



Junos[®] OS

Translational Cross-Connect and Layer 2.5 VPNs Feature Guide

Release

10.4



Published: 2010-10-05

Revision 1

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

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Junos® OS Translational Cross-Connect and Layer 2.5 VPNs Feature Guide

Release 10.4

Copyright © 2010, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Writing: Roy Spencer, Fawn Damitio, Ines Salazar, Richard Hendricks, and Walter Goralski

Editing: Nancy Kurahashi, Dawn Spencer

Illustration: Faith Bradford, Fawn Damitio, Nathaniel Woodward, Richard Hendricks, and Dawn Spencer

Cover Design: Edmonds Design

Revision History

October 2010—R1 Junos 10.4

The information in this document is current as of the date listed in the revision history.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The 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

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

- a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.
- b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.
- c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.
- d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.
- e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14 (ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at <http://www.gnu.org/licenses/gpl.html>, and a copy of the LGPL at <http://www.gnu.org/licenses/lgpl.html>.

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Table of Contents

Part 1	Translational Cross-Connect and Layer 2.5 VPNs	
Chapter 1	Translational Cross-Connect and Layer 2.5 VPNs Concepts and Reference Materials	3
	Translational Cross-Connect and Layer 2.5 VPNs Overview	3
	Translational Cross-Connect and Layer 2.5 VPNs System Requirements	4
	Translational Cross-Connect and Layer 2.5 VPNs Terms and Acronyms	5
	Configuring Layer 2.5 VPNs	5
	Configuring the Encapsulation on Interfaces Participating in the Layer 2.5 VPN	6
	Configuring the Layer 2.5 VPN	7
Chapter 2	Translational Cross-Connect and Layer 2.5 VPNs Configuration	9
	Configuring TCC Interface Switching	9
	Defining the Encapsulation for Layer 2 TCC Switching	10
	Configuring ARP Functionality for Ethernet TCC Circuits	11
	Configuring ARP Functionality for Extended VLAN Circuits	12
	Configuring Static ARP For Ethernet and Extended VLAN TCCs	12
	Defining the Connection for Layer 2 TCC Switching	13
	Configuring MPLS	13
	Option: Configuring ISO or MPLS Traffic on T Series, M320, and MX Series Routers	14
Chapter 3	Translational Cross-Connect and Layer 2.5 VPNs Configuration Example	15
	Example: PPP to ATM TCC Configuration	15
	Verifying Your Work	17
	Example: Frame Relay to Fast Ethernet TCC Configuration	17
	Verifying Your Work	19
	Example: Layer 2.5 VPN Configuration	19
	Verifying Your Work	25
	Router PE1 Status	25
	Router PE2 Status	27
	Router P Status	28
	For More Information	29
Part 2	Index	
	Index	33

List of Figures

Part 1	Translational Cross-Connect and Layer 2.5 VPNs	
Chapter 1	Translational Cross-Connect and Layer 2.5 VPNs Concepts and Reference Materials	3
	Figure 1: TCC Concept Example	4
Chapter 2	Translational Cross-Connect and Layer 2.5 VPNs Configuration	9
	Figure 2: Layer 2 TCC Switching	9
Chapter 3	Translational Cross-Connect and Layer 2.5 VPNs Configuration Example	15
	Figure 3: TCC Interface Switching—PPP to ATM	15
	Figure 4: TCC Interface Switching—Frame Relay to Fast Ethernet	17
	Figure 5: Layer 2.5 VPN Topology Diagram	20

PART 1

Translational Cross-Connect and Layer 2.5 VPNs

- Translational Cross-Connect and Layer 2.5 VPNs Concepts and Reference Materials on page 3
- Translational Cross-Connect and Layer 2.5 VPNs Configuration on page 9
- Translational Cross-Connect and Layer 2.5 VPNs Configuration Example on page 15

CHAPTER 1

Translational Cross-Connect and Layer 2.5 VPNs Concepts and Reference Materials

This chapter covers these topics:

- Translational Cross-Connect and Layer 2.5 VPNs Overview on page 3
- Translational Cross-Connect and Layer 2.5 VPNs System Requirements on page 4
- Translational Cross-Connect and Layer 2.5 VPNs Terms and Acronyms on page 5
- Configuring Layer 2.5 VPNs on page 5
- Configuring the Encapsulation on Interfaces Participating in the Layer 2.5 VPN on page 6
- Configuring the Layer 2.5 VPN on page 7

Translational Cross-Connect and Layer 2.5 VPNs Overview

Translational cross-connect (TCC) is a switching concept that allows you to establish interconnections between a variety of Layer 2 protocols or circuits. It is similar to its predecessor, circuit cross-connect (CCC). However, while CCC requires the same Layer 2 encapsulations on both sides of a Juniper Networks router (such as PPP-to-PPP or Frame Relay-to-Frame Relay), TCC lets a network administrator connect different types of Layer 2 protocols interchangeably. With TCC, combinations such as PPP-to-ATM and Ethernet-to-Frame Relay cross-connections are possible. Also, TCC can be used to create Layer 2.5 VPNs and Layer 2.5 circuits.

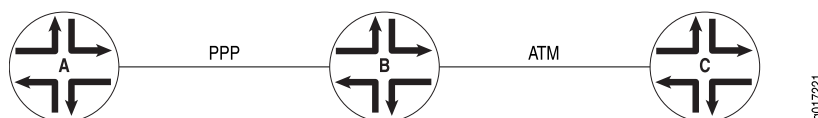
The Junos OS makes interworking between unlike protocols possible. The software strips off the Layer 2 header when frames enter the router and adds a different Layer 2 header before the frames exit the router. TCC supports these Layer 2 protocols:

- ATM
- Cisco HDLC
- Ethernet
- Extended VLAN

- Frame Relay
- PPP

In Figure 1 on page 4, the PPP header is stripped from frames arriving at Router B and an ATM header is added before the frames are sent to Router C. All Layer 2 negotiations are terminated at the interconnecting router (Router B). Examples include Link Control Protocol (LCP) and Network Control Protocol (NCP) for PPP, keepalives for Cisco HDLC, and Local Management Interface (LMI) for Frame Relay.

Figure 1: TCC Concept Example



TCC functionality is different from standard Layer 2 switching. TCC only swaps Layer 2 headers. No other processing, such as header checksums, time-to-live (TTL) decrementing, or protocol handling, is performed. Currently, TCC is supported in IPv4, ISO, and MPLS.

