Configuring SMDS 20

This chapter describes the Switched Multimegabit Data Service (SMDS) tunneling functionality in the ERX system.

Overview

SMDS is a wide area networking service designed for LAN interconnection. SMDS is a connectionless service. An SMDS network is composed of:

- A series of SMDS switches inside a service provider’s network
- A series of channel service units/data service units (CSUs/DSUs) that connect subscribers to the network
- Routers and gateways to connect to each CSU/DSU

Terms

Table 20-1 defines terms and abbreviations that are used in this discussion of SMDS.

Table 20-1 SMDS terms and abbreviations

Term	Definition
CBF	connection-based forwarding. A method of forwarding frames in which forwarding decisions are made using only the identity of the ingress interface. No part of a packet's contents is used to determine how a packet should be forwarded.
GRE	generic routing encapsulation. In this application, GRE encapsulates SMDS packets to enable data transmission through an IP tunnel. The resulting encapsulated packet contains a GRE header and a delivery header.
HSSI	high-speed serial interface. SMDS runs only over HSSI interfaces.
SMDS	Switched Multimegabit Data Service

Application

The ERX system's SMDS implementation addresses an application in which the ERX system offloads the central express ring in an SMDS network. Figure 20-1 shows an SMDS network before ERX systems are added.

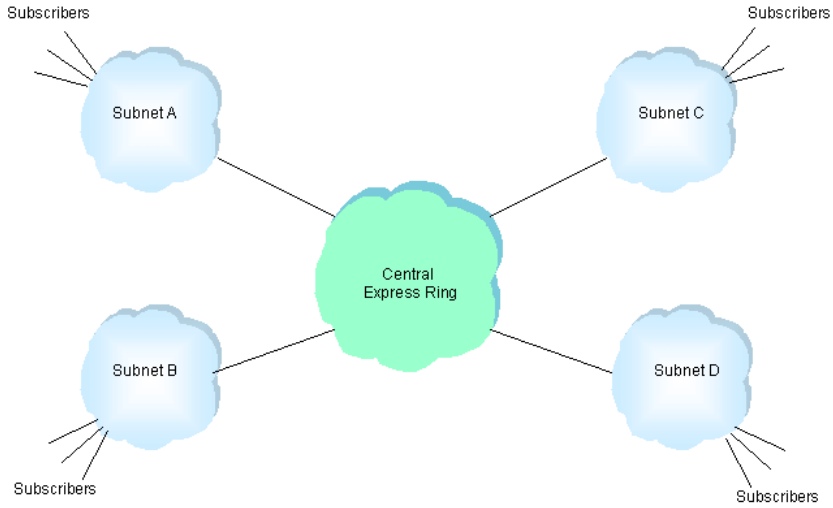

Figure 20-1 SMDS network with central express ring and subnets

As shown in Figure 20-2, the ERX system provides a mesh of GRE tunnel connections as a means of offloading the central express ring.

Figure 20-2 ERX system with GRE tunnels to offload the central express ring

To connect the GRE tunnels to each of the subnets, the ERX system connects to SMDS switches using the 3-port HSSI line module.

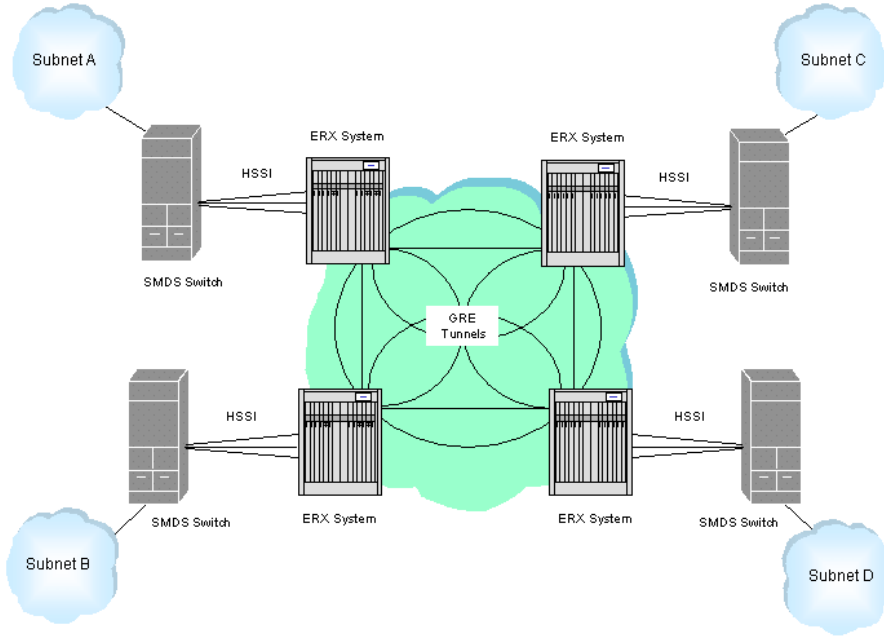

Figure 20-3 HSSI line modules in ERX systems providing connections to SMDS switches

