Nortel Multiservice Switch 7400/15000/20000

ATM Configuration Management

NN10600-710

Document status: Standard
Document issue: 7.2S1
Document date: March 2006
Product release: PCR7.2 and up
Job function: Operations
Type: NTP
Language type: U.S. English

Contents

What's new

The following features were added to this document:

- Load Re-Balancing on Parallel Links (page 7)

Other changes made to this document include the following:

- The procedures Configuring connection administrator (page 47), Configuring traffic management for a VPT (page 74), and Configuring traffic management (page 109) were updated to include additional information about the *holdingPriority* attribute.
- Updated Configuring connection and buffer space for ATM IP FPs (page 22) with a new maximum value for configured connection resources.

Attention: To ensure that you are using the most current version of an NTP, check the current NTP list in NN10600-000 *Nortel Multiservice Switch 7400/15000/20000 What's New*.

Load Re-Balancing on Parallel Links

The following section was added for this feature:

- Configuring Load Re-Balancing on Parallel Links (page 132)

ATM configuration

Configure asynchronous transfer mode (ATM) to add and set features and connections.

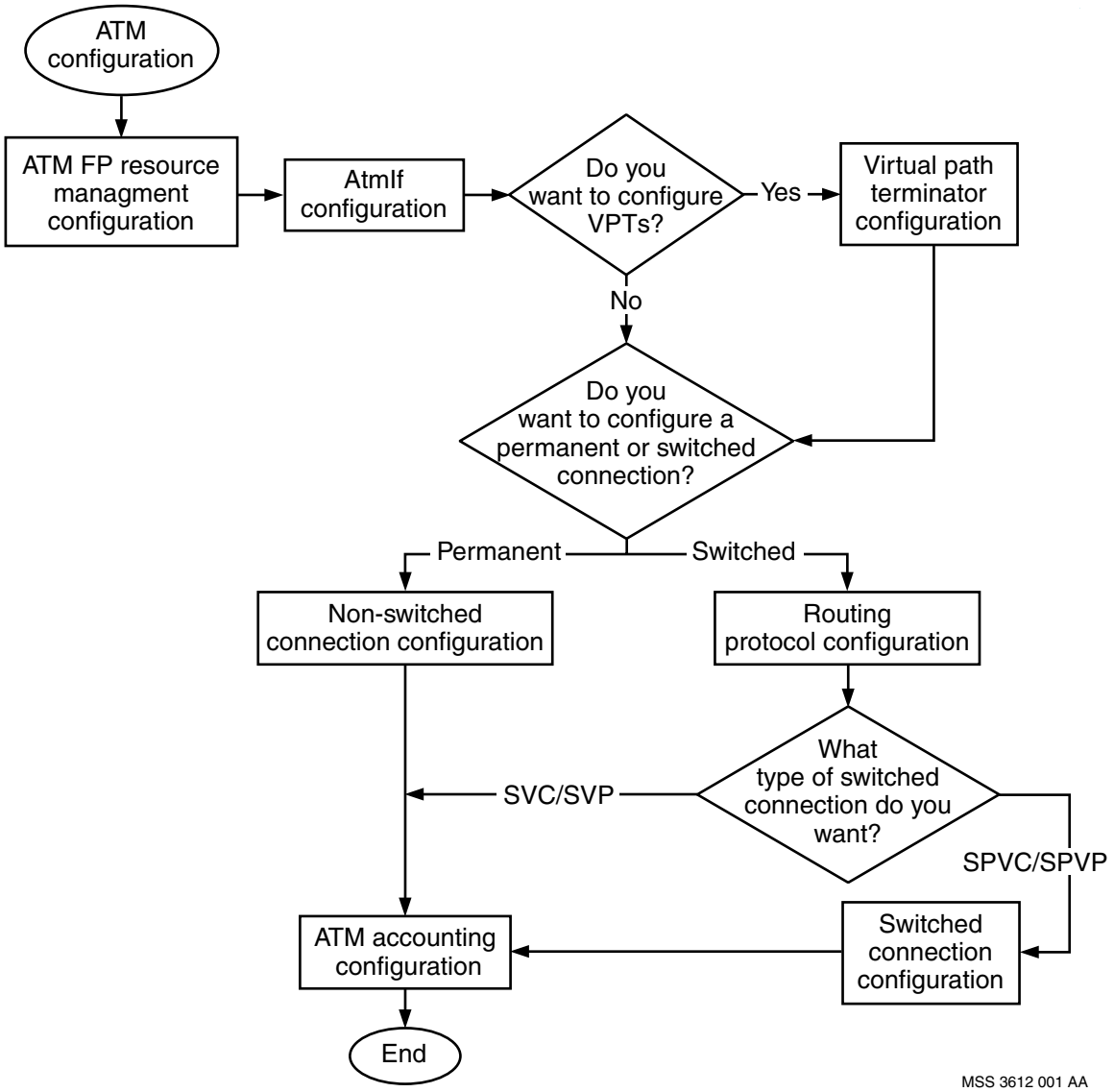
Prerequisites to ATM configuration

- The supporting network infrastructure, including Nortel Multiservice Switch nodes, must be installed and configured.
- If you are unfamiliar with ATM concepts and procedures, see NN10600-700 *Nortel Multiservice Switch 7400/15000/20000 ATM Technology Fundamentals*.

ATM configuration tasks

This workflow shows you the sequence of tasks you perform to configure ATM. To link to any task, go to ATM configuration task navigation (page 9).

ATM configuration tasks



ATM configuration task navigation

- ATM FP resource management configuration (page 11)
- AtmIf configuration (page 32)
- Virtual path terminator configuration (page 66)
- Non-switched connection configuration (page 100)
- Routing protocol configuration (page 118)
- Switched connection configuration (page 161)

- ATM accounting configuration (page 179)

ATM FP resource management configuration

Configure ATM FP resource management to change the default configuration for queue and memory resources and to monitor and control common hardware resources on a function processor (FP) supporting ATM services.

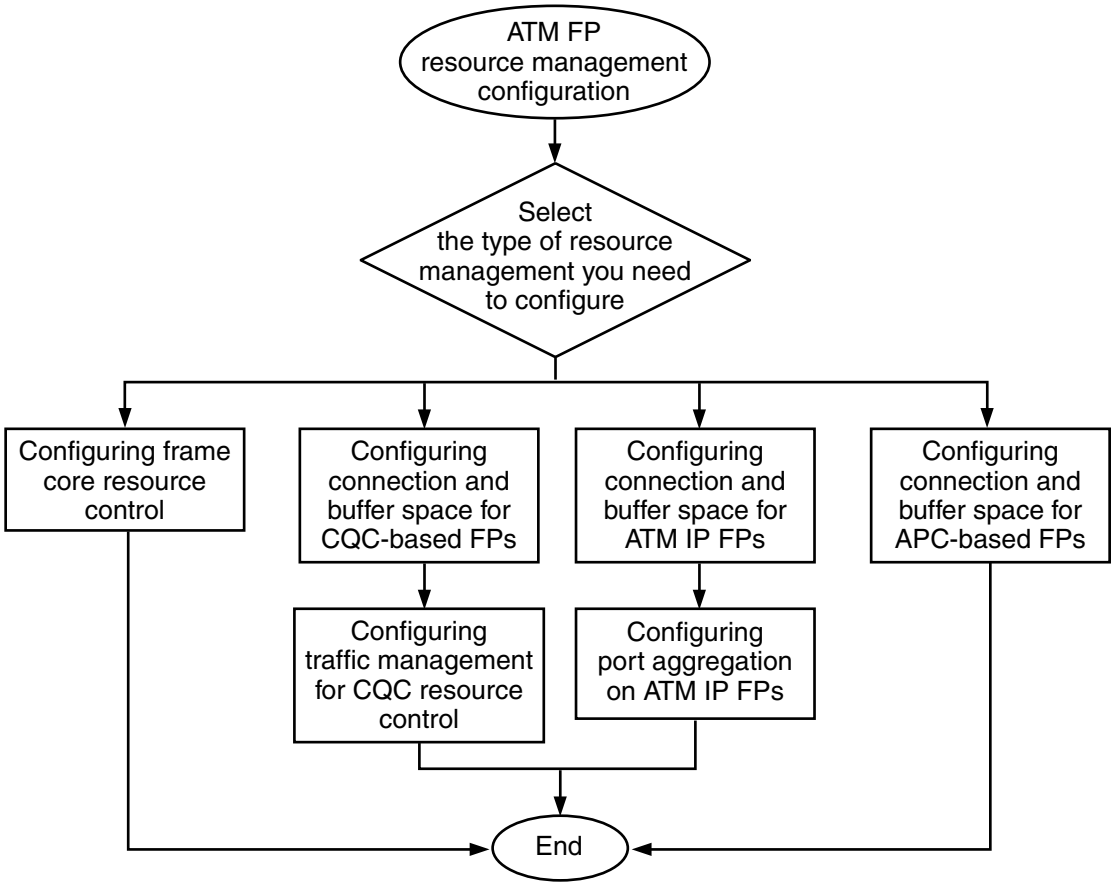
Prerequisites to ATM FP resource management configuration

- See NN10600-707 *Nortel Multiservice Switch 7400/15000/20000 ATM Queuing and Scheduling Fundamentals* for more detailed information.

ATM FP resource management configuration procedures

This task flow shows you the sequence of procedures you perform to configure ATM FP resource management. To link to any procedure, go to ATM FP resource management procedure navigation (page 12).

ATM FP resource management configuration procedures



MSS 3612 002 AA

ATM FP resource management procedure navigation

Configuring frame core resource control

Configure frame core resource control (FCRC) to control the use of frame-specific resources on a logical processor that interworks ATM services with frame services. You can change the size of the subconnection pool for connection-oriented services or change the size of the logical network number (LNN) connection pool for connectionless services.

Procedure steps

Step	Action
1	Add the frame core resource control override component. **`add Lp/<lp> Eng Fcrc Ov`**
2	Set the number of sub-connection resource records for each LP. **`set Lp/<lp> Eng Fcrc Ov subConnectionPoolCapacity <subConnCap>`**
3	Set the number of LNN connection resource records for each LP. Frame and cell queues also compete for this memory. **`set Lp/<lp> Eng Fcrc Ov lnnConnectionPoolCapacity <lnnConnCap>`**
4	Add the queue controller component. The system adds the *Override* component automatically. **`add Lp/<lp> Eng Fcrc Pqc`**
5	Set the maximum number of IP routes permitted in the ATM IP FPs forwarding table through the *ipRoutesPoolCapacity* attribute. **`set Lp/<lp> Eng Fcrc Pqc Ov ipRoutesPoolCapacity <ipRtsCap>`**

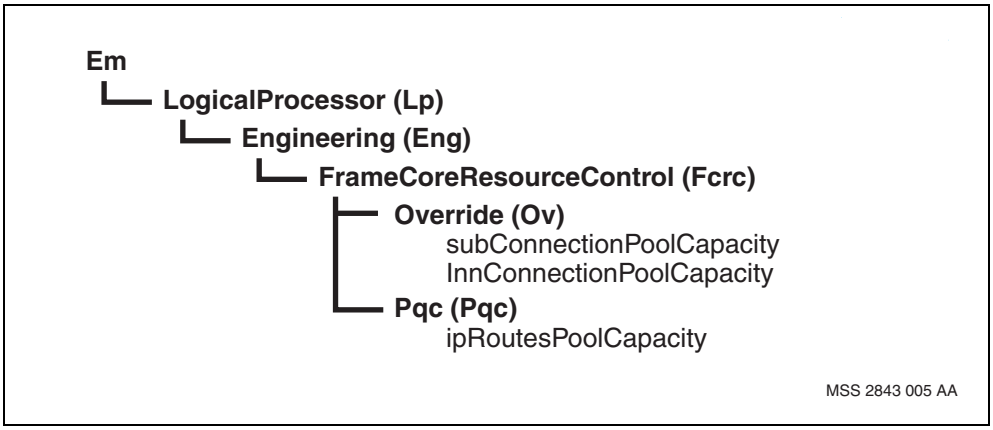
--End--

Variable definitions

Variable	Value
<ipRtsCap>	is a numeric value between 0 and 65 536. The default value is 4096.
<lnnConnCap>	is a numeric value between 1 to 2048 or cardDependent for ATM IP FPs. For CQC-based FPs, the maximum value is 256. The default value is cardDependent. The default value sets the size of the pool to 256 for ATM IP FPs and 128 for CQC-based FPs.
<lp>	is the number of the LP on which the interface is configured
<subConnCap>	is a numeric value between 0 to 49 152 or cardDependent for ATM IP FPs. For CQC-based FPs, the maximum value is 8192. The default value is cardDependent. The default value sets the size of this pool to 1024. IP routing entries also compete for this memory.