This guide shows you how to use the Layer 2 interworking nature of TCC for interface switching and for constructing Layer 2.5 virtual private networks (VPNs).

Translational Cross-Connect and Layer 2.5 VPNs System Requirements

To implement TCC, your system must meet these minimum requirements:

- Junos OS Release 10.1 or later for support of IS-IS and MPLS traffic over Layer 2.5 VPNs on MX Series routers.
- Junos OS Release 8.3 or later for support of IS-IS and MPLS traffic over Layer 2.5 VPNs on T Series and M320 routers.
- Junos OS Release 8.2 or later for support on M120 routers and MX Series routers
- Junos OS Release 6.2 or later for CCC, TCC, and Layer 2.5 VPN support on M320 routers
- Junos OS Release 6.0 or later for CCC, TCC, and Layer 2.5 VPN support on T Series routers
- Junos OS Release 5.6 or later for proxy Address Resolution Protocol (ARP) functionality in Ethernet TCC and extended VLAN TCC networks on M Series routers
- Junos OS Release 5.4 or later for Ethernet TCC and extended VLAN TCC interface encapsulation types on M Series routers
- Junos OS Release 5.2 or later for PPP TCC, Cisco HDLC TCC, ATM TCC, and Frame Relay TCC interface encapsulation types on M Series routers
- Ethernet-based PICs to support Ethernet TCC and extended VLAN TCC interface encapsulation types, with the following exceptions:
 - All 8-port, 12-port, and 48-port Fast Ethernet PICs do not support TCC.

- 4-port Gigabit Ethernet PICs do not support extended VLAN TCC.
- Any combination of three Juniper Networks M Series or T Series routers for TCC or five routers for Layer 2.5 VPNs or Layer 2.5 circuits

Translational Cross-Connect and Layer 2.5 VPNs Terms and Acronyms

C

circuit cross-connect (CCC)	A method of exchanging frames between two router interfaces running the same Layer 2 protocol, such as ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP. For more information about CCC, see the <i>Junos Network Interfaces Configuration Guide</i> .
customer edge router (CE)	Any router that connects a customer site to a PE router.

L

Layer 2.5 circuits	A method of privately connecting sites across the Internet. Layer 2.5 circuits function like Layer 2 circuits, but connect two sites that use different Layer 2 protocols. For more information about Layer 2 circuits, see the <i>Junos VPNs Configuration Guide</i> .
Layer 2.5 VPNs	A method of privately connecting sites across the Internet. Layer 2.5 VPNs function like Layer 2 VPNs, but connect two sites that use different Layer 2 protocols. For more information about Layer 2 VPNs, see the <i>Junos VPNs Configuration Guide</i> .

P

provider core router (P)	Any router that lies between PE routers in the provider core network.
provider edge router (PE)	Any router that connects customer edge (CE) routers to the provider core network.

T

translational cross-connect (TCC)	A Juniper Networks method of exchanging frames between two router interfaces running different Layer 2 protocols, such as ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP. For more information about TCC, see the <i>Junos Network Interfaces Configuration Guide</i> .
--	---

Configuring Layer 2.5 VPNs

Layer 2.5 VPNs are an extension to Layer 2 VPNs. The main difference is that Layer 2.5 VPNs are not required to have the same media on both ends of the connection. In general, Layer 2.5 VPNs combine the capabilities of TCC with Layer 2 VPNs.

Layer 2.5 VPNs support the same media types as TCC: ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP. They support IPv4, IS-IS, and MPLS traffic types, but do not support IPv6 traffic.

Although not covered in this manual, you can also use TCC in conjunction with Layer 2 circuits to connect two CE sites that use different Layer 2 protocols. You can use these

so-called Layer 2.5 circuits as an alternative to Layer 2.5 VPNs. For more information about Layer 2 circuits, see the *Junos VPNs Configuration Guide*.

To configure Layer 2.5 VPNs, complete the following tasks:

- Configuring the Encapsulation on Interfaces Participating in the Layer 2.5 VPN on page 6
- Configuring the Layer 2.5 VPN on page 7
- Option: Configuring ISO or MPLS Traffic on T Series, M320, and MX Series Routers on page 14

Configuring the Encapsulation on Interfaces Participating in the Layer 2.5 VPN

The encapsulation types used for Layer 2.5 VPNs are parallel to CCC encapsulations and identical to the TCC encapsulations explained in “Configuring TCC Interface Switching” on page 9. The encapsulation types are:

- **atm-tcc-vc-mux**
- **atm-tcc-snap**
- **cisco-hdlc-tcc**
- **ethernet-tcc**
- **extended-vlan-tcc**
- **frame-relay-tcc**
- **ppp-tcc**

When you configure a TCC encapsulation type, some modifications are needed to handle VPN connections over unlike Layer 2 links and to terminate the Layer 2 protocol locally. The router performs the following media-specific changes:

- **PPP TCC**—Both Link Control Protocol (LCP) and Network Control Protocol (NCP) are terminated on the router. Internet Protocol Control Protocol (IPCP) IP address negotiation is not supported. The Junos OS strips all PPP encapsulation data from incoming frames before forwarding them. For frames destined to a PPP-connected neighbor, PPP encapsulation is added.
- **Cisco HDLC TCC**—Keepalive processing is terminated on the router. The Junos OS strips all Cisco HDLC encapsulation data from incoming frames before forwarding them. Cisco-HDLC encapsulation is added onto frames that are destined to a Cisco HDLC-connected neighbor.
- **Frame Relay TCC**—All Local Management Interface (LMI) processing is terminated on the router. The Junos OS strips all Frame Relay encapsulation data from incoming frames before forwarding them. For frames destined to a Frame Relay-connected neighbor, Frame Relay encapsulation is added.
- **ATM**—Operation, Administration, and Maintenance (OAM) and Interim Local Management Interface (ILMI) processing is terminated at the router. Cell relay is not

supported. The Junos OS strips all ATM encapsulation data from incoming frames before forwarding them. ATM encapsulation is added onto frames that are destined to an ATM-connected neighbor.