SMDS Components

The following components support the SMDS application:

- HSSI line module – provides the physical interface to connect to the SMDS switch.
- GRE tunnels – transport SMDS traffic among ERX systems.
- SMDS trunk interface – runs over the HSSI line module and GRE tunnels.
- Connection-based forwarding (CBF) interfaces and connections – forward SMDS traffic among SMDS trunk interfaces.

Figure 20-4 shows how these components make up the interface columns for the SMDS application.

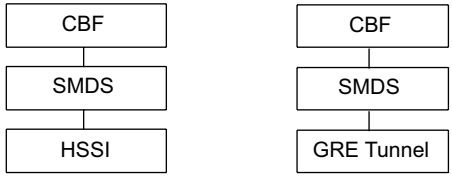

Figure 20-4 Interface columns for the SMDS application

Two sets of interfaces are required for SMDS:

- The first set provides the connection between the ERX system and the SMDS switch. It has a physical HSSI line module on which you configure an SMDS trunk interface and a CBF interface.
- The second set provides the connection between ERX systems. It has a GRE tunnel, an SMDS trunk interface, and a CBF interface.

When you configure SMDS, you set up a CBF connection that connects the two interface columns (see Figure 20-5).

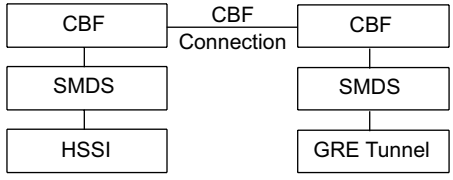

Figure 20-5 A CBF connection connecting the SMDS interface columns

The next sections describe each of the SMDS components in more detail.

HSSI Line Module

The ERX HSSI line module supports the SMDS trunk interface. This connection supports speeds of 34.368 or 44.736 Mbps, which you can select using the CLI.

GRE Tunnels

To support the application of offloading the express ring, you configure a mesh of GRE tunnels between ERX systems. These tunnels do not provide demultiplexing and will carry traffic only from a specific SMDS interface. Thus, they can be viewed as extensions of the ingress physical HSSI interface. All SMDS traffic received on a HSSI must be mapped one-to-one with a specific GRE tunnel.

GRE does not have a protocol ID type for an SMDS payload. Therefore, every GRE tunnel carrying SMDS traffic must terminate at an ERX system. Although these tunnels do not need to be marked as SMDS, the peer connection guarantees that header parsing does not take place.

The GRE tunnel can run over any layer 2 medium such as ATM, PPP, and Frame Relay.

For more information on GRE tunnels, see *ERX Routing Protocols Configuration Guide, Vol. 1, Chapter 4, Configuring IP Tunnels*.

Layer Two Connection-Based Forwarding

CBF provides a simple mechanism for forwarding non-IP traffic directly between ERX interfaces. CBF is a method of forwarding frames in which CBF makes forwarding decisions using only the identity of the ingress interface. In particular, no part of a packet's contents is used to determine how a packet should be forwarded.

CBF Interfaces

Connection-based forwarding uses dedicated layer 2 interfaces, called CBF interfaces, which are configured at the top of interface columns. CBF interfaces used in connection-based forwarding are analogous to IP interfaces used in IP routing.

A CBF interface runs over the SMDS trunk interface. Only one CBF interface is permitted per SMDS interface.

CBF Connections

CBF forwards frames along connections configured between pairs of interfaces. Frames received on a connection's ingress interface are forwarded directly to that connection's egress interface. The packet is transmitted from ingress to egress forwarding interfaces as the raw layer 2 payload received from the layer 2 interface. Both the ingress and egress interfaces are SMDS encapsulated and have CBF interfaces associated with them. Once the connection between the ingress and egress interfaces is made, traffic begins to flow.

SMDS Trunk Interface

The SMDS trunk interface supports SMDS framed in standard HDLC. This interface is like other layer 2 interfaces, such as PPP, Frame Relay, and ATM, with the exception that the system does not de-encapsulate the packet and route the traffic as it does for the other layer 2 interfaces.