Procedure job aid

Frame core resource control component hierarchy



Configuring connection and buffer space for CQC-based FPs

Configure connection and buffer space for CQC-based FPs to set the maximum number of connections or change the size of cell or frame memory.

On CQC-based FPs, you can also configure traffic management parameters for the FP.

Prerequisites

WARNING

This procedure resets the FP

Configure the *Arc* component before you activate any ATM service to prevent service interruption.

Procedure steps

Step	Action
1	Add an ATM resource control Override component for the LP. **`add Lp/<lp> Eng Arc Ov`**
2	For multicast connections, define the number of point-to-multipoint branches for all unspared ATM interfaces at an ATM CQC-based FP. **`set Lp/<lp> Eng Arc Ov multicastBranchesCapacity <mCastCap>`**
3	Set the maximum number of connections that can be enabled on the logical processor. This attribute equally distributes connections between ports. If you set the value of this attribute, the value of the *Cqc Ov connectionPoolCapacity* must be 0 for each port. **`set Lp/<lp> Eng Arc Ov connectionPoolCapacity <total>`**
4	Add the cell queue controller component. If you want to divide the number of connections between ports, set the maximum number of connections that can be enabled on each port. **`add Lp/<lp> Eng Arc Cqc`** The system adds the *Override* component automatically.
5	Set the *connectionPoolCapacity* attribute. **`set Lp/<lp> Eng Arc Cqc Ov connectionPoolCapacity 0 <connCap> 1 <connCap> 2 <connCap>`**
6	Set the percentage of the transmit memory allocated to the transmission of cells on this logical processor. The system allocates the remainder of the memory to transmit frames. **`set Lp/<lp> Eng Arc Ov txCellMemoryAllocation <txCell>`**

7 Set the percentage of the receive memory allocated to the reception of cells on this logical processor. The system allocates the remainder of the memory to receive frames.

```
set Lp/<lp> Eng Arc Ov rxCellMemoryAllocation <rxCell>
```

--End--

Variable definitions

Variable	Value
<connCap>	is a numeric value between 512 and 4096 inclusive (in multiples of 256). The default is 0. Ports 0 and 1 can each support 4096 connections. Port 2 (in the event of the 3-port ATM FPs) can support 2560 connections.
<lp>	is the number of the LP associated with the ATM FP.
<mCastCap>	is a decimal entry from 0 to 10 752. The default value is 0.
<rxCell>	is a percentage (1 to 99) of the total cell queue memory that the system allocates to receive traffic. The default value is 20.

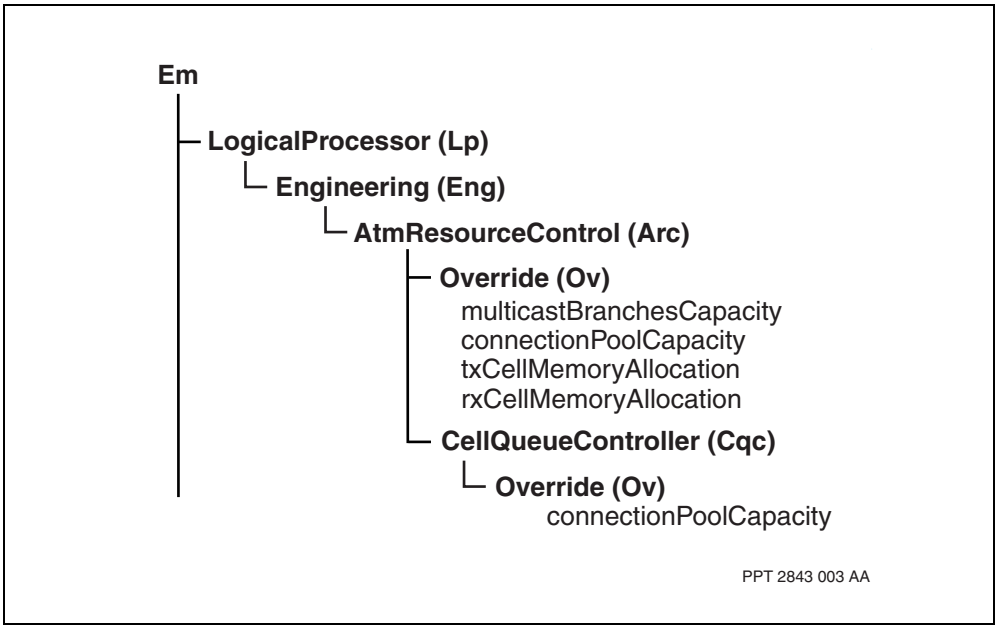
(1 of 2)

Variable	Value
<total>	is a numeric value from 0 to 64 000 inclusive. The default is 3072 for all FPs. This value defines the maximum number of connections, including multicast connections, available on an LP. If you set the value of the *connectionPoolCapacity* attribute, the value for *Arc Ov connectionPoolCapacity* must be 0. When map mode trunks are configured across equipment protection sparing configurations, PORS connections will not come up if the *connectionPoolCapacity* attribute is set to 0. The *connectionPoolCapacity* attribute must be set to a value which is greater than the sum of all the *Trunk Pa maxLc* attribute values for map mode trunks on the relevant LPs. For 8-port DS1/E1 ATM FPs, the value of <total> must not exceed the sum of VPCs, VPTs, and VCCs on all ATM interfaces bound to the LP. (This includes all ATM interfaces served by either independent ATM links and IMA link groups.) For 2-port FPs, <total> can be a numeric value from 1024 to 8192 inclusive (in multiples of 512). For 3-port FPs, <total> can be a numeric value from 1536 to 7680 inclusive (in multiples of 768) For all other ATM FPs, the value of <total> is divided evenly among all ports available. The number of connections supported on each port must not exceed the range specified in the connection map for each ATM interface bound to a port on the LP. A restricted set of values are supported on all non 8-port DS1/E1 ATM FPs:
<txCell>	is a percentage (1 to 99) of the total cell queue memory that the system allocates to transmit traffic. The default value is 50.

(2 of 2)

Procedure job aid

Connection and buffer space for CQC-based FPs component hierarchy



Configuring traffic management for CQC resource controls

Configure traffic management for CQC resource controls by changing the default traffic management parameters on CQC-based FPs.

Procedure steps

Step	Action
1	Specify the number of ports or IMA instances that can support traffic shaping and per-VC queuing on this FP. **`set Lp/<lp> Eng Arc Cqc Ov perVcQueueInterfaces <perVcIf>`**
2	Define the traffic shaping rate for all ports of the FP by setting the global scaling factor. **`set Lp/<lp> Eng Arc Cqc Ov shapingScalingFactor <scFactor>`**
3	Specify the reduction of CDV of shaped transmit traffic. **`set Lp/<lp> Eng Arc Cqc Ov cdvReduction <cdv>`**
4	Select the congestion behavior of ports when you queue a cell (or frame) to a common queue. **`set lp/<lp> Eng Arc Cqc Ov portCongestionPolicy <portCong>`**

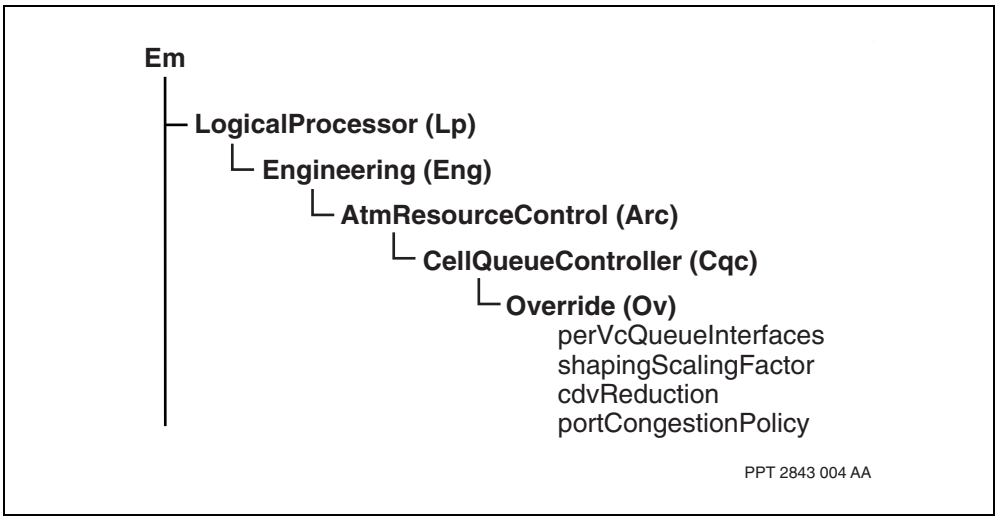
--End--

Variable definitions

Variable	Value
<cdv>	is disabled, enabled, or cardDependent. The default value is disabled on 3pDs3Atm and 3pOC3Atm cards. The default is cardDependent for all other types of cards. Set the value to enabled to reduce CDV. Setting the *cdvReduction* attribute to enabled reduces the transmit bandwidth use of the 3pDS3Atm and 3pOC3Atm cards.
<lp>	is the number of the LP associated with the FP.
<perVcIf>	is a value of 0 to 4. The default value is 0 which disables traffic shaping. The following list details the meaning of the other values: • A value of 1 indicates that port or IMA instance 0 can have traffic shaping and per-VC queuing enabled. • A value of 2 indicates that ports or IMA instances 0 and 1 can have traffic shaping and per-VC queuing enabled. • A value of 3 is permitted for independent port instances only (not IMA) and indicates that ports 0 to 2 can have traffic shaping and per-VC queuing enabled. • A value of 4 indicates that ports or IMA instances 0 to 3 can have traffic shaping and per-VC queuing enabled. The number of ports available on the 8-port DS1/E1 ATM FP depends on whether traffic shaping and per-VC queuing is enabled: • If traffic shaping and per-VC queuing is disabled, up to eight independent links or IMA link groups are supported. • If traffic shaping and per-VC queuing is enabled, only four independent links or IMA link groups are supported. For the 8-port DS1/E1 ATM FPs, the ports specified by the *perVcQueueInterfaces* attribute apply to either independent ATM links or IMA link groups. Each IMA instance can contain any of the port instances on the FP, including port instances 4 to 7. See NN10600-730 *Nortel Multiservice Switch 7400/15000/20000 Operations: Inverse Multiplexing for ATM* for details. The fewer the ports configured for traffic shaping and per-VC queuing, the greater the number of shaping rates available per port.
<portCong>	is aggregate or individualQueue. The default value is aggregate. Set the value to aggregate to ensure low priority common queues are serviced in the event of congestion. Set the value to individualQueue to prioritize traffic in the event of congestion.
<scFactor>	is 1.0, 1.4, 2.0, 2.8, 4.0, or 5.6. The default is 1.0. This scaling factor applies to all shaping stacks on the FP; that is, one global scaling factor is configurable per LP.

Procedure job aid

Traffic management for CQC resource controls component hierarchy



Configuring connection and buffer space for ATM IP FPs

Configure connection and buffer space for ATM IP FPs to set the amount of memory required to support the required number of ATM connections and the remaining memory used for cell or frame queue memory.

Prerequisites

WARNING

This procedure resets the FP

Configure the *Arc* component before you activate any ATM service to prevent service interruption.

- Ensure that the sum of the *Arc Ov protectedConnectionPoolCapacity* attribute and the *Arc Ov connectionPoolCapacity* attribute do not exceed the practical limit of the number of connection resources for the FP.
- Configured connection space beyond 29440 (29696 for PQC 12-based FPs) for connection resources may not be achieved on all ATM IP FPs unless you disable the IP hardware assist functions of the card. The disabling of the IP hardware assist functions will prevent the IP protocol stack from working and the encapsulation of IP over ATM or Frame Relay.
- To determine if the number of configured connection resources has not been satisfied, determine the number of configured resources in the corresponding *LpEngArcOv* component and compare it to the number of available and used resources in the *LpEngArc* component.
- If the number of configured connection resources is greater than 29440 (29696 for PQC 12-based FPs) in the *LpEngArcOv* component and the number of available and used connection resources in the *LpEngArc* component is equal to exactly 29440 (29696 for PQC 12-based FPs), then disabling the IP hardware assist functions will allow greater numbers of connection resources to be made available. To disable the IP hardware assist function, set the *ipRoutesCapacity* attribute of the *LpEnFcrcPqc* component to 0. This action will reset the card.

