic Interfaces on page 525

Autodetection

The router performs *autodetection,* also referred to as *autosensing,* to determine the layers of each dynamic interface. The autodetection process occurs when the router conditionally constructs interface layers based on the encapsulation type of the incoming packet.

Autodetection only uses system resources on demand based on what is detected in the incoming packet. Dynamic interfaces are created as a result of traffic on the interface. Dynamic interfaces can also be dynamically deleted without your intervention, thereby enabling any consumed system resources to be returned.

Unlike dynamic interfaces, static interfaces always allocate system resources upon creation, and always consume system resources, even when the interface is quiescent.

Types of Dynamic Interfaces

There are two types of dynamic interfaces: upper-layer and bulk-configured. Bulk-configured dynamic interfaces enable you to dynamically create ATM 1483 subinterfaces and VLAN subinterfaces by bulk-configuring a range of identifiers. There are two types of bulk-configured dynamic interfaces:

- ATM 1483 interfaces over static ATM AAL5 interface
- VLAN subinterface over static VLAN major interface

For more information, see "Configuring Dynamic Interfaces Using Bulk Configuration" on page 619.

Upper-layer dynamic interfaces enable you to dynamically create the following configurations:

- Dynamic IP, Point-to-Point Protocol over Ethernet (PPPoE), Point-to-Point Protocol (PPP), Multilink Point-to-Point Protocol (MLPPP), and bridged Ethernet interfaces over a static ATM 1483 interface
- IP or PPPoE interfaces over VLAN interfaces and Ethernet, Gigabit Ethernet, and 10-Gigabit-Ethernet interfaces.

NOTE: **Ethernet interfaces in this topic refer to any of these interfaces Fast Ethernet, Gigabit Ethernet, or 10–Gigabit-Ethernet.**

Upper-Layer Dynamic Interface Configurations

E Series routers support the following types of upper-layer dynamic interface configurations:

- Dynamic IP over static ATM 1483 (IPoA)
- Dynamic IP over dynamic PPP over static ATM 1483
- Dynamic IP over dynamic PPP over dynamic PPPoE over static ATM 1483
- Dynamic IP over dynamic bridged Ethernet over static ATM 1483
- Dynamic IP over dynamic MLPPP over static ATM 1483
- Dynamic IP over dynamic MLPPP over dynamic PPPoE over static ATM 1483
- Dynamic IP over dynamic PPP over dynamic PPPoE subinterface over static PPPoE major interface (with or without VLANs)
- Dynamic IP over dynamic MLPPP over dynamic PPPoE subinterface over static PPPoE major interface (with or without VLANs)
- Dynamic IP over dynamic MLPPP over dynamic PPPoE (with or without VLANs)

E Series routers also support the following types of upper-layer dynamic interfaces over static VLAN interfaces:

- Dynamic Subscriber Interfaces
- IP over Ethernet
- IP over PPP over Ethernet

IP version 4 (IPv4) is supported for all of these upper-layer dynamic interface configurations.

Currently, IP version 6 (IPv6) is supported only when PPP or MLPPP is the layer immediately below the IPv6 layer in the interface column. Dynamic IPv6 is *not* supported directly over static ATM 1483, dynamic bridged Ethernet, or dynamic VLANs. Upper-layer dynamic interface columns that support IPv6 include the following:

- Dynamic IPv6 over dynamic PPP over static ATM 1483
- Dynamic IPv6 over dynamic MLPPP over static ATM 1483
- Dynamic IPv6 over dynamic PPP over dynamic PPPoE over static ATM 1483
- Dynamic IPv6 over dynamic MLPPP over dynamic PPPoE over static ATM 1483
- Dynamic IPv6 over dynamic PPP over dynamic PPPoE subinterface over static PPPoE major interface (with or without VLANs)
- Dynamic IPv6 over dynamic MLPPP over dynamic PPPoE subinterface over static PPPoE major interface (with or without VLANs)

For more information about IPv4, see *Configuring IP* and *Monitoring IP* in *JunosE IP, IPv6, and IGP Configuration Guide*. For more information about IPv6, see *Configuring IPv6* and *Monitoring IPv6* in *JunosE IP, IPv6, and IGP Configuration Guide*.

Profiles

You can use profiles to configure dynamic interfaces over ATM, VLAN, or Ethernet Interfaces. A *profile* is a set of characteristics that can be dynamically assigned to interfaces. By using a profile, you reduce the management of a large number of interfaces by applying a set of characteristics to multiple interfaces.

When you are configuring a large number of interfaces with the same attributes at the higher layers, you can use a profile to factor out all the common attributes of each layer into one place. This action affects one or more dynamic layers of the interface column. After you define the static lower layers, you assign a profile to the highest static layer of the interface column.

When a dynamic interface is configured, the configuration data received from the RADIUS authentication server typically overrides configuration data obtained from a profile.

In contrast to static PPP interfaces (above which only dynamic IP interfaces can be created), static ATM 1483 subinterfaces support recognition and creation of the following upper dynamic interface types or *encapsulations*:

- Bridged Ethernet
- IP
- IPv6
- Multilink PPP
- PPP
- PPPoE

The **auto-configure** command identifies the encapsulation type. For flexibility, the router provides the ability to configure an ATM 1483 subinterface with distinct profile assignments for each encapsulation type supported by the **auto-configure** command.

For more information about using this command, see "Configuring a Dynamic Interface over an ATM 1483 Subinterface" on page 530.

RADIUS Authentication

RADIUS helps protect your network against unauthorized access. To accomplish this, RADIUS clients running on your router send authentication requests to a central RADIUS server. You can configure dynamic interfaces over interfaces through RADIUS authentication.

When a packet is received, the authenticating interface, either PPP or ATM 1483, establishes a session with RADIUS and passes the username and password to the RADIUS server. For dynamic IPoA or dynamic bridged Ethernet, the RADIUS username and password are obtained from the information specified by the **subscriber** command. The RADIUS server returns a grant or deny indication. If authentication is granted, the RADIUS attributes are returned, a user login is created, and the dynamic interfaces are configured from the RADIUS attributes. For more information about using this command, see "Configuring a Local Subscriber for a Dynamic IPoA or Bridged Ethernet Interface" on page 528.

ATM 1483 interfaces may receive configuration data from the RADIUS server in the form of *traffic-shaping* parameters.

Any changes made to a RADIUS configuration for a given dynamic interface do not take effect until an existing dynamic interface configured from this RADIUS entry is re-created, that is, deleted and then dynamically created.

ATM Oversubscription for Dynamic Interfaces

You can take advantage of oversubscription of static ATM 1483 subinterfaces and bulk-configured ATM VCs with the following dynamic interface configurations:

- The router supports oversubscription of static ATM 1483 subinterfaces when you configure the static ATM 1483 subinterface to support one of the following dynamic upper-layer encapsulation types: bridged Ethernet, IP, Multilink PPP, PPP, and PPPoE interfaces. For information about configuring dynamic upper-layer encapsulation types over a static ATM 1483 subinterface, see "Upper-Layer Dynamic Interfaces over Static ATM Overview" on page 527.
- The router supports oversubscription of bulk-configured VC ranges when you create a bulk-configured VC range on a static ATM AAL5 interface for use by a dynamic ATM 1483 subinterface. For information about configuring dynamic ATM 1483 subinterfaces with bulk-configured VC ranges, see "Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview" on page 626 in "Configuring Dynamic Interfaces Using Bulk Configuration" on page 619.

This section describes the following:

- How Oversubscription Works on page 524
- Static ATM 1483 Subinterfaces on page 524

- Bulk-Configured VC Ranges on page 524
- Combination of Static ATM 1483 Subinterfaces and Bulk-Configured VC Ranges on page 525

How Oversubscription Works

Oversubscription is based on the capabilities of the ATM line module on which the dynamic interface is configured. For details about the capabilities of specific ATM line modules, see either "Module Capabilities" on page 12 in "Configuring ATM" on page 3, or the *Link Layer Maximums* tables in *JunosE Release Notes*, *Appendix A*, *System Maximums.*

Each ATM line module supports a maximum number of configured subinterfaces or VCs, and a smaller maximum number of subinterfaces or VCs that can be active at any one time. The maximum number of active subinterfaces or VCs determines the number of subscribers that can connect to the router through this line module at any one time.

As a result, you can oversubscribe static ATM 1483 subinterfaces or bulk-configured VC ranges by creating up to the maximum number of configured subinterfaces or VCs supported on the module, knowing that no more than the maximum number of active subinterfaces or VCs can be connected to the router at any one time.

Static ATM 1483 Subinterfaces

An active static ATM 1483 subinterface currently supports a dynamic upper-layer encapsulation type such as PPP or PPPoE. For ATM line modules that support ATM subinterface oversubscription, the maximum number of active subinterfaces supported per module is less than the maximum number of configured subinterfaces supported per module.

When the maximum number of active ATM 1483 subinterfaces has been reached, the router prevents all additional subscribers from connecting to the line module until at least one currently active subscriber logs out, which causes the router to tear down the dynamic interface column for that subscriber. When a dynamic interface column is torn down, the router enables the first currently inactive subscriber that receives traffic to connect to the router and become active as a replacement for the subscriber that logged out.

Consider an ATM line module that supports a maximum of 16,000 configured subinterfaces and a maximum of 8000 active subinterfaces. If all 16,000 static ATM 1483 subinterfaces attempt to connect to the router, only the first 8000 subinterfaces to receive traffic are able to log in, generate dynamic interface columns, and become active. When a subscriber connected through one of these active subinterfaces logs out, the router enables the first of the remaining 8000 inactive subinterfaces that receives traffic to connect as a replacement for the subscriber that logged out.

Bulk-Configured VC Ranges

An active bulk-configured VC range is associated with a dynamic ATM 1483 subinterface that supports a dynamic upper-layer encapsulation type. For ATM line modules that support VC oversubscription, the maximum number of active bulk-configured VCs per line module is less than the maximum number of individual VCs created from the total number of bulk-configured VC ranges that the line module supports.

For details about how oversubscription works for bulk-configured VC ranges, see "Configuring Dynamic Interfaces Using Bulk Configuration Overview" on page 620 in "Configuring Dynamic Interfaces Using Bulk Configuration" on page 619.

Combination of Static ATM 1483 Subinterfaces and Bulk-Configured VC Ranges

ATM line modules are sometimes configured with a combination of static ATM 1483 subinterfaces and bulk-configured VC ranges. In these configurations, both the static ATM 1483 subinterfaces and bulk-configured VC ranges can support active subinterfaces. The combined total of active static ATM 1483 subinterfaces, and active dynamic ATM 1483 subinterfaces created from bulk-configured VC ranges, cannot exceed the maximum number of active subinterfaces supported by the line module.

For details about how oversubscription works for ATM modules configured with both static ATM 1483 subinterfaces and bulk-configured VC ranges, see "Configuring Dynamic Interfaces Using Bulk Configuration Overview" on page 620 in "Configuring Dynamic Interfaces Using Bulk Configuration" on page 619.

Ethernet Oversubscription for Dynamic Interfaces

When you configure S-VLAN subinterfaces over Ethernet interfaces to support dynamic PPPoE subinterfaces, you can take advantage of VLAN and S-VLAN oversubscription.

For more information on S-VLAN oversubscription, see "S-VLAN Oversubscription" on page 179.

Related Documentation

- Upper-Layer Dynamic Interfaces Platform Considerations on page 525
- RADIUS References for Upper-Layer Dynamic Interfaces on page 526

Upper-Layer Dynamic Interfaces Platform Considerations

You can configure dynamic interfaces on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support dynamic interfaces on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.

- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support dynamic interfaces.

For information about the modules that support dynamic interfaces on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support dynamic interfaces.

Interface Specifiers

The interface specifier format that you use depends on the router that you are using. The examples in this section use the *slot/port*[.*subinterface*] format and the *slot/adapter/port*[.*subinterface*] format to specify the physical interface that you want to configure to support dynamic interfaces.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Related Documentation

- Dynamic Interfaces Overview on page 519
- RADIUS References for Upper-Layer Dynamic Interfaces on page 526

RADIUS References for Upper-Layer Dynamic Interfaces

For more information about RADIUS, consult the following resources:

- RFC 2865—Remote Authentication Dial In User Service (RADIUS) (June 2000)
- RFC 2866—RADIUS Accounting (June 2000)

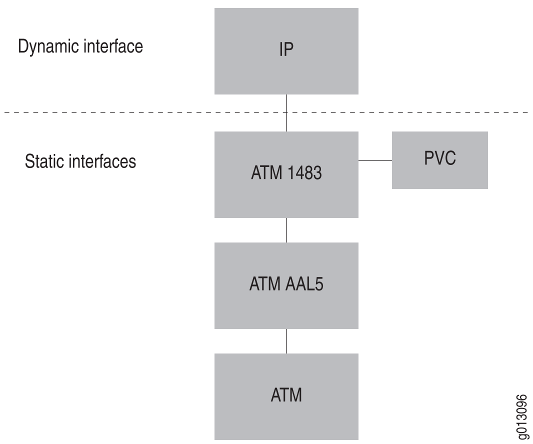
Related Documentation

- Dynamic Interfaces Overview on page 519
- Upper-Layer Dynamic Interfaces Platform Considerations on page 525

Upper-Layer Dynamic Interfaces over Static ATM Overview

To create dynamic interfaces over Asynchronous Transfer Mode (ATM), you create the static layers of the interface first, and then configure them to support a dynamic interface by means of autodetection. Figure 46 on page 527 shows an example of the interface stack for a dynamic IP over ATM 1483 interface.

Figure 46: Configuring an ATM 1483 Interface to Support Dynamic Interfaces



On receipt of a packet, the router creates all dynamic layers above the ATM 1483 layer, starting with the lowest dynamic layer. For example, in the case of a dynamic Point-to-Point Protocol over Ethernet (PPPoE) interface, the router creates the PPPoE interface first, then the Point-to-Point (PPP) interface, and then the IP interface.

If any layer of the dynamic portion of the interface column fails to be created, then the interface creation fails and the connection is denied. All dynamic layers above the ATM 1483 subinterface are destroyed, starting with the highest dynamic layer.

When you configure a dynamic interface, you must assign (or create and assign) a profile to the interface. Profile creation and assignment topics are discussed in depth in "Dynamic Interface Configuration Using a Profile" on page 565.

Related Documentation

- How to Configure Upper-Layer Dynamic Interfaces Using the RADIUS Server on page 527
- Configuring a Dynamic Interface over an ATM 1483 Subinterface on page 530
- Dynamic Encapsulation Type Lockout on page 532
- Creating a PVC on an ATM 1483 Subinterface on page 537
- Dynamic IPoA Interfaces Overview on page 556

How to Configure Upper-Layer Dynamic Interfaces Using the RADIUS Server

Dynamic interfaces can be configured automatically through authentication and authorization by the RADIUS server.

On Asynchronous Transfer Mode (ATM) interfaces, you initially create the static portion of the interface column by creating an ATM interface, ATM 1483 subinterface, and underlying ATM permanent virtual circuit (PVC).

- Configuring a Local Subscriber for a Dynamic IPoA or Bridged Ethernet Interface on page 528
- Subscriber Authentication on Dynamic Bridged Ethernet over Static ATM Interfaces on page 528
- Dynamic IP Route Insertion in the Routing Table Overview on page 530

Configuring a Local Subscriber for a Dynamic IPoA or Bridged Ethernet Interface

For dynamic interfaces that do not have a PPP layer, such as IPoA, you can use the **subscriber** command to configure an ATM 1483 subinterface to be authenticated automatically by the RADIUS server. The **subscriber** command uses a RADIUS username and optional password for identification and is available only for bridged Ethernet and IPoA configurations. This command is used for dynamic encapsulations that do not provide the authentication information remotely, as PPP does.

For dynamic interfaces with a PPP layer, the RADIUS username and password are obtained from the remote client, and authentication is performed with the RADIUS server. The attributes obtained from RADIUS can then be used to configure any higher-layer dynamic interfaces, such as IP, that are built over PPP.

If your router is running stateful SRP switchover (high availability), the use of the **subscriber** command to configure RADIUS authentication for subscribers on dynamic bridged Ethernet interfaces might suspend stateful SRP switchover on the router or prevent stateful SRP switchover from becoming active. For more information about using the subscriber management application to bypass this limitation, see "Subscriber Authentication on Dynamic Bridged Ethernet over Static ATM Interfaces" on page 528.

To configure a local subscriber on the E Series router to support authentication and configuration from RADIUS for a dynamic IPoA or bridged Ethernet interface:

- Issue the **subscriber** command in Subinterface Configuration mode.
  a. To set IP as the dynamic-layer upper interface:

     host1(config-subif)#**subscriber ip user-prefix charlie domain myisp password-prefix lucy**

  b. To set bridged Ethernet as the dynamic-layer upper interface:

     host1(config-subif)#**subscriber bridgedEthernet user westford003 domain acmecorp.east password xyz123**

  Use the **no** version to remove the subscriber.

Subscriber Authentication on Dynamic Bridged Ethernet over Static ATM Interfaces

You can use either of the following methods to configure and manage RADIUS authentication for IP subscribers on dynamic bridged Ethernet over static ATM interfaces:

- The **subscriber** command

- The subscriber management application

The **subscriber** command *does not support* running stateful switch route processor (SRP) switchover (high availability) on the router. Therefore, the configuration method you choose depends on whether stateful SRP switchover is or is not running on your router.

This section describes the following:

- Configuration Method Using subscriber Command on page 529
- Configuration Method Using Subscriber Management Application on page 529

Configuration Method Using subscriber Command

When you use the **subscriber** command to configure IP subscribers on dynamic bridged Ethernet over static ATM 1483 interface columns to support RADIUS authentication, the **subscriber** command provides the subscriber's authentication parameters. The static ATM 1483 subinterface acts as the authenticating layer that establishes a session with RADIUS and passes the subscriber's locally configured username and password information to the RADIUS server.

However, if your router is running stateful SRP switchover (high availability), the use of the **subscriber** command in this configuration might suspend stateful SRP switchover on the router or prevent stateful SRP switchover from becoming active. To bypass this limitation, you can use the subscriber management application to configure IP subscribers on dynamic bridged Ethernet interfaces.

Configuration Method Using Subscriber Management Application

You can use the JunosE subscriber management application to configure and manage IP subscribers associated with a dynamic bridged Ethernet interface column. The subscriber management application uses an IP service profile to manage and authenticate IP subscribers with RADIUS. An IP service profile contains user and password information, and is used in a route map for subscriber management and to authenticate subscribers with RADIUS.

In this configuration, the IP service profile provides the subscriber's authentication parameters, and the subscriber management application acts as the authenticating layer to obtain information from RADIUS for configuration of dynamic IP subscribers. To assign the IP service profile to the interface profile from which the dynamic bridged Ethernet interface is created, you use the **bridge1483 service-profile** command in Profile Configuration mode.

If stateful SRP switchover is disabled or not running on your router, you can continue to use the **subscriber** command to configure IP subscribers on dynamic bridged Ethernet interfaces to support RADIUS authentication.

Alternatively, you can use the subscriber management application to create and configure dynamic IP interfaces regardless of whether stateful SRP switchover is running on the router. In addition, using subscriber management enables you to take advantage of several useful features such as the IP inactivity timer.

In the event that an interface profile for a dynamic bridged Ethernet interface includes the **subscriber** command to configure a local subscriber as well as the **bridge1483 service-profile** command to reference an IP service profile, the values specified with the **subscriber** command take precedence. The router ignores the values in the IP service profile in this case.

For details about using the subscriber management application to configure RADIUS authentication for IP subscribers on dynamic bridged Ethernet interfaces, see "Configuring Subscriber Management for IP Subscribers on Dynamic Bridged Ethernet Interfaces" on page 561.

For more information about using the subscriber management application, see *JunosE Broadband Access Configuration Guide*.

Dynamic IP Route Insertion in the Routing Table Overview

If you want to insert a dynamic IP route into the routing table of the relevant virtual router to point to the subscriber's subinterface, you can use the Framed-Route [22] RADIUS attribute to do so. Defined by RFC 2865—Remote Authentication Dial In User Service (RADIUS) (June 2000), the Framed-Route attribute can be returned in Access-Accept messages to specify the route as follows:

Framed-Route = *ipAddress/mask nextHop*

For dynamic IP interfaces, the next hop might not be known when you create the user record. In this case, use the value 0.0.0.0 for the next hop; the E Series router then assigns the subinterface associated with the user as the next hop in the routing table.

Related Documentation

- Upper-Layer Dynamic Interfaces over Static ATM Overview on page 527
- Benefits of Encapsulation Type Lockout on page 533
- Scripts and Macros for Dynamic Interfaces Overview on page 589

- *bridge1483 service-profile*
- *subscriber*

Configuring a Dynamic Interface over an ATM 1483 Subinterface

You use the **auto-configure** command to configure an ATM 1483 subinterface to support a dynamic interface. After the subinterface is configured, it performs autodetection to identify the encapsulation, resulting in the dynamic creation of the higher protocol layers. This command specifies one or more types of next upper dynamic encapsulations that the static interfaces can detect or accept.

This command creates the layers above ATM 1483 *dynamically*. You can enter the command repetitively in Subinterface Configuration mode to support multiple dynamic interface types.

NOTE: **On static ATM 1483 interfaces, dynamic encapsulation types can be bridged Ethernet, IP, IPv6, PPP, or PPPoE.**

Encapsulation type lockout is performed on a per-encapsulation-type basis for each subinterface. An encapsulation type not configured for autodetection with the **auto-configure** command is automatically locked out. The lockout temporarily prevents the static ATM 1483 subinterface from detecting, accepting, and creating the encapsulation type until the lockout time expires.

You can use the **lockout-time** keyword to set the minimum lockout time and maximum lockout time, each of which can be in the range 1–86400 seconds (24 hours). The default range is 1–300 seconds (5 minutes).

You can use the **none** keyword with the **lockout-time** keyword to disable lockout for the specified encapsulation type.

NOTE: **Disabling lockout can result in undesirable CPU loading; we recommend that you not disable lockout for general use. At a minimum, use the default lockout time.**

For information about the rules that apply when you configure the lockout time for dynamic encapsulation type lockout, see "Guidelines for Configuring Encapsulation Type Lockout for PPPoE Sessions" on page 535.

To configure a static ATM 1483 subinterface to support a dynamic interface:

- Issue the **auto-configure** command in Subinterface Configuration mode.
  a. Enable autodetection for the PPPoE encapsulation type using the default lockout time range, 1–300 seconds.

     host1(config-subif)#**auto-configure pppoe**

  b. Enable autodetection for the PPP encapsulation type using a nondefault lockout time range, 5–60 seconds.

     host1(config-subif)#**auto-configure ppp lockout-time 5 60**

  c. Disable encapsulation type lockout for the PPPoE encapsulation type.

     host1(config-subif)#**auto-configure pppoe lockout-time none**

  d. Reenable encapsulation type lockout for the PPPoE encapsulation type using the default lockout time range.

     host1(config-subif)#**auto-configure pppoe**

     Or

     host1(config-subif)#**no auto-configure pppoe lockout-time**

  e. Permanently lock out the PPP encapsulation type until the **auto-configure ppp** command is issued.

host1(config-subif)#**no auto-configure ppp**

Use the **no** version to terminate detection of the specified encapsulation type or, if the **lockout-time** keyword is specified, to restore the lockout time range to its default value, 1–300 seconds.

Related Documentation

- Upper-Layer Dynamic Interfaces over Static ATM Overview on page 527
- Dynamic Encapsulation Type Lockout on page 532
- Dynamic PPP and PPPoE Interfaces over Static ATM Overview on page 538
- Monitoring Status or Summary Information for ATM Subinterfaces on page 595
- Monitoring Total Static and Dynamic Interface Counts for Interface Columns on page 602

- *auto-configure*

Dynamic Encapsulation Type Lockout

You can configure E Series routers to support dynamic encapsulation type lockout. With this feature, you can temporarily prevent an Asynchronous Transfer Mode (ATM) 1483 subinterface from autodetecting, accepting, and creating dynamic interface columns for a configurable time period.

On ATM 1483 subinterfaces, encapsulation type lockout is the default behavior for Internet Protocol over Asynchronous Transfer Mode (IPoA), bridged Ethernet, Point-to-Point Protocol (PPP), and Point-to-Point Protocol over Ethernet (PPPoE) encapsulation types.

- How Encapsulation Type Lockout Works on page 532
- Benefits of Encapsulation Type Lockout on page 533
- Encapsulation Type Lockout Based on DSL Forum VSAs for IWF PPPoE Sessions on page 534
- Guidelines for Configuring Encapsulation Type Lockout for PPPoE Sessions on page 535
- Guidelines for Configuring Encapsulation Type Lockout for IWF PPPoE Sessions on page 535
- Encapsulation Type Lockout for IWF PPPoE Sessions Overview on page 536

How Encapsulation Type Lockout Works

For a given encapsulation type, such as bridged Ethernet, lockout occurs when a dynamic interface of this type cannot be created. For example, an authentication denial from RADIUS causes a lockout. When lockout occurs, the router applies the lockout time range. If you do not configure a lockout-time range, the router uses the default time range.

Encapsulation type lockout is performed by default. You can configure the lockout time range by issuing the **auto-configure** command with the optional **lockout-time** keyword.

The following guidelines describe lockout behavior:

- Any encapsulation type that you do not configure for autodetection with the **auto-configure** command is automatically locked out.
- You can permanently lock out a specified encapsulation type from autodetection and prevent dynamic interface creation by issuing a **no auto-configure** command for the specified encapsulation type, if previously configured.
- When an encapsulation type is locked out, the router continues to autodetect the remaining encapsulation types and create the dynamic interfaces.

For the IP and bridged Ethernet encapsulation types, temporary lockout occurs automatically on receipt of an authentication deny response from RADIUS when you attempt to create and configure a dynamic IPoA or dynamic bridged Ethernet interface.

The lockout time range comprises two values: a minimum lockout time and a maximum lockout time. The initial lockout time begins with the minimum lockout time. From this point, the lockout time increases exponentially for every successive lockout event within the greater of 15 minutes or the maximum configured lockout time. The lockout time never exceeds the maximum value of the time range.

For example, using the default lockout time range of 1–300 seconds, the increasing lockout time sequence is: 1 second, 2 seconds, 4 seconds, 8 seconds, 16 seconds, 32 seconds, 64 seconds, 128 seconds, 256 seconds, and finally, 300 seconds (5 minutes).

Benefits of Encapsulation Type Lockout

Using dynamic encapsulation type lockout provides the following benefits:

- Enables autodetection of other encapsulation types when a dynamic interface for a specified encapsulation type cannot be created.

  For example, when running a PPPoE client, digital subscriber line (DSL) modems might transmit bridged Ethernet frames among the PPPoE frames. When bridged Ethernet and PPPoE encapsulation types are configured for autodetection with the **auto-configure** command, and a subscriber is configured for the bridged Ethernet encapsulation type, RADIUS sends a deny response after the router attempts to authenticate a received bridged Ethernet frame. Receiving an authentication denial from RADIUS causes the router to lock out bridged Ethernet. By locking out bridged Ethernet frames, the router can receive PPPoE frames unimpeded, facilitating rapid creation of dynamic PPPoE interfaces.

- Reduces loading on the RADIUS server.

  In some cases, IP and bridged Ethernet interfaces configured with a local subscriber do not have a corresponding subscriber entry in the RADIUS database. This can occur inadvertently due to misconfiguration of the E Series router or RADIUS server, or intentionally as a way to prevent creation of dynamic IPoA or bridged Ethernet interfaces.

  In previous releases, when the ATM 1483 interface received a deny response from RADIUS due to the missing subscriber entry, it performed continuous authentication retries every few seconds, which caused significant loading on the RADIUS server. Locking out autodetection of the IP or bridged Ethernet encapsulation type for a configurable time period prevents detection of dynamic IPoA or bridged Ethernet interfaces and reduces loading on the RADIUS server.

For PPP and PPPoE encapsulation types, incorrect logins coupled with clients configured to perform frequent authentication retries results in significant loading on the RADIUS server. When an incorrect login occurs, the process of autodetecting, creating partial dynamic interface columns, and tearing down the columns due to authentication failures consumes router bandwidth. Enabling temporary lockout of PPP and PPPoE encapsulation types reduces loading on the RADIUS server caused by incorrect logins and auto-retry clients.

- Reduces loading on line modules.

  The repeated creation of multiple short-cycle dynamic interfaces causes excessive loading on line modules. A *short-cycle dynamic interface* is one that is detected, partially or completely created, and torn down within 60 seconds.

  Events that can cause short-cycle dynamic interfaces include:

  - Authentication denials from RADIUS due to the absence of a corresponding entry in the RADIUS database or due to improper login attempts
  - Misconfiguration within a dynamic interface profile or RADIUS record
  - Insufficient memory resources to create a dynamic interface column
  - Protocol failure or error that occurs within a dynamic interface column
  - Client logout shortly after a successful login; this action creates a complete dynamic interface column before the column is torn down

Encapsulation Type Lockout Based on DSL Forum VSAs for IWF PPPoE Sessions

JunosE Software supports the dynamic encapsulation type lockout functionality for PPPoE sessions that contain the IWF-Session DSL Forum vendor-specific attribute (VSA) (26-254) in the PPPoE active discovery request (PADR) packets. For interworking function (IWF) sessions that involve a set of functions to be processed to interconnect two networks of different technologies (such as PPPoE over ATM to PPPoE), the encapsulation type lockout for the PPPoE clients associated with the dynamic PPPoE subinterface column on the PPPoE major interface is determined using a combination of the Agent-Circuit-Id (26-1) and Agent-Remote-Id (26-2) DSL Forum VSAs, in addition to the media access control (MAC) address.

The DSL Forum VSAs used in the encapsulation type lockout process for IWF PPPoE sessions comprise Agent-Circuit-Id (26-1) and Agent-Remote-Id (26-2). The Agent-Circuit-Id VSA is the identifier for the subscriber agent circuit that corresponds to the digital subscriber line access multiplexer (DSLAM) interface from which subscriber requests are initiated. The Agent-Remote-Id VSA is the unique identifier for the subscriber associated with the DSLAM interface from which requests are initiated. For PPPoE sessions with the IWF-Session VSA, if you configured the **pppoe auto-configure lockout-time** command in Interface Configuration mode or Subinterface Configuration mode, the MAC address, Agent-Circuit-Id, and Agent-Remote-Id values are used together to identify a subscriber to implement PPPoE lockout.

If subscriber PPP sessions are transported on PPPoE, the PPPoE intermediate agent on the DSLAM adds the Agent-Circuit-Id and Agent-Remote-Id VSAs to the PPPoE PADI and PADR packets. The PPPoE implementation technique captures both the Agent-Circuit-Id and the Agent-Remote-Id sub-options from every PADR packet for every PPPoE session. Dynamic encapsulation type lockout is enabled by default for all IWF PPPoE sessions.

Guidelines for Configuring Encapsulation Type Lockout for PPPoE Sessions

The following rules apply when you configure the lockout time for dynamic encapsulation type lockout:

- The lockout time value is defined as

  (minimum lockout time) * (2 ^ *n*-1)

  where *n* represents the number of consecutive lockout events.
- The router increments the value of *n* when the time between lockout events is either within 15 minutes or the maximum lockout time, whichever is greater.
- When the time between lockout events is greater than either 15 minutes or the maximum lockout time, the value of *n* reverts to 1. This condition is referred to as a *grace period*.
- The lockout time never exceeds the maximum configured lockout time. For example, for a configured lockout time in the range 20–120 seconds, the increasing lockout time sequence is 20 seconds, 40 seconds, 80 seconds, and finally, 120 seconds.
- A *short-cycle event* is a dynamic interface that is created and torn down within 60 seconds. The router tracks the time between short-cycle events to determine whether to increase the lockout time for a subsequent short-cycle event.

  NOTE: When the calculated lockout time is equal to or exceeds the maximum lockout time, the router uses the maximum lockout time value until the time to the next event exceeds the greater of 15 minutes or the maximum lockout time value. At that point, the lockout time reverts to the minimum lockout time value.

- The minimum lockout time value cannot exceed the maximum lockout time value. When the minimum and maximum values are equal, the encapsulation type lockout time becomes fixed.

Guidelines for Configuring Encapsulation Type Lockout for IWF PPPoE Sessions

Keep the following points in mind while configuring dynamic encapsulation type lockout for IWF PPPoE sessions:

- If the DSLAM network does not add the DSL Forum VSAs (Agent-Circuit-Id and Agent-Remote-Id) that are required to accurately and uniquely identify the subscriber access loop, dynamic encapsulation type lockout for PPPoE clients is performed using MAC address as the only criterion to identify subscribers.

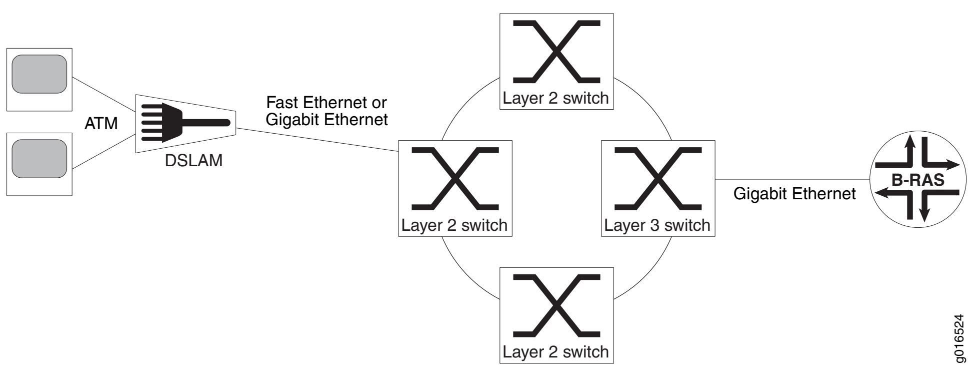
In such conditions, the mode of working of encapsulation type lockout for PPPoE clients is the same as the behavior that existed when lockout of PPPoE clients was performed using the MAC address of the client as the only matching parameter.

- If the Agent-Circuit-Id DSL Forum VSA (26-1) is not contained in the PADR packet sent from the PPPoE client or is not unique for each IWF PPPoE session that originates from the client, the Agent-Remote-Id DSL VSA (26-2) is used along with the MAC address as the filter criteria to uniquely identify the subscriber session that needs to be locked out.
- When you enter the **pppoe auto-configure lockout-time** command in Interface Configuration mode or Subinterface Configuration mode to configure dynamic encapsulation type lockout, even if PPPoE sessions associated with a particular MAC address are locked out, other PPPoE sessions that originated with the same MAC address are not terminated (continue to remain logged in) if they are IWF sessions from different access loops (PPPoE clients) and this information is available to the B-RAS application.

Encapsulation Type Lockout for IWF PPPoE Sessions Overview

Consider a sample configuration scenario in which subscriber access loops are connected using ATM links to a digital subscriber line access multiplexer (DSLAM) device. Figure 47 on page 536 shows a topology in which one DSLAM is connected one PPPoE access concentrator. The ATM traffic is forwarded using an Ethernet circuit from the DSLAM device to the PPPoE access concentrator. The PPPoE over ATM or PPP over ATM (PPPoA) traffic is converted to PPPoE by the DSLAM device, which is connected to an ATM access loop on one side and an Ethernet aggregate network on the other side. The interworking function (IWF) describes the set of mechanisms used to convert a PPPoE over ATM or PPPoA session to a PPPoE session.

Figure 47: Single DSLAM Connected to a PPPoE Access Concentrator



The conversion of PPP, PPPoE, Ethernet, ATM 1483, or ATM connections at the DSLAM access loop interface to PPP, PPPoE, or Ethernet connections at the DSLAM aggregate network interface signifies that the MAC addresses of all such IWF sessions are translated to be the MAC address of the DSLAM Ethernet interface connected to the aggregate network. Therefore, you cannot uniquely identify a subscriber using the media access control (MAC) address only. If the MAC address alone is used to identify a particular subscriber, a single erroneous IWF session can cause other IWF sessions to be locked out.

In this setup, if the PPPoE session contains the IWF-Session DSL Forum VSA (26-254) in the PPPoE active discovery request (PADR) packet that is sent from the PPPoE client, the Agent-Circuit-Id DSL Forum VSA (26-1) is used in addition to the MAC address to identify the PPPoE session to be locked out. This method enables backward compatibility with all non-IWF topologies. The Agent-Circuit-Id is of the format, *DSLAM name Slot/Port VPI: VCI*, which enables unique identification of the subscriber that has initiated the session. This feature of encapsulation type lockout based on the Agent-Circuit-Id, in addition to the MAC address retrieved from the client, is enabled by default only for IWF sessions.

Related Documentation

- Upper-Layer Dynamic Interfaces over Static ATM Overview on page 527
- Configuring a Local Subscriber for a Dynamic IPoA or Bridged Ethernet Interface on page 528
- Configuring a Dynamic Interface over an ATM 1483 Subinterface on page 530
- Encapsulation Type Lockout for PPPoE Clients Overview on page 550
- Configuring Encapsulation Type Lockout for PPPoE Clients on page 552

- *auto-configure*
- *pppoe auto-configure*

Creating a PVC on an ATM 1483 Subinterface

You use the **atm pvc** command to define the underlying circuit supporting an ATM 1483 subinterface. When you define a circuit with this command by using the **aal5autoconfig** option, it causes the ATM 1483 encapsulation (LLC/SNAP encapsulation or VC multiplexed) to be autodetected. Alternatively, if you use the **aal5snap** or **aal5mux ip** option, the ATM 1483 encapsulation becomes fixed, but higher layers can be dynamic.

To configure a circuit for autodetection of the ATM 1483 encapsulation and all higher layers:

- Issue the **atm pvc** command with the **aal5autoconfig** option in Subinterface Configuration mode.

  ```
  host1(config-subif)#atm pvc 100 0 100 aal5autoconfig 0 0 0
  ```

  Use the **no** version to remove the specified PVC.

You can also include the **atm pvc** command in a base profile assigned to a dynamic ATM 1483 interface to apply encapsulation and traffic-shaping parameters to a bulk-configured range of PVCs. For information, see "Configuring Dynamic Interfaces Using Bulk Configuration" on page 619.

Related Documentation

- Upper-Layer Dynamic Interfaces over Static ATM Overview on page 527
- Monitoring Configuration Information of an ATM AAL5 Interface on page 593
- Monitoring Status or Summary Information for ATM Subinterfaces on page 595
- Monitoring Summary Information for ATM VCs and Reserved VC Ranges on page 600

- *atm pvc*

Dynamic PPP and PPPoE Interfaces over Static ATM

E Series routers support dynamic Point-to-Point Protocol (PPP) and Point-to-Point Protocol over Ethernet (PPPoE) interfaces over static Asynchronous Transfer Mode (ATM).

This topic describes the following:

- Dynamic PPP and PPPoE Interfaces over Static ATM Overview on page 538
- Configuring a PPP or PPPoE Dynamic Interface over an ATM 1483 Subinterface on page 539
- Overview of Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations on page 540

Dynamic PPP and PPPoE Interfaces over Static ATM Overview

E Series routers support dynamic PPP and PPPoE interfaces. The configuration procedure is very similar for each.

When using the **auto-configure** command, select only **ppp** or **pppoe**. The router automatically builds the necessary interfaces for you. When you indicate pppoe, on receipt of a PPPoE packet, the dynamic interface built is IP over PPP over PPPoE over ATM. Likewise, when you indicate **ppp**, the dynamic interface built is IP over PPP over ATM.

Figure 48 on page 538 shows dynamic PPP interface columns on ATM interfaces.

Figure 48: Dynamic PPP Interface Columns

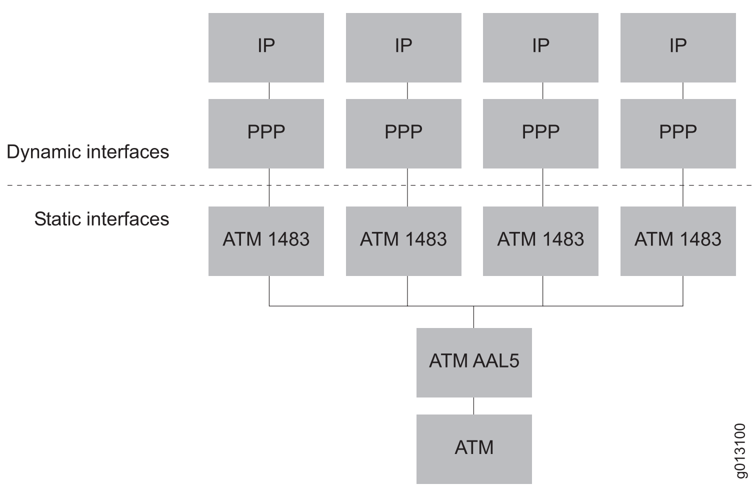
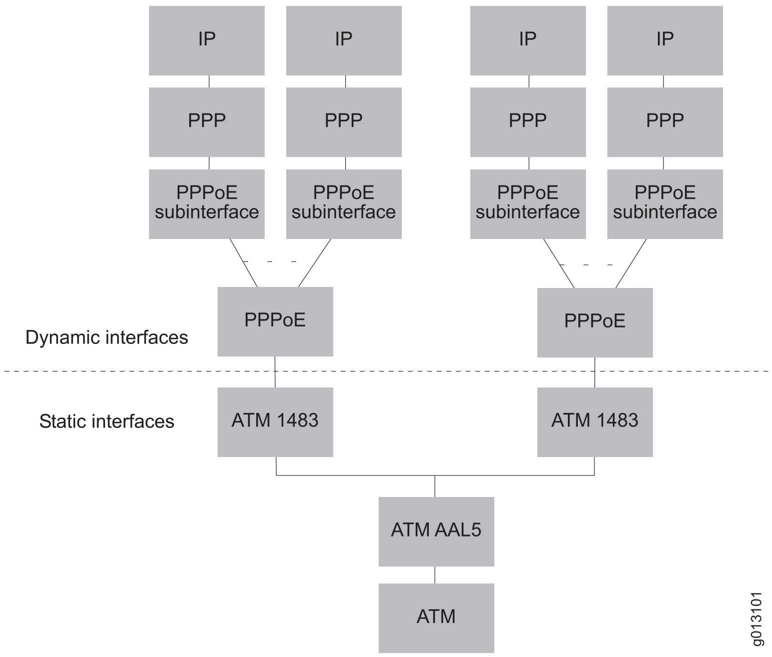


Figure 49 on page 539 shows dynamic PPPoE interface columns and illustrates how PPPoE supports multiple IP sessions over each ATM 1483 circuit.

Figure 49: Dynamic PPPoE Interface Columns



You can specify either or both **ppp** and **pppoe** for the interface by specifying the **auto-configure** command for each type of interface. The first packet received defines the type of dynamic interface that is created.

Configuring a PPP or PPPoE Dynamic Interface over an ATM 1483 Subinterface

To configure an ATM 1483 subinterface to support a PPP or PPPoE dynamic interface:

1. Configure a physical interface.

   host1(config)#**interface atm 5/0**

2. Configure an ATM 1483 subinterface.

   host1(config-if)#**interface atm 5/0.1**

3. Configure a PVC by specifying the virtual circuit descriptor, the virtual path identifier, the virtual channel identifier, and the encapsulation type. For more information, see "Creating a PVC on an ATM 1483 Subinterface" on page 537.

   host1(config-subif)#**atm pvc 10 10 22 aal5snap**

   If you want the router to autodetect the encapsulation type, use the **aal5autoconfig** option.

   host1(config-subif)#**atm pvc 10 10 22 aal5autoconfig**

4. Assign a profile to the PPP or PPPoE encapsulation types.

   host1(config-subif)#**profile ppp foo**
   host1(config-subif)#**profile pppoe foo**

5. Configure the subinterface to detect and accept dynamic PPP or PPPoE. For more information, see “Configuring a Dynamic Interface over an ATM 1483 Subinterface” on page 530.

   host1(config-subif)#**auto-configure ppp**
   host1(config-subif)#**auto-configure pppoe**

   In addition to **ppp** and **pppoe**, you can also specify **ip** or **bridgedEthernet**.

6. (Optional) Verify your configuration. For more information, see “Monitoring Status or Summary Information for ATM Subinterfaces” on page 595.

   host1#**show atm subinterface atm 5/0.1**

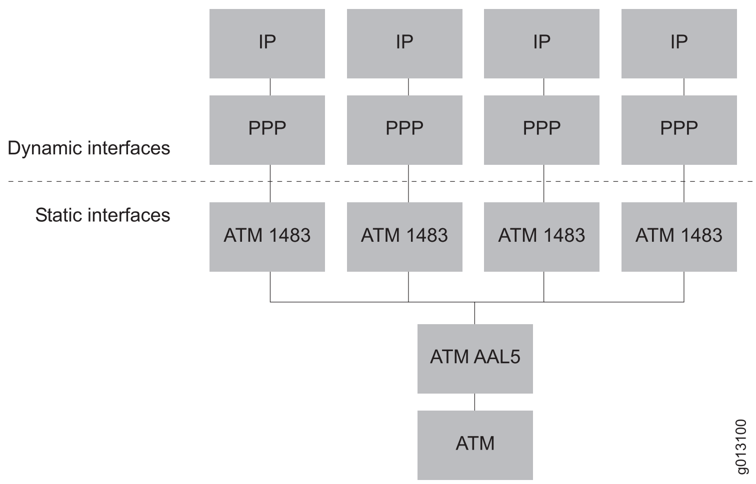
```
                               Circuit                                  Interface
Interface ATM-Prot VCD VPI VCI  Type   Encap MTU      Status         Type
--------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 5/0.1 RFC-1483  10  10  22 PVC     SNAP  9180 lowerLayerDown Static

Auto configure status         : dynamic
Auto configure interface(s)   : PPP  PPPoE
Detected 1483 encapsulation   : none
Detected dynamic interface    : none
Interface types in lockout    : none

Lockout state (seconds)       : Min Max Current Elapsed Next
------------------------------- --- --- ------- ------- ----
PPP                              1 300       0       0    1
PPPoE                            1 300       0       0    1

Assigned profile (IP)         : none assigned
Assigned profile (BridgedEnet): none assigned
Assigned profile (PPP)        : foo
Assigned profile (PPPoE)      : foo
Assigned profile (any)        : none assigned

SNMP trap link-status: disabled
Assigned VC Class: none assigned

InPackets:                 0
InBytes:                   0
OutPackets:                0
OutBytes:                  0
InErrors:                  0
OutErrors:                 0
InPacketDiscards:          0
InPacketsUnknownProtocol:  0
OutDiscards:               0
InPolicedPackets:          0
OutPolicedPackets:         0
1 interface(s) found
```

Overview of Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations

In configurations of dynamic IP over dynamic PPP over a static ATM 1483 subinterface, as shown in Figure 50 on page 541, any of the following conditions might cause the static ATM 1483 subinterface to transition to a dormant state as the result of an ungraceful subscriber logout:

- Rebooting the router
- Rebooting a line module
- Transitioning the physical (for example, SONET) interface, ATM major interface, or ATM AAL5 interface from up to down to up again
- Transitioning the ATM 1483 subinterface or the ATM permanent virtual connection (PVC) from up to down to up again
- Any other lowerLayerDown operational status condition that affects the dynamic PPP interface; a lowerLayerDown status indicates that a lower-layer interface below the dynamic PPP interface is down

Figure 50: Dynamic PPP Interface Columns



When the ATM 1483 subinterface transitions to a dormant state as a result of any of these conditions, the router tears down the dynamic PPP interface column. The dynamic PPP interface is unable to send an Link Control Protocol (LCP) terminate request to its peer because its own lower-layer interface is down. This action causes a loss of connectivity between the router and the Point-to-Point Protocol over ATM (PPPoA) customer premises equipment (CPE). If the CPE supports the PPP keepalive feature, it can detect the loss of connectivity and restart LCP negotiations in order to initiate a new connection. However, if the CPE does not support PPP keepalive, it cannot detect that the connection is down, and continues to send PPP data packets to the router.

On receipt of an IPv4-over-PPP data packet or an IPv6-over-PPP data packet from the CPE when the ATM 1483 subinterface transitions to a dormant state, the router sends an LCP terminate request packet to the CPE. Receipt of the LCP terminate request packet causes the CPE to restart LCP negotiations in order to initiate a new connection. After the CPE restarts LCP negotiations, the router recreates the dynamic PPP and IP upper-layer interfaces above the static ATM 1483 subinterface. This behavior is always in effect on the router and does not require command-line interface (CLI) or SNMP configuration.

Sending an LCP terminate request packet in response to receipt of an IPv4-over-PPP data packet or an IPv6-over-PPP data packet from a PPPoA CPE device offers the following benefits:

- For CPEs that support PPP keepalive, receipt of an LCP terminate request packet from the router restarts the LCP negotiations more quickly.
- For CPEs that do not support PPP keepalive, receipt of an LCP terminate request packet from the router enables the CPE to detect the connection termination and restart LCP negotiations in response.

The router also sends an LCP terminate request packet to a PPPoA CPE device in configurations of dynamic IP over dynamic PPP over a dynamic (bulk-configured) ATM 1483 subinterface. For more information, see "Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations Overview" on page 634 in "Configuring Dynamic Interfaces Using Bulk Configuration" on page 619.

Related Documentation

- Configuring a Dynamic Interface over an ATM 1483 Subinterface on page 530
- Dynamic Interface Configuration Using a Profile on page 565
- Configuring Profile Characteristics on page 576
- Monitoring Summary Information for ATM VCs and Reserved VC Ranges on page 600
- Monitoring Total Static and Dynamic Interface Counts for Interface Columns on page 602

- *atm pvc*
- *auto-configure*
- *interface atm*
- *profile*
- *show atm subinterface*

Dynamic PPPoE Interfaces over PPPoE Static Interfaces

E Series routers support dynamic Point-to-Point Protocol over Ethernet (PPPoE) subinterfaces over static PPPoE major interfaces. The PPPoE major interfaces can be created over:

- Asynchronous Transfer Mode (ATM)
- Ethernet
- Ethernet with virtual LANs (VLANs)
- Ethernet with service VLANs (S-VLANs)

The following sections describe how to create each of these configurations on the router:

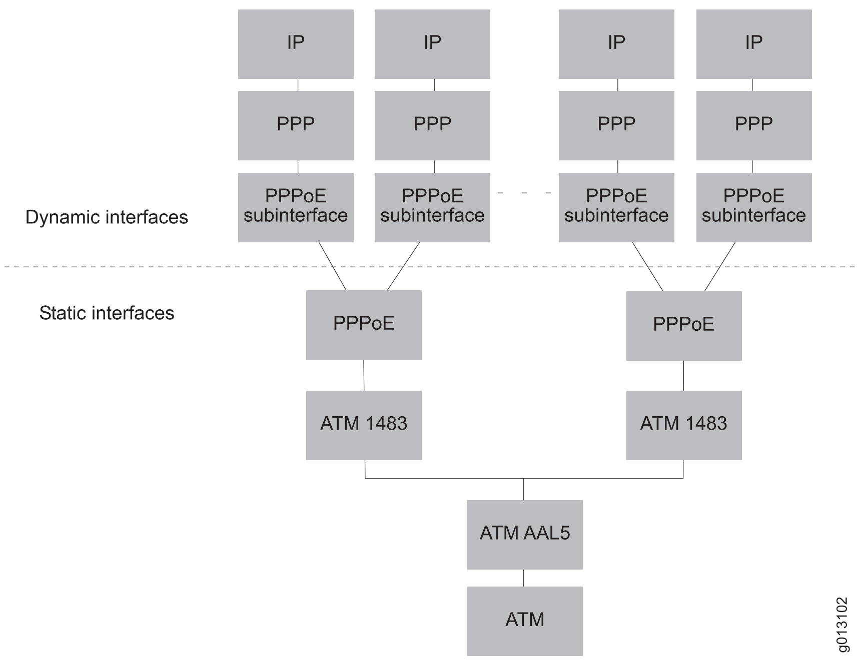
- Configuring Dynamic PPPoE over Static PPPoE with ATM Interface Columns on page 543
- Configuring Dynamic PPPoE over Static PPPoE with Ethernet Interface Columns on page 545

- Configuring Dynamic PPPoE over Static PPPoE with Ethernet and VLAN Interface Columns on page 546
- Configuring IPv4 and IPv6 Interface Columns over Static and Dynamic PPPoE on page 548
- Configuring Dynamic PPPoE over Static PPPoE with Ethernet and S-VLAN Interface Columns on page 549
- S-VLAN Oversubscription for Dynamic PPPoE Interfaces over Static PPPoE Overview on page 550
- Encapsulation Type Lockout for PPPoE Clients Overview on page 550
- Configuring Encapsulation Type Lockout for PPPoE Clients on page 552

Configuring Dynamic PPPoE over Static PPPoE with ATM Interface Columns

Figure 51 on page 543 shows dynamic PPPoE subinterface columns and illustrates an alternative method for PPPoE to support multiple IP sessions over each ATM 1483 circuit.

Figure 51: Dynamic PPPoE over Static PPPoE with ATM Interface Columns



To configure an ATM 1483 subinterface to support a dynamic PPPoE subinterface:

1. Configure a physical interface.

   host1(config)#interface atm 5/0

2. Configure an ATM 1483 subinterface.

   host1(config-if)#interface atm 5/0.1

3. Configure a PVC by specifying the virtual circuit descriptor, the virtual path identifier, the virtual channel identifier, and the encapsulation type. For more information, see "Creating a PVC on an ATM 1483 Subinterface" on page 537.

   host1(config-subif)#atm pvc 10 10 22 aal5snap

If you want the router to autodetect the encapsulation type, use the **aal5autoconfig** option.

host1(config-subif)#**atm pvc 10 10 22 aal5autoconfig**

4. Set the encapsulation type to PPPoE to create the PPPoE major interface.

   host1(config-subif)#**encapsulation pppoe**

5. Assign a profile.

   host1(config-subif)#**pppoe profile pppoeProfile1**

   The default encapsulation type, **any**, applies to any autoconfigured encapsulation that does not have a specific profile assignment.

6. Configure the interface to detect and accept dynamic PPPoE subinterfaces.

   host1(config-subif)#**pppoe auto-configure**

   When you configure dynamic encapsulation type lockout for PPPoE sessions that contain the IWF-Session DSL Forum VSA (26-254) in the PPPoE active discovery request (PADR) packets, even if PPPoE sessions associated with a particular MAC address are locked out, other PPPoE sessions that originated with the same MAC address are not terminated (continue to remain logged in) if they are IWF sessions from different access loops (PPPoE clients) and this information is available to the B-RAS application.

7. (Optional) Verify your configuration. For more information, see "Monitoring Status or Summary Information for ATM Subinterfaces" on page 595.

   host1#**show atm subinterface atm 5/0.1**

```
                               Circuit                                  Interface
Interface ATM-Prot VCD VPI VCI  Type   Encap MTU      Status         Type
--------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 5/0.1 RFC-1483  10  10  22 PVC     SNAP  9180 lowerLayerDown Static

Auto configure status         : static
Auto configure interface(s)   : none
Detected 1483 encapsulation   : none
Detected dynamic interface    : none
Interface types in lockout    : none

Assigned profile (IP)         : none assigned
Assigned profile (BridgedEnet): none assigned
Assigned profile (PPP)        : none assigned
Assigned profile (PPPoE)      : none assigned
Assigned profile (any)        : none assigned

SNMP trap link-status: disabled
Assigned VC Class: none assigned

InPackets:                0
InBytes:                  0
OutPackets:               0
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         0
InPacketsUnknownProtocol: 0
OutDiscards:              0
```

```
InPolicedPackets:         0
OutPolicedPackets:        0
1 interface(s) found
```

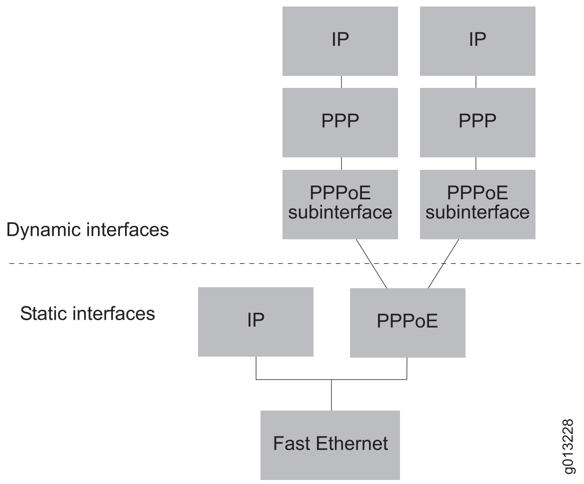
host1#**show pppoe interface atm 5/0.1**

```
PPPoE interface ATM 5/0.1 is operStatusLowerLayerDown
    PPPoE interface ATM 5/0.1 has max sessions = 8000
    PPPoE interface ATM 5/0.1 MTU 1494
    PPPoE interface ATM 5/0.1 has no acName set
    PPPoE interface ATM 5/0.1 autoconfigured subinterfaces
    PPPoE interface ATM 5/0.1 has 0 active connections,
        out of 0 configured subinterfaces
Assigned profile (any)        : pppoeProfile1

PPPoE Statistics
  Counters:
    PADI received       0
    PADI transmitted    0
    PADO received       0
    PADO transmitted    0
    PADR received       0
    PADR transmitted    0
    PADS received       0
    PADS transmitted    0
    PADT received       0
    PADT transmitted    0
    PADM received       0
    PADM transmitted    0
    PADN received       0
    PADN transmitted    0
  PAD packets received        0
  PAD packets transmitted     0

  Invalid PAD Packets:
    Invalid Version           0
    Invalid PAD Code          0
    Invalid PAD Tags          0
    Invalid PAD Tag length    0
    Invalid PAD Type          0
    Invalid PADI Session      0
    Invalid PADR Session      0
    Invalid PAD packet length 0
    Invalid PAD packets       0
   Total Invalid PAD packets  0
   Ingress Policed Packets 0
   Egress Policed Packets 0

  Insufficient Resources 0
```

Configuring Dynamic PPPoE over Static PPPoE with Ethernet Interface Columns

Figure 52 on page 546 shows dynamic PPPoE subinterface columns configured over an Ethernet interface without VLANs.

Figure 52: Dynamic PPPoE over Static PPPoE with Non-VLAN Interface Columns



To configure an Ethernet interface without VLANs to support a dynamic PPPoE subinterface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface.

   host1(config)#**interface fastEthernet 4/1**

2. Assign an IP address and mask.

   host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

3. Specify PPPoE as the encapsulation method on the interface.

   host1(config-subif)#**encapsulation pppoe**

   This command creates the static PPPoE major interface.

4. Assign a profile to the PPPoE major interface.

   host1(config-subif)#**pppoe profile pppoeProfile3**

   The default encapsulation type, **any**, applies to any autoconfigured encapsulation that does not have a specific profile assignment.