Configuring the Layer 2.5 VPN

Layer 2.5 VPNs use essentially the same configuration format as Layer 2 VPNs. You should already have experience configuring Layer 2 VPNs before you attempt to configure a Layer 2.5 VPN. The major steps required to create a Layer 2 VPN are:

- Enable MPLS on interfaces pointing toward the core and the edge on your provider edge (PE) and provider core (P) routers.
- Configure Label Distribution Protocol (LDP) on all P and PE routers for traffic traveling from the PEs, through the core, and to the remote PEs.
- Establish an internal BGP (IBGP) Layer 2 VPN peering relationship between PE routers.
- Set up policies on your PE routers that will set a private community tag on outbound BGP traffic heading to the remote PEs and accept incoming traffic that matches similar community traffic from the remote PEs.
- Build VPN routing and forwarding (VRF) instances on your PE routers and apply the previously configured policies to deliver private traffic to the customer edge (CE) routers.
- In addition to the above configurations procedures, you can specify a Layer 3 protocol family for a Layer 2.5 VPN on T Series and M320 routers. See the “Option: Configuring ISO or MPLS Traffic on T Series, M320, and MX Series Routers” on page 14 section for configuration information.

CHAPTER 2

Translational Cross-Connect and Layer 2.5 VPNs Configuration

This chapter contains the following:

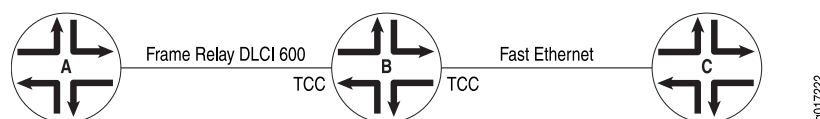
- Configuring TCC Interface Switching on page 9
- Defining the Encapsulation for Layer 2 TCC Switching on page 10
- Configuring ARP Functionality for Ethernet TCC Circuits on page 11
- Configuring ARP Functionality for Extended VLAN Circuits on page 12
- Configuring Static ARP For Ethernet and Extended VLAN TCCs on page 12
- Defining the Connection for Layer 2 TCC Switching on page 13
- Configuring MPLS on page 13
- Option: Configuring ISO or MPLS Traffic on T Series, M320, and MX Series Routers on page 14

Configuring TCC Interface Switching

TCC joins disparate Layer 2 logical interfaces by means of a virtual Layer 2 switching technique.

Figure 2 on page 9 illustrates Layer 2 TCC switching. In this example, Router A is connected by Frame Relay to Router B and Router B is connected by Fast Ethernet to Router C. Router B is a Juniper Networks router. TCC allows you to configure Router B to act as a multiservice Layer 2 switch. To do this, you configure a connection from Router A to Router C that passes through Router B. This effectively establishes Router B as a Frame Relay-to-Fast Ethernet switch and allows the router to switch frames transparently between Router A and Router C without regard to the frames' contents or the Layer 3 protocols. Router B removes the Frame Relay header from frames arriving on data-link connection identifier (DLCI) 600 and adds an Ethernet header onto the frames before sending them to the Fast Ethernet link.

Figure 2: Layer 2 TCC Switching



You can configure Layer 2 TCC switching on ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP circuits. TCC enables unlike interfaces to be connected with a single cross-connect.

To configure Layer 2 TCC switching, perform the following tasks on the router that is acting as the switch (Router B in Figure 2 on page 9):

- Defining the Encapsulation for Layer 2 TCC Switching on page 10
- Configuring ARP Functionality for Ethernet TCC Circuits on page 11
- Configuring ARP Functionality for Extended VLAN Circuits on page 12
- Configuring Static ARP For Ethernet and Extended VLAN TCCs on page 12
- Defining the Connection for Layer 2 TCC Switching on page 13
- Configuring MPLS on page 13

Defining the Encapsulation for Layer 2 TCC Switching

To begin implementation of Layer 2 TCC switching, configure TCC encapsulation on the desired interfaces of the router that is acting as the switch (Router B in Figure 2 on page 9).



NOTE: You cannot configure standard protocol families on TCC or CCC interfaces. Only the CCC family is allowed on CCC-encapsulated interfaces. Likewise, only the TCC family is allowed on TCC-encapsulated interfaces.

For ATM connections, specify the encapsulation type when configuring ATM virtual circuits (VCs) at the `[edit interfaces interface-name unit unit-number]` hierarchy level. For each VC, you configure whether it is a circuit or a regular logical interface. The default interface type is **point-to-point**.

```
[edit]
interfaces {
  at-fpc/pic/port {
    atm-options {
      vpi vpi-identifier maximum-vcs maximum-vcs;
    }
    unit logical-unit-number {
      point-to-point; # This is the default interface type.
      encapsulation (atm-tcc-vc-mux | atm-tcc-snap);
      vci vpi-identifier.vci-identifier;
    }
  }
}
```

For Cisco HDLC and PPP circuits, specify the encapsulation in the **encapsulation** statement at the `[edit interfaces interface-name]` hierarchy level. This statement configures the entire physical device. Also, you must configure the logical interface **unit 0**.

```
[edit]
```

```

interfaces {
  type-fpc/pic/port {
    encapsulation (cisco-hdlc-tcc | ppp-tcc);
    unit 0;
  }
}

```

You can specify the encapsulation for Frame Relay circuits at the **[edit interfaces *interface-name*]** hierarchy level and the **[edit interfaces *interface-name* unit *unit-number*]** hierarchy level. For TCC and CCC interfaces, the DLCI value must be configured in the range of 512 through 1022. The default interface type is **point-to-point**.

```

[edit]
interfaces {
  type-fpc/pic/port {
    encapsulation frame-relay-tcc;
    unit logical-unit-number {
      point-to-point; # This is the default interface type.
      encapsulation frame-relay-tcc;
      dlci dlci-identifier;
    }
  }
}

```

Configuring ARP Functionality for Ethernet TCC Circuits

For Ethernet TCC circuits, include the **encapsulation ethernet-tcc** statement. This statement configures the encapsulation for the entire physical interface.

To provide Address Resolution Protocol (ARP) functionality for an Ethernet-based neighbor, configure the **remote** statement at the **[edit interfaces *interface-name* unit *unit-number* family tcc]** hierarchy level and specify either the MAC address or IP address of the TCC router's Ethernet neighbor. To complete the setup of ARP, configure the **proxy** statement at the **[edit interfaces *interface-name* unit *unit-number* family tcc]** hierarchy level and specify the IP address of the TCC router's non-Ethernet neighbor.

```

[edit]
interfaces
  EthernetType-fpc/pic/port {
    encapsulation ethernet-tcc;
    unit 0 {
      family tcc {
        remote { # Addresses associated with the Ethernet TCC neighbor.
          mac-address mac-address; # Select a MAC or IP address.
          inet-address inet-address;
        }
        proxy { # Addresses belonging to the non-Ethernet TCC neighbor.
          inet-address inet-address;
        }
      }
    }
  }
}

```

Configuring ARP Functionality for Extended VLAN Circuits

For extended VLAN circuits, include the **encapsulation extended-vlan-tcc** statement. This statement configures the encapsulation for the entire physical interface.