Procedure steps

Step	Action
1	Add an ATM resource control override component. **`add Lp/<lp> Eng Arc Ov`**
2	For multicast connections, define the number of point-to-multipoint branches for all unspared ATM interfaces at an ATM IP FP. **`set Lp/<lp> Eng Arc Ov multicastBranchesCapacity <mCastCap>`**

3 For multicast connections, define the number of point-to-multipoint branches for all spared ATM interfaces at an ATM IP FP.

set Lp/<lp> Eng Arc Ov protectedMcastBranchesCapacity <proMCastCap>

4 Set the maximum number of resources for unspared VPT, VCC and VPC connections that can be created on the FP. The remaining space is used for buffer space.

set Lp/<lp> Eng Arc Ov connectionPoolCapacity <total>

5 Set the maximum number of resources for spared VPT, VCC and VPC connections that can be created on the FP. The remaining space is used for buffer space.

set Lp/<lp> Eng Arc Ov protectedConnectionPoolCapacity <prototal>

6 Add the ATM queue manager component and set the maximum number of connections for each AQM instance.If you want greater control over the number of connections that can occur on each AQM instance. This number must be less than the sum of the total number of connections (*maxVpcs* + *maxVccs* + [*maxVpts* x 3]) associated with the AQM instance plus one.

add Lp/<lp> Eng Arc Aqm/<n>

The system adds the *Override* component automatically.

7 Set the connection capacity for the AQM instance. This value must be less than the sum of *connectionPoolCapacity* attribute plus the *protectedConnectionPoolCapacity* attribute of the *Arc* component.

set Lp/<lp> Eng Arc Aqm/<n> Ov connectionPoolCapacity <connCap>

8 Set the percentage of the PQC transmit memory allocated to the transmission of cells on the FP. The system allocates the remainder of the PQC memory to transmit frames.

set Lp/<lp> Eng Arc Ov txCellMemoryAllocation <txCell>

9 Set the percentage of the PQC receive memory allocated to the reception of cells on the FP. The system allocates the remainder of the PQC memory to receive frames.

set Lp/<lp> Eng Arc Ov rxCellMemoryAllocation <rxCell>

10 Set the early packet discard (EPD) offsets for each AQM on the FP.

set Lp/<lp> Eng Arc Aqm/<n> Ov highPriorityEpdOffset <hiEpdO>

set Lp/<lp> Eng Arc Aqm/<n> Ov lowPriorityEpdOffset <loEpdO>

--End--

Variable definitions

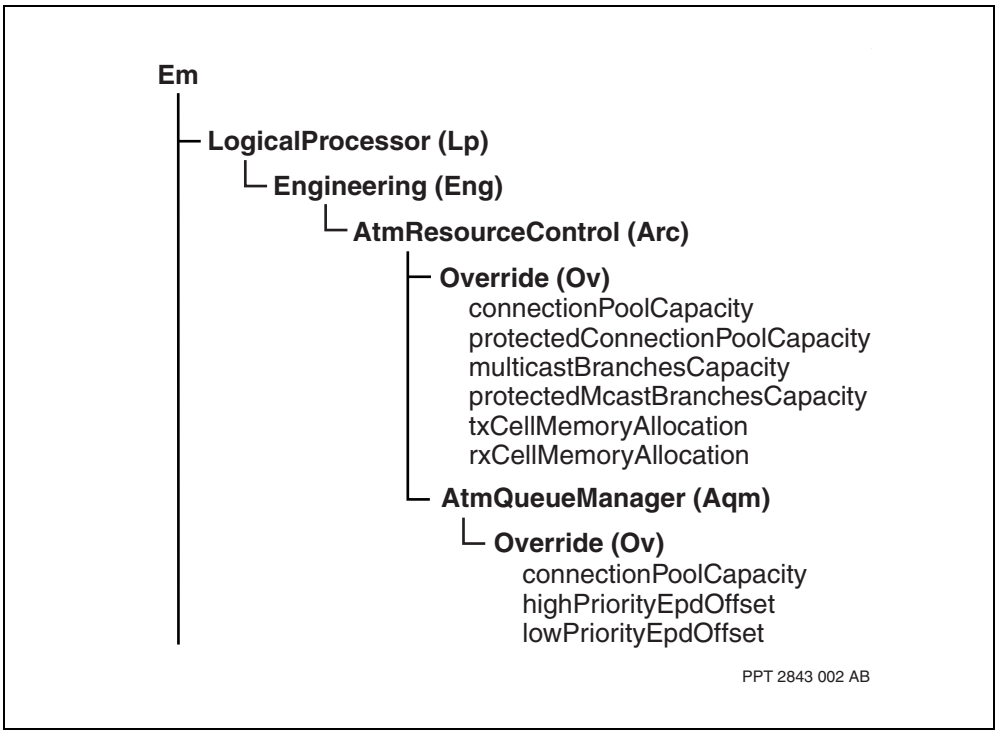
Variable	Value
<connCap>	is a numeric value between 1 and 16 000 or derivedFromArc. The default is derivedFromArc. The system arrives at the default value by dividing the value of the *connectionPoolCapacity* attribute by the number of AQM components associated with this logical processor.
<hiEpdO>	is a numeric value between 1 and 1024. The default value is 50. This attribute defines the offset for the exclusive VBR shaper emission priority and the first four (higher priority) unshaped emission priorities.
<loEpdO>	is a numeric value between 1 and 1024. The default value is 200. This attribute defines the offset for the last two (lower priority) unshaped emission priorities.
<lp>	is the number of the LP associated with the ATM FP.
<mCastCap>	is a decimal entry from 0 to 16 384. The default value is 0. The 1-port OC48c ATM IP FP does not support multicast connections.
<n>	is the instance of the ATM queue manager. The value is 0, 1, 2 or 3. The value represents either a related port or a group of related ports on the ATM FP. The number of instances depends on the type of FP.
<proMCastCap>	is a decimal entry from 0 to 16 384. The default value is 0. The 1-port OC48c ATM IP FP does not support multicast connections.
<prototal>	is a numeric value from 0 to 64000. The default is 0. For ATM IP FPs, the default value of <prototal> is generally sufficient for cases where an LP is linked to an FP that provides one to four unspared ATM interfaces, using the default values of *maxVpcs*, *maxVpts* and *maxVccs*. For ATM FPs that can provide spared ATM interfaces, see Hitless ATM services on FPs (page 186). When map mode trunks are configured across equipment protection sparing configurations, PORS connections will not come up if the *connectionPoolCapacity* attribute is set to 0. The *connectionPoolCapacity* attribute must be set to a value that is greater than the sum of all the *Trunk Pa maxLc* attribute values for map mode trunks on the relevant LPs.
<rxCell>	is a percentage (1 to 99) of the total PQC cell queue memory that the system allocates to receive traffic. The default value is 20.

(1 of 2)

Variable	Value
<total>	is a numeric value from 0 to 64000. The default is 3072. For ATM IP FPs, the default value of <total> is generally sufficient for cases where an LP is linked to an FP that provides one to four ATM interfaces, using the default values of *maxVpcs*, *maxVpts* and *maxVccs*.
<txCell>	is a percentage (1 to 99) of the total PQC cell queue memory that the system allocates to transmit traffic. The default value is 50.
(2 of 2)	

Procedure job aid

Connection and buffer space for ATM IP FPs component hierarchy



Configuring port aggregation on ATM IP FPs

Configure port aggregation on ATM IP FPs to establish resource control.

Prerequisites

WARNING

This procedure resets the FP

Configure the *portCongestionPolicy* component before you activate any ATM service to prevent service interruption.

- Port aggregation on ATM IP FPs applies to all ports and is configured under the *Arc Aqm/0* component. You need to configure only the value of *Aqm/0* to affect the change for all AQMs. A semantic check ensures that when there are multiple *Aqm* instances, all instances have the same settings for port aggregation. This check simplifies configuration of the *AtmIf* service category emission priority levels if the AQM port aggregation is turned on.

Procedure steps

Step	Action
1	If an AQM/0 component does not exist, add the ATM queue manager component for AQM/0. The system adds the *Override* component automatically. **`add Lp/<lp> Eng Arc Aqm/0`**
2	Define the congestion behavior for all ports on the FP. This attribute applies to transmit traffic only, and does not apply to connections under the CBR service category. **`set Lp/<lp> Eng Arc Aqm/0 Ov portCongestionPolicy <policy>`**

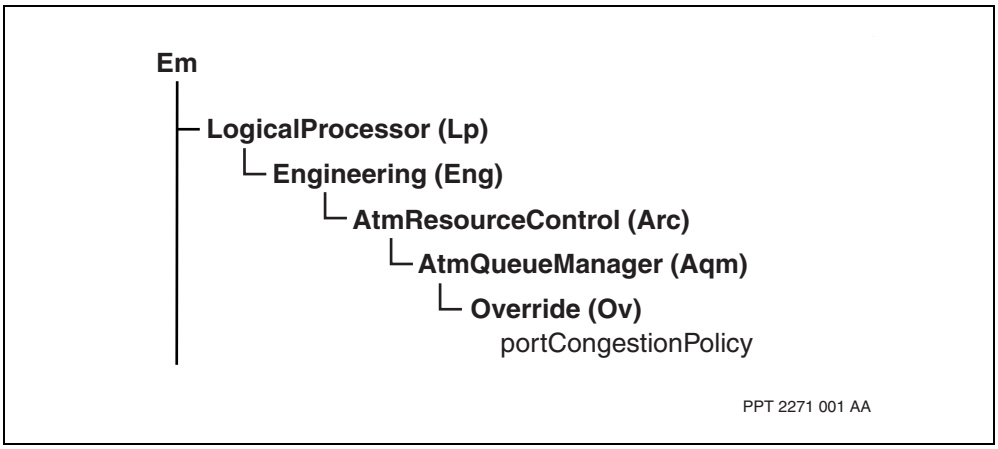
--End--

Variable definitions

Variable	Value
<lp>	the number of the LP associated with the ATM FP.
<policy>	aggregate or individualQueue. The default value is individualQueue. Set the value to aggregate to base queuing decisions on the congestion state of the entire port to which the queue belongs. The most congested queue associated with the port determines port congestion. Set the value to aggregate to ensure that low-priority common queues are serviced in the event of congestion. Set the value to individualQueue to base queuing decisions on the congestion state of the queue. Port congestion is not taken into consideration. This value ensures that the node services transmit queues according to absolute emission priority of traffic without regard to the status of the lower priority queues. On 3-port OC3 ATM IP FPs, all existing override components must have the same setting for this attribute.

Procedure job aid

Port aggregation on ATM IP FPs component hierarchy



Configuring connection and buffer space for APC-based FPs

Configure connection and buffer space for APC-based FPs by setting the maximum number of connections or changing the sizes of cell or frame PQC memory.

Prerequisites

WARNING

This procedure resets the FP

Configure the *Arc* component before you activate any ATM service to prevent service interruption.

- Calculate the sum of the *Arc Ov protectedConnectionPoolCapacity* attribute and the *Arc Ov connectionPoolCapacity* attribute to ensure these values do not exceed the practical limit of the number of connection resources for the FP. This limit varies depending on the configuration. Please contact your Nortel Networks' representative to determine the connection resource limit for the FP.

Procedure steps

Step	Action
1	Add an ATM resource control override component. **`add Lp/<lp> Eng Arc Ov`**
2	For multicast connections, define the number of point-to-multipoint branches for all unspared ATM interfaces at an APC-based FP. **`set Lp/<lp> Eng Arc Ov multicastBranchesCapacity <mCastCap>`**
3	For multicast connections, define the number of point-to-multipoint branches for all spared ATM interfaces at an APC-based FP. **`set Lp/<lp> Eng Arc Ov protectedMcastBranchesCapacity <proMCastCap>`**
4	Set the maximum number of resources for unspared VPT, VCC and VPC connections that can be created on the FP. The remaining space is used for buffer space. **`set Lp/<lp> Eng Arc Ov connectionPoolCapacity <total>`**
5	Set the maximum number of resources for spared VPT, VCC and VPC connections that can be created on the FP. The remaining space is used for buffer space. **`set Lp/<lp> Eng Arc Ov protectedConnectionPoolCapacity <prototal>`**

6 Add the Apc component. If you want greater control over the number of connections per *Apc* instance, set the maximum number of connections that can occur on each instance. This number must be less than the sum of the total number of connections (*maxVpcs* + *maxVccs* + [*maxVpts* x 3]) associated with the *Apc* instance plus one.

```
add Lp/<lp> Eng Arc Apc/<n>
```

The system adds the *Override* component automatically.