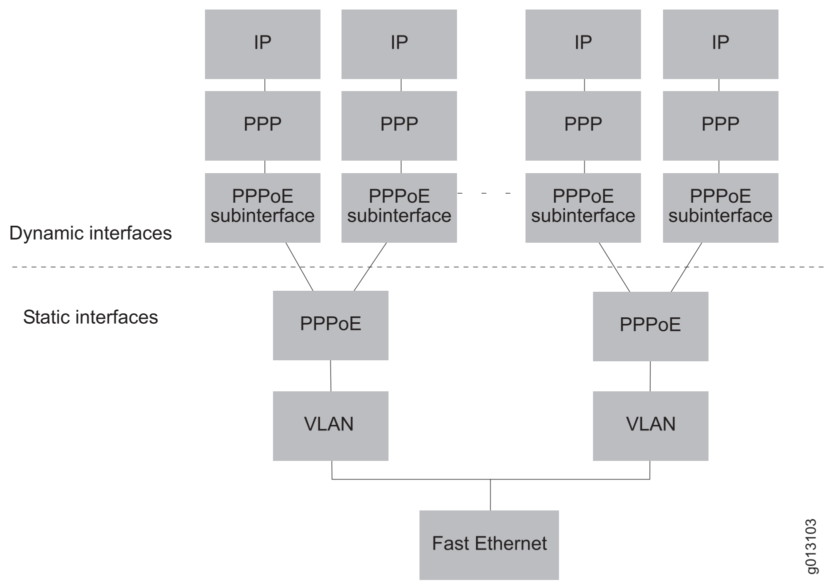
5. Configure the interface to detect and accept dynamic PPPoE subinterfaces.

   host1(config-subif)#**pppoe auto-configure**

Configuring Dynamic PPPoE over Static PPPoE with Ethernet and VLAN Interface Columns

Figure 53 on page 547 shows dynamic PPPoE subinterface columns and illustrates an alternative method for PPPoE to support multiple IP sessions over each VLAN.

Figure 53: Dynamic PPPoE over Static PPPoE with VLAN Interface Columns



To configure a VLAN subinterface to support a dynamic PPPoE subinterface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface.

   host1(config)#**interface fastEthernet 4/1**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   This command adds the VLAN major interface.

3. Create a VLAN subinterface by adding a subinterface number to the interface identifier.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Assign a VLAN ID for the subinterface.

   host1(config-if)#**vlan id 400**

5. Set the encapsulation type to PPPoE.

   host1(config-subif)#**encapsulation pppoe**

6. Assign a profile.

   host1(config-subif)#**pppoe profile pppoeProfile2**

   The default encapsulation type, **any**, applies to any autoconfigured encapsulation that does not have a specific profile assignment.

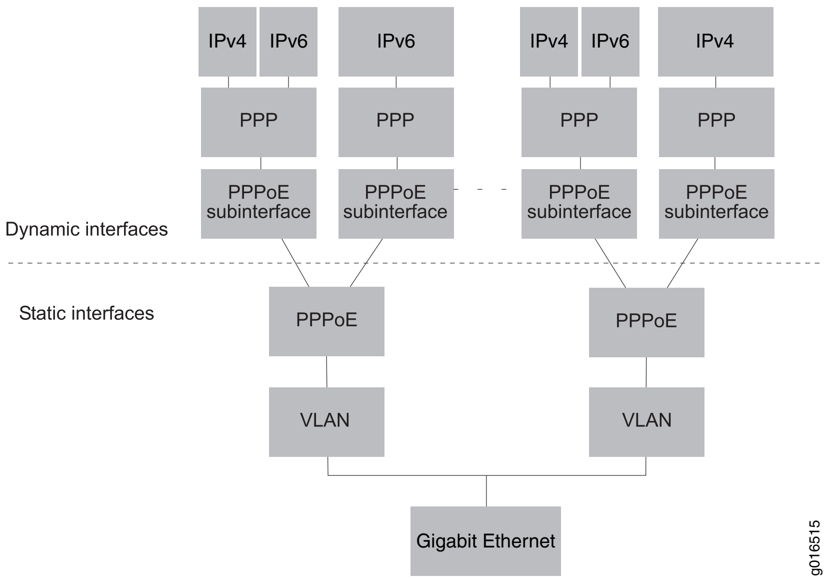
7. Configure the interface to detect and accept dynamic PPPoE subinterfaces.

   host1(config-subif)#**pppoe auto-configure**

Configuring IPv4 and IPv6 Interface Columns over Static and Dynamic PPPoE

You can configure IPv4 and IPv6 interface columns over static and dynamic PPPoE, as shown in Figure 54 on page 548.

Figure 54: IPv4 and IPv6 Interface Columns over Static and Dynamic PPPoE



To configure IPv4 and IPv6 interface columns over dynamic PPPoE:

1. Specify the loopback mode for an interface and assign both IPv6 and IPv4 addresses to the interface.

   host1(config)#**interface loopback 1**
   host1(config-if)#**ipv6 address 2000::1/64**
   host1(config-if)#**ip address 2.2.2.1/24**

2. Create a profile that defines attributes for the dynamic interface. You can use this profile to configure IPv4 or IPv6, or both IPv4 and IPv6 PPP interfaces.

   host1(config)#**profile ipv4Ipv6Profile**
   host1(config-profile)#**ip virtual-router ppp**
   host1(config-profile)#**ip unnumbered loopback 1**
   host1(config-profile)#**ipv6 virtual-router ppp**
   host1(config-profile)#**ipv6 unnumbered loopback 1**
   host1(config-profile)#**ipv6 nd other-config-flag**
   host1(config-profile)#**ipv6 nd ra-interval 10**
   host1(config-profile)#**ppp authentication chap**
   host1(config-profile)#**exit**

3. Specify the interface.

   host1(config-if)#**interface gigabitEthernet 4/0/6**

4. Set the encapsulation type to PPPoE.

   host1(config-subif)#**pppoe**

5. Configure the interface to detect and accept dynamic PPPoE subinterfaces.

   host1(config-subif)#**pppoe auto-configure**

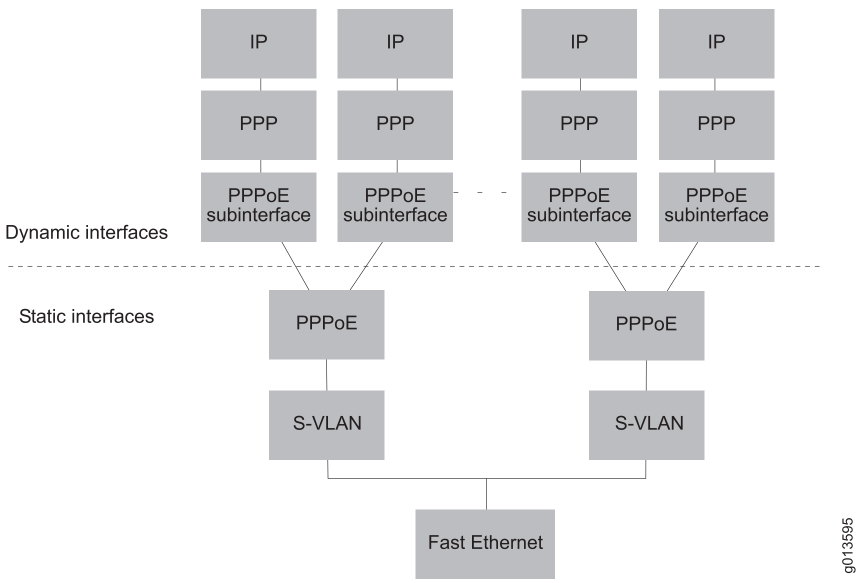
6. Assign the profile to any autoconfigured encapsulation.

   host1(config-if)#**pppoe profile any ipv4Ipv6Profile**

Configuring Dynamic PPPoE over Static PPPoE with Ethernet and S-VLAN Interface Columns

Figure 55 on page 549 shows dynamic PPPoE subinterface columns over PPPoE major interfaces using S-VLANs over Ethernet.

Figure 55: Dynamic PPPoE over Static PPPoE with S-VLAN Interface Columns



To configure an S-VLAN subinterface to support a dynamic PPPoE subinterface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface.

   host1(config)#**interface fastEthernet 4/1**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   This command creates the VLAN major interface.

3. Create a VLAN subinterface by adding a subinterface number to the interface identifier.

   host1(config-if)#**interface fastEthernet 3/1.1**

4. Assign an S-VLAN ID and a VLAN ID for the subinterface.

   host1(config-if)#**svlan id 3 300**

5. Assign an S-VLAN Ethertype.

   host1(config-if)#**svlan ethertype 9200**

6. Specify PPPoE as the encapsulation method on the interface.

   host1(config-subif)#**encapsulation pppoe**

   This command creates the PPPoE major interface.

7. Assign a profile.

   host1(config-subif)#**pppoe profile pppoeProfile3**

   The default encapsulation type, **any**, applies to any autoconfigured encapsulation that does not have a specific profile assignment.

8. Configure the interface to detect and accept dynamic PPPoE subinterfaces.

   host1(config-subif)#**pppoe auto-configure**

S-VLAN Oversubscription for Dynamic PPPoE Interfaces over Static PPPoE Overview

When you configure S-VLAN subinterfaces over Ethernet interfaces to support dynamic PPPoE subinterfaces, you can take advantage of S-VLAN oversubscription.

The maximum number of S-VLANs that you can create per I/O module or input/output adapter (IOA) with PPPoE major interfaces stacked over them is greater than the maximum number of dynamic PPPoE subinterfaces. The maximum number of Point-to-Point Protocol (PPP) interfaces supported per line module is directly proportional to the maximum number of PPPoE subinterfaces.

As a result, you can oversubscribe S-VLANs by configuring up to the maximum number of S-VLANs supported on the I/O module or IOA, knowing that no more than the maximum number of supported PPP sessions can be connected to the router at any one time.

For information about the module combinations that support S-VLAN oversubscription, see "S-VLAN Oversubscription" on page 179.

For specific information about the maximum number of S-VLANs supported per I/O module or IOA and the maximum number of PPP interfaces and PPPoE subinterfaces supported per line module, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

NOTE: S-VLAN oversubscription is not currently supported for S-VLANs configured over bridged Ethernet interfaces.

The E120 and E320 routers can support up to two IOAs per line module. This maximum number of S-VLANs per line module does not change whether one or two IOAs are installed. For more information about configuration options for the ES2-S1 GE-4 IOA, see *Configuring Ethernet Interfaces* in *JunosE Physical Layer Configuration Guide*.

Encapsulation Type Lockout for PPPoE Clients Overview

In configurations with dynamic PPPoE subinterfaces over static PPPoE major interfaces, you can configure dynamic encapsulation type lockout for the PPPoE clients associated with a dynamic PPPoE subinterface column. Using this feature enables you to temporarily prevent the static PPPoE major interface from autodetecting, accepting, and creating dynamic PPPoE subinterface columns for a configurable time period.

By default, encapsulation type lockout is disabled for PPPoE clients. To configure a lockout time range for the PPPoE clients associated with the dynamic PPPoE subinterface columns on the PPPoE major interface, use the **pppoe auto-configure** command with the **lockout-time** keyword. You can also use the **show pppoe interface lockout-time** command to display detailed information about the current lockout condition for each PPPoE client, and the **pppoe clear lockout interface** command to clear (reset) the lockout condition for an individual PPPoE client.

For information about the working of the dynamic encapsulation type lockout feature for PPPoE sessions that contain the interworking function (IWF)-Session digital subscriber line (DSL) Forum vendor-specific attribute (VSA) (26-254) in the PPPoE active discovery request (PADR) packets, see "Encapsulation Type Lockout Based on DSL Forum VSAs for IWF PPPoE Sessions" on page 534.

For illustrations of the interface stacking in dynamic PPPoE over static PPPoE configurations, see the figures provided in "Configuring Dynamic PPPoE over Static PPPoE with ATM Interface Columns" on page 543, "Configuring Dynamic PPPoE over Static PPPoE with Ethernet Interface Columns" on page 545, "Configuring Dynamic PPPoE over Static PPPoE with Ethernet and VLAN Interface Columns" on page 546, and "Configuring Dynamic PPPoE over Static PPPoE with Ethernet and S-VLAN Interface Columns" on page 549.

Differences from Lockout Configuration for PPPoE over Static ATM

Table 49 on page 551 lists the important differences between how encapsulation type lockout works for dynamic PPPoE over static PPPoE configurations and how lockout works for dynamic PPPoE over static ATM 1483 configurations.

Table 49: Differences in Lockout Operation for Dynamic PPPoE Configurations

Dynamic PPPoE over Static PPPoE	Dynamic PPPoE over Static ATM 1483
Encapsulation type lockout is disabled by default.	Encapsulation type lockout is enabled by default with a lockout time range of 1–300 seconds.
You must explicitly configure encapsulation type lockout for PPPoE clients with the **pppoe auto-configure** command.	PPPoE clients automatically inherit their lockout setting from the lockout parameters configured for the underlying static ATM 1483 subinterface with the **auto-configure** command. Currently, the dynamic PPPoE interface layer must be configured directly above the static ATM 1483 interface layer to support inheritance of lockout parameters. For an illustration of dynamic PPPoE over static ATM 1483 interface stacking, see the figure *Dynamic PPPoE Interface Columns* in "Dynamic PPP and PPPoE Interfaces over Static ATM Overview" on page 538.

For more information about the benefits and operation of dynamic encapsulation type lockout, see "Dynamic Encapsulation Type Lockout" on page 532. In particular, see

"Guidelines for Configuring Encapsulation Type Lockout for PPPoE Sessions" on page 535 for information about the rules that apply when you configure the lockout time. These rules are common to both dynamic PPPoE over static PPPoE configurations and dynamic PPPoE over static ATM 1483 configurations.

Configuring Encapsulation Type Lockout for PPPoE Clients

Configuring dynamic encapsulation type lockout for PPPoE clients includes the following tasks:

- Configuring and Verifying Lockout for PPPoE Clients on page 552
- Clearing the Lockout Condition for a PPPoE Client on page 553

Configuring and Verifying Lockout for PPPoE Clients

To configure and verify encapsulation type lockout for a PPPoE client:

1. Configure the underlying physical interface.

   For example, the following commands configure a static ATM 1483 subinterface and corresponding ATM PVC. For more information about the **atm pvc** command, see "Creating a PVC on an ATM 1483 Subinterface" on page 537

   ```
   host1(config)#interface atm 3/0
   host1(config-if)#interface atm 3/0.101
   host1(config-subif)#atm pvc 10 10 20 aal5snap
   ```

2. Create a static PPPoE major interface.

   ```
   host1(config-subif)#encapsulation pppoe
   ```

3. Configure the PPPoE major interface to detect and accept dynamic PPPoE subinterfaces. Use the **lockout-time** keyword to configure a nondefault lockout time range for the PPPoE clients associated with the dynamic PPPoE subinterface column.

   For example, the following command configures a lockout time in the range 5–60 seconds for the PPPoE clients associated with the dynamic PPPoE subinterface column on the PPPoE major interface.

   ```
   host1(config-subif)#pppoe auto-configure lockout-time 5 60
   ```

4. Assign a profile to the PPPoE major interface.

   ```
   host1(config-subif)#pppoe profile pppoeLockoutProfile
   ```

   The default encapsulation type, **any**, applies to any autoconfigured encapsulation that does not have a specific profile assignment.

   For information about creating and using profiles, see "Dynamic Interface Configuration Using a Profile" on page 565 and "Configuring Profile Characteristics" on page 576.

5. (Optional) Verify the lockout configuration by using either of the following commands.

   - To display summary information about the lockout configuration, use the **show pppoe interface** command. (The following example shows only the portion of the command display relevant to the PPPoE lockout configuration.)

```
host1#show pppoe interface atm 3/0.101
PPPoE interface ATM 3/0.101 is operStatusUp (dynamic)
. . .
Lockout Configuration (seconds): Min 5, Max 60
 Total clients in active lockouts: 0
 Total clients in lockout grace period: 0
```

- To display detailed information about the current lockout condition for each PPPoE client associated with a specific source MAC address, use the **show pppoe interface lockout-time** command.

  This command displays multiple entries for the same MAC address if multiple IWF sessions contain the same MAC address. In the following example, more than one entry for the same PPPoE client MAC address, 0090.1a42.527c, is displayed under the Client Address column head. This method of display occurs because the MAC address in the Client Address field denotes the MAC address of the DSLAM device at which multiplexing functions are performed and not the address of the originating PPPoE client (access loop) for PPPoE sessions that contain the IWF-Session DSL Forum VSA (26-154).

```
host1#show pppoe interface atm 12/1/1.1.1 lockout-time
PPPoE interface atm 12/1/1.1
Lockout Configuration (seconds): Min 90, Max 120
 Total clients in active lockout: 1
 Total clients in lockout grace period: 0
Client Address Current Elapsed Next
-------------- ------- ------- ----
0090.1a42.527c     120      30  120
0090.1a42.527c       0       0   90
```

  For a description of the fields in the command display, see “Monitoring Summary Information About the Encapsulation Type Lockout for PPPoE Clients” on page 603 and “Monitoring Detailed Information About the Current Encapsulation Type Lockout Condition for PPPoE Clients” on page 604.

Clearing the Lockout Condition for a PPPoE Client

You can use the **pppoe clear lockout interface** command to clear the lockout condition for an individual PPPoE client associated with a dynamic PPPoE subinterface column on a static PPPoE major interface. To identify the PPPoE client, you must specify its source MAC address.

For information about the working of the dynamic encapsulation type lockout feature for PPPoE sessions that contain the IWF-Session DSL Forum VSA (26-254) in PADR packets, see “Encapsulation Type Lockout Based on DSL Forum VSAs for IWF PPPoE Sessions” on page 534.

NOTE: Issuing the **pppoe clear lockout interface** command resets the current lockout condition for the specified PPPoE client, but does *not* disable dynamic encapsulation type lockout for that PPPoE client.

To clear the current lockout condition for a PPPoE client:

1. Display the source MAC address assigned to the PPPoE client by issuing one of the following **show** commands:

   - To display the source MAC address when there is no available PPPoE session in progress, use the **show pppoe interface lockout-time** command.

     This command displays multiple entries for the same MAC address if multiple IWF sessions contain the same MAC address. In the following example, more than one entry for the same PPPoE client MAC address, 0090.1a42.527c, is displayed under the Client Address column head. This method of display occurs because the MAC address in the Client Address field denotes the MAC address of the DSLAM device at which multiplexing functions are performed and not the address of the originating PPPoE client (access loop) for PPPoE sessions that contain the IWF-Session DSL Forum VSA (26-154).

     ```
     host1#show pppoe interface atm 12/1/1.1.1 lockout-time
     PPPoE interface atm 12/1/1.1
     Lockout Configuration (seconds): Min 90, Max 120
      Total clients in active lockout: 1
      Total clients in lockout grace period: 0
     Client Address Current Elapsed Next
     -------------- ------- ------- ----
     0090.1a42.527c     120      30  120
     0090.1a42.527c       0       0   90
     ```

   - To display the source MAC address when a subscriber is connected to the router through an available PPPoE session, use either the **show pppoe interface lockout-time** command or the **show pppoe subinterface full** command. (The following example shows only the portion of the command display relevant to the source MAC address.)

     ```
     host1#show pppoe subinterface full
     ...
         PPPoE subinterface ATM 3/0.101 has source MAC address 0090.1a10.165e
     ...
     ```

   For a description of the fields in the command display, see “Monitoring Detailed Information About the Current Encapsulation Type Lockout Condition for PPPoE Clients” on page 604 and “Monitoring the Source MAC Address of a PPPoE Client” on page 605.

2. Clear the current lockout condition for the PPPoE client associated with the specified source MAC address on the static PPPoE major interface.

   ```
   host1#pppoe clear lockout interface atm 3/0.101 0090.1a10.165e
   ```

   If the specified PPPoE client is undergoing active lockout or is in a lockout grace period, issuing the **pppoe clear lockout interface** command causes the router to reset the current lockout condition and start the next lockout interval at the minimum configured lockout time.

   The lockout grace period occurs when the time between lockout events is greater than either 15 minutes or the maximum lockout time. When a PPPoE client is in a lockout grace period, the router resets the number of consecutive lockout events to 1. (For more information, see “Guidelines for Configuring Encapsulation Type Lockout for PPPoE Sessions” on page 535.)

Related Documentation

- Dynamic Encapsulation Type Lockout on page 532
- Dynamic Interface Configuration Using a Profile on page 565
- Configuring Profile Characteristics on page 576
- Monitoring Summary Information for ATM VCs and Reserved VC Ranges on page 600
- Monitoring Total Static and Dynamic Interface Counts for Interface Columns on page 602
- Monitoring Summary Information About the Encapsulation Type Lockout for PPPoE Clients on page 603
- Monitoring Detailed Information About the Current Encapsulation Type Lockout Condition for PPPoE Clients on page 604
- Monitoring the Source MAC Address of a PPPoE Client on page 605

- *atm pvc*
- *encapsulation pppoe*
- *encapsulation vlan*
- *interface atm*
- *interface fastEthernet*
- *interface gigabitEthernet*
- *interface loopback*
- *ip address*
- *ip unnumbered*
- *ip virtual-router*
- *ipv6 address*
- *ipv6 nd other-config-flag*
- *ipv6 nd ra-interval*
- *ipv6 unnumbered*
- *ipv6 virtual-router*
- *ppp authentication*
- *pppoe*
- *pppoe auto-configure*
- *pppoe clear lockout interface*
- *pppoe profile*
- *profile*
- *show atm subinterface*
- *show pppoe interface*
- *show pppoe interface lockout-time*

- *show pppoe subinterface*
- *svlan ethertype*
- *svlan id*
- *vlan id*

Dynamic IPoA Interfaces Overview

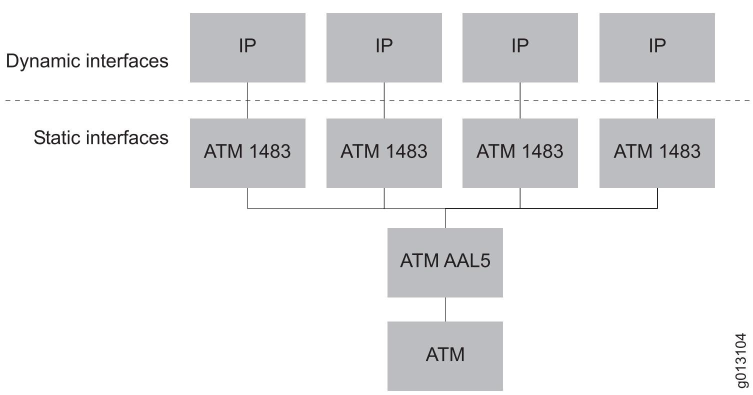
E Series routers support dynamic IP over ATM (IPoA) interfaces. An IPoA interface is IP over Asynchronous Transfer Mode (ATM) 1483 over ATM AAL5 over ATM. See the figure *Configuring an ATM 1483 Interface to Support Dynamic Interfaces* provided in "Upper-Layer Dynamic Interfaces over Static ATM Overview" on page 527.

An IPoA configuration is typically used as a high-speed access service or uplink to another router. A common use is to provision IP over ATM circuits over digital subscriber line (DSL) to connect businesses to the Internet—a Broadband Remote Access Server (B-RAS) alternative to circuit aggregation. All provisioning can be through the RADIUS server to minimize any configuration of the router.

When IP packets are received over ATM circuits, the IP interfaces are dynamically constructed over the corresponding ATM 1483 interfaces from the configuration data received from the RADIUS server, a profile, or both.

Figure 56 on page 556 shows the protocol layers that represent the IPoA interface columns, and the layers within the interface columns that are static and dynamic.

Figure 56: Dynamic IPoA over Static ATM 1483 Interface Columns



When you configure dynamic IPoA interfaces, you must assign a profile. Optionally, you can also assign a subscriber identification.

Related Documentation

- Configuring a Dynamic IPoA Interface on page 556

Configuring a Dynamic IPoA Interface

To configure dynamic IPoA interfaces:

1. Configure a physical interface.

   host1(config)#**interface atm 5/0**

2. Configure an ATM subinterface.

   host1(config-if)#**interface atm 5/0.1**

3. Configure a PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type.

   host1(config-subif)#**atm pvc 10 10 22 aal5snap**

   If you want the router to autodetect the encapsulation type, use the **aal5autoconfig** option. For more information, see "Creating a PVC on an ATM 1483 Subinterface" on page 537.

   host1(config-subif)#**atm pvc 10 10 22 aal5autoconfig**

4. Assign a profile. For more information, see "Dynamic Interface Configuration Using a Profile" on page 565.

   host1(config-subif)#**profile ip foo**

5. (Optional) Assign subscriber identification. For more information, see "Configuring a Local Subscriber for a Dynamic IPoA or Bridged Ethernet Interface" on page 528.

   host1(config-subif)#**subscriber ip user charlie domain myispname password lucy**

6. Do either of the following:

   - Configure the subinterface to detect and accept the dynamic IP encapsulation type using the default lockout time range, 1–300 seconds. For more information, see "Configuring a Dynamic Interface over an ATM 1483 Subinterface" on page 530.

     host1(config-subif)#**auto-configure ip**

   - Configure the subinterface to detect and accept the dynamic IP encapsulation type using a nondefault lockout time range. For example, the following command configures 3600 seconds (1 hour) as the minimum lockout time and 7200 seconds (2 hours) as the maximum lockout time.

     host1(config-subif)#**auto-configure ip lockout-time 3600 7200**

7. (Optional) Verify your configuration.

   host1#**show atm subinterface atm 5/0.1**

```
                               Circuit                          Interface
Interface ATM-Prot VCD VPI VCI  Type   Encap MTU      Status       Type
--------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 5/0.1 RFC-1483  10  10  22 PVC     SNAP  9180 lowerLayerDown Static

Auto configure status         : dynamic
Auto configure interface(s)   : IP
Detected 1483 encapsulation   : none
Detected dynamic interface    : none
Interface types in lockout    : none

Lockout state (seconds)       : Min  Max  Current Elapsed Next
------------------------------- ---- ---- ------- ------- ----
IP                              3600 7200       0       0 3600

Assigned profile (IP)         : foo
```

```
Assigned profile (BridgedEnet): none assigned
Assigned profile (PPP)        : none assigned
Assigned profile (PPPoE)      : none assigned
Assigned profile (any)        : none assigned

IP subscriber info            :
Username: charlie@myispname
Password: lucy
Authenticate: enabled

SNMP trap link-status: disabled
Assigned VC Class: none assigned

InPackets:                 0
InBytes:                   0
OutPackets:                0
OutBytes:                  0
InErrors:                  0
OutErrors:                 0
InPacketDiscards:          0
InPacketsUnknownProtocol:  0
OutDiscards:               0
InPolicedPackets:          0
OutPolicedPackets:         0
1 interface(s) found
```

Related Documentation

- Dynamic IPoA Interfaces Overview on page 556
- Configuring IPv4 Characteristics for a Profile on page 577
- Monitoring Status or Summary Information for ATM Subinterfaces on page 595
- Monitoring Summary Information for ATM VCs and Reserved VC Ranges on page 600

- *atm pvc*
- *auto-configure*
- *interface atm*
- *profile*
- *show atm subinterface*
- *subscriber*

Dynamic Bridged Ethernet Interfaces

This topic describes the following:

- Dynamic Bridged Ethernet Interfaces Overview on page 559
- Configuring a Dynamic Bridged Ethernet Interface on page 559
- Configuring Subscriber Management for IP Subscribers on Dynamic Bridged Ethernet Interfaces on page 561

- Example: Configuring IP Subscribers on a Dynamic Bridged Ethernet Interface Using the subscriber Command on page 562
- Example: Configuring IP Subscribers on a Dynamic Bridged Ethernet Interface Using IP Subscriber Management on page 563

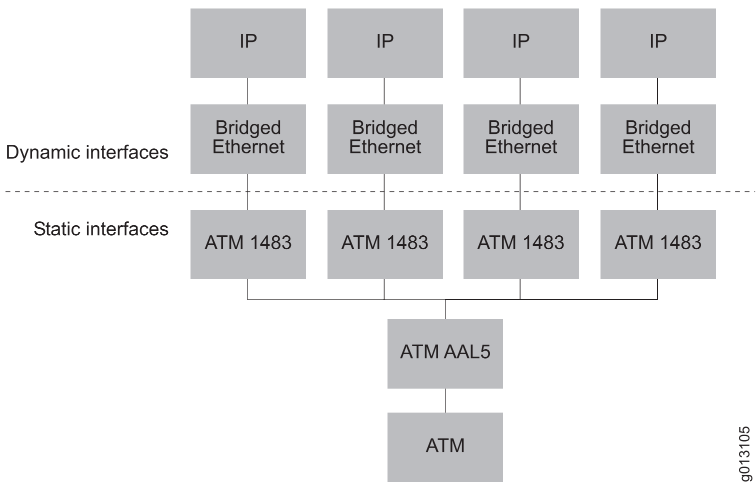
Dynamic Bridged Ethernet Interfaces Overview

A bridged Ethernet interface is IP over bridged Ethernet over Asynchronous Transfer Mode (ATM) 1483 over ATM AAL5 over ATM.

When bridged Ethernet packets are received over ATM circuits, the bridged Ethernet and IP interfaces are dynamically constructed over the corresponding ATM 1483 interfaces and use the configuration data received from the RADIUS server, a profile, or both.

Figure 57 on page 559 shows the protocol layers that represent the bridged Ethernet interface columns, and the layers within the interface columns that are static and dynamic.

Figure 57: Dynamic Bridged Ethernet over Static ATM 1483 Interface Columns



Configuring a Dynamic Bridged Ethernet Interface

When you configure dynamic bridged Ethernet interfaces, you must assign a profile. You may optionally assign a subscriber identification.

To configure dynamic bridged Ethernet interfaces:

1. Configure a physical interface.

   host1(config)#interface atm 5/0

2. Configure an ATM subinterface.

   host1(config-if)#interface atm 5/0.1

3. Configure a PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type. For more information, see "Creating a PVC on an ATM 1483 Subinterface" on page 537.

   host1(config-subif)#atm pvc 10 10 22 aal5snap

If you want the router to autodetect the encapsulation type, use the **aal5autoconfig** option.

host1(config-subif)#**atm pvc 10 10 22 aal5autoconfig**

4. Do either of the following:

   - Configure the subinterface to detect and accept the dynamic bridged Ethernet encapsulation type with the default lockout time range, 1–300 seconds. For more information, see "Configuring a Dynamic Interface over an ATM 1483 Subinterface" on page 530.

     host1(config-subif)#**auto-configure bridgedEthernet**

   - Configure the subinterface to detect and accept the dynamic bridged Ethernet encapsulation type with a nondefault lockout time range. For example, the following command configures 3600 seconds (1 hour) as the minimum lockout time and 7200 seconds (2 hours) as the maximum lockout time.

     host1(config-subif)#**auto-configure bridgedEthernet lockout-time 3600 7200**

5. Assign a profile to match the encapsulation type of bridged Ethernet. For more information, see "Dynamic Interface Configuration Using a Profile" on page 565.

   host1(config-subif)#**profile bridgedEthernet foo**

6. (Optional) Assign subscriber identification. For more information, see "Configuring a Local Subscriber for a Dynamic IPoA or Bridged Ethernet Interface" on page 528.

   host1(config-subif)#**subscriber bridgedEthernet user charlie domain myisp password lucy**

7. (Optional) Verify your configuration.

   host1#**show atm subinterface atm 5/0.1**

```
                              Circuit                            Interface
Interface ATM-Prot VCD VPI VCI  Type   Encap MTU      Status       Type
--------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 5/0.1 RFC-1483  10  10  22 PVC     SNAP  9180 lowerLayerDown Static

Auto configure status          : dynamic
Auto configure interface(s)    : bridgedEthernet
Detected 1483 encapsulation    : none
Detected dynamic interface     : none
Interface types in lockout     : none

Lockout state (seconds)        : Min Max Current Elapsed Next
------------------------------- --- --- ------- ------- ----
BridgedEthernet                    1 300       0       0    1

Assigned profile (IP)          : none assigned
Assigned profile (BridgedEnet): foo
Assigned profile (PPP)         : none assigned
Assigned profile (PPPoE)       : none assigned
Assigned profile (any)         : none assigned

BridgedEnet subscriber info    :
Username: charlie@myisp
Password: lucy
Authenticate: enabled
```



```
SNMP trap link-status: disabled
Assigned VC Class: none assigned

InPackets:                0
InBytes:                  0
OutPackets:               0
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         0
InPacketsUnknownProtocol: 0
OutDiscards:              0
InPolicedPackets:         0
OutPolicedPackets:        0
1 interface(s) found
```

Configuring Subscriber Management for IP Subscribers on Dynamic Bridged Ethernet Interfaces

You can use the JunosE subscriber management application to configure and manage IP subscribers associated with a dynamic bridged Ethernet over static ATM 1483 interface column, as described in "Subscriber Authentication on Dynamic Bridged Ethernet over Static ATM Interfaces" on page 528.

To use the subscriber management application to configure IP subscribers on a dynamic bridged Ethernet interface for RADIUS authentication:

1. Define an IP service profile that contains the subscriber's RADIUS authentication parameters including the username, domain, and password.
2. Configure the interface profile from which the router creates a dynamic bridged Ethernet interface column. For more information, see "Dynamic Interface Configuration Using a Profile" on page 565 and "Configuring Profile Characteristics" on page 576.
   a. Include the desired characteristics for the upper-layer encapsulation types.
   b. (Optional) Specify the name of the route map used to configure the IP subscriber interface.
   c. Use the **bridge1483 service-profile** command to assign the specified IP service profile to the interface profile. The IP service profile contains the RADIUS authentication parameters for subscribers on the dynamic bridged Ethernet interface.
3. Define the underlying static or dynamic ATM 1483 subinterface on which the dynamic bridged Ethernet interface column is built. For more information, see "Configuring a Dynamic Bridged Ethernet Interface" on page 559.
   a. Assign the specified interface profile to the ATM 1483 subinterface.
   b. Enable autodetection (autoconfiguration) of the bridged Ethernet upper-layer encapsulation type.

c. Define the ATM PVC over which data is transmitted.

4. (Optional) Use the **show profile** command to verify assignment of the IP service profile to the interface profile. For more information, see "Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces" on page 606.

Example: Configuring IP Subscribers on a Dynamic Bridged Ethernet Interface Using the subscriber Command

This example shows you how to use the **subscriber** command to configure RADIUS authentication for IP subscribers on a dynamic bridged Ethernet interface. This configuration method *does not support* running stateful SRP switchover on the router.

- Requirements on page 562
- Overview on page 562
- Configuring IP Subscribers on a Dynamic Bridged Ethernet Interface on page 562

Requirements

This example uses the following software and hardware components:

- JunosE Release 7.1.0 or higher-numbered releases
- E Series router (ERX7xx models, ERX14xx models, the ERX310 router, the E120 router, or the E320 router)
- ASIC-based line modules that support Fast Ethernet or Gigabit Ethernet

Overview

You can use the **subscriber** command to configure RADIUS authentication for IP subscribers on a dynamic bridged Ethernet interface. Assume that you have issued the following commands to configure IP subscribers on a dynamic bridged Ethernet interface for RADIUS authentication. In this configuration, the **subscriber** command provides the subscriber's authentication parameters, and the static ATM 1483 subinterface is the authenticating layer. Keep in mind that the **subscriber** command does not support running stateful SRP switchover on the router.

Configuring IP Subscribers on a Dynamic Bridged Ethernet Interface

Step-by-Step Procedure

To configure IP subscribers on a dynamic bridged Ethernet interface:

1. Configure the interface profile from which to create a dynamic bridged Ethernet interface. For more information, see "Dynamic Interface Configuration Using a Profile" on page 565.

   ```
   host1(config)#profile east
   ```

2. Include the desired attributes for the profile (in this case, IGMP). For more information, see "Configuring IPv4 Characteristics for a Profile" on page 577.

   ```
   host1(config-profile)#ip igmp
   host1(config-profile)#ip igmp immediate-leave
   host1(config-profile)#ip igmp group limit 6
   ```

3. (Optional) Configure the name of the route map used to configure the IP subscriber interface.

   ```
   host1(config-profile)#ip route-map ip-subscriber eastRouteMap
   host1(config-profile)#exit
   ```

4. Configure the static ATM 1483 subinterface to assign the east profile.

   ```
   host1(config)#interface atm 2/1.100 point-to-point
   host1(config-subif)#profile bridgedEthernet east
   ```

5. Configure a local subscriber for the static ATM 1483 subinterface to support RADIUS authentication. For more information, see "Configuring a Local Subscriber for a Dynamic IPoA or Bridged Ethernet Interface" on page 528.

   ```
   host1(config-subif)#subscriber bridgedEthernet user westford001 domain
     xyzcorp.east password abc123
   ```

6. Enable autodetection of the bridged Ethernet upper-layer encapsulation type. For more information, see "Configuring a Dynamic Interface over an ATM 1483 Subinterface" on page 530.

   ```
   host1(config-subif)#auto-configure bridgedEthernet
   ```

7. Define the ATM PVC. For more information, see "Creating a PVC on an ATM 1483 Subinterface" on page 537.

   ```
   host1(config-subif)#atm pvc 100 10 101 aal5snap 6400 0 0
   host1(config-subif)#exit
   ```

Example: Configuring IP Subscribers on a Dynamic Bridged Ethernet Interface Using IP Subscriber Management

This example shows you how to configure IP subscribers on a dynamic bridged Ethernet interface for RADIUS authentication using the IP subscriber management application. This configuration method uses the **bridge1483 service-profile** command to assign the specified IP service profile to the interface profile, and *does support* running stateful SRP switchover on the router.

- Requirements on page 563
- Overview on page 564
- Configuring IP Subscribers on a Dynamic Bridged Ethernet Interface on page 564

Requirements

This example uses the following software and hardware components:

- JunosE Release 7.1.0 or higher-numbered releases
- E Series router (ERX7xx models, ERX14xx models, the ERX310 router, the E120 router, or the E320 router)
- ASIC-based line modules that support Fast Ethernet or Gigabit Ethernet

Overview

In this configuration, the IP service profile provides the subscriber's authentication parameters, and the subscriber management application is the authenticating layer. To assign the IP service profile to the interface profile, use the **bridge1483 service-profile** command.

For more information about using the subscriber management application, see *JunosE Broadband Access Configuration Guide*.

Configuring IP Subscribers on a Dynamic Bridged Ethernet Interface

Step-by-Step Procedure

To configure IP subscribers on a dynamic bridged Ethernet interface:

1. Define an IP service profile.

   ```
   host1(config)#ip service-profile eastServiceProfile
   ```

2. Assign the subscriber's username, domain, and password to the IP service profile.

   ```
   host1(config-service-profile)#user-name westford001
   host1(config-service-profile)#domain xyzcorp.east
   host1(config-service-profile)#password abc123
   host1(config-service-profile)#exit
   ```

3. Configure the interface profile from which to create a dynamic bridged Ethernet interface. For more information, see "Dynamic Interface Configuration Using a Profile" on page 565.

   ```
   host1(config)#profile east
   ```

4. Include the desired attributes for the profile (in this case, IGMP). For more information, see "Configuring IPv4 Characteristics for a Profile" on page 577.

   ```
   host1(config-profile)#ip igmp
   host1(config-profile)#ip igmp immediate-leave
   host1(config-profile)#ip igmp group limit 6
   ```

5. (Optional) Configure the name of the route map used to configure the IP subscriber interface.

   ```
   host1(config-profile)#ip route-map ip-subscriber eastRouteMap
   ```

6. Configure the name of the IP service profile containing the authentication parameters for the dynamic bridged Ethernet interface.

   ```
   host1(config-profile)#bridge1483 service-profile eastServiceProfile
   host1(config-profile)#exit
   ```

7. Configure the static ATM 1483 subinterface to assign the east profile.

   ```
   host1(config)#interface atm 2/1.100 point-to-point
   host1(config-subif)#profile bridgedEthernet east
   ```

8. Enable autodetection of the bridged Ethernet upper-layer encapsulation type. For more information, see "Configuring a Dynamic Interface over an ATM 1483 Subinterface" on page 530

   ```
   host1(config-subif)#auto-configure bridgedEthernet
   ```

9. Define the ATM PVC. For more information, see "Creating a PVC on an ATM 1483 Subinterface" on page 537

```
host1(config-subif)#atm pvc 100 10 101 aal5snap 6400 0 0
host1(config-subif)#exit
```

Related Documentation

- Dynamic Interface Configuration Using a Profile on page 565
- Configuring Profile Characteristics on page 576
- Monitoring Status or Summary Information for ATM Subinterfaces on page 595
- Monitoring Summary Information for ATM VCs and Reserved VC Ranges on page 600
- Monitoring Total Static and Dynamic Interface Counts for Interface Columns on page 602
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606

- *atm pvc*
- *auto-configure*
- *bridge1483 service-profile*
- *domain*
- *interface atm*
- *ip igmp*
- *ip igmp group limit*
- *ip igmp immediate-leave*
- *ip route-map ip-subscriber*
- *ip service-profile*
- *password*
- *profile*
- *show atm subinterface*
- *show profile*
- *subscriber*
- *username*

Dynamic Interface Configuration Using a Profile

You define profiles by using command-line interface (CLI) commands similar to the ones you use to configure static interfaces. When configuring profiles, you can specify every layer explicitly or specify a subset of layers.

- Profile Considerations for Dynamic Interfaces on page 566
- Profile Characteristics on page 566
- How to Work with Profiles on page 571

- Creating a Profile for Dynamic Interfaces on page 571
- Assigning a Profile to a Dynamic Interface on page 572
- Example: Configuring a Profile for Dynamic Interfaces on page 573

Profile Considerations for Dynamic Interfaces

When a dynamic interface is configured, the configuration data received from the RADIUS authentication server typically overrides configuration data obtained from a profile.

In contrast to static Point-to-Point Protocol (PPP) interfaces (above which only dynamic IP interfaces can be created), static Asynchronous Transfer Mode (ATM) 1483 subinterfaces support recognition and creation of the following upper dynamic interface types or *encapsulations*: bridged Ethernet, IP, IPv6, Multilink PPP, PPP, and Point-to-Point Protocol over Ethernet (PPPoE) interfaces. The **auto-configure** command identifies the encapsulation type. For flexibility, the router provides the ability to configure an ATM 1483 subinterface with distinct profile assignments for each encapsulation type supported by the **auto-configure** command. For more information, see "Configuring a Dynamic Interface over an ATM 1483 Subinterface" on page 530.

In contrast to dynamic ATM 1483 subinterfaces, dynamic virtual LAN (VLAN) subinterfaces support recognition and creation of simultaneous IP and PPPoE upper dynamic interface types. The **vlan auto-configure** command identifies the encapsulation type. For flexibility, the router provides the ability to configure a VLAN subinterface with distinct profile assignments for each encapsulation type supported by the **vlan auto-configure** command.

Each profile typically contains configuration attributes for the expected encapsulation, in addition to attributes for other higher-interface layers through IP. If your configuration of upper layers is intended to be different depending on which incoming encapsulation is received by the subinterface, configure and assign separate profiles for each encapsulation type. If your configuration of upper layers is the same for more than one encapsulation type, configure one profile and assign it for those encapsulation types.

Profile Characteristics

Currently, profiles support bridged Ethernet, IP, IPv6, L2TP, Multilink PPP, PPP, PPPoE, and VLANs. You create a profile with a specific set of characteristics. You then assign the profile to multiple interfaces instead of creating separate interfaces with identical attributes. After you create a profile, you can assign it to static ATM 1483, static PPP, or static VLAN major interfaces on different devices.

Bridged Ethernet Characteristics

A profile can contain the following bridged Ethernet characteristic:

- mtu—Sets the maximum allowable size, in bytes, of the MTU for dynamic bridged Ethernet interfaces

IP Characteristics

A profile can contain one or more of the following IP characteristics:

- access-routes—Enables the creation of host access routes on an interface
- address—Configures an IP address on an interface
- auto-configure ip-subscriber—Configures a primary IP interface to enable dynamic creation of subscriber interfaces
- auto-detect ip-subscriber—Enables packet detection on the router and specifies that IP automatically detects packets that do not match any entries in the demultiplexer table
- directed-broadcast—Enables directed broadcast forwarding
- filter-options all—Filters out packets that include IP options
- igmp—Configures an IGMP interface
- ignore-df-bit—Specifies that the don't-fragment bit is ignored
- inactivity-timer—Configures an inactivity timer value for IP interfaces
- inspection—Associates an inspection list to the interface for firewalling
- mtu—Configures the MTU for a network
- nat—Configures the interface as inside or outside for NAT
- policy—Assigns a policy to the ingress or egress of an interface
- redirects—Enables transmission of ICMP redirect messages
- route-cache flow sampled—Enables J-Flow statistics on an interface
- route-map ip-subscriber—Configures the interface for route-map processing
- sa-validate—Verifies that a packet has been sent from a valid source address
- tcp adjust-mss—Modifies MSS on TCP connections when path MTU detection is not sufficient
- unnumbered—Configures IP on this interface without a specific address
- virtual-router—Specifies a virtual router to which interfaces created by this profile attach

IPv6 Characteristics

A profile can contain one or more of the following IPv6 characteristics:

- address—Configures an IPv6 address on an interface
- http—Configures the HTTP local server for IPv6
- http redirectUrl—Configures the URL to which a subscriber's initial Web browser session is redirected
- nd—Enables Neighbor Discovery on an interface
- nd managed-config-flag—Sets the "managed address configuration" flag in IPv6 router advertisements

- nd other-config-flag—Sets the “other stateful configuration” flag in IPv6 router advertisements
- nd prefix-advertisement—Specifies which IPv6 prefixes are included in IPv6 router advertisements
- nd ra-interval—Configures the interval between IPv6 router advertisements
- nd ra-lifetime—Configures the router advertisement lifetime
- nd reachable-time—Configures the amount of time the router can reach an IPv6 node after a reachability confirmation event occurs
- nd suppress-ra—Disables router advertisement transmissions
- mld—Configures the MLD interface
- mtu—Configures the MTU for a network
- policy—Attaches (or removes) a policy to (or from) an interface
- sa-validate—Enables source address validation
- unnumbered—Configures IPv6 on this interface without a specific address
- virtual-router—Specifies a virtual router to which interfaces created by this profile attach

L2TP Characteristics

A profile can contain the following L2TP characteristic:

- policy—Assigns an L2TP policy list to a profile

MLPPP and PPP Characteristics

A profile can contain one or more of the following MLPPP or PPP characteristics:

- aaa-accounting-broadcast—Assigns a broadcast virtual router group to enable broadcast accounting in the PPP profile
- aaa-profile—Assigns an AAA profile
- authentication—Requests PAP or CHAP authentication from a PPP peer
- authentication virtual router—Specifies a virtual router for the authentication virtual router context
- chap challenge length—Modifies the length of the CHAP challenge
- fragmentation—Enables fragmentation on an MLPPP link interface
- hash-link-selection—Enables use of a hash-based algorithm to select the link on which the router transmits non-best-effort (high-priority) packets, such as voice or video, on dynamic MLPPP interfaces
- initiate-ip—Initiates IPv4 for passive clients
- initiate-ipv6—Initiates IPv6 for passive clients

- ipcp lockout—Terminates an invalid subscriber entry and prevents additional Internet Protocol Control Protocol negotiations
- ipcp netmask—Controls the negotiation of the IPCP netmask option 0x90; *disabled* indicates do not negotiate, *enabled* indicates negotiate
- ipcp-lockout-duration—Configures the time period during which additional IPCP negotiations are prevented
- ipcp-max-negotiation—Configures the maximum number of requests for IPv4 addresses that can be received per subscriber during the time interval configured for IPCP renegotiations to be received
- ipcp-nego-duration—Configures the time period during which IPCP renegotiations for IPv4 addresses that the router or the provider edge device can receive from a subscriber are restricted
- keepalive—Specifies a keepalive value, in seconds
- log—Enables packet or state machine logging for any dynamic interfaces that use the profile
- magic-number disable—Disables negotiation of the local magic number
- magic-number ignore-mismatch—Causes the router to ignore a mismatch of the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number.
- max-negotiations—Configures the maximum number of LCPl, IPCP, or IPv6CP renegotiation attempts that the router accepts before terminating a PPP session
- mru—Configures the maximum receive unit size for the interface
- multilink enable—For MLPPP interfaces only, enables the creation of dynamic MLPPP interfaces
- multilink multiclass—Enables the creation of multilink classes on a multiclass MLPPP interface
- multilink multiclass fragmentation—Enables fragmentation on a multiclass MLPPP interface
- multilink multiclass multilink-classes—Configures the maximum number of multilink classes that can be created
- multilink multiclass reassembly—Enables reassembly on a multiclass MLPPP interface
- multilink multiclass traffic-class—Configures mapping of QoS traffic classes to multilink classes on a multiclass MLPPP interface
- passive-mode—Forces the interface into passive mode before LCP negotiation begins, for a period of one second to enable slow clients to start up and initiate the LCP negotiation
- peer dns—Resolves conflicts when the E Series router and the PPP peer system have the primary and secondary DNS addresses configured with different values

- peer wins—Resolves conflicts when the E Series router and the PPP peer system have the primary and secondary WINS addresses configured with different values
- reassembly—Enables reassembly on an MLPPP link interface

PPPoE Characteristics

A profile can contain one or more of the following PPPoE characteristics:

- AC name—Adds an access concentrator name to the profile configuration
- always-offer—Causes the router to offer to set up a session for the client, even when the router has insufficient resources to establish a session
- duplicate-protection—Prevents a client from establishing more than one session using the same MAC address

NOTE: When the duplicate protection feature is enabled for PPPoE sessions that contain the IWF-Session DSL VSA (26–254) in the Point-to-Point Protocol over Ethernet Active Discovery Request packet sent from PPPoE clients to the access concentrator, multiple IWF PPPoE sessions that contain the same MAC address are still processed and can access network services until the maximum number of PPPoE sessions configured per major interface (configured using the **pppoe sessions** command) is reached.

- log pppoeControlPacket—Enables packet trace logging on PPPoE dynamic interfaces created with this profile
- motm—Causes the router to send a PADM message of the minute
- mtu—Configures the MTU
- remote-circuit-id—Enables the router to capture and process a vendor-specific tag containing a remote circuit ID transmitted from a digital subscriber line access multiplexer device
- service-name-table—Assigns a PPPoE service name table to dynamic interfaces created with this profile
- sessions—Specifies the maximum number of subinterfaces permitted on a PPPoE major interface
- url—Causes the PPPoE application to send a URL string to the new client

VLAN Characteristics

A profile can contain one or more of the following VLAN characteristics:

- advisory-rx-speed—Sets an advisory receive speed for VLAN subinterfaces
- advisory-tx-speed—Sets an advisory connect speed for VLAN subinterfaces
- auto-configure—Specifies the types of upper-interface encapsulations that are accepted or detected by the dynamic VLAN subinterface

- auto-configure agent-circuit-identifier—Enables the creation of VLAN subinterfaces that are based on agent-circuit-identifier information
- description—Assigns a description to VLAN subinterfaces that are created with this profile
- policy—Attaches (or removes) a policy to (or from) a dynamically created VLAN
- profile—Adds a nested profile assignment, which references another profile that dynamically configures an upper-interface encapsulation type over the VLAN subinterface
- service-profile—Specifies a service profile name to a dynamically created VLAN
- svlan ethertype—Specifies that the packet must use this Ethertype to create the dynamic VLAN subinterface

How to Work with Profiles

Figure 58 on page 571 shows how to create a profile and assign characteristics to it.

Figure 58: Creating and Configuring a Profile

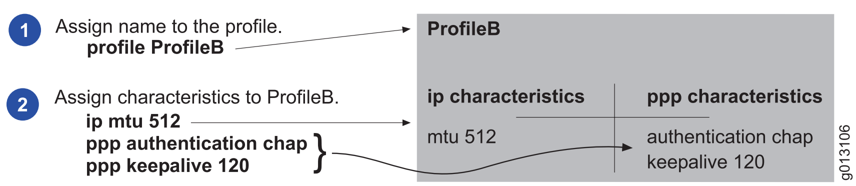
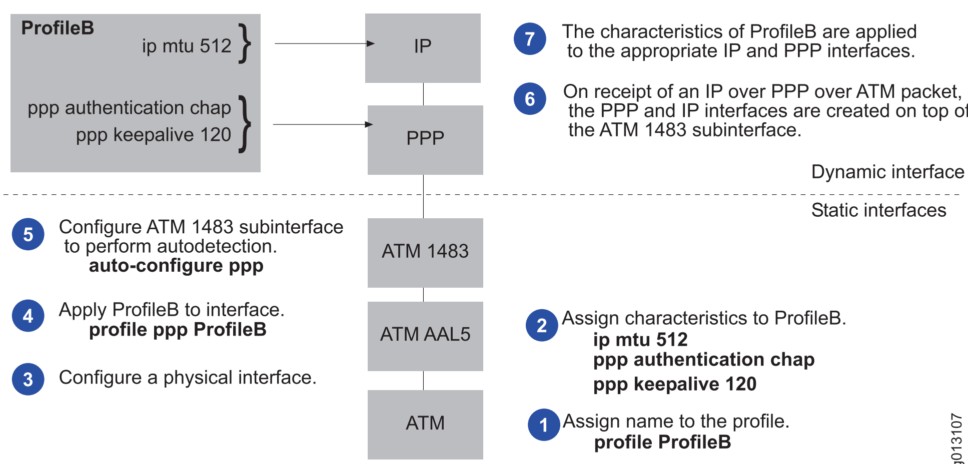


Figure 59 on page 571 shows how to assign a profile to static interfaces. These static interfaces create dynamic interfaces above them.

Figure 59: Assigning a Profile to a Static Interface



Creating a Profile for Dynamic Interfaces

You can create a profile by using CLI commands similar to those used to create the equivalent static interfaces. You can configure a profile for bridged Ethernet, IP, IPv6, MLPPP, PPP, PPPoE, or VLAN interfaces.

To configure a profile:

1. Create a profile by assigning it a name.

   host1(config)#**profile foo**

2. Specify a virtual router to which to assign dynamic IP interfaces created with this profile.

   host1(config-profile)#**ip virtual-router egypt**

3. Specify an IP loopback interface for dynamic IP interfaces created with this profile to be associated.

   host1(config-profile)#**ip unnumbered loopback 0**

4. Configure IPCP option 0x90.

   host1(config-profile)#**ppp ipcp netmask**

5. Optionally set IP, IPv6, MLPPP, PPP, or PPPoE characteristics. For more information, see "Configuring Profile Characteristics" on page 576.

NOTE: When configuring either IP or IPv6 to operate over PPP, you might want to initiate IP or IPv6 by using the appropriate **ppp initiate** command, either **ppp initiate-ip** or **ppp initiate-ipv6**. This command initiates either IPv4 or IPv6 in the event you are connecting to a passive client.

Assigning a Profile to a Dynamic Interface

Use the **profile** command from Interface Configuration mode when you assign a profile to an interface.

For static PPP interfaces, you can assign only a profile for IP encapsulations. For static ATM 1483 subinterfaces, you can assign one profile for each bridged Ethernet, IP, PPP, and PPPoE encapsulation. For static VLAN subinterfaces, you can assign one profile for each IP or PPPoE encapsulation. You can also use the default keyword **any**, which applies to any autoconfigured encapsulation that does not have specific profile assignment.

For example, the following commands cause the router to use ProfileB when an IPoA packet is received, and to use ProfileA for any other received encapsulation that is autoconfigured. When you omit the keyword, it defaults to **any**.

host1(config-subif)#**profile any ProfileA**
host1(config-subif)#**profile ip ProfileB**

To assign a profile to an interface:

1. Configure a physical interface.

   host1(config-if)#**interface atm 2/1.10**

2. Configure a PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type. For more information, see "Creating a PVC on an ATM 1483 Subinterface" on page 537.

   host1(config-subif)#**atm pvc 10 10 22 aal5snap** host1(config-subif)#**atm pvc 10 10 22 aal5autoconfig**

3. Apply an existing profile.

   host1(config-subif)#**profile ip holland**

4. Assign subscriber identification. For more information, see "Configuring a Local Subscriber for a Dynamic IPoA or Bridged Ethernet Interface" on page 528.

   host1(config-subif)#**subscriber ip user ispname domain abc.com password 3fds9jpt**

5. Enable the dynamic encapsulation type. For more information, see "Configuring a Dynamic Interface over an ATM 1483 Subinterface" on page 530.

   host1(config-subif)#**auto-configure ip**

Example: Configuring a Profile for Dynamic Interfaces

This example shows different ways to configure profiles.

- Requirements on page 573
- Overview on page 573
- Configuring and Assigning Profiles on page 573

Requirements

This example uses the following software and hardware components:

- JunosE Release 7.1.0 or higher-numbered releases
- E Series router (ERX7xx models, ERX14xx models, the ERX310 router, the E120 router, or the E320 router)
- ASIC-based line modules that support Fast Ethernet or Gigabit Ethernet

Overview

Currently, profiles support bridged Ethernet, IP, IPv6, L2TP, Multilink PPP, PPP, PPPoE, and VLANs. You create a profile with a specific set of characteristics. You then assign the profile to multiple interfaces instead of creating separate interfaces with identical attributes. After you create a profile, you can assign it to static ATM 1483, static PPP, or static VLAN major interfaces on different devices.

Configuring and Assigning Profiles

This example explains various ways to assign the created profiles to multiple interfaces.

- Creating Profiles on page 574
- Assigning Distinct Profiles for Each Encapsulation on page 574
- Assigning a Single Profile for All Encapsulations on page 574
- Assigning a Profile Using any Wildcard on page 575
- Assigning a Profile for bridgedEthernet Encapsulation on page 575

Creating Profiles

Step-by-Step Procedure

To create profiles with various characteristics assigned:

1. Create a new profile with IP characteristics only.

   host1(config)#**profile ProfileA**
   host1(config-profile)#**ip mtu 1024**
   host1(config-profile)#**exit**

2. Create a new profile with both IP and PPP characteristics.

   host1(config)#**profile ProfileB**
   host1(config-profile)#**ip mtu 512**
   host1(config-profile)#**ppp authentication chap**
   host1(config-profile)#**ppp keepalive 120**
   host1(config-profile)#**exit**

3. Create a new profile with IP, PPP, and PPPoE characteristics.

   host1(config)#**profile ProfileC**
   host1(config-profile)#**ip mtu 1400**
   host1(config-profile)#**ppp authentication chap**
   host1(config-profile)#**ppp keepalive 60**
   host1(config-profile)#**pppoe sessions 64**
   host1(config-profile)#**exit**

Assigning Distinct Profiles for Each Encapsulation

Step-by-Step Procedure

Distinct profiles are assigned for each encapsulation, where the configuration of dynamic layers varies according to which incoming encapsulation the ATM 1483 subinterface detects.

1. Assign the created profiles for each encapsulation.

   host1(config)#**interface atm 4/0.1**
   host1(config-subif)#**atm pvc 10 10 22 aal5autoconfig**
   host1(config-subif)#**profile ip ProfileA**
   host1(config-subif)#**profile ppp ProfileB**
   host1(config-subif)#**profile pppoe ProfileC**
   host1(config-subif)#**subscriber ip user atm1 domain isp1 password atm1pw**

2. Enable autodetection for the encapsulation types with the default lockout time range.

   host1(config-subif)#**auto-configure ip**
   host1(config-subif)#**auto-configure ppp**
   host1(config-subif)#**auto-configure pppoe**
   host1(config-subif)#**exit**

Assigning a Single Profile for All Encapsulations

Step-by-Step Procedure

The same profile is assigned for all encapsulations. The configuration of dynamic layers is the same regardless of incoming encapsulations detected by ATM. Only relevant profile attributes are used for whichever dynamic interface layers are actually constructed.