You must also enable VLAN tagging. Ethernet interfaces in VLAN mode can have multiple logical interfaces. For encapsulation type **extended-vlan-tcc**, all VLAN IDs from 0 through 4094 are valid, up to a maximum of 1024 VLANs.

To enable ARP functionality, configure the **remote** statement at the **[edit interfaces interface-name unit unit-number family tcc]** hierarchy level with either the MAC address or IP address of your Ethernet TCC neighbor. To complete the ARP setup, configure the **proxy** statement at the **[edit interfaces interface-name unit unit-number family tcc]** hierarchy level and specify the IP address of the non-Ethernet TCC neighbor.

```
[edit]
interfaces {
  EthernetType-fpc/pic/port {
    vlan-tagging;
    encapsulation extended-vlan-tcc;
    unit 0 {
      vlan-id 600;
      family tcc {
        remote { # Addresses associated with the Ethernet TCC neighbor.
          mac-address mac-address; # Select a MAC or IP address.
          inet-address inet-address;
        }
        proxy { # Addresses belonging to the non-Ethernet TCC neighbor.
          inet-address inet-address;
        }
      }
    }
  }
}
```

Configuring Static ARP For Ethernet and Extended VLAN TCCs

If you do not use the **proxy** statement to configure ARP functionality in an Ethernet TCC or extended VLAN TCC, you must use another method to allow ARP to function. To retain the functionality of ARP for Ethernet networks, you must configure static ARP on the Ethernet neighbor. Use of static ARP assumes that you have already configured the **remote** statement on the TCC router (see “Configuring ARP Functionality for Ethernet TCC Circuits” on page 11 and “Configuring ARP Functionality for Extended VLAN Circuits” on page 12).

You configure the **arp** statement on the Ethernet neighbor at the **[edit interfaces interface-number unit unit-number family inet address ip-address]** hierarchy level. Your static ARP statement must contain the IP address of the non-Ethernet neighbor on the opposite side of the TCC router and the Ethernet interface MAC address of the TCC router. This static ARP configuration enables return path ARP functionality and complements the **remote** statement previously set on the TCC router.

In Figure 4 on page 17, you would configure an ARP statement on the **fe-0/0/0** interface of Router C. The ARP statement would contain the IP address for interface **so-0/1/0.600** on Router A and the MAC address of the **fe-1/0/0** interface of Router B.

Configure static ARP on an Ethernet neighbor at the **[edit interfaces *interface-name* unit *unit-number* family inet address *ip-address*]** hierarchy level.

```
[edit]
interfaces
  EthernetType-fpc/pic/port {
    unit 0 {
      family inet {
        address ip-address { # The local IP address.
          arp ip-address mac mac-address; # IP address of the non-Ethernet
        } # TCC neighbor and MAC address of the TCC
      } # router's Ethernet interface.
    }
  }
}
```

Defining the Connection for Layer 2 TCC Switching

The next step in configuring Layer 2 TCC switching is to define the connection between the two circuits. You configure this on the router acting as the TCC switch. When you specify the interface names, include the logical portion of the name, which corresponds to the logical unit number. The cross-connect is bidirectional, so packets received on the first interface are transmitted by the second interface, and those received on the second interface are transmitted by the first.

```
[edit]
protocols {
  connections {
    interface-switch connection-name {
      interface first-interface-name.unit-number;
      interface second-interface-name.unit-number;
    }
  }
}
```

Configuring MPLS

You must also configure MPLS for a Layer 2 cross-connect to work. The following is a minimal MPLS configuration:

```
[edit]
interfaces {
  interface-name {
    unit logical-unit-number;
  }
}
protocols {
  mpls {
    interface all;
  }
}
```

Option: Configuring ISO or MPLS Traffic on T Series, M320, and MX Series Routers

Layer 2.5 VPNs on T Series, M320, and MX Series routers support IPv4, IS-IS, and MPLS traffic types. Layer 2.5 VPNs running on M Series routers support only IPv4 traffic. Layer 2.5 VPNs do not support IPv6.

By default, IPv4 traffic runs on T Series, M320, and MX Series routers. To configure IS-IS (ISO traffic) or MPLS traffic on Layer 2.5 VPNs, you must configure both ends of the VPN with the protocol configuration.

The same type of Layer 3 traffic must be configured on both ends of a TCC connection. By default, TCC connections carry IPv4 traffic. To specify which traffic can run over a TCC interface, include the `[inet | mpls | iso]` statement at the `[edit interfaces interface-name encapsulation encapsulation-type logical-unit-number family tcc protocol]` hierarchy level:

```
[edit]
interfaces {
  interface so-2/0/0 {
    encapsulation ppp-tcc;
    unit 0 {
      family tcc {
        protocols [iso | inet | mpls];
      }
    }
  }
}
```

When enabling ISO over a Layer 2.5 VPN that is configured on a CE Ethernet interface, include the `[point-to-point]` statement at the `[edit protocols isis interface interface-name]` hierarchy level:

```
[edit]
protocols {
  isis {
    interface fe-1/0/0.0 {
      point-to-point;
    }
  }
}
```

For a full explanation of Layer 2 VPNs and configuration samples, see the *Junos VPNs Configuration Guide*.

CHAPTER 3

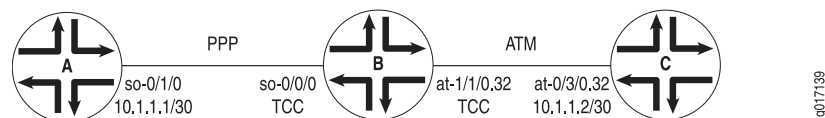
Translational Cross-Connect and Layer 2.5 VPNs Configuration Example

This section contains configuration examples and commands you can issue to verify your Layer 2 TCC switching configuration:

- Example: PPP to ATM TCC Configuration on page 15
- Example: Frame Relay to Fast Ethernet TCC Configuration on page 17
- Example: Layer 2.5 VPN Configuration on page 19
- For More Information on page 29

Example: PPP to ATM TCC Configuration

Figure 3: TCC Interface Switching—PPP to ATM



In Figure 3 on page 15, Router A uses PPP to connect with Router B, while Router C connects with Router B through ATM. Router B acts as the Layer 2 virtual switch and transparently connects Router A to Router C.