The ERX system does not examine the SMDS payload, and the SMDS payload is not required to be IP, because the SMDS header stays intact. The entire SMDS frame, including the internal header provided by the SMDS switch, is transported over a GRE tunnel to a destination SMDS switch. The SMDS trunk interface supports an MTU of up to 9188 octets.

A HSSI maps one-to-one with an SMDS logical interface. The data received on an SMDS interface is forwarded to the destination interface in the ERX system, which has an associated GRE tunnel defined. If a CBF interface does not exist for the SMDS interface, all received traffic is dropped. In addition, the connection must be to another HSSI or a GRE-tunneled interface and never to a purely routed interface.

The SMDS trunk interface supports only connection-based forwarding to another SMDS trunk-encapsulated interface; that is, a GRE tunnel.

Scalability and Performance

Because the number of SMDS interfaces is limited by the number of physical ports, there is no restriction on the number of SMDS interfaces supported on the system (you can configure up to 12 HSSI cards on an ERX-1410 system). SMDS termination and connection-based forwarding to GRE tunnels should run at wire rate on the ERX system. A typical configuration might consist of redundant SRPs, 10 three-port HSSI line modules, one GRE tunnel interface, and a quad-port OC3 or OC12 ATM line module.

Requirements

The ERX system SMDS implementation has the following requirements and restrictions:

- The SMDS switch provides SMDS address screening and source address validation.
- The SMDS switch provides all SMDS group address elaboration.
- The SMDS switch will not fragment SMDS frames sent to ERX system interfaces.
- All SMDS traffic received on an ERX physical interface is mapped one-to-one with a specific GRE tunnel.
- Because GRE does not have a protocol ID for SMDS, both ends of GRE tunnels must be ERX systems so that the SMDS payload can be understood.
- You cannot place an IP interface on an SMDS trunk interface.

Configuring SMDS

This section shows how to set up ERX system B in the sample configuration in Figure 20-6. In the example, the GRE tunnel runs over an OC3 ATM interface.

Figure 20-6 Sample SMDS configuration

ATM Interface

To configure an ATM interface:

1 Specify an ATM interface, and set the clock source.

```
host1(config)#interface atm 10/0
host1(config-if)#atm clock internal
```

2 Configure a subinterface on the ATM interface.

```
host1(config)#interface atm 10/0.1
```

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-subif)#atm pvc 1 0 1 aal5snap
```

4 Assign an IP address to the subinterface.

```
host1(config-subif)#ip address 160.1.0.1
```

GRE Tunnel and CBF Interface

To configure a GRE tunnel and CBF interface:

1 Create and configure a GRE tunnel interface called newport1.

```
host1(config)#interface tunnel gre:newport1
```

2 Assign a source address for the GRE tunnel. The tunnel source address must match an address configured on the system where the GRE tunnel originates.

```
host1(config-if)#tunnel source 160.1.0.1
```

3 Assign a destination address for the GRE tunnel. The tunnel destination address must match an address configured on the system where the tunnel terminates.

```
host1(config-if)#tunnel destination 160.1.0.2
```

4 Add SMDS trunk encapsulation over the GRE tunnel interface.

```
host1(config-if)#encapsulation smds-trunk
```

5 Create a CBF interface over the GRE tunnel interface.

```
host1(config-if)#cbf
```

HSSI and CBF Interface

To configure a HSSI and a CBF interface:

1 Specify a HSSI.

```
host1(config)#interface hssi 1/0
```

2 Set the HSSI to CRC-32 mode.

```
host1(config-if)#crc 32
```

3 Specify the HSSI DCE clock rate of 34 MHz.

```
host1(config-if)#clock rate 34
```

4 Set the HSSI to DCE mode by setting clocking to internal.

```
host1(config-if)#hssi internal-clock
```

Note: *If the SMDS switch acts as the DTE, you must set up the ERX system to be the DCE.*

5 Add SMDS trunk encapsulation over the HSSI.

```
host1config-if)#encapsulation smds-trunk
```

6 Create a CBF interface over the HSSI.

```
host1(config-if)#cbf
```

Note: *For information on HSSI commands, see Chapter 7, Configuring HSSIs*.

CBF Connection

Create a CBF connection between the HSSI in slot 1, port 0 and the GRE tunnel, newport1. All SMDS traffic received on a HSSI must be mapped one-to-one with a specific GRE tunnel. The CBF connection provides that mapping.