1. Assign the same profile for all encapsulations.

```
host1(config)#interface atm 4/0.2
host1(config-subif)#atm pvc 200 0 200 aal5autoconfig
host1(config-subif)#profile any ProfileC
host1(config-subif)#subscriber ip user atm2 domain isp2 password atm2pw
```

2. Enable autodetection for the encapsulation types with the default lockout time range.

```
host1(config-subif)#auto-configure ip
host1(config-subif)#auto-configure ppp
host1(config-subif)#auto-configure pppoe
host1(config-subif)#exit
```

Assigning a Profile Using any Wildcard

Step-by-Step Procedure

The profile is implicitly assigned via the **any** encapsulation wildcard. Configuration of dynamic layers is the same regardless of incoming encapsulation detected by ATM.

1. Assign the profile using the **any** keyword.

```
host1(config)#interface atm 4/0.3
host1(config-subif)#atm pvc 300 0 300 aal5autoconfig
host1(config-subif)#profile any ProfileC
host1(config-subif)#subscriber ip user atm2 domain isp3 password atm3pw
```

2. Enable autodetection for the IP encapsulation type with a lockout time range of 3600–7200 seconds (1–2 hours).

```
host1(config-subif)#auto-configure ip lockout-time 3600 7200
```

3. Enable autodetection for other encapsulation types with the default lockout time range.

```
host1(config-subif)#auto-configure ppp
host1(config-subif)#auto-configure pppoe
host1(config-subif)#exit
```

Assigning a Profile for bridgedEthernet Encapsulation

Step-by-Step Procedure

The profile is assigned for the bridgedEthernet encapsulation type.

1. Assign the profile for the bridgedEthernet encapsulation.

```
host1(config)#interface atm 4/0.3
host1(config-subif)#atm pvc 300 0 300 aal5autoconfig
host1(config-subif)#profile bridgedEthernet ProfileA
host1(config-subif)#subscriber bridgedEthernet user atm3 domain isp1 password
  fjdkei
```

2. Enable autodetection for the bridged Ethernet encapsulation type with a lockout time range of 3600–21600 seconds (1–6 hours).

```
host1(config-subif)#auto-configure bridgedEthernet lockout-time 3600 21600
```

Related Documentation

- Dynamic PPP and PPPoE Interfaces over Static ATM on page 538
- Dynamic PPPoE Interfaces over PPPoE Static Interfaces on page 542

- Dynamic Bridged Ethernet Interfaces on page 558
- Configuring Profile Characteristics on page 576
- Monitoring Status or Summary Information for ATM Subinterfaces on page 595
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606

- *atm pvc*
- *auto-configure*
- *interface atm*
- *ip mtu*
- *ip unnumbered*
- *ip virtual-router*
- *ppp authentication*
- *ppp ipcp netmask*
- *ppp keepalive*
- *pppoe sessions*
- *profile*
- *subscriber*
- *vlan auto-configure*

Configuring Profile Characteristics

This topic describes the various characteristics that can be configured for a profile.

- Configuring Bridged Ethernet Characteristics for a Profile on page 576
- Configuring IPv4 Characteristics for a Profile on page 577
- Configuring IPv6 Characteristics for a Profile on page 580
- Configuring L2TP Characteristics for a Profile on page 582
- Configuring MLPPP and PPP Characteristics for a Profile on page 582
- Configuring PPPoE Characteristics for a Profile on page 586
- Configuring VLAN Characteristics for a Profile on page 588

Configuring Bridged Ethernet Characteristics for a Profile

To configure bridged Ethernet characteristics for a profile:

- Set the maximum allowable size, in bytes, of the MTU for bridged Ethernet interfaces.

  ```
  host1(config-profile)#bridge1483 mtu 1684
  ```

  Use the **no** version to restore the default MTU size for bridged Ethernet interfaces, 1518 bytes.

Configuring IPv4 Characteristics for a Profile

To configure IPv4 characteristics for a profile:

- Enable an access route in a profile.

  host1(config-profile)#**ip access-routes**

  Use the **no** version to remove the access route.

- Assign an IP address to a profile.

  host1(config-profile)#**ip address 192.13.5.61**

  Use the **no** version to remove the IP address assignment from the profile.

- Allow more than one subscriber to have the same IP address across different virtual routers in the dynamic subscriber interface configuration by appending the virtual router name to the interface. You can use this command from either Interface Configuration mode or Profile Configuration mode.

  host1(config-profile)#**ip auto-configure append-virtual-router-name**

  Use the **no** version to disable ip auto-configure on the static primary interface if it is already configured. This feature is enabled by default in non-dynamic subscriber interface configuration with DHCP-Local Server

- Configure a primary IP interface to enable dynamic creation of subscriber interfaces.

  You can use the **include-primary** keyword to specify that the primary interface is assigned to the first subscriber.

  You can use the **exclude-primary** keyword to specify that the primary interface is not used for dynamic subscribers. By default, the primary interface is not assigned to a dynamic subscriber.

  host1(config-profile)#**ip auto-configure ip-subscriber include-primary**

  Use the **no** version to disable creation of dynamic subscriber interfaces associated with this primary IP interface. Use the **no** version with the **include-primary** keyword to specify that the primary interface is not assigned to a subscriber. Use the **no** version with the **exclude-primary** keyword to specify that the primary interface is assigned to a subscriber.

- Enable packet detection on the router and specify that IP automatically detect packets that do not match any entries in the demultiplexer table.

  host1(config-profile)#**ip auto-detect ip-subscriber**

  Use the **no** version to restore the default behavior, which disables packet detection.

- Enable a directed broadcast address in a profile.

  host1(config-profile)#**ip directed-broadcast**

  Use the **no** version to remove the directed broadcast address from the profile.

- Filter out packets that include IP options.

  host1(config-profile)#**ip filter-options all**

Use the **no** version to disable filtering of packets with IP options.

- Enable IGMP on an interface, and set the IGMP version to IGMPv2.

  host1(config-profile)#ip igmp

  Use the **no** version to disable IGMP on an interface.

- Force the router to ignore the DF bit if it is set in the IP packet header for packets on an interface.

  NOTE: You can also use RADIUS VSA [26-70] to configure the router's DF bit support. The action configured by the RADIUS VSA takes precedence over the action configured by the **ip ignore-df-bit** command. For more information, see *Juniper Networks VSAs* and *Juniper Networks VSAs Supported for Subscriber AAA Access Messages*.

  host1(config-profile)#ip ignore-df-bit

  Use the **no** version to restore the default behavior, which is to consider the DF bit before fragmentation.

- Configure an inactivity timer value for an IP interface. IP polls the dynamic interface at the configured interval to determine whether the interface was active during the interval. Inactive interfaces are deleted only when the period of inactivity is equal to or greater than the configured value.

  For example, if you configure an inactivity timer of 15 minutes, IP polls the interface every 15 minutes. If a poll determines that the interface was last active 14 minutes earlier, the inactive time is less than the configured value so nothing happens. IP polls again 15 minutes later. If the interface is still inactive then the total period of inactivity is now 29 minutes. This is greater than the configured value and the interface is deleted.

  host1(config-profile)#ip inactivity-timer 100

  Use the **no** version to restore the default behavior, which disables the inactivity timer.

- Associate an inspection list to the inbound or outbound side of the IP interface.

  host1(config-profile)#ip inspection list1

  Use the **no** version to remove the inspection list association to this interface.

- Assign the maximum transmission unit size sent on an IP interface.

  host1(config-profile)#ip mtu 1000

  Use the **no** version to restore the default value, 0, which means that the router takes the value from a lower protocol layer.

- Mark interfaces that participate in NAT translation as residing on the inside or the outside network.

  host1(config-profile)#ip nat inside

  Use the **no** version to unmark the interface (the default) so that it does not participate in NAT translation.

- Assign a policy list to the ingress or egress of an interface to which the profile is attached.

  host1(config-profile)#**ip policy secondary-input my-policy**

  Use the **no** version to remove the association between a policy list and a profile.

- Enable the sending of redirect messages if the software is forced to resend a packet through the same interface on which it was received.

  host1(config-profile)#**ip redirects**

  Use the **no** version to remove the assignment from the profile.

- Enable J-Flow statistics on the interface.

  host1(config-profile)#**ip route-cache flow sampled**

  Use the **no** version to delete J-Flow statistics from the profile.

- Configure an interface for route-map processing and specify the route map that is applied to the IP interface subscriber.

  host1(config-profile)#**ip route-map ip-subscriber chicagoRouteMap**

  Use the **no** version to delete the route map.

- Enable source address validation on an IP interface. Source address validation verifies that a packet has been sent from a valid source address.

  host1(config-profile)#**ip sa-validate**

  Use the **no** version to disable source address validation.

- Modify the maximum segment size for TCP SYN packets traveling through the interface.

  host1(config-profile)#**ip tcp adjust-mss 200**

  Use the **no** version to remove the maximum segment size modification.

- Specify the unnumbered interface with which dynamic interfaces created with the profile are associated.

  You can configure a loopback using RADIUS instead of adding one to the profile using the **ip unnumbered loopback** command.

  host1(config-profile)#**ip unnumbered loopback 5**

  Use the **no** version to remove the assignment from the profile.

- Assign a virtual router to a profile. Interfaces created by the profile are attached to this virtual router.

  If the virtual router specified in a profile with the **ip virtual-router** command differs from the virtual router provided by AAA, IP uses the virtual router provided by AAA when the dynamic IP upper-layer interface is created. For more information about using the **ppp authentication virtual-router** command, see "Configuring MLPPP and PPP Characteristics for a Profile" on page 582.

  host1(config-profile)#**ip virtual-router salem1**

  Use the **no** version to remove the virtual router assignment from the profile. If no virtual router is specified via RADIUS, then any subsequent use of the profile to create a dynamic interface fails for lack of a virtual router.

Configuring IPv6 Characteristics for a Profile

To configure IPv6 characteristics for a profile:

- Configure an IPv6 address on an interface to which the profile is attached.

  host1(config-profile)#**ipv6 address 1::1/64**

  Use the **no** version to remove the IPv6 address from the interface.

- Create the HTTP local server to listen and process for IPv6 exception packets. For more information, see *Configuring the HTTP Server to Support Guided Entrance* in the *JunosE Broadband Access Configuration Guide*.

  host1(config)#**ipv6 http**

  Use the **no** version to delete the HTTP local server.

- Specify the URL to which a subscriber's HTTP access session is redirected. For more information, see *Configuring the HTTP Server to Support Guided Entrance* in the *JunosE Broadband Access Configuration Guide*.

  The first access session is typically used by the Service Manager application to provide initial provisioning and service selection for the subscriber.

  HTTP redirect is per-interface; use the command in Interface, Subinterface or Profile Configuration mode for static interfaces.

  The redirect URL can be a maximum of 64 characters.

  NOTE: The HTTP local server must be configured and enabled in the virtual router for the interface on which you use the **ipv6 http redirectUrl** command. Otherwise, the URL redirect operation will fail.

  host1(config-profile)#**ipv6 http redirectUrl http://ispsite.redirect.com**

  Use the **no** version to restore the default, which disables the HTTP redirect feature.

- Enable MLD on an interface, and set the MLD version to MLDv2.

  host1(config-profile)#**ipv6 mld**

  Use the **no** version to disable MLD on an interface.

- Set the maximum transmission unit size of IPv6 packets sent on an interface.

  host1(config-profile)#**ipv6 mtu 1000**

  Use the **no** version to restore the default value, 0, which means that the router takes the value from a lower protocol layer.

- Enable the IPv6 Neighbor Discovery process on an interface.

  host1(config-profile)#**ipv6 nd**

  Use the **no** version to disable the Neighbor Discovery process.

- Set the "managed address configuration" flag in IPv6 router advertisements.

host1(config-profile)#**ipv6 nd managed-config-flag**

Use the **no** version to clear the flag from IPv6 router advertisements.

- Set the "other stateful configuration" flag in IPv6 router advertisements.

  host1(config-profile)#**ipv6 nd other-config-flag**

  Use the **no** version to clear the flag from IPv6 router advertisements.

- Specify which IPv6 prefixes the system includes in IPv6 router advertisements.

  host1(config-profile)#**ipv6 nd prefix-advertisement 2002:1::/64 60000 45000 onlink autoconfig**

  Use the **no** version to remove any prefixes from the IPv6 routing advertisements.

- Specify the interval, in seconds, between IPv6 router advertisement retransmissions on an interface.

  host1(config-profile)#**ipv6 nd ra-interval 500**

  Use the **no** version to restore the default interval, 200 seconds.

- Specify the router lifetime value, in seconds, in IPv6 router advertisements on an interface. The router lifetime value is the amount of time the router is considered the default router on this interface.

  host1(config-profile)#**ipv6 nd ra-lifetime 900**

  Use the **no** version to restore the default lifetime, 1800 seconds.

- Specify the amount of time, in milliseconds, that the E Series router can reach a remote IPv6 node after some reachability confirmation event has occurred.

  host1(config-profile)#**ipv6 nd reachable-time 30000**

  Use the **no** version to restore the default value 0 milliseconds for router advertisements and 3,600,000 milliseconds (1 hour) for Neighbor Discovery activity of the E Series router.

- Suppress IPv6 router advertisement transmissions on a LAN local area network (Ethernet) interface.

  host1(config-profile)#**ipv6 nd suppress-ra**

  Use the **no** version to reenable the sending of IPv6 router advertisement transmissions on the LAN (Ethernet) interface

- Assign a policy list to the ingress or egress of an interface to which the profile is attached.

  host1(config-profile)#**ipv6 policy secondary-input my-policy**

  Use the **no** version to remove the association between a policy list and a profile.

- Enable source address validation on an IPv6 interface.

  host1(config-profile)#**ipv6 sa-validate**

  Use the **no** version to disable source address validation.

- Enable IPv6 processing on an interface without assigning an explicit IPv6 address to that interface.

host1(config-profile)#**ipv6 unnumbered loopback 0**

Use the **no** version to remove the IPv6 address from the interface.

- Specify a virtual router in an IPv6 profile. Dynamic interfaces created with the profile are assigned to this virtual router

  host1(config-profile)#**ipv6 virtual-router westford01**

  Use the **no** version to remove the virtual router assignment from the profile. If no virtual router is specified via RADIUS, then any subsequent use of the profile to create a dynamic interface fails for lack of a virtual router.

Configuring L2TP Characteristics for a Profile

To configure L2TP characteristics for a profile:

- Assign a policy list to the ingress or egress of an interface to which the profile is attached.

  host1(config-profile)#**l2tp policy secondary-input my-policy**

  Use the **no** version to remove the association between a policy list and a profile.

Configuring MLPPP and PPP Characteristics for a Profile

To configure MLPPP and PPP characteristics for a profile:

- Assign a broadcast virtual router group to enable broadcast accounting in a PPP profile. For more information about broadcast accounting, see *RADIUS Authentication and Accounting Servers Configuration Overview* and "Broadband Remote Access Support for PPP Overview" on page 265.

  NOTE: When the broadcast virtual router group is configured at both PPP profile and virtual router levels, AAA sends accounting messages only to broadcast accounting servers in the broadcast virtual router group that is configured at both PPP profile and virtual router levels.

  host1(config-profile)#**ppp aaa-accounting-broadcast groupxyz**

  Use the **no** version to remove the broadcast virtual router group assignment.

- Assign an AAA profile to static and dynamic, multilink and nonmultilink PPP interfaces.

  NOTE:
  - Although an AAA profile and an interface profile have similar functionality, they are not related and you need to treat them differently.
  - For more information about AAA profiles, see *JunosE Broadband Access Configuration Guide*.

  host1(config-profile)#**ppp aaa-profile westford24**

  Use the **no** version to remove the AAA profile assignment.

- Request authentication from a PPP peer router.

  host1(config-profile)#**ppp authentication pap chap**

  To specify the name of a virtual router to be used as the authentication virtual router context, use the **virtual-router** keyword.

  host1(config-profile)#**ppp authentication virtual-router boston pap chap**

  Use the **no** version to specify that the router does not require authentication.

- Modify the length of the CHAP challenge by specifying the minimum length and maximum length.

  CAUTION: Do *not* use the **ppp chap-challenge-length** command; increasing the minimum length (from the default 16 bytes) or decreasing the maximum length (from the default 32 bytes) reduces the security of your router.

  host1(config-profile)#**ppp chap-challenge-length 24 28**

  Use the **no** version to restore the default minimum 16 bytes and default maximum 32 bytes.

- Enable fragmentation on an MLPPP link interface and optionally specify the maximum fragment size, in octets, to be used on the link.

  host1(config-profile)#**ppp fragmentation 128**

  Use the **no** version to disable fragmentation on the link and restore the default fragment size, which is the link's MTU.

- Enable use of a hash-based algorithm to select the link on which the router transmits non-best-effort (high-priority) packets, such as voice or video, on the dynamic MLPPP interfaces created by this profile.

  host1(config-profile)#**ppp hash-link-selection**

  Use the **no** version to restore the default round-robin algorithm for MLPPP link selection.

- Initiate IPv4 for passive clients. By default, PPP creates IP instances when it receives client requests.

  host1(config-profile)#**ppp initiate-ip**

  Use the **no** version to disable initiation of IP.

- Initiate IPv6 for passive clients. By default, PPP creates IPv6 instances when it receives client requests.

  host1(config-profile)#**ppp initiate-ipv6**

  Use the **no** version to disable initiation of IPv6.

- Terminate invalid IPv4 subscribers and prevent additional IPCP negotiations.

  For more information about how the IPv4 addresses are restored, see *Configuring Point-to-Point Protocol* in *JunosE Link Layer Configuration Guide*.

  host1(config-profile)#**ppp ipcp lockout**

Use the **no** version to disable the IPCP lockout option on the interface.

- Specify IPCP option 0x90 for each PPP interface. By default, IPCP option 0x90 is disabled on the interface.

  host1(config-profile)#**ppp ipcp netmask**

  Use the **no** version to disable IPCP option 0x90 option on the interface.

- Configure the time period during which additional IPCP negotiations are prevented.

  host1(config-profile)#**ppp ipcp-lockout-duration 400**

  Use the **no** version to restore the default value, 600 seconds.

- Configure the maximum number of requests for IPv4 addresses that can be received per subscriber during the time interval configured for IPCP renegotiations to be received.

  host1(config-profile)#**ppp ipcp-max-negotiation 4**

  Use the **no** version to restore the default value, 6.

- Configure the time period during which IPCP renegotiations for IPv4 addresses that the router or the provider edge device can receive from a subscriber are restricted.

  host1(config-profile)#**ppp ipcp-nego-duration 300**

  Use the **no** version to restore the default value, 60 seconds.

- Configure to prompt the CPE to negotiate the IPCP primary and secondary DNS options that are locally available with the broadband remote access server.

  host1(config-profile)#**ppp ipcp prompt-option dns**

  Use the **no** version to disable the command.

- Specify the keepalive timeout value.

  You can use the **ppp keepalive** command without a value to restore the default, 30 seconds.

  host1(config-profile)#**ppp keepalive 50**

  Use the **no** version to disable keepalive.

- Enable PPP packet or state machine logging on any dynamic interface that uses the profile being configured.

  host1(config-profile)#**ppp log pppPacket**

  Or

  host1(config-profile)#**ppp log pppStateMachine**

  NOTE: This command is equivalent to the **log severity debug pppPacket** and **log severity debug pppStateMachine** commands.

  Use the **no** version to disable packet or state machine logging.

- Disable negotiation of the local magic number.

  host1(config-profile)#**ppp magic-number disable**

Issuing this command prevents the router from detecting loopback configurations. Use the **no** version to restore negotiation of the local magic number.

- Configure the router to ignore a mismatch of the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number.

  For more information about using this command and LCP peer magic number validation, see "Understanding PPP Link Control Protocol" on page 263.

  host1(config-profile)#**ppp magic-number ignore-mismatch**

  Use the **no** version to restore the default behavior, in which the router terminates the PPP connection if it detects an LCP peer magic number mismatch.

- Configure the maximum number of LCP, IPCP, or IPv6CP renegotiation attempts, in the range 1–65535, that the router accepts before terminating a PPP session.

  host1(config-profile)#**ppp max-negotiations 15**

  NOTE: If you do not specify the optional **lcp**, **ipcp**, or **ipv6cp** keyword, the **ppp max-negotiations** command sets the maximum number of renegotiation attempts for each of LCP, IPCP, and IPv6CP to the value you specify, or to the default value (30) if you omit the optional value for maximum renegotiation attempts.

  Use the **no** version to restore the default value, 30 renegotiation attempts.

- Control the negotiation of the MRU.

  host1(config-profile)#**ppp mru 576**

  Use the **no** version to restore the default value, which causes PPP to use the lower-layer MRU minus the PPP header length as the MRU value.

- Enable the creation of dynamic MLPPP interfaces.

  host1 (config-profile)#**ppp multilink enable**

  Use the **no** version to cause the LNS to reject any incoming requests to create dynamic MLPPP interfaces.

- Enable multiclass MLPPP and the creation of multilink classes on a dynamic MLPPP interface.

  host1 (config-profile)#**ppp multilink multiclass multilink-classes 6**

  Use the **no** version to disable multiclass MLPPP or to restore the number of multilink classes to the default value, 1.

- Enable fragmentation on a multilink class on a dynamic MLPPP interface.

  host1(config-profile)#**ppp multilink multiclass fragmentation best-effort voice low-loss video**

  Use the **no** version to disable fragmentation on a multilink class.

- Enable reassembly on a multilink class on a dynamic MLPPP interface.

host1(config-profile)#**ppp multilink multiclass reassembly best-effort voice low-loss video**

Use the **no** version to disable reassembly on a multilink class.

- Configure mapping of QoS traffic classes to multilink classes on a dynamic MLPPP interface.

  host1(config-profile)#**ppp multilink multiclass traffic-class best-effort voice low-loss video**

  Use the **no** version to delete the mapping of QoS traffic classes to multilink classes.

- Force a static or dynamic PPP interface into passive mode before LCP negotiation begins, for a period of one second. This delay enables slow clients to start up and initiate the LCP negotiation.

  host1(config-profile)#**ppp passive-mode**

  Use the **no** version to disable passive mode.

- Resolve conflicts when the router and the PPP peer system have the primary and secondary DNS and WINS addresses configured with different values.

  host1(config-profile)#**ppp peer dns**

  Use the **no ppp peer dns** command or the **no ppp peer wins** command when you want the router to take precedence during setup negotiations between the router and the remote PC client. If the IP addresses passed to the router by the remote PC client differ from the ones you have configured on your router, the router returns the values that you configured as the correct values to the remote PC client.

- Enable reassembly on an MLPPP link interface and optionally specify the administrative MRRU value, in octets, for the link.

  host1(config-profile)#**ppp reassembly 1590**

  Use the **no** version to disable reassembly on the link and restore the default value, which is the link's local MRU.

Configuring PPPoE Characteristics for a Profile

To configure PPPoE characteristics for a profile:

- Add an access concentrator name to the profile configuration.

  host1(config-profile)#**pppoe acName CYM9876**

  Use the **no** version to remove the AC name.

- Set up the router to offer to set up a session for the client, even if the router has insufficient resources to establish a session.

  host1(config-profile)#**pppoe always-offer**

  Use the **no** version to disable this feature.

- Prevent a client from establishing more than one session using the same MAC address.

For a list of considerations to be observed when you use the duplicate protection feature for IWF PPPoE sessions, see "Guidelines for Configuring Duplicate Protection for IWF PPPoE Sessions" on page 405.

host1(config-profile)#**pppoe duplicate-protection**

Use the **no** version to disable duplicate protection.

- Enable packet trace logging on PPPoE dynamic interfaces created with this profile. Packet trace information is logged to the pppoeControlPacket log.

  host1(config-profile)#**pppoe log pppoeControlPacket**

  Use the **no** version to turn off packet trace logging.

- Configure the PPPoE application to send a PADM Message Of The Minute message. The recipient of the message is determined by the mode from which the command is issued.

  host1(config-profile)#**pppoe motm string**

  Use the **no** version to disable the command.

- Set the MTU using a combination of lower layer restrictions and controls.

  You can use the **use-lower-layer** keyword to use the lower layer interface value minus any PPPoE overhead. You can use the **use-mtu-tag** keyword to use the provided PPPoE mtu tag value.

  host1(config-profile)#**pppoe mtu 1380**

  Use the **no** version to restore the default value, 1494.

- Enable the router to capture and process a vendor-specific tag containing a remote circuit ID transmitted from a DSLAM device.

  Optionally, the router can use the remote circuit ID in place of either or both of the Calling-Station-Id [31] and NAS-Port-Id [87] RADIUS attributes to uniquely identify subscriber locations.

  host1(config-profile)#**pppoe remote-circuit-id**

  Use the **no** version to restore the default behavior, which is not to capture and process the remote circuit ID.

- Assign a PPPoE service name table to dynamic interfaces created with this profile.

  host1(config-profile)#**pppoe service-name-table myServiceTable1**

  Use the **no** version to remove the PPPoE service name table assignment.

- Specify the maximum number of PPPoE subinterfaces permitted on an interface, in the range 1–8000 (ERX routers) or 1–32,000 (E120 and E320 routers). On the ES2 10G ADV LM (E120 and E320 routers), you can have PPPoE subinterfaces in the range 1–32,000. The default value is 8000 (ERX routers) or 16,000 (E120 and E320 Broadband Services Routers) or 32,000 (ES2 10G ADV LM).

  host1(config-profile)#**pppoe sessions 3000**

  Use the **no** version to restore the default value, 8000 (ERX routers) or 16,000 (E120 and E320 routers) or 32,000 (ES2 10G ADV LM).

- Configure the PPPoE application to send the string to the new client created when the profile is dynamically attached to an IP interface.

  host1(config-profile)#**pppoe url http://www.relevanturl.com**

  Use the **no** version to disable the command.

NOTE: If you assign profiles to static PPPoE major interfaces, the PPPoE settings that you configure on the interfaces become effective and the PPPoE parameters that you configure in the profiles are not processed. However, for dynamic PPPoE interfaces that are assigned with profiles, the PPPoE settings in the profile are processed and applied on the interface.

Configuring VLAN Characteristics for a Profile

To configure VLAN characteristics for a profile:

- Assign an Ethertype value for the S-VLAN subinterface.

  host1(config-profile)#**svlan ethertype 8100**

  Or

  host1(config-profile)#**svlan ethertype 88a8**

  Use the **no** version to restore the default value, 9100.

- Set an advisory receive speed for VLAN subinterfaces that are created with the profile you are configuring.

  host1(config-profile)#**vlan advisory-rx-speed 2000**

  Use the **no** version to restore the default behavior—the Rx speed is not sent to the LNS.

- Set an advisory connect speed for VLAN subinterfaces that are created with the profile that you are configuring.

  host1(config-profile)#**vlan advisory-tx-speed 2000**

  Use the **no** version to restore the default behavior—the Tx speed is not sent to the LNS.

- Specify the types of dynamic upper-interface encapsulations that are accepted or detected by a dynamic VLAN subinterface.

  host1(config-profile)#**vlan auto-configure ip**

  Use the **no** version to terminate detection of the specified encapsulation type.

- Create a VLAN subinterface that is based on the agent-circuit-id information in the option 82 field of DHCP messages or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets.

  host1(config-profile)#**vlan auto-configure agent-circuit-identifier**

  Use the **no** version to disable creation of VLAN subinterfaces based on agent-circuit-identifier information.

- Assign a description to VLAN subinterfaces that are created with this profile.

host1(config-profile)#**vlan description test1**

Use the **no** version to remove the VLAN description.

- Assign a VLAN policy list to an interface.

  host1(config-profile)#**vlan policy input VlanPolicy33 statistics enabled preserve**

  Use the **no** version to remove the association between a policy list and an interface or a profile.

- Add a nested profile assignment to a base profile for a dynamic VLAN subinterface.

  A nested profile assignment references another profile that configures attributes for a dynamic upper-interface type over the VLAN subinterface.

  host1(config-profile)#**vlan profile pppoe vlanProfilePppoe**
  host1(config-profile)#**vlan profile ip vlanProfileIP**

  Use the **no** version to remove the profile assignment for the upper-interface encapsulation type.

- Specify a service profile name for a dynamic VLAN and to enter Service Profile Configuration mode. Service profiles contain user and password information, and are used in route maps for subscriber management and to authenticate subscribers with RADIUS.

  host1(config)#**vlan service-profile vlanClass1Service**
  host1(config-service-profile)#

  Use the **no** version to delete the service profile.

Related Documentation

- Dynamic PPP and PPPoE Interfaces over Static ATM on page 538
- Dynamic PPPoE Interfaces over PPPoE Static Interfaces on page 542
- Dynamic Bridged Ethernet Interfaces on page 558
- Profile Characteristics on page 566
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606
- Monitoring Status and Configuration Information for VLAN Subinterfaces on page 614

Scripts and Macros for Dynamic Interfaces Overview

Scripts and macros are intended to reduce the management of static interfaces. Because dynamic interfaces have static lower layers, you can use scripts and macros to configure the static portion of all dynamic interfaces.

A script or macro can specify the static interface by using the **interface**, **auto-configure**, **subscriber**, or **profile** commands. These commands enable you to configure the interface as dynamic and to specify configuration sources for the dynamic upper layers. These files can then be executed by the router as though the commands were entered at the terminal.

- Scripts—You can create script files containing a series of command-line interface (CLI) commands. The resulting script can be executed via the **configure file** command.
- Macros—You can create macros that generate and execute CLI commands. You first write macros on a computer and then copy them to the router. You issue the **macro** command from the CLI to execute both local macros or macros stored remotely. The macro command is available from all command modes. See *Writing CLI Macros* in *JunosE System Basics Configuration Guide*.

NOTE: For a list of vendor-specific attributes (VSAs) that apply to dynamic interfaces, see *JunosE Broadband Access Configuration Guide*.

Related Documentation

- Configuring a Local Subscriber for a Dynamic IPoA or Bridged Ethernet Interface on page 528
- Configuring a Dynamic Interface over an ATM 1483 Subinterface on page 530

- *auto-configure*
- *configure*
- *interface*
- *macro*
- *profile*
- *subscriber*

Reassigning a Debug Profile Before Troubleshooting PPP and PPPoE Dynamic Interfaces

You can issue the **profile-reassign** command to help you use PPP and PPPoE packet–logging capabilities to debug and troubleshoot PPP and PPPoE dynamic interfaces. To use the **profile-reassign** command, you must access Privileged Exec mode at privilege level 5 or higher.

The **profile-reassign** command enables you to reassign the profile currently assigned to the specified encapsulation type for the specified ATM 1483 subinterface. In effect, you swap the currently assigned nondebug profile for a debug profile that has identical attributes with the addition of one or more PPP or PPPoE packet–logging attributes enabled.

To reassign a profile:

1. Create a debug profile that includes the same attributes as an existing nondebug profile, with the addition of one or more PPP or PPPoE packet–logging attributes enabled.

   Observe the following guidelines when you create the debug profile:

- Because PPP and PPPoE packet logging is performed at log severity 7 (debug priority), configure a destination such as the console to log severity level 7 and issue the **log here** command to enable packet capture using the debug profile you created.
- Before you reassign the debug profile to the ATM 1483 subinterface, make sure that the number of PPP dynamic interfaces has not already exceeded the maximum number of aggregate dynamic and static PPP interfaces for which you can log PPP packets. For more information about this and other system maximums, see *JunosE Release Notes, Appendix A, System Maximums*.

For details about creating and using profiles, see "Dynamic Interface Configuration Using a Profile" on page 565.

2. Access Privileged Exec mode at privilege level 5 or higher.

host1>**enable 5**
Password: *********
host1#

NOTE: The router prompts you for a password only if you have configured a password to control access to Privileged Exec mode. For details about setting passwords to access various command privilege levels, see *Passwords and Security* in *JunosE System Basics Configuration Guide*.

3. From Privileged Exec mode, issue the **profile-reassign** command to replace the nondebug profile currently assigned to the specified encapsulation type for the specified ATM 1483 interface with the debug profile created in Step 1.

You must specify one of the following encapsulation types to which the debug profile applies: **ppp**, **pppoe**, or **any**. You can use the **any** encapsulation type if neither the **ppp** encapsulation type nor the **pppoe** encapsulation type has an existing profile assignment.

NOTE: Issuing the **profile-reassign** command causes the router to tear down any dynamic interfaces that exist above the ATM 1483 subinterface. After the profile is reassigned, the router restores the interfaces based on the necessary client reconnections. If the ATM 1483 subinterface is currently shut down, issuing the **profile-reassign** command does not reestablish the interface connection.

- To facilitate debugging for the **ppp** encapsulation type:

host1#**profile-reassign atm 2/0.101 ppp pppLogConfig**
WARNING: Execution of this command will cause all dynamic interfaces over
  atm 2/0.101 to be torn-down.
Proceed with profile reassignment? [confirm] **yes**
Profile pppConfig replaced by profile pppLogConfig for ppp.

- To facilitate debugging for the **any** encapsulation type:

host1#**profile-reassign atm 3/0.101 any anyLogConfig**
WARNING: Execution of this command will cause all dynamic interfaces over

```
  atm 3/0.101 to be torn-down.
Proceed with profile reassignment? [confirm] yes
Profile anyConfig replaced by profile anyLogConfig for any.
```

4. (Optional) Use the appropriate **show** command to verify the profile reassignment. For example:

   ```
   host1#show atm subinterface atm 2/0.101
   ```

   When you reassign a debug profile to an ATM 1483 subinterface, the reassignment is stored in NVS and persists after a reboot. If you issue the **show atm subinterface** or **show configuration** command after the profile is reassigned, these commands display the new profile assignment.

5. (Optional) To restore the initial (nondebug) profile assignment after you troubleshoot the dynamic interface, issue the **profile-reassign** command again using the name of the nondebug profile.

   ```
   host1#profile-reassign atm 2/0.101 ppp pppConfig
   WARNING: Execution of this command will cause all dynamic interfaces over
     atm 2/0.101 to be torn-down.
   Proceed with profile reassignment? [confirm] yes
   Profile pppLogConfig replaced by profile pppConfig for ppp.
   ```

Related Documentation

- *enable*
- *log here*
- *profile-reassign*
- *show atm subinterface*

CHAPTER 20

Monitoring Upper-Layer Dynamic Interfaces

You can use the **show** commands described in this chapter to monitor configurations created with dynamic interfaces and profiles.

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

- Monitoring Configuration Information of an ATM AAL5 Interface on page 593
- Monitoring Status or Summary Information for ATM Subinterfaces on page 595
- Monitoring Summary Information for ATM VCs and Reserved VC Ranges on page 600
- Monitoring Total Static and Dynamic Interface Counts for Interface Columns on page 602
- Monitoring Summary Information About the Encapsulation Type Lockout for PPPoE Clients on page 603
- Monitoring Detailed Information About the Current Encapsulation Type Lockout Condition for PPPoE Clients on page 604
- Monitoring the Source MAC Address of a PPPoE Client on page 605
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606
- Monitoring Status and Configuration Information for VLAN Subinterfaces on page 614

Monitoring Configuration Information of an ATM AAL5 Interface

Purpose Display information about a configured ATM AAL5 interface.

Action To display information about a configured ATM AAL5 interface:

```
host1#show atm aal5 interface atm 3/0
AAL5 Interface ATM 3/0 operational status:    lowerLayerDown
      time since last status change: 00:08:46
```

```
SNMP trap link-status: disabled
Auto configure ATM 1483 status: disabled

InPackets:        0
InBytes:          0
OutPackets:       0
OutBytes:         0
InErrors:         0
OutErrors:        0
InPacketDiscards: 0
OutDiscards:      0
```

Meaning Table 50 on page 594 lists the **show atm aal5 interface** command output fields.

Table 50: show atm aal5 interface Output Fields

Field Name	Field Description
AAL5 Interface operational status	Operational status of the AAL5 interface: up, down, lowerLayerDown
time since last status change	Time since last reported change to the AAL5 interface operational status in hh:mm:ss format
SNMP trap link-status	Whether SNMP link status traps are enabled or disabled on the ATM AAL5 interface
Auto configure ATM 1483 status	Whether the autoconfiguration feature for a dynamic ATM 1483 subinterface configured over the ATM AAL5 interface is enabled or disabled
InPackets	Number of packets received on this interface
InBytes	Number of bytes received on this interface
OutPackets	Number of packets transmitted on this interface
OutBytes	Number of bytes transmitted on this interface
InErrors	Number of incoming errors received on this interface
OutErrors	Number of outgoing errors on this interface
InPacketDiscards	Number of incoming packets discarded on this interface
OutDiscards	Number of outgoing packets discarded on this interface

Related Documentation

- Creating a PVC on an ATM 1483 Subinterface on page 537
- *show atm aal5 interface*

Monitoring Status or Summary Information for ATM Subinterfaces

Purpose Display the current state of all ATM subinterfaces, all ATM subinterfaces configured on a specified ATM physical interface, or a specific ATM subinterface. You can use the **summary** keyword to display brief summary information for all ATM subinterfaces, or for ATM subinterfaces configured on a specified ATM physical interface. You can use the **status** keyword with one of the following operating status values to display status information only for ATM subinterfaces with a specific operating status:

- dormant
- dormantLockout
- down
- lowerLayerDown
- notPresent
- up

You can use the **atm** *slot/port/vpi/vci* format (for ERX7xx models, ERX14xx models, and ERX310 router) or the *slot/adapter/port/vpi/vci* format (for E120 and E320 routers) to display the current state of an ATM subinterface created on the PVC with the specified VPI and VCI values.

NOTE: You can use the **atm** *slot/port/vpi/vci* format as an alternative to the **atm** *slot/port.subinterface* format with the specific **show interface** and **show subinterface** commands to monitor all ATM 1483 subinterfaces (except NBMA interfaces) as well as the upper-layer interfaces configured over an ATM 1483 subinterface. You cannot, however, use the **atm** *slot/port/vpi/vci* format to create or modify an ATM 1483 subinterface.

These guidelines also apply to E120 and E320 routers when you use the **atm** *slot/adapter/port/vpi/vci* format as an alternative to the **atm** *slot/adapter/port.subinterface* format.

For more information, see "Creating a Basic Configuration" on page 21 in "Configuring ATM" on page 3.

Action To display the current state of all ATM subinterfaces:

```
host1#show atm subinterface
                                Circuit                         Interface
Interface   ATM-Prot VCD VPI VCI Type  Encap MTU    Status      Type
----------- -------- --- --- --- ----- ----- ---- ----------- -------
ATM 2/0.101 RFC-1483 101   0 101 PVC   AUTO  9180 dormantLockout Static
ATM 2/0.102 RFC-1483 102   0 102 PVC   AUTO  9180 up             Dynamic
ATM 2/0.103 RFC-1483 103   0 103 PVC   AUTO  9180 dormant        Static
3 interface(s) found
```


To display summary information for all ATM subinterfaces:

```
host1#show atm subinterface summary
3 subinterfaces: 1 up, 0 down,
1 dormant, 1 dormantLockout,
0 lowerLayerDown, 0 not present
```

To display status information for all ATM subinterfaces in the dormantLockout state:

```
host1#show atm subinterface status dormantLockout
                                Circuit                        Interface
 Interface  ATM-Prot VCD VPI VCI  Type Encap MTU  Status       Type
---------- -------- --- --- --- ------ ----- ---- ------------- ---------
ATM 2/0.101 RFC-1483 101   0 101 PVC   AUTO  9180 dormantLockout Static
1 interface(s) found
```

To display the current state of a specific ATM subinterface:

```
host1#show atm subinterface atm 2/0.101
                                Circuit                        Interface
 Interface  ATM-Prot VCD VPI VCI Type Encap MTU    Status      Type
----------- -------- --- --- --- ----- ----- ---- ------------ ---------
ATM 2/0.101 RFC-1483 101  0 101 PVC   AUTO  9180 dormantLockout Static

Auto configure status         : dynamic
Auto configure interface(s)   : IP  bridgedEthernet  PPP  PPPoE
Detected 1483 encapsulation   : AUTO
Detected dynamic interface    : none
Interface types in lockout    : IP

Lockout state (seconds)       : Min Max  Current Elapsed Next
------------------------------ --- ---- ------- ------- ----
IP                              1   30      16       7   30
BridgedEnet                   900 3600       0       0  900
PPP                             1  300       0       0    1
PPPoE                           1  300       0       0    1

Assigned profile (IP)         : ipoa
Assigned profile (BridgedEnet): beth
Assigned profile (PPP)        : ppptest
Assigned profile (PPPoE)      : pppoetest
Assigned profile (any)        : none assigned

BridgedEnet subscriber info   :
Username: elaine@jpeterman.com
Password: putty
Authenticate: enabled

SNMP trap link-status: disabled

InPackets:                0
InBytes:                  1904
OutPackets:               0
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         14
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found
```


To display the current state of a specific ATM subinterface created on the PVC with the specified VPI and VCI values:

```
host1#show atm subinterface atm 0/0/0/101
                                 Circuit                         Interface
 Interface   ATM-Prot VCD VPI VCI  Type   Encap MTU  Status   Type
----------- -------- --- --- --- ------- ----- ---- ------ ---------
ATM 0/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 up     Static

Auto configure status         : dynamic
Auto configure interface(s)   : PPPoE
Detected 1483 encapsulation   : SNAP
Detected dynamic interface    : PPPoE
Interface types in lockout    : none

Lockout state (seconds)       : Min Max Current Elapsed Next
------------------------------ --- --- ------- ------- ----
PPPoE                            1 300       0       0    1

Assigned profile (IP)         : none assigned
Assigned profile (BridgedEnet): none assigned
Assigned profile (PPP)        : none assigned
Assigned profile (PPPoE)      : pppoeprofile
Assigned profile (any)        : none assigned

SNMP trap link-status: disabled

InPackets:                 5119
InBytes:                   358672
OutPackets:                5107
OutBytes:                  357510
InErrors:                  0
OutErrors:                 0
InPacketDiscards:          3
InPacketsUnknownProtocol:  0
OutDiscards:               0
1 interface(s) found
```

Meaning Table 51 on page 597 lists the **show atm subinterface** command output fields.

Table 51: show atm subinterface Output Fields

Field Name	Field Description
Interface	Interface identifier
ATM-Prot	One of the following ATM protocol types: • RFC-1483—Multiprotocol encapsulation over AAL5 • NBMA—Nonbroadcast multiaccess interface • ATM/MPLS—Local ATM passthrough interface
VCD	Virtual circuit descriptor
VPI	Virtual path identifier
VCI	Virtual circuit (or channel) identifier
Circuit Type	Type of circuit: PVC

Table 51: show atm subinterface Output Fields *(continued)*

Field Name	Field Description
Encap	Administered encapsulation method based on what was configured with the **atm pvc** command
MTU	Maximum transmission unit size for the interface
Status	One of the following ATM 1483 subinterface states: • absent—Represents the notPresent state and indicates that, although the SRP detects the ATM 1483 subinterface, the module on which the subinterface resides has not completed booting up, has failed, or is disabled. • dormant—Indicates that the ATM 1483 subinterface is performing autodetection of one or more upper-layer encapsulation types and is waiting to receive a packet of that type on a lower-layer interface. An ATM 1483 subinterface transitions from the dormant state to the up state when the router receives a valid packet of the specified encapsulation type on the interface. • dormantLockout—Indicates that a dormant ATM 1483 subinterface has one or more upper-layer encapsulation types currently undergoing encapsulation type lockout. An ATM 1483 subinterface transitions from the dormantLockout state to the dormant state when the lockout time expires for all upper-layer encapsulation types undergoing lockout. An ATM 1483 subinterface transitions from the dormantLockout state to the up state when the router receives a valid packet for an encapsulation type that is configured for autodetection but is not undergoing lockout. • down—Indicates that the ATM 1483 subinterface is administratively disabled or has a circuit that is down or not configured. • lowerLayerDown—Indicates that a lower-layer interface below the ATM 1483 subinterface is down. • up—Indicates that the ATM 1483 subinterface is up and able to transfer data. For an ATM 1483 subinterface that supports one or more dynamic upper-layer interfaces, indicates that the router has created the dynamic upper-layer interface or is in the process of creating it.
Interface Type	Type of ATM 1483 subinterface: dynamic or static
Auto configure status	Setting of the autoconfiguration feature: • dynamic—Autodetection is on; the router automatically detects the next upper interface • static—Autodetection is off

Table 51: show atm subinterface Output Fields *(continued)*

Field Name	Field Description
Auto configure interface(s)	Types of dynamic upper interfaces configured with the **auto-configure** command: bridged Ethernet, IP, PPP, or PPPoE
Detected 1483 encapsulation	If the encapsulation type is set to **aal5autoconfig**, displays the 1483 encapsulation type detected on the subinterface (displays AUTO until a packet is detected)
Detected dynamic interface	Type of dynamic upper interface detected during autoconfiguration: bridged Ethernet, IP, PPP, PPPoE, or (if no packet has been received) none
Interface types in lockout	Encapsulation types currently experiencing lockout: bridged Ethernet, IP, PPP, PPPoE, or none
Lockout state (seconds)	Settings of encapsulation type lockout for the upper-layer encapsulation type indicated • Min—Minimum lockout time, in seconds • Max—Maximum lockout time, in seconds • Current—Current lockout time, in seconds; displays 0 (zero) if lockout is not occurring • Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if lockout is not occurring • Next—Lockout time for the router to use for the next lockout event, in seconds
Assigned profile	For each dynamic interface type, indicates whether or not a profile is assigned and, if assigned, displays the profile name
Subscriber info	Subscriber login information for the specified dynamic interface type (bridged Ethernet or IP)
SNMP trap link-status	Trap link status: enabled or disabled
InPackets	Number of packets received on this interface
InBytes	Number of bytes received on this interface
OutPackets	Number of packets transmitted on this interface
OutBytes	Number of bytes transmitted on this interface
InErrors	Number of errors received on this interface
OutErrors	Number of outgoing errors on this interface

Table 51: show atm subinterface Output Fields *(continued)*

Field Name	Field Description
InPacketDiscards	Number of incoming packets discarded on this interface
InPacketsUnknownProtocol	Number of incoming packets with an unknown protocol type
OutDiscards	Number of outgoing packets discarded on this interface

Related Documentation

- Configuring a Dynamic Interface over an ATM 1483 Subinterface on page 530
- Creating a PVC on an ATM 1483 Subinterface on page 537
- Configuring Dynamic PPPoE over Static PPPoE with ATM Interface Columns on page 543
- Configuring a Dynamic IPoA Interface on page 556
- Configuring a Dynamic Bridged Ethernet Interface on page 559
- Assigning a Profile to a Dynamic Interface on page 572
- *atm pvc*
- *auto-configure*
- *show atm subinterface*

Monitoring Summary Information for ATM VCs and Reserved VC Ranges

Purpose Display a summary of all configured ATM VCs and reserved VC ranges. You can specify the **reserved** keyword with no other keywords to display only a summary of all reserved VC ranges on the router. This includes VPI/VCI ranges reserved for use by dynamic ATM 1483 subinterfaces.

Action To display all VCs and reserved VC ranges on the router:

```
host1#show atm vc
                                         Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
ATM 3/0.3      0 102  4376 PVC  AUTO  CBR   1000     0     0 DOWN
...
ATM 3/0.8099   1 8099 8099 PVC  SNAP  UBR      0     0     0 UP
ATM 3/0.8100   1 8100 8100 PVC  SNAP  UBR      0     0     0 DOWN
8000 circuit(s) found

Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
```



```
ATM 2/0       3   300   3 303
2 reservation(s) found
```

To display all reserved VC ranges on the router:

```
host1#show atm vc reserved
Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

Meaning Table 52 on page 601 lists the **show atm vc** command output fields.

Table 52: show atm vc Output Fields

Field Name	Field Description
Interface	Interface identifier
VPI	Virtual path identifier
VCI	Virtual channel identifier
VCD	Virtual circuit descriptor
Type	Type of circuit: PVC
Encap	Encapsulation method: AUTO, AAL5, MUX, SNAP, ILMI, F4-OAM
Category	Service type configured on the VC: UBR, UBR-PCR, NRT-VBR, RT-VBR, or CBR
Rx/Tx Peak	Peak rate, in Kbps
Rx/Tx Avg	Average rate, in Kbps
Rx/Tx Burst	Maximum number of cells that can be burst at the peak cell rate
Status	State of the virtual circuit: Up or Down
Start VPI	Starting virtual path identifier (inclusive) of the reserved VC range
Start VCI	Starting virtual circuit identifier (inclusive) of the reserved VC range
End VPI	Ending virtual path identifier (inclusive) of the reserved VC range

Table 52: show atm vc Output Fields *(continued)*

Field Name	Field Description
End VCI	Ending virtual circuit identifier (inclusive) of the reserved VC range

Related Documentation

- Creating a PVC on an ATM 1483 Subinterface on page 537
- Configuring a PPP or PPPoE Dynamic Interface over an ATM 1483 Subinterface on page 539
- Configuring Dynamic PPPoE over Static PPPoE with ATM Interface Columns on page 543
- Configuring and Verifying Lockout for PPPoE Clients on page 552
- Configuring a Dynamic IPoA Interface on page 556
- Dynamic Bridged Ethernet Interfaces on page 558
- *show atm vc*

Monitoring Total Static and Dynamic Interface Counts for Interface Columns

Purpose Display static and dynamic interface counts for each interface column. Counts for PPP and PPPoE interface columns are updated when the PPP layer comes up. Counts for bridged Ethernet and IP over ATM columns are updated when the ATM layer comes up.

Action To display static and dynamic interface counts for each interface column:

```
host#show columns
                  Interface columns:
                  ------------------
        Type           Total      Static    Dynamic
-------------------- ---------- ---------- ----------
Bridged Ethernet              4          2          2
IP over ATM                   4          2          2
PPP                          22         12         10
PPPoE                        10          5          5
```

Meaning Table 53 on page 602 lists the **show columns** command output fields.

Table 53: show columns Output Fields

Field Name	Field Description
Type	Interface type
Total	Total number of interfaces on this column
Static	Number of static interfaces on this column
Dynamic	Number of dynamic interfaces on this column

Related Documentation

- Configuring a Dynamic Interface over an ATM 1483 Subinterface on page 530
- Dynamic PPP and PPPoE Interfaces over Static ATM on page 538
- Dynamic PPPoE Interfaces over PPPoE Static Interfaces on page 542
- Dynamic Bridged Ethernet Interfaces on page 558
- *show columns*

Monitoring Summary Information About the Encapsulation Type Lockout for PPPoE Clients

Purpose Display summary information about the encapsulation type lockout parameters configured for PPPoE clients on a dynamic PPPoE subinterface column.

Action To display summary information about the encapsulation type lockout parameters configured for PPPoE clients on a dynamic PPPoE subinterface column:

```
host1#show pppoe interface atm 3/0.101
. . .
Lockout Configuration (seconds): Min 5, Max 60
 Total clients in active lockouts: 0
 Total clients in lockout grace period: 0
```

NOTE: The output includes only the portion of the **show pppoe interface** command display relevant to lockout configuration for PPPoE clients.

Meaning Table 54 on page 603 lists the **show pppoe interface** command encapsulation type lockout output fields.

Table 54: show pppoe interface Encapsulation Type Lockout Output Fields

Field Name	Field Description
Lockout Configuration (seconds)	Encapsulation type lockout settings for the PPPoE client associated with the dynamic PPPoE subinterface column: • Min—Minimum lockout time, in seconds • Max—Maximum lockout time, in seconds • Total clients in active lockouts—Number of PPPoE clients currently undergoing dynamic encapsulation type lockout • Total clients in lockout grace period—Number of PPPoE clients currently in a lockout grace period; for more information about the lockout grace period, see "Guidelines for Configuring Encapsulation Type Lockout for PPPoE Sessions" on page 535

Related Documentation

- Configuring and Verifying Lockout for PPPoE Clients on page 552
- *show pppoe interface*

Monitoring Detailed Information About the Current Encapsulation Type Lockout Condition for PPPoE Clients

Purpose Display detailed information about the current encapsulation type lockout condition for each PPPoE client associated with a dynamic PPPoE subinterface column on a static PPPoE major interface.

Action To display detailed information about the current encapsulation type lockout condition for each PPPoE client associated with a dynamic PPPoE subinterface column on a static PPPoE major interface:

```
host1#show pppoe interface atm 12/1/1.1.1 lockout-time
PPPoE interface atm 12/1/1.1
Lockout Configuration (seconds): Min 90, Max 120
 Total clients in active lockout: 1
 Total clients in lockout grace period: 0
Client Address Current Elapsed Next
-------------- ------- ------- ----
0090.1a42.527c     120      30  120
0090.1a42.527c       0       0   90
```

Meaning Table 55 on page 604 lists the **show pppoe interface lockout-time** command output fields.

Table 55: show pppoe interface lockout-time Output Fields

Field Name	Field Description
PPPoE interface	Specifier for the PPPoE interface
Lockout Configuration (seconds)	Encapsulation type lockout settings for the PPPoE client associated with the dynamic PPPoE subinterface column: • Min—Minimum lockout time, in seconds • Max—Maximum lockout time, in seconds • Total clients in active lockouts—Number of PPPoE clients currently undergoing dynamic encapsulation type lockout • Total clients in lockout grace period—Number of PPPoE clients currently in a lockout grace period; for more information about the lockout grace period, see “Guidelines for Configuring Encapsulation Type Lockout for PPPoE Sessions” on page 535

Table 55: show pppoe interface lockout-time Output Fields *(continued)*

Field Name	Field Description
Client Address	Source MAC address of the PPPoE client NOTE: Because PPPoE sessions that contain the IWF-Session DSL Forum VSA (26-154) use the same source MAC address of the DSLAM for all subscriber connections, multiple entries are displayed in the Client Address field for the same MAC address if multiple IWF sessions contain the same MAC address.
Current	Current lockout time, in seconds; displays 0 (zero) if the PPPoE client is not undergoing lockout
Elapsed	Time elapsed into the lockout time, in seconds; displays 0 (zero) if the PPPoE client is not undergoing lockout
Next	Lockout time that the router uses for the next lockout event, in seconds

Related Documentation

- Configuring and Verifying Lockout for PPPoE Clients on page 552
- Clearing the Lockout Condition for a PPPoE Client on page 553

- *show pppoe interface lockout-time*

Monitoring the Source MAC Address of a PPPoE Client

Purpose Display the source MAC address of a PPPoE client when a subscriber is connected to the router through an available PPPoE session. You can use the **full** keyword to display configuration, status, and statistics information, including the source MAC address of the PPPoE client.

Action To display the source MAC address of a PPPoE client when a subscriber is connected to the router through an available PPPoE session:

```
host1#show pppoe subinterface full
...
    PPPoE subinterface ATM 3/0.101 has source MAC address 0090.1a10.165e
...
```

NOTE: The output includes only the portion of the **show pppoe subinterface** command display relevant to the source MAC address for PPPoE clients.

Meaning Table 56 on page 606 lists the **show pppoe subinterface** command source MAC address output fields.

Table 56: show pppoe subinterface Source MAC Address Output Fields

Field Name	Field Description
PPPoE subinterface	Specifier for the PPPoE subinterface
source MAC address	MAC address of the PPPoE client associated with the dynamic PPPoE subinterface column

Related Documentation

- Clearing the Lockout Condition for a PPPoE Client on page 553
- *pppoe clear lockout interface*
- *show pppoe subinterface*

Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces

Purpose Display information about profiles. You can use the **name** keyword to display information about a specific profile. You can use the **brief** keyword to display a list of profiles configured on the router.

Action To display configuration information for a profile assigned to a dynamic interface:

```
host1#show profile name pppoe
PPP Keepalive                 : 30
PPP Magic Number              : enabled
PPP Magic Number Mismatch    :reject
PPP Peer DNS Priority         : disabled
PPP Peer WINS Priority        : disabled
PPP Authentication            :
PPP Authentication Router     :
PPP Negotiate MRU             : (use lower layer MRU)
PPP Packet Log                : disabled
PPP State Log                 : disabled
PPP Chap Challenge Length     : 16 - 32
PPP Passive Mode              : disabled
PPP Multilink                 : disabled
PPP IPCP Netmask Option       : disabled
PPP IPCP Lockout Option       : disabled
PPP IPCP Lockout              : disabled
PPP AAA Profile               :
PPP Multilink Fragmentation : disabled
PPP Multilink Fragment Size : (use MTU)
PPP Multilink Reassembly      : disabled
PPP Multilink Mrru            : (use MRU)
PPP Hash Link Selection       : disabled
PPP Initiate IP               : disabled
PPP Initiate IPv6             : disabled
PPP Max-negotiations LCP      : 30
PPP Max-negotiations IPCP     : 30
PPP Max-negotiations IPv6CP : 30
PPP IPCP prompt-option DNS    : enabled
PPP Multilink Multiclass      : disabled
PPP Multilink Multiclasses    : 0
PPP Ipcp Max Negotiation      : 6
PPP Ipcp Negotiation time     : 60
```



```
PPP Ipcp Lockout time       : 600
PPP broadcast Accounting Group Name     : groupxyz
PPP Client Username          :
PPP Client Password         :
PPP Client Authentication   :
PPP Client Ip Address       : 0.0.0.0
PPP Client Primary Dns Address   : 0.0.0.0
PPP Client Secondary Dns Address   : 0.0.0.0
PPP Client Primary Wins Address   : 0.0.0.0
PPP Client Secondary Wins Address   : 0.0.0.0
```

To display configuration information for a base profile assigned to a dynamic ATM 1483 subinterface:

```
host1#show profile name atm1483BaseProfile
ATM1483 Auto-configure ip                : disabled
ATM1483 Auto-configure bridgedEthernet    : disabled
ATM1483 Auto-configure ppp               : enabled
ATM1483 lockout (seconds) ppp            : range : 1-300
ATM1483 Auto-configure pppoe             : enabled
ATM1483 lockout (seconds) pppoe          : range : 1-300
ATM1483 PVC circuit type         :  aal5autoconfig
ATM1483 PVC service category     :  Nrt-Vbr
ATM1483 PVC Peak rate : 10000, Avg rate : 2000, Burst size : 500
ATM1483 Description              :  VC_atm1483
ATM1483 Advisory Rx Speed        :  2000000000


ATM1483 PVC OAM Administrative status: enabled
ATM1483 PVC OAM Loopback frequency:    30

ATM1483 Ip Subscriber information:
  user              : elaine
  domain            : jpeterman.com
  password          : putty
ATM1483 IP Profile              : none assigned
ATM1483 Bridged Ethernet Profile : none assigned
ATM1483 PPP Profile             : none assigned
ATM1483 PPPoE Profile           : pppoeProfile
```

To display configuration information for a base profile assigned to a dynamic VLAN subinterface:

```
host1#show profile name vlanProfile
VLAN Auto-configure ip             : enabled
VLAN Auto-configure pppoe          : enabled
VLAN Svlan Ethertype               : auto-configure
VLAN Advisory Rx Speed             : 100 Kbps
VLAN Advisory Tx Speed             : 2500 Kbps
VLAN Description                   : testing
VLAN IP Profile                    : ipProfile
VLAN PPPoE Profile                 : pppoeProfile
VLAN Service Profile               : none assigned
Bridged Ethernet Mtu               : 1971
Bridged Ethernet Service Profile   : eastServiceProfile
```

To display profile configuration information related to IPv6 Neighbor Discovery router advertisements:

```
host1#show profile name ipv6Profile
IPv6 Unnumbered interface  : loopback 0
```

```
IPv6 Router                 : default
IPv6 Src-Addr Validation    : Disabled
IPv6 Administrative MTU     : 0
IPv6 ND Enabled             : Enabled
IPv6 ND ManagedConfig       : Disabled
IPv6 ND OtherConfig         : Enabled
IPv6 ND SuppressRa          : Disabled
IPv6 ND RaInterval          : 50
IPv6 ND RaLifeTime          : 1800
IPv6 ND ReachableTime       : 0
IPv6 ND RaPrefix            : 1234::/64
IPv6 ND ValidLifetime       : 60
IPv6 ND PreferredLifetime   : 60
IPv6 ND PrefixOnLink        : Enabled
IPv6 ND PrefixAutoConfig    : Enabled
IPv6 http redirect Url      : http://www.google.com
```

Meaning Table 57 on page 608 lists the **show profile** command output fields.

Table 57: show profile Output Fields

Field Name	Field Description
Profile	Name of the profile that is displayed
IP address	IP address and subnet mask of the interface, or none if the interface is unnumbered
Unnumbered interface	Specifier for the unnumbered interface, or none if the interface is numbered
Router	Name of the virtual router assigned to the profile; interfaces created by the profile are attached to this virtual router
Directed Broadcast	Enabled or disabled
ICMP Redirects	Enabled or disabled
Access Route Addition	Enabled or disabled
Network Address Translation	Enabled or disabled; domain location (inside or outside)
Source-Address Validation	Enabled or disabled
Ignore DF Bit	Enabled or disabled
Filter Option Packets	Router filters out packets with IP options; enabled or disabled
Administrative MTU	MTU size configured on the profile
TCP MSS value	Maximum segment size for TCP SYN packets traveling through the interface

Table 57: show profile Output Fields *(continued)*

Field Name	Field Description
Inactivity Timer	Inactivity timer setting; enabled or disabled
Route Map Name	Route map applied to the IP interface subscriber; enabled or disabled
Auto Detect	Router automatically detects packets that do not match any entries in the demultiplexer table; enabled or disabled
Auto Configure	Dynamic creation of subscriber interfaces on a primary IP interface; enabled or disabled
IGMP	Enabled or disabled
static-groups	Displays address of any static groups configured for IGMP
Input policy	Name of input policy and whether statistics are enabled or disabled
Output policy	Name of output policy and whether statistics are enabled or disabled
PPP Keepalive	PPP keepalive period, in seconds
PPP Magic Number	Enabled or disabled
PPP Magic Number Mismatch	Indicates whether the router is configured to ignore the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number: ignore (ignore the peer magic number mismatch and retain the PPP connection), or reject (router terminates the PPP connection if it detects a peer magic number mismatch)
PPP Peer DNS Priority	Enabled or disabled
PPP Peer WINS Priority	Enabled or disabled
PPP Authentication	Type of authentication configured: PAP, CHAP, or none
PPP Authentication Router	Name of authentication virtual router
PPP Negotiate MRU	MRU configured for the profile
PPP Packet Log	Enabled or disabled
PPP State Log	Enabled or disabled

Table 57: show profile Output Fields *(continued)*

Field Name	Field Description
PPP Chap Challenge Length	Minimum and maximum Chap Challenge length
PPP Passive Mode	Enabled or disabled
PPP Multilink	Enabled or disabled
PPP IPCP Netmask Option	Enabled or disabled
PPP IPCP Lockout Option	Enabled or disabled
PPP Max-negotiations IPCP	Maximum number of renegotiation attempts that the router accepts before terminating a PPP session
PPP AAA Profile	AAA profile associated with this PPP interface
PPP Multilink Fragmentation	Enabled or disabled
PPP Multilink Fragment Size	Multilink fragment size for this PPP interface
PPP Multilink Reassembly	Enabled or disabled
PPP Multilink Mrru	Multilink MRRU value for this PPP interface
PPP Hash Link Selection	Hash-based link selection for a PPP interface; enabled or disabled
PPP Initiate IP	Initiation of IPv4 over this PPP interface; enabled or disabled
PPP Initiate IPv6	Initiation of IPv6 over this PPP interface; enabled or disabled
PPP IPCP prompt-option dns	Prompts the CPE (Customer Premises Equipment) to negotiate the IPCP primary and secondary DNS options that are locally available with the broadband remote access server; enabled or disabled
PPP Multilink Multiclass	Configuration of multiclass MLPPP on the MLPPP interface: enabled or disabled
PPP Multilink Multiclasses	Number of multilink classes created on the MLPPP interface: 1 through 8
PPP Ipcp Max Negotiation	Maximum number of successful IPCP renegotiations for IPv4 addresses that the router can receive per subscriber: 3 through 6

Table 57: show profile Output Fields *(continued)*

Field Name	Field Description
PPP Ipcp Negotiation time	Time period during which IPCP renegotiations for IPv4 addresses that the router or the provider edge device can receive from a subscriber are restricted: 60 through 600 seconds
PPP Ipcp Lockout time	Time period during which additional IPCP negotiations are prevented: 300 through 600 seconds
PPP broadcast Accounting Group Name	Name of the broadcast virtual router group associated with this PPP profile
PPPoE Max Sessions	Maximum number of PPPoE subinterfaces that can be on an interface
PPPoE Always-offer	Router offers to set up a session for the client, even if the router has insufficient resources to establish a session; enabled or disabled
PPPoE Remote-Circuit-Id	The router captures and processes a vendor-specific tag containing a remote circuit ID transmitted from a digital subscriber line access multiplexer; enabled or disabled
PPPoE Log PPPoeControlPacket	Enabled or disabled
PPPOE duplicate-protect	Enabled or disabled
PPPoE ACNAME	Access concentrator name
PPPoE URL	URL sent in PADM message to PPPoE clients
PPPoE MOTM	Message of the minute sent in the PADM message to PPPoE clients
PPPoE Service-Name Table	Name of the PPPoE service name table, if configured for the specified profile
ATM1483 Auto-configure	Whether autodetection of the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE) is enabled or disabled for a dynamic ATM 1483 subinterface
ATM1483 lockout (seconds)	Encapsulation type lockout setting for the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE) configured on a dynamic ATM 1483 subinterface: • range—Minimum lockout time–maximum lockout time, in seconds • no lockout—Encapsulation type lockout is disabled

Table 57: show profile Output Fields *(continued)*

Field Name	Field Description
ATM1483 PVC circuit type	Encapsulation setting for the PVC configured on a dynamic ATM 1483 subinterface: • aal5autoconfig—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed) • aal5mux ip—VC-based multiplexed circuit for IP only • aal5snap—LLC encapsulated circuit; the LLC/SNAP header precedes the protocol datagram
ATM1483 PVC service category	Service type setting for the PVC configured on a dynamic ATM 1483 subinterface: UBR (the default), UBR PCR, NRT-VBR, RT-VBR, or CBR
ATM1483 PVC Peak rate	Peak cell rate, in Kbps, for the PVC configured on a dynamic ATM 1483 subinterfaces
ATM1483 PVC Avg rate	Average rate, in Kbps, for the PVC configured on a dynamic ATM 1483 subinterface; also referred to as sustained cell rate
ATM1483 PVC Burst size	Length in cells of the burst for the PVC configured on a dynamic ATM 1483 subinterface; also referred to as maximum burst size
ATM1483 Description	Text description assigned to ATM 1483 subinterfaces that are created with this profile
ATM1483 Advisory Rx Speed	Configured receive speed, in Kbps, for the dynamic ATM 1483 subinterface. The E Series LAC sends this value to the LNS in the RX Connect-Speed AVP [38].
ATM1483 PVC OAM Administrative status	Status of OAM F5 loopback cell generation (for VC integrity) on a circuit created with this profile: enabled or disabled
ATM1483 PVC OAM Loopback frequency	Number of seconds between transmissions of OAM F5 end-to-end loopback cells on a circuit created with this profile
ATM1483 Ip Subscriber information	Subscriber login information for the specified dynamic interface type
ATM1483 Profile	Name of the profile assigned to the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE); these profiles are referenced in the base profile for a dynamic ATM 1483 subinterface as nested profile assignments
VLAN Auto-configure	Whether auto detection of the specified upper-interface encapsulation type (IP or PPPoE) is enabled or disabled for a dynamic VLAN subinterface

Table 57: show profile Output Fields *(continued)*

Field Name	Field Description
VLAN Advisory Rx Speed	Configured advisory receive speed, in Kbps, for the dynamic VLAN subinterface; the E Series LAC sends this value to the LNS in the RX Connect-Speed AVP [38]
VLAN Advisory Tx Speed	Configured advisory speed, in Kbps, for the dynamic VLAN subinterface.
VLAN Description	Text description assigned to VLAN subinterfaces that are created with this profile
VLAN Profile	Name of the profile assigned to the specified upper-interface encapsulation type (IP or PPPoE); these profiles are referenced in the base profile for a dynamic VLAN subinterface as nested profile assignments
VLAN Service Profile	Service profile name for a VLAN
VLAN Svlan Ethertype	Ethertype that the packet must use this to create the dynamic VLAN subinterface
Bridged Ethernet Mtu	MTU size configured for a dynamic bridged Ethernet interface
Bridged Ethernet Service Profile	Name of the IP service profile associated with the interface profile for this dynamic bridged Ethernet interface
IPv6 http redirect Url	URL to which a subscriber's initial web browser session is redirected
IPv6 Unnumbered interface	Name of interface without a specific address
IPv6 Router	Router name or default
IPv6 Src-Addr Validation	Source-Address Validation; enabled or disabled
IPv6 Administrative MTU	MTU size
IPv6 ND Enabled	State of the Neighbor Discovery; enabled or disabled
IPv6 ND ManagedConfig	State of the Neighbor Discovery router advertisement managed flag; enabled or disabled
IPv6 ND OtherConfig	State of the Neighbor Discovery router advertisement other config flag; enabled or disabled
IPv6 ND SuppressRa	Status IPv6 router advertisement suppression; enabled or disabled

Table 57: show profile Output Fields *(continued)*

Field Name	Field Description
IPv6 ND RaInterval	Interval (in seconds) of the Neighbor Discovery router advertisement
IPv6 ND RaLifeTime	Lifetime (in seconds) of the Neighbor Discovery router advertisement
IPv6 ND ReachableTime	Amount of time (in milliseconds) that the neighbor is expected to remain reachable
IPv6 ND RaPrefix	Configured prefixes for Neighbor Discovery router advertisement
IPv6 ND ValidLifetime	Amount of time (in seconds) that the router advertises the IPv6 prefix as valid
IPv6 ND PreferredLifetime	Amount of time (in seconds) that the router advertises the specified IPv6 prefix as preferred
IPv6 ND PrefixOnLink	State of the on-link flag; enabled or disabled
IPv6 ND PrefixAutoConfig	State of the use the specified prefix for IPv6 autoconfiguration; enabled or disabled

Related Documentation

- Configuring Subscriber Management for IP Subscribers on Dynamic Bridged Ethernet Interfaces on page 561
- Creating a Profile for Dynamic Interfaces on page 571
- Configuring Profile Characteristics on page 576

- *show profile*

Monitoring Status and Configuration Information for VLAN Subinterfaces

Purpose

Display configuration and status information for a specified VLAN subinterface or for all VLAN subinterfaces configured on the router. You can use the **summary** keyword to display only the counts of all VLAN subinterfaces and VLAN major interfaces configured on the router. You can use the **vlan** or **svlan** keywords to display information about specific VLAN IDs or S-VLAN IDs.

Use the **agent-circuit-identifier** keyword to display information about VLAN subinterfaces that are created based on the agent-circuit-id information in the option 82 field of DHCP messages or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets. Using this keyword causes the router to display the agent-circuit-identifier string in the command output.

Action To display full status and configuration information for all VLAN subinterfaces configured on the router:

```
host1#show vlan subinterface
        Interface        Status   MTU  Svlan Id Vlan Id  Ethertype  Type
--------------------- ---------- ---- -------- -------- --------- -------
ATM 3/0.1.2               Up       1522 ----     11       ----      Static
ATM 3/0.1.3               Up       1522 ----     12       ----      Static
ATM 3/1.1.1               Up       1522 ----     13       ----      Static
ATM 3/1.1.2               Up       1522 ----     14       ----      Static
ATM 3/2.1.1               Down     1526 4        255      0x9100    Static
FastEthernet 4/5.1        Up       1522 ----     1        ----      Dynamic
6 vlan subinterfaces found
```

To display full status and configuration information for the specified VLAN subinterface:

```
host1#show vlan subinterface fastEthernet 4/5.1
        Interface       Status   MTU  Svlan Id Vlan Id  Ethertype  Type
---------------------- -------- ---- -------- -------- --------- -------
FastEthernet 4/5.1       Up     522 ----       1       ----      Dynamic
1 vlan subinterface found
```

To display full status and configuration information for the specified S-VLAN ID:

```
host1#show vlan subinterface svlan id 100 53
        Interface       Status   MTU  Svlan Id Vlan Id Ethertype  Type
---------------------- -------- ---- -------- -------- --------- -------
FastEthernet 0/0.1       Up     1526 100      53       0x9100    Static
FastEthernet 4/6.1       Up     1526 100      53       0x9100    Dynamic
2 vlan subinterfaces found
```

To display full status and configuration information for the specified dynamic VLAN subinterface:

```
host1#show vlan subinterface fastEthernet 4/6.1000053
        Interface         Status  MTU  Svlan Id Vlan Id Ethertype Type
------------------------- ------ ---- -------- -------- --------- -------
FastEthernet 4/6.1000053  Up      1526 100      53       0x9100   Dynamic
Auto configure interface(s)   : IP  PPPoE
Detected dynamic interface    : PPPoE
Interface types in lockout    : none
Lockout state (seconds)       : Min Max Current Elapsed Next
------------------------------ --- --- ------- ------- ----
IP                              1 300       0       0    1
PPPoE                           1 300       0       0    1
  In: Bytes 1040, Packets 15
   Multicast 0, Broadcast 1
   Errors 0, Discards 0
  Out: Bytes 984, Packets 15
   Multicast 0, Broadcast 1
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

To display status information for dynamic VLAN subinterfaces that are created based on agent-circuit-identifier information:

```
host1#show vlan subinterface
     Interface          Status  MTU  Svlan Id Vlan Id  Ethertype   Type
```

```
--------------------- -------- ---- -------- -------- --------- ---------
FastEthernet 4/0.1       Up     1522  2       ----     ----      Dynamic *
FastEthernet 4/0.2       Up     1522  2       ----     ----      Dynamic *
2 vlan subinterfaces found
* Created via agent circuit identifier
host1#show vlan subinterface agent-circuit-identifier
        Interface         Svlan Id Agent-Circuit-Identifier
------------------------- -------- --------------------------
FastEthernet 4/0.1        2        ----
FastEthernet 4/0.2        2        0200D0CB2729E5
```

Meaning Table 58 on page 616 lists the **show vlan subinterface** command output fields.

Table 58: show vlan subinterface Output Fields

Field Name	Field Description
Interface	Type and specifier of the VLAN subinterface
Status	Status of the VLAN subinterface: up, down, dormant, lowerLayerDown, absent
MTU	Maximum allowable size (in bytes) of the MTU for the VLAN subinterface
Svlan Id	S-VLAN ID value, if configured
Vlan Id	VLAN ID value for the VLAN subinterface
Ethertype	S-VLAN Ethertype value, if configured
Type	Type of VLAN subinterface: • Static—VLAN or S-VLAN subinterface was configured statically • Dynamic—VLAN or S-VLAN subinterface was configured dynamically
Auto configure interface(s)	Types of dynamic upper interfaces configured with the **auto-configure** command: IP or PPPoE
Detected dynamic interface	Type of dynamic upper interface detected during autoconfiguration: IP, PPPoE, or (if no packet has been received) none
Interface types in lockout	Encapsulation types currently experiencing lockout: IP, PPPoE, or none

Table 58: show vlan subinterface Output Fields *(continued)*

Field Name	Field Description
Lockout state (seconds)	Settings of encapsulation type lockout for the upper-layer encapsulation type indicated: • Min—Minimum lockout time, in seconds • Max—Maximum lockout time, in seconds • Current—Current lockout time, in seconds; displays 0 (zero) if lockout is not occurring • Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if lockout is not occurring • Next—Lockout time for the router to use for the next lockout event, in seconds
In	Analysis of inbound traffic on this interface: • Bytes—Number of bytes received on the VLAN or S-VLAN subinterface • Packets—Sum of all unicast, broadcast, and multicast packets received on the VLAN or S-VLAN subinterface • Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface • Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface • Errors—Total number of errors in all received packets; some packets might contain more than one error • Discards—Total number of discarded incoming packets
Out	Analysis of outbound traffic on this interface: • Bytes—Number of bytes sent on the VLAN or S-VLAN subinterface • Packets—Number of packets sent on the VLAN or S-VLAN subinterface • Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface • Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface • Errors—Total number of errors in all transmitted packets; some packets might contain more than one error • Discards—Total number of discarded outgoing packets

Table 58: show vlan subinterface Output Fields *(continued)*

Field Name	Field Description
ARP Statistics	Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface: • ARP requests—Number of ARP requests • ARP responses—Number of ARP responses • Errors—Total number of errors in all ARP packets • Discards—Total number of discarded ARP packets
Total VLAN interfaces	Total numbers of VLAN subinterfaces and VLAN major interfaces configured on the router; only this field appears when you specify the **summary** keyword

Related Documentation

- Configuring VLAN Characteristics for a Profile on page 588
- *auto-configure*
- *show vlan subinterface*

CHAPTER 21

Configuring Dynamic Interfaces Using Bulk Configuration

This topic explains dynamic interfaces and describes the procedures for configuring them on E Series routers.

This topic contains the following sections:

- Configuring Dynamic Interfaces Using Bulk Configuration Overview on page 620
- Configuring Dynamic Interfaces Using Bulk Configuration Platform Considerations on page 624
- Configuring Dynamic Interfaces Using Bulk Configuration References on page 625
- Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview on page 626
- ATM Base Profile Overview on page 627
- Bulk Configuration of VC Ranges Overview on page 629
- Bulk Configuration and VC Classes Overview on page 631
- Dynamic ATM 1483 Subinterface Creation Overview on page 632
- Overriding ATM 1483 Base Profile Assignments Overview on page 632
- Static ATM Interfaces Within VC Subranges Overview on page 633
- Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations Overview on page 634
- Subscriber Authentication on Dynamic Bridged Ethernet over Dynamic ATM Interfaces on page 635
- Configuring a Dynamic ATM 1483 Subinterface on page 636
- Creating a Base Profile to Configure Attributes for a Dynamic ATM 1483 Subinterface on page 639
- Configuring a Base Profile for Dynamic ATM 1483 Subinterface on page 640
- Creating a Bulk-Configured VC Range on a Static ATM AAL5 Interface on page 644
- Configuring the Static ATM AAL5 Interface to Support Autodetection of an ATM 1483 Dynamic Encapsulation Type on page 646
- Configuring Overriding Profile Assignments on page 646

- Assigning the Base Profile Configured for a Dynamic ATM 1483 Subinterface to the VC Range Configured on a Static ATM AAL5 Interface on page 649
- Assigning an Overriding Profile to a Single ATM PVC that Exists Within a Bulk-Configured VC Subrange on page 650
- Disabling a Dynamic ATM 1483 Interface on page 651
- Changing VC Subranges on page 651
- Configuring Static ATM Interfaces Within VC Subranges on page 655
- Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview on page 657
- VLAN Base Profile Overview on page 660
- Bulk Configuration of VLAN Ranges Overview on page 662
- Bulk Configuration of VLAN Ranges Using Agent-Circuit-Identifier Information Overview on page 663
- Dynamic VLAN Subinterface Creation Overview on page 665
- Overriding VLAN Base Profile Assignments Overview on page 665
- Static VLAN Subinterfaces Within VLAN Subranges Overview on page 666
- Configuring Dynamic VLAN Subinterfaces on page 667
- Creating a Base Profile to Configure Attributes for a Dynamic VLAN Subinterface on page 669
- Configuring a Base Profile for VLAN Subinterface on page 669
- Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface on page 674
- Configuring the Static VLAN Major Interface to Support Autodetection of a VLAN Dynamic Interface Type on page 675
- Configuring Dynamic VLAN Subinterfaces Based on Agent Circuit Identifier Information on page 676
- Configuring Overriding Profile Assignments for VLAN Major Interfaces on page 678
- Assigning the Base Profile Configured for a Dynamic VLAN Subinterface to the Single-Tagged VLAN IDs or Double-Tagged S-VLAN IDs on page 681
- Assigning an Overriding Profile to a Single-Tagged VLAN ID or Double-Tagged S-VLAN ID on page 682
- Configuring VLAN as the Encapsulation Method for the Interface on page 682
- Disabling a Dynamic VLAN subinterface on page 683
- Changing VLAN Subranges on page 683
- Configuring Static VLAN Subinterfaces Within VLAN Subranges on page 688

Configuring Dynamic Interfaces Using Bulk Configuration Overview

Like upper-layer dynamic interfaces, bulk-configured dynamic interfaces are created automatically and transparently through the receipt of data over a lower-layer link, such as an ATM virtual circuit (VC) or a virtual LAN (VLAN) using autodetection. The layers of a dynamic interface are created based on the packets received on the link and can be configured through any one of the following:

- RADIUS authentication (through PPP or ATM 1483)
- Profiles
- A combination of RADIUS authentication and profiles

You create and configure each layer of a static interface manually through an existing configuration mechanism such as the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

For more information about dynamic interfaces, autodetection, and RADIUS, see "Dynamic Interfaces Overview" on page 519.

This topic describes the following:

- Bulk Dynamic Interface Configurations on page 621
- Profiles on page 622
- ATM Oversubscription for Bulk-Configured VC Ranges on page 622

Bulk Dynamic Interface Configurations

E Series routers support dynamic interfaces on two types of static interfaces: ATM and VLAN. This section provides configuration information for ATM and then for VLANs.

E Series routers support dynamic ATM 1483 subinterfaces over static ATM interfaces.

E Series routers support the following types of dynamic interfaces over VLAN major interfaces:

- Dynamic VLAN subinterface over static VLAN major interface
- IP over dynamic VLAN subinterface
- IP over PPPoE over dynamic VLAN subinterface

Internet Protocol version 4 (IPv4) is supported for all bulk-configured dynamic interface columns over dynamic ATM 1483 subinterfaces and over dynamic VLAN subinterfaces.

Currently, Internet Protocol version 6 (IPv6) is supported only when PPP or MLPPP is the layer immediately below the IPv6 layer in the interface column. IPv6 is *not* supported directly over dynamic ATM 1483, dynamic bridged Ethernet, or dynamic VLANs. Bulk-configured dynamic interface columns that support IPv6 include the following:

- Dynamic IPv6 over dynamic PPP over dynamic ATM 1483
- Dynamic IPv6 over dynamic MLPPP over dynamic ATM 1483
- Dynamic IPv6 over dynamic PPP over dynamic PPPoE over dynamic ATM 1483
- Dynamic IPv6 over dynamic MLPPP over dynamic PPPoE over dynamic ATM 1483
- Dynamic IPv6 over dynamic PPP over dynamic PPPoE over dynamic VLAN
- Dynamic IPv6 over dynamic MLPPP over dynamic PPPoE over dynamic VLAN

For more information about IPv4, see *Configuring IP* in *JunosE IP, IPv6, and IGP Configuration Guide*. For more information about IPv6, see *Configuring IPv6* in *JunosE IP, IPv6, and IGP Configuration Guide*.

Profiles

You can use profiles to configure dynamic interfaces over ATM and VLAN interfaces. A *profile* is a set of characteristics that can be dynamically assigned to interfaces. By using a profile, you reduce the management of a large number of interfaces by applying a set of characteristics to multiple interfaces.

When you are configuring a large number of interfaces with the same attributes at the higher layers, you can use a profile to factor out all the common attributes of each layer into one place. This action affects one or more dynamic layers of the interface column. After you define the static lower layers, you assign a profile to the highest static layer of the interface column.

When a dynamic interface is configured, the configuration data received from the RADIUS authentication server typically overrides configuration data obtained from a profile.

The **atm atm1483 auto-configure** command specifies the types of dynamic upper-interface encapsulations that are accepted or detected by a dynamic ATM 1483 subinterface. For flexibility, the router provides the ability to configure an ATM 1483 subinterface with distinct profile assignments for each encapsulation type supported by the **atm atm1483 auto-configure** command. For more information about using this command, see "Specifying the Types of Dynamic Upper-Interface Encapsulations Accepted or Detected by a Dynamic ATM 1483 Subinterface" on page 640.

In contrast to dynamic ATM 1483 subinterfaces, dynamic VLAN subinterfaces support recognition and creation of simultaneous IP and PPPoE upper dynamic interface types. The **vlan auto-configure** command identifies the encapsulation type. For flexibility, the router provides the ability to configure a VLAN subinterface with distinct profile assignments for each encapsulation type supported by the **vlan auto-configure** command. For more information about using this command, see "Specifying the Types of Dynamic Upper-Interface Encapsulations that are Accepted or Detected by a Dynamic VLAN Subinterface" on page 670.

For more information about configuring profiles, see "Dynamic Interface Configuration Using a Profile" on page 565.

ATM Oversubscription for Bulk-Configured VC Ranges

You can take advantage of oversubscription of bulk-configured ATM VCs. The router supports oversubscription of bulk-configured VC ranges when you create a bulk-configured VC range on a static ATM AAL5 interface for use by a dynamic ATM 1483 subinterface.

Oversubscription of bulk-configured VC ranges works in a similar, but not identical, manner to oversubscription of static ATM 1483 subinterfaces that support dynamic upper-layer encapsulation types. For more information, see *ATM Oversubscription for Dynamic Interfaces* in "Dynamic Interfaces Overview" on page 519.

Bulk-Configured VC Ranges

An active bulk-configured VC range is associated with a dynamic ATM 1483 subinterface that supports a dynamic upper-layer encapsulation type. For ATM line modules that support VC oversubscription, the maximum number of active bulk-configured VCs per line module is less than the maximum number of individual VCs created from the total number of bulk-configured VC ranges that the line module supports. For information about configuring dynamic ATM 1483 subinterfaces with bulk-configured VC ranges, see "Bulk Configuration of VC Ranges Overview" on page 629.

When the maximum number of active bulk-configured VCs has been reached, the router prevents all additional subscribers associated with the remaining inactive bulk-configured VCs from connecting to the line module until one of the following conditions occurs:

- At least one currently active subscriber logs out, which causes the router to tear down the dynamic interface column for that subscriber. Although the dynamic ATM 1483 subinterface and its associated VC remain configured on the router, the subinterface becomes inactive and can be replaced by one of the bulk-configured VCs waiting to become active.
- The router tears down at least one dynamic interface column in its entirety, which involves administratively shutting down the associated dynamic ATM 1483 subinterface.

When either of these conditions occurs, the router enables the first inactive bulk-configured VC that receives traffic to connect to the router as a replacement for the subscriber that logged out.

For example, consider an ATM line module that supports a maximum of 32,000 individual VCs created from bulk-configured VC ranges, of which only 8000 VCs can be active at any one time. If all 32,000 bulk-configured VCs attempt to connect to the router, only the first 8000 VCs to receive traffic are able to log in, generate dynamic subinterface columns, and become active. When a subscriber connected through one of these active VCs logs out, the router enables the first of the remaining 24,000 inactive bulk-configured VCs that receives traffic to connect. The router replaces the inactive dynamic ATM 1483 subinterface and associated VC that remain after the subscriber logout with a new dynamic ATM 1483 subinterface and its newly activated circuit.

Combination of Static ATM 1483 Subinterfaces and Bulk-Configured VC Ranges

ATM line modules are sometimes configured with a combination of static ATM 1483 subinterfaces and bulk-configured VC ranges. In these configurations, both the static ATM 1483 subinterfaces and bulk-configured VC ranges can support active subinterfaces. The combined total of active static ATM 1483 subinterfaces, and active dynamic ATM 1483 subinterfaces created from bulk-configured VC ranges, cannot exceed the maximum number of active subinterfaces supported by the line module.

The number of active dynamic subinterfaces created from the bulk-configured VC ranges is limited by both of the following:

- The number of static ATM subinterfaces that exist on the line module, which cannot exceed the maximum number of configured ATM 1483 subinterfaces supported on the line module.
- The number of static ATM subinterfaces that are active on the line module, which cannot exceed the maximum number of active ATM 1483 subinterfaces supported on the line module.

For example, consider an ATM line module that supports a maximum of 8000 active ATM 1483 subinterfaces. The module has 4000 static ATM 1483 subinterfaces configured, all of which are active, and 8000 individual VCs created from bulk-configured VC ranges. Because the 4000 static ATM 1483 subinterfaces are already active, the router enables only 4000 of the bulk-configured VCs to create dynamic ATM 1483 subinterface columns and become active, yielding a combined total of 8000 active subinterfaces on the line module. The router prevents the remaining 4000 inactive bulk-configured VCs from connecting and becoming active until at least one subscriber connected through an active ATM subinterface logs out, thereby making the associated subinterface inactive and eligible for replacement.

Related Documentation

- Configuring Dynamic Interfaces Using Bulk Configuration Platform Considerations on page 624
- Configuring Dynamic Interfaces Using Bulk Configuration References on page 625

Configuring Dynamic Interfaces Using Bulk Configuration Platform Considerations

You can configure dynamic interfaces on the following E Series Broadband Services Routers:

- E120 router
- E320 router
- ERX1440 router
- ERX1410 router
- ERX710 router
- ERX705 router
- ERX310 router

Module Requirements

For information about the modules that support dynamic interfaces on ERX14xx models, ERX7xx models, and the ERX310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support dynamic interfaces.

For information about the modules that support dynamic interfaces on the E120 and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support dynamic interfaces.

Interface Specifiers

The configuration task examples in this topic use the *slot/port*[.*subinterface*] format to specify the physical interface that you want to configure to support dynamic interfaces. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 router, use the *slot/port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E Series routers, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.

Related Documentation

- Configuring Dynamic Interfaces Using Bulk Configuration Overview on page 620
- Configuring Dynamic Interfaces Using Bulk Configuration References on page 625

Configuring Dynamic Interfaces Using Bulk Configuration References

For more information about RADIUS, consult the following resources:

- DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006)
- RFC 2865—Remote Authentication Dial In User Service (RADIUS) (June 2000)
- RFC 2866—RADIUS Accounting (June 2000)
- RFC 3046—DHCP Relay Agent Information Option (January 2001)

Related Documentation

- Configuring Dynamic Interfaces Using Bulk Configuration Overview on page 620

- Configuring Dynamic Interfaces Using Bulk Configuration Platform Considerations on page 624

Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview

E Series routers support configuration of dynamic ATM 1483 subinterfaces over static ATM AAL5 interfaces over ATM. The dynamic ATM 1483 subinterface can perform autodetection and dynamic creation of the following upper-layer encapsulation types:

- Bridged Ethernet
- IP
- PPP
- PPPoE

When you use dynamic interfaces over static ATM 1483 subinterfaces, you must configure the ATM interface and each ATM 1483 subinterface, including the ATM PVC and the attributes of the subinterface. Subinterface attributes include profile assignments, autoconfiguration settings, and subscriber configurations.

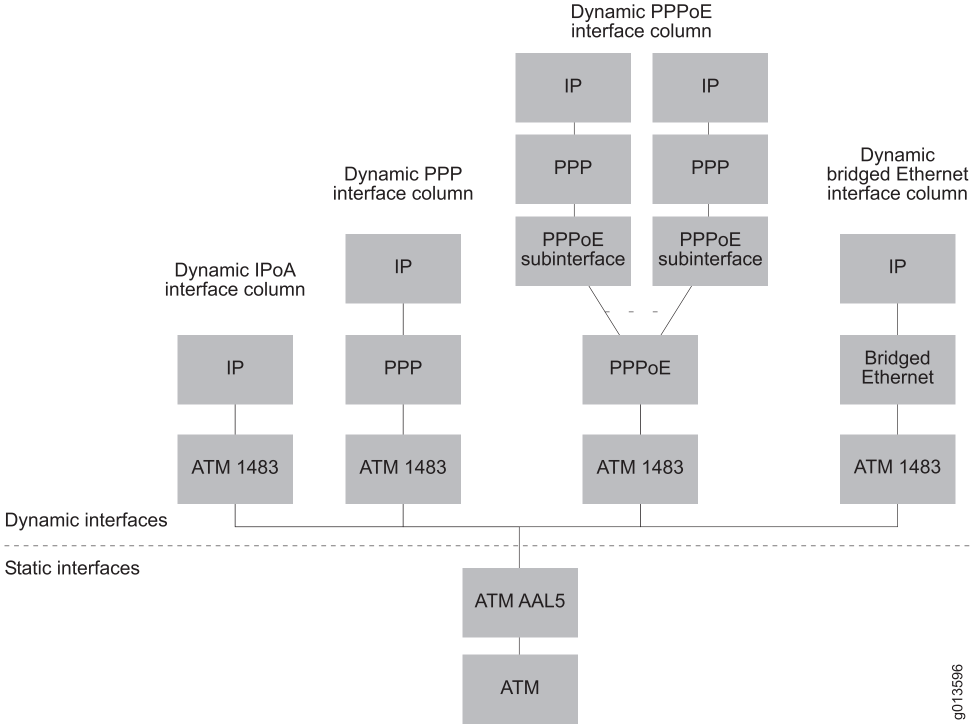
By contrast, when you use dynamic ATM 1483 subinterfaces over static ATM AAL5 interfaces, you use a process called *bulk configuration* to configure a range of ATM PVCs that support dynamic interfaces. On receipt of an incoming packet on the virtual circuit, the router dynamically creates the ATM 1483 subinterface. As part of the configuration process, you create an ATM 1483 base profile, which can optionally include nested profile assignments, to define the attributes required to configure the dynamic ATM 1483 subinterface and the dynamic upper-layer encapsulation types built over it.

Bulk configuration provides an efficient and timesaving way to specify a range of ATM PVCs for dynamic ATM 1483 subinterfaces. Because bulk configuration requires significantly less configuration of the router, it results in reduced output when you issue the **show configuration** command to display the current router configuration.

Dynamic ATM 1483 subinterfaces function identically to static ATM 1483 subinterfaces, except for the manner in which they are created and configured. The creation of dynamic upper-layer encapsulation types is essentially the same regardless of whether they are configured over static ATM 1483 subinterfaces or dynamic ATM 1483 subinterfaces.

Figure 60 on page 627 shows the dynamic upper-interface columns supported by dynamic ATM 1483 subinterfaces, and indicates which layers in the columns are static and dynamic.

Figure 60: Dynamic Interface Columns over Dynamic ATM 1483 Subinterfaces



Related Documentation

- Configuring Dynamic Interfaces Using Bulk Configuration Overview on page 620
- Bulk Configuration of VC Ranges Overview on page 629
- Bulk Configuration and VC Classes Overview on page 631
- Dynamic ATM 1483 Subinterface Creation Overview on page 632
- Configuring a Dynamic ATM 1483 Subinterface on page 636

ATM Base Profile Overview

To configure a dynamic ATM 1483 subinterface over a static ATM AAL5 interface, you must create a base profile. The base profile includes one or more of the following attributes for the ATM 1483 subinterface, listed alphabetically:

- advisory-rx-speed—Sets an advisory receive speed for ATM 1483 subinterfaces that are created with this base profile. For information, see "Configuring a Base Profile to Set Up an Advisory Receive Speed for a Dynamic ATM 1483 Subinterface" on page 640.
- atm pvc—Applies encapsulation, traffic-shaping, and OAM parameters to the range of ATM PVCs configured on the ATM AAL5 interface for use by the dynamic ATM 1483 subinterface. For information, see "Configuring a Base Profile to Apply Encapsulation, Traffic-Shaping and OAM Parameters to a Bulk Range of PVCs Configured on an ATM AAL5 Interface" on page 643.
- auto-configure—Specifies the types of upper-interface encapsulations that are accepted or detected by the dynamic ATM 1483 subinterface. For information, see

"Specifying the Types of Dynamic Upper-Interface Encapsulations Accepted or Detected by a Dynamic ATM 1483 Subinterface" on page 640.

- atm class-vc—Specifies the VC class assigned to the bulk-configured VC ranges created on the dynamic ATM 1483 subinterfaces associated with this base profile. For information, see "Assigning a Previously Configured VC Class to a Base Profile" on page 642.
- description—Assigns a description to ATM 1483 subinterfaces that are created with this base profile. For information, see "Configuring a Base Profile to Assign a Text Description for a Dynamic ATM 1483 Subinterface" on page 641. You can then set up the router to send this description to AAA by using the **atm atm1483 export-subinterface-description** command, as described in "Sending Interface Descriptions to AAA" on page 41 in "Configuring ATM" on page 3.
- profile—Adds a nested profile assignment, which references another profile that dynamically configures an upper-interface encapsulation type over the ATM 1483 subinterface. For information, see "Adding a Nested Profile Assignment to a Base Profile for a Dynamic ATM 1483 Subinterface" on page 641.
- subscriber—Configures a local subscriber for a dynamic upper-interface encapsulation type. For information, see "Configuring a Local Subscriber for a Dynamic Upper-Interface Encapsulation Type Configured over a Dynamic ATM 1483 Subinterface" on page 641.

You can override the base profile assignment for a single ATM PVC that exists within a bulk-configured VC subrange with a profile that includes debugging attributes. This feature is useful for troubleshooting problems with the ATM 1483 dynamic subinterface columns created on the specified PVC. For more information, see "Overriding ATM 1483 Base Profile Assignments Overview" on page 632.

This topic describes the following:

- Nested Profile Assignments Overview on page 628
- Additional Profile Characteristics for Upper Interfaces on page 629

Nested Profile Assignments Overview

The configuration for each dynamic upper-interface encapsulation type might differ, depending on the column type built by the router. To manage these differences, you can include one or more nested profile assignments within the ATM 1483 base profile. A nested profile assignment references another profile that configures attributes for a dynamic upper-interface encapsulation type. You can create different profiles for each upper-interface encapsulation type, or you can create a single profile that includes attributes for multiple encapsulation types.

For example, the following commands create a base profile named atm1483BaseProfile with two nested profile assignments. The first nested profile assignment references an IP profile named atm1483ProfileIp, and the second nested profile assignment references a PPP profile named atm1483ProfilePpp.

host1(config)#**profile atm1483BaseProfile**
host1(config-profile)#**atm atm1483 profile ip atm1483ProfileIp**
host1(config-profile)#**atm atm1483 profile ppp atm1483ProfilePpp**

In this example, atm1483ProfileIp and atm1483ProfilePpp have different IP configurations depending on the dynamic interface column constructed. For an IP over ATM (IPoA) dynamic interface column, the router uses the IP attributes in atm1483ProfileIp. For an IP over PPP over ATM dynamic interface column, the router uses the IP attributes in atm1483ProfilePpp.

The concepts that apply to profiles created for upper-interface encapsulation types configured over static ATM 1483 subinterfaces also apply to profiles created for upper-interface encapsulation configured over dynamic ATM 1483 subinterfaces. For information about creating profiles for upper-interface encapsulation types, see "Creating a Profile for Dynamic Interfaces" on page 571.

Additional Profile Characteristics for Upper Interfaces

In addition to ATM 1483 attributes and nested profile assignments, the base profile for a dynamic ATM 1483 subinterface can also include individual characteristics for several upper-interface encapsulation types, provided that no nested profile assignment for the specified encapsulation type is in the base profile. If, on the other hand, a nested profile assignment for this encapsulation type exists in the base profile, the router obtains all characteristics for that encapsulation type from the nested profile and not from the base profile.

For lists of the characteristics for each supported upper-interface encapsulation type, see "Profile Characteristics" on page 566.

Related Documentation

- Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview on page 626
- Creating a Base Profile to Configure Attributes for a Dynamic ATM 1483 Subinterface on page 639
- Configuring a Base Profile for Dynamic ATM 1483 Subinterface on page 640
- *profile atm1483 bulk-config-name*
- *atm atm1483 profile*

Bulk Configuration of VC Ranges Overview

When you create a static ATM 1483 subinterface, you must configure a permanent virtual circuit (PVC), also known as a virtual circuit (VC). The ATM protocol requires one or more VCs over which data traffic is transmitted to higher layers in the protocol stack.

Similarly, dynamic creation of ATM 1483 subinterfaces requires you to configure a range of ATM PVCs on the ATM AAL5 interface and assign a name to this range. Each VC range consists of one or more nonoverlapping VC subranges. A VC subrange is a group of VCs that resides within the virtual path identifier (VPI) and virtual circuit identifier (VCI) ranges you specify.

The process of configuring a VC range for a dynamic ATM 1483 subinterface is referred to as *bulk configuration*. You create a bulk configuration by issuing the **atm bulk-config** command. For example, the following commands create an ATM 1483 bulk configuration named myBulkConfig on the specified ATM AAL5 interface.

```
host1(config)#interface atm 2/0
host1(config-if)#atm bulk-config myBulkConfig vc-range 0 3 101 1100
  vc-range 4 7 201 700
```

In this example, the **atm bulk-config** command configures a VC range made up of two VC subranges. The first subrange, with VPIs 0–3 and VCIs 101–1100, configures 1000 VCs on each of four VPIs, for a total of 4000 VCs. The second subrange, with VPIs 4–7 and VCIs 201–700, configures 500 VCs on each of four VPIs, for a total of 2000 VCs. The entire myBulkConfig VC range configures a combined total of 6000 VCs.

NOTE: For information about the maximum number of ATM 1483 bulk configurations supported per router, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

After you issue the **atm bulk-config** command, the router provisions all circuits in the specified VC range at the same time. This provisioning can take several seconds, depending on the number of VCs being created. The router does not dynamically create the ATM 1483 subinterface for the circuit until it receives incoming data traffic on the circuit.

After you create a named VC range, you cannot remove the underlying ATM AAL5 interface until you issue the **no atm bulk-config** command to remove the VC range from that interface.

NOTE: For information about the maximum number of VCs (sum of the VPI/VCI addresses within all VC subranges) that you can configure with the **atm bulk-config** command per line module and per chassis, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

Do not use any reserved VCI values when configuring VCs with the **atm bulk-config** command. For information about reserved VCIs, see "Configuring F4 OAM" on page 32 in "Configuring ATM" on page 3.

This topic describes the following:

- Bulk Configuration and CAC Overview on page 630
- Changing VC Subranges Overview on page 631

Bulk Configuration and CAC Overview

You cannot create a bulk-configured VC range on an ATM interface on which you have configured connection admission control (CAC). Conversely, you cannot configure CAC on an ATM interface on which you have created a bulk-configured VC range.

If you are upgrading to the current JunosE Software release from a lower-numbered release, configurations that use CAC and bulk configuration on the same ATM interface continue to work. However, we recommend that you disable CAC on these ATM interfaces to ensure continued compatibility with future JunosE releases.

For information about how to use the **atm cac** command to configure CAC, see "Setting Optional Parameters" on page 23 in "Configuring ATM" on page 3.

Changing VC Subranges Overview

You can add, remove, modify, merge, disable, and enable VC subranges within an existing bulk-configured VC range. Previously, changes to VC subranges were possible only if you removed the VC range and then configured it again with different subrange values. The ability to make changes to VC subranges without first having to remove the entire VC range avoids potentially disrupting all subscribers on existing dynamic ATM 1483 subinterfaces associated with the deleted VC range. For configuration instructions and examples, see "Changing VC Subranges" on page 651

Related Documentation

- Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview on page 626
- Creating a Bulk-Configured VC Range on a Static ATM AAL5 Interface on page 644

- *atm bulk-config*
- *interface atm*

Bulk Configuration and VC Classes Overview

You can assign a previously configured VC class to a bulk-configured VC range. A *VC class* is a set of attributes for virtual circuits that can include the service category, encapsulation method, F5 OAM options, and Inverse ARP. Using VC classes to configure VC attributes provides the following benefits:

- VC classes enable you to classify and group VCs based on the OAM and traffic requirements of their associated subscribers.
- When subscriber requirements change, a VC class is easier and less time-consuming to modify than individual PVC attributes.

To assign a VC class to a bulk-configured VC range, you use the **atm class-vc** command from Profile Configuration mode to associate the VC class to a base profile. Issuing this command applies the set of attributes in the specified VC class to all bulk-configured VC ranges that are dynamically created from this base profile.

For details about configuring and using VC classes, including information about how precedence levels affect how the router determines attributes values for dynamically created circuits, see "Configuring ATM VC Classes" on page 52 in "Configuring ATM" on page 3.

NOTE: Using the **atm class-vc** command inside a nested profile that is referenced in a base profile has no effect on the bulk-configured VC ranges associated with the base profile. The router accepts only those VC class assignments that are configured in a base profile and ignores any VC class assignments made in a nested profile.

Related Documentation

- Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview on page 626
- Assigning a Previously Configured VC Class to a Base Profile on page 642
- *atm class-vc*

Dynamic ATM 1483 Subinterface Creation Overview

After you configure the ATM 1483 base profile and create the range of VCs on the ATM AAL5 interface, you associate these two components by assigning the base profile to the VC range with the **profile atm1483 bulk-config-name** command.

As a final step, you must issue the **auto-configure atm1483** command. This command configures the ATM AAL5 interface to support autodetection of the ATM 1483 dynamic encapsulation type.

When the router receives an incoming data packet on a circuit, it dynamically creates the ATM 1483 subinterface, using the attributes specified in the base profile. After examining the contents of the data packet, the router dynamically creates the required interface columns above the ATM 1483 subinterface, using the configuration attributes contained in the nested profiles, if specified, or in the base profile itself.

Related Documentation

- Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview on page 626
- Assigning the Base Profile Configured for a Dynamic ATM 1483 Subinterface to the VC Range Configured on a Static ATM AAL5 Interface on page 649
- Configuring the Static ATM AAL5 Interface to Support Autodetection of an ATM 1483 Dynamic Encapsulation Type on page 646
- *auto-configure atm1483*
- *profile atm1483 bulk-config-name*

Overriding ATM 1483 Base Profile Assignments Overview

You can use the **profile atm1483 bulk-config-name pvc** command to assign an overriding profile to a single ATM PVC that exists within a bulk-configured VC subrange. The VC subrange that encompasses the PVC must have been previously configured with the **atm bulk-config** command for use by a dynamic ATM 1483 subinterface. After you assign the overriding profile, the router uses the information in this profile instead of the information in the previously assigned base profile to create any subsequent ATM 1483 dynamic subinterface columns on the specified PVC.

Overriding the base profile assignment for an ATM PVC with a profile that includes debugging attributes enables you to troubleshoot problems with ATM 1483 dynamic subinterface columns created on the specified PVC. The overriding profile, like the original base profile, can include ATM 1483 attributes, nested profile assignments, and individual characteristics for dynamic upper-interface encapsulation types.

NOTE: See *JunosE Release Notes*, *Appendix A*, *System Maximums* for information about the maximum number of overriding profile assignments currently supported per router.

Related Documentation

- ATM Base Profile Overview on page 627
- Assigning an Overriding Profile to a Single ATM PVC that Exists Within a Bulk-Configured VC Subrange on page 650
- Creating a Bulk-Configured VC Range on a Static ATM AAL5 Interface on page 644
- Configuring Overriding Profile Assignments on page 646
- *atm bulk-config*
- *profile atm1483 bulk-config-name pvc*

Static ATM Interfaces Within VC Subranges Overview

You can configure a static ATM interface with an ATM PVC whose VPI and VCI addresses fall within an existing bulk-configured VC subrange. Conversely, you can also create a bulk-configured VC subrange that includes the VPI and VCI addresses belonging to an existing ATM PVC on a static ATM interface. Previously, configurations that caused VPI/VCI address conflicts between a static ATM interface and a bulk-configured VC subrange were prohibited on the router.

In certain ATM network configurations, you might need to transparently forward traffic from selected circuits with unrelated addresses to another location in the network. The ability to create a static ATM interface on a circuit within a bulk-configured VPI/VCI address range is particularly useful when you use ATM layer 2 services over MPLS with Martini encapsulation to forward the traffic from the selected circuits. You must create the interface stack for ATM layer 2 statically and define the configuration parameters individually on a per-interface basis.

The following rules apply when you configure either a static ATM interface within an existing bulk-configured VC subrange, or a subrange that includes an existing static ATM interface:

- All of the following ATM configurations are supported on the static ATM interface: ATM layer 2 services over MPLS including local cross-connects, point-to-point connections, and nonbroadcast multiaccess (NBMA) connections.
- Static ATM interfaces and circuits defined within a bulk-configured VC subrange are stored in NVS and preserved after a reboot.
- The base profile associated with the VC subrange does not apply to any statically defined ATM interfaces that fall within the subrange.
- If a VC subrange includes a statically defined ATM interface, overriding profile assignments configured for the same VPI/VCI address as a statically defined ATM interface become inactive until the static ATM 1483 subinterface is removed. The

overriding profile becomes active again when you remove the static ATM 1483 subinterface. You can display the current operational status (active or inactive) of overriding profile assignments by using "Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface" on page 693.

- Operations that add, remove, modify, merge, disable, or enable VC subranges within a bulk-configured VC range do not affect any static ATM interfaces defined within the VC subrange.
- You cannot create a static ATM circuit if the VPI/VCI address conflicts with an existing ATM 1483 dynamic subinterface column. Such a configuration would disrupt subscribers already connected to the router via the dynamic subinterface.
- You cannot create a static ATM interface with a VPI/VCI address that falls within a range of circuits reserved for use by the MPLS downstream-on-demand label distribution method.
- You cannot configure CAC on a static ATM interface within an existing bulk-configured VC subrange. Conversely, you cannot create a bulk-configured VC subrange that includes a static ATM interface on which CAC is configured. (For information about how to use the **atm cac** command to configure CAC, see "Setting Optional Parameters" on page 23 in "Configuring ATM" on page 3.)

Related Documentation

- Bulk Configuration of VC Ranges Overview on page 629
- Changing VC Subranges on page 651
- Configuring Static ATM Interfaces Within VC Subranges on page 655

Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations Overview

In configurations of dynamic IP over dynamic PPP over a dynamic (bulk-configured) ATM 1483 subinterface, the router sends an LCP terminate request packet to a PPPoA CPE device in response to receipt of an IPv4-over-PPP data packet or an IPv6-over-PPP data packet when the dynamic ATM 1483 subinterface transitions to a dormant state due to an ungraceful subscriber logout. This action terminates stale PPPoA subscribers and causes the CPE to restart LCP negotiations. This behavior is always in effect on the router and does not require CLI or SNMP configuration.

The implementation of this feature for dynamic ATM 1483 subinterfaces is almost identical to the implementation for static ATM 1483 subinterfaces, with the following difference:

- For *static* ATM 1483 subinterfaces, the restart of LCP negotiations by the CPE causes the router to re-create the dynamic PPP and IP upper-layer interfaces above the static ATM 1483 subinterface.
- For *dynamic* ATM 1483 subinterfaces, the receipt of a PPP data packet from the CPE causes the router to re-create only the dynamic ATM 1483 subinterface to send the LCP terminate request packet, but not the dynamic PPP and IP upper-layer interfaces above the dynamic ATM 1483 subinterface. The router re-creates the dynamic PPP and IP upper-layer interfaces when the CPE restarts LCP negotiations.

For details about the operation and benefits of this feature, see “Overview of Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations” on page 540, which describes the router behavior for static ATM 1483 subinterfaces.

Related Documentation

• Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview on page 626

Subscriber Authentication on Dynamic Bridged Ethernet over Dynamic ATM Interfaces

You can use either of the following methods to configure and manage RADIUS authentication for IP subscribers on dynamic bridged Ethernet over dynamic ATM 1483 subinterfaces:

- The **atm atm1483 subscriber** command
- The subscriber management application

The **atm atm1483 subscriber** command *does not support* running stateful SRP switchover (high availability) on the router. Therefore, the configuration method you choose depends on whether stateful SRP switchover is or is not running on your router.

This section describes the following:

- Configuration Method Using atm atm1483 subscriber Command on page 635
- Configuration Method Using Subscriber Management Application on page 635

Configuration Method Using atm atm1483 subscriber Command

When you use the **atm atm1483 subscriber** command, as described in “Configuring a Local Subscriber for a Dynamic Upper-Interface Encapsulation Type Configured over a Dynamic ATM 1483 Subinterface” on page 641, to configure IP subscribers on dynamic bridged Ethernet over dynamic ATM 1483 subinterface columns to support RADIUS authentication, the **atm atm1483 subscriber** command provides the subscriber’s authentication parameters. The dynamic ATM 1483 subinterface acts as the authenticating layer that establishes a session with RADIUS and passes the subscriber’s locally configured username and password information to the RADIUS server.

However, if your router is running stateful SRP switchover (high availability), the use of the **atm atm1483 subscriber** command in this configuration might suspend stateful SRP switchover on the router or prevent stateful SRP switchover from becoming active. To bypass this limitation, you can use the subscriber management application to configure IP subscribers on dynamic bridged Ethernet interfaces.

Configuration Method Using Subscriber Management Application

You can use the JunosE subscriber management application to configure and manage IP subscribers associated with a dynamic bridged Ethernet interface column. The subscriber management application uses an IP service profile to manage and authenticate IP subscribers with RADIUS. An IP service profile contains user and password information, and is used in a route map for subscriber management and to authenticate subscribers with RADIUS.

In this configuration, the IP service profile provides the subscriber's authentication parameters, and the subscriber management application acts as the authenticating layer to obtain information from RADIUS for configuration of dynamic IP subscribers. To assign the IP service profile to the interface profile from which the dynamic bridged Ethernet interface is created, you use the **bridge1483 service-profile** command in Profile Configuration mode.

If stateful SRP switchover is disabled or not running on your router, you can continue to use the **atm atm1483 subscriber** command to configure IP subscribers on dynamic bridged Ethernet interfaces to support RADIUS authentication.

Alternatively, you can use the subscriber management application to create and configure dynamic IP interfaces regardless of whether stateful SRP switchover is running on the router. In addition, using subscriber management enables you to take advantage of several useful features such as the IP inactivity timer.

In the event that an interface profile for a dynamic bridged Ethernet interface includes the **atm atm1483 subscriber** command to configure a local subscriber as well as the **bridge1483 service-profile** command to reference an IP service profile, the values specified with the **atm atm1483 subscriber** command take precedence. The router ignores the values in the IP service profile in this case.

For details about using the subscriber management application to configure RADIUS authentication for IP subscribers on dynamic bridged Ethernet interfaces, see "Subscriber Authentication on Dynamic Bridged Ethernet over Static ATM Interfaces" on page 528 and "Configuring Subscriber Management for IP Subscribers on Dynamic Bridged Ethernet Interfaces" on page 561. The information in these sections, which explains how to use subscriber management to achieve the same functionality as the **subscriber** command without adversely affecting stateful SRP switchover, applies equally to the **atm atm1483 subscriber** command.

For more information about using the subscriber management application, see *JunosE Broadband Access Configuration Guide*.

Related Documentation

- Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview on page 626
- *atm atm1483 subscriber*
- *bridge1483 service-profile*

Configuring a Dynamic ATM 1483 Subinterface

When you use dynamic ATM 1483 subinterfaces over static ATM AAL5 interfaces, you use a process called bulk configuration to configure a range of ATM PVCs that support dynamic interfaces. On receipt of an incoming packet on the virtual circuit, the router dynamically creates the ATM 1483 subinterface. As part of the configuration process, you create an ATM 1483 base profile, which can optionally include nested profile assignments, to define the attributes required to configure the dynamic ATM 1483 subinterface and the dynamic upper-layer encapsulation types built over it.

To configure a dynamic ATM 1483 subinterface:

1. (Optional) Configure profiles containing characteristics for the dynamic upper-interface encapsulation types to be created over the dynamic ATM 1483 subinterface.

   These profiles are referenced in the base profile for the dynamic ATM subinterface as nested profile assignments. For detailed instructions on creating profiles, see "Dynamic Interface Configuration Using a Profile" on page 565.

2. Create the base profile for the dynamic ATM 1483 subinterface by assigning the profile a name. For more information, see "Creating a Base Profile to Configure Attributes for a Dynamic ATM 1483 Subinterface" on page 639.

   host1(config)#**profile atm1483BaseProfile**

   This command accesses Profile Configuration mode, which enables you to configure attributes in the base profile.

3. Define attributes for the ATM 1483 subinterface in the base profile. For more information, see "Configuring a Base Profile for Dynamic ATM 1483 Subinterface" on page 640.

   a. Apply traffic-shaping parameters to the VC range on the ATM AAL5 interface.

      host1(config-profile)#**atm pvc aal5autoconfig cbr 10000**

   b. Configure the ATM 1483 subinterface for autodetection of the PPP upper-interface encapsulation type.

      host1(config-profile)#**atm atm1483 auto-configure ppp**

   c. Configure the ATM 1483 subinterface for autodetection of the IP upper-interface encapsulation type using a nondefault lockout time range of 3600–7200 seconds (1–2 hours).

      host1(config-profile)#**atm atm1483 auto-configure ip lockout-time 3600 7200**

   d. Configure a subscriber for the IP upper-interface encapsulation type.

      host1(config-profile)#**atm atm1483 subscriber ip user-prefix joesmith domain myisp password-prefix abc123**

   e. Configure a description for ATM 1483 subinterfaces that are created with this base profile.

      host1(config-profile)#**atm atm1483 description VC_atm1**

   f. Set an advisory speed for ATM subinterfaces that are created with this base profile.

      host1(config-profile)#**atm atm1483 advisory-rx-speed 2000**

   g. Assign a VC class to the bulk-configured VC ranges created on the dynamic ATM 1483 subinterfaces associated with this base profile. You must issue the **exit** command from Profile Configuration mode for the VC class association to take effect.

      host1(config-profile)#**atm class-vc premium-subscriber-class**

4. (Optional) In the base profile, create nested profile assignments for the upper-interface encapsulation types, and include additional profile characteristics for other encapsulation types as needed. For more information, see "Adding a Nested Profile Assignment to a Base Profile for a Dynamic ATM 1483 Subinterface" on page 641.

For example, the following commands configure nested profile assignments for the PPP and IP upper-interface encapsulation types, and define additional attributes for the PPPoE upper-interface encapsulation type.

```
host1(config-profile)#atm atm1483 profile ppp myPppProfile
host1(config-profile)#atm atm1483 profile ip myIpProfile
host1(config-profile)#pppoe duplicate-protection
host1(config-profile)#pppoe sessions 3000
```

5. Exit Profile Configuration mode.

```
host1(config-profile)#exit
```

6. Configure the ATM and ATM AAL5 interface. For more information, see “Creating a Basic Configuration” on page 21 in “Configuring ATM” on page 3.

```
host1(config)#interface atm 5/0
```

7. Configure a range of VCs on the static ATM AAL5 interface, and assign a name to this range. This operation can take several minutes to complete, depending on the number of VCs being configured. For more information, see “Creating a Bulk-Configured VC Range on a Static ATM AAL5 Interface” on page 644.