7 Set the connection capacity for the Apc instance. This value must be less than or equal to the *connectionPoolCapacity* attribute of the *Arc* component.

```
set Lp/<lp> Eng Arc Apc/<n> Ov connectionPoolCapacity
<connCap>
```

8 Set the maximum buffer spaces allowed for each of the *EmissionPriorities* in the *APC* instance of the *Lp* component.

```
set Lp/<lp> Eng Arc Apc/<n> Ov bufferLimitPerEP
<buffperEP>
```

9 Set the percentage of the PQC transmit memory allocated to the transmission of cells on the FP. The system allocates the remainder of the PQC memory to transmit frames.

```
set Lp/<lp> Eng Arc Ov txCellMemoryAllocation <txCell>
```

10 Set the percentage of the PQC receive memory allocated to the reception of cells on the FP. The system allocates the remainder of the PQC memory to receive frames.

```
set Lp/<lp> Eng Arc Ov rxCellMemoryAllocation <rxCell>
```

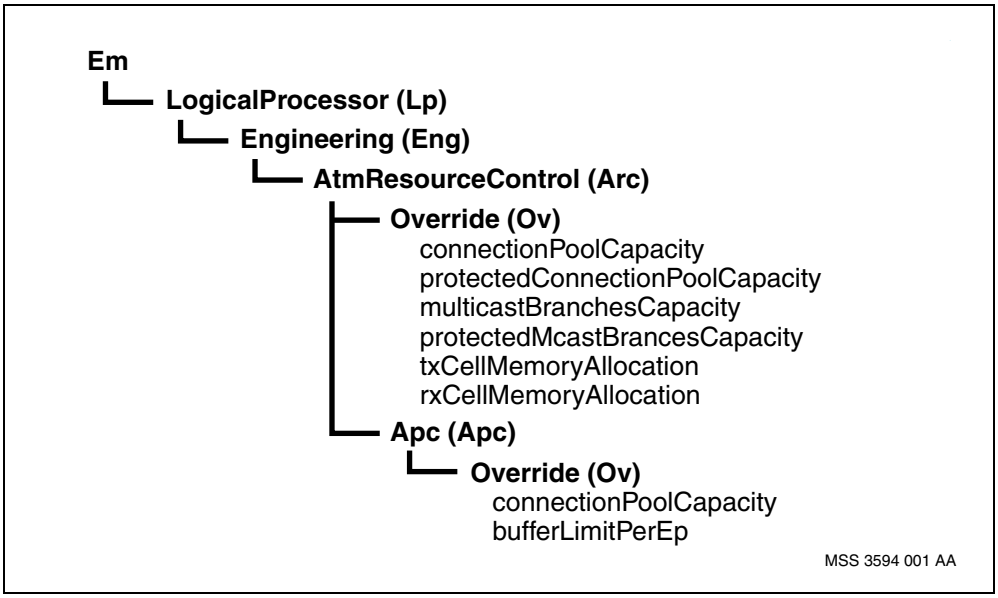
--End--

Variable definitions

Variable	Value
<buffperEP>	is a vector array of eight numeric values. The bufferLimitperEP in cells, partitions the total buffer space available on an APC device at the egress. For more information, see NN10600-060 *Nortel Multiservice Switch 7400/15000/20000 Component Reference*.
<connCap>	is a numeric value between 1 and 16 000 or *derivedFromArc*. The default is *derivedFromArc*. The system arrives at the default value by dividing the value of the *Arc* component's *connectionPoolCapacity* attribute by the number of *Apc* components associated with this logical processor. The value of the *Lp Eng Arc Apc Ov connectionPoolCapacity* attribute restricts the sum of the *maxVccs* and *maxVpcs* connections across all AtmIfs bound to this *Apc* instance.
<lp>	is the number of the LP associated with the ATM FP.
<mCastCap>	is a decimal entry from 0 to 16 384. The default value is 0.
<n>	is the instance of the Apc. The value is 0, 1, 2 or 3. The value represents a port on the ATM FP.
<proMCastCap>	is a decimal entry from 0 to 16 384. The default value is 0.
<prototal>	is a numeric value from 0 to 64 000. The default is 0. For APC-based FPs, the default value of <prototal> is generally sufficient for cases where an LP is linked to a FP that provides one to four unspared ATM interfaces, using the default values of *maxVpcs*, *maxVpts* and *maxVccs*.
<rxCell>	is a percentage (1 to 99) of the total PQC cell queue memory that the system allocates to receive traffic. The default value is 20.
<total>	is a numeric value from 0 to 64 000. The default is 3 072. For APC-based FPs, the default value of <total> is generally sufficient for cases where an LP is linked to a FP that provides one to four ATM interfaces, using the default values of *maxVpcs*, *maxVpts* and *maxVccs*.
<txCell>	is a percentage (1 to 99) of the total PQC cell queue memory that the system allocates to transmit traffic. The default value is 50.

Procedure job aid

Connection and buffer space for APC-based FPs component hierarchy



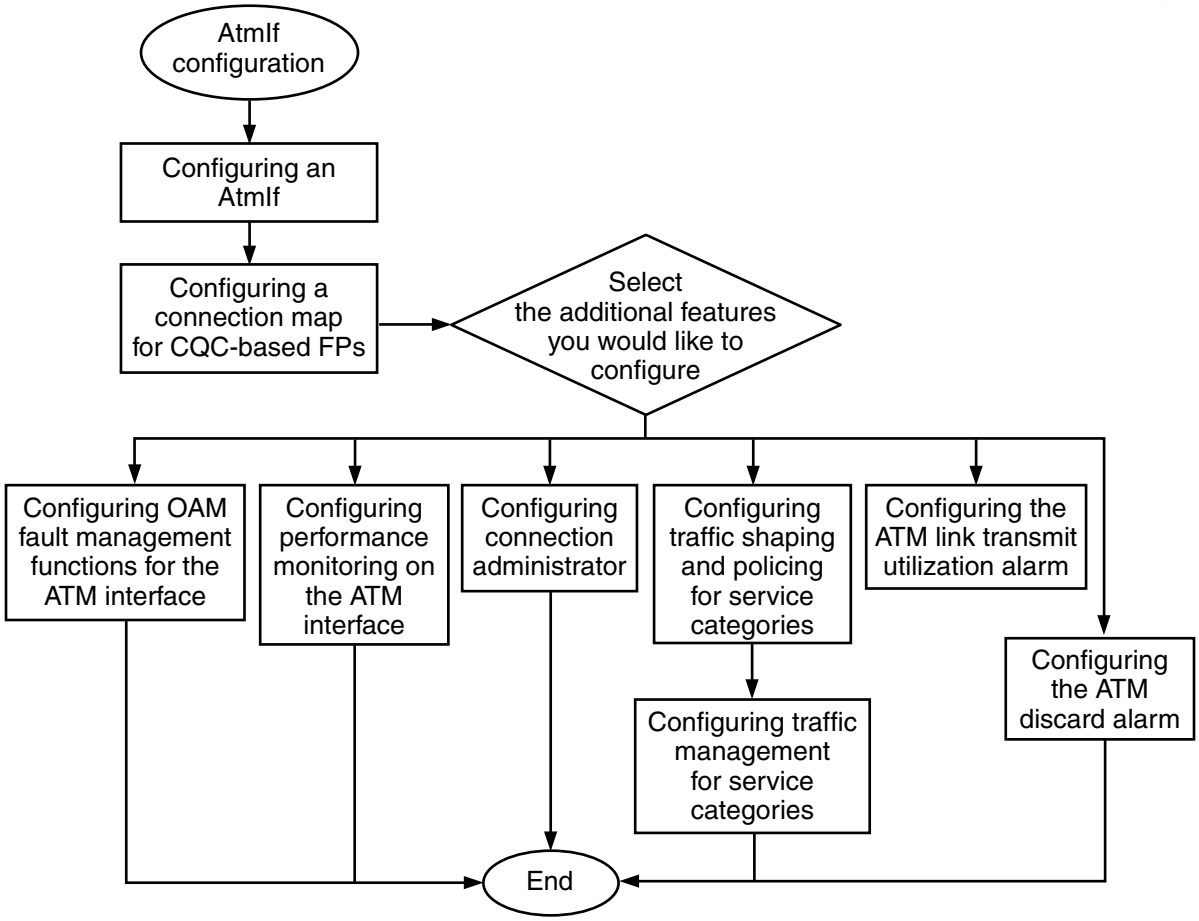
AtmIf configuration

Configure an ATM interface (AtmIf) to provide an ATM connection between two network elements. Each network element may be a terminal, another Nortel Multiservice Switch node, or an external ATM network.

AtmIf configuration procedures

This task flow shows you the sequence of procedures you perform to configure AtmIf. To link to any procedure, go to AtmIf configuration procedure navigation (page 33).

AtmIf configuration procedures



AtmIf configuration procedure navigation

- Configuring an AtmIf (page 34)
- Configuring the connection map for CQC-based FPs (page 36)
- Configuring OAM fault management functions for the ATM interface (page 39)
- Configuring performance monitoring on the ATM interface (page 41)
- Configuring connection administrator (page 47)
- Configuring traffic shaping and policing for service categories (page 53)
- Configuring traffic management for service categories (page 57)
- Configuring the ATM link transmit utilization alarm (page 63)
- Configuring the ATM discard alarm (page 64)

Configuring an AtmIf

Configure an AtmIf to add an ATM interface and link it to a port to provide connectivity to an ATM network.

Procedure steps

Step	Action
1	Add an ATM Interface component. **`add AtmIf/<n>`**
2	Link the interface to a port. **`set AtmIf/<n> interfaceName <port>`**
3	Specify a label for the remote ATM interface located at the opposite end of the physical link. **`set AtmIf/<n> remoteAtmInterfaceLabel <remoteAtmIf>`**
4	Optionally, set the maximum number of VPI bits that are active for this interface if you are configuring an NNI connection. **`set AtmIf/<n> maxVpiBits <vpiBits>`**

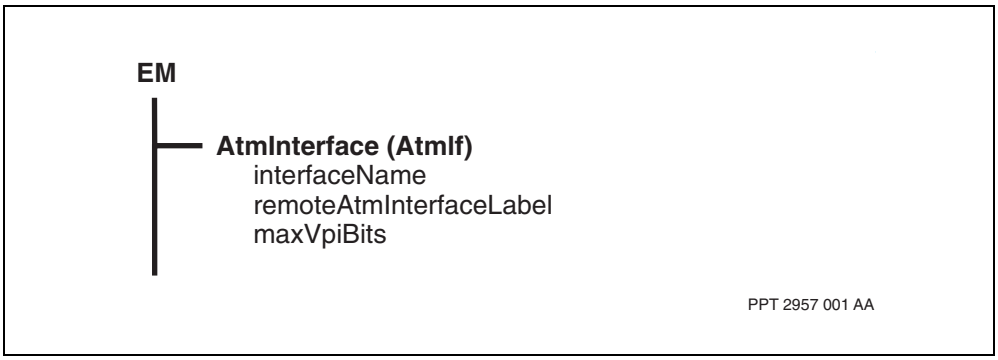
--End--

Variable definitions

Variable	Value
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095. For each 2pSTM1eCh FP (NTNQ91), up to 57 ATM interfaces can be configured per port. The ATMIF cannot be used on E1 57 to 62 (or k=3, l = 2 to 7, and m=3).
<port>	is the logical processor on which the ATM interface is activated. The values for <port> depend on the FP being used. For example it could be *Lp/4 DS1/2 chan/0*, *Lp/4 E3/0 chan/0*, or *Lp/4 SONET/2* Path/0. The port identified in the <port> parameter must already be configured.
<remoteAtmIf>	is an ASCII text string from 0 to 60 characters. The default value is empty.
<vpiBits>	is the VPI range for the interface. The value is either 8 or 12. The default value is 8. Set the value to 8 if you require a VPI range from 0 to 255. Set the value to 12 to increase the VPI range to 4 095.

Procedure job aid

AtmIf component hierarchy



Configuring the connection map for CQC-based FPs

Configure the connection map for CQC-based FPs to define the range of address spaces for connections under the ATM interface. The connection map is also referred to as the available or usable connection map.