```
host1(config)#cbf connection hssi 1/0 tunnel gre:newport1
```

atm clock internal

- Use to cause the ATM interface to generate the transmit clock internally.
- You can add one of the following keywords:
  - **module** – internal clock is from the line module
  - **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.

atm pvc

- Use to configure a PVC on the ATM interface.
- You must include each of the following parameters:
  - *vcd* – unique number that you assign to identify a virtual circuit. The range is 1–4294967293.
  - *vpi* – 8-bit field in the ATM cell header. The VPI value must match the value of the VPI on the switch. You cannot set both the VPI and VCI to 0; if one is 0, the other cannot be 0.
  - *vci* – 16-bit field in the ATM cell header. You cannot set both the VPI and VCI to 0; if one is 0, the other cannot be 0.
  - *encapsulation type*:
    - **aal5snap** – specifies a logical link control (LLC) encapsulated circuit. An LLC/Subnetwork Access Protocol (LLC/SNAP) precedes the protocol datagram.
    - **aal5mux ip** – specifies a multiplexed circuit used for IP only.
    - **aal5autoconfig** – enables the autodetection of a 1483 encapsulation (LLC/SNAP or VC multiplexed). See *Chapter 21, Configuring Dynamic Interfaces* for more explanation.
- Example

```
host1(config-subif)#atm pvc 1 0 1 aal5snap
```

- Use the **no** version to remove the specified PVC

cbf

- Use to create a CBF interface over the SMDS interface that you are configuring.
- Example

```
host1(config-if)#cbf
```

- Use the **no** version to remove the CBF interface.

cbf connection

- Use to create a bidirectional CBF connection between two CBF interfaces.
- Note that the bidirectional connection consists of a symmetrical pair of unidirectional connections, one from *interface1* to *interface2* and a second from *interface2* to *interface1*.
- Example – the following example creates a CBF connection between the HSSI in slot 1, port 0 and the GRE tunnel newport1.

```
host1config)#cbf connection hssi 1/0 tunnel gre:newport1
```

- Use the **no** version to remove an existing connection.

cbf shutdown

- Use to administratively disable a CBF interface.
- Example

```
host1(config-if)#cbf shutdown
```

- Use the **no** version to enable a CBF interface. (Newly created CBF interfaces are enabled by default.)

encapsulation smds-trunk

- Use to configure SMDS as the encapsulation method on an interface.
- Example

```
host1(config-if)#encapsulation smds-trunk
```

- Use the **no** version to disable SMDS encapsulation on an interface.

interface atm

- Use to configure an ATM interface or subinterface type in the *slot/port.subinterface* format:
  - *slot* – number of the slot in which the line module resides
  - *port* – port number on the I/O module
  - *subinterface* – number of the subinterface in the range 1–4294967293
- Example

```
host1(config)#interface atm 10/0
```

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

interface tunnel

- Use to create a GRE IP tunnel interface.
- Specify the type of tunnel as **gre**, and assign a name to the tunnel.
- Example

```
host1(config)#interface tunnel gre:boston-tunnel-1
```

- Use the **no** version to remove the tunnel.

tunnel destination

- Use to configure the remote end of the tunnel.
- Specify either the IP address of an interface on the remote system or the hostname of the remote system.
  - The IP address is the address for the destination interface.
  - The hostname is the name of the destination interface.
- Example 1

```
host1(config)#interface tunnel gre:tunnel2
host1(config-if)#tunnel destination 192.13.7.1
```

- Example 2

```
host1(config)#interface tunnel gre:tunnel2
host1(config-if)#tunnel destination remoteHost
```

- Use the **no** version to remove the destination of a tunnel.

tunnel source

- Use to configure the source of the tunnel.
- Specify either the primary IP address or the type and number of a new interface.
- Do not specify an unnumbered interface.
- Example 1

```
host1(config)#interface tunnel gre:boston-tunnel-1
host1(config-if)#tunnel source 192.10.2.1
```

- Example 2

```
host1(config)#interface tunnel gre:boston-tunnel-1
host1(config-if)#tunnel source atm 5/0.12
```

- Use the **no** version to remove the source of a tunnel.

Monitoring SMDS

Use the versions of the **show smds interface** command described in this section to monitor SMDS interfaces.

You can set a statistics baseline for SMDS interfaces using the **baseline smds** command. To display baselined statistics, use the **delta** keyword with SMDS **show** commands.

Output Filtering

You can include or exclude lines of output based on a text string you specify. Use the output filtering feature of the **show** command. See *show Commands* in *ERX System Basics Configuration Guide, Chapter 2, Command Line Interface*, for details.

baseline smds interface

- Use to set a baseline for SMDS statistics on an interface.
- The system 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.
- Use the **delta** keyword with SMDS **show** commands to specify that baselined statistics are to be shown.

- Example