On Router A, configure basic PPP encapsulation and any desired Layer 3 protocol families on the SONET/SDH interface.

```
Router A [edit]
interfaces {
  so-0/1/0 {
    description "to Router B so-0/0/0";
    unit 0 {
      encapsulation ppp;
      family inet {
        address 10.1.1.1/30;
      }
    }
  }
}
```

Router B acts as the virtual Layer 2 switch. Here you configure the appropriate TCC encapsulations on the corresponding interfaces. In this case, **encapsulation ppp-tcc** is bound to physical interface **so-0/0/0**, and **encapsulation atm-tcc-vc-mux** is placed on VC 32 of interface **at-1/1/0**. Because the switching occurs at Layer 2, you cannot configure IP addresses or other Layer 3 family information on these interfaces.

You also need to configure MPLS and establish the cross-connect by adding the necessary interfaces to the **interface-switch** statement at the **[edit protocols connections]** hierarchy level.

```
Router B [edit]
interfaces {
  so-0/0/0 {
    description "to Router A so-0/1/0";
    encapsulation ppp-tcc;
    unit 0 {
    }
  }
  at-1/1/0 {
    description "to Router C at-0/3/0";
    atm-options {
      vpi 0 maximum-vc 2000;
    }
    unit 32 {
      vci 32;
      encapsulation atm-tcc-vc-mux;
    }
  }
}
protocols {
  mpls {
    interface so-0/0/0.0;
    interface at-1/1/0.32;
  }
  connections {
    interface-switch PPP-to-ATM {
      interface so-0/0/0.0;
      interface at-1/1/0.32;
    }
  }
}
```

On Router C, the encapsulation option used to connect to the TCC-encapsulated ATM interface on Router B is **atm-vc-mux**. Since this ATM connection is switched at Layer 2 to reach the PPP link, it is transparent to Layer 3 addressing. As a result, the IP address must be configured in the same address space as Router A's **so-0/1/0** interface.

```
Router C [edit]
interfaces {
  at-0/3/0 {
    description "to Router B at-1/1/0";
    atm-options {
      vpi 0 maximum-vc 2000;
    }
    unit 32 {
```

```

vci 32;
encapsulation atm-vc-mux;
family inet {
    address 10.1.1.2/30;
}
}
}
}

```

Verifying Your Work

To verify your TCC connection, use the **show connections** command on Router B:

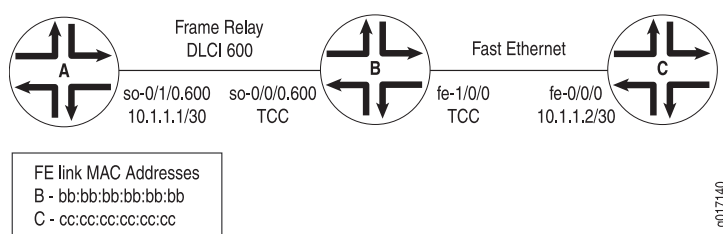
```

user@router_b> show connections
CCC and TCC connections [Link Monitoring On]
Legend for status (St)
UN -- uninitialized
NP -- not present
WE -- wrong encapsulation
DS -- disabled
Dn -- down
-> -- only outbound conn is up
<- -- only inbound conn is up
Up -- operational
Legend for connection types
if-sw: interface switching
rmt-if: remote interface switching
lsp-sw: LSP switching
Legend for circuit types
intf -- interface
tlsp -- transmit LSP
rlsp -- receive LSP
Connection/Circuit
PPP-to-ATM
at-1/1/0.32
so-0/0/0.0
Type    St    Time last up    # Up trans
if-sw   Up    Nov 30 08:57:53    1
intf    Up
intf    Up

```

Example: Frame Relay to Fast Ethernet TCC Configuration

Figure 4: TCC Interface Switching—Frame Relay to Fast Ethernet



In the configuration example in Figure 4 on page 17, Router A uses Frame Relay to connect with Router B, while Router C connects to Router B by using Fast Ethernet. Router B acts as the Layer 2 virtual switch and transparently connects Router A to Router C.

You must enable Frame Relay encapsulation on Router A at the physical interface level.

```

Router A [edit]
interfaces {
    so-0/1/0 {
        description "to Router B so-0/0/0";
        encapsulation frame-relay;
        unit 600 {
            point-to-point;
            dlci 600;
            family inet {

```

```

        address 10.1.1.1/30;
    }
}
}

```

Router B acts as the virtual switch. Enable the appropriate TCC encapsulations on the corresponding interfaces. In this case, configure the **encapsulation frame-relay-tcc** option on the logical and physical interfaces of **so-0/0/0.600**. Next, add the **ethernet-tcc** encapsulation type to the physical interface of **fe-1/0/0**. To enable ARP, configure the remote MAC address or IP address of Router C's Fast Ethernet interface with the **remote** statement at the **[edit interfaces *interface-name* unit 0 family tcc]** hierarchy level. To enable proxy ARP, include the **proxy** statement at the **[edit interfaces *interface-name* unit 0 family tcc]** hierarchy level and specify the IP address of Router A.

After configuring the correct interface encapsulations, complete your cross-connect by adding both interfaces into your MPLS configuration. Include the same interfaces in the **interface-switch** statement at the **[edit protocols connections]** hierarchy level.

```

Router B [edit]
interfaces {
  so-0/0/0 {
    description "to Router A so-0/1/0";
    dce;
    encapsulation frame-relay-tcc;
    unit 600 {
      point-to-point;
      encapsulation frame-relay-tcc;
      dlci 600;
    }
  }
  fe-1/0/0 {
    description "to Router C fe-0/0/0";
    encapsulation ethernet-tcc;
    unit 0 {
      family tcc {
        protocol inet
        remote { # Addresses associated with the Ethernet TCC neighbor Router C.
          mac-address cc:cc:cc:cc:cc:cc; # Or, specify Router C's IP address here.
        }
        proxy { # Addresses associated with the other TCC neighbor—Router A.
          inet-address 10.1.1.1;
        }
      }
    }
  }
}
protocols {
  mpls {
    interface so-0/0/0.600;
    interface fe-1/0/0.0;
  }
  connections {
    interface-switch FR-to-Ether {
      interface so-0/0/0.600;

```

```

        interface fe-1/0/0.0;
    }
}

```

Ethernet encapsulation is the default for Router C. Because the Fast Ethernet connection is switched at Layer 2 to reach the Frame Relay link, it is transparent to Layer 3 addressing. As a result, you must configure the IP address for the **fe-0/0/0** interface in the same address space as Router A's **so-0/1/0.600** interface.