Prerequisites

- For a detailed description of how the connection map works, see NN10600-702 *Nortel Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals*. For detailed engineering considerations and constraints, see Nortel Multiservice Switch Release Notes.
- Carefully consider the size of the connection map for CQC-based FPs (excluding the eight-port DS1/E1 FPs). If the size cannot accommodate future expansion and needs to be increased at a later date, all ATM connections under the interface must be then manually re-configured. See NN10600-702 *Nortel Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals* for a discussion of how to set up the connection map.
- To configure the connection, you must observe the following rules:
  - <connection map range for VPCs> >= <maxVpcs>
  - <connection map range for VCCs> >= <maxVccs + maxVpts>
- For an overview of the relationship between resource management and CQC-based interfaces, see Relationship between the resource controls and the interface for CQC-based FPs (page 223).
- For some sample values you can use to configure the connection map, see Connection map templates for CQC-based FPs (page 223).

Procedure steps

Step	Action
1	Add the connection mapping component. The system automatically adds the *Override* subcomponent. **`add AtmIf/<n> ConnMap`**
2	Specify the number of VCCs for VPI Zero VCC Space. **`set AtmIf/<n> ConnMap Ov numVccsForVpiZero <nZVccs>`**
3	Specify the initial VPI of the programmable VCC space. **`set AtmIf/<n> ConnMap Ov firstNonZeroVpiForVccs <firstVpi>`**
4	Specify the number of VPI values supported by the programmable VCC space. **`set AtmIf/<n> ConnMap Ov numNonZeroVpisForVccs <nVpis>`**

5 Specify the number of VCCs supported for each VPI in the programmable VCC space.

```
set AtmIf/<n> ConnMap Ov numVccsPerNonZeroVpi <nVccs>
```

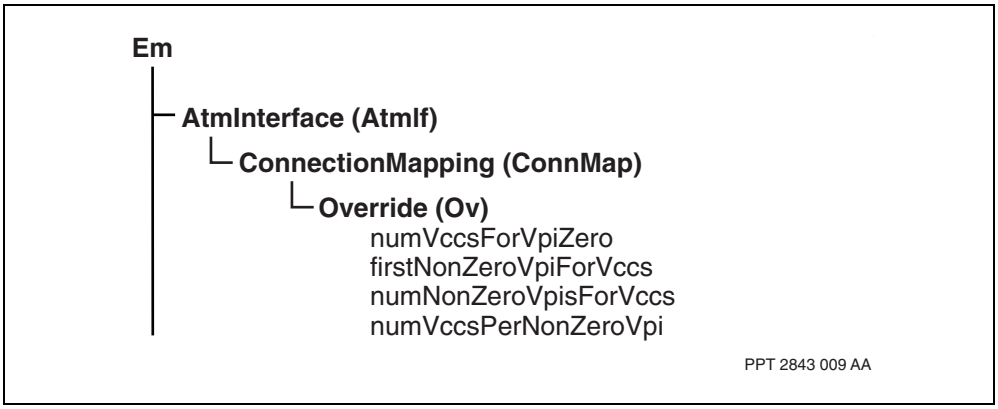
--End--

Variable definitions

Variable	Value
<firstVpi>	is a decimal value representing the initial VPI of the programmable VCC space. The default for all FP types is 1. For APC-based FPs, the value can be any integer number between 1 and 255 for UNI applications and between 1 and 4 095 for NNI applications. For CQC-based FPs, the value is either 1 or a non-zero multiple of 16 having a maximum value of 240.
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095.
<nVccs>	is a number expressed in powers of two with a minimum value to 2 for all FP types. The default for all FP types is 64. For APC-based FPs, the maximum value is 16 384. For CQC-based FPs, the maximum value is 2 048. If you are configuring a VPT VPC, you must define this attribute with a value of 8 or greater.
<nVpis>	is any value from 0 to 4 095. The default for all FP types is 0. A value of 0 indicates that the programmable VCC space is not used. For APC-based FPs, the value can be any integer number between 0 and 255 for UNI applications and between 0 and 4 095 for NNI applications. For CQC-based FPs, the value ranges from 0 to 255.
<nZVccs>	is the number of VCCs supported in VPI Zero VCC space. For APC-based FPs, the default is 1 024. The allowed values for APC-based FPs are either 1 or powers of 2 to a maximum value equal to 16 384. For CQC-based FPs, the default is 768. the size of this space for CQC-based FPs must be a multiple of 256, cannot be smaller than 256 where VPI=0 is used for VCCs, and the maximum value is 16 128. Where VPI=0 is used for a VPC, then this value must be 0. The address defined by VPI.VCI = 0.0 is reserved. VCI 0-31 is reserved for future use by standards.

Procedure job aid

Configuring the connection map component hierarchy



Configuring OAM fault management functions for the ATM interface

Configure OAM fault management functions for the ATM interface to provide operational, maintenance, and management functions for detecting errors and notifying the management stations of congestion.

Procedure steps

Step	Action
1	Set the OAM segment boundary. **`set AtmIf/<n> oamSegmentBoundary <sb>`**
2	If you want VCCs to inherit VP-layer faults detected by the VPT, set the fault hold-off time. Changing this attribute does not take effect until any existing fault clears. **`set AtmIf/<n> faultHoldOffTime <faultHOT>`**
3	Specify the *segSwitchSideLoopback* for the ATM interface. This setting enables or disables the loopback on the ATM interface. **`set AtmIf/<n> segSwitchSideLoopback <segSwLbk>`**
4	Specify the default *endToEndLoopback* for the ATM interface. This setting enables or disables the loopback on the ATM interface. **`set AtmIf/<n> endToEndLoopback <eeLbk>`**

CAUTION
Use of CTD may affect SPVC operation
The CTD feature makes use of the ATM OAM loopback cells. The loss of three consecutive loopback cells will fail the loopback test and take down the SPVC. Possible causes of this include discarded cells due to congestion and traffic management, loss of connectivity in the network, and loopback cells not being returned at the other segment endpoint.

5	Enable or disable cell transfer delay (CTD) calculation. **`set AtmIf/<n> ctdCalculation on|off`**

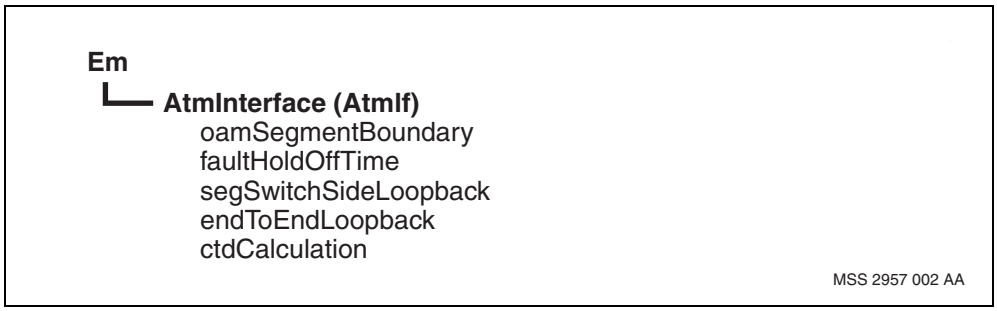
--End--

Variable definitions

Variable	Value
<eeLbk>	specifies whether end-to-end loopback insertion and termination should be performed on this connection. This value can be on or off. The default value is off.
<faultHOT>	is the amount of delay before VP-layer faults are propagated to the associated VCCs. A value of 0 (zero) allows the VC layer to react immediately to VP-layer faults. A value of infinity prevents VP-layer faults from passing to the VC layer. The default is infinity.
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095.
<sb>	defines the interface for OAM segment boundaries. The value is either yes or no. The default is yes.
<segSwLbk>	specifies whether device-side segment loopback insertion and termination should be performed on this connection. This value can be on or off. The default value is off.

Procedure job aid

OAM fault management functions for the ATM interface component hierarchy



Configuring performance monitoring on the ATM interface

Configure performance monitoring on the ATM interface to periodically evaluate ATM hardware and software through the use of performance management cells.

Procedure steps

Step	Action
1	Add a *Pm* component under the *VCD* or *VPD* component. **add AtmIf/<n> <connection_type>/<x> <descriptor_type> pm**
2	Choose the measurement to calculate: availability ratio, cell loss ratio, or both, and the side of the segment to test. **set atmif/<n> pm segSwitchsideMeasurement <segSwitchSideMeasurement>** **set atmif/<n> pm segLinksideMeasurement <segLinkSideMeasurement>**
3	Choose how the cell loss ratio is measured on the device side of a connection. **set atmif/<n> pm switchSideClrType <switchSideClrType>**
4	To enable auto start mode performance monitoring, set the control mode to autoStart and activate the provisioning view. **set atmIf/<n> pm controlMode <mode>** Activate, confirm and commit the configuration changes, Activating configuration changes (page 238).
5	Stop the measurement process by turning off the AtmIf Pm measurements. **set atmIf/<n> pm <segmentSide> <measurement>** Activate, confirm and commit the configuration changes, Activating configuration changes (page 238).
6	To enable on-demand performance monitoring, set the control mode to onDemand and activate the provisioning view. **set atmIf/<n> pm controlMode <mode>** Activate, confirm and commit the configuration changes, Activating configuration changes (page 238).
7	Start measurements for on-demand mode performance monitoring. **start atmIf/<n> vcc/<x> pm/<side>** **start atmIf/<n> vpc/<x> pm/<side>**
8	When you have taken enough measurements, stop measurements for on-demand mode performance monitoring.

```
stop atmIf/<n> vpc/<x> pm/<side>

stop atmIf/<n> vcc/<x> pm/<side>
```

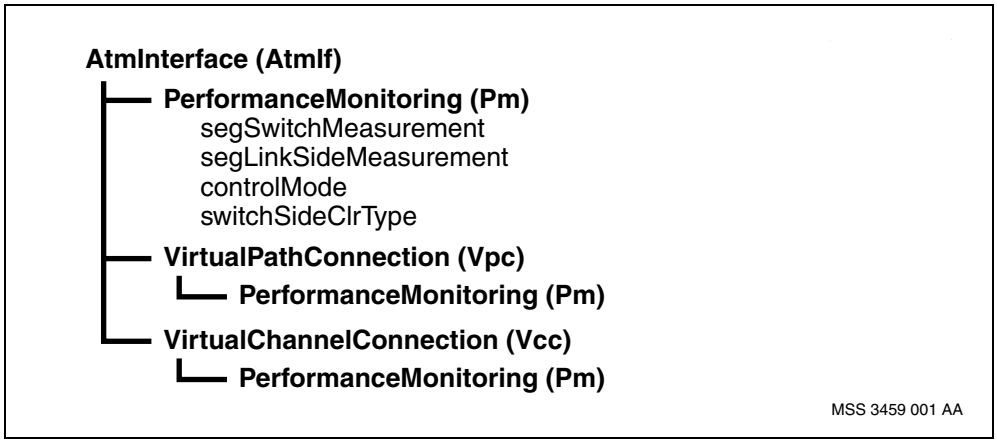
--End--

Variable definitions

Variable	Value
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<descriptor_type>	is vcd (for a virtual channel connection) or vpd (for a virtual path connection).
<measurement>	is CLR, AR, or both CLR and AR.
<mode>	is autoStart for auto mode performance monitoring, or onDemand for on demand mode performance monitoring. On demand mode provides the ability to control measurement collection through the use of start and stop verbs.
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095.
<segLinkSideMeasurement>	is a value that specifies the default PM measurements on the link side configuration for all AtmIf Vcc and AtmIf Vpc connections on the Atm interface. The default values are ~ar for availability ratio and ~clr for cell loss ratio.
<segmentSide>	is segSwitchSide for device side testing or segLinkSide for link side testing, or both segSwitchSide and segLinkSide.
<segSwitchSideMeasurement>	is a value that specifies the default PM measurements on the device side configuration for all AtmIf Vcc and AtmIf Vpc connections on the Atm interface. The default values are ~ar for availability ratio and ~clr for cell loss ratio.
<side>	is switch for device side or link for link side.
<switchSideClrType>	is complete or partial. The default value is complete. Complete includes egress discards at the segment endpoints as per the ITU standards. Partial indicates that egress discards do not contribute to the CLR measurement.
<x>	is the instance of the *Vcc* or *Vpc*. For VCCs, the instance value defines the VPI and VCI values for this connection. For VPCs, the instance value defines the VPI value for this connection.

Procedure job aid

Performance monitoring on the ATM interface component hierarchy



Configuring performance monitoring on a connection under the ATM interface

Configure performance monitoring on a connection under the ATM interface to periodically evaluate ATM hardware and software through the use of performance management cells.