```
host1#baseline smds interface hssi 4/0
```

- There is no **no** version.

show smds interface

- Use to display the status and statistical information of SMDS interfaces.
- Field descriptions
  - SMDS interface – interface type, interface specifier, and status: up, down
  - Interface characteristics – Link Status Trap Enable: true, false
  - Baseline status – indicates whether a statistics baseline is set
  - 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)
- Example 1

```
host1#show smds interface
SMDS interface hssi 4/0 is up
SMDS interface hssi 4/1 is up
```

- Example 2

```
host1#show smds interface full
SMDS interface hssi 4/0 is up
Interface characteristics
   Link Status Trap Enable        true
No baseline has been set
Interface statistics              in                  out
   packets                        0                   0
   octets                         0                   0
   errors                         0                   0
   discards                       0                   0
SMDS interface hssi 4/1 is up
Interface characteristics
   Link Status Trap Enable        false
No baseline has been set
Interface statistics              in                  out
   packets                        0                   0
   octets                         0                   0
   errors                         0                   0
   discards                       0                   0
```

show smds interface statistics

- Use to show statistics for SMDS interfaces.
- Field descriptions
  - › Configured network protocol – network protocol configured on the interface
  - › Baseline status – indicates whether a statistics baseline is set
  - › 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)
- Example

```
host1#show smds interface statistics
SMDS interface hssi 4/0 is up
No baseline has been set
Interface statistics              in                  out
   packets                        0                   0
   octets                         0                   0
   errors                         0                   0
   discards                       0                   0
SMDS interface hssi 4/1 is up
No baseline has been set
Interface statistics              in                  out
   packets                        0                   0
   octets                         0                   0
   errors                         0                   0
   discards                       0                   0
```

Monitoring CBF Interfaces

You can set a statistics baseline for CBF interfaces using the **baseline cbf interface** command, and you can clear statistics with the **clear cbf interface** command. To display baselined statistics, use the **delta** keyword with CBF **show** commands.

Output Filtering

To include or exclude lines of output based on a text string that you specify, use the filtering feature of the **show** command. See *show Commands* in *ERX System Basics Configuration Guide, Chapter 2, Command Line Interface*, for details.

baseline cbf interface

- Use to establish a baseline for statistics associated with a CBF interface.
- The system 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.
- Use the **delta** keyword with CBF **show** commands to specify that baselined statistics are to be shown.
- Example

```
host1#baseline cbf interface 2/0
```

- There is no **no** version.

clear cbf interface

- Use to clear statistics associated with a CBF interface.
- Example

```
host1#clear cbf interface 2/0
```

- There is no **no** version.

show cbf connection

- Use to display all configured CBF connections.
- Example

```
host1#show cbf connection
   hssi 4/0 <--> hssi 4/1
```

show cbf interface

- Use to display information about configured CBF interfaces.
- Field descriptions
  - › Interface specifier and status
    - Lower interface type – type of the lower interface in the interface column
    - Administrative status – interface is administratively enabled (Up), which means that the **no cbf shutdown** command is operational; interface is administratively disabled (Down), which means that the **cbf shutdown** command is operational
    - Operational status – operational status of the interface: up, down, not present, lower layer down, local status, path status
  - › Interface statistics
    - Received bytes – number of bytes received from the lower layer
    - Received packets – number of packets received and forwarded to the egress interface
    - Receive discards – number of packets discarded during receive
    - Receive errors – number of packet errors detected during receive
    - Transmitted bytes – number of bytes transmitted to the lower layer
    - Transmitted packets – number of packets forwarded from an ingress interface and transmitted

  - Transmit discards – number of packets discarded during transmit
  - Transmit errors – number of packet errors detected during transmit

- Example 1

```
host1#show cbf interface
hssi 4/0 is up
   Lower interface type:   Smds
   Administrative status:  Up
   Operational status:     Up
hssi 4/1 is up
   Lower interface type:   Smds
   Administrative status:  Up
   Operational status:     Up
```

- Example 2

```
host1#show cbf interface full
hssi 4/0 is up
   Lower interface type:   Smds
   Administrative status:  Up
   Operational status:     Up
   Statistics:
     Received bytes:       0
     Received packets:     0
     Receive discards:     0
     Receive errors:       0
     Transmitted bytes:    0
     Transmitted packets:  0
     Transmit discards:    0
     Transmit errors:      0
hssi 4/1 is up
   Lower interface type:   Smds
   Administrative status:  Up
   Operational status:     Up
   Statistics:
     Received bytes:       0
     Received packets:     0
     Receive discards:     0
     Receive errors:       0
     Transmitted bytes:    0
     Transmitted packets:  0
     Transmit discards:    0
     Transmit errors:      0
```