Optionally, configure static ARP on the **fe-0/0/0** interface if you omit the **proxy** statement on Router B. The **arp** statement must contain the IP address from interface **so-0/1/0.600** on Router A and the MAC address of the Fast Ethernet interface on Router B.

```

Router C [edit]
interfaces
fe-0/0/0 {
  description "to Router B fe-1/0/0";
  unit 0 {
    family inet {
      address 10.1.1.2/30 {
        arp 10.1.1.1 mac bb:bb:bb:bb:bb:bb; # Configure this only if you did not
      } # enter a proxy statement on Router B.
    }
  }
}

```

Verifying Your Work

To verify the operational status of your TCC connection, use the **show connections** command on Router B:

```

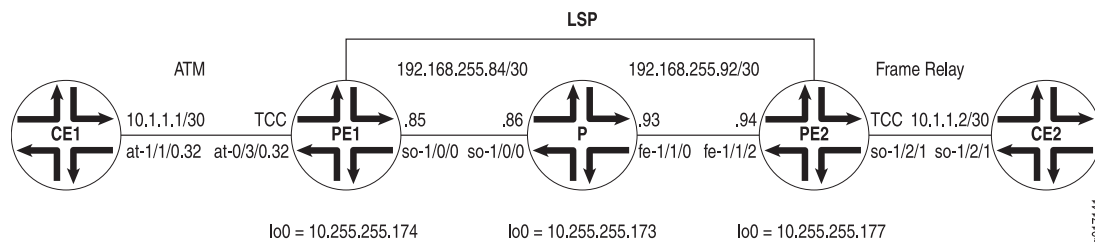
user@router_b> show connections
CCC and TCC connections [Link Monitoring On]
Legend for status (St)
UN -- uninitialized
NP -- not present
WE -- wrong encapsulation
DS -- disabled
Dn -- down
-> -- only outbound conn is up
<- -- only inbound conn is up
Up -- operational
Legend for connection types
if-sw: interface switching
rmt-if: remote interface switching
lsp-sw: LSP switching
Legend for circuit types
intf -- interface
tlsp -- transmit LSP
rlsp -- receive LSP
Connection/Circuit      Type  St  Time last up    # Up trans
FR-to-Ether             if-sw Up  Dec 30 09:57:23    1
so-0/0/0.600            intf  Up
fe-1/0/0.0               intf  Up

```

Example: Layer 2.5 VPN Configuration

This section contains a configuration example and commands you can issue to verify your Layer 2.5 VPN configuration:

Figure 5: Layer 2.5 VPN Topology Diagram



In Figure 5 on page 20, ATM is configured between CE1 and PE1 and Frame Relay is configured between PE2 and CE2. To begin the Layer 2 VPN configuration, enable ATM and the corresponding encapsulation on CE1.

```
Router CE1 [edit]
interfaces
at-1/1/0 {
  description "to PE1 at-0/3/0";
  atm-options {
    vpi 0 maximum-vcs 2000;
  }
  unit 32 {
    vci 32;
    encapsulation atm-vc-mux;
    family inet {
      address 10.1.1.1/30;
    }
  }
}
```

The first provider edge (PE1) router uses ATM TCC encapsulation on the ATM VC connecting to CE1. After this, standard Layer 2 VPN design rules apply. You use MPLS on interfaces pointing toward the core and the edge, establish a Layer 2 VPN BGP peer relationship with PE2, use LDP or Resource Reservation Protocol (RSVP) for traffic traveling through the core, and configure the proper VRF instance. Finally, you create policies for PE1 that will set a private community tag on outbound BGP traffic heading to PE2 and accept incoming traffic that matches similar community traffic from PE2.

```
Router PE1 [edit]
interfaces {
  at-0/3/0 {
    description "to CE1 at-1/1/0";
    atm-options {
      vpi 0 maximum-vcs 2000;
    }
    unit 32 {
      encapsulation atm-tcc-vc-mux;
      vci 32;
    }
  }
  so-1/0/0 {
    description "to P so-1/0/0";
    unit 0 {
      family inet {
        address 192.168.255.86/30;
      }
    }
  }
}
```

```
    }
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 10.255.255.174/32;
    }
  }
}
protocols {
  mpls {
    interface at-0/3/0.32;
    interface so-1/0/0.0;
  }
  bgp {
    group my-internal-peers {
      type internal;
      local-address 10.255.255.174;
      family l2vpn {
        signaling;
      }
      neighbor 10.255.255.177;
    }
  }
  ldp {
    interface so-1/0/0.0;
  }
}
policy-options {
  policy-statement companyA-import {
    term T1 {
      from {
        protocol bgp;
        community companyA;
      }
      then accept;
    }
    term Final {
      then reject;
    }
  }
  policy-statement companyA-export {
    term T1 {
      then {
        community add companyA;
        accept;
      }
    }
    term Final {
      then reject;
    }
  }
}
community companyA members target:100:1;
```

```
}
routing-instances {
  companyA {
    instance-type l2vpn;
    interface at-0/3/0.32;
    route-distinguisher 10.255.255.174:1;
    vrf-import companyA-import;
    vrf-export companyA-export;
    protocols {
      l2vpn {
        encapsulation-type interworking;
        site Denver {
          site-identifier 1;
          interface at-0/3/0.32 {
            remote-site-id 2;
          }
        }
      }
    }
  }
}
```

On the provider core router (P), you need only enable MPLS and LDP on the interfaces that bridge the gap between the PE routers.

```
Router P [edit]
interfaces {
  so-1/0/0 {
    description "to PE1 so-1/0/0";
    unit 0 {
      family inet {
        address 192.168.255.85/30;
      }
      family mpls;
    }
  }
  fe-1/1/0 {
    description "to PE2 fe-1/1/2";
    unit 0 {
      family inet {
        address 192.168.255.93/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.255.173/32;
      }
    }
  }
}
protocols {
  mpls {
    interface so-1/0/0.0;
```



```

    interface fe-1/1/0.0;
  }
  ldp {
    interface so-1/0/0.0;
    interface fe-1/1/0.0;
  }
}