NOTE: For information about the maximum number of ATM 1483 bulk configurations supported per chassis, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

For example, the following command creates a VC range named myBulkConfig made up of two VC subranges that configure a total of 5,000 virtual circuits.

```
host1(config-if)#atm bulk-config myBulkConfig vc-range 0 2 101 1100
  vc-range 3 6 201 700
```

NOTE: For information about the maximum number of VCs (sum of the VPI/VCI addresses within all VC subranges) that you can configure with the **atm bulk-config** command per line module and per chassis, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

Do not use any reserved VCI values when configuring VCs with the **atm bulk-config** command. For information about reserved VCIs, see “Configuring F4 OAM” on page 32 in “Configuring ATM” on page 3.

8. Assign the base profile configured for the ATM 1483 subinterface to the VC range configured on the ATM AAL5 interface. For more information, see “Assigning the Base Profile Configured for a Dynamic ATM 1483 Subinterface to the VC Range Configured on a Static ATM AAL5 Interface” on page 649.

```
host1(config-if)#profile atm1483 bulk-config-name myBulkConfig atm1483BaseProfile
```

9. Configure the ATM AAL5 interface to support autodetection of the ATM 1483 dynamic encapsulation type. For more information, see “Configuring the Static ATM AAL5 Interface to Support Autodetection of an ATM 1483 Dynamic Encapsulation Type” on page 646.

host1(config-if)#**auto-configure atm1483**

Related Documentation

- Dynamic ATM 1483 Subinterfaces over Static ATM AAL5 Interfaces Overview on page 626
- Configuring a Base Profile to Apply Encapsulation, Traffic-Shaping and OAM Parameters to a Bulk Range of PVCs Configured on an ATM AAL5 Interface on page 643
- Configuring a Local Subscriber for a Dynamic Upper-Interface Encapsulation Type Configured over a Dynamic ATM 1483 Subinterface on page 641
- Specifying the Types of Dynamic Upper-Interface Encapsulations Accepted or Detected by a Dynamic ATM 1483 Subinterface on page 640
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606
- Monitoring Configuration Information of an ATM AAL5 Interface on page 593
- Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface on page 693
- *atm atm1483 advisory-rx-speed*
- *atm atm1483 auto-configure*
- *atm atm1483 description*
- *atm atm1483 profile*
- *atm atm1483 subscriber*
- *atm bulk-config*
- *atm pvc*
- *interface atm*
- *profile*

Creating a Base Profile to Configure Attributes for a Dynamic ATM 1483 Subinterface

You can use the **profile** command to create a base profile to configure attributes for a dynamic ATM 1483 subinterface. You can specify a profile name of up to 80 alphanumeric characters.

To create a base profile to configure attributes for a dynamic ATM 1483 subinterface:

- Issue the **profile** command in Global Configuration mode.

  host1(config)#**profile atm1483BaseProfile**

Use the **no** version to delete the specified profile if it is not being used by any existing VC subranges.

NOTE: If VC ranges are configured for the dynamic ATM 1483 subinterface associated with the base profile you want to delete, you must use the **no atm bulk-config** command to remove the VC ranges before you can use the **no profile** command to remove the associated base profile.

Related Documentation

- Configuring a Base Profile for Dynamic ATM 1483 Subinterface on page 640
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606
- *profile*

Configuring a Base Profile for Dynamic ATM 1483 Subinterface

This topic includes the following tasks:

- Configuring a Base Profile to Set Up an Advisory Receive Speed for a Dynamic ATM 1483 Subinterface on page 640
- Specifying the Types of Dynamic Upper-Interface Encapsulations Accepted or Detected by a Dynamic ATM 1483 Subinterface on page 640
- Configuring a Base Profile to Assign a Text Description for a Dynamic ATM 1483 Subinterface on page 641
- Adding a Nested Profile Assignment to a Base Profile for a Dynamic ATM 1483 Subinterface on page 641
- Configuring a Local Subscriber for a Dynamic Upper-Interface Encapsulation Type Configured over a Dynamic ATM 1483 Subinterface on page 641
- Assigning a Previously Configured VC Class to a Base Profile on page 642
- Configuring a Base Profile to Apply Encapsulation, Traffic-Shaping and OAM Parameters to a Bulk Range of PVCs Configured on an ATM AAL5 Interface on page 643

Configuring a Base Profile to Set Up an Advisory Receive Speed for a Dynamic ATM 1483 Subinterface

You can use the **atm atm1483 advisory-rx-speed** command to set an advisory receive speed for a dynamic ATM 1483 subinterface created with a base profile. This setting has no effect on data forwarding. It indicates the speed of the client interface. When traffic is tunneled with L2TP, the advisory receive speed is sent from the LAC to the LNS. See *LAC Configuration Prerequisites* for additional information about the advisory receive speed. The range is 0–2147483647 kbps.

To set an advisory receive speed for a dynamic ATM 1483 subinterface created with a base profile:

- Issue the **atm atm1483 advisory-rx-speed** command in Profile Configuration mode.

```
host1(config-profile)#atm atm1483 advisory-rx-speed 2000
```

Use the **no** version to restore the default behavior—the RX speed is not sent to the LNS.

Specifying the Types of Dynamic Upper-Interface Encapsulations Accepted or Detected by a Dynamic ATM 1483 Subinterface

You can use the **atm atm1483 auto-configure** command to specify the types of dynamic upper-interface encapsulations that are accepted or detected by a dynamic ATM 1483 subinterface created with a base profile. For the bridged Ethernet, IP, PPP, and PPPoE encapsulation types, you can optionally specify the lockout time range for the encapsulation type.

To specify the types of dynamic upper-interface encapsulations for a dynamic ATM 1483 subinterface created with a base profile:

- Issue the **atm atm1483 auto-configure** command in Profile Configuration mode.

```
host1(config-profile)#atm atm1483 auto-configure ip lockout-time 3600 7200
host1(config-profile)#atm atm1483 auto-configure pppoe
```

Use the **no** version to terminate detection of the specified encapsulation type.

Configuring a Base Profile to Assign a Text Description for a Dynamic ATM 1483 Subinterface

You can use the **atm atm1483 description** command to assign a text description for a dynamic ATM 1483 subinterface created with a base profile. The description can be up to 255 characters.

To assign a text description for a dynamic ATM 1483 subinterface created with a base profile:

- Issue the **atm atm1483 description** command in Profile Configuration mode.

```
host1(config-profile)#atm atm1483 description VC_atm1
```

Use the **no** version to remove the text description.

Adding a Nested Profile Assignment to a Base Profile for a Dynamic ATM 1483 Subinterface

You can use the **atm atm1483 profile** command to add a nested profile assignment to a base profile for a dynamic ATM 1483 subinterface. A nested profile assignment references another profile that configures attributes for a dynamic upper-interface type over the ATM 1483 subinterface.

To add a nested profile assignment to a base profile for a dynamic ATM 1483 subinterface:

- Issue the **atm atm1483 profile** command in Profile Configuration mode.

```
host1(config-profile)#atm atm1483 profile pppoe atm1483ProfilePppoe
```

Use the **no** version to remove the profile assignment for the upper-interface encapsulation type.

Configuring a Local Subscriber for a Dynamic Upper-Interface Encapsulation Type Configured over a Dynamic ATM 1483 Subinterface

You can use the **atm atm1483 subscriber** command to configure a local subscriber for a dynamic upper-interface encapsulation type configured over a dynamic ATM 1483 subinterface created with a base profile. A subscriber supports authentication and configuration from the RADIUS server.

When you configure a subscriber, you must specify the following:

- *upperInterfaceType*—Type of dynamic interface, **bridgedEthernet** or **ip**
- *userNameUsage*—How the dynamic interface uses the username for authentication purposes
  - **user**—Use the name as specified.
  - **user-prefix**—Use the name as a prefix to the interface physical location. The router automatically postpends the physical location of the user to the username string. The username format is userName.slot.port.vpi.vci. The resulting username string is then used to authenticate the subscriber with the RADIUS server.
- *userName*—RADIUS username
- *domainName*—Domain name

You can optionally supply password information:

- *passwordUsage*—How the dynamic interface uses the password for authentication purposes
  - **password**—Use the password as specified.
  - **password-prefix**—Use the password as a prefix to the interface physical location. The router automatically postpends the physical location of the user to the password string. The password format is password.slot.port.vpi.vci. The resulting password string is then used to authenticate the subscriber with the RADIUS server.
- *password*—RADIUS password

If your router is running stateful SRP switchover (high availability), the use of the **atm atm1483 subscriber** command to configure RADIUS authentication for subscribers on dynamic bridged Ethernet interfaces might suspend stateful SRP switchover on the router or prevent stateful SRP switchover from becoming active. For more information about using the subscriber management application to bypass this limitation, see "Subscriber Authentication on Dynamic Bridged Ethernet over Dynamic ATM Interfaces" on page 635.

To configure a local subscriber for a dynamic upper-interface encapsulation type configured over a dynamic ATM 1483 subinterface with a base profile::

- Issue the **atm atm1483 subscriber** command in Profile Configuration mode.

```
host1(config-profile)#atm atm1483 subscriber ip user-prefix boston01
  domain myisp password-prefix abc123
```

Use the **no** version to remove the subscriber.

Assigning a Previously Configured VC Class to a Base Profile

You can use the **atm class-vc** command to assign a previously configured VC class to a base profile for a dynamic ATM 1483 subinterface. This command applies the set of attributes in the specified VC class to all bulk-configured VC ranges that are dynamically created from this base profile.

You must issue the **exit** command from Profile Configuration mode for the VC class association to take effect.

NOTE: **Changes to a VC class specified in a base profile apply only to those PVCs that are dynamically created after the change is made. These changes do not apply to dynamic PVCs that were created prior to the VC class modification.**

To assign a previously configured VC class to a base profile for a dynamic ATM 1483 subinterface:

- Issue the **atm class-vc** command in Profile Configuration mode.

```
host1(config-profile)#atm class-vc gold-subscriber-class
host1(config-profile)#exit
```

Use the **no** version to remove the VC class association with the base profile.

Configuring a Base Profile to Apply Encapsulation, Traffic-Shaping and OAM Parameters to a Bulk Range of PVCs Configured on an ATM AAL5 Interface

You can use the **atm pvc** command to apply encapsulation, traffic-shaping, and OAM parameters to the range of ATM PVCs configured on an ATM AAL5 interface for use by a dynamic ATM 1483 subinterface created with a base profile.

You must specify one of the following encapsulation types:

- **aal5autoconfig**—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed)
- **aal5snap**—Specifies a logical link control (LLC) encapsulated circuit; the LLC/Subnetwork Access Protocol (LLC/SNAP) header precedes the protocol datagram
- **aal5mux ip**—Specifies a VC-based multiplexed circuit used for IP only

You can optionally set the *peak*, *average*, and *burst* sizes. To use VBR-RT or VBR-NRT as the service type, you must specify each of these options.

The default service type is UBR. To set a different service type, specify one of the following keywords:

- **rt**—Selects VBR-RT as the service type; you can select **rt** only if you set the *peak*, *average*, and *burst* parameters
- **cbr**—Selects CBR as the service type; you must set the CBR rate in Kbps

You can optionally include the **oam** keyword and a number of seconds in the range 1–600 to enable generation of OAM F5 loopback cells on this circuit. This option enables VC integrity features that affect the operational state of the ATM PVC.

To apply encapsulation, traffic-shaping, and OAM parameters to the range of ATM PVCs configured on an ATM AAL5 interface for use by a dynamic ATM 1483 subinterface created with a base profile:

- Issue the **atm pvc** command in Profile Configuration mode.

```
host1(config-profile)#atm pvc aal5autoconfig cbr 10000 oam 120
```

Use the **no** version to restore the default service type, UBR, on the VC range.

Related Documentation

- ATM Base Profile Overview on page 627
- Assigning a Previously Configured VC Class to a Base Profile on page 642
- Bulk Configuration and VC Classes Overview on page 631
- *Configuring the RX Speed on the LAC*
- Creating a PVC on an ATM 1483 Subinterface on page 537
- Dynamic Encapsulation Type Lockout on page 532
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606
- *atm atm1483 advisory-rx-speed*
- *atm atm1483 auto-configure*
- *atm atm1483 description*
- *atm atm1483 profile*
- *atm atm1483 subscriber*
- *atm class-vc*
- *atm pvc*

Creating a Bulk-Configured VC Range on a Static ATM AAL5 Interface

You can use the **atm bulk-config** command to create a bulk-configured VC range on a static ATM AAL5 interface for use by a dynamic ATM 1483 subinterface.

NOTE: **For information about the maximum number of ATM 1483 bulk configurations supported per chassis, see** ***JunosE Release Notes*****,** ***Appendix A*****,** ***System Maximums*****.**

Each VC range consists of one or more nonoverlapping VC subranges. A VC subrange is a group of VCs that resides within the VPI and VCI ranges you specify. You can configure multiple VC ranges on an ATM AAL5 interface.

NOTE: For information about the maximum number of VCs (sum of the VPI/VCI addresses within all VC subranges) that you can configure with the **atm bulk-config** command per line module and per chassis, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

Do not use any reserved VCI values when configuring VCs with the **atm bulk-config** command. For information about reserved VCIs, see "Configuring F4 OAM" on page 32 in "Configuring ATM" on page 3.

When you create a bulk-configured VC range, you must specify the following:

- A name of up to 80 alphanumeric characters; this is also referred to as the bulk configuration name
- The starting and ending VPI values (inclusive) for each VC subrange
- The starting and ending VCI values (inclusive) for each VC subrange

You can create a placeholder VC range by issuing the **atm bulk-config** command without specifying any subranges. You can assign a profile to this placeholder and add subranges to it later.

You can add and remove individual VC subranges. You cannot remove a VC subrange if any dynamic ATM 1483 subinterfaces currently exist for any circuit within the subrange. You can use the **atm bulk-config shutdown** command to remove dynamic ATM 1483 interfaces created within a subrange. Removal of a subrange automatically results in the removal of all overriding profile assignments on that subrange. You can create a bulk-configured VC subrange that includes the VPI and VCI addresses belonging to an existing ATM PVC on a static ATM interface.

You cannot create a bulk-configured VC range on an ATM interface on which you have configured CAC. Conversely, you cannot configure CAC on an ATM interface on which you have created a bulk-configured VC range. For information about configuring CAC, see "Setting Optional Parameters" on page 23 in "Configuring ATM" on page 3.

To create a bulk-configured VC range on a static ATM AAL5 interface for use by a dynamic ATM 1483 subinterface:

- Configure a VC range named myBulkConfig with a single VC subrange containing VPIs 0–2 and VCIs 101–1100; this command configures a total of 3000 VCs

  host1(config-if)#**atm bulk-config myBulkConfig vc-range 0 2 101 1100**

- Configure a VC range named myMultiBulkConfig with two VC subranges containing VPIs 0–1 and VCIs 101–600 (first subrange) and VPIs 3–5 and VCIs 201–3200 (second subrange); this command configures a total of 10,000 VCs

  host1(config-if)#**atm bulk-config myMultiBulkConfig vc-range 0 1 101 600 vc-range 3 5 201 3200**

Use the **no** version to remove the specified VC range from the ATM AAL5 interface, to remove the specified subranges from the specified VC range, or to remove all subranges from the specified VC range. The **no** version also removes any overriding profile assignments for ATM PVCs within the deleted VC range or VC subrange.

Related Documentation

- Bulk Configuration of VC Ranges Overview on page 629
- Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface on page 693
- *atm bulk-config*

Configuring the Static ATM AAL5 Interface to Support Autodetection of an ATM 1483 Dynamic Encapsulation Type

You can use the **auto-configure atm1483** command to configure the static ATM AAL5 interface to enable all bulk configurations and support autodetection of an ATM 1483 dynamic encapsulation type. This command enables creation of a dynamic ATM 1483 subinterface.

To configure the static ATM AAL5 interface to enable all bulk configurations and support autodetection of an ATM 1483 dynamic encapsulation type:

- Issue the **auto-configure atm1483** command in Interface Configuration mode.

```
host1(config-if)#auto-configure atm1483
```

Use the **no** version to terminate autodetection of the ATM 1483 encapsulation type.

Related Documentation

- Dynamic ATM 1483 Subinterface Creation Overview on page 632
- Monitoring Configuration Information of an ATM AAL5 Interface on page 593
- *auto-configure atm1483*

Configuring Overriding Profile Assignments

Configuring overriding profile assignments includes the following tasks:

- Assigning an Overriding Profile to an ATM PVC Within a Bulk-Configured VC Subrange on page 646
- Removing an Overriding Profile Assignment from an ATM PVC on page 648
- Removing an Overriding Profile Assignment from a VC Range or VC Subrange on page 649

Assigning an Overriding Profile to an ATM PVC Within a Bulk-Configured VC Subrange

You can assign an overriding profile to a single ATM PVC within a bulk-configured VC subrange. Typically, the overriding profile includes debugging attributes to help you identify and troubleshoot problems with the ATM 1483 dynamic subinterface column created on the specified PVC.

To assign an overriding profile to an ATM PVC within a bulk-configured VC subrange:

1. Configure both of the following:

- Base profile for the bulk-configured VC range on the static ATM AAL5 interface. The VC range consists of one or more VC subranges. For more information, see "Configuring a Base Profile for Dynamic ATM 1483 Subinterface" on page 640
- Overriding profile for an ATM PVC within a bulk-configured VC subrange

For information about configuring profiles, see "Dynamic Interface Configuration Using a Profile" on page 565.

2. Create a bulk-configured range of VCs on a static ATM AAL5 interface. The following commands create a bulk-configured VC range named myBulkConfig that consists of two VC subranges. The first subrange encompasses VPIs 0–4 and VCIs 21–1000. The second subrange encompasses VPIs 5–7 and VCIs 21–2000. For more information, see "Creating a Bulk-Configured VC Range on a Static ATM AAL5 Interface" on page 644.

```
host1(config)#interface atm 4/0
host1(config-if)#atm bulk-config myBulkConfig vc-range 0 4 21 1000
  vc-range 5 7 21 2000
```

3. Assign the previously configured base profile (atm1483BaseProfile) to the bulk-configured VC range. For more information, see "Assigning the Base Profile Configured for a Dynamic ATM 1483 Subinterface to the VC Range Configured on a Static ATM AAL5 Interface" on page 649.

```
host1(config-if)#profile atm1483 bulk-config-name myBulkConfig atm1483BaseProfile
```

4. Assign the previously configured overriding profile to a single ATM PVC within the bulk-configured VC subrange. The following command assigns the overriding profile myDebugProfile to the PVC with VPI 0 and VCI 101. This PVC exists within the first VC subrange (VPIs 0–4 and VCIs 21–1000) configured in Step 2. For more information, see Assigning an Overriding Profile to a Single ATM PVC that Exists Within a Bulk-Configured VC Subrange on page 650

```
host1(config-if)#profile atm1483 bulk-config-name myBulkConfig pvc 0 101
  myDebugProfile
```

The router now uses the information in the overriding profile instead of the information in the base profile to create subsequent ATM 1483 dynamic subinterface columns over this PVC.

5. (Optional) You can assign the same overriding profile to a different ATM PVC within the same VC subrange or within a different VC subrange. For example, the following command assigns the overriding profile myDebugProfile to the PVC with VPI 6 and VCI 901. This PVC exists within the second VC subrange (VPIs 5–7 and VCIs 21–2000) configured in Step 2.

```
host1(config-if)#profile atm1483 bulk-config-name myBulkConfig pvc 6 901
  myDebugProfile
```

NOTE: You can reverse the order of Step 3 and Step 4 with identical results. That is, you can assign the overriding profile to the ATM PVC and then assign the base profile to the entire VC range. In either case, you must first create the bulk-configured VC range with the **atm bulk-config** command.

6. Configure the ATM AAL5 interface to enable all bulk configurations and to support autodetection of the ATM 1483 dynamic encapsulation type. For more information, see “Configuring the Static ATM AAL5 Interface to Support Autodetection of an ATM 1483 Dynamic Encapsulation Type” on page 646.

   host1(config-if)#**auto-configure atm1483**

7. (Optional) Use the **show atm bulk-config** command to verify the overriding profile configuration. For more information, see “Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface” on page 693.

Removing an Overriding Profile Assignment from an ATM PVC

After you troubleshoot the ATM 1483 dynamic subinterface column created on the specified PVC, make sure that you remove the overriding profile assignment to restore the original base profile assignment. This action ensures that subsequent ATM 1483 dynamic subinterface columns are created using the same attributes defined in the base profile.

To remove an overriding profile assignment from an ATM PVC within a bulk-configured VC range:

1. Remove the overriding profile assignment from the specified ATM PVC. For more information, see “Assigning the Base Profile Configured for a Dynamic ATM 1483 Subinterface to the VC Range Configured on a Static ATM AAL5 Interface” on page 649

   host1(config-if)#**no profile atm1483 bulk-config-name myBulkConfig pvc 0 101**

2. Select the dynamic ATM 1483 subinterface on which the ATM 1483 dynamic subinterface column resides. For more information, see “Creating a Basic Configuration” on page 21 in “Configuring ATM” on page 3.

   host1(config)#**interface atm 4/0.101**

3. Use the **shutdown** command to disable the dynamic ATM 1483 subinterface. The **shutdown** command deletes the ATM 1483 dynamic subinterface column and removes the dynamic ATM 1483 subinterface. For more information, see “Disabling a Dynamic ATM 1483 Interface” on page 651.

   host1(config-subif)#**shutdown**

4. Send traffic over the specified PVC (VPI 0 and VCI 101) on the ATM AAL5 interface. This action re-creates the ATM 1483 dynamic subinterface column with the original base profile association.

   The router now uses the information in the base profile instead of the information in the overriding profile to create subsequent ATM 1483 dynamic subinterface columns for the specified PVC.

5. (Optional) Use the **show atm bulk-config** command to verify the removal of the overriding profile assignment. For more information, see “Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface” on page 693.

Removing an Overriding Profile Assignment from a VC Range or VC Subrange

When you issue the **no atm bulk-config** command to remove an entire VC range (and all VC subranges within that VC range), the router also removes any overriding profile assignments configured for PVCs within those VC subranges.

To remove the bulk-configured VC range named myBulkConfig and any overriding profile assignments for PVCs within the VC subranges belonging to myBulkConfig:

- Issue the **no atm bulk-config** command in Interface Configuration mode without VC subranges.

  host1(config-if)#**no atm bulk-config myBulkConfig**

When you issue the **no atm bulk-config** command to remove a particular VC subrange in a bulk-configured VC range, the router also removes any overriding profile assignments for PVCs within that VC subrange. However, overriding profile assignments for PVCs within other VC subranges in the VC range remain intact.

To remove one VC subrange (VPIs 0–4 and VCIs 21–1000) and only those overriding profile assignments associated with this subrange:

- Issue the **no atm bulk-config** command in Interface Configuration mode with VC subrange.

  host1(config-if)#**no atm bulk-config myBulkConfig vc-range 0 4 21 1000**

Related Documentation

- Overriding ATM 1483 Base Profile Assignments Overview on page 632
- Creating a Bulk-Configured VC Range on a Static ATM AAL5 Interface on page 644
- Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface on page 693
- *atm bulk-config*
- *interface atm*
- *profile atm1483 bulk-config-name*
- *show atm bulk-config*
- *shutdown*

Assigning the Base Profile Configured for a Dynamic ATM 1483 Subinterface to the VC Range Configured on a Static ATM AAL5 Interface

You can use the **profile atm1483 bulk-config-name** command to assign the base profile configured for a dynamic ATM 1483 subinterface to the VC range configured on a static ATM AAL5 interface.

When assigning the base profile configured for a dynamic ATM 1483 subinterface to the VC range configured on a static ATM AAL5 interface, you must specify the following:

- Name assigned to the VC range on an ATM AAL5 interface, as specified in the ***atm bulk-config*** command
- Name assigned to the base profile for a dynamic ATM 1483 subinterface

To assign the base profile configured for a dynamic ATM 1483 subinterface to the VC range configured on a static ATM AAL5 interface:

- Issue the **profile atm1483 bulk-config-name** command in Interface Configuration mode.

```
host1(config-if)#profile atm1483 bulk-config-name myBulkConfig  atm1483BaseProfile
```

Use the **no** version to remove the profile assignment.

Related Documentation

- Dynamic ATM 1483 Subinterface Creation Overview on page 632
- Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface on page 693
- *atm bulk-config*
- *profile atm1483 bulk-config-name*

Assigning an Overriding Profile to a Single ATM PVC that Exists Within a Bulk-Configured VC Subrange

You can use the **profile atm1483 bulk-config-name pvc** command to assign an overriding profile to a single ATM PVC that exists within a bulk-configured VC subrange.

An overriding profile typically includes debugging attributes that help you troubleshoot problems with the ATM 1483 dynamic subinterface column created on the specified PVC.

The VPI and VCI values of the PVC you specify must exist between the starting VPI/VCI values and ending VPI/VCI values of a VC subrange previously configured with the **atm bulk-config** command.

To assign or remove an overriding profile to a single ATM PVC that exists within a bulk-configured VC subrange:

a. Assign an overriding profile (test1DebugProfile) to the ATM PVC with VPI 4 and VCI 301 that is within VPIs 3–5 and VCIs 201–3200 subrange:

```
host1(config-if)#profile atm1483 bulk-config-name test1 pvc 4 301  test1DebugProfile
```

b. Remove the overriding profile assignment from the ATM PVC with VPI 4 and VCI 301, and restore the original base profile assignment:

```
host1(config-if)#no profile atm1483 bulk-config-name test1 pvc 4 301
```

Related Documentation

- Overriding ATM 1483 Base Profile Assignments Overview on page 632
- Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface on page 693
- *atm bulk-config*
- *profile atm1483 bulk-config-name pvc*

Disabling a Dynamic ATM 1483 Interface

You can use the **shutdown** command to disable an interface.

When you disable a dynamic ATM 1483 interface, the **shutdown** command deletes the ATM 1483 dynamic subinterface column and removes the dynamic ATM 1483 subinterface.

To disable an interface:

- Issue the **shutdown** command in Subinterface Configuration mode.

```
host1(config-subif)#shutdown
```

Because the **shutdown** command removes the dynamic ATM 1483 subinterface from the router, issuing a subsequent **no** version of this command has no effect; that is, it does not restart the disabled subinterface.

Related Documentation

- Configuring a Dynamic ATM 1483 Subinterface on page 636
- Monitoring Summary Information for ATM VCs and Reserved VC Ranges on page 600

- *shutdown*

Changing VC Subranges

Changing VC subranges within a bulk-configured VC range includes the following tasks:

- Adding New VC Subranges to an Existing VC Range on page 651
- Removing VC Subranges from an Existing VC Range on page 652
- Modifying a VC Subrange by Shortening or Expanding the Subrange Values on page 653
- Merging Multiple VC Subranges Belonging to an Existing VC Range on page 653
- Changing the Administrative State of VC Subranges on page 654

Adding New VC Subranges to an Existing VC Range

You can add a new VC subrange to an existing VC range only when the new subrange does not overlap with any existing subrange. Any overlap causes the addition to fail.

You can add multiple subranges to an existing VC range simultaneously. However, the entire operation fails if even one of the new subranges overlaps with an existing subrange.

The following example specifies the original VC subranges.

```
host1(config-if)#atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250 vc-range
  5 5 501 550 vc-range 3 3 301 350
```

To add subranges to this bulk-configured VC range, you can choose either of the following methods. Each method adds a new subrange (4, 4, 401, 450) to the existing VC range, test.

- Specify one new subrange at a time.

host1(config-if)#**atm bulk-config test vc-range 4 4 401 450**

- Specify the new subrange and all the existing subranges. If you use this method, all the existing subranges and their order must match exactly, or the operation fails.

  host1(config-if)#**atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250 vc-range 5 5 501 550 vc-range 3 3 301 350 vc-range 4 4 401 450**

  The following operation fails because the order of subranges does not match the existing order.

  host1(config-if)#**atm bulk-config test vc-range 2 2 201 250 vc-range 1 1 101 150 vc-range 5 5 501 550 vc-range 3 3 301 350 vc-range 4 4 401 450 vc-range 6 6 601 650**

You can create a placeholder VC range by specifying a VC range name without specifying any subrange parameters. This VC range has no circuit reservation, but you can assign a profile to it, and add subranges later as desired. The following commands illustrate this approach.

host1(config-if)#**atm bulk-config test**
host1(config-if)#**profile atm1483 bulk-config-name test atmProfile**
host1(config-if)#**atm bulk-config test vc-range 4 4 401 450 vc-range 6 6 601 650**

Removing VC Subranges from an Existing VC Range

You can remove VC subranges from an existing VC range if no dynamic ATM 1483 subinterfaces currently exists for any circuit within those subranges. The removal operation fails if any such dynamic ATM 1483 subinterface exists. You must first remove the dynamic ATM 1483 subinterfaces before you can remove the subranges. Removal of a subrange automatically results in the removal of all overriding profile assignments on that subrange.

You can remove only a single specific VC subrange at a time. The following example specifies the original VC subranges.

host1(config-if)#**atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250 vc-range 5 5 501 550 vc-range 3 3 301 350**

To remove one subrange (1, 1, 101, 150) and leave the remaining subranges, and the named VC range, test, intact:

- Issue the **no atm bulk-config** command in Interface Configuration mode.

host1(config-if)#**no atm bulk-config test vc-range 1 1 101 150**

To remove more than one VC subrange, you must issue multiple removal commands, one for each subrange. You cannot remove only part of a subrange. A removal command cannot encompass more than one subrange, even if the subranges are adjacent. However, if you do not specify any subranges, you can remove all subranges in the VC, and the named VC range, at the same time.

- Issue the **no atm bulk-config** command in Interface Configuration mode.

host1(config-if)#**no atm bulk-config test**

Modifying a VC Subrange by Shortening or Expanding the Subrange Values

You can shorten or expand a subrange by modifying the subrange values of a VC range. You can expand a subrange if none of the circuits added overlap with any other subrange. You can shorten a subrange if none of the circuits dropped have existing dynamic ATM 1483 subinterfaces.

You can modify a subrange so that it completely includes at least one other subrange from within the same VC range effectively merges the subranges. Each subrange that is merged with another frees up a subrange for subsequent configuration. The subranges that are merged do not need to be adjacent to each other.

The router retains any overriding profiles assigned to a subrange if the assignment falls within the modified subrange. If the assignment falls outside of the newly modified subrange, the router drops the overriding profile assignment. If two subranges are merged, the router retains overriding profiles that were assigned to the separate subranges and applies the overriding profiles to the newly merged subrange.

You can modify only a single specific subrange at a time. To modify a VC subrange:

- Issue the **atm bulk-config** command in Interface Configuration mode.

```
host1(config-if)#atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250  vc-range
   5 5 501 550 vc-range 3 3 301 350
```

To modify the second subrange from (2, 2, 201, 250) to (2, 3, 210, 230):

- Issue the **atm bulk-config modify** command in Interface Configuration mode.

```
host1(config-if)#atm bulk-config test modify vc-range 2 3 210 230
```

The router retains any overriding profiles assigned to a subrange after you modify the subrange if the override assignment still falls within the modified subrange. If the assignment falls outside of the newly modified subrange, the router drops the overriding profile assignment.

You cannot modify a subrange at the same time you are adding or removing a subrange. If the new modified values for a subrange partially overlap with another subrange, the operation fails and the router displays an error message.

Merging Multiple VC Subranges Belonging to an Existing VC Range

You can merge multiple subranges of any particular VC range to form a single unified subrange, conserving subrange resources. Merging takes place only when you modify a subrange so that it completely includes at least one other subrange of the same VC range. The merged subranges do not need to be adjacent to each other.

If the encompassing subrange has any circuits that are outside the subranges to be merged, those circuits are added. The encompassing subrange must cover a subrange completely to incorporate it in the merged subrange. The merge operation fails if the encompassing subrange completely overlaps some subranges but only partially overlaps with another subrange. The encompassing subrange does not have to encompass all subranges of the VC range.

Each subrange that is merged with another frees up a subrange. E Series routers currently support a maximum of 300 bulk-configured VC ranges per chassis. Therefore, if a VC range consists of 5 subranges, 295 subranges are still available for subsequent configuration. If you merge 2 of those subranges, resulting in a new total of 4 subranges in the VC range, then 296 subranges are available for configuration.

The router retains any overriding profile assignments on the subranges made before the merger, and applies them to the new merged subrange. You can separate merged subranges either by removing the merged subrange and then adding new separate subranges or by modifying the merged subrange to remove some portion of the subrange and then adding a new subrange.

The following example specifies the original VC subranges.

```
host1(config-if)#atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250  vc-range
  5 5 501 550 vc-range 3 3 301 350
```

To merge two subranges, (1, 1, 101, 150) and
(2, 2, 201, 250), and effectively replace them with the new subrange (1, 2, 101, 250):

- Issue the **atm bulk-config modify** command in Interface Configuration mode.

```
host1(config-if)#atm bulk-config test modify vc-range 1 2 101 250
```

To separate the merged subranges, you can modify the unified subrange and add subranges as needed, provided that no dynamic ATM 1483 subinterfaces currently exist for any circuit within those subranges.

If you merge subranges by using SNMP, the new merged subrange takes the lowest instance value of the incorporated subranges. For example, if a VC range has three subranges with instance values of 2, 4, and 5 and the subranges with instance values of 2 and 5 are merged, the new merged subrange has an instance value of 2.

Changing the Administrative State of VC Subranges

VC subranges have an administrative state that enables you to remove dynamic ATM 1483 subinterfaces on various subranges that belong to a single VC range. This functionality is important because subrange removal requires that no dynamic ATM 1483 subinterfaces exist for any circuit on that subrange. The removal operation fails if any such interfaces exist.

By default, the administrative state of a VC subrange is up. When you change the administrative state to down by using the **atm bulk-config shutdown** command, the router deletes all dynamic ATM 1483 subinterfaces on the affected subranges. You can use the **show atm subinterface** command or the **show atm vc** command to monitor the progress of the removal of all dynamic ATM 1483 subinterfaces for the specified subrange.

No additional dynamic ATM 1483 subinterfaces can be created for the subrange until you restore the administrative state to up by using the **no atm bulk-config shutdown** command.

The following example specifies the original VC subranges.

host1(config-if)#**atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250 vc-range 5 5 501 550 vc-range 3 3 301 350**

You cannot specify a partial subrange; the specified subrange must exactly match a subrange that has already been configured. The router removes all dynamic interface columns built on any of the circuits in this subrange. No additional dynamic ATM 1483 subinterfaces can be created until you change the administrative state to up.

To change the administrative state of VC subranges:

a. Change the administrative state of the second subrange (2, 2, 201, 250) to down:

   host1(config-if)#**atm bulk-config test shutdown vc-range 2 2 201 250**

b. Change the administrative state of the same VC subrange to up:

   host1(config-if)#**no atm bulk-config test shutdown vc-range 2 2 201 250**

c. Shut down all four subranges belonging to the named VC range, test, regardless of their current state:

   host1(config-if)#**atm bulk-config test shutdown**

The time required for the router to complete an administrative state change depends on the number of VC subranges configured.

Related Documentation

- Bulk Configuration of VC Ranges Overview on page 629
- *Monitoring ATM Subinterfaces for Layer 2 Services over MPLS*
- *Monitoring ATM VCs and VPI/VCI Ranges Used for MPLS*
- Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface on page 693
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606
- *atm bulk-config*
- *atm bulk-config modify*
- *profile atm1483 bulk-config-name*

Configuring Static ATM Interfaces Within VC Subranges

The following topics describe how to configure static ATM interfaces within VC subranges:

- Creating a Static ATM Interface Within an Existing Bulk-Configured VC Subrange on page 655
- Creating a Bulk-Configured VC Subrange that Includes an Existing Static ATM Interface on page 656

Creating a Static ATM Interface Within an Existing Bulk-Configured VC Subrange

You can configure a static ATM interface with an ATM PVC whose VPI and VCI addresses fall within an existing bulk-configured VC subrange.

To create a static ATM interface within a VC subrange:

1. Create a bulk-configured VC range that includes one or more VC subranges.

   host1(config)#interface atm 0/0
   host1(config-if)#atm bulk-config test vc-range 1 3 32 1031

2. Specify a static ATM 1483 subinterface. For more information, see "Creating a Basic Configuration" on page 21 in "Configuring ATM" on page 3.

   host1(config-if)#interface atm 0/0.2100

3. Configure an ATM PVC with VPI and VCI values that fall within the bulk-configured VC subrange. In this example, the VPI value (2) is within the VPI range 1–3, and the VCI value (100) is within the VCI range 32–1031. For more information, see "Creating a Basic Configuration" on page 21 in "Configuring ATM" on page 3.

   host1(config-subif)#atm pvc 2100 2 100 aal0

4. Configure the static ATM interface. For example, the **mpls-relay** command creates a ATM layer 2 services over MPLS tunnel on the circuit. For detailed information about using the **mpls-relay** command, see *Configuring Layer 2 Services over MPLS* in *JunosE BGP and MPLS Configuration Guide*.

   host1(config-subif)#mpls-relay 192.168.0.1 2100

Creating a Bulk-Configured VC Subrange that Includes an Existing Static ATM Interface

You can configure a bulk-configured VC subrange that includes the VPI and VCI addresses belonging to an existing ATM PVC on a static ATM interface.

To create a VC subrange that includes a static ATM interface:

1. Specify a static ATM 1483 subinterface. For more information, see "Creating a Basic Configuration" on page 21 in "Configuring ATM" on page 3.

   host1(config-if)#interface atm 3/1.201

2. Configure an ATM PVC on the static ATM 1483 subinterface. In this example, the VPI value is 1 and the VCI value is 101. For more information, see "Creating a Basic Configuration" on page 21 in "Configuring ATM" on page 3.

   host1(config-subif)#atm pvc 201 1 101 aal0

3. Configure the static ATM interface. For example, the **mpls-relay** command creates an ATM layer 2 services over MPLS tunnel on the circuit. For detailed information about using the **mpls-relay** command, see *Configuring Layer 2 Services over MPLS* in *JunosE BGP and MPLS Configuration Guide*.

   host1(config-subif)#mpls-relay 5.1.1.1 201

4. Create a bulk-configured VC range that includes the VPI and VCI values of the previously configured ATM PVC. In this example, the VPI range (0–2) includes VPI 1, and the VCI range (100–250) includes VCI 101.

   host1(config)#interface atm 3/1
   host1(config-if)#atm bulk-config test2 vc-range 0 2 100 250

Related Documentation

- Static ATM Interfaces Within VC Subranges Overview on page 633
- Creating a Bulk-Configured VC Range on a Static ATM AAL5 Interface on page 644
- Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface on page 693
- Monitoring Status or Summary Information for ATM Subinterfaces on page 595

- *atm bulk-config*
- *atm pvc*
- *interface atm*
- *mpls-relay*

Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview

E Series routers support configuration of dynamic VLAN subinterfaces over static VLAN major interfaces over Ethernet.

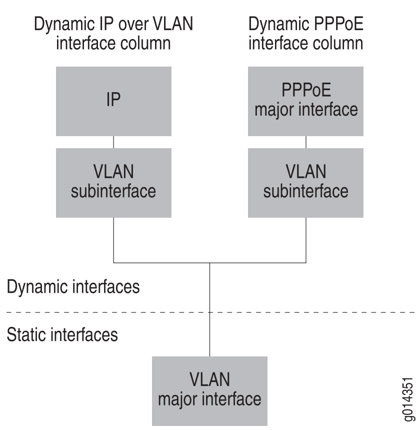
When you configure the dynamic VLAN subinterface, you can enable autodetection and dynamic creation of the following upper-layer encapsulation types:

- IP
- PPPoE

NOTE: **Unlike ATM, which supports dynamic upper interfaces over static ATM 1483 subinterfaces, you must configure a dynamic VLAN subinterface to enable autodetection and dynamic creation of IP and PPPoE interfaces.**

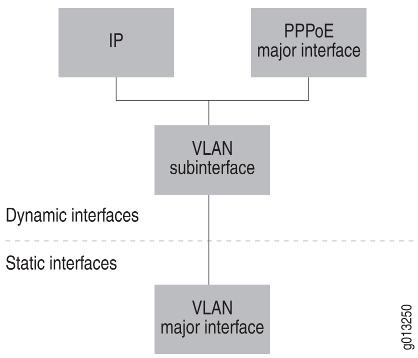
Figure 61 on page 657 shows the dynamic upper-interface columns supported by dynamic VLAN subinterfaces, and indicates which layers in the columns are static and dynamic.

Figure 61: Dynamic Interface Columns over Dynamic VLAN Subinterfaces



Unlike ATM 1483, you can configure both IP and PPPoE over a single dynamic VLAN subinterface (Figure 62 on page 658).

Figure 62: Dynamic IP and PPPoE over Single Dynamic VLAN Subinterface



When you configure dynamic VLAN subinterfaces over static VLAN major interfaces, you must configure the VLAN major interface, including the attributes of the VLAN major interface. VLAN major interface attributes include profile assignments and autoconfiguration settings.

As part of the configuration process, you create a VLAN base profile, which can optionally include nested profile assignments, to define the attributes required to configure the dynamic VLAN subinterface and the dynamic upper-layer encapsulation types built over it.

When the router receives a packet, it examines the packet for a VLAN ID or double-tagged S-VLAN ID. You can also configure the router to further examine the packet for agent-circuit-identifier information. Based on these values and the configuration data received from a profile, the router creates all dynamic layers above the VLAN layer, starting with the lowest dynamic layer. For example, in the case of a dynamic PPPoE interface, the router creates the interfaces in the following order:

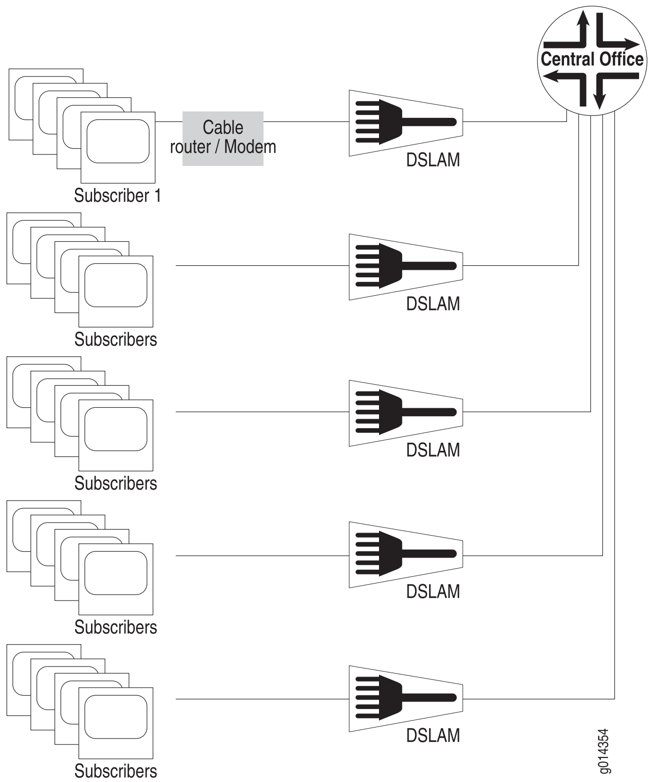
- Dynamic VLAN subinterface
- PPPoE interface
- PPP interface
- IP interface

If any layer of the dynamic portion of the interface column fails to be created, then the interface creation fails and the connection is denied. All dynamic layers above the VLAN subinterface are destroyed, starting with the highest dynamic layer. VLAN subinterfaces are persistent; after they are created, they cannot be destroyed, unless the operational state changes to down.

Dynamic VLAN subinterfaces function identically to static VLAN subinterfaces, except for the manner in which they are created and configured. However, dynamic VLANs provide you with the flexibility of having the dynamic interface column created automatically only when the subscriber logs in.

Figure 63 on page 659 displays the relationship between the central office, digital subscriber line access multiplexers (DSLAMs), and subscribers. The subscribers are connected to the DSLAMS through Gigabit Ethernet interfaces.

Figure 63: Dynamic VLAN Subinterfaces for Subscribers



For example, if an S-VLAN is assigned at the DSLAM, and each DSLAM subscriber at the DSLAM is assigned a unique VLAN ID, the JunosE Software dynamically constructs a VLAN-based interface column using that S-VLAN/VLAN ID pair when the subscriber logs in.

For more information about the attributes of VLAN and S-VLAN subinterfaces, see "VLAN Overview" on page 165 and "S-VLAN Overview" on page 166.

Related Documentation

- Configuring Dynamic Interfaces Using Bulk Configuration Overview on page 620
- Bulk Configuration of VLAN Ranges Overview on page 662
- Bulk Configuration of VLAN Ranges Using Agent-Circuit-Identifier Information Overview on page 663
- Dynamic VLAN Subinterface Creation Overview on page 665
- Configuring Dynamic VLAN Subinterfaces on page 667

VLAN Base Profile Overview

To configure a dynamic VLAN subinterface over a static VLAN major interface, you must create a base profile. The base profile includes one or more of the following attributes for the VLAN subinterface:

- advisory-rx-speed—Sets an advisory receive speed for VLAN subinterfaces that are created with this profile. For information, see "Setting an Advisory Receive Speed for VLAN Subinterfaces" on page 670.
- advisory-tx-speed—Sets an advisory connect speed for VLAN subinterfaces that are created with this profile. For information, see "Setting an Advisory Connect Speed for VLAN Subinterfaces" on page 670.
- auto-configure—Specifies the types of upper-interface encapsulations that are accepted or detected by the dynamic VLAN subinterface. For information, see "Specifying the Types of Dynamic Upper-Interface Encapsulations that are Accepted or Detected by a Dynamic VLAN Subinterface" on page 670.
- auto-configure agent-circuit-identifier—Enables the creation of VLAN subinterfaces that are based on agent-circuit-identifier information. For information, see "Creating a VLAN Subinterface that is Based on the Agent-Circuit-ID Information" on page 671.
- description—Assigns a description to VLAN subinterfaces that are created with this profile. For information, see "Assigning Descriptions to VLAN Subinterfaces" on page 671.
- policy—Assigns a policy to a VLAN. For information, see "Assigning a VLAN Policy List to a Profile" on page 671.
- profile—Adds a nested profile assignment, which references another profile that dynamically configures an upper-interface encapsulation type over the VLAN subinterface. For information, see "Adding a Nested Profile Assignment to a Base Profile for a Dynamic VLAN Subinterface" on page 672.
- service-profile—Specifies a service profile name for a VLAN. For information, see "Specifying a Service Profile Name for a Dynamic VLAN" on page 673.
- svlan ethertype—Specifies that the packet must use this Ethertype to create the dynamic VLAN subinterface. For more information, see "Specifying the Available Ethertypes to Create a Dynamic VLAN Subinterface" on page 673.

You can override the base profile assignment for a VLAN or S-VLAN that exists with a profile. For more information, see "Overriding VLAN Base Profile Assignments Overview" on page 665.

This topic describes the following:

- VLAN Nested Profile Assignments on page 661
- VLAN Additional Profile Characteristics for Upper Interfaces on page 661

VLAN Nested Profile Assignments

The configuration for each dynamic upper-interface encapsulation type might differ, depending on the column type built by the router. To manage these differences, you can include one or more nested profile assignments within the VLAN base profile. A nested profile assignment references another profile that configures attributes for a dynamic upper-interface encapsulation type. You can create different profiles for each upper-interface encapsulation type, or you can create a single profile that includes attributes for multiple encapsulation types.

For example, the following commands create a base profile named vlanBaseProfile with two nested profile assignments. The first nested profile assignment references an IP profile named vlanProfileIp, and the second nested profile assignment references a PPPoE profile named vlanProfilePppoe.

```
host1(config)#profile vlanBaseProfile
host1(config-profile)#vlan profile ip vlanProfileIp
host1(config-profile)#vlan profile pppoe vlanProfilePppoe
```

In this example, vlanProfileIp and vlanProfilePppoe have different IP configurations depending on the dynamic interface column constructed. For an IP over VLAN dynamic interface column, the router uses the IP attributes in vlanProfileIp. For an IP over PPPoE dynamic interface column, the router uses the IP attributes in vlanProfilePppoe.

For information about creating profiles for upper-interface encapsulation types, see "Dynamic Interface Configuration Using a Profile" on page 565.

VLAN Additional Profile Characteristics for Upper Interfaces

In addition to VLAN attributes and nested profile assignments, the base profile for a dynamic VLAN subinterface can also include individual characteristics for several upper-interface encapsulation types, provided that no nested profile assignment for the specified encapsulation type is in the base profile. If, on the other hand, a nested profile assignment for this encapsulation type exists in the base profile, the router obtains all characteristics for that encapsulation type from the nested profile and not from the base profile.

For lists of the characteristics for each supported upper-interface encapsulation type, see "Profile Characteristics" on page 566.

Related Documentation

- Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview on page 657
- Creating a Base Profile to Configure Attributes for a Dynamic VLAN Subinterface on page 669
- Configuring a Base Profile for VLAN Subinterface on page 669

- *profile*
- *vlan profile*

Bulk Configuration of VLAN Ranges Overview

Dynamic creation of VLAN subinterfaces requires you to configure a range of single-tagged VLAN IDs and double-tagged S-VLAN IDs on the VLAN major interface and assign a name to this range. You can also configure a range of S-VLAN IDs that is based on agent-circuit-identifier information. For more information, see "Bulk Configuration of VLAN Ranges Using Agent-Circuit-Identifier Information Overview" on page 663.

Each VLAN range consists of one or more nonoverlapping VLAN subranges. A VLAN subrange is a group of VLAN IDs and S-VLAN IDs that reside within the VLAN range you specify.

The process of configuring a VLAN range for a dynamic VLAN subinterface is referred to as *bulk configuration*. You create a bulk configuration by issuing the **vlan bulk-config** command. For example, the following commands create a VLAN bulk configuration named myBulkConfig on the specified VLAN interface.

```
host1(config)#interface gigabitEthernet 2/0
host1(config-if)#vlan bulk-config myBulkConfig svlan-range 101 1100 1 375 svlan-range
  1300 1500 500 650
```

In the example, the **vlan bulk-config** command configures a VLAN range made up of two VLAN subranges. The first subrange configures S-VLANs 101–1100 and VLANs 1–375. The second subrange configures S-VLANs 1300–1500 and VLANs 500–650.

NOTE: **For information about the maximum number of VLAN bulk configurations supported per router and line module, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.**

After you issue the **vlan bulk-config** command, the router provisions all VLAN IDs and S-VLAN IDs in the specified VLAN range at the same time. The router does not dynamically create the VLAN subinterface until it receives incoming data traffic on the VLAN ID or S-VLAN ID.

After you create a named VLAN range, you cannot remove the underlying VLAN major interface until you issue the **no vlan bulk-config** command to remove the VLAN range from that interface.

This topic describes the following:

- Changing VLAN Subranges Overview on page 662

Changing VLAN Subranges Overview

You can add, remove, modify, merge, disable, and enable VLAN subranges within an existing bulk-configured VLAN range. For configuration instructions and examples, see "Changing VLAN Subranges" on page 683

Related Documentation

- Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview on page 657

- Bulk Configuration of VLAN Ranges Using Agent-Circuit-Identifier Information Overview on page 663
- Overriding VLAN Base Profile Assignments Overview on page 665
- Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface on page 674
- Changing VLAN Subranges on page 683

Bulk Configuration of VLAN Ranges Using Agent-Circuit-Identifier Information Overview

Using bulk configuration to create S-VLAN IDs based on agent-circuit-identifier information is similar to the process of creating a bulk-configured VLAN range that is not based on agent-circuit-identifier information. However, when you issue the **vlan bulk-config** command with the **svlan-range** keyword to specify the S-VLAN ID range, you then specify the **agent-circuit-identifier** keyword instead of a VLAN ID range. This technique creates a unique type of S-VLAN range in which the agent-circuit-identifier information is used in place of the second tag.

The agent-circuit-identifier string is contained in the option 82 field of DHCP messages for DHCP traffic, or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets for PPPoE traffic. The agent-circuit-identifier information identifies the subscriber's access node and the DSL line on the access node. You can repeat the **svlan-range** and **agent-circuit-identifier** keywords to provide nonoverlapping VLAN subranges that reside within the VLAN range.

The following example configures a VLAN ID range made up of two subranges. The first subrange configures S-VLANs 200–250 and the second subrange configures S-VLANs 3000–3500. Both subranges configure the subscriber identification based on agent-circuit-identifier information.

```
host1(config)#interface gigabitEthernet 2/0
host1(config-if)#vlan bulk-config myAgent2BulkConfig svlan-range 200 250
  agent-circuit-identifier svlan-range 3000 3500 agent-circuit-identifier
```

After you issue the **vlan bulk-config** command with the **agent-circuit-identifier** keyword, the router provisions the S-VLAN IDs in the specified bulk-configured VLAN range at the same time. The router does not dynamically create the VLAN subinterface until it receives incoming data traffic. The user information is generated from the incoming data traffic that contains the agent-circuit-identifier string.

Conceptually, a VLAN subinterface in this configuration has two attributes, an S-VLAN ID and an agent-circuit-identifier string. This is analogous to a regular S-VLAN that also has two attributes, an S-VLAN ID and a VLAN ID. However, the packet that the router receives is singly-tagged with only a VLAN ID. The use of the **agent-circuit-identifier** keyword in the **vlan bulk-config** command causes the router to further examine the packet and extract the agent-circuit-identifier string in order to generate the subscriber identification information.

In a DSL access network, subscriber information can be conveyed through either of the following methods:

- VLAN encapsulation; that is, the S-VLAN ID and the VLAN ID
- Insertion of the agent-circuit-identifier string in DHCP or PPPoE messages

For example, the following configurations uniquely identify subscribers by means of VLAN encapsulation:

- Subscriber packets received from the DSLAM are single-tagged with a VLAN ID
- Subscriber packets received from the DSLAM are double-tagged with both an S-VLAN ID and a VLAN ID

The DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006) refers to the behavior of these configurations as the 1:1 forwarding model because there is a one-to-one correspondence between an individual subscriber and the VLAN encapsulation.

In contrast, the following configurations do *not* uniquely identify subscribers by means of VLAN encapsulation:

- Subscriber packets received from the DSLAM are single-tagged with the same VLAN ID for a group of subscribers. This configuration is typically used to implement service VLANs where the VLAN ID corresponds to the type of service for which the VLAN is used, such as voice or video. In this configuration, the VLAN ID does not correspond to an individual subscriber.
- Subscriber packets received from the DSLAM are untagged.

Instead, these configurations identify subscribers by means of the agent-circuit-identifier information present in DHCP and PPPoE control messages. DSL Forum TR-101 refers to the behavior of these configurations as the N:1 forwarding model because there is a many-to-one correspondence between subscribers and a VLAN.

Creating dynamic VLANs based on agent-circuit-identifier information enables you to manage subscribers in single-tagged or untagged N:1 configurations that do not use encapsulation to uniquely identify subscribers. In these configurations, the router intercepts the agent-circuit-identifier string from DHCP messages or from PPPoE PADR and PADI packets to build a unique subscriber interface.

For double-tagged 1:1 configurations, the router uses standard dynamic VLAN procedures to uniquely identify subscribers. In these configurations, the S-VLAN ID typically represents the DSLAM, and the VLAN ID represents the individual subscriber accessing the router through that DSLAM.

For configuration instructions, see "Configuring Dynamic VLAN Subinterfaces Based on Agent Circuit Identifier Information" on page 676.

NOTE: You must configure the DHCP local or external server to support the creation of dynamic subscriber interfaces that are based on the agent-circuit-id option (suboption 1) of the option 82 field in DHCP messages. See *Configuring the DHCP Local Server* or *DHCP External Server Overview* for information.

Related Documentation

- Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview on page 657
- Bulk Configuration of VLAN Ranges Overview on page 662
- Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface on page 674
- *vlan bulk-config*

Dynamic VLAN Subinterface Creation Overview

After you configure the base profile, you associate it with the VLAN major interface by issuing ***profile vlan bulk-config*** command.

As a final step, you must issue ***auto-configure vlan*** command. This command configures the VLAN major interface to support autodetection of the VLAN dynamic encapsulation type.

When the router receives an incoming data packet on a circuit, it dynamically creates the VLAN subinterface, using the attributes specified in the base profile. After examining the contents of the data packet, the router dynamically creates the required interface columns above the VLAN subinterface, using the configuration attributes contained in the nested profiles, if specified, or in the base profile itself.

Related Documentation

- Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview on page 657
- Assigning the Base Profile Configured for a Dynamic VLAN Subinterface to the Single-Tagged VLAN IDs or Double-Tagged S-VLAN IDs on page 681
- Configuring the Static VLAN Major Interface to Support Autodetection of a VLAN Dynamic Interface Type on page 675
- *auto-configure vlan*
- *profile vlan bulk-config*

Overriding VLAN Base Profile Assignments Overview

You can also use the ***profile vlan override bulk-config*** command to assign an overriding profile to a single VLAN ID or double-tagged S-VLAN ID that exists within a bulk-configured VLAN subrange. The VLAN ID subrange that encompasses the major interface must have been previously configured with the ***vlan bulk-config*** command for use by a dynamic VLAN subinterface. After you assign the overriding profile, the router uses the information in this profile instead of the information in the previously assigned base profile to create any subsequent VLAN dynamic subinterface columns on the specified VLAN major interface, as long as they match the VLAN or S-VLAN specified in the override.

The overriding profile, like the original base profile, can include VLAN attributes, nested profile assignments, and individual characteristics for dynamic upper-interface encapsulation types.

Overriding the base profile assignment for a VLAN with a profile enables you to create a special profile for a subscriber in a DSLAM. For example, you can use the overriding profile to create dynamic VLAN subinterfaces for subscribers with an S-VLAN ID of 200 and a VLAN ID of 100.

You can also use an overriding profile with debugging attributes to troubleshoot problems with VLAN dynamic subinterface columns.

NOTE: See *JunosE Release Notes*, *Appendix A*, *System Maximums* for information about the maximum number of overriding profile assignments currently supported per chassis.

Related Documentation

- VLAN Base Profile Overview on page 660
- Bulk Configuration of VLAN Ranges Overview on page 662
- Assigning an Overriding Profile to a Single-Tagged VLAN ID or Double-Tagged S-VLAN ID on page 682
- Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface on page 674
- Configuring Overriding Profile Assignments for VLAN Major Interfaces on page 678
- *profile vlan override bulk-config*
- *vlan bulk-config*

Static VLAN Subinterfaces Within VLAN Subranges Overview

You can configure a static VLAN subinterface with a single-tagged VLAN ID or double-tagged S-VLAN ID, or an S-VLAN ID with agent-circuit-identifier information that falls within an existing bulk-configured VLAN subrange. Conversely, you can also create a bulk-configured VLAN subrange that includes the single-tagged VLAN ID or double-tagged S-VLAN ID on a static VLAN subinterface. Configuring static VLAN subinterfaces within VLAN subranges can be useful when you want to create a column statically for users who have difficulty logging on. You might also want to configure static VLAN subinterface within a VLAN subrange as a static column to the DSLAM; the dynamic column can be for subscribers.