Procedure steps

Step	Action
1	Choose the measurements to calculate CL, AR, or both, and the side of the segment to test. **set atmif/<n> <connection_type>/<x> <descriptor_type> pm <segment side> <measurement>**
2	Turn the measurements off for the side of the segment you do not wish to test. **set atmif/<n> <connection_type>/<x> <descriptor_type> pm <segment side> ~<measurement>**
3	Choose how the cell loss ratio is measured on the device side of a connection. **set atmif/<n> <connection_type>/<x> <descriptor_type> pm switchSideClrType <switchSideClrType>**
4	Set the control mode. **set atmif/<n> <connection_type>/<x> <descriptor_type> pm controlMode <mode>**
5	To start measurements in auto-start mode, activate the provisioning view, Activating configuration changes (page 238).
6	To start measurements in on-demand mode, activate the provisioning view, see Activating configuration changes (page 238), and then start performance monitoring under the connection. **start atmif/<n> <connection_type>/<x> pm/<side>**
7	When you have collected enough measurements in on-demand mode, stop measurements for on-demand mode performance monitoring. **stop atmif/<n> <connection_type>/<x> pm/<side>**
8	When you have collected enough measurements during auto-mode performance monitoring, stop measurements by activating a different provisioning view, Activating configuration changes (page 238).

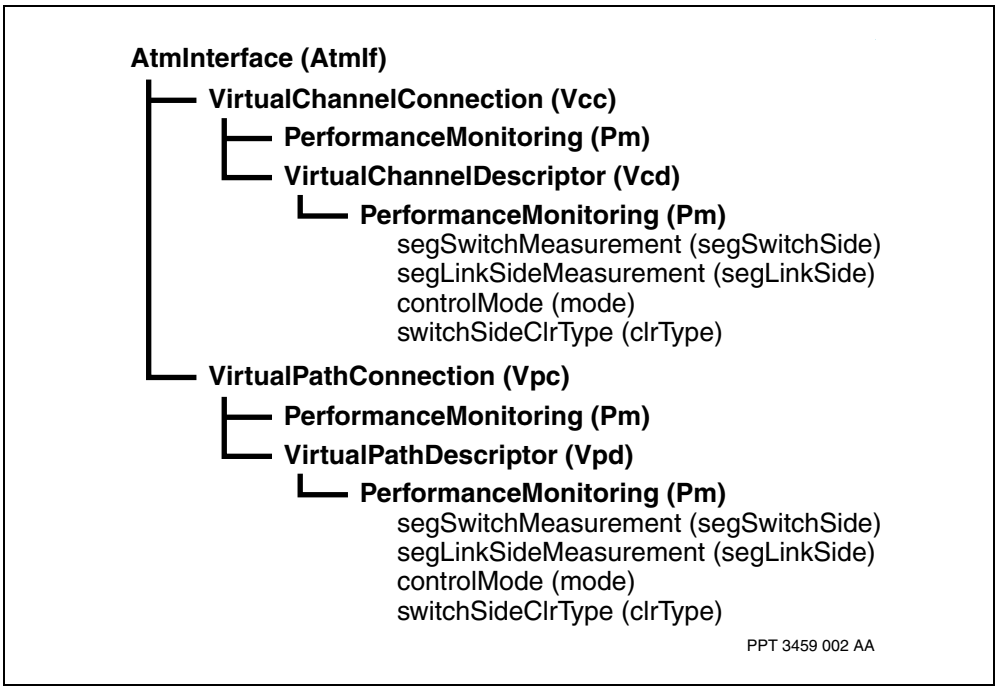
--End--

Variable definitions

Variable	Value
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<descriptor_type>	is vcd (for a virtual channel connection) or vpd (for a virtual path connection).
<measurement>	is CLR, AR, or both CLR and AR.
<mode>	is autoStart for auto mode performance monitoring, or onDemand for on demand mode performance monitoring. On demand mode provides the ability to control measurement collection through the use of start and stop verbs.
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095.
<segment side>	is segSwichSide for device side testing or segLinkSide for link side testing, or both segSwitchSide and segLinkSide.
<side>	is switch for device side or link for link side.
<switchSideClrType>	is complete or partial. The default value is complete. Complete includes egress discards at the segment endpoints as per the ITU standards. Partial indicates that egress discards do not contribute to the CLR measurement.
<x>	is the instance of the *Vcc* or *Vpc*. For VCCs, the instance value defines the VPI and VCI values for this connection. For VPCs, the instance value defines the VPI value for this connection.

Procedure job aid

Performance monitoring on a connection under the ATM interface component hierarchy



Configuring connection administrator

Configure connection administrator to define the maximum number of connections for each instance of the ATM interface. The connection administrator is also used to define and configure bandwidth pools.

Procedure steps

Step	Action
1	Set the maximum number of connections (VPCs, VPTs, and VCCs) for the interface. Select values for *maxVpcs*, *maxVccs* and *maxVpts* as necessary. **set AtmIf/<n> Ca maxVpcs <max_connections>** **set AtmIf/<n> Ca maxVpts <max_connections>** **set AtmIf/<n> Ca maxVccs <max_connections>**
2	Set the maximum number of connections for the UBR service category. **set AtmIf/<n> Ca Ubr/0 maxVpcs <max_connections2>** **set AtmIf/<n> Ca Ubr/0 maxVpts <max_connections2>** **set AtmIf/<n> Ca Ubr/0 maxVccs <max_connections2>** **set AtmIf/<n> Ca Ubr/0 minimumCellRate <mcr>**
3	Configure the connection pools capacity for the service category. **set AtmIf/<n> Ca bandwidthPool 1 <percentage1> 2 <percentage2> 3 <percentage3> 4 <percentage4> 5 <percentage5>**
4	Configure the connection pool assignments for the service category. **set AtmIf/<n> Ca <ServiceCategory>/0 pool <pool>**
5	Set the cell loss ratio (CLR) for each applicable service category. **set AtmIf/<n> Ca <ServiceCategory>/0 cellLossRatio <clr>**
6	Set the cell delay variation tolerance (a parameter that affects the equivalent cell rate) for the service category. **set AtmIf/<n> Ca <ServiceCategory>/0 cdvt <cdvt>**
7	Configure the VPI range for SVPs. **set AtmIf/<n> Ca minAutoSelectedVpi <minVpi>** **set AtmIf/<n> Ca maxAutoSelectedVpi <maxVpi>**
8	Define the minimum range of VPI.VCI values in the connection map space that can be used for switched connections. **set AtmIf/<n> Ca minAutoSelectedVciForVpiZero <vciZVpi>** **set AtmIf/<n> Ca minAutoSelectedVciForNonZeroVpi <vciNZVpi>**

On the user side of the interface, these values must be less than or equal to (creating a larger range) the corresponding values on the network side. Because the network side dynamically selects the VPI.VCI that is assigned to the switched connection, a larger range on the user side ensures that the network side assigns a valid VPI.VCI in the connection map space.

9 Define the maximum range of VPI.VCI values in the connection map space that can be used for switched connections.

```
set AtmIf/<n> Ca maxAutoSelectedVciForVpiZero
<maxVciVpiZero>

set AtmIf/<n> Ca maxAutoSelectedVciForNonZeroVpi
<maxVciNonZeroVpi>
```

10 Set the permitted ATM service categories.

```
set AtmIf/<n> Ca permittedSvcAtmServiceCategories
<ServiceCategory>
```

11 Set the default holding priority for SVCs containing CBR, RTVBR, NRTVBR, and UBR traffic.

```
set AtmIf/<n> Ca <ServiceCategory>/0 holdingPriority
<hPri>
```

Attention: Set the *holdingPriority* attribute with a different value for each service category to keep higher priority connections during a bandwidth decreases.

12 Define the maximum number of point-to-multipoint branches at an ATM FP port.

```
set AtmIf/<n> Ca maxMulticastBranches <maxMCast>
```

13 Define the minimum number of signalled point-to-multipoint branches that are guaranteed at an ATM FP port.

```
set AtmIf/<n> Ca minMulticastBranches <minMCast>
```

14 If the interface supports point-to-multipoint connections, set the default holding priority for these connections containing CBR, RT-VBR, NRT-VBR, and UBR traffic.

```
set AtmIf/<n> Ca <ServiceCategory>/0
svcMpHoldingPriority <svcMpHrpi>
```

--End--

Variable definitions

Variable	Value
<cdvt>	is a numeric value from 1 to10 000. The default is 250. This value applies to all CBR traffic on the interface. If usage parameter control (UPC) is enabled for the connection, the value in this attribute affects traffic policing configuration.
<clr>	is a negative logarithmic value (base 10) between 0 and 15. The default is 10 for CBR and RT-VBR service categories, and 7 for the NRT-VBR service category. The value for this attribute must be identical for CBR and RT-VBR service categories because traffic in these service categories have the same discard priority. The value of this attribute for the NRT-VBR service category must be less than or equal to the value in the CBR and RT-VBR service categories. In Multiservice Switch systems, CLR is configured as a power of 10. For example, to configure a CLR of 10^{-10}, the <clrValue> would be 10.
<hPri>	is the default holding priority for the specified traffic type, and can be any number from 0 (highest priority) to 4 (lowest priority).
<lp>	is the number of the LP associated with the ATM card.

(1 of 4)

Variable	Value
<max_connections>	is the maximum number of virtual channel connections that can be configured on this *AtmIf* component. For virtual channel connections, this value is between 0 to 16 384. For virtual path connections and virtual path terminators, this value is between 0 to 4 096. The default is autoConfigure depending on the card type. The actual value of this autoConfigure default is displayed by the operational attribute actualMaxVccs, actualMaxVpcs, or actualMaxVpts under the AtmIf component. The total number of connections for all the ATM interfaces on a FP cannot be more than the sum of the *connectionPoolCapacity* and the *protectedConnectionPoolCapacity* attributes of the *Arc* component or the *Arc Ov* component. This includes connections for unspared ATM interfaces and connection resources that are reserved for spared ATM interfaces. The default values for *maxVpcs*, *maxVccs* and *maxVpts* are generally sufficient for cases where one to four unspared ATM interfaces are running on a single FP. If the FP is running more than four ATM interfaces, see ATM FP resource management configuration (page 11). The *maxVcc* attribute is used as a limit for the admission of all permanent Vccs (PVCs or SPVCs). When an ATM networking protocol (UNI, PNNI, IISP, or AINI) is configured, if the number of all currently admitted Vccs is less than the maxVccs, then the reserved channels for signalling Vccs of the networking protocol can be established, provided there is still enough bandwidth. However, upon a critical change to the AtmIf or an FP reset, all permanent Vccs (PVCs or SPVCs) will be established first, and then the reserved signalling Vccs for ATM networking protocol. Therefore, it is possible that the reserved signalling Vccs will fail to establish if the number of these permanent Vccs has already reached the configured maxVccs or there is a lack of bandwidth available. Consideration for reserved signalling Vccs for ATM networking must be taken into account when configuring the maxVccs. If an ATM networking protocol and n permanent Vccs (PVCs and SPVCs) are to be configured under an AtmIf, then the maxVccs under AtmIf Ca should be configured with a value of n + 2 (where 2 is the maximum number of reserved signalling Vccs for an ATM networking protocol).
<max_connections2>	is either sameAsCa or a decimal in the range of 0 to 16 384 for Vccs, or a decimal in the range of 0 to 4096 for Vpcs or Vpts. The default is sameAsCa. This attribute defines the maximum number of VCCs, VPCs, and VPTs for the UBR service category.