```

The PE2 router uses Frame Relay TCC encapsulation on the Frame Relay DLCI connecting to CE2. To establish the Layer 2.5 VPN, follow the same steps you used to configure PE1. You use MPLS on interfaces pointing toward the core and the edge, establish a Layer 2 VPN BGP peer relationship with PE1, use LDP or RSVP for traffic traveling through the core, and configure the proper VRF instance. Finally, you create policies on PE2 that will set a private community tag on outbound BGP traffic heading to PE1 and accept incoming traffic that matches similar community traffic from PE1.

```

Router PE2 [edit]
interfaces {
  fe-1/1/2 {
    description "to P fe-1/1/0";
    unit 0 {
      family inet {
        address 192.168.255.94/30;
      }
      family mpls;
    }
  }
  so-1/2/1 {
    description "to CE2 so-1/2/1";
    dce;
    encapsulation frame-relay-tcc;
    unit 600 {
      encapsulation frame-relay-tcc;
      dlci 600;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.255.177/32;
      }
    }
  }
}
protocols {
  mpls {
    interface fe-1/1/2.0;
    interface so-1/2/1.600;
  }
  bgp {
    group my-internal-peers {
      type internal;
      local-address 10.255.255.177;
      family l2vpn {
        signaling;
      }
    }
  }
}

```

```
    }
    neighbor 10.255.255.174;
  }
}
ldp {
  interface fe-1/1/2.0;
}
policy-options {
  policy-statement companyA-import {
    term T1 {
      from {
        protocol bgp;
        community companyA;
      }
      then accept;
    }
    term Final {
      then reject;
    }
  }
  policy-statement companyA-export {
    term T1 {
      then {
        community add companyA;
        accept;
      }
    }
    term Final {
      then reject;
    }
  }
}
community companyA members target:100:1;
}
routing-instances {
  companyA {
    instance-type l2vpn;
    interface so-1/2/1.600;
    route-distinguisher 10.255.255.177:1;
    vrf-import companyA-import;
    vrf-export companyA-export;
    protocols {
      l2vpn {
        encapsulation-type interworking;
        site NewYork {
          site-identifier 2;
          interface so-1/2/1.600 {
            remote-site-id 1;
          }
        }
      }
    }
  }
}
}
```

To complete the Layer 2.5 VPN configuration, enable Frame Relay encapsulation on CE2.

```

Router CE2 [edit]
interfaces
so-1/2/1 {
  description "to PE2 so-1/2/1";
  encapsulation frame-relay;
  unit 600 {
    dlci 600;
    family inet {
      address 10.1.1.2/30;
    }
  }
}

```

Verifying Your Work

To verify the operational status of your Layer 2.5 VPN, use the following commands:

- **show route forwarding-table**
- **show ldp database**
- **show l2vpn connections**
- **show bgp summary**
- **show route**

To view sample output of these commands as used with the configuration example, see the following:

- Router PE1 Status on page 25
- Router PE2 Status on page 27
- Router P Status on page 28

Router PE1 Status

```
user@PE1> show route forwarding-table
```

```
<snip>
```

```
Routing table:: ccc
```

```
MPLS:
```

Interface.Label	Type	RtRef	NextHop	Type	Index	NhRef	Netif
default	perm	0		dscd	10	1	
0	user	0		recv	12	2	
1	user	0		recv	12	2	
100128	user	0		Pop			so-1/0/0.0
100128(S=0)	user	0		Pop			so-1/0/0.0
100129	user	0		Swap	100000		so-1/0/0.0
800001	user	0		ucst	137	1	at-0/3/0.32
at-0/3/0. (CCC)	user	0		indr	133	2	
				Push	800000		Push 100000(top)

```
so-1/0/0.0
```

```
<snip>
```

```
user@PE1> show ldp database
```

```
Input label database, 10.255.255.174:0-10.255.255.173:0
```

Label	Prefix
100002	10.255.255.174/32
100000	10.255.255.177/32

```

      3      10.255.255.173/32
Output label database, 10.255.255.174:0-10.255.255.173:0
Label      Prefix
100128     10.255.255.173/32
100129     10.255.255.177/32
      3      10.255.255.174/32

```

user@PE1> show l2vpn connections

L2VPN Connections:

Legend for connection status (St) Legend for interface status

OR -- out of range

up -- operational

EI -- encapsulation invalid

Dn -- down

EM -- encapsulation mismatch

NP -- no present

CN -- circuit not present

DS -- disabled

OL -- no outgoing label

WE -- wrong encapsulation

Dn -- down

UN -- uninitialized

VC-Dn -- Virtual circuit down

WE -- intf encaps != instance encaps

-> -- only outbound conn is up

<- -- only inbound conn is up

UP -- operational

XX -- unknown

Instance: companyA

Local site: Denver (1)

connection-site	Type	St	Time last up	# Up trans
2	rmt	Up	Nov 30 08:21:07 2001	1

Local interface: at-0/3/0.32, Status: Up, Encapsulation: INTERWORKING

Remote PE: 10.255.255.177

Incoming label: 800001, Outgoing label: 800000

user@PE1> show bgp summary

Groups: 1 Peers: 1 Down peers: 0

Table	Tot Paths	Act Paths	Suppressed	History	Damp	State	Pending
inet.0	0	0	0	0	0	0	0
bgp.l2vpn.0	1	1	0	0	0	0	0

Peer	AS	InPkt	OutPkt	OutQ	Flaps	Last
------	----	-------	--------	------	-------	------

Up/DwnState|#Active/Received/Damped...

10.255.255.177	69	49	45	0	1	19:16 Estab1
----------------	----	----	----	---	---	--------------

bgp.l2vpn.0: 1/1/0

companyA.l2vpn.0: 1/1/0

user@PE1> show route

<snip>

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

0 *[MPLS/0] 1d 18:54:24, metric 1

Receive

1 *[MPLS/0] 1d 18:54:24, metric 1

Receive

100128 *[LDP/9] 00:24:03, metric 1

> via so-1/0/0.0, Pop

100128(S=0) *[LDP/9] 00:24:03, metric 1

> via so-1/0/0.0, Pop

100129 *[LDP/9] 00:24:03, metric 1

> via so-1/0/0.0, Swap 100000

800001 *[L2VPN/7] 00:10:35

> via at-0/3/0.32, Pop [0]

at-0/3/0.32 *[L2VPN/7] 00:10:35

> via so-1/0/0.0, Push 800000, Push 100000(top)

```

companyA.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:1:1          /96
                  *[L2VPN/7] 00:19:55
                  Discard
1:1:2:1          /96
                  *[BGP/170] 00:06:46, localpref 100, from 10.255.255.177
                  AS path: I
                  > via so-1/0/0.0, Push 100000
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:2:1          /96
                  *[BGP/170] 00:10:35, localpref 100, from 10.255.255.177
                  AS path: I
                  > via so-1/0/0.0, Push 100000

<snip>