The following rules apply when you configure either a static VLAN subinterface within an existing bulk-configured VLAN subrange or a subrange that includes an existing static VLAN interface:

- You have no restrictions on how to configure the static VLAN subinterface.
- Static VLAN interfaces defined within a bulk-configured VLAN subrange are stored in NVS and preserved after a reboot.
- The base profile associated with the VLAN subrange does not apply to any statically defined VLAN interfaces that fall within the subrange.
- If a VLAN subrange includes a statically defined VLAN subinterface, overriding profile assignments configured for the same VLAN ID as a statically defined VLAN subinterface become inactive until the static VLAN subinterface is removed. The overriding profile

becomes active again when you remove the static VLAN subinterface. You can display the current operational status (active or inactive) of overriding profile assignments by using the **show vlan bulk-config** command.

- Operations that add, remove, modify, merge, disable, or enable VLAN subranges within a bulk-configured VLAN range do not affect any static VLAN subinterfaces defined within the VLAN subrange.
- You cannot create a static VLAN if the single-tagged VLAN ID or double-tagged S-VLAN ID conflicts with an existing VLAN dynamic subinterface column. Such a configuration would disrupt subscribers already connected to the router via the dynamic subinterface.

Related Documentation

- Bulk Configuration of VLAN Ranges Overview on page 662
- Configuring Dynamic VLAN Subinterfaces Based on Agent Circuit Identifier Information on page 676
- Changing VLAN Subranges on page 683
- Configuring Static VLAN Subinterfaces Within VLAN Subranges on page 688

Configuring Dynamic VLAN Subinterfaces

To configure a dynamic VLAN subinterface:

1. Configure profiles containing characteristics for the dynamic upper-interface encapsulation types to be created over the dynamic VLAN subinterface.

   These profiles are referenced in the base profile for the dynamic VLAN subinterface as nested profile assignments. For detailed instructions on creating profiles, see "Dynamic Interface Configuration Using a Profile" on page 565.

2. (Optional) Create the profile for an upper-interface encapsulation type, and include additional profile characteristics for other encapsulation types as needed. Perform this step if you want to create a nested profile assignment in Step 5.

   ```
   host1(config)#profile myIpProfile
   host1(config-profile)#ip inactivity-timer 200
   host1(config-profile)#ip auto-configure ip-subscriber include-primary
   ```

3. Create the base profile for the dynamic VLAN subinterface by assigning the profile a name. For information, see "Creating a Base Profile to Configure Attributes for a Dynamic VLAN Subinterface" on page 669.

   ```
   host1(config)#profile vlanBaseProfile
   ```

   This command accesses Profile Configuration mode, which enables you to configure attributes in the base profile.

4. Define attributes for the VLAN subinterface in the base profile. For more information, see "Configuring a Base Profile for VLAN Subinterface" on page 669.

   a. Configure the VLAN major interface for autodetection of the PPPoE upper-interface encapsulation type.

      ```
      host1(config-profile)#vlan auto-configure pppoe
      ```

b. Configure the VLAN subinterface for autodetection of the IP upper-interface encapsulation type.

```
host1(config-profile)#vlan auto-configure ip
```

c. Configure an Ethertype value for any S-VLANs configured on the VLAN.

```
host1(config-profile)#svlan ethertype 8100
```

5. (Optional) In the base profile, create nested profile assignments for the upper-interface encapsulation types.

```
host1(config-profile)#vlan profile ip myIpProfile
```

6. Exit Profile Configuration mode.
7. Configure the VLAN major interface. For more information, see "Configuring VLAN as the Encapsulation Method for the Interface" on page 682.

```
host1(config)#interface gigabitEthernet 5/0
host1(config-if)#encapsulation vlan
```

8. Configure a VLAN range on the major VLAN interface, and assign a name to this range. For more information, see "Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface" on page 674

NOTE: For information about the maximum number of VLAN bulk configurations supported per chassis, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

The following command creates a VLAN range named myBulkConfig made up of two VLAN subranges.

```
host1(config-if)#vlan bulk-config myBulkConfig vlan-range 0 100
  vlan-range 110 200
```

9. Assign the base profile configured for the VLAN subinterface to the VLAN range configured on the major VLAN interface. For more information, see "Assigning the Base Profile Configured for a Dynamic VLAN Subinterface to the Single-Tagged VLAN IDs or Double-Tagged S-VLAN IDs" on page 681.

```
host1(config-if)#profile vlan bulk-config myBulkConfig vlanBaseProfile
```

10. Configure the VLAN major interface to support autodetection of the VLAN dynamic encapsulation type. For more information, see "Configuring the Static VLAN Major Interface to Support Autodetection of a VLAN Dynamic Interface Type" on page 675.

```
host1(config-if)#auto-configure vlan
```

Related Documentation

- Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview on page 657
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606
- Monitoring Status and Configuration Information for VLAN Subinterfaces on page 614
- Monitoring Base Profile Assignments and Overriding Profile Assignments for VLAN Major Interface on page 716

- *auto-configure vlan*
- *encapsulation vlan*
- *profile*
- *profile vlan bulk-config*
- *svlan ethertype*
- *vlan auto-configure*
- *vlan bulk-config*
- *vlan profile*

Creating a Base Profile to Configure Attributes for a Dynamic VLAN Subinterface

You can use the **profile** command to create a base profile to configure attributes for a dynamic VLAN subinterface. You can specify a profile name of up to 80 alphanumeric characters.

To create a base profile to configure attributes for a dynamic VLAN subinterface:

- Issue the **profile** command in Global Configuration mode.

```
host1(config)#profile vlanBaseProfile
```

Use the **no** version to delete the specified profile.

Related Documentation

- VLAN Base Profile Overview on page 660
- Configuring a Base Profile for VLAN Subinterface on page 669
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606

- *profile*

Configuring a Base Profile for VLAN Subinterface

This topic includes the following tasks:

- Setting an Advisory Receive Speed for VLAN Subinterfaces on page 670
- Setting an Advisory Connect Speed for VLAN Subinterfaces on page 670
- Specifying the Types of Dynamic Upper-Interface Encapsulations that are Accepted or Detected by a Dynamic VLAN Subinterface on page 670
- Creating a VLAN Subinterface that is Based on the Agent-Circuit-ID Information on page 671
- Assigning Descriptions to VLAN Subinterfaces on page 671
- Assigning a VLAN Policy List to a Profile on page 671
- Adding a Nested Profile Assignment to a Base Profile for a Dynamic VLAN Subinterface on page 672

- Specifying the Available Ethertypes to Create a Dynamic VLAN Subinterface on page 673
- Specifying a Service Profile Name for a Dynamic VLAN on page 673

Setting an Advisory Receive Speed for VLAN Subinterfaces

You can use the **vlan advisory-rx-speed** command to set an advisory receive speed for VLAN subinterfaces that are created with a VLAN base profile. This setting has no effect on data forwarding. You can use it to indicate the speed of the client interface. When traffic is tunneled with L2TP, the advisory receive speed is sent from the LAC to the LNS. See *LAC Configuration Prerequisites* for additional information about the advisory receive speed. The range is 0–2147483647 kbps; 0 indicates no advisory speed setting.

To set an advisory receive speed for VLAN subinterfaces:

- Issue the **vlan advisory-rx-speed** command in Profile Configuration mode.

```
host1(config-profile)#vlan advisory-rx-speed 2000
```

Use the **no** version to restore the default behavior—the Rx speed is not sent to the LNS.

Setting an Advisory Connect Speed for VLAN Subinterfaces

You can use the **vlan advisory-tx-speed** command to set an advisory connect speed for VLAN subinterfaces that are created with a VLAN base profile..

This setting has no effect on data forwarding. You can use it to indicate the speed of the client interface. When traffic is tunneled with L2TP, the advisory receive speed is sent from the LAC to the LNS. See *LAC Configuration Prerequisites* for additional information about the advisory receive speed. The range is 0–2147483647 kbps; 0 indicates no advisory speed setting.

To set an advisory connect speed for VLAN subinterfaces:

- Issue the **vlan advisory-tx-speed** command in Profile Configuration mode.

```
host1(config-profile)#vlan advisory-tx-speed 2000
```

Use the **no** version to restore the default behavior—the Tx speed is not sent to the LNS.

Specifying the Types of Dynamic Upper-Interface Encapsulations that are Accepted or Detected by a Dynamic VLAN Subinterface

You can use the **vlan auto-configure** command to specify the types of dynamic upper-interface encapsulations that are accepted or detected by a dynamic VLAN subinterface. You can include this command in the base profile for a dynamic VLAN subinterface.

You can use the **lockout-time** keyword to specify the minimum and maximum lockout time range for the encapsulation type. For more information, see "Dynamic Encapsulation Type Lockout" on page 532.

To specify the types of dynamic upper-interface encapsulations that are accepted or detected by a dynamic VLAN subinterface:

- Issue the **vlan auto-configure** command in Profile Configuration mode.

  host1(config-profile)#**vlan auto-configure ip**

Use the **no** version to terminate detection of the specified encapsulation type.

Creating a VLAN Subinterface that is Based on the Agent-Circuit-ID Information

You can use the **vlan auto-configure agent-circuit-identifier** command to create a VLAN subinterface that is based on the agent-circuit-id information in the option 82 field of DHCP messages or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets. You can include this command in the base profile for a dynamic VLAN subinterface.

To create a VLAN subinterface:

- Issue the **vlan auto-configure agent-circuit-identifier** command in Profile Configuration mode.

  host1(config-profile)#**vlan auto-configure agent-circuit-identifier**

Use the **no** version to disable creation of VLAN subinterfaces based on agent-circuit-identifier information.

Assigning Descriptions to VLAN Subinterfaces

You can use the **vlan description** command to assign a description to VLAN subinterfaces that are created with a base profile. You can use a maximum of 64 characters for the description or to name the alias.

To assign a description to VLAN subinterfaces:

- Issue the **vlan description** command in Profile Configuration mode.

  host1(config-profile)#**vlan description test1**

Use the **no** version to remove the VLAN description.

Assigning a VLAN Policy List to a Profile

You can use the **vlan policy** command to assign a VLAN policy list to a profile, which then assigns the policy to an interface.

You can use the **input** or **output** keyword to assign the policy list to the ingress or egress of the interface. You can enable or disable the recording of routing statistics for bytes and packets affected by the policy.

You can use the **preserve** keyword to save the existing statistics when you attach a policy to an interface that already has a policy attached. This keyword saves the statistics for any classifier-list that is the same for both the new and old policy attachments. Without the **preserve** keyword, all statistics are deleted when you attach the new policy.

For example, when you replace a policy attachment that references the original policy-list plOne with a new attachment referencing policy-list plTwo, the existing statistics for the classifier group referencing clOne and the default classifier group are saved.

Original Policy Attachment	New Policy Attachment	Comment
ip policy-list plOne	ip policy-list plTwo	–
ip classifier-list clOne	ip classifier-list clOne	statistics from plOne are saved
Forward	Forward	–
ip classifier-list clTwo	ip classifier-list clFour	–
Forward	Forward	–
ip classifier-list clThree	ip classifier-list clFive	–
Forward	Forward	–
classifier-list *	classifier-list *	statistics from plOne are saved
Filter	Filter	–

To assign a VLAN policy list to a profile:

- Issue the **vlan policy** command in Profile Configuration mode.

```
host1(config-profile)#vlan policy input VlanPolicy33 statistics enabled preserve
```

Use the **no** version to remove the policy reference from the profile.

Adding a Nested Profile Assignment to a Base Profile for a Dynamic VLAN Subinterface

You can use the **vlan profile** command to add a nested profile assignment to a base profile for a dynamic VLAN subinterface. A nested profile assignment references another profile that configures attributes for a dynamic upper-interface type over the VLAN subinterface.

To add a nested profile assignment to a base profile for a dynamic VLAN subinterface:

- Issue the **vlan profile** command in Profile Configuration mode.

```
host1(config-profile)#vlan profile pppoe vlanProfilePppoe
host1(config-profile)#vlan profile ip vlanProfileIP
```

Use the **no** version to remove the profile assignment for the upper-interface encapsulation type.

Specifying the Available Ethertypes to Create a Dynamic VLAN Subinterface

You can use the **svlan ethertype** command to specify the available Ethertypes that a packet must use to create a dynamic VLAN subinterface.

You can choose one of the following Ethertype values:

- **8100**—Specifies Ethertype value 0x8100, as defined in IEEE Standard 802.1q
- **88a8**—Specifies Ethertype value 0x88a8, as defined in draft IEEE Standard 802.1ad
- **9100**—Specifies Ethertype value 0x9100
- **autoconfig**—Specifies that the packet can use any Ethertype to create a dynamic VLAN subinterface

To specify the available Ethertypes that a packet must use to create a dynamic VLAN subinterface:

- Issue the **svlan ethertype** command in Profile Configuration mode.

```
host1(config-profile)#svlan ethertype 8100
host1(config-profile)#svlan ethertype autoconfig
```

Use the **no** version to restore the default value, autoconfig.

Specifying a Service Profile Name for a Dynamic VLAN

You can use the **vlan service-profile** command to assign an IP service profile to a VLAN subinterface. Service profiles contain user and password information, and are used in route maps for subscriber management and to authenticate subscribers with RADIUS. You can specify a service profile name with up to 80 alphanumeric characters.

To assign an IP service profile to a VLAN subinterface:

- Issue the **vlan service-profile** command in Profile Configuration mode.

```
host1(config-profile)#vlan service-profile ipprofile
```

Use the **no** version removes the IP service profile from the VLAN subinterface.

Related Documentation

- VLAN Base Profile Overview on page 660
- Creating a Base Profile to Configure Attributes for a Dynamic VLAN Subinterface on page 669
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606

- *svlan ethertype*
- *vlan advisory-rx-speed*
- *vlan advisory-tx-speed*
- *vlan auto-configure*
- *vlan auto-configure agent-circuit-identifier*

- *vlan description*
- *vlan policy*
- *vlan profile*

Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface

You can use the **vlan bulk-config** command to create a bulk-configured VLAN range on a static VLAN major interface for use by a dynamic VLAN subinterface.

NOTE: For information about the maximum number of VLAN bulk configurations supported per chassis, see *JunosE Release Notes*, *Appendix A*, *System Maximums*.

Each VLAN range consists of one or more nonoverlapping VLAN subranges. A VLAN subrange is a group of VLAN IDs or S-VLAN IDs that reside within the VLAN range you specify. You can configure multiple VLAN ranges on a VLAN subinterface.

When you create a bulk-configured VLAN range, you must specify the following:

- A name of up to 80 alphanumeric characters; this is also referred to as the bulk configuration name
- The starting and ending VLAN ID or S-VLAN ID values (inclusive) for each VLAN subrange

You can use the **any** keyword to specify a VLAN ID as a wildcard. When you specify the **any** keyword with an S-VLAN ID of a DSLAM, all subscribers associated with the DSLAM will be created with the same profile.

You can use the **agent-circuit-identifier** keyword to configure a VLAN range that is based on the agent-circuit-id information in the option 82 field of DHCP messages or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets. When you specify the **agent-circuit-identifier** keyword with an S-VLAN ID of a DSLAM, all subscribers associated with the DSLAM are created with the same profile.

You can create a placeholder VLAN range by issuing the **vlan bulk-config** command without specifying any subranges. You can assign a profile to this placeholder and add subranges to it later.

You can add and remove individual VLAN subranges. You cannot remove a VLAN subrange if any dynamic VLAN subinterfaces currently exist within the subrange. Use the vlan bulk-config shutdown command to remove dynamic VLAN interfaces created within a subrange. Removal of a subrange automatically results in the removal of all overriding profile assignments on that subrange.

You can create a bulk-configured VLAN subrange that includes the VLAN IDs and S-VLAN IDs belonging to an existing VLAN major interface on a static VLAN subinterface.

To create a bulk-configured VLAN range on a static VLAN major interface for use by a dynamic VLAN subinterface:

- Configure a VLAN range named myBulkConfig with a single VLAN subrange containing VLAN IDs 100–500

  host1(config-if)#**vlan bulk-config myBulkConfig vlan-range 100 500**

- Configure a VLAN range named myMultiBulkConfig with two VLAN subranges containing S-VLAN IDs 101–600 with VLAN IDs 0–1 (first subrange) and S-VLAN IDs 201–3200 with VLAN IDs 3–5 (second subrange)

  host1(config-if)#**vlan bulk-config myMultiBulkConfig svlan-range 101 600 0 1 svlan-range 201 3200 3 5**

- Configure a VLAN range named myAciBulkConfig containing S-VLAN IDs 200–400. Subscriber information is determined by the packet's agent-circuit-identifier information

  host1(config-if)#**vlan bulk-config myAciBulkConfig svlan-range 200 400 agent-circuit-identifier**

Use the **no** version to remove the specified VLAN range from the VLAN interface, to remove the specified subranges from the specified VLAN range, or to remove all subranges from the specified VLAN range. The **no** version also removes any overriding profile assignments for VLAN major interfaces within the deleted VLAN range or VLAN subrange.

Related Documentation

- Bulk Configuration of VLAN Ranges Overview on page 662
- Bulk Configuration of VLAN Ranges Using Agent-Circuit-Identifier Information Overview on page 663
- Monitoring Base Profile Assignments and Overriding Profile Assignments for VLAN Major Interface on page 716

- *vlan bulk-config*

Configuring the Static VLAN Major Interface to Support Autodetection of a VLAN Dynamic Interface Type

You can use the **auto-configure vlan** command to configure the static VLAN major interface to support autodetection of an VLAN dynamic interface type. Issue this command to enable creation of a dynamic VLAN subinterface. By default, all valid VLAN IDs and S-VLAN IDs are accepted.

To configure the static VLAN major interface to support autodetection of an VLAN dynamic interface type:

- Issue the **auto-configure vlan** command in Interface Configuration mode.

host1(config-if)#**auto-configure vlan**

Use the **no** version to terminate autodetection of the VLAN dynamic interface type.

Related Documentation

- Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview on page 657
- Dynamic VLAN Subinterface Creation Overview on page 665

- Monitoring Status and Configuration Information for VLAN Subinterfaces on page 614
- *auto-configure vlan*

Configuring Dynamic VLAN Subinterfaces Based on Agent Circuit Identifier Information

The procedure you use to configure a dynamic VLAN subinterface that is based on agent-circuit-identification information is similar to the procedure described in "Configuring Dynamic VLAN Subinterfaces" on page 667.

NOTE: S-VLAN Ethertype is not supported for ACI-based subscribers because the S-VLAN tag is not present in received frames from subscribers to validate the Ethertype value.

For ACI-based subscribers, the frames that the router receivers contain only the single-tagged VLAN ID and the ACI value is derived either from DHCP options or from PPPoE. In the interface controller (IC) on the router, the Ethernet application does not update the S-VLAN Ethertype value as a measure of bits of VLAN. As a result, the IC sends an exception for the received frame to the system controller (SC). The frame is not processed and is discarded because the Ethertype value of the frame does not match with the S-VLAN Ethertype of 0x8100 in the profile for ACI-based subscribers. This behavior is expected for ACI-based subscribers for which the configuration of S-VLAN Ethertype in dynamic profiles is not valid.

1. Configure profiles containing characteristics for the dynamic upper-interface encapsulation types to be created over the dynamic VLAN subinterface.
2. (Optional) If you want to create a nested profile assignment, create the profile for an upper-interface encapsulation type, and include additional profile characteristics for other encapsulation types as needed.
3. Create the base profile for the dynamic VLAN subinterface and enter Profile Configuration mode by assigning the profile a name. For more information, see "Creating a Base Profile to Configure Attributes for a Dynamic VLAN Subinterface" on page 669.

   host1(config)#**profile vlanMyBaseProfile**

4. Define attributes for the VLAN subinterface in the base profile. For more information, see "Configuring a Base Profile for VLAN Subinterface" on page 669.

   a. Enable autoconfiguration for the PPPoE upper-interface encapsulation type.

      host1(config-profile)#**vlan auto-configure pppoe**

   b. Enable autoconfiguration for the IP upper-interface encapsulation type.

      host1(config-profile)#**vlan auto-configure ip**

   c. Enable autoconfiguration of VLANs that are based on agent-circuit-identifier information.


```
host1(config-profile)#vlan auto-configure agent-circuit-identifier
host1(config-profile)#exit
host1(config)#
```

d. (Optional) Create nested profile assignments for the upper-interface encapsulation types.

5. Configure the VLAN major interface. For more information, see "Configuring VLAN as the Encapsulation Method for the Interface" on page 682.

```
host1(config)#interface gigabitEthernet 5/0
host1(config-if)#encapsulation vlan
```

6. On the VLAN major interface, configure a VLAN range that is based on agent-circuit-identifier information, and assign a name to this range. For more information, see "Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface" on page 674

```
host1(config-if)#vlan bulk-config myNewBulkConfig svlan-range 50 100
  agent-circuit-identifier
```

7. Assign the base profile configured for the VLAN subinterface to the VLAN range configured on the major VLAN interface. For more information, see "Assigning the Base Profile Configured for a Dynamic VLAN Subinterface to the Single-Tagged VLAN IDs or Double-Tagged S-VLAN IDs" on page 681.

```
host1(config-if)#profile vlan bulk-config myNewBulkConfig vlanMyBaseProfile
```

8. Configure the VLAN major interface to support autodetection of the VLAN dynamic encapsulation type. For more information, see "Configuring the Static VLAN Major Interface to Support Autodetection of a VLAN Dynamic Interface Type" on page 675.

```
host1(config-if)#auto-configure vlan
```

Related Documentation

- Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview on page 657
- Bulk Configuration of VLAN Ranges Using Agent-Circuit-Identifier Information Overview on page 663
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 606
- Monitoring Base Profile Assignments and Overriding Profile Assignments for VLAN Major Interface on page 716

- *auto-configure vlan*
- *encapsulation vlan*
- *vlan profile*
- *vlan auto-configure*
- *vlan auto-configure agent-circuit-identifier*
- *vlan bulk-config*

Configuring Overriding Profile Assignments for VLAN Major Interfaces

Configuring overriding profile assignments for VLAN major interfaces includes the following tasks:

- Assigning an Overriding Profile to a VLAN Within a Bulk-Configured VLAN Subrange on page 678
- Removing an Overriding Profile Assignment from a VLAN on page 679
- Removing Overriding Profile Assignments from a VLAN Range or VLAN Subrange on page 680

Assigning an Overriding Profile to a VLAN Within a Bulk-Configured VLAN Subrange

You can assign an overriding profile to a single VLAN major interface within a bulk-configured VLAN subrange.

The overriding profile includes debugging attributes to help you identify and troubleshoot problems with the VLAN dynamic subinterface column created on the specified VLAN ID.

To assign an overriding profile to a VLAN within a bulk-configured VLAN subrange:

1. Configure both of the following:
   - Base profile for the bulk-configured dynamic VLAN on the static VLAN major interface. The VLAN range consists of one or more VLAN subranges. For more information, see "Configuring a Base Profile for VLAN Subinterface" on page 669
   - Overriding profile for a dynamic VLAN within a bulk-configured VLAN subrange.

   For information about configuring profiles, see "Dynamic Interface Configuration Using a Profile" on page 565.

2. Create a bulk-configured range of single-tagged VLAN IDs or double-tagged S-VLAN IDs on a static VLAN major interface. The following commands create a bulk-configured VLAN range named myBulkConfig that consists of two VLAN subranges. The first subrange encompasses VLAN IDs 150–250. The second subrange encompasses VLAN IDs 300–500. For more information, see *Selecting a Gigabit Ethernet Interface or a 10-Gigabit Ethernet Interface* and "Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface" on page 674.

   ```
   host1(config)#interface gigabitEthernet 4/0.101
   host1(config-if)#vlan bulk-config myBulkConfig vlan-range 150 250
     vlan-range 300 500
   ```

3. Assign the previously configured base profile (vlanBaseProfile) to the bulk-configured VLAN range. For more information, see "Assigning the Base Profile Configured for a Dynamic VLAN Subinterface to the Single-Tagged VLAN IDs or Double-Tagged S-VLAN IDs" on page 681.

   ```
   host1(config-if)#profile vlan bulk-config myBulkConfig vlanBaseProfile
   ```


4. Assign the previously configured overriding profile to a single VLAN ID or double-tagged S-VLAN ID within the bulk-configured VLAN subrange. The following command assigns the overriding profile overrideVoiceSubscriber to the VLAN ID 202. This VLAN ID exists within the first VLAN subrange (VLAN IDs 150–250) configured in Step 2.

   host1(config-if)#**profile vlan override bulk-config myBulkConfig vlan 202 overrideVoiceSubscriber**

   The router now uses the information in the overriding profile instead of the information in the base profile to create subsequent VLAN dynamic subinterface columns over this VLAN ID. For more information, see "Assigning an Overriding Profile to a Single-Tagged VLAN ID or Double-Tagged S-VLAN ID" on page 682.

5. (Optional) You can assign the same overriding profile to a VLAN ID within the same VLAN range or within a different VLAN range. For example, the following command assigns the overriding profile overrideVoiceSubscriber to the VLAN ID 160. This S-VLAN ID exists within the VLAN subrange configured in Step 2.

   host1(config-if)#**profile vlan override bulk-config-name myBulkConfig svlan 120 202 overrideVoiceSubscriber**

   NOTE: You can reverse the order of Step 2 and Step 4 with identical results. That is, you can assign the overriding profile to an S-VLAN ID and then assign the base profile to the entire VLAN subinterface.

6. Configure the VLAN major interface to support autodetection of the VLAN dynamic encapsulation type. For more information, see "Configuring the Static VLAN Major Interface to Support Autodetection of a VLAN Dynamic Interface Type" on page 675.

   host1(config-if)#**auto-configure vlan**

7. (Optional) Use the **show vlan profile** command to verify the overriding profile configuration. For more information, see "Monitoring Dynamic VLAN Subinterfaces Created with an Overriding Profile Assignment" on page 718.

Removing an Overriding Profile Assignment from a VLAN

You can remove an overriding profile assignment from a VLAN major interface.

If you use the overriding profile to troubleshoot the VLAN dynamic subinterface column created on the specified VLAN ID, make sure that you remove the overriding profile assignment to restore the original base profile assignment. This action ensures that subsequent VLAN dynamic subinterface columns are created using the same attributes defined in the base profile.

To remove an overriding profile assignment from a VLAN:

1. Remove the overriding profile assignment from the specified VLAN ID or S-VLAN ID.

   host1(config-if)#**no profile vlan override bulk-config-name myBulkConfig vlan 202 overrideVoiceSubscriber**

2. Select the dynamic VLAN subinterface on which the VLAN dynamic subinterface column resides. For more information, see *Selecting a Gigabit Ethernet Interface or a 10-Gigabit Ethernet Interface*.

   host1(config)#interface gigabitEthernet 4/0.101

3. Use the **shutdown** command to disable the dynamic VLAN subinterface. The **shutdown** command deletes the VLAN dynamic subinterface column and removes the dynamic VLAN subinterface. For more information, see "Disabling a Dynamic VLAN subinterface" on page 683.

   host1(config-if)#shutdown

4. Send traffic over the VLAN subinterface. This action re-creates the VLAN dynamic subinterface column with the original base profile association.

   The router now uses the information in the base profile instead of the information in the overriding profile to create subsequent VLAN dynamic subinterface columns for the specified VLAN ID or S-VLAN ID.

5. (Optional) Use the **show vlan profile override** command to verify the removal of the overriding profile assignment. For more information, see "Monitoring Dynamic VLAN Subinterfaces Created with an Overriding Profile Assignment" on page 718.

Removing Overriding Profile Assignments from a VLAN Range or VLAN Subrange

You can use the **no vlan bulk-config** command to remove an entire VLAN range (and all VLAN subranges within that VLAN range). When you issue the **no vlan bulk-config** command, the router removes any overriding profile assignments configured for VLAN IDs within those VLAN subranges.

To remove the bulk-configured VLAN range named myBulkConfig and any overriding profile assignments for VLAN IDs within the VLAN subranges belonging to myBulkConfig:

- Issue the **no vlan bulk-config** command in Interface Configuration mode.

  host1(config-if)#no vlan bulk-config myBulkConfig

You can also use the **no vlan bulk-config** command to remove a particular VLAN subrange in a bulk-configured VLAN range. When you issue the **no vlan bulk-config** command, the router removes any overriding profile assignments for VLAN IDs within that VLAN subrange. However, overriding profile assignments for VLAN IDs within other VLAN subranges in the VLAN range remain intact.

To remove one VLAN subrange (S-VLAN IDs 50–150 and VLAN IDs 150–250) and only those overriding profile assignments associated with this subrange:

- Issue the **no vlan bulk-config** command in Interface Configuration mode.

  host1(config-if)#no vlan bulk-config myBulkConfig svlan-range 50 150 150 250

Related Documentation

- Overriding VLAN Base Profile Assignments Overview on page 665
- Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface on page 674

- Monitoring Base Profile Assignments and Overriding Profile Assignments for VLAN Major Interface on page 716
- Monitoring Dynamic VLAN Subinterfaces Created with an Overriding Profile Assignment on page 718
- *interface gigabitEthernet*
- *profile vlan bulk-config*
- *profile vlan override bulk-config*
- *show vlan bulk-config*
- *show vlan profile*
- *shutdown*
- *vlan bulk-config*

Assigning the Base Profile Configured for a Dynamic VLAN Subinterface to the Single-Tagged VLAN IDs or Double-Tagged S-VLAN IDs

You can use the **profile vlan bulk-config** command to assign the base profile configured for a dynamic VLAN subinterface to the single-tagged VLAN IDs or double-tagged S-VLAN IDs configured on a static VLAN major interface.

To assign the base profile configured for a dynamic VLAN subinterface to the single-tagged VLAN IDs or double-tagged S-VLAN IDs configured on a static VLAN major interface, you must specify the following:

- Name assigned to the VLAN range on a VLAN subinterface, as specified in "Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface" on page 674
- Name assigned to the base profile for a dynamic VLAN subinterface

To assign the base profile configured for a dynamic VLAN subinterface:

- Issue the **profile vlan bulk-config** command in Interface Configuration mode.

```
host1(config-if)#profile vlan bulk-config myBulkConfig vlanBaseProfile
```

Use the **no** version to remove the base profile assignment.

Related Documentation

- Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview on page 657
- Bulk Configuration of VLAN Ranges Overview on page 662
- Bulk Configuration of VLAN Ranges Using Agent-Circuit-Identifier Information Overview on page 663
- Monitoring Base Profile Assignments and Overriding Profile Assignments for VLAN Major Interface on page 716
- *profile vlan bulk-config*
- *vlan bulk-config*

Assigning an Overriding Profile to a Single-Tagged VLAN ID or Double-Tagged S-VLAN ID

You can use the **profile vlan override bulk-config** command to assign an overriding profile to a single VLAN ID or double-tagged S-VLAN ID. Using an overriding profile enables you to assign a special profile for the subscribers associated with a specific DSLAM.

You can also use an overriding profile to troubleshoot the specified VLAN or S-VLAN by overriding the currently assigned base profile with one that has debugging attributes enabled.

You can use the **any** keyword to specify a VLAN ID as a wildcard. When you specify the **any** keyword with an S-VLAN ID of a DSLAM, all subscribers associated with the DSLAM will be created with the same profile.

To assign or remove an overriding profile to the dynamic VLAN subinterface:

a. Assign an overriding profile (test1OverridingProfile) to the dynamic VLAN subinterface with VLAN ID 20

   host1(config-if)#**profile vlan override bulk-config vlan 202 test1OverridingProfile**

b. Assign an overriding profile (test1DebugProfile) to the S-VLAN subinterface with S-VLAN ID 100 within the VLAN subinterface with V-LAN ID 202

   host1(config-if)#**profile vlan override bulk-config svlan 100 202 test1OverridingProfile**

c. Remove the overriding profile assignment from the VLAN subinterface with VLAN ID 202, and restores the original base profile assignment

   host1(config-if)#**no profile vlan override bulk-config vlan 202 test1OverridingProfile**

Related Documentation

- Overriding VLAN Base Profile Assignments Overview on page 665
- Monitoring Dynamic VLAN Subinterfaces Created with an Overriding Profile Assignment on page 718

- *profile vlan override bulk-config*

Configuring VLAN as the Encapsulation Method for the Interface

You can use the **encapsulation vlan** command to configure VLAN as the encapsulation method for the interface.

To configure VLAN as the encapsulation method for the interface:

- Issue the **encapsulation vlan** command in Interface Configuration mode.

  host1(config-if)#**encapsulation vlan**

Use the **no** version to disable VLAN on an interface.

Related Documentation

- Dynamic VLAN Subinterfaces over Static VLAN Major Interfaces Overview on page 657
- *encapsulation vlan*

Disabling a Dynamic VLAN subinterface

You can use the **shutdown** command to disable an interface.

When you disable a dynamic VLAN subinterface, the **shutdown** command deletes the VLAN dynamic subinterface column and removes the dynamic VLAN subinterface.

To disable an interface:

- Issue the **shutdown** command in Subinterface Configuration mode.

```
host1(config-subif)#shutdown
```

Because the **shutdown** command removes the dynamic VLAN subinterface from the router, issuing a subsequent **no** version of this command has no effect; that is, it does not restart the disabled subinterface.

Related Documentation

- Configuring Dynamic VLAN Subinterfaces on page 667
- *shutdown*

Changing VLAN Subranges

Changing VLAN subranges within a bulk-configured VLAN range includes the following tasks:

- Adding VLAN Subranges on page 683
- Removing VLAN Subranges on page 684
- Modifying VLAN Subranges on page 685
- Merging VLAN Subranges on page 685
- Changing the Administrative State of VLAN Subranges on page 686

Adding VLAN Subranges

You can add a new VLAN subrange to an existing VLAN range only when the new subrange does not overlap with any existing subrange. Any overlap causes the addition to fail.

You can add multiple subranges to an existing VLAN range simultaneously. However, the entire operation fails if even one of the new subranges overlaps with an existing subrange.

The following example specifies the original VLAN subranges.

```
host1(config-if)#vlan bulk-config test svlan-range 201 250 2 2
  svlan-range 501 550 5 5 svlan-range 301 350 3 3
```

To add subranges to this bulk-configured VLAN range, you can choose either of the following methods. Each method adds a new subrange encompassing S-VLAN IDs 401–450 with VLAN ID 4 to the existing VLAN range, test.

- Specify one new subrange at a time.

  host1(config-if)#**vlan bulk-config test svlan-range 401 450 4 4**

- Specify the new subrange and all the existing subranges. If you use this method, all the existing subranges and their order must match exactly, or the operation fails.

  host1(config-if)#**vlan bulk-config test svlan-range 201 250 2 2 svlan-range 501 550 5 5 svlan-range 301 350 3 3 svlan-range 401 450 4 4**

  The following operation fails because the order of subranges does not match the existing order.

  host1(config-if)#**vlan bulk-config test svlan-range 201 250 2 2 svlan-range 101 150 1 1 svlan-range 501 550 5 5 svlan-range 301 350 3 3 svlan-range 401 450 4 4 svlan-range 601 650 6 6**

You can create a placeholder VLAN range by specifying a VLAN range name without specifying any subrange parameters. This VLAN range has no VLAN ID reservation, but you can assign a profile to it, and add subranges later as desired. The following commands illustrate this approach.

host1(config-if)#**vlan bulk-config test**
host1(config-if)#**profile vlan bulk-config-name test vlanProfile**
host1(config-if)#**vlan bulk-config test svlan-range 401 450 4 4 svlan-range 601 650 6 6**

Removing VLAN Subranges

You can remove VLAN subranges from an existing VLAN range if no dynamic VLAN subinterfaces currently exists for any circuit within those subranges. The removal operation fails if any such dynamic VLAN subinterface exists. You must first remove the dynamic VLAN subinterfaces before you can remove the subranges. Removal of a subrange automatically results in the removal of all overriding profile assignments on that subrange.

You can remove only a single specific VLAN subrange at a time. The following example specifies the original VLAN subranges.

host1(config-if)#**vlan bulk-config test svlan-range 101 150 1 1 svlan-range 201 250 2 2 svlan-range 501 550 5 5 svlan-range 301 350 3 3**

The following command removes one subrange encompassing S-VLAN IDs 101–150 with VLAN ID 1 and leaves the remaining subranges, and the named VLAN range, test, intact.

host1(config-if)#**no vlan bulk-config test svlan-range 101 150 1 1**

The following command removes a subrange that includes S-VLAN IDs 700–750, and that is based on agent-circuit-identifier information from the named VLAN range, test.

host1(config-if)#**no vlan bulk-config test svlan-range 700 750 agent-circuit-identifier**

To remove more than one VLAN subrange, you must issue multiple removal commands, one for each subrange. You cannot remove only part of a subrange. A removal command cannot encompass more than one subrange, even if the subranges are adjacent. However, if you do not specify any subranges, you can remove all subranges in the VLAN, and the named VLAN range, at the same time.

```
host1(config-if)#no vlan bulk-config test
```

Modifying VLAN Subranges

You can use the **vlan bulk-config modify** command to shorten or expand a subrange by modifying the subrange values of a VLAN range. You can expand a subrange if none of the VLAN IDs or S-VLAN IDs added overlap with any other subrange. You can shorten a subrange if none of the VLAN IDs or S-VLAN IDs have existing dynamic VLAN subinterfaces. You can also modify an existing subrange by configuring it to use agent-circuit-identifier information rather than a range of VLAN IDs.

You can modify a subrange so that it completely includes at least one other subrange from within the same VLAN range effectively merges the subranges. Each subrange that is merged with another frees up a subrange for subsequent configuration. The subranges that are merged do not need to be adjacent to each other.

You can modify only a single specific subrange at a time. The following example specifies the original VLAN subranges encompassing S-VLAN IDs 201–250 with VLAN ID 2.

```
host1(config-if)#vlan bulk-config test svlan-range 101 150 1 1
  svlan-range 201 250 2 2 svlan-range 501 550 5 5 svlan-range 301 350 3 3
```

The following command modifies the second subrange from S-VLAN IDs 201–250 with VLAN ID 2 to S-VLAN IDs 210–230 with VLAN IDs 2–3.

```
host1(config-if)#vlan bulk-config test modify svlan-range 210 230 2 3
```

The following command modifies the third subrange from S-VLAN IDs 501–550 with VLAN ID 5 to S-VLAN IDs 501–550 with user identification that is based on agent-circuit-identifier information.

```
host1(config-if)#vlan bulk-config test modify svlan-range 501 550 agent-circuit-identifier
```

The router retains any overriding profiles assigned to a subrange after you modify the subrange if the override assignment still falls within the modified subrange. If the assignment falls outside of the newly modified subrange, the router drops the overriding profile assignment. If two subranges are merged, the router retains overriding profiles that were assigned to the separate subranges and applies the overriding profiles to the newly merged subrange.

You cannot modify a subrange at the same time you are adding or removing a subrange. If the new modified values for a subrange partially overlap with another subrange, the operation fails and the router displays an error message.

Merging VLAN Subranges

You can merge multiple subranges of any particular VLAN range to form a single unified subrange, conserving subrange resources. Merging takes place only when you modify a subrange so that it completely includes at least one other subrange of the same VLAN range. The merged subranges do not need to be adjacent to each other.

If the encompassing subrange has any VLAN IDs or S-VLAN IDs that are outside the subranges to be merged, those VLAN IDs or S-VLAN IDs are added. The encompassing subrange must cover a subrange completely to incorporate it in the merged subrange. The merge operation fails if the encompassing subrange completely overlaps some subranges but only partially overlaps with another subrange. The encompassing subrange does not have to encompass all subranges of the VLAN range.

Each subrange that is merged with another frees up a subrange. E Series routers currently support a maximum of 300 bulk-configured VLAN ranges per chassis. Therefore, if a VLAN range consists of 5 subranges, 295 subranges are still available for subsequent configuration. If you merge 2 of those subranges, resulting in a new total of 4 subranges in the VLAN range, then 296 subranges are available for configuration.

The router retains any overriding profile assignments on the subranges made before the merger, and applies them to the new merged subrange. You can separate merged subranges either by removing the merged subrange and then adding new separate subranges or by modifying the merged subrange to remove some portion of the subrange and then adding a new subrange.

The following example specifies the original VLAN subranges.

```
host1(config-if)#vlan bulk-config test svlan-range 101 150 1 1
   svlan-range 201 250 2 2 svlan-range 501 550 5 5 svlan-range 301 350 3 3
```

The following command merges two subranges (S-VLAN IDs 101–150 and VLAN ID 1) and (S-VLAN IDs 201–250 and VLAN ID 2) and effectively replaces them with the new subrange encompassing S-VLAN IDs 101–250 and VLAN IDs 1–2.

```
host1(config-if)#vlan bulk-config test modify svlan-range 101 250 1 2
```

To separate the merged subranges, you can modify the unified subrange and add subranges as needed, provided that no dynamic VLAN subinterfaces currently exist for any VLAN ID within those subranges.

If you merge subranges by using SNMP, the new merged subrange takes the lowest instance value of the incorporated subranges. For example, if a VLAN range has three subranges with instance values of 2, 4, and 5 and the subranges with instance values of 2 and 5 are merged, the new merged subrange has an instance value of 2.

Changing the Administrative State of VLAN Subranges

VLAN subranges have an administrative state that enables you to remove dynamic VLAN subinterfaces on various subranges that belong to a single VLAN range. This functionality is important because subrange removal requires that no dynamic VLAN subinterfaces exist for any circuit on that subrange. The removal operation fails if any such interfaces exist.

By default, the administrative state of a VLAN subrange is up. When you change the administrative state to down by using the **vlan bulk-config shutdown** command, the router deletes all dynamic VLAN subinterfaces on the affected subranges. You can use the **show vlan subinterface** command to monitor the progress of the removal of all dynamic VLAN subinterfaces for the specified subrange.

No additional dynamic VLAN subinterfaces can be created for the subrange until you restore the administrative state to up by using the **no vlan bulk-config shutdown** command.

The following example specifies the original VLAN subranges.

host1(config-if)#**vlan bulk-config test svlan-range 101 150 1 1 svlan-range 201 250 2 2 svlan-range 501 550 5 5 svlan-range 301 350 3 3**

You cannot specify a partial subrange; the specified subrange must exactly match a subrange that has already been configured. The following command changes the administrative state of the second subrange (S-VLAN IDs 201–250 and VLAN ID 2) to down. The router removes all dynamic interface columns built on any of the VLAN IDs or S-VLAN IDs in this subrange. No additional dynamic VLAN subinterfaces can be created until you change the administrative state to up.

host1(config-if)#**vlan bulk-config test shutdown svlan-range 201 250 2 2**

The following command changes the administrative state of this same VLAN subrange to up.

host1(config-if)#**no vlan bulk-config test shutdown svlan-range 201 250 2 2**

You can also change the administrative state of VLAN subranges that are based on agent-circuit-identifier information. For example, assume that the following command is issued to configure a VLAN subrange based on agent-circuit-identifier information:

host1(config-if)#**vlan bulk-config myNewBulkConfig svlan-range 50 100 agent-circuit-identifier**

The following command changes the administrative state of this same VLAN subrange to down:

host1(config-if)#**vlan bulk-config myNewBulkConfig shutdown svlan-range 50 100 agent-circuit-identifier**

You can change the administrative state of all subranges in a bulk-configured VLAN range at the same time by issuing the command without specifying any subranges. When you shut down a named bulk configuration, all VLAN ranges belonging to that bulk configuration, including those based on double-tagged S-VLANs or agent-circuit-identifier information, are disabled.

The following command shuts down all four subranges belonging to the named VLAN range, test, regardless of their current state.

host1(config-if)#**vlan bulk-config test shutdown**

The time required for the router to complete an administrative state change depends on the number of VLAN subranges configured.

Related Documentation

- Bulk Configuration of VLAN Ranges Overview on page 662
- Monitoring Base Profile Assignments and Overriding Profile Assignments for VLAN Major Interface on page 716
- *profile vlan bulk-config*

- *vlan bulk-config*
- *vlan bulk-config modify*
- *vlan bulk-config shutdown*

Configuring Static VLAN Subinterfaces Within VLAN Subranges

The example procedures in this section show how to configure static VLAN subinterfaces within VLAN subranges by using the same loopback interface referenced by multiple unnumbered IP interfaces. Instead of assigning a different IP address to each physical interface, the first example assigns an IP address to a loopback interface (loopback 0). Each physical interface is then configured as an unnumbered IP interface, referencing the same loopback interface.

The following topics describe how to configure static VLAN subinterfaces within VLAN subranges:

- Creating a Static VLAN Subinterface Within a VLAN Subrange on page 688
- Creating a VLAN Subrange that Includes a Static VLAN Subinterface on page 689

Creating a Static VLAN Subinterface Within a VLAN Subrange

You can configure a static VLAN subinterface with a VLAN whose VLAN ID falls within an existing bulk-configured VLAN subrange.

To create a static VLAN subinterface within a VLAN subrange:

1. Create the VLAN major interface. For more information, see "Configuring VLAN as the Encapsulation Method for the Interface" on page 682.

   host1(config)#interface gigabitEthernet 0/0
   host1(config-if)#encapsulation vlan

2. Create a bulk-configured VLAN range that includes one or more VLAN subranges. For more information, see "Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface" on page 674.

   host1(config-if)#vlan bulk-config test vlan-range 200 250

3. Create a static VLAN subinterface by adding a subinterface number to the interface identification command. For more information, see *Selecting a Gigabit Ethernet Interface or a 10-Gigabit Ethernet Interface*.

   host1(config-if)#interface gigabitEthernet 0/0.2100

4. Do one of the following:

   - Assign a VLAN ID for the subinterface.

     host1(config-if)#vlan id 201

   - Assign a VLAN ID and the optional unique MAC address for the subinterface.

     host1(config-if)#vlan id 201 mac-address 0090.1a01.1234

5. To fully configure the VLAN subinterface, assign an IP address, or make it unnumbered. For more information, see *Setting Up an Unnumbered Interface*.

   host1(config-if)#ip unnumbered loopback 0

Creating a VLAN Subrange that Includes a Static VLAN Subinterface

You can configure a bulk-configured VLAN subrange that includes the VLAN ID belonging to an existing VLAN on a static VLAN subinterface.

To create a VLAN subrange that includes a static VLAN subinterface:

1. Create the VLAN major interface. For more information, see "Configuring VLAN as the Encapsulation Method for the Interface" on page 682.

   host1(config)#interface gigabitEthernet 3/1
   host1(config-if)#encapsulation vlan

2. Specify a static VLAN subinterface. For more information, see *Selecting a Gigabit Ethernet Interface or a 10-Gigabit Ethernet Interface*.

   host1(config-if)#interface gigabitEthernet 3/1.201

3. Do one of the following:

   - Assign a VLAN ID for the subinterface.

     host1(config-if)#vlan id 201

   - Assign a VLAN ID and the optional unique MAC address for the subinterface.

     host1(config-if)#vlan id 201 mac-address 0090.1a01.1234

4. Create a bulk-configured VLAN range that includes the VLAN ID of the previously configured VLAN. In this example, the VLAN range 100–250 includes VLAN ID 201. For more information, see "Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface" on page 674.

   host1(config)#interface gigabitEthernet 3/1
   host1(config-if)#vlan bulk-config test2 vlan-range 100 250

5. To fully configure the VLAN subinterface, assign an IP address or make it unnumbered. For more information, see *Setting Up an Unnumbered Interface*.

   host1(config-if)#ip unnumbered loopback 0

Related Documentation

- Static VLAN Subinterfaces Within VLAN Subranges Overview on page 666
- Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface on page 674
- Monitoring Base Profile Assignments and Overriding Profile Assignments for VLAN Major Interface on page 716

- *encapsulation vlan*
- *interface gigabitEthernet*
- *interface tenGigabitEthernet*
- *ip unnumbered*

- *vlan bulk-config*

CHAPTER 22

Monitoring Dynamic Interfaces and Profiles

You can use the **show** commands described in this chapter to monitor configurations created with dynamic interfaces and profiles.

NOTE: The E120 and E320 routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 routers output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

- Monitoring Configuration Information of an ATM AAL5 Interface on page 691
- Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface on page 693
- Monitoring Status or Summary Information for ATM Subinterfaces on page 696
- Monitoring Summary Information for ATM VCs and Reserved VC Ranges on page 702
- Monitoring Total Static and Dynamic Interface Counts for Interface Columns on page 704
- Monitoring Summary Information About the Encapsulation Type Lockout for PPPoE Clients on page 705
- Monitoring Detailed Information About the Current Encapsulation Type Lockout Condition for PPPoE Clients on page 706
- Monitoring the Source MAC Address of a PPPoE Client on page 707
- Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces on page 708
- Monitoring Base Profile Assignments and Overriding Profile Assignments for VLAN Major Interface on page 716
- Monitoring Dynamic VLAN Subinterfaces Created with an Overriding Profile Assignment on page 718
- Monitoring Status and Configuration Information for VLAN Subinterfaces on page 719

Monitoring Configuration Information of an ATM AAL5 Interface

Purpose Display information about a configured ATM AAL5 interface.

Action To display information about a configured ATM AAL5 interface:

```
host1#show atm aal5 interface atm 3/0
AAL5 Interface ATM 3/0 operational status:    lowerLayerDown
      time since last status change: 00:08:46

SNMP trap link-status: disabled
Auto configure ATM 1483 status: disabled

InPackets:        0
InBytes:          0
OutPackets:       0
OutBytes:         0
InErrors:         0
OutErrors:        0
InPacketDiscards: 0
OutDiscards:      0
```

Meaning Table 50 on page 594 lists the **show atm aal5 interface** command output fields.

Table 59: show atm aal5 interface Output Fields

Field Name	Field Description
AAL5 Interface operational status	Operational status of the AAL5 interface: up, down, lowerLayerDown
time since last status change	Time since last reported change to the AAL5 interface operational status in hh:mm:ss format
SNMP trap link-status	Whether SNMP link status traps are enabled or disabled on the ATM AAL5 interface
Auto configure ATM 1483 status	Whether the autoconfiguration feature for a dynamic ATM 1483 subinterface configured over the ATM AAL5 interface is enabled or disabled
InPackets	Number of packets received on this interface
InBytes	Number of bytes received on this interface
OutPackets	Number of packets transmitted on this interface
OutBytes	Number of bytes transmitted on this interface
InErrors	Number of incoming errors received on this interface
OutErrors	Number of outgoing errors on this interface
InPacketDiscards	Number of incoming packets discarded on this interface
OutDiscards	Number of outgoing packets discarded on this interface

Related Documentation

- Creating a PVC on an ATM 1483 Subinterface on page 537
- *show atm aal5 interface*

Monitoring Bulk-Configured VC Ranges on an ATM AAL5 Interface

Purpose Display information, including base profile assignments and overriding profile assignments, for the bulk-configured VC ranges on an ATM AAL5 interface. You can display information for all VC ranges on the router by using the **show atm bulk-config** command with no keywords.

To display information for all VC ranges on a specified ATM AAL5 interface, use the **show atm bulk-config** command with the **atm** keyword and interface specifier.

For information about a particular VC range on a specified ATM AAL5 interface, use the **show atm bulk-config** command with the **atm** keyword and interface specifier and the **name** keyword.

You can display information only about overriding profile assignments configured for specific ATM PVCs within bulk-configured VC subranges by using the **show atm bulk-config** command with the **override** keyword. When you specify the **override** keyword, the command does not display information about base profile assignments.

Action To display information about base profile assignments and overriding profile assignments for all bulk-configured VC ranges on the router, use the **show atm bulk-config** command with no keywords. The VC range named test consists of a single VC subrange (1, 1, 101, 200), has a base profile named atm1483BaseProfile assigned, and has an overriding profile named overrideProfile1 assigned to two ATM PVCs within the VC subrange. The VC range named test2 is a placeholder range that has no VC subranges configured and no base profile assigned:

```
host1#show atm bulk-config

              Bulk
             Config Start End Start End                    Admin
 Interface    Name   VPI  VPI  VCI  VCI  Assigned Profile  State
------------ ------ ----- --- ----- --- ------------------ -----
ATM AAL5 3/0   test     1   1   101 200 atm1483BaseProfile    up
ATM AAL5 3/2  test2    --  --   --- ---      none assigned   ---

2 bulk configuration(s) found

Profile override(s):

              Bulk
             Config
 Interface    Name  VPI VCI Assigned Profile Status
------------ ------ --- --- ---------------- ------
ATM AAL5 3/0   test   1 151 overrideProfile1 Active
ATM AAL5 3/0   test   1 161 overrideProfile1 Active
2 profile override(s) found
```

To display information about base profile assignments and overriding profile assignments for all VC ranges configured on a specified ATM AAL5 interface:

host1#**show atm bulk-config atm 3/0**

```
              Bulk
             Config Start End Start End                    Admin
 Interface    Name   VPI  VPI  VCI  VCI  Assigned Profile  State
------------ ------ ----- --- ----- --- ------------------ -----
ATM AAL5 3/0   test     1   1   101 200 atm1483BaseProfile    up
1 bulk configuration(s) found
```

```
Profile override(s):
```

```
              Bulk
             Config
 Interface    Name  VPI VCI Assigned Profile Status
------------ ------ --- --- ---------------- ------
ATM AAL5 3/0   test   1 151 overrideProfile1 Active
ATM AAL5 3/0   test   1 161 overrideProfile1 Active
2 profile override(s) found
```

To display information about base profile assignments and overriding profile assignments for a particular bulk-configured VC range:

host1#**show atm bulk-config name test**

```
              Bulk
             Config Start End Start End                    Admin
 Interface    Name   VPI  VPI  VCI  VCI  Assigned Profile  State
------------ ------ ----- --- ----- --- ------------------ -----
ATM AAL5 3/0   test     1   1   101 200 atm1483BaseProfile    up
1 bulk configuration(s) found
```

```
Profile override(s):
```

```
              Bulk
             Config
 Interface    Name  VPI VCI Assigned Profile Status
------------ ------ --- --- ---------------- ------
ATM AAL5 3/0   test   1 151 overrideProfile1 Active
ATM AAL5 3/0   test   1 161 overrideProfile1 Active
2 profile override(s) found
```

To display information only about overriding profile assignments for all bulk-configured VC ranges on the router:

host1#**show atm bulk-config override**

```
Profile override(s):
```

```
              Bulk
             Config
 Interface    Name  VPI VCI Assigned Profile Status
------------ ------ --- --- ---------------- ------
ATM AAL5 3/0   test   1 151 overrideProfile1 Active
ATM AAL5 3/0   test   1 161 overrideProfile1 Active
2 profile override(s) found
```

To display information only about overriding profile assignments for a particular VC range configured on a specified ATM AAL5 interface:

host1#**show atm bulk-config atm 3/0 override**

```
Profile override(s):
```



```
                Bulk
               Config
 Interface      Name   VPI VCI Assigned Profile Status
------------ ------ --- --- ---------------- ------
ATM AAL5 3/0   test   1 151 overrideProfile1 Active
ATM AAL5 3/0   test   1 161 overrideProfile1 Active
2 profile override(s) found
```

Meaning Table 60 on page 695 lists the output fields for the **show atm bulk-config** command.

Table 60: show atm bulk-config Output Fields

Field Name	Field Description
Interface	Identifier of the ATM AAL5 physical interface on which the bulk-configured VC range resides. For more information about specifying the ATM interface, see *Interface Types and Specifiers* in the *JunosE Command Reference Guide*.
Bulk Config Name	Name of the bulk-configured VC range on this interface
Start VPI	Starting virtual path identifier (inclusive) of the VC subrange
End VPI	Ending virtual path identifier (inclusive) of the VC subrange
Start VCI	Starting virtual circuit identifier (inclusive) of the VC subrange
End VCI	Ending virtual circuit identifier (inclusive) of the VC subrange
Assigned Profile	Base profile name for the dynamic ATM 1483 subinterface assigned to this VC subrange with "Assigning an Overriding Profile to a Single ATM PVC that Exists Within a Bulk-Configured VC Subrange" on page 650. When no profile is assigned to the VC subrange, the field displays none assigned.
Admin State	Administrative state of the VC subrange: up or down

Table 60: show atm bulk-config Output Fields *(continued)*

Field Name	Field Description
Profile override(s)	When overriding profile assignments are configured on the router, the command displays the following fields: • Interface—Identifier of the ATM AAL5 physical interface • Bulk Config Name—Name of the bulk-configured VC range on this interface that includes the VC subrange encompassing the specified ATM PVC • VPI—Virtual path identifier of the PVC within the bulk-configured VC subrange • VCI—Virtual circuit identifier of the PVC within the bulk-configured VC subrange • Assigned Profile—Name of the overriding profile assigned to the specified PVC with "Assigning an Overriding Profile to a Single ATM PVC that Exists Within a Bulk-Configured VC Subrange" on page 650 • Status—Operational status of the overriding profile assignment: Active or Inactive. Active indicates that the router uses the overriding profile to create dynamic interface columns because no static ATM circuits with the same VPI/VCI values exist on this interface. Inactive indicates that the router does not use the overriding profile to create dynamic interface columns because a static ATM circuit with the same VPI/VCI values exists on this interface.

Related Documentation

- Creating a Bulk-Configured VC Range on a Static ATM AAL5 Interface on page 644

- *show atm bulk-config*

Monitoring Status or Summary Information for ATM Subinterfaces

Purpose Display the current state of all ATM subinterfaces, all ATM subinterfaces configured on a specified ATM physical interface, or a specific ATM subinterface. You can use the **summary** keyword to display brief summary information for all ATM subinterfaces, or for ATM subinterfaces configured on a specified ATM physical interface. You can use the **status** keyword with one of the following operating status values to display status information only for ATM subinterfaces with a specific operating status:

- dormant
- dormantLockout
- down
- lowerLayerDown
- notPresent
- up

You can use the **atm** *slot/port/vpi/vci* format (for ERX7xx models, ERX14xx models, and ERX310 router) or the *slot/adapter/port/vpi/vci* format (for E120 and E320 routers) to display the current state of an ATM subinterface created on the PVC with the specified VPI and VCI values.

NOTE: You can use the **atm** *slot/port/vpi/vci* format as an alternative to the **atm** *slot/port.subinterface* format with the specific **show interface** and **show subinterface** commands to monitor all ATM 1483 subinterfaces (except NBMA interfaces) as well as the upper-layer interfaces configured over an ATM 1483 subinterface. You cannot, however, use the **atm** *slot/port/vpi/vci* format to create or modify an ATM 1483 subinterface.

These guidelines also apply to E120 and E320 routers when you use the **atm** *slot/adapter/port/vpi/vci* format as an alternative to the **atm** *slot/adapter/port.subinterface* format.

For more information, see “Creating a Basic Configuration” on page 21 in “Configuring ATM” on page 3.