(2 of 4)

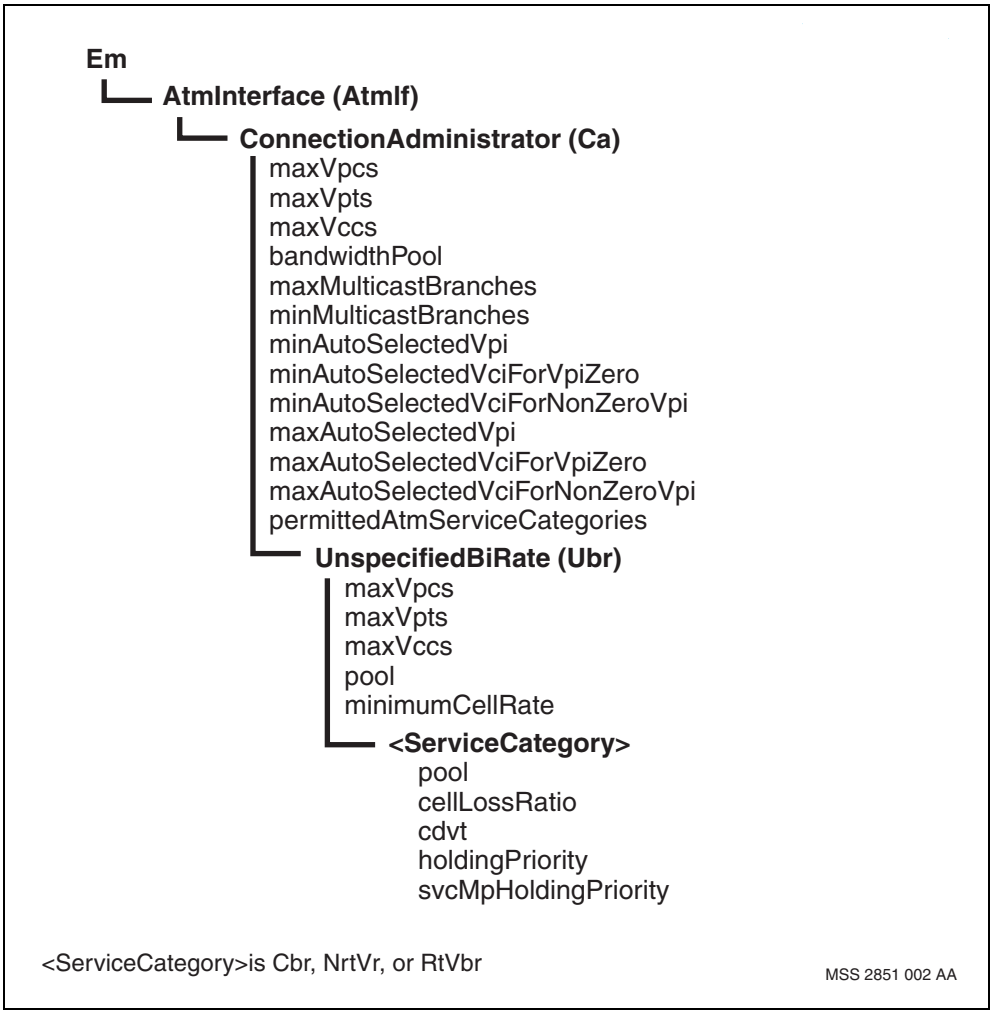
Variable	Value
<maxMcast>	is a decimal entry from 0 to 16 384 or autoConfigure. The default value is autoConfigure. The value autoConfigure selects an appropriate value depending on card type (10 752 for CQC cards and 16 384 for ATM IP cards).
<maxVciNonZeroVpi>	is the maximum VCI value for a switched connection with VPI values equal to non-zero. The value can be in the range from 32 to 65 535. The default value is 63.
<maxVciVpiZero>	is the maximum VCI value for a switched connection with a VPI value of 0. The value can be in the range from 32 to 65 535. The default value is 767.
<maxVpi>	is the maximum VPI value automatically allocated for the switched connection. This value is an integer between 1 and 4 095. The default value is 128.
<mcr>	is a decimal in the range of 0 to 2 147 483 647. The default is 0. This attribute defines a measure of the minimum useful data rate for the given ATM UBR service category for all UBR connections under AtmIf.
<minMCast>	is a decimal entry from 0 to 16 384 for ATM IP cards. For CQC cards, the range of values for this attribute is from 0 to 10 752. The default value is 0.
<minVpi>	is the minimum VPI value automatically allocated for the switched connection. This value is an integer between 1 and 4 095. The default value is 1.
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095.
<percentage1>, <percentage2>, <percentage3>, <percentage4>, <percentage5>	are vectors that define the bandwidth pools. These vectors consist of an index entry ranging from 1 to 5 and a decimal entry (<percentage1>, <percentage2>, <percentage3>, <percentage4>, and <percentage5>) that has a value between 0 and 12 800. The decimal entry sets the percentage of link bandwidth allowed in the bandwidth pool defined by the index entry. The default values of this vector are *1 100 2 0 3 0 4 0 5 0*.
<pool>	is pool1, pool2, pool3, pool4, or pool5. The default is pool1.
<ServiceCategory>	is Cbr, RtVbr, or NrtVbr. Only the 0 instance of a service category exists.
<svcMpHpri>	is the default holding priority for the specified traffic type, and can be any number from 0 (lowest priority) to 4 (highest priority).

(3 of 4)

Variable	Value
<vciNZVpi>	is the minimum VCI value for a switched connection with VPI values equal to non-zero. The value can be in the range from 5 to 65 535. The default value is 32. The VCI range 0-31 is currently reserved for use by ATM Forum and ITU.
<vciZVpi>	is the minimum VCI value for a switched connection with a VPI value of 0. The value can be in the range from 5 to 65 535. The default value is 32. The VCI range 0-31 is currently reserved for use by ATM Forum and ITU. Do not use the value 65 535 for the *minAutoSelectedVciForVpiZero* attribute. This value is for restricted use.
	(4 of 4)

Procedure job aid

Connection administrator component hierarchy



Configuring traffic shaping and policing for service categories

Configure traffic shaping and policing for service categories to optimize service levels.

If you enable traffic shaping or policing in a service category, you can disable it for each connection using the *Vpd Tm* (for *Vpcs* and *Vpts*), or *Vcd Tm* components. If you enable traffic shaping or policing in a service category, you can disable it for signaling or routing control channels using the *Vcd* component of the interface type (*Uni*, *Iisp*, Aini, or *Pnni*). If the interface is Pnni, you can enable traffic shaping for the signaling or routing control channels (RCC).

Traffic shaping is not available on the 1-port OC-12c and 1-port OC48c FPs, or for CBR connections on a CQC-based FP.

Prerequisites

- To enable traffic shaping on ATM IP FPs, you must configure the emission priorities for each applicable service category. For information on configuring emission priorities, see Configuring traffic management for service categories (page 57).
- The prerequisites to enable traffic shaping for the interface on CQC-based FPs are as follows:
  - the applicable interfaces must be set up
  - the value in the *perVcQueueInterfaces* attribute of the *Arc Cqc Ov* component must be a non zero value
- Identify the type of FP and the value of the *txTrafficDescType* attribute in the connection (*Vpc*, *Vpt*, or *Vcc*).
- Identify the value of the *rxTrafficDescType* attribute in the connection (*Vpc*, *Vpt*, or *Vcc*). See Relationship between rxTrafficDescType and policing action values (page 55) when configuring usage parameter control (UPC) for service categories.

Procedure steps

Step	Action
1	Enable traffic shaping for the service category. **`set AtmIf/<n> Ca <ServiceCategory>/0 trafficShaping <trShaping>`**
2	Enable usage parameter control (UPC) for the service category. **`set AtmIf/<n> Ca <Service Category>/0 usageParameterControl <upc>`**

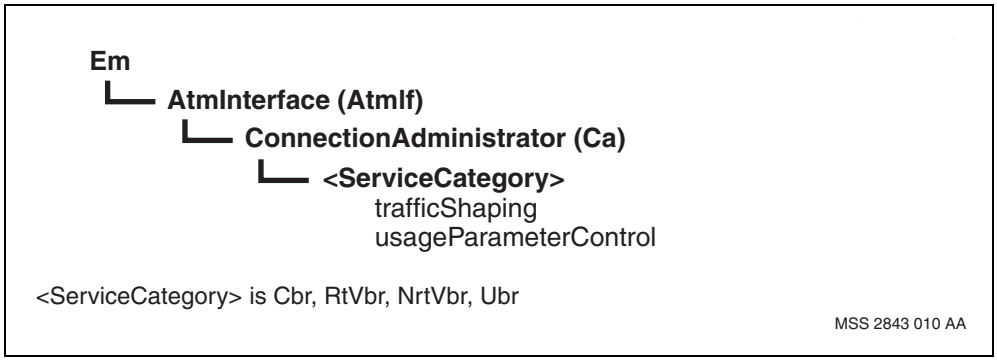
--End--

Variable definitions

Variable	Value
<n>	is the instance value of the *AtmIf* component (any unique value from 1 to 4 095).
<ServiceCategory>	is Cbr, RtVbr, NrtVbr, or Ubr. Only the 0 instance of the service category exists. On CQC-based FPs, shaping does not apply when the service category is CBR.
<trShaping>	is enabled, disabled, or inverseUpc for all card types except for the APC-based FPs. For the APC-based FPs, the value is either enabled or disabled. The default value for all FP types is disabled, which means traffic shaping is disabled for the connections in the service category. Set the value to disabled to turn off traffic shaping for connections in the service category. Set the value to enabled to permit linear traffic shaping on connections in the specified service category. This attribute must be set to disabled for the 1pOC12SmIrAtm and the 1pOC48ChSmIrAtm FP types. Set the value to inverseUpc to enable traffic shaping according to the value of the *txTrafficDescType* attribute in the connections. This value is only valid for ATM IP FPs with the value 6, 7, or 8 in the *txTrafficDescType* attribute. See the table Relationship between the txTrafficDescType and trafficShaping attribute values (page 55) for details.
<upc>	is enforced, monitored, or disabled. The default is disabled. Set the value to enforced to tag or discard cells depending on the value of the *rxTrafficDescriptorType* attribute. See the table Relationship between the txTrafficDescType and trafficShaping attribute values (page 55)for details Set the value to monitored to count UPC violations and pass cells unchanged. For CQC-based FPs, the value of monitored disables UPC. The node cannot monitor UPC violations on a CQC-based FP. If a UPC value is enforced when the txtdt or rxtdt value is set to 9, no shaping is applied. In this case, warning messages for Tx and Dx are displayed which state that to enable UPC on UBR with MDCR connections, the Rx and Tx traffic descriptor types need to be set to 3, 4, 5, 6, 7, or 8.

Procedure job aid

Traffic shaping and policing for service categories component hierarchy



Relationship between the txTrafficDescType and trafficShaping attribute values

Type of FP	txTrafficDescType value	trafficShaping value	Type of traffic shaping applied for service categories
AQM-based, CQC-based	1 or 2	enabled, disabled, inverseUpc	traffic shaping not applicable for the connection
APC-based	1 or 2	enabled, disabled	traffic shaping not applicable for the connection
AQM-based, CQC-based, APC-based	3, 4, 5, 6, 7, 8, or 9	enabled disabled	linear traffic shaping enabled for the connection
AQM-based, CQC-based	3, 4, 5, or 9	inverseUpc	linear traffic shaping enabled for the connection
AQM-based	6, 7, or 8	inverseUpc	inverse UPC traffic shaping enabled for the connection
CQC-based	6, 7, or 8	inverseUpc	linear traffic shaping enabled for the connection

Relationship between rxTrafficDescType and policing action values

rxTrafficDescType value	PCR enforcer	SCR enforcer
1 or 2	n/a	n/a
3	discard CLP 0 +1	n/a

(1 of 2)

Relationship between rxTrafficDescType and policing action values (continued)

rxTrafficDescType value	PCR enforcer	SCR enforcer
4	discard CLP 0 + 1	discard CLP 0
5	discard CLP 0 + 1	tag CLP 0
6	discard CLP 0 + 1	discard CLP 0 + 1
7	discard CLP 0 + 1	discard CLP 0
8	discard CLP 0 + 1	tag CLP 0
9	discard CLP 0 +1	n/a

(2 of 2)

Configuring traffic management for service categories

Configure traffic management for service categories to optimize traffic management for each applicable service category.

Prerequisites

- For information about additional attributes available for AQM and APC based FPs, and about restrictions related to this procedure, see Attributes for traffic management on AQM and APC-based FPs (page 221).

Procedure steps

Step	Action
1	Set the emission priority for the service category. **`set AtmIf/<n> Ca <ServiceCategory>/0 emissionPriority <ep>`**
2	If you are configuring an AQM, APC, or GQM-based FP and need to set the emission priority, add the *EmissionPriority* component for the interface. **`add AtmIf/<n> Ep/<ep2>`**
3	Set the percentage of minimum bandwidth guarantee after the premium emission priority traffic has been served. For the AQM and GQM-based FPs, the premium emission priorities are EP 0 and EP 1. For APC-based FPs, the premium priority is EP 0. **`set AtmIf/<n> Ep/<ep> minimumBandwidthGuarantee <minBw>`**
4	Configure the recoup policy. This policy recoups shaping opportunities that are lost when multiple connections schedule a cell for the same transmit opportunity. **`set AtmIf/<n> Ca <ServiceCategory>/0 shapeRecoupPolicy <recoup>`**
5	Set the default queuing option for the service category. **`set AtmIf/<n> Ca <ServiceCategory>/0 unshapedTransmitQueueing <unshaped>`**
6	Configure the fairness weighting. **`set AtmIf/<n> Ca <ServiceCategory>/0 weightPolicy <weight>`**

CAUTION

Configuring txQueueLimit may reset the ATM interface

Configuring the *txQueueLimit* attribute is a critical change that causes the ATM interface to reset. Any active call is dropped.