```

Router PE2 Status

```
user@vpn07> show route forwarding-table
```

```
<snip>
```

```
Routing table:: ccc
```

```
MPLS:
```

Interface.Label	Type	RtRef	Nexthop	Type	Index	NhRef	Netif
default	perm	0		dscd	8	1	
0	user	0		recv	10	2	
1	user	0		recv	10	2	
100002	user	0		Pop			fe-1/1/2.0
100002(S=0)	user	0		Pop			fe-1/1/2.0
100003	user	0		Swap	100002		fe-1/1/2.0
800000	user	0		ucst	60	1	so-1/2/1.0
so-1/2/1. (CCC)	user	0		indr	59	2	

```

<snip>

```

```
user@vpn07> show ldp database
```

```
Input label database, 10.255.255.177:0-10.255.255.173:0
```

Label	Prefix
100000	10.255.255.177/32
3	10.255.255.173/32
100002	10.255.255.174/32

```
Output label database, 10.255.255.177:0-10.255.255.173:0
```

Label	Prefix
100002	10.255.255.173/32
3	10.255.255.177/32
100003	10.255.255.174/32

```
user@vpn07> show l2vpn connections
```

```
L2VPN Connections:
```

```
Legend for connection status (St)
```

```
OR -- out of range
```

```
EI -- encapsulation invalid
```

```
EM -- encapsulation mismatch
```

```
CN -- circuit not present
```

```
OL -- no outgoing label
```

```
Dn -- down
```

```
VC-Dn -- Virtual circuit down
```

```
WE -- intf encaps != instance encaps
```

```
-> -- only outbound conn is up
```

```
<- -- only inbound conn is up
```

```
UP -- operational
```

```
Legend for interface status
```

```
up -- operational
```

```
Dn -- down
```

```
NP -- no present
```

```
DS -- disabled
```

```
WE -- wrong encapsulation
```

```
UN -- uninitialized
```

```

XX -- unknown
Instance: companyA
Local site: NewYork (2)
  connection-site      Type  St      Time last up      # Up trans
  1                    rmt   Up      Nov 30 08:21:01 2001      1
    Local interface: so-1/2/1.0, Status: Up, Encapsulation: INTERWORKING
    Remote PE: 10.255.255.174
    Incoming label: 800000, Outgoing label: 800001

user@vpn07> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table      Tot Paths  Act Paths Suppressed  History  Damp State  Pending
bgp.l2vpn.0      1          1          0          0          0          0
inet.0           0          0          0          0          0          0
Peer          AS      InPkt  OutPkt  OutQ   Flaps Last
Up/DwnState|#Active/Received/Damped...
10.255.255.174  69      45      52      0      0      20:20 Estab1
  bgp.l2vpn.0: 1/1/0
  companyA.l2vpn.0: 1/1/0

user@vpn07> show route
<snip>
mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0          *[MPLS/0] 02:34:04, metric 1
  Receive
1          *[MPLS/0] 02:34:04, metric 1
  Receive
100002      *[LDP/9] 00:25:39, metric 1
  > via fe-1/1/2.0, Pop
100002(S=0) *[LDP/9] 00:25:39, metric 1
  > via fe-1/1/2.0, Pop
100003      *[LDP/9] 00:25:01, metric 1
  > via fe-1/1/2.0, Swap 100002
800000      *[L2VPN/7] 00:07:50
  > via so-1/2/1.0, Pop      [0]
so-1/2/1.0  *[L2VPN/7] 00:07:50
  > via fe-1/1/2.0, Push 800001, Push 100002(top)
companyA.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:1:1      /96
  *[BGP/170] 00:04:59, localpref 100, from 10.255.255.174
    AS path: I
  > via fe-1/1/2.0, Push 100002
1:1:2:1      /96
  *[L2VPN/7] 00:11:34
    Discard
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:1:1      /96
  *[BGP/170] 00:11:38, localpref 100, from 10.255.255.174
    AS path: I
  > via fe-1/1/2.0, Push 100002
<snip>

```

Router P Status

```
user@P> show ldp database
```

```
Input label database, 10.255.255.173:0-10.255.255.174:0
  Label Prefix
100128 10.255.255.173/32
100129 10.255.255.177/32
   3   10.255.255.174/32
Output label database, 10.255.255.173:0-10.255.255.174:0
  Label Prefix
   3   10.255.255.173/32
100000 10.255.255.177/32
100002 10.255.255.174/32
Input label database, 10.255.255.173:0-10.255.255.177:0
  Label Prefix
   3   10.255.255.177/32
100002 10.255.255.173/32
100003 10.255.255.174/32
Output label database, 10.255.255.173:0-10.255.255.177:0
  Label Prefix
   3   10.255.255.173/32
100000 10.255.255.177/32
100002 10.255.255.174/32
```

For More Information

For additional information about TCC or Layer 2.5 VPNs, see the following documents:

- *Junos VPNs Configuration Guide*
- *Junos MPLS Applications Configuration Guide*
- *Junos Network Interfaces Configuration Guide*

PART 2

Index

- Index on page 33

Index

A

ATM encapsulation	
Layer 2 switching cross-connects.....	10

C

Cisco HDLC encapsulation	
Layer 2 switching cross-connect.....	10
connections statement	
usage guidelines.....	13

E

encapsulation	
TCC.....	10
encapsulation statement	
Layer 2 switching cross-connect.....	10

F

Frame Relay encapsulation	
Layer 2 switching cross-connect.....	11

L

Layer 2 switching	
TCC	
configuration procedure.....	9
example configuration.....	15, 17
overview.....	3
system requirements.....	4
Layer 2 switching cross-connect	
MPLS.....	13
TCC connections.....	13
TCC encapsulation.....	10
Layer 2.5 VPNs	
TCC	
configuration procedure.....	5
example configuration.....	19
operational mode commands.....	25

M

MPLS	
Layer 2 switching cross-connect.....	13

S

system requirements	
TCC.....	4

T

TCC	
configuration procedure.....	9
encapsulation.....	6
example configuration	
Frame Relay to Fast Ethernet.....	17
PPP to ATM.....	15
Layer 2.5 VPNs	
configuration procedure.....	5
example configuration.....	19
operational mode commands.....	25
operational mode commands	
Frame Relay to Fast Ethernet.....	19
PPP to ATM.....	17
options	
static ARP.....	12
overview.....	3
system requirements.....	4