Action To display the current state of all ATM subinterfaces:

```
host1#show atm subinterface
                                Circuit                         Interface
Interface   ATM-Prot VCD VPI VCI Type  Encap MTU    Status      Type
----------- -------- --- --- --- ----- ----- ---- ----------- -------
ATM 2/0.101 RFC-1483 101   0 101 PVC   AUTO  9180 dormantLockout Static
ATM 2/0.102 RFC-1483 102   0 102 PVC   AUTO  9180 up             Dynamic
ATM 2/0.103 RFC-1483 103   0 103 PVC   AUTO  9180 dormant        Static
3 interface(s) found
```

To display summary information for all ATM subinterfaces:

```
host1#show atm subinterface summary
3 subinterfaces: 1 up, 0 down,
1 dormant, 1 dormantLockout,
0 lowerLayerDown, 0 not present
```

To display status information for all ATM subinterfaces in the dormantLockout state:

```
host1#show atm subinterface status dormantLockout
                                Circuit                        Interface
 Interface  ATM-Prot VCD VPI VCI  Type Encap MTU  Status        Type
---------- -------- --- --- --- ------ ----- ---- ------------- ---------
ATM 2/0.101 RFC-1483 101   0 101 PVC   AUTO  9180 dormantLockout Static
1 interface(s) found
```

To display the current state of a specific ATM subinterface:

```
host1#show atm subinterface atm 2/0.101
                                Circuit                        Interface
 Interface  ATM-Prot VCD VPI VCI Type Encap MTU    Status      Type
----------- -------- --- --- --- ----- ----- ---- ------------ ---------
ATM 2/0.101 RFC-1483 101  0 101 PVC   AUTO  9180 dormantLockout Static
```



```
Auto configure status         : dynamic
Auto configure interface(s)   : IP  bridgedEthernet  PPP  PPPoE
Detected 1483 encapsulation   : AUTO
Detected dynamic interface    : none
Interface types in lockout    : IP

Lockout state (seconds)       : Min Max  Current Elapsed Next
------------------------------ --- ---- ------- ------- ----
IP                              1   30      16       7   30
BridgedEnet                   900 3600       0       0  900
PPP                             1  300       0       0    1
PPPoE                           1  300       0       0    1

Assigned profile (IP)         : ipoa
Assigned profile (BridgedEnet): beth
Assigned profile (PPP)        : ppptest
Assigned profile (PPPoE)      : pppoetest
Assigned profile (any)        : none assigned

BridgedEnet subscriber info   :
Username: elaine@jpeterman.com
Password: putty
Authenticate: enabled

SNMP trap link-status: disabled

InPackets:               0
InBytes:                 1904
OutPackets:              0
OutBytes:                0
InErrors:                0
OutErrors:               0
InPacketDiscards:        14
InPacketsUnknownProtocol: 0
OutDiscards:             0
1 interface(s) found
```

To display the current state of a specific ATM subinterface created on the PVC with the specified VPI and VCI values:

```
host1#show atm subinterface atm 0/0/0/101
                                 Circuit                   Interface
 Interface   ATM-Prot VCD VPI VCI  Type   Encap MTU  Status   Type
----------- -------- --- --- --- ------- ----- ---- ------ ---------
ATM 0/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 up     Static

Auto configure status         : dynamic
Auto configure interface(s)   : PPPoE
Detected 1483 encapsulation   : SNAP
Detected dynamic interface    : PPPoE
Interface types in lockout    : none

Lockout state (seconds)       : Min Max Current Elapsed Next
------------------------------ --- --- ------- ------- ----
PPPoE                           1 300       0       0    1

Assigned profile (IP)         : none assigned
Assigned profile (BridgedEnet): none assigned
Assigned profile (PPP)        : none assigned
Assigned profile (PPPoE)      : pppoeprofile
Assigned profile (any)        : none assigned

SNMP trap link-status: disabled
```



```
InPackets:                 5119
InBytes:                   358672
OutPackets:                5107
OutBytes:                  357510
InErrors:                  0
OutErrors:                 0
InPacketDiscards:          3
InPacketsUnknownProtocol:  0
OutDiscards:               0
1 interface(s) found
```

Meaning Table 51 on page 597 lists the **show atm subinterface** command output fields.

Table 61: show atm subinterface Output Fields

Field Name	Field Description
Interface	Interface identifier
ATM-Prot	One of the following ATM protocol types: • RFC-1483—Multiprotocol encapsulation over AAL5 • NBMA—Nonbroadcast multiaccess interface • ATM/MPLS—Local ATM passthrough interface
VCD	Virtual circuit descriptor
VPI	Virtual path identifier
VCI	Virtual circuit (or channel) identifier
Circuit Type	Type of circuit: PVC
Encap	Administered encapsulation method based on what was configured with the **atm pvc** command
MTU	Maximum transmission unit size for the interface

Table 61: show atm subinterface Output Fields *(continued)*

Field Name	Field Description
Status	One of the following ATM 1483 subinterface states: • absent—Represents the notPresent state and indicates that, although the SRP detects the ATM 1483 subinterface, the module on which the subinterface resides has not completed booting up, has failed, or is disabled. • dormant—Indicates that the ATM 1483 subinterface is performing autodetection of one or more upper-layer encapsulation types and is waiting to receive a packet of that type on a lower-layer interface. An ATM 1483 subinterface transitions from the dormant state to the up state when the router receives a valid packet of the specified encapsulation type on the interface. • dormantLockout—Indicates that a dormant ATM 1483 subinterface has one or more upper-layer encapsulation types currently undergoing encapsulation type lockout. An ATM 1483 subinterface transitions from the dormantLockout state to the dormant state when the lockout time expires for all upper-layer encapsulation types undergoing lockout. An ATM 1483 subinterface transitions from the dormantLockout state to the up state when the router receives a valid packet for an encapsulation type that is configured for autodetection but is not undergoing lockout. • down—Indicates that the ATM 1483 subinterface is administratively disabled or has a circuit that is down or not configured. • lowerLayerDown—Indicates that a lower-layer interface below the ATM 1483 subinterface is down. • up—Indicates that the ATM 1483 subinterface is up and able to transfer data. For an ATM 1483 subinterface that supports one or more dynamic upper-layer interfaces, indicates that the router has created the dynamic upper-layer interface or is in the process of creating it.
Interface Type	Type of ATM 1483 subinterface: dynamic or static
Auto configure status	Setting of the autoconfiguration feature: • dynamic—Autodetection is on; the router automatically detects the next upper interface • static—Autodetection is off
Auto configure interface(s)	Types of dynamic upper interfaces configured with the **auto-configure** command: bridged Ethernet, IP, PPP, or PPPoE

Table 61: show atm subinterface Output Fields *(continued)*

Field Name	Field Description
Detected 1483 encapsulation	If the encapsulation type is set to **aal5autoconfig**, displays the 1483 encapsulation type detected on the subinterface (displays AUTO until a packet is detected)
Detected dynamic interface	Type of dynamic upper interface detected during autoconfiguration: bridged Ethernet, IP, PPP, PPPoE, or (if no packet has been received) none
Interface types in lockout	Encapsulation types currently experiencing lockout: bridged Ethernet, IP, PPP, PPPoE, or none
Lockout state (seconds)	Settings of encapsulation type lockout for the upper-layer encapsulation type indicated • Min—Minimum lockout time, in seconds • Max—Maximum lockout time, in seconds • Current—Current lockout time, in seconds; displays 0 (zero) if lockout is not occurring • Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if lockout is not occurring • Next—Lockout time for the router to use for the next lockout event, in seconds
Assigned profile	For each dynamic interface type, indicates whether or not a profile is assigned and, if assigned, displays the profile name
Subscriber info	Subscriber login information for the specified dynamic interface type (bridged Ethernet or IP)
SNMP trap link-status	Trap link status: enabled or disabled
InPackets	Number of packets received on this interface
InBytes	Number of bytes received on this interface
OutPackets	Number of packets transmitted on this interface
OutBytes	Number of bytes transmitted on this interface
InErrors	Number of errors received on this interface
OutErrors	Number of outgoing errors on this interface
InPacketDiscards	Number of incoming packets discarded on this interface
InPacketsUnknownProtocol	Number of incoming packets with an unknown protocol type

Table 61: show atm subinterface Output Fields *(continued)*

Field Name	Field Description
OutDiscards	Number of outgoing packets discarded on this interface

Related Documentation

- Configuring a Dynamic Interface over an ATM 1483 Subinterface on page 530
- Creating a PVC on an ATM 1483 Subinterface on page 537
- Configuring Dynamic PPPoE over Static PPPoE with ATM Interface Columns on page 543
- Configuring a Dynamic IPoA Interface on page 556
- Configuring a Dynamic Bridged Ethernet Interface on page 559
- Assigning a Profile to a Dynamic Interface on page 572

- *atm pvc*
- *auto-configure*
- *show atm subinterface*

Monitoring Summary Information for ATM VCs and Reserved VC Ranges

Purpose Display a summary of all configured ATM VCs and reserved VC ranges. You can specify the **reserved** keyword with no other keywords to display only a summary of all reserved VC ranges on the router. This includes VPI/VCI ranges reserved for use by dynamic ATM 1483 subinterfaces.

Action To display all VCs and reserved VC ranges on the router:

```
host1#show atm vc
                                      Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
ATM 3/0.3      0 102  4376 PVC  AUTO  CBR   1000     0     0 DOWN
...
ATM 3/0.8099   1 8099 8099 PVC  SNAP  UBR      0     0     0 UP
ATM 3/0.8100   1 8100 8100 PVC  SNAP  UBR      0     0     0 DOWN
8000 circuit(s) found

Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

To display all reserved VC ranges on the router:

```
host1#show atm vc reserved
Reserved VCC ranges:
```



```
          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

Meaning Table 52 on page 601 lists the **show atm vc** command output fields.

Table 62: show atm vc Output Fields

Field Name	Field Description
Interface	Interface identifier
VPI	Virtual path identifier
VCI	Virtual channel identifier
VCD	Virtual circuit descriptor
Type	Type of circuit: PVC
Encap	Encapsulation method: AUTO, AAL5, MUX, SNAP, ILMI, F4-OAM
Category	Service type configured on the VC: UBR, UBR-PCR, NRT-VBR, RT-VBR, or CBR
Rx/Tx Peak	Peak rate, in Kbps
Rx/Tx Avg	Average rate, in Kbps
Rx/Tx Burst	Maximum number of cells that can be burst at the peak cell rate
Status	State of the virtual circuit: Up or Down
Start VPI	Starting virtual path identifier (inclusive) of the reserved VC range
Start VCI	Starting virtual circuit identifier (inclusive) of the reserved VC range
End VPI	Ending virtual path identifier (inclusive) of the reserved VC range
End VCI	Ending virtual circuit identifier (inclusive) of the reserved VC range

Related Documentation

- Creating a PVC on an ATM 1483 Subinterface on page 537

- Configuring a PPP or PPPoE Dynamic Interface over an ATM 1483 Subinterface on page 539
- Configuring Dynamic PPPoE over Static PPPoE with ATM Interface Columns on page 543
- Configuring and Verifying Lockout for PPPoE Clients on page 552
- Configuring a Dynamic IPoA Interface on page 556
- Dynamic Bridged Ethernet Interfaces on page 558
- *show atm vc*

Monitoring Total Static and Dynamic Interface Counts for Interface Columns

Purpose Display static and dynamic interface counts for each interface column. Counts for PPP and PPPoE interface columns are updated when the PPP layer comes up. Counts for bridged Ethernet and IP over ATM columns are updated when the ATM layer comes up.

Action To display static and dynamic interface counts for each interface column:

```
host#show columns
                  Interface columns:
                  ------------------
        Type          Total      Static    Dynamic
-------------------- ---------- ---------- ----------
Bridged Ethernet              4          2          2
IP over ATM                   4          2          2
PPP                          22         12         10
PPPoE                        10          5          5
```

Meaning Table 53 on page 602 lists the **show columns** command output fields.

Table 63: show columns Output Fields

Field Name	Field Description
Type	Interface type
Total	Total number of interfaces on this column
Static	Number of static interfaces on this column
Dynamic	Number of dynamic interfaces on this column

Related Documentation

- Configuring a Dynamic Interface over an ATM 1483 Subinterface on page 530
- Dynamic PPP and PPPoE Interfaces over Static ATM on page 538
- Dynamic PPPoE Interfaces over PPPoE Static Interfaces on page 542
- Dynamic Bridged Ethernet Interfaces on page 558
- *show columns*

Monitoring Summary Information About the Encapsulation Type Lockout for PPPoE Clients

Purpose Display summary information about the encapsulation type lockout parameters configured for PPPoE clients on a dynamic PPPoE subinterface column.

Action To display summary information about the encapsulation type lockout parameters configured for PPPoE clients on a dynamic PPPoE subinterface column:

```
host1#show pppoe interface atm 3/0.101
. . .
Lockout Configuration (seconds): Min 5, Max 60
 Total clients in active lockouts: 0
 Total clients in lockout grace period: 0
```

NOTE: The output includes only the portion of the **show pppoe interface** command display relevant to lockout configuration for PPPoE clients.

Meaning Table 54 on page 603 lists the **show pppoe interface** command encapsulation type lockout output fields.

Table 64: show pppoe interface Encapsulation Type Lockout Output Fields

Field Name	Field Description
Lockout Configuration (seconds)	Encapsulation type lockout settings for the PPPoE client associated with the dynamic PPPoE subinterface column: • Min—Minimum lockout time, in seconds • Max—Maximum lockout time, in seconds • Total clients in active lockouts—Number of PPPoE clients currently undergoing dynamic encapsulation type lockout • Total clients in lockout grace period—Number of PPPoE clients currently in a lockout grace period; for more information about the lockout grace period, see “Guidelines for Configuring Encapsulation Type Lockout for PPPoE Sessions” on page 535

Related Documentation

- Configuring and Verifying Lockout for PPPoE Clients on page 552
- *show pppoe interface*

Monitoring Detailed Information About the Current Encapsulation Type Lockout Condition for PPPoE Clients

Purpose Display detailed information about the current encapsulation type lockout condition for each PPPoE client associated with a dynamic PPPoE subinterface column on a static PPPoE major interface.

Action To display detailed information about the current encapsulation type lockout condition for each PPPoE client associated with a dynamic PPPoE subinterface column on a static PPPoE major interface:

```
host1#show pppoe interface atm 12/1/1.1.1 lockout-time
PPPoE interface atm 12/1/1.1
Lockout Configuration (seconds): Min 90, Max 120
 Total clients in active lockout: 1
 Total clients in lockout grace period: 0
Client Address Current Elapsed Next
-------------- ------- ------- ----
0090.1a42.527c     120      30  120
0090.1a42.527c       0       0   90
```

Meaning Table 55 on page 604 lists the **show pppoe interface lockout-time** command output fields.

Table 65: show pppoe interface lockout-time Output Fields

Field Name	Field Description
PPPoE interface	Specifier for the PPPoE interface
Lockout Configuration (seconds)	Encapsulation type lockout settings for the PPPoE client associated with the dynamic PPPoE subinterface column: • Min—Minimum lockout time, in seconds • Max—Maximum lockout time, in seconds • Total clients in active lockouts—Number of PPPoE clients currently undergoing dynamic encapsulation type lockout • Total clients in lockout grace period—Number of PPPoE clients currently in a lockout grace period; for more information about the lockout grace period, see “Guidelines for Configuring Encapsulation Type Lockout for PPPoE Sessions” on page 535
Client Address	Source MAC address of the PPPoE client NOTE: Because PPPoE sessions that contain the IWF-Session DSL Forum VSA (26-154) use the same source MAC address of the DSLAM for all subscriber connections, multiple entries are displayed in the Client Address field for the same MAC address if multiple IWF sessions contain the same MAC address.

Table 65: show pppoe interface lockout-time Output Fields *(continued)*

Field Name	Field Description
Current	Current lockout time, in seconds; displays 0 (zero) if the PPPoE client is not undergoing lockout
Elapsed	Time elapsed into the lockout time, in seconds; displays 0 (zero) if the PPPoE client is not undergoing lockout
Next	Lockout time that the router uses for the next lockout event, in seconds

Related Documentation

- Configuring and Verifying Lockout for PPPoE Clients on page 552
- Clearing the Lockout Condition for a PPPoE Client on page 553
- *show pppoe interface lockout-time*

Monitoring the Source MAC Address of a PPPoE Client

Purpose Display the source MAC address of a PPPoE client when a subscriber is connected to the router through an available PPPoE session. You can use the **full** keyword to display configuration, status, and statistics information, including the source MAC address of the PPPoE client.

Action To display the source MAC address of a PPPoE client when a subscriber is connected to the router through an available PPPoE session:

```
host1#show pppoe subinterface full
...
    PPPoE subinterface ATM 3/0.101 has source MAC address 0090.1a10.165e
...
```

NOTE: The output includes only the portion of the **show pppoe subinterface** command display relevant to the source MAC address for PPPoE clients.

Meaning Table 56 on page 606 lists the **show pppoe subinterface** command source MAC address output fields.

Table 66: show pppoe subinterface Source MAC Address Output Fields

Field Name	Field Description
PPPoE subinterface	Specifier for the PPPoE subinterface
source MAC address	MAC address of the PPPoE client associated with the dynamic PPPoE subinterface column

Related Documentation

- Clearing the Lockout Condition for a PPPoE Client on page 553
- *pppoe clear lockout interface*
- *show pppoe subinterface*

Monitoring the Characteristics of a Profile Assigned to Dynamic Interfaces

Purpose Display information about profiles. You can use the **name** keyword to display information about a specific profile. You can use the **brief** keyword to display a list of profiles configured on the router.

Action To display configuration information for a profile assigned to a dynamic interface:

```
host1#show profile name pppoe
PPP Keepalive                : 30
PPP Magic Number             : enabled
PPP Magic Number Mismatch    :reject
PPP Peer DNS Priority        : disabled
PPP Peer WINS Priority       : disabled
PPP Authentication           :
PPP Authentication Router    :
PPP Negotiate MRU            : (use lower layer MRU)
PPP Packet Log               : disabled
PPP State Log                : disabled
PPP Chap Challenge Length    : 16 - 32
PPP Passive Mode             : disabled
PPP Multilink                : disabled
PPP IPCP Netmask Option      : disabled
PPP IPCP Lockout Option      : disabled
PPP IPCP Lockout             : disabled
PPP AAA Profile              :
PPP Multilink Fragmentation : disabled
PPP Multilink Fragment Size : (use MTU)
PPP Multilink Reassembly     : disabled
PPP Multilink Mrru           : (use MRU)
PPP Hash Link Selection      : disabled
PPP Initiate IP              : disabled
PPP Initiate IPv6            : disabled
PPP Max-negotiations LCP     : 30
PPP Max-negotiations IPCP    : 30
PPP Max-negotiations IPv6CP : 30
PPP IPCP prompt-option DNS   : enabled
PPP Multilink Multiclass     : disabled
PPP Multilink Multiclasses   : 0
PPP Ipcp Max Negotiation     : 6
PPP Ipcp Negotiation time    : 60
PPP Ipcp Lockout time        : 600
PPP broadcast Accounting Group Name     : groupxyz
PPP Client Username           :
PPP Client Password          :
PPP Client Authentication    :
PPP Client Ip Address        : 0.0.0.0
PPP Client Primary Dns Address    : 0.0.0.0
PPP Client Secondary Dns Address    : 0.0.0.0
PPP Client Primary Wins Address    : 0.0.0.0
PPP Client Secondary Wins Address    : 0.0.0.0
```


To display configuration information for a base profile assigned to a dynamic ATM 1483 subinterface:

```
host1#show profile name atm1483BaseProfile
ATM1483 Auto-configure ip                 : disabled
ATM1483 Auto-configure bridgedEthernet    : disabled
ATM1483 Auto-configure ppp                : enabled
ATM1483 lockout (seconds) ppp             : range : 1-300
ATM1483 Auto-configure pppoe              : enabled
ATM1483 lockout (seconds) pppoe           : range : 1-300
ATM1483 PVC circuit type         :  aal5autoconfig
ATM1483 PVC service category     :  Nrt-Vbr
ATM1483 PVC Peak rate : 10000, Avg rate : 2000, Burst size : 500
ATM1483 Description              :  VC_atm1483
ATM1483 Advisory Rx Speed        :  2000000000



ATM1483 PVC OAM Administrative status: enabled
ATM1483 PVC OAM Loopback frequency:    30

ATM1483 Ip Subscriber information:
  user            : elaine
  domain          : jpeterman.com
  password        : putty
ATM1483 IP Profile               : none assigned
ATM1483 Bridged Ethernet Profile : none assigned
ATM1483 PPP Profile              : none assigned
ATM1483 PPPoE Profile            : pppoeProfile
```

To display configuration information for a base profile assigned to a dynamic VLAN subinterface:

```
host1#show profile name vlanProfile
VLAN Auto-configure ip              : enabled
VLAN Auto-configure pppoe           : enabled
VLAN Svlan Ethertype                : auto-configure
VLAN Advisory Rx Speed              : 100 Kbps
VLAN Advisory Tx Speed              : 2500 Kbps
VLAN Description                    : testing
VLAN IP Profile                     : ipProfile
VLAN PPPoE Profile                  : pppoeProfile
VLAN Service Profile                : none assigned
Bridged Ethernet Mtu                : 1971
Bridged Ethernet Service Profile    : eastServiceProfile
```

To display profile configuration information related to IPv6 Neighbor Discovery router advertisements:

```
host1#show profile name ipv6Profile
IPv6 Unnumbered interface  : loopback 0
IPv6 Router                : default
IPv6 Src-Addr Validation   : Disabled
IPv6 Administrative MTU    : 0
IPv6 ND Enabled            : Enabled
IPv6 ND ManagedConfig      : Disabled
IPv6 ND OtherConfig        : Enabled
IPv6 ND SuppressRa         : Disabled
IPv6 ND RaInterval         : 50
IPv6 ND RaLifeTime         : 1800
IPv6 ND ReachableTime      : 0
IPv6 ND RaPrefix           : 1234::/64
```

```
IPv6 ND ValidLifetime      : 60
IPv6 ND PreferredLifetime  : 60
IPv6 ND PrefixOnLink       : Enabled
IPv6 ND PrefixAutoConfig   : Enabled
IPv6 http redirect Url     : http://www.google.com
```

Meaning Table 57 on page 608 lists the **show profile** command output fields.

Table 67: show profile Output Fields

Field Name	Field Description
Profile	Name of the profile that is displayed
IP address	IP address and subnet mask of the interface, or none if the interface is unnumbered
Unnumbered interface	Specifier for the unnumbered interface, or none if the interface is numbered
Router	Name of the virtual router assigned to the profile; interfaces created by the profile are attached to this virtual router
Directed Broadcast	Enabled or disabled
ICMP Redirects	Enabled or disabled
Access Route Addition	Enabled or disabled
Network Address Translation	Enabled or disabled; domain location (inside or outside)
Source-Address Validation	Enabled or disabled
Ignore DF Bit	Enabled or disabled
Filter Option Packets	Router filters out packets with IP options; enabled or disabled
Administrative MTU	MTU size configured on the profile
TCP MSS value	Maximum segment size for TCP SYN packets traveling through the interface
Inactivity Timer	Inactivity timer setting; enabled or disabled
Route Map Name	Route map applied to the IP interface subscriber; enabled or disabled
Auto Detect	Router automatically detects packets that do not match any entries in the demultiplexer table; enabled or disabled

Table 67: show profile Output Fields *(continued)*

Field Name	Field Description
Auto Configure	Dynamic creation of subscriber interfaces on a primary IP interface; enabled or disabled
IGMP	Enabled or disabled
static-groups	Displays address of any static groups configured for IGMP
Input policy	Name of input policy and whether statistics are enabled or disabled
Output policy	Name of output policy and whether statistics are enabled or disabled
PPP Keepalive	PPP keepalive period, in seconds
PPP Magic Number	Enabled or disabled
PPP Magic Number Mismatch	Indicates whether the router is configured to ignore the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number: ignore (ignore the peer magic number mismatch and retain the PPP connection), or reject (router terminates the PPP connection if it detects a peer magic number mismatch)
PPP Peer DNS Priority	Enabled or disabled
PPP Peer WINS Priority	Enabled or disabled
PPP Authentication	Type of authentication configured: PAP, CHAP, or none
PPP Authentication Router	Name of authentication virtual router
PPP Negotiate MRU	MRU configured for the profile
PPP Packet Log	Enabled or disabled
PPP State Log	Enabled or disabled
PPP Chap Challenge Length	Minimum and maximum Chap Challenge length
PPP Passive Mode	Enabled or disabled
PPP Multilink	Enabled or disabled
PPP IPCP Netmask Option	Enabled or disabled

Table 67: show profile Output Fields *(continued)*

Field Name	Field Description
PPP IPCP Lockout Option	Enabled or disabled
PPP Max-negotiations IPCP	Maximum number of renegotiation attempts that the router accepts before terminating a PPP session
PPP AAA Profile	AAA profile associated with this PPP interface
PPP Multilink Fragmentation	Enabled or disabled
PPP Multilink Fragment Size	Multilink fragment size for this PPP interface
PPP Multilink Reassembly	Enabled or disabled
PPP Multilink Mrru	Multilink MRRU value for this PPP interface
PPP Hash Link Selection	Hash-based link selection for a PPP interface; enabled or disabled
PPP Initiate IP	Initiation of IPv4 over this PPP interface; enabled or disabled
PPP Initiate IPv6	Initiation of IPv6 over this PPP interface; enabled or disabled
PPP IPCP prompt-option dns	Prompts the CPE (Customer Premises Equipment) to negotiate the IPCP primary and secondary DNS options that are locally available with the broadband remote access server; enabled or disabled
PPP Multilink Multiclass	Configuration of multiclass MLPPP on the MLPPP interface: enabled or disabled
PPP Multilink Multiclasses	Number of multilink classes created on the MLPPP interface: 1 through 8
PPP Ipcp Max Negotiation	Maximum number of successful IPCP renegotiations for IPv4 addresses that the router can receive per subscriber: 3 through 6
PPP Ipcp Negotiation time	Time period during which IPCP renegotiations for IPv4 addresses that the router or the provider edge device can receive from a subscriber are restricted: 60 through 600 seconds
PPP Ipcp Lockout time	Time period during which additional IPCP negotiations are prevented: 300 through 600 seconds
PPP broadcast Accounting Group Name	Name of the broadcast virtual router group associated with this PPP profile

Table 67: show profile Output Fields *(continued)*

Field Name	Field Description
PPPoE Max Sessions	Maximum number of PPPoE subinterfaces that can be on an interface
PPPoE Always-offer	Router offers to set up a session for the client, even if the router has insufficient resources to establish a session; enabled or disabled
PPPoE Remote-Circuit-Id	The router captures and processes a vendor-specific tag containing a remote circuit ID transmitted from a digital subscriber line access multiplexer; enabled or disabled
PPPoE Log PPPoeControlPacket	Enabled or disabled
PPPOE duplicate-protect	Enabled or disabled
PPPoE ACNAME	Access concentrator name
PPPoE URL	URL sent in PADM message to PPPoE clients
PPPoE MOTM	Message of the minute sent in the PADM message to PPPoE clients
PPPoE Service-Name Table	Name of the PPPoE service name table, if configured for the specified profile
ATM1483 Auto-configure	Whether autodetection of the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE) is enabled or disabled for a dynamic ATM 1483 subinterface
ATM1483 lockout (seconds)	Encapsulation type lockout setting for the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE) configured on a dynamic ATM 1483 subinterface: • range—Minimum lockout time–maximum lockout time, in seconds • no lockout—Encapsulation type lockout is disabled
ATM1483 PVC circuit type	Encapsulation setting for the PVC configured on a dynamic ATM 1483 subinterface: • aal5autoconfig—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed) • aal5mux ip—VC-based multiplexed circuit for IP only • aal5snap—LLC encapsulated circuit; the LLC/SNAP header precedes the protocol datagram

Table 67: show profile Output Fields *(continued)*

Field Name	Field Description
ATM1483 PVC service category	Service type setting for the PVC configured on a dynamic ATM 1483 subinterface: UBR (the default), UBR PCR, NRT-VBR, RT-VBR, or CBR
ATM1483 PVC Peak rate	Peak cell rate, in Kbps, for the PVC configured on a dynamic ATM 1483 subinterfaces
ATM1483 PVC Avg rate	Average rate, in Kbps, for the PVC configured on a dynamic ATM 1483 subinterface; also referred to as sustained cell rate
ATM1483 PVC Burst size	Length in cells of the burst for the PVC configured on a dynamic ATM 1483 subinterface; also referred to as maximum burst size
ATM1483 Description	Text description assigned to ATM 1483 subinterfaces that are created with this profile
ATM1483 Advisory Rx Speed	Configured receive speed, in Kbps, for the dynamic ATM 1483 subinterface. The E Series LAC sends this value to the LNS in the RX Connect-Speed AVP [38].
ATM1483 PVC OAM Administrative status	Status of OAM F5 loopback cell generation (for VC integrity) on a circuit created with this profile: enabled or disabled
ATM1483 PVC OAM Loopback frequency	Number of seconds between transmissions of OAM F5 end-to-end loopback cells on a circuit created with this profile
ATM1483 Ip Subscriber information	Subscriber login information for the specified dynamic interface type
ATM1483 Profile	Name of the profile assigned to the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE); these profiles are referenced in the base profile for a dynamic ATM 1483 subinterface as nested profile assignments
VLAN Auto-configure	Whether auto detection of the specified upper-interface encapsulation type (IP or PPPoE) is enabled or disabled for a dynamic VLAN subinterface
VLAN Advisory Rx Speed	Configured advisory receive speed, in Kbps, for the dynamic VLAN subinterface; the E Series LAC sends this value to the LNS in the RX Connect-Speed AVP [38]
VLAN Advisory Tx Speed	Configured advisory speed, in Kbps, for the dynamic VLAN subinterface.

Table 67: show profile Output Fields *(continued)*

Field Name	Field Description
VLAN Description	Text description assigned to VLAN subinterfaces that are created with this profile
VLAN Profile	Name of the profile assigned to the specified upper-interface encapsulation type (IP or PPPoE); these profiles are referenced in the base profile for a dynamic VLAN subinterface as nested profile assignments
VLAN Service Profile	Service profile name for a VLAN
VLAN Svlan Ethertype	Ethertype that the packet must use this to create the dynamic VLAN subinterface
Bridged Ethernet Mtu	MTU size configured for a dynamic bridged Ethernet interface
Bridged Ethernet Service Profile	Name of the IP service profile associated with the interface profile for this dynamic bridged Ethernet interface
IPv6 http redirect Url	URL to which a subscriber's initial web browser session is redirected
IPv6 Unnumbered interface	Name of interface without a specific address
IPv6 Router	Router name or default
IPv6 Src-Addr Validation	Source-Address Validation; enabled or disabled
IPv6 Administrative MTU	MTU size
IPv6 ND Enabled	State of the Neighbor Discovery; enabled or disabled
IPv6 ND ManagedConfig	State of the Neighbor Discovery router advertisement managed flag; enabled or disabled
IPv6 ND OtherConfig	State of the Neighbor Discovery router advertisement other config flag; enabled or disabled
IPv6 ND SuppressRa	Status IPv6 router advertisement suppression; enabled or disabled
IPv6 ND RaInterval	Interval (in seconds) of the Neighbor Discovery router advertisement
IPv6 ND RaLifeTime	Lifetime (in seconds) of the Neighbor Discovery router advertisement

Table 67: show profile Output Fields *(continued)*

Field Name	Field Description
IPv6 ND ReachableTime	Amount of time (in milliseconds) that the neighbor is expected to remain reachable
IPv6 ND RaPrefix	Configured prefixes for Neighbor Discovery router advertisement
IPv6 ND ValidLifetime	Amount of time (in seconds) that the router advertises the IPv6 prefix as valid
IPv6 ND PreferredLifetime	Amount of time (in seconds) that the router advertises the specified IPv6 prefix as preferred
IPv6 ND PrefixOnLink	State of the on-link flag; enabled or disabled
IPv6 ND PrefixAutoConfig	State of the use the specified prefix for IPv6 autoconfiguration; enabled or disabled

Related Documentation

- Configuring Subscriber Management for IP Subscribers on Dynamic Bridged Ethernet Interfaces on page 561
- Creating a Profile for Dynamic Interfaces on page 571
- Configuring Profile Characteristics on page 576

- *show profile*

Monitoring Base Profile Assignments and Overriding Profile Assignments for VLAN Major Interface

Purpose Display information, including base profile assignments and overriding profile assignments, for the bulk-configured VLAN ranges on a VLAN major interface. You can use the **show vlan bulk-config** command with no keywords to display information for all VLAN ranges on the router.

You can use the **show vlan bulk-config** command with the **name** keyword to display information for the VLAN range associated with a particular bulk configuration name.

You can use the **show vlan bulk-config** command with the interface specifier and the **name** keyword to display information for a particular VLAN range on a specified VLAN interface.

Action To display information about base profile assignments and overriding profile assignments for all bulk-configured VLAN ranges on the router:

```
host1#show vlan bulk-config
                   Bulk    Start  End  Start End
                  Config   Svlan Svlan Vlan  Vlan    Assigned
   Interface       Name     Id    Id    Id    Id      Profile     Status
---------------- -------- ----- ----- ----- ---- ------------- ------
```



```
FastEthernet 4/6 vlanOnly     1     1     0    0 vlanProfile   Up
FastEthernet 4/6 vlanOnly     2     2   any  any vlanProfile   Up
FastEthernet 0/5 vlanOnly ----- ----- ----- ---- none assigned ------
FastEthernet 4/0 vlanOnly     2     2   any  any none assigned Up
% 4 vlan bulk-config(s) found

Profile override(s):

                   Bulk
                  Config  Svlan Vlan Assigned
   Interface       Name    Id    Id   Profile  Status
---------------- -------- ----- ---- --------- ------
FastEthernet 4/6 vlanOnly     2    3 ipProfile Active
FastEthernet 4/6 vlanOnly     2    4 ipProfile Active
% 2 profile override(s) found
```

To display information about base profile assignments and overriding profile assignments for all VLAN ranges configured on a specified Fast Ethernet interface:

```
host1#show vlan bulk-config interface fastEthernet 4/6
                   Bulk   Start  End  Start End
                  Config  Svlan Svlan Vlan  Vlan   Assigned
   Interface       Name    Id    Id    Id    Id     Profile    Status
---------------- -------- ----- ----- ----- ---- ------------- ------
FastEthernet 4/6 vlanOnly     1     1     0    0 vlanProfile   Up
FastEthernet 4/6 vlanOnly     2     2   any  any vlanProfile   Up

% 2 vlan bulk-config(s) found

Profile override(s):

                   Bulk
                  Config  Svlan Vlan Assigned
   Interface       Name    Id    Id   Profile  Status
---------------- -------- ----- ---- --------- ------
FastEthernet 4/6 vlanOnly     2    3 ipProfile Active
FastEthernet 4/6 vlanOnly     2    4 ipProfile Active
% 2 profile override(s) found
```

Meaning Table 68 on page 717 lists the output fields for the **show vlan bulk-config** command.

Table 68: show vlan bulk-config Output Fields

Field Name	Field Description
Interface	Identifier of the physical interface on which the bulk-configured VLAN range resides. For more information about specifying the VLAN subinterface, see *Interface Types and Specifiers* in *JunosE Command Reference Guide*.
Bulk Config Name	Name of the bulk-configured VLAN range on this interface
End Svlan Id	Ending S-VLAN ID (inclusive) of the S-VLAN group in the subrange
Start Vlan Id	Starting VLAN ID (inclusive) of the VLAN group in the subrange

Table 68: show vlan bulk-config Output Fields *(continued)*

Field Name	Field Description
End Vlan Id	Ending VLAN ID (inclusive) of the VLAN group in the subrange
Assigned Profile	Base profile name for the dynamic VLAN subinterface assigned to this VLAN subrange with "Assigning the Base Profile Configured for a Dynamic VLAN Subinterface to the Single-Tagged VLAN IDs or Double-Tagged S-VLAN IDs" on page 681. When no profile is assigned to the VLAN subrange, the field displays none assigned.
Admin State	Administrative state of the VLAN subrange: up or down

Related Documentation

- Creating a Bulk-Configured VLAN Range on a Static VLAN Major Interface on page 674
- *show vlan bulk-config*

Monitoring Dynamic VLAN Subinterfaces Created with an Overriding Profile Assignment

Purpose Display information about the dynamic VLAN subinterfaces that have been created with an overriding profile assignment. You can use the **bulk-config** keyword to display information about bulk-configured ranges.

Action To display information about the dynamic VLAN subinterfaces that have been created with an overriding profile assignment:

```
host1#show vlan profile override

Profile override(s):

                   Bulk
                  Config Svlan Vlan Assigned
   Interface       Name   Id    Id   Profile  Status
---------------- ------ ----- ---- --------- ------
FastEthernet 4/6 vlanB2  ----    2 ipProfile Active

% 1 profile override(s) found
```

Meaning Table 69 on page 718 lists the output fields for the **show vlan profile** command.

Table 69: show vlan profile Output Fields

Field Name	Field Description
Interface	Type and specifier of the VLAN subinterface
Svlan Id	S-VLAN ID value, if configured
Vlan Id	VLAN ID for the interface

Table 69: show vlan profile Output Fields *(continued)*

Field Name	Field Description
Assigned Profile	Overriding profile to be assigned to the VLAN
Status	Operational status of the overriding profile assignment: Active or Inactive. Active indicates that the router uses the overriding profile to create dynamic interface columns because no static VLAN subinterfaces exist on this interface. Inactive indicates that the router does not use the overriding profile to create dynamic interface columns because a static VLAN subinterface exists on this interface.

Related Documentation

- Adding a Nested Profile Assignment to a Base Profile for a Dynamic VLAN Subinterface on page 672
- *show vlan profile*

Monitoring Status and Configuration Information for VLAN Subinterfaces

Purpose

Display configuration and status information for a specified VLAN subinterface or for all VLAN subinterfaces configured on the router. You can use the **summary** keyword to display only the counts of all VLAN subinterfaces and VLAN major interfaces configured on the router. You can use the **vlan** or **svlan** keywords to display information about specific VLAN IDs or S-VLAN IDs.

Use the **agent-circuit-identifier** keyword to display information about VLAN subinterfaces that are created based on the agent-circuit-id information in the option 82 field of DHCP messages or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets. Using this keyword causes the router to display the agent-circuit-identifier string in the command output.

Action

To display full status and configuration information for all VLAN subinterfaces configured on the router:

```
host1#show vlan subinterface
        Interface        Status    MTU  Svlan Id Vlan Id  Ethertype  Type
--------------------- ---------- ---- -------- -------- --------- -------
ATM 3/0.1.2               Up       1522 ----     11       ----      Static
ATM 3/0.1.3               Up       1522 ----     12       ----      Static
ATM 3/1.1.1               Up       1522 ----     13       ----      Static
ATM 3/1.1.2               Up       1522 ----     14       ----      Static
ATM 3/2.1.1               Down     1526 4        255      0x9100    Static
FastEthernet 4/5.1        Up       1522 ----     1        ----      Dynamic
6 vlan subinterfaces found
```

To display full status and configuration information for the specified VLAN subinterface:

```
host1#show vlan subinterface fastEthernet 4/5.1
        Interface       Status   MTU  Svlan Id Vlan Id  Ethertype  Type
---------------------- -------- ---- -------- -------- --------- -------
```

```
FastEthernet 4/5.1        Up     522 ----       1       ----      Dynamic
1 vlan subinterface found
```

To display full status and configuration information for the specified S-VLAN ID:

host1#**show vlan subinterface svlan id 100 53**
```
        Interface       Status   MTU  Svlan Id Vlan Id Ethertype  Type
---------------------- -------- ---- -------- -------- --------- -------
FastEthernet 0/0.1        Up     1526 100      53       0x9100    Static
FastEthernet 4/6.1        Up     1526 100      53       0x9100    Dynamic
2 vlan subinterfaces found
```

To display full status and configuration information for the specified dynamic VLAN subinterface:

host1#**show vlan subinterface fastEthernet 4/6.1000053**
```
        Interface         Status  MTU  Svlan Id Vlan Id Ethertype Type
------------------------- ------ ---- -------- -------- --------- -------
FastEthernet 4/6.1000053  Up      1526 100      53       0x9100    Dynamic
Auto configure interface(s)    : IP  PPPoE
Detected dynamic interface     : PPPoE
Interface types in lockout     : none
Lockout state (seconds)        : Min Max Current Elapsed Next
------------------------------- --- --- ------- ------- ----
IP                                1 300       0       0    1
PPPoE                             1 300       0       0    1
  In: Bytes 1040, Packets 15
   Multicast 0, Broadcast 1
   Errors 0, Discards 0
  Out: Bytes 984, Packets 15
   Multicast 0, Broadcast 1
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

To display status information for dynamic VLAN subinterfaces that are created based on agent-circuit-identifier information:

host1#**show vlan subinterface**
```
     Interface           Status  MTU  Svlan Id Vlan Id  Ethertype   Type
---------------------- -------- ---- -------- -------- --------- ---------
FastEthernet 4/0.1       Up     1522  2       ----     ----      Dynamic *
FastEthernet 4/0.2       Up     1522  2       ----     ----      Dynamic *
2 vlan subinterfaces found
* Created via agent circuit identifier
```
host1#**show vlan subinterface agent-circuit-identifier**
```
        Interface         Svlan Id Agent-Circuit-Identifier
------------------------- -------- --------------------------
FastEthernet 4/0.1        2        ----
FastEthernet 4/0.2        2        0200D0CB2729E5
```

Meaning Table 58 on page 616 lists the **show vlan subinterface** command output fields.

Table 70: show vlan subinterface Output Fields

Field Name	Field Description
Interface	Type and specifier of the VLAN subinterface

Table 70: show vlan subinterface Output Fields *(continued)*

Field Name	Field Description
Status	Status of the VLAN subinterface: up, down, dormant, lowerLayerDown, absent
MTU	Maximum allowable size (in bytes) of the MTU for the VLAN subinterface
Svlan Id	S-VLAN ID value, if configured
Vlan Id	VLAN ID value for the VLAN subinterface
Ethertype	S-VLAN Ethertype value, if configured
Type	Type of VLAN subinterface: • Static—VLAN or S-VLAN subinterface was configured statically • Dynamic—VLAN or S-VLAN subinterface was configured dynamically
Auto configure interface(s)	Types of dynamic upper interfaces configured with the **auto-configure** command: IP or PPPoE
Detected dynamic interface	Type of dynamic upper interface detected during autoconfiguration: IP, PPPoE, or (if no packet has been received) none
Interface types in lockout	Encapsulation types currently experiencing lockout: IP, PPPoE, or none
Lockout state (seconds)	Settings of encapsulation type lockout for the upper-layer encapsulation type indicated: • Min—Minimum lockout time, in seconds • Max—Maximum lockout time, in seconds • Current—Current lockout time, in seconds; displays 0 (zero) if lockout is not occurring • Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if lockout is not occurring • Next—Lockout time for the router to use for the next lockout event, in seconds

Table 70: show vlan subinterface Output Fields *(continued)*

Field Name	Field Description
In	Analysis of inbound traffic on this interface: • Bytes—Number of bytes received on the VLAN or S-VLAN subinterface • Packets—Sum of all unicast, broadcast, and multicast packets received on the VLAN or S-VLAN subinterface • Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface • Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface • Errors—Total number of errors in all received packets; some packets might contain more than one error • Discards—Total number of discarded incoming packets
Out	Analysis of outbound traffic on this interface: • Bytes—Number of bytes sent on the VLAN or S-VLAN subinterface • Packets—Number of packets sent on the VLAN or S-VLAN subinterface • Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface • Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface • Errors—Total number of errors in all transmitted packets; some packets might contain more than one error • Discards—Total number of discarded outgoing packets
ARP Statistics	Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface: • ARP requests—Number of ARP requests • ARP responses—Number of ARP responses • Errors—Total number of errors in all ARP packets • Discards—Total number of discarded ARP packets
Total VLAN interfaces	Total numbers of VLAN subinterfaces and VLAN major interfaces configured on the router; only this field appears when you specify the **summary** keyword

Related Documentation

- Configuring VLAN Characteristics for a Profile on page 588
- *auto-configure*
- *show vlan subinterface*

PART 2

Index

- Index on page 725

Index

Symbols

10-Gigabit Ethernet modules
specifying an interface....419, 485, 688
802.3ad link aggregation
E120 and E320 routers....196
802.3ah link-fault management
E120 and E320 routers....227
802.3ah OAM link-fault management *See* Ethernet OAM link-fault management

A

AAA (authentication, authorization, accounting)
sending ATM interface descriptions to....41
aaa commands
aaa domain-map....415
aaa tunnel calling-number-format....421
aaa tunnel calling-number-format fallback....421
AAL (ATM Adaptation Layer)....5
AAL5 encapsulation
functions of access nodes....399
interworking function....399
AAL5 encapsulation types....6
AC-Cookie tag, PPPoE....420
AC-Name tag, PPPoE....420
access nodes
as Ethernet switches....399
digital subscriber lines (DSL) and....399
in Ethernet aggregation networks, overview....399
interworking functions....399
accounting statistics
terminated PPP session....296
active discovery mode
and OAM link-fault management....232
configuration on
provider edge devices....232
functions of....232
performance variables
responding to....240
retrieval of....240
starting remote loopback
by the local entity....237
triggering the discovery process....232
address ranges
VPI/VCI....7, 12
Agent-Circuit-Id
format of
configuration scenario for encapsulation lockout, IWF PPPoE sessions....536
Agent-Circuit-Id DSL Forum VSA
usage in dynamic encapsulation lockout for IWF PPPoE sessions....534
agent-circuit-identifier and dynamic VLAN subinterfaces....663, 676
Agent-Remote-Id DSL Forum VSA
usage in dynamic encapsulation lockout for IWF PPPoE sessions....534
aggregating T1 or E1 lines....129, 316
aggregation, Ethernet link....193
AIS (alarm indication signal) cells....14
alarm indication signal cells. *See* AIS cells
ARP (Address Resolution Protocol) table....13
arp commands
arp....486
ARP, Inverse....52
Asynchronous Transfer Mode. *See* ATM
ATM (Asynchronous Transfer Mode)....13, 14
AAL....5
AIS cells, handling....14
ATM passthrough....5
cable lengths....28
CAC....7
CC cells....14
cell scrambling payload....30
configuring....21, 418, 419
dynamic interfaces....527
E120 and E320 routers....10, 23, 83, 447, 454, 479, 595, 696
E320 routers....10, 73
fault management....14
ILMI....7
interface description....3
loopback cells, handling....16, 19
Martini encapsulation....5
module capabilities (statistics)....12
monitoring....73
OAM....14
oversubscription for dynamic interfaces....523, 622

overview....3
ping
configuring....36
overview....18
platform considerations....10
PVCs
configuring....34
configuring individual parameters for....43
defining....535
RDI cells, handling....14
testing....30
VC classes....52
VC integrity....16
VCC-layer connectivity verification....16
See also OAM, ATM
ATM 1483 subinterfaces, dynamic
assigning VC classes to....64, 627, 629
benefits of using....626
bulk configuration of VC ranges....629
changing VC subranges....632
configuring....636
creating static ATM interfaces in VC subranges....632, 655
exporting descriptions....41
monitoring....693
overriding base profile assignments....632
oversubscription....523, 622
overview....626
profiles for....626
restarting LCP negotiations for PPPoA clients....634
atm aal5 commands
atm aal5 description....40
atm aal5 shutdown....25
atm aal5 snmp link-status....25
ATM Adaptation Layer. *See* AAL
atm atm1483 commands
atm atm1483 advisory-rx-speed....25, 588, 670
atm atm1483 description....41
atm atm1483 export-subinterface-description....41
atm atm1483 mtu....25
atm atm1483 shutdown....25
atm atm1483 snmp trap link-status....25
atm atm1483 subscriber....635
atm commands....100
atm auto-configuration....25
atm bulk-config....644, 655
atm cac....27
atm class-vc....642
atm clock internal....27
atm description....41
atm framing....27
atm ilmi-enable....27
atm ilmi-keepalive....7, 27, 28
atm lbo....28
atm oam....32
atm oam flush....35, 36
atm oam loopback-location....35
atm pvc....22, 34, 38, 537, 539, 643
atm shutdown....28
atm snmp trap link-status....28
atm sonet....28
atm sonet stm-1....28
atm uni-version....28, 29
atm vc-per-vp....7, 29, 30
atm vp-description....41
atm vp-tunnel....30
monitor atm vc....69
monitor atm vp....69
ping atm interface atm....36
See also show atm commands
ATM VC Configuration mode....44
ATM virtual circuit (VC)....165
ATM-to-Ethernet interworking
arrival of a PPPoA session at the DSLAM and conversion into a PPPoE session at the DSLAM....401
configuration examples for
1:1 association between DSLAM and B-RAS....406
N:1 association between DSLAMs and B-RAS....407
conversion from ATM to Ethernet
mapping method for tags....401
conversion from Ethernet to ATM
mapping method for tags....401
cross-connect
provision of Layer 2 switching....401
distinction between IWF and non-IWF PPPoE sessions
using the IWF PPPoE tag....401
method of mapping of ATM to Ethernet packets....400
overview....400
translation of ATM into Ethernet traffic....400
transmission of a PPPoE session
to the access concentrator....401

authentication
- EAP....267
- MLPPP....329
- PPP....264, 329
- subscribers on dynamic bridged Ethernet interfaces....528, 561, 635

auto-configure commands
- auto-configure....419, 530, 559
- auto-configure atm1483....643
- auto-configure vlan....675

autodetection of dynamic interfaces....520, 620

B

B-RAS application
- authentication of PPPoE packets....400
- IWF PPPoE sessions
  - multiple DSLAMs connected to an access concentrator....407
  - one DSLAM connected to an access concentrator....406
- usage of the IWF PPPoE tag
  - in identifying IWF sessions....401

B-RAS applications, with PPP sessions....264

B-RAS device
- operation for
  - interworked PPPoA sessions....399
  - interworked PPPoE over ATM sessions....399

bandwidth
- SONET/SDH....375

baseline commands
- baseline atm....66
- baseline atm vp interface....66
- baseline bridge....495
- baseline bridge interface....495
- baseline frame-relay....120
- baseline frame-relay interface....136
- baseline frame-relay multilinkinterface....136
- baseline hdlc interface serial....515
- baseline interface atm....66
- baseline interface pos....382
- baseline ppp (MLPPP)....342
- baseline ppp interface....297
- baseline pppoe interface....425

bridge commands....481
- bridge....480
- bridge acquire....481
- bridge address....481
- bridge aging time....481
- bridge crb....491
- bridge learn....481
- bridge route....491
- bridge snmp-trap link-status....481
- bridge subscriber-policy....486
- bridge-group....481
- *See also* clear bridge commands; show bridge commands; subscriber policy commands

bridge groups
- configuring optional attributes....481
- creating....480
- defined....474
- monitoring....495

bridge interfaces
- configuring....481
- defined....474
- monitoring....495
- supported configurations....475
- types of....475

bridge1483 commands
- bridge1483 mtu....468
- bridge1483 service-profile....563

bridged Ethernet dynamic interfaces....559
- authenticating subscribers....528, 561, 635
- profile characteristics....566

bridged Ethernet static interfaces....451
- application....451
- backward compatible configuration....455
- configuring....455
- configuring for terminated traffic....455
- configuring MTU size....468
- configuring S-VLANs....465
- configuring VLANs....460
- E120 and E320 routers....454
- MAC address validation....455
- monitoring....468
- platform considerations....454
- terminating and routing traffic....452
- VLAN and S-VLAN support....452

bridged IP
- configuring....448
- E120 and E320 routers....446
- overview....445
- platform considerations....446

bridging, transparent
- concurrent routing and bridging....477, 491
- configuration examples
  - bridged Ethernet....493
  - VLANs....493

configuration tasks....480
configuring
bridge groups and bridge interfaces....474, 481
optional bridge group attributes....481
routing....491
subscriber policies....486
creating bridge groups....480
E120 and E320 routers....478
MAC addresses....473, 495
monitoring....495
overview....473
platform considerations....478
prerequisites....479
references....479
setting statistics baselines....495
subscriber policies....476
unsupported features....478
broadcast command....486
bulk configuration of VC ranges....629
assigning VC classes....64, 627, 629
oversubscription....523, 622
restarting LCP negotiations for PPPoA clients....634
with CAC....7, 629
bulk configuration of VLAN ranges....662
bundle
MLFR....129
MLPPP....316

C

C-VLAN tags
virtual path identifier (VPI) mapping to
in ATM-to-Ethernet conversion....400
cable length
ATM interfaces....28
CAC (connection admission control) for ATM
configuring....7, 27
overview....7
with bulk configuration....7, 629
Calling Number AVP
descriptive format configuration....421
format configuration with agent ID suboptions....421
cbr command....46, 57
CE-side load balancing....195
CE1 line modules
MLPPP features....320
cells
AIS....14
ATM....5
F4 OAM....16
fault management....14
handling of ATM loopback....19
loopback....16
RDI....14
Challenge Handshake Authentication Protocol. *See* CHAP
channelized T1 interfaces
MLPPP features....320
channelized T3 interfaces
end-to-end fragmentation and reassembly....116
MLPPP features....320
CHAP (Challenge Handshake Authentication Protocol)....264
circuit ID, capturing for PPPoE
configuring....421
dsl-forum-1 format and examples....394, 421
formatting....392, 421
monitoring....427
overview....392
sending to RADIUS or L2TP....394, 421
troubleshooting....394
using in profiles....566, 586
Cisco HDLC
configuring....511
E120 and E320 routers....510
error frames....509
framing....509
line modules supported....510
monitoring....515
overview....509
platform considerations....510
shutting down the interface....512
SLARP Address Resolution protocol....509
SLARP Keep-Alive protocol....509
SLARP keepalive interval....512
class-int command....63
class-vc command....62
classes, VC
assigning to
ATM major interfaces....62
dynamic ATM 1483 subinterfaces....64, 629, 642

PVCs....61
static ATM 1483 subinterfaces....63
benefits....52
configuring....56
examples
configuration....56
precedence levels....64
monitoring....95
overview....52
precedence levels....54
upgrade considerations....54
clear bridge commands
clear bridge....495
clear bridge address....495
clear bridge interface....495
clock commands
clock source
POS interfaces....378
clock source, selecting
ATM interfaces....27
POS interfaces....378
COCX-F3 line modules
MLPPP features....320
cOCx/STMx interfaces
end-to-end fragmentation and reassembly....116
MLPPP features....320
cold router restart
preservation of
settings for enabling and disabling remote loopback....244
concurrent routing and bridging. *See* CRB
configuration examples
ATM 1483 subinterfaces, dynamic....636
ATM VC classes....56, 64
bridged Ethernet, dynamic....559
bridged Ethernet, static....455
Cisco HDLC....512
IPoA, dynamic....556
MLFR....134
MLPPP, static....326
PPP, serial....281
PPPoE, dynamic....538, 542
PPPoE, static....402, 408
profiles for dynamic interfaces....573
transparent bridging....493
VLAN subinterfaces, dynamic....667, 676
configure file command....590
configuring. *See* specific feature, product, or protocol
connection admission control. *See* CAC for ATM
connection types, ATM
multipoint....4
point-to-point....4
control block
functions of
implementation of discovery process....231
rules management of received OAM PDUs from the parser....231
transmission of OAM PDUs to the multiplexer....231
interface between
OAM client and internal sublayer elements....231
control PVCs, creating....44
conventions
notice icons....xxxi
text and syntax....xxxii
CRB (concurrent routing and bridging)
bridge crb command....491
bridge route command....491
configuring routing....491
defined....477
enabling....491
crc command....107, 379
creating an IP profile
tcp adjust-mss....566
critical event
denoting an unspecified vendor-specific error condition....236
detection of error conditions
in the receive path of the link....236
influencing the link state
using the bit in the Flags field....236
OAM remote failure detection
notified constantly to the local entity....236
critical event PDU
received from the remote peer
influencing the link state....235
OAM link monitoring....235
CT3/T3-F0 line modules
MLPPP features....320
customer support....xxxiii
contacting JTAC....xxxiii

D

data communication equipment. *See* DCE
data PVCs, creating....45
data terminal equipment. *See* DTE
data-link connection identifier. *See* DLCI
DCE (data communication equipment)
  configuring Frame Relay interface as....106, 121, 136
  frame-relay lmi commands....106
  show frame-relay lmi command....121
debugging PPP and PPPoE dynamic interfaces....590
description, interface
  ATM 1483 subinterfaces, exporting....41
  ATM interfaces....40
  ATM virtual paths....41
  Frame Relay interfaces....106
  POS interfaces....112, 380
  sending to AAA....41
  serial interfaces....112
DHCP option 82 field....663
DHCP relay
  with bridged IP....446
digital subscriber line access multiplexers. *See* DSLAMs
discovery process *See* OAM discovery
Discovery protocol with PPPoE....388
DLCI (data-link connection identifier)....106, 116, 125
documentation set
  comments on....xxxiii
ds3-scramble command....30
DSL Forum VSAs
  Agent-Circuit-Id
    encapsulation type lockout....534
  Agent-Remote-Id
    encapsulation type lockout....534
  as criteria for encapsulation type lockout....534
  IWF-Session
    dynamic encapsulation type lockout....534
    PPPoE clients with duplicate MAC addresses....401
dsl-forum-1 format for PPPoE remote circuit ID....394
DSLAM devices
  IWF PPPoE sessions and
    multiple DSLAMs connected to an access concentrator....407
    single DSLAM connected to an access concentrator....406
DSLAMs (digital subscriber line access multiplexers)....264
DTE (data terminal equipment)
  configuring Frame Relay interface as....106, 121, 136
  frame-relay lmi command....106
  show frame-relay lmi command....121
Duplicate event notification OAM PDUs
  viewing....250
duplicate protection
  disabled for
    IWF PPPoE sessions, method of processing....405
    non-IWF PPPoE sessions, method of processing....405
  enabled for
    IWF PPPoE sessions, method of processing....405
    non-IWF PPPoE sessions, method of processing....405
  for an underlying interface
    PPPoE logical interface activation and....404
  for IWF PPPoE sessions
    guidelines for configuring....405
  of IWF PPPoE sessions
    and access to network services....405
    based on MAC addresses....404
    guidelines for configuring....405
    prevention of unauthorized access....405
    scenarios of configuration....405
  topologies for IWF PPPoE sessions
    1:1 association between DSLAMs and the B-RAS application....406
    N:1 association between DSLAMs and the B-RAS application....407
  types of PPPoE sessions
    with the IWF tag absent in PADR....405
    with the IWF tag present in PADR....405
dying gasp
  cases in which it occurs....236
  influencing the link state
    using the bit in Flags field....236
  OAM remote failure detection
    using Flags field in Information PDUs....236
  vendor-specific condition
    notified constantly to the local entity....236

dynamic encapsulation type lockout....532
- based on DSL Forum VSAs
  - for IWF PPPoE sessions....534
- benefits....533
- configuring....530, 532, 556
- criteria used in determining
  - for IWF PPPoE sessions....534
- for IWF PPPoE sessions
  - guidelines for configuring....535
  - same as the behavior for non-IWF sessions, scenario....535
  - sample configuration scenario....536
  - single DSLAM connected to a B-RAS device....536
  - with DSL Forum VSAs not present in PADR....535
- for PPPoE clients....550
  - clearing lockout condition....553
  - configuring....552
  - differences from PPPoE over static ATM....550
  - monitoring....552, 603, 604, 705, 706
- grace period....535
- guidelines....535
- parameters used to perform
  - Agent-Circuit-Id....534
  - Agent-Remote-Id....534
  - MAC address....534

Dynamic Host Configuration Protocol. *See* DHCP relay

dynamic interfaces
- ATM 1483 subinterfaces....527, 626
- autodetection....520, 620
- bridged Ethernet....559
  - authenticating subscribers....528, 561, 635
- configuring from a profile....522, 565, 622
- configuring from RADIUS....527
- configuring IPoA....556
- configuring Multiclass MLPPP....368
- configuring PPP and PPPoE over ATM....538
  - restarting LCP negotiations....540
- configuring PPPoE over static PPPoE....542
  - ATM interface columns....543
  - encapsulation type lockout....550
  - Ethernet and S-VLAN interface columns....549
  - Ethernet and VLAN interface columns....546
  - S-VLAN oversubscription....179, 549
  - static Ethernet interface columns....545
- E120 and E320 routers....525, 624, 625
- E320 routers....150, 168, 526
- inserting dynamic IP routes into routing table....530
- monitoring....593, 691
- oversubscription, ATM....523, 622
- overview....519, 620
- platform considerations....196, 525, 624
- profiles, reassigning....590
- RADIUS authentication....523, 620
- troubleshooting....590
- types supported....520, 621, 657
- VLAN subinterfaces....657
- VLAN subinterfaces with agent-circuit-identifier information....663, 676

E

E120 and E320 routers
- 802.3ad link aggregation....196
- 802.3ah link-fault management....227
- ATM interfaces....10, 23, 73, 83, 447, 454, 479, 595, 696
- bridged IP interfaces....446
- Cisco HDLC interfaces....510
- dynamic interfaces....525, 624, 625
- POS interfaces....282, 376, 380, 383
- PPP interfaces....280, 319
- PPPoE interfaces....398
- transparent bridging....478

e3-scramble command....30

E320 routers
- ATM interfaces....10
- bridged Ethernet....454
- dynamic interfaces....150, 168, 526
- POS interfaces....376
- PPP interfaces....280

EAP (Extensible Authentication Protocol)....267
- authentication methods....267
- authentication negotiation....267
- components....267
- L2TP....267
- limitations....267
- MRU....267
- MTU....267
- performance....267
- retransmission of packets....267

scalability....267
types....267
ECMP (equal-cost multipath)
MLFR alternative to....129
MLPPP alternative to....316
enable commands
enable....590
enabling LMI....106
enabling Multiclass MLPPP....365
encapsulation commands
encapsulation....48, 56
encapsulation bridge1483....448, 455, 462
encapsulation frame-relay ietf....106, 116, 378
encapsulation hdlc....511
encapsulation mlframe-relay ietf....134
encapsulation ppp....174, 176, 378, 455, 462
encapsulation pppoe....418, 419
encapsulation vlan....179, 462, 675, 688
encapsulation method
ATM interface....448
Cisco HDLC interface....511
Frame Relay interface....409, 417, 419
encapsulation type lockout. *See* dynamic encapsulation type lockout
encapsulation, configuring for PVCs....48
end-to-end fragmentation and reassembly, Frame Relay....116
endpoint discriminator
MLPPP....317
equal-cost multipath. *See* ECMP
error frame seconds
configuring
high threshold....241
low threshold....241
window....241
error frame seconds summary events
configuring
high threshold....241
low threshold....241
window....241
error frames
Frame Relay....103
PPP....262
Errored Frame Event TLV
in Event notification OAM PDUs
for tracking frame errors....235
Errored Frame Seconds Summary Event TLV
in Event notification OAM PDUs
for tracking frame seconds summary errors....235
Errored Symbol Period TLV
in Event notification OAM PDUs
for tracking symbol errors....235
ES2-S1 Service IOA
MLPPP features....320
Ethernet aggregation networks
access nodes in, overview....399
Ethernet interfaces
CE-side load balancing....155
changing of the operational status
and generation of a link-fault event....233
commands....155, 186
interface fastEthernet....179
interface lag....200
See also show commands
failing over to the redundant link
when high threshold is exceeded....235
IEEE 802.1Q....165
IEEE 802.3ad....193
L2TP....154
link aggregation (LAG)....193
link-fault management
guidelines for configuring....239
monitoring....155, 186, 216, 221
moving to the down state
when high threshold is exceeded....235
MPLS....149, 200
multinetting....154
OAM link-fault management....233
on SRP modules
and support for link-fault management....240
operational states
down state on OAM association tear-down....233
up state on OAM association rebuilding....233
PPPoE over S-VLANs....175
S-VLANs....166, 179
subnetwork attachment point (SNAP)....165
VLANs....165
Ethernet link aggregation
CE-side load balancing....197
configuring....197
enabling CE-side load balancing....197

interoperation with OAM link-fault
management
operation of 802.3ah below the LAG
sublayer....238
IP interfaces
example....201
MPLS....200, 204
MPLS over VLAN....205
overview....193
PPPoE subinterfaces....199
example....202
VLAN subinterfaces....199
example....203
Ethernet link aggregation commands
interface lag....200
lacp....200
lacp port-priority....200
member-interface....200
minimum-links....200
Ethernet link redundancy
configuration models....206
configuring....215
link behavior....211
overview....206
Ethernet link-fault management
outage time of forwarding controller
and unified ISSU....239
stateful SRP switchover
effect on OAM settings....239
of more than 2 seconds....239
recommendation for PDU loss threshold
setting....239
stateful SRP switchover support and
dependence on completion time....239
unified ISSU
halted when upgrade in progress....239
resumption when upgrade is
complete....239
Ethernet OAM link-fault management
actions on exceeding high threshold
disabling interface....235
failing over to redundant link....235
configuring remote failure detection
critical events....244
dying gasp....244
link fault....244
configuring, procedure....240
disabling
remote loopback functionality on a remote
peer....244
discovery, overview....232
elements, overview
client....229
sublayer....229
enabling
on a particular interface....240
remote loopback functionality on a remote
peer....244
error message
while enabling remote loopback
functionality on a remote peer....244
example
for configuring discovery mode....246
for configuring link monitoring....246
for configuring remote fault
detection....246
for enabling remote loopback on local
entity....246
features, understanding....232
figure showing interconnection with LAG....238
guidelines for configuring....239
implementing
at the interface level....240
Information OAM PDUs
components of....233
initiation of discovery in active mode....233
transmission settings for....233
interrelationship with link aggregation
sublayer....238
Marker response PDUs
and exceeding of high threshold....239
modifying
discovery mode....240
OAM client, functions of....230
OAM link monitoring, overview....234
OAM messages, overview....228
OAM remote and local loopback mode....237
OAM remote fault detection, overview....236
overview....226
performance impact
on ES2 4G LMs....239
performance impact and memory usage
with link monitoring enabled....229
platform considerations....227

platforms supported for
  E120 and E320 routers....227
  ERX7xx, ERX14xx, and ERX310 models....227
prerequisite
  for configuring settings....240
references for....228
remote loopback start and stop settings
  not stored across reboots....244
setting
  the local entity into loopback mode....244
slow protocols
  destination address of OAM PDUs and....228
specifying
  action on exceeding high threshold, disabling interface....243
  action on exceeding high threshold, failover....243
  detection of remote failure conditions....244
  high threshold for frame error events....241
  high threshold for frame seconds summary error events....241
  high threshold for symbol error events....241
  interface type....240
  link events to monitor....241
  low threshold for frame error events....241
  low threshold for frame seconds summary error events....241
  low threshold for symbol error events....241
  mode of discovery....240
  PDU loss threshold before link-fault event is generated....241
  rate of transmission of Information OAM PDUs....241
  remote loopback functionality....244
  tracking of error frame events....241
  tracking of error frame second events....241
  tracking of symbol error events....241
  window for frame error events....241
  window for frame seconds summary error events....241
  window for symbol error events....241
support for unified ISSU....239
support on SRP Ethernet interfaces....240
supported standards....228
viewing
  configurations for all interfaces on which OAM is enabled....255
  details about processing of packets, statistics....250
  discovery mode....247
  general OAM parameters configured on an interface....252
  link monitoring configured on an interface....252
warning message
  while placing local entity in remote loopback mode....244
Ethernet OAM link-fault management commands
  ethernet oam lfm....240
  ethernet oam lfm high-threshold....244
  ethernet oam lfm link-monitor frame-seconds....241
  ethernet oam lfm link-monitor frame-seconds-summary....242
  ethernet oam lfm link-monitor symbol-period....243
  ethernet oam lfm mode....240
  ethernet oam lfm pdu-lost-threshold....241
  ethernet oam lfm pdu-transmit-interval....241
  ethernet oam lfm remote-failure....244
  ethernet oam lfm remote-loopback start....245
  ethernet oam lfm remote-loopback stop....245
  ethernet oam lfm remote-loopback supported....244
  show ethernet oam lfm discovery....247
  show ethernet oam lfm statistics....250
  show ethernet oam lfm status....252
  show ethernet oam lfm summary....255
Ethernet OAM messages *See* OAM messages
Ethertype value, assigning to S-VLANs....179, 466, 588, 673
Event notification OAM PDUs
  description of....229
  generating to the remote peer....237
  generation of and sending to the remote peer when low threshold is exceeded....234
  identification of duplicate events using sequence number....235
  link monitoring function and....234
  receive link path monitoring and transmission to the remote peer....236
  responding to received notices....237

supported TLVs for tracking link faults
Errored Frame Event....234
Errored Frame Seconds Summary Event....234
Errored Symbol Period Event....234
transmission of, multiple times
owing to failure in delivery....235
viewing....250
example....204, 205
examples, configuration
ATM 1483 subinterfaces, dynamic....636
ATM VC classes....56, 64
bridged Ethernet, dynamic....559
bridged Ethernet, static....455
Cisco HDLC....512
IPoA, dynamic....556
MLFR....134
MLPPP, static....326
Multiclass MLPPP, dynamic....368
Multiclass MLPPP, static....369
PPP, serial....281
PPPoE, dynamic....538, 542
PPPoE, static....402, 408
profiles for dynamic interfaces....573
transparent bridging....493
VLAN subinterfaces, dynamic....667, 676
exporting ATM 1483 subinterface descriptions....43
Extensible Authentication Protocol. *See* EAP
external loopback....30

F

F4 OAM cells, for ATM....16
configuring....32
handling of received cells....19
F5 OAM cells, for ATM
configuring....33, 49
disabling automatically....19, 25, 28, 33
handling of received cells....19
failover
action on exceeding high threshold
OAM link monitoring and....235
to secondary link of redundant port
on GE-2 and GE-HDE LMs....235
when high threshold is exceeded....235
Fast Ethernet interfaces
specifying an interface....179
Fast Ethernet modules
configuring PPPoE....408
specifying an interface....419, 481
fault management, ATM....14
Flags field
in Information OAM PDUs
for OAM remote failure detection....236
OAM remote fault detection
Critical Event bit set in....236
Dying Gasp bit set in....236
Link Fault bit set in....236
viewing
configured settings on an interface....255
flush, ATM OAM....35
fragmentation
Frame Relay packets....116
MLPPP
configuring static....334
overview....332
Multiclass MLPPP....364, 366
frame errors
cumulative count of
threshold exceptions....235
display of aggregated
in link event TLVs....235
in the show command output....235
types of
tracked with OAM link monitoring....234
Frame Relay
configuring....106
disabling interface....112
end-to-end fragmentation and reassembly....116
configuring....116
error frames....103
framing....103
interconnection and relationship of NNIs and subnetworks....103
map class....116
maximum payload size....116
monitoring....120
multicast addressing....103
Network-to-Network Interface....103
overview....103
platform considerations....105
SNMP link status processing....113
unicast addressing....103
User-to-Network Interface....103
frame-relay commands....106
frame-relay class....116
frame-relay description....106
frame-relay fragment....116

frame-relay interface-dlci ietf........................106, 116
frame-relay intf-type..106
frame-relay keepalive...106
frame-relay lmi-n391dte...106
frame-relay lmi-n392dce..106
frame-relay lmi-n392dte...106
frame-relay lmi-n393dce..106
frame-relay lmi-n393dte...106
frame-relay lmi-t391dte..106
frame-relay lmi-t392dce...106
frame-relay lmi-type...106
map-class frame-relay..116
See also show frame-relay commands
framing
ATM interfaces...27
capabilities of E Series routers...........103, 262, 509
POS interfaces...380

G

GE-2 line modules
with GE-2 SFP I/O modules
action when high threshold is
exceeded...235
action when link fault occurs........................236
failing over to the redundant link.................235
failover for remote faults................................236
GE-HDE line modules
with GE-2 SFP I/O modules
action when high threshold is
exceeded...235
action when link fault occurs........................236
failing over to the redundant link.................235
failover for remote faults................................236
Gigabit Ethernet modules
configuring MLPPP..320
configuring PPPoE...408
specifying an interface.........................419, 485, 688
grace period, dynamic encapsulation type
lockout...535
group (multicast) addressing..103
groups, bridge. *See* bridge groups

H

hash-based packet distribution with MLPPP 320
HDLC (High-Level Data Link Control), Cisco. *See*
Cisco HDLC
HDLC (High-Speed Data Link Control)
PPP framing...262
serial encapsulation...509
hdlc commands
hdlc down-when-looped..512
hdlc keepalive...512
hdlc shutdown...512
high availability
and atm atm1483 subscriber command...........635
and subscriber command.............................528, 562
high threshold
action when exceeded on
GE-2 LMs with GE-2 SFP I/O
modules..235
GE-HDE LMs with GE-2 SFP I/O
modules..235
actions performed on exceeding
failing over to the redundant link..................235
moving the interface state to down,
disabling..235
for OAM link monitoring
action taken when exceeded........................234
when exceeded, disabling the
interface..234
when exceeded, failover to secondary
link..234
specifying
for frame error events.......................................241
for frame seconds summary error
events..241
for symbol error events....................................241
when exceeded
default action taken..235
high threshold setting
configuring for an interface
for link monitoring..243
high threshold value
recommendation on setting
for link-fault monitoring..................................239
High-Level Data Link Control. *See* Cisco HDLC
High-Speed Data Link Control. *See* HDLC
higher-level protocols over Ethernet
platform considerations...150

I

IEEE 802.1Q...165
IEEE 802.1w..197
IEEE 802.1w (Rapid Reconfiguration of Spanning
Tree)...197
IEEE 802.3ad...193
IEEE 802.3ad (Link Aggregation)...................................197

IEEE 802.3ah OAM link-fault management *See* Ethernet OAM link-fault management
ILMI....106
ILMI (integrated local management interface)
about....7
enabling....27
keepalive timer....28
inarp command....52, 58
individual (unicast) addressing....103
Information OAM PDUs
bandwidth allocation for OAM traffic
using the OAM sublayer capabilities TLV....233
bidirectional transmission of
keepalive for OAM association....233
components, overview....233
contained TLV fields
OAM sublayer capabilities....233
state....233
vendor OUI....233
vendor-specific details....233
description of....229
forwarding to the remote peer in cases
where no other types of OAM PDUs are ready for sending....233
loss threshold setting
action taken when configured value is exceeded....233
definition of....233
maximum transmission rate
impact on performance....239
receipt by the remote peer
and wait time for obtaining local entity details....233
determination of formation of OAM association....233
reestablishment of OAM association
link-fault event clearance....233
restart of discovery
link-fault event generation....233
scenario of OAM association failure
with remote peer not supporting loopback....233
timer setting
maximum value....233
minimum value....233
rate of transmission in milliseconds....233
transmission settings for, overview....233
transmission to the remote entity
triggering of the discovery process and....233
using the Flags field of
and OAM remote fault detection....236
viewing....250
with Link Fault event bit set
not containing TLV data....236
integrated local management interface. *See* ILMI
interface commands
interface atm....23, 450, 455, 462
interface fastEthernet....179, 419, 481
interface gigabitEthernet....419, 485, 688
interface lag....200
interface mlframe-relay....134
interface pos....106, 282, 380
interface serial
Cisco HDLC interfaces....512
Frame Relay interfaces....112, 116
PPPoE interfaces....434
interface tenGigabitEthernet....419, 485, 688
interface description
Frame Relay interfaces....106
POS interfaces....112, 380
serial interfaces....112
interfaces
10-Gigabit Ethernet....419, 485, 688
bridge. *See* bridge interfaces
Fast Ethernet....419, 481
Gigabit Ethernet....419, 485, 688
serial. *See* serial interfaces
interfaces, monitoring
Ethernet....155, 186, 216, 221
internal loopback....30
International Telecommunication Union. *See* ITU
Internet Protocol Control Protocol. *See* IPCP
interworking function *See* IWF
Inverse ARP....52
IP addresses
assigning to Cisco HDLC interfaces....512
assigning to Frame Relay subinterfaces....112, 116
ip commands
ip (subscriber policies)....486
ip access-routes....577
ip address....112, 116, 155, 455, 462, 512, 577
ip auto-configure ip-subscriber....577
ip auto-detect ip-subscriber....577
ip directed-broadcast....577

ip filter-options all....577
ip igmp....577
ip ignore-df-bit....577
ip inactivity-timer....577
ip mac-validate....455
ip mtu....577
ip nat....577
ip policy....577
ip redirects....577
ip route-map ip-subscriber....577
ip sa-validate....577
ip tcp adjust-mss....577
ip unnumbered....577, 688
ip virtual-router....577
IP over ATM. *See* IPoA
IP over VLAN over bridged Ethernet....461
IP profile....566
IP routes, inserting dynamic routes into routing table....530
IPCP (Internet Protocol Control Protocol)
option 0x90....582
overview....272
IPCP negotiation
based on RADIUS returned attributes....274
IPoA (IP over ATM)
configuring dynamic interfaces....556
dynamic interfaces....523, 556
IPv4 and IPv6 over Static PPPoE
IPv4 and IPv6 interface columns....409
IPv6
neighbor discovery, defining....580
ipv6 commands
ipv6 address....580
ipv6 mld....580
ipv6 mtu....580
ipv6 nd....580
ipv6 nd managed-config-flag....580
ipv6 nd other-config-flag....580
ipv6 nd prefix-advertisement....580
ipv6 nd ra-interval....580
ipv6 nd reachable-time....580
ipv6 nd suppress-ra....580
ipv6 policy....580
ipv6 sa-validate....580
ipv6 unnumbered....580
ipv6 virtual-router....580
IPv6 neighbor discovery commands
ipv6 nd reachable-time....580
IPv6 profile....567
IPv6CP negotiation
based on RADIUS returned attributes....274
ITU (International Telecommunication Union) recommendation, ATM OAM standards....14
IWF
configuration scenarios
multiple DSLAMs connected to an access concentrator....407
single DSLAM connected to an access concentrator....406
defined by the DSL Forum
for conversion of PPPoA and PPPoE over ATM sessions....401
definition of, uses....401
enabling migration of DSL networks
from ATM to Ethernet links....401
PPPoE sessions
dynamic encapsulation type lockout....534
with duplicate MAC addresses....399, 401, 405
with duplicate protection feature enabled....403
IWF PPPoE sessions
duplicate protection of
guidelines for configuring....405
dynamic encapsulation type lockout
configuration example....536
guidelines for configuring....535
using DSL Forum VSAs as criteria....534
processing of
duplicate MAC addresses in multiple sessions....404
dynamic encapsulation type lockout....534
with duplicate MAC addresses of clients
ATM-to-Ethernet interworking....401
dependence on pppoe sessions command configuration....403
limit on maximum number of sessions....403
scenario, multiple DSLAMs linked to a B-RAS....407
scenario, single DSLAM linked to a B-RAS....406

IWF PPPoE tag
  insertion in the PADR packet
    sent from the PPPoE client to the access concentrator....401
  used to distinguish
    between an IWF and a non-IWF PPPoE session....401
IWF-Session DSL Forum VSA
  absence in PADR packets
    duplicate protection, disabled....405
    duplicate protection, enabled....405
  presence in PADR packets
    and B-RAS application....399, 401
    duplicate protection, disabled....405
    duplicate protection, enabled....405
    duplicate protection, guidelines....405
    dynamic encapsulation type lockout....534
    enabling multiple sessions with the same MAC address....403
  suboption for
    field code....407
    length field, value....407

J

J-Flow commands
  ip route-cache flow sampled....577

K

keepalive timer, setting....28

L

L2TP (Layer 2 Tunneling Protocol)....154
  profile characteristics....566
  using PPPoE remote circuit ID....394, 421
l2tp commands
  l2tp policy....582
L2TP Proxy Authenticate Extensions for EAP—draft-ietf-l2tpext-proxy-authen-ext-eap-01.txt (December 2006 expiration)....269, 281
LACP (Link Aggregation Control Protocol)....194
  PPPoE subinterfaces....199, 202
  redundant member link behavior....211
LAG bundle
  position of
    and OAM link-fault management....229
    between OAM sublayer and MAC client....229
LAG member link
  marked as redundant
    using link monitoring for failover and link reversal....239
LAG multiplexer
  feeding LAC protocol logic....239
  feeding link events....239
layer 2 services over MPLS
  802.3ad link aggregation....194
  802.3ad switch....194
Layer 2 Tunneling Protocol. *See* L2TP
LCP (Link Control Protocol)
  configuration options....263, 316, 363
    endpoint discriminator....316
    MRRU....316
    Multilink header format 363
    Prefix elision....363
    SSN header format....316
  restarting negotiations for PPPoA clients....540
Link Control Protocol. *See* LCP
link events
  and processing of LAC protocol logic....239
  local generation of
    using health monitoring....239
  supplied to the LAG multiplexer....239
link fault
  bit set in the Flags field
    influencing the link state based on....236
  example condition of occurrence....236
  loss of signal detection
    by the receiver....236
  rate of transmission
    in Information OAM PDUs....236
Link fault information OAM PDUs
  detection of receive path failure....240
  sending of
    not supported....240
Link Integrity Protocol. *See* LIP
link monitoring
  frequent polling of error counters
    when enabled on the local entity....229
  impact on performance and memory usage
    when disabled on the local entity....229
    when enabled on the local entity....229
link monitoring, of Ethernet interfaces *See* OAM link monitoring

link-fault event
clearing of
interface transition to the up state..............233
reestablishment of OAM association.........233
generation of
non-receipt of OAM PDUs within the loss threshold..........233
link-fault management commands
for Ethernet interfaces
show ethernet oam..........235
link-fault management, Ethernet interfaces *See* Ethernet OAM link-fault management
LIP (Link Integrity Protocol)..........262
LMI (local management interface)
configuring counters and timers..........106
configuring type..........106
enabling..........106
monitoring..........121, 136
load-interval command
ATM interfaces..........30
POS interfaces..........380
local loopback mode *See* OAM loopback mode
local management interface. *See* LMI;ILMI
local OAM entity
health monitoring threshold exceeded
failover to secondary link..........239
prerequisite for configuring
remote loopback on..........244
preventing both local and remote entities
from simultaneous loopback..........237
retrieval and response to performance variables
in active mode..........240
sending of performance variables to peer
support for..........240
transmission of Information OAM PDUs
receipt by the remote peer..........233
lockout, encapsulation type. *See* dynamic encapsulation type lockout
loopback
detection on Cisco HDLC interfaces..........512
loopback command
ATM interfaces..........30
POS interfaces..........380
Loopback control OAM PDUs
description of..........229
sending of
for termination of loopback..........238
start of loopback of remote peer
by local entity..........237
viewing..........250
loopback for ATM VC integrity
cells..........16
configuring..........32, 33
F4 OAM cells..........16
handling of received cells..........19
loopback frames
forwarding of
from the parser to the subordinate sublayer..........231
processing by the packet parser
transmission to the multiplexer..........231
lost-link timer
and Ethernet OAM link-fault management..........234
equivalent to
product of PDU timer and loss threshold..........234
resetting OAM discovery..........234
low threshold
for OAM link monitoring
generation of Event notification PDU.........234
response when exceeded..........234
specifying
for frame error events..........241
for frame seconds summary error events..........241
for symbol error events..........241
low threshold value
recommendation for setting
for link-fault monitoring..........239

M

MAC (media access control) addresses
bridging overview..........473
configuring for S-VLANs..........166
configuring for VLANs..........166
removing from forwarding table..........495
using with bridged Ethernet..........455
validation on bridged Ethernet interfaces.........455
MAC address
usage in dynamic encapsulation lockout
for IWF PPPoE sessions..........534
macro command..........590
macros
using to configure dynamic interfaces..........589
magic numbers..........263

manuals
- comments on....xxxiii

map class, Frame Relay fragmentation....116
map entries, Frame Relay....121, 136
map list, using to configure NBMA interfaces....38
map-class frame-relay command....116
map-group command....40
map-list command....40
Marker response OAM PDUs
- circumstances of non-transmission of....239
- link event caused by exceeding high threshold and disabling sending of....239

maximum payload size, Frame Relayfragmentation....116
maximum receive unit. *See* MRU
media access control addresses *See* MAC addresses
member-interface command....136
message of the minute messages. *See* MOTM messages
MFR
- aggregating OC3/STM1 lines....130

MLFR (Multilink Frame Relay)
- aggregating T1 or E1 lines....129
- aggregation limits....132
- bundle....129
- bundle limits....132
- configuring....134
- interfaces, monitoring....136
- Link Integrity Protocol....130
- member link sequence numbers....132
- NxT1 service....129
- overview....129
- packet distribution, round-robin....132
- platform considerations....131
- protocol layering....131
- unsupported features....133

MLP. *See* MLPPP
MLPPP (Multilink PPP)
- aggregating T1 or E1 lines....316
- aggregation limits....320
- authentication....329
- bundle....316
- bundle limits....320
- commands, contextual differences of....328
- configuring....326
- ECMP....316
- endpoint discriminator....317
- features....320
- fragmentation
  - configuring static....334
  - overview....332
- interface stacking....316
- interfaces, monitoring....342
- member link sequence numbers....320
- MRRU....317
- NxT1 service....316
- overview....316
- packet distribution....320
- platform considerations....318
- profiles....566
- protocol layering....316
- reassembly
  - configuring static....334
  - overview....332
- SSN header format....317
- statistics and baselining....342
- unsupported features....325

modules
- ATM capabilities....12

monitor commands
- monitor atm vc....69
- monitor atm vp....69
- monitor vlan interface....186

Monitoring and Troubleshooting Point-to-Point Protocol over Ethernet. *See* PPPoE
monitoring. *See* specific feature, product, or protocol
MOTM (message of the minute) messages....412
MP. *See* MLPPP
MPLS (Multiprotocol Label Switching)
- configuring over Ethernet link aggregation....194
- configuring S-VLAN tunnels....177, 179

mpls commands
- mpls....154, 200
- mpls (subscriber policies)....486
- mpls atm vci range 462

MPLS over VLAN over bridged Ethernet....461
MRRU LCP configuration option....317
MRU (maximum receive unit)
- POS interfaces....380
- PPP interfaces....263

mru command....380
MTU (maximum transmission unit) *See* MTU
- bridged Ethernet interfaces....468
- Ethernet interfaces....200
- POS interfaces....380

mtu command....380
Ethernet interfaces....200
multicast addressing....103
multicast command....486
Multiclass MLPPP
configuration guidelines....361
configuring....364
enabling....365
example, dynamic....368
example, static....369
fragmentation....361, 364, 366
monitoring....369
overview....361
platform considerations....363
reassembly....361, 364, 367
traffic-class....361, 362, 364, 366
Multilink Frame Relay. *See* MLFR
multilink maximum received reconstructed unit (MRRU)....317
Multilink PPP. *See* MLPPP
multinetting....154
multiplexer
pass-through of MAC client-generated frames without any modification....231
transfer of loopback frames
forwarded to the subordinate sublayer....231
received from the parser....231
working of
for link-fault management of Ethernet interfaces....231
multipoint, ATM connections....4, 13

N

NBMA....13
NBMA (nonbroadcast multiaccess)....13
configuring....37
description....13
Inverse ARP....14
point-to-multipoint....13
removing circuits....14
static mapping....14
NCP
component of PPP sessions
overview of....283
NCP packets
arriving on POS interfaces
in a sequenced manner....283
in an out-of-order format....283
nested profile assignments
VLAN subinterfaces....627, 660
Network Control Protocol *See* NCP
NNI (Network-to-Network Interface),Frame Relay....103, 106, 116
nonbroadcast multiaccess. *See* NBMA
notice icons....xxxi
NxT1 service
MLFR....129
MLPPP....316

O

OAM (Operation, Administration, and Management), ATM....14
cc (continuity check) cells....14
configuring....31, 49
disabling F5 OAM services....19, 25, 28, 33
flush....36
standards....14
OAM association
after establishment of
transmission rate and packet loss threshold settings....233
formation of 233
keepalive for
using bidirectional transfer of OAM PDUs....233
rebuilding of
operational up state of the interface and....233
successful rediscovery with the remote peer
link transition to the up state....234
tearing down of
generation of the link-fault event....233
OAM client
enabling and configuration of OAM sublayer....230
enabling OAM operations on the link
during the discovery phase....230
management of
remote loopback functionality....230
responses to OAM PDUs received from the remote peer....230
monitoring OAM PDUs from the remote peer....230
oam commands
oam ais-rdi....50, 56
oam cc....50, 56

oam retry....51, 56
oam-pvc....51, 56
OAM discovery
automatic activation of....232
configuration at the interface level....233
controlling other OAM attributes for an entity....232
enabling on an interface....232
exceeding of PDU loss threshold
link transition to the down state....234
first phase of OAM association....232
implementation of
by the control block....231
interconnection with
loopback operation....232
variable requests....232
monitoring the remote peer....232
restarting of
Information OAM PDU loss threshold....233
restriction on performing
scenario in which it occurs....232
successful rediscovery with the remote peer
link transition to the up state....234
supported types
active mode....232
passive mode....232
topology for configuration of
customer edge devices....232
provider edge devices....232
OAM discovery mode
configuring....240
example, configuring....246
monitoring....247
OAM elements
operations of
and link-fault management....229
parts of
client....229
sublayer....229
OAM features
backward compatibility with older networks....232
benefits of 232
IEEE 802.3ah enhancements and....232
supported....232
types of
discovery....232
link monitoring....232
remote and local loopback....232
remote fault detection....232
OAM link monitoring
actions on high threshold excess
disabling interface....235
failing over to the redundant link....235
advertisement of OAM attributes
using Information OAM PDUs....234
continuation with a new window
when the link is in up state....234
detecting frame and symbol errors....234
error frame seconds summary
overview of....235
error frames per second
overview of....235
error symbols per second
overview of....235
example, configuring....246
generation of Event notification OAM PDUs
when low threshold is exceeded....234
health and quality analysis of the link
using tracking of frame and symbol errors....234
improving failover and recovery times
configuration of OAM sublayer....238
influencing the link state
on exceeding link quality threshold....235
on receipt of a critical event PDU from peer....235
influencing the operational link state
when high threshold is exceeded....234
overview of feature....234
period for detecting errors....234
physical redundancy
preservation of MAC bindings....239
preservation of the same MAC address....239
switchover from primary to secondary link....239
without indicating to higher-layer protocols....239
repeated sending of Event notification PDUs
preventing duplication....235
supported error events
error frame....235
error frame seconds summary....235
error symbol period....235

supported frame errors tracked
alignmentError 234
frameCheckSequence 234
frameTooLong....234
lengthError (runts)....234
symbol errors counting
on IOAs that do not support....235
threshold for frame, symbol errors....234
using Event notification OAM PDUs....234
using the results of
to configure LAG sublayer....238
viewing configured values....250
OAM link-fault event *See* link-fault event
OAM link-fault management *See* Ethernet OAM link-fault management
OAM link-fault management, 802.3ah....227
OAM loopback mode
cessation by the initiating entity
using Loopback control PDUs....238
configuring....244
higher priority for OAM PDUs
over data traffic....238
in disabled state
receipt of Information PDUs by local entity....237
restarting sending of locally sourced non-OAM PDUs....237
in enabled state
received data calculation....238
transmitted data calculation....238
initiation phase
discarding locally sourced non-OAM PDUs....237
intentional discard of data frames
by remote peer....238
local and remote functionalities
overview....237
preventing loopback on both local and remote entities
verification to perform....237
processing of OAM PDUs
measuring link performance....238
remote
feature supported by the remote peer....237
link quality between local and remote entities....237
non-OAM PDUs resent to the local entity....237
placing a peer into loopback mode....237
processing of OAM PDUs....237
settings for enabling and disabling remote loopback
not stored across reboots....244
viewing configuration....250
warning message
while placing local entity in remote loopback....244
OAM messages
format of
Code field, usage....228
Data/Pad field, usage....228
Flags field, usage....228
format of, figure....228
forwarding to the control block....231
higher priority over data traffic
during remote loopback....238
length of untagged frames....240
maximum transmission rate
performance impact....239
processing by multiplexer....231
processing by the packet parser....231
processing during remote loopback
measuring link performance....238
properties of
determination of maximum frame size....228
rate of transmission per second....228
size range....228
value of destination MAC address....228
value of Ethertype....228
value of frame payload....228
rules management of received
by the control block....231
supported types
Event notification OAM PDUs....229
Information OAM PDUs....229
Loopback control OAM PDUs....229
Vendor-specific OAM PDUs....229
transmission of
control block and....231
transmission rate and
impact on router operations....229
with link-fault event bit in Flags field
higher preferential processing....236

OAM PDU loss threshold
actions taken when exceeded
link problem signaled....234
link transition to the down state....234
configuring....241
definition of
action taken when configured value is exceeded....233
product of transmission rate and
equals lost-link timer....234
recommendation for setting
for link-fault monitoring....239
viewing configured value....252
OAM PDU transmission rate
configuring....241
definition of
range for configuration....233
product of loss threshold and
equals lost-link timer....234
viewing configured setting....252
OAM PDUs *See* OAM messages
OAM remote failure monitoring *See* OAM remote fault detection
OAM remote fault detection
action when link fault occurs
disabling interface....236
failing over to secondary port....236
conditions notified using
Flags field of Information OAM PDU....236
example, configuring....246
monitoring the receive link path
generation of Event notification PDUs....236
types of failures monitored
critical event....236
dying gasp....236
link fault....236
OAM remote loopback settings
viewing....252
OAM session
changed to down, nonfunctional
after the action for high threshold excess is taken....235
in down state with loss of OAM association
status in show command output....234
OAM state machine
port in the down state
until discovery process completion....233
OAM sublayer
below the 802.3ad LAG sublayer
interworking with link aggregation....238
capabilities of
assessment of supported functions for remote entities....233
bandwidth utilization for OAM traffic....233
maximum OAM PDU size for sending and receiving....233
rate limit on number of frames transferred....233
components of
control block, functions....231
multiplexer, functions....231
packet parser, functions....231
interworking with link aggregation
situated below the LAG bundle....229
placement of
above the MAC layer....229
below the LLC layer....229
presentation of interface to clients
for MAC clients....229
for OAM clients....229
OAM traffic
higher priority over data frames
when oversubscription occurs....238
OC3 modules
monitoring ATM interfaces....73
testing ATM interfaces....30
OCx/STMx interfaces
MLPPP features....320
Operation, Administration, and Maintenance, Ethernet interfaces *See* Ethernet OAM link-fault management
Operation, Administration, and Management. *See* OAM, ATM
option 82 field, DHCP....663
OSI Network Layer Control Protocol. *See* OSINLCP
OSINLCP (OSI Network Layer Control Protocol)....262
overriding base profile assignments
monitoring....693
overview....632, 665
removing from PVC....647
removing from VC range or subrange....647
removing from VLAN range or subrange....678
removing from VLAN subinterface....678
VLANs....678

oversubscription
dynamic ATM interfaces.....523, 622
S-VLANs.....179
oversubscription of allotted bandwidth
discarding of data traffic
higher priority for OAM PDUs.....238
with peer in loopback.....238

P

packet logging, PPP and PPPoE.....590
packet over SONET. *See* POS
packet parser
classification of received frames
and forwarding to appropriate entities.....231
delivery of OAM PDUs
to the control block.....231
working of
for OAM link-fault management.....231
PADM (PPPoE Active Discovery Message), configuring.....412
PADN (PPPoE Active Discovery Network) messages, configuring.....415
PADR (PPPoE Active Discovery Request)
IWF PPPoE sessions
criteria for encapsulation type lockout.....534
IWF-Session DSL VSA absent in
duplicate protection disabled.....405
duplicate protection enabled.....405
IWF-Session DSL VSA present in
duplicate protection disabled.....405
duplicate protection enabled.....405
suboptions, field code.....406
suboptions, length field.....406
with IWF-Session DSL VSA
for differentiating between IWF and non-IWF sessions.....401
from access nodes to the DSLAM.....401
PADS (PPPoE Active Discovery Session) packets, configuring.....420
PAP (Password Authentication Protocol).....264
passive discovery mode
and OAM link-fault management.....232
configuration on
customer edge devices.....232
enabled on local and remote entities
and influence on discovery process.....232
working of.....232
Password Authentication Protocol. *See* PAP
payload, ATM cell.....5
PCR (peak cell rate).....9
PDU loss threshold *See* OAM PDU loss threshold
performance impact
with link-fault management
frequency of querying MAC-layer statistics.....239
frequency of retrieval of errored symbols.....239
maximum frequency of PDUs transmission.....239
on ES2 4G LMs.....239
performance variables
for link-fault management
retrieval and response to.....240
sending of.....240
permanent virtual circuit. *See* PVC
physical level redundancy
OAM link monitoring
preservation of MAC bindings.....239
preservation of the same MAC address.....239
switchover from primary to secondary link.....239
without indicating to higher-layer protocols.....239
physical link redundancy
on GE-2 and GE-HDE line modules
action performed on exceeding high threshold.....235
failing over to redundant link.....235
ping atm interface atm command.....36
ping, ATM
configuring.....36
overview.....18
platform considerations
ATM.....10
bridged Ethernet.....454
bridged IP.....446
Cisco HDLC.....510
dynamic interfaces.....196, 525, 624
Ethernet interfaces.....167
Frame Relay.....105
higher-level protocols over Ethernet.....150
MLFR.....131
MLPPP.....318
Multiclass MLPPP.....363
OAM link-fault management.....227

POS....376
PPP....279
PPPoE....397
transparent bridging....478
point-to-multipoint, NBMA connection....13
Point-to-Point Protocol over ATM. *See* PPPoA
Point-to-Point Protocol over Ethernet. *See* PPPoE
Point-to-Point Protocol. *See* PPP
policies, subscriber. *See* subscriber policies for transparent bridging
policy commands
atm policy....671
frame-relay policy....671
gre-tunnel policy....671
ip policy....671
l2tp policy....671
mpls policy....671
vlan policy....588, 671
policy management
baselining statistics....671
preserving statistics....671
statistics....671
POS (packet over SONET)
configuring interface....378
disabling interface....380
E120 and E320 routers....282, 376, 380, 383
E320 routers....376
line modules supported....376
monitoring interface....382
overview....375
platform considerations....376
references....377
pos commands
pos description....112, 380
pos framing....380
pos scramble-atm....380
POS interfaces
remaining in the down state
when NCP packets are out-of-order....283
with NCP packets in out-of-order format....283
shifting to the up state
with NCP packets in out-of-order format....283
transitioning to the up state
when NCP packets are out-of-order....283
with PPP encapsulation enabled
and receipt of NCP packets....283
PPP (Point-to-Point Protocol)
accounting statistics for terminated sessions....296
Async-Control-Character-Map (ACCM) option....262
authentication....264, 329
configuring....281, 285, 286, 287, 288, 289, 290, 291, 292, 295
optional....284
configuring dynamic interfaces....538
E120 and E320 routers....280, 319
E320 routers....280
EAP....267
Extensible Authentication Protocol....267
magic numbers....263
network control protocol....316
packet logging....590
platform considerations....279
PPP profiles....566
sequencing of NCP packets,
disabling....283
enabling....283
troubleshooting
dynamic interfaces....590
static interfaces....312
ppp commands
ppp aaa-accounting-broadcast....582
ppp aaa-profile....582
ppp authentication....292, 329, 582
ppp chap-challenge-length....295, 582
ppp description....285
ppp fragmentation....582
ppp hash-link-selection (MLPPP)....582
ppp initiate-ip....582
ppp initiate-ipv6....582
ppp ipcp lockout....286
ppp ipcp netmask....286, 582
ppp ipcp-lockout-duration....582
ppp ipcp-max-negotiation....582
ppp ipcp-nego-duration....582
ppp keepalive....287, 582
ppp log....312, 582
ppp magic-number disable....287, 582
ppp magic-number ignore-mismatch....288, 582
ppp max-bad-auth....295
ppp max-bad-auth (MLPPP)....329
ppp max-negotiations....288, 582
ppp mru....289, 582
ppp multilink enable (MLPPP)....332, 582

ppp passive-mode....290, 582
ppp peer....290, 582
ppp peer-ip-address-optional....291
ppp reassembly....582
ppp shutdown....291
PPP encapsulation
on POS interfaces
NCP packets in a sequenced format....283
NCP packets in out-of-order format....283
PPP Multilink. *See* MLPPP
PPP sessions
enabling establishment of
when NCP packets are in sequence....283
when NCP packets are out-of-sequence....283
PPPoA (Point-to-Point Protocol over ATM)
terminating stale subscribers and restarting LCP negotiations....540
PPPoE 199
PPPoE (Point-to-Point Protocol over Ethernet)
Active Discovery Initiation (PADI) packets....388
Active Discovery Message (PADM)....412
Active Discovery Network (PADN) messages....415
Active Discovery Offer (PADO) packets....388
Active Discovery Request (PADR) packets....388
Active Discovery Session (PADS) packets....388, 420
agent-circuit-identifier information....663
capturing remote circuit ID
configuring....421
dsl-forum-1 format and examples....394, 421
formatting....392, 421
monitoring....427
overview....392
sending to RADIUS or L2TP....394, 421
troubleshooting....394
using in profiles....566, 586
configuring dynamic interfaces over ATM....538
configuring dynamic interfaces over PPPoE....542
configuring for Ethernet....408, 545
configuring over ATM....402
Discovery protocol....388
DSL Forum VSA 26-1....663
duplicate protection of
IWF PPPoE sessions, guidelines for configuring....405
IWF PPPoE sessions, processing of 404
IWF PPPoE sessions, scenarios....405
non-IWF PPPoE sessions, scenarios....405
duplicate protection scenarios for IWF PPPoE sessions
1:1 association between DSLAMs and the B-RAS....406
N:1 association between DSLAMs and the B-RAS....407
dynamic encapsulation type lockout....534, 550
clearing lockout condition....553
configuring....552
differences from PPPoE over static ATM....550
monitoring....552, 603, 604, 705, 706
See also and IWF PPPoE sessions
E120 and E320 routers....398
overview....388
packet logging....590
platform considerations....397
subinterfaces for LACP....199, 202
troubleshooting
dynamic interfaces....590
static interfaces....441
PPPoE access concentrator
connected to a single DSLAM
for IWF sessions....406
connected to multiple DSLAMs
for IWF sessions....407
IWF PPPoE sessions
duplicate protection of....406
dynamic encapsulation type lockout....536
pppoe commands....412
pppoe....455, 462
pppoe (subscriber policies)....486
pppoe acname....586
pppoe always-offer....586
pppoe duplicate-protection....586
pppoe log pppoeControlPacket....586
pppoe motm....412, 586
pppoe pads disable-ac-info....420
pppoe remote-circuit-id....421, 586
pppoe service-name-table....417, 418, 419
pppoe sessions....586
pppoe subinterface....455, 462

pppoe url...412, 586
service...417
See also show pppoe commands
PPPoE over S-VLAN over bridged Ethernet...467
PPPoE over VLAN over bridged Ethernet...461
PPPoE service name tables
action, defined...390
enabling for dynamic interfaces...419
enabling for static interfaces...418
Ethernet configurations...419
overview...390
service name tag, defined...390
PPPoE sessions
with IWF-Session DSL Forum VSA *See* IWF PPPoE sessions
with IWF-Session DSL VSA
contained in PADR packets...401
duplicate MAC addresses of the DSLAM...401
limit on maximum number of sessions...401
precedence levels for VC classes
examples...64
overview...54
prerequisite
for configuring link-fault management on Ethernet interfaces...240
for enabling remote peer to loop back PDUs...244
privilege level
for troubleshooting dynamic interfaces...590
profile commands...332
profile...332, 419, 589
profile atm1483 bulk-config-name pvc...650
profile vlan override...682
profile-reassign...590
vlan profile...588
vlan service-profile...673
See also show profile commands
profiles
assigning to a static interface...571
assigning to an interface...572
characteristics for...566
creating for debugging...590
description...522, 622
for dynamic ATM 1483 subinterfaces...627
for dynamic VLAN subinterfaces...658
monitoring...593, 691
overriding base profile assignments...632, 665
reassigning for troubleshooting...590
using to configure dynamic interfaces...565
working with...566
protocol data units (PDUs)
number that the remote peer can fail to receive
specifying...241
rate of transmission of
specifying...241
standard length of
for link-fault management of Ethernet interfaces...241
proxy ARP...445
PVC (permanent virtual circuit)
assigning VC classes...61
ATM interfaces...400, 535
bulk configuration of...629
configuring encapsulation...48
configuring F5 OAM...49
configuring for bridged IP...448
configuring individual parameters for...43
configuring Inverse ARP...52
configuring service category...46
creating control PVCs...44
creating data PVCs...45
Frame Relay...106, 116, 121, 136
overview...3
pvc command
for control PVCs...44
for data PVCs...45

R

RADIUS (Remote Authentication Dial-In User Service)
authentication of dynamic interfaces...523, 527, 620
configuring dynamic interfaces from...527
overriding attributes for PPPoE...394, 421
using PPPoE remote circuit ID...394, 421
radius commands
radius override calling-station-idremote-circuit-id...421
radius override nas-port-id remote-circuit-id...421
radius remote-circuit-id-delimiter...392
radius remote-circuit-id-format...392
radius remote-circuit-id-format (dsl-forum-1 keyword)...394
ranges, VC. *See* VC ranges, bulk configuration of

ranges, VLAN *See* VLAN ranges, bulk configuration
of RDI (remote defect indication) cells
RDI (remote defect indication) cells....14
reassembly
Frame Relay packets....116
MLPPP
configuring static....334
overview....332
Multiclass MLPPP....364, 367
reassigning profiles to dynamic interfaces....590
relearn command....486
remote circuit ID, capturing for PPPoE
configuring....421
dsl-forum-1 format and examples....394, 421
formatting....392, 421
monitoring....427
overview....392
sending to RADIUS or L2TP....394, 421
troubleshooting....394
using in profiles....566, 586
remote defect indication cells. *See* RDI cells
remote loopback mode *See* OAM loopback mode
remote loopback operation
halting
calculation and display of received data at the prompt....245
calculation and display of transmitted data at the prompt....245
viewing calculated loopback details....245
prerequisite for enabling remote peer to loop back PDUs....244
starting....244
stopping....244
remote loopback request
error message
if local entity is not placed in loopback mode....244
if remote entity is not configured for loopback....244
Loopback control OAM PDU and....237
placing the interface of remote peer in loopback....237
received from the local entity
by the remote peer....237
remote OAM entity
prerequisite for enabling loopback operation on....244
remote OAM peer
Ethernet OAM link-fault management and....233
example scenario
for not building the OAM association....233
intentional discard of data frames
when loopback is enabled....238
notifying forwarding state
to local OAM entity....237
preventing both local and remote entities
from simultaneous loopback....237
receipt of Information PDUs
determination of OAM connection establishment....233
processing of configured rules on local entity details....233
supporting remote loopback
discarding locally sourced non-OAM PDUs....237
forwarding non-OAM PDUs to local entity....237
round-robin packet distribution with MLPPP....320
routing, configuring for transparent bridging....491
RSTP (Rapid Spanning Tree Protocol)....197
Ethernet link redundancy 206, 213

S

S-VLAN tags
virtual circuit identifier (VCI) mapping to
in ATM-to-Ethernet conversion....400
S-VLANs (stacked virtual local area networks)
address possibilities....166
bridged Ethernet configurations....465
configuring PPPoE over S-VLAN....166
configuring to support dynamic PPPoE....549
configuring tunnel interfaces
advantages....177
example....178
interface stacking....178
displaying status....186
oversubscription....179
overview....166
PPPoE over S-VLAN over bridged Ethernet....467
scrambling ATM cell payload....30
scripts
using to configure dynamic interfaces....589
SDH (Synchronous Digital Hierarchy)....375, 376
serial description command....112

serial interfaces
- configuring....112, 116, 434, 512

Serial Line Address Resolution Protocol. *See* SLARP
service category, configuring for PVCs....46
service name tables, PPPoE. *See* PPPoE service name tables
SFPs (small form-factor pluggable transceivers)....156, 159
short sequence number (SSN) header format....317
show aaa commands
- show aaa ipv4-addr-saving....311

show atm commands
- show atm aal5 interface....71, 593, 691
- show atm atm1483....71
- show atm bulk-config....693
- show atm interface....73
- show atm map....75
- show atm oam....75
- show atm ping....75
- show atm subinterface....83, 590, 595, 696
- show atm vc....88, 600, 702
- show atm vc atm 90
- show atm vc-class....95
- show atm vp....95
- show atm vp-description....95
- show atm vp-tunnel....71, 100, 593, 691
- show nbma arp....101

show bridge commands
- show bridge....495
- show bridge groups....495
- show bridge interface....495
- show bridge port....495
- show bridge table....495

show bridge1483 interface command....469
show columns command....602, 704
show frame-relay commands
- show frame-relay interface....121, 136
- show frame-relay lip....136
- show frame-relay lmi....121, 136
- show frame-relay map....121, 136
- show frame-relay multilinkInterface....136
- show frame-relay pvc....121, 136
- show frame-relay subinterface....127, 146
- show frame-relay summary....127, 146

show hdlc interface command....516
show interfaces commands
- show interfaces fastEthernet....155, 186, 188
- show interfaces gigabitEthernet....155, 190
- show interfaces lag....216, 221
- show interfaces pos 383
- show interfaces tenGigabitEthernet....155, 190

show ip mac-validate command....471
show mpls cross-connects atm command....101
show ppp commands
- show ppp interface 297, 299, 300
- show ppp interface summary....297, 309
- show ppp peer-ip-address-optional....311
- show ppp session-To-Thirteen-Years....313

show pppoe commands
- show aaa tunnel-parameters....426
- show pppoe interface....427, 428, 603, 705
- show pppoe interface full....428, 429
- show pppoe interface lockout-time....604, 706
- show pppoe interface summary....430
- show pppoe subinterface....434, 605, 707
- show pppoe subinterface full....434
- show pppoe subinterface summary....434
- show pppoe-service-name-table....433, 434
- show profile....436
- show radius override....441

show profile commands
- show profile 606, 708

show subscriber-policy command....495
show vlan commands
- show vlan subinterface....468, 614, 719

show vlan subinterface command....183
shutdown command....113
- dynamic interfaces....650, 682
- Frame Relay....112
- POS....380

shutting down interfaces
- dynamic ATM 1483....650
- dynamic VLAN....682
- Frame Relay....112
- POS....380

Simple Network Management Protocol. *See* SNMP link status processing
SLARP (Serial Line Address Resolution Protocol)....509
- Keep-Alive protocol....509
- keepalive interval....512
- overview....509
- *See also* Cisco HDLC

slow protocols
- bandwidth requirements....228
- definition of
  - rate of transmission of OAM PDUs....228

multicast address of
and destination MAC address of remote entities....233
OAM messages and....228
SMs (Service modules)
MLPPP features....320
SNAP (subnetwork attachment point)....165
SNMP (Simple Network Management Protocol)
link status processing....113
snmp trap frame-relay link-status command....113
SONET (Synchronous Optical Network)....28, 375
source, clock
ATM interfaces....27
SRP Ethernet interfaces
support for link-fault management....240
SSN LCP (short sequence number Link Control Protocol) configuration option....317
stacked virtual local area networks. *See* S-VLANs
stateful SRP switchover
and atm atm1483 subscriber command....635
and subscriber command....528, 562
Ethernet link-fault management
dependence on completion time....239
effect on OAM settings....239
of more than 2 seconds....239
recommendation for high and low threshold configuration....239
recommendation for PDU loss threshold setting....239
static interfaces....520, 621
configuring Multiclass MLPPP....369
creating in VC subranges....632, 655
creating in VLAN subranges....688
static mapping, and NBMA interfaces....38
subinterfaces
ATM 1483, dynamic....626
ATM 1483, static....3
Frame Relay, monitoring....127
MLFR, monitoring....146
VLAN, dynamic....657
subnetwork attachment point. *See* SNAP
subranges, VC. *See* VC subranges
subscriber command....523, 528, 589
subscriber management, using for subscriber authentication....528, 635
subscriber policies for transparent bridging....476
configuring....486
defined....476
monitoring....495
See also subscriber policy commands
subscriber policy commands....495
arp....486
bridge subscriber-policy....486
broadcast....486
ip....486
multicast....486
pppoe....486
relearn....486
show subscriber-policy....495
subscriber-policy....486
unicast....486
unknown-destination....486
unknown-protocol....486
subscribers
authenticating on dynamic bridged Ethernet interfaces....528, 561, 635
identifying for PPPoE....392
support, technical *See* technical support
SVC (switched virtual circuit)
ATM....5
Frame Relay....106, 116
svlan commands
svlan ethertype....179, 466, 588, 673
svlan id
for S-VLAN tunnels....179
for standard S-VLANs....466
switch, 802.3ad....196
switched virtual circuit. *See* SVC
symbol errors
cumulative count of
threshold exceptions....235
detected using OAM link monitoring
for Ethernet interfaces....234
display of aggregated
in link event TLVs....235
in the show command output....235
enabling on IOAs that do not support
and events generation for that link....235
monitoring of
on IOAs that do not support....235

symbol period errors
configuring
high threshold....241
low threshold....241
window....241
effect of configuration on
interfaces that do not support....243
exceeding threshold for
configuring LAG for removal of member link....239
configuring LAG to rebalance bundle....239
recommendation for configuring....243
Synchronous Digital Hierarchy. *See* SDH
Synchronous Optical Network. *See* SONET
system
framing capabilities....262
system clock
selecting clock source....27

T

technical support
contacting JTAC....xxxiii
terminated PPP session accounting statistics....296
text and syntax conventions....xxxii
traffic management, ATM
types....12
traffic-class
Multiclass MLPPP....364, 366
traffic-shaping parameters....523
transmit clock source, configuring
ATM interfaces....27
transparent bridging. *See* bridging, transparent
troubleshooting interfaces
dynamic PPP and PPPoE....590
PPP....312
PPPoE....394, 441
TSMs (Tunnel Service modules)
MLPPP features....320

U

ubr command....47, 60
UNI (User-Network Interface)
version....7, 29, 73, 104, 116
unicast addressing....103
unicast command....486
unified ISSU
and Ethernet link-fault management....239
completion of
resumption of link-fault monitoring....239
in progress
halt of link-fault management....239
Uniform Resource Locator messages. *See* URL messages
unknown- commands
unknown-destination....486
unknown-protocol....486
Unsupported OAM PDUs
viewing....250
URL (Uniform Resource Locator) messages....412
User-Network Interface. *See* UNI version

V

vbr commands
vbr-nrt....47, 56
vbr-rt....47, 56
VC (virtual channel)
integrity....16
per port, ATM....12
VC (virtual circuit), monitoring....66
VC classes
assigning to
ATM major interfaces....62
dynamic ATM 1483 subinterfaces....64, 629, 642
PVCs....61
static ATM 1483 subinterfaces....63
benefits....52
configuring....56
examples
configuration....56
precedence levels....64
monitoring....95
overview....52
precedence levels....54
upgrade considerations....54
VC ranges, bulk configuration of
assigning VC classes....64, 627, 629
example....636
oversubscription....523, 622
overview....629
restarting LCP negotiations for PPPoA clients....634

VC subranges
- changing
  - monitoring....693
  - overview....632
- creating static ATM interfaces in
  - configuring....655
  - monitoring....693
  - overview....632
- overriding profile assignments....632
- oversubscription....523, 622

vc-class atm command....61
VCC (virtual channel connection)....5
VCD (virtual circuit descriptor)....448
VCI (virtual channel identifier)....5, 449
vendor OUI TLV
- component of Information OAM PDU
  - value of, usage....233

vendor-specific attributes. *See* VSAs
vendor-specific information
- and Ethernet OAM link-fault management
  - Information OAM PDU component....233
- format of, size of
  - usage....233

Vendor-specific OAM PDUs
- description of....229

versions
- UNI (User-Network Interface)....29, 73

virtual channel connection. *See* VCC
virtual channel. *See* VC
virtual circuit descriptor. *See* VCD
virtual circuit identifier (VCI) tags
- mapped to S-VLAN tags
  - in ATM-to-Ethernet conversion....400

virtual circuit. *See* VC, monitoring
virtual connections, ATM....5
virtual local area networks. *See* VLANs
- IPv4 and IPv6 over Dynamic PPPoE....548

virtual path connection. *See* VPC
virtual path descriptions, assigning....41
virtual path identifier (VPI) tags
- mapped to C-VLAN tags
  - in ATM-to-Ethernet conversion....400

virtual path identifier. *See* VPI
virtual path tunnels. *See* VP tunnels, ATM
virtual path. *See* VP, monitoring
vlan commands
- encapsulation vlan....675
- monitor vlan interface....186
- profile vlan override....682
- vlan advisory-rx-speed....588, 670
- vlan advisory-tx-speed....588
- vlan auto-configure....588
- vlan auto-configure agent-circuit-identifier....588, 671
- vlan bulk-config....688
- vlan description....588
- vlan id....462
- vlan profile....588
- vlan service-profile....673

VLAN subinterfaces
- base profiles....659
- nested profile assignments....627, 660
- overriding profile assignments....665

VLAN subinterfaces, dynamic
- bulk configuration of VLAN ranges....662
- changing VLAN subranges....665, 683
- configuring....667
- configuring with agent-circuit-identifier information....663, 676
- creating static VLAN interfaces in VLAN subranges....688
- overriding base profile assignments....665
- overview....657
- profiles for....626

VLAN subranges
- changing
  - adding to VLAN range....683
  - changing administrative state....683
  - configuring....683
  - merging....683
  - overview....665
  - removing from VLAN range....683
  - shortening or expanding....683
- creating static VLAN interfaces in
  - configuring....688

VLANs (virtual local area networks)
- bridged Ethernet configurations....460
- configuring....168
- configuring dynamic subinterfaces for....657
- configuring for dynamic IP....657
- configuring for dynamic PPPoE....657
- configuring IPv4 and IPv6 over dynamic PPPoE....548
- configuring to support dynamic PPPoE....546
- displaying status....186
- IP over VLAN over bridged Ethernet....461
- monitoring....186
- MPLS over VLAN over bridged Ethernet....461

overview....165
PPPoE over VLAN over bridged Ethernet....461
VP (virtual path) tunnels, ATM
overview....7
per module assembly....12
traffic rate....30
VP (virtual path), monitoring....66
VPC (virtual path connection)....5
VPI (virtual path identifier)....5, 165, 449
VSAs (vendor-specific attributes)
DSL Forum 26-1....663
levels of CLI access....590

W

warm router restart
preservation of
remote loopback enabling and disabling settings....244

X

X.21/V.35 interfaces
MLPPP features....320
XFPs (10-gigabit small form-factor pluggable transceivers)....159