7 Define the size of the queue in the transmit direction.

```
set AtmIf/<n> Ca <ServiceCategory>/0 txQueueLimit <txq1>
```

8 Configure the minimum queue limit for per-VC queues for shaped connections.

```
set AtmIf/<n> Ca <ServiceCategory>/0 minPerVcQueueLimit
<minpervcql>
```

9 Configure the reference rate for per-VC queues in the service category.

```
set AtmIf/<n> Ca <ServiceCategory>/0
perVcQueueLimitReferenceRate <refrate>
```

10 Force the value for cell loss priority to equal 1 for cells in the transmit direction.

```
set AtmIf/<n> Ca <ServiceCategory>/0 forceTagging <tag>
```

--End--

Variable definitions

Variable	Value
<ep>	is a decimal value. For AQM-based FPs, the applicable range is 0 to 7 for all ATM service categories. For APC-based FPs, the values can be 0, 2, 3, 4, or 7. CQC-based FPs ignore the *emissionPriority* attribute. AQM default values: CBR 0; rt-VBR 1; nrt-VBR 4; UBR 7 APC default values: CBR 0; rt-VBR 2; nrt-VBR 3; UBR 7
<ep2>	is the instance value that corresponds to the value of the *emissionPriority* attribute. For AQM-based FPs, the value can be 2, 3, 4, 5, 6, or 7. For APC-based FPs, the value can be 2, 3, 4 or 7. For AQM-based FPs, if the value for the *emissionPriority* attribute in any service category is 2, 3, 4, 5, 6, or 7, configure the percentage of minimum bandwidth guarantee for traffic in that service category. For APC-based FPs, if the value for the *emissionPriority* attribute in any service category is 2, 3, 4, or 7, configure the percentage of minimum bandwidth guarantee for traffic in that service category.
<minBw>	is a decimal value or the value priority. The default value for all FPs is priority which means that this EP has bandwidth based on the hierarchy of emission priorities. For APC-based FPs, the allowed value of the attribute is between 0 and 100, inclusive. For non-APC-based FPs, the allowed value of the attribute is between 1 and 48, inclusive. Set the value to a decimal to specify the percentage of remaining bandwidth for transmit opportunities for this EP. The total of the minimum bandwidth guarantee for all EPs (premium and regular emission priorities) must be less than or equal to 100%.
<minpervcql>	is a decimal in the range of 5 to 12 000. (For APC-based cards, the minimum is 36.) The default is 88.
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095.
<recoup>	is either maximumEfficiency or minimumCdv. Set the value to maximumEfficiency to configure each connection to recoup scheduling delays. Refer to impact of this value in NN10600-706 *Nortel Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*, the section on shaping congestion. Set the value to minimumCdv to turn the recoup policy off for connections under this service category.

(1 of 3)

Variable	Value
<refrate>	is either autoConfigure or a decimal value that varies depending on the FP. The default for all FPs is autoConfigure which means that the software determines the appropriate value based on the FP type. For the value corresponding to a particular FP, see NN10600-060 *Nortel Multiservice Switch 7400/15000/20000 Component Reference*.
<ServiceCategory>	is Cbr, RtVbr, NrtVbr, or Ubr. Only the 0 instance of the service category exists. You can define the *ServiceCategory* attribute as Abr, but the command will fail, as the ABR service category is not supported. The forceTagging feature is only valid for switched connections with a UBR service category and permanent connections of all service categories.
<tag>	is either enabled or disabled. The default value is disabled. This attribute does not apply to interfaces on CQC-based FPs or to Vpts. Use enabled on ATM IP and APC-based FPs to force CLP=1 on all cells in the transmit direction. Use disabled on ATM IP and APC-based FPs to leave the CLP tag according to the CC to CLP mapping for cells in the transmit direction.
<txq1>	is either autoConfigure or a decimal value that varies depending on the FP. The default for all FPs is autoConfigure which means that the software determines the appropriate value based on the FP type. For the value corresponding to a particular FP, see NN10600-060 *Nortel Multiservice Switch 7400/15000/20000 Component Reference*. If you use autoConfigure, the software determines the appropriate value based on FP type. If you enabled common queueing on an AQM-based FP, set the *txQueueLimit* attribute to *autoConfigure*.

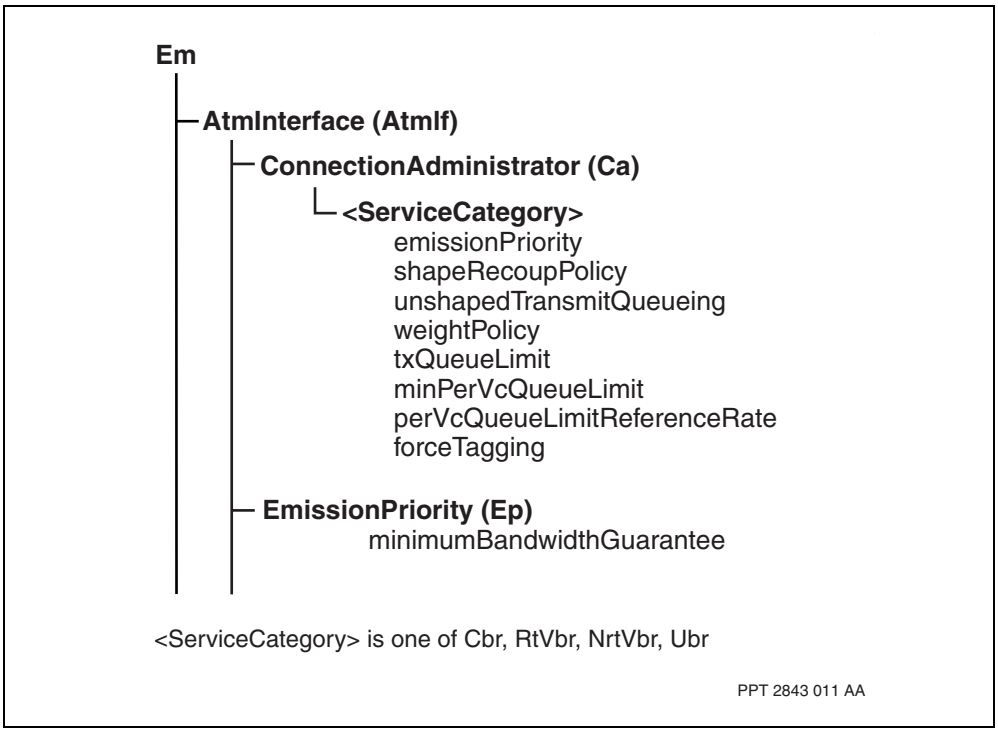
(2 of 3)

Variable	Value
<unshaped>	*is autoConfigure*, *common* or *perVc* depending on the FP. The default for all FPs is autoConfigure, which means that the software determines the appropriate value based on the FP type. For the value corresponding to a particular FP, see NN10600-060 *Nortel Multiservice Switch 7400/15000/20000 Component Reference*. PerVc queuing is not available on 1pOC12SmLrAtm and 1pOC48ChSmIrAtm. Common queueing is not available on the APC-based FPs. Set the attribute to perVc to indicate that each connection has its own buffering and queue limits. Set the attribute to common to queue the cells in first-in-first-out order on a common queuing schema. If you set the attribute to common on an AQM-based FP, set the *txQueueLimit* attribute to *autoConfigure*. Set the value to autoConfigure to set the value based the FP and traffic shaping, see the table Interpretation of the value autoConfigure for the unshapedTransmitQueueing attribute (page 222) for details. The value autoConfigure is invalid for the CBR service category.
<weight>	is scr, pcr, or ecr. The default is ecr. Use scr to set the fairness weight to be equivalent to the transmit sustained cell rate of the connection. For connections with no configured sustained cell rate, the sustained cell rate is equal to the peak cell rate. Use pcr to set the fairness weight to be equivalent to the transmit peak cell rate of the connection. Use ecr to set the fairness weight to be equivalent to the transmit equivalent cell rate of the connection.

(3 of 3)

Procedure job aid

Traffic management for service categories component hierarchy



Configuring the ATM link transmit utilization alarm

Configure the ATM link transmit utilization alarm to notify a management workstation of links exceeding the configured threshold.

Procedure steps

Step	Action
1	To add the ATM link transmit utilization alarm for the ATM interface, configure the *LinkTxUtilization* (*Ltu*) component under AtmIf. **`add AtmIf/<n> Ltu`**
2	To change the critical alarm threshold, change *critThresh* to a different value. **`set AtmIf/<n> Ltu critThresh <m>`**
3	Disable an ATM link transmit utilization alarm. **`set AtmIf/<n> Ltu critThresh disabled`**

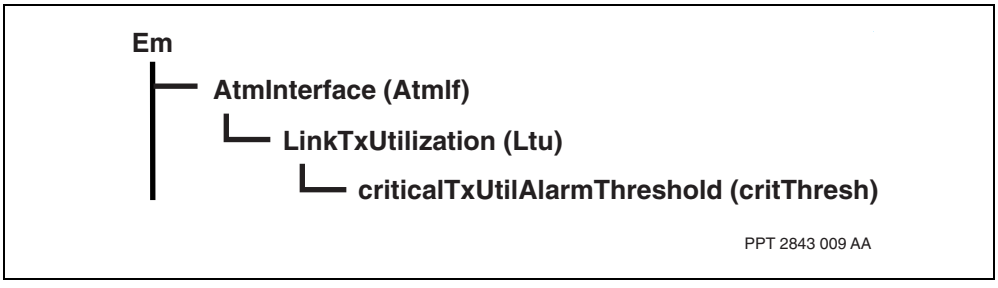
--End--

Variable definitions

Variable	Value
<m>	is a value from 1 to 99. The default value is 80 percent.
<n>	is the instance value of the *AtmIf* component (any unique value from 1 to 4 095).

Procedure job aid

ATM link transmit utilization alarm component hierarchy



Configuring the ATM discard alarm

Configure ATM discard alarm for each ATM service category to inform the network operator when a significant number of discards are occurring on the link or within one of the service categories.

Procedure steps

Step	Action
1	Set the corresponding thresholds to non-zero values for the ATM interface discards. **set AtmIf/<n> txCellDiscardThreshold <m>** **set AtmIf/<n> rxCellDiscardThreshold <m>** **set AtmIf/<n> txFrameDiscardThreshold <m>** **set AtmIf/<n> rxFrameDiscardThreshold <m>**
2	Set the corresponding thresholds to non-zero values for the CBR service category. **set AtmIf/<n> Ca Cbr/0 txCellDiscardThreshold <m>** **set AtmIf/<n> Ca Cbr/0 rxCellDiscardThreshold <m>** **set AtmIf/<n> Ca Cbr/0 txFrameDiscardThreshold <m>** **set AtmIf/<n> Ca Cbr/0 rxFrameDiscardThreshold <m>**
3	Set the corresponding thresholds to non-zero values for the rt-VBR service category. **set AtmIf/<n> Ca RtVbr/0 txCellDiscardThreshold <m>** **set AtmIf/<n> Ca RtVbr/0 rxCellDiscardThreshold <m>** **set AtmIf/<n> Ca RtVbr/0 txFrameDiscardThreshold <m>** **set AtmIf/<n> Ca RtVbr/0 rxFrameDiscardThreshold <m>**
4	Set the corresponding thresholds to non-zero values for the nrt-VBR service category. **set AtmIf/<n> Ca NrtVbr/0 txCellDiscardThreshold <m>** **set AtmIf/<n> Ca NrtVbr/0 rxCellDiscardThreshold <m>** **set AtmIf/<n> Ca NrtVbr/0 txFrameDiscardThreshold <m>** **set AtmIf/<n> Ca NrtVbr/0 rxFrameDiscardThreshold <m>**
5	Set the corresponding thresholds to non-zero values for the UBR service category. **set AtmIf/<n> Ca Ubr/0 txCellDiscardThreshold <m>** **set AtmIf/<n> Ca Ubr/0 rxCellDiscardThreshold <m>** **set AtmIf/<n> Ca Ubr/0 txFrameDiscardThreshold <m>** **set AtmIf/<n> Ca Ubr/0 rxFrameDiscardThreshold <m>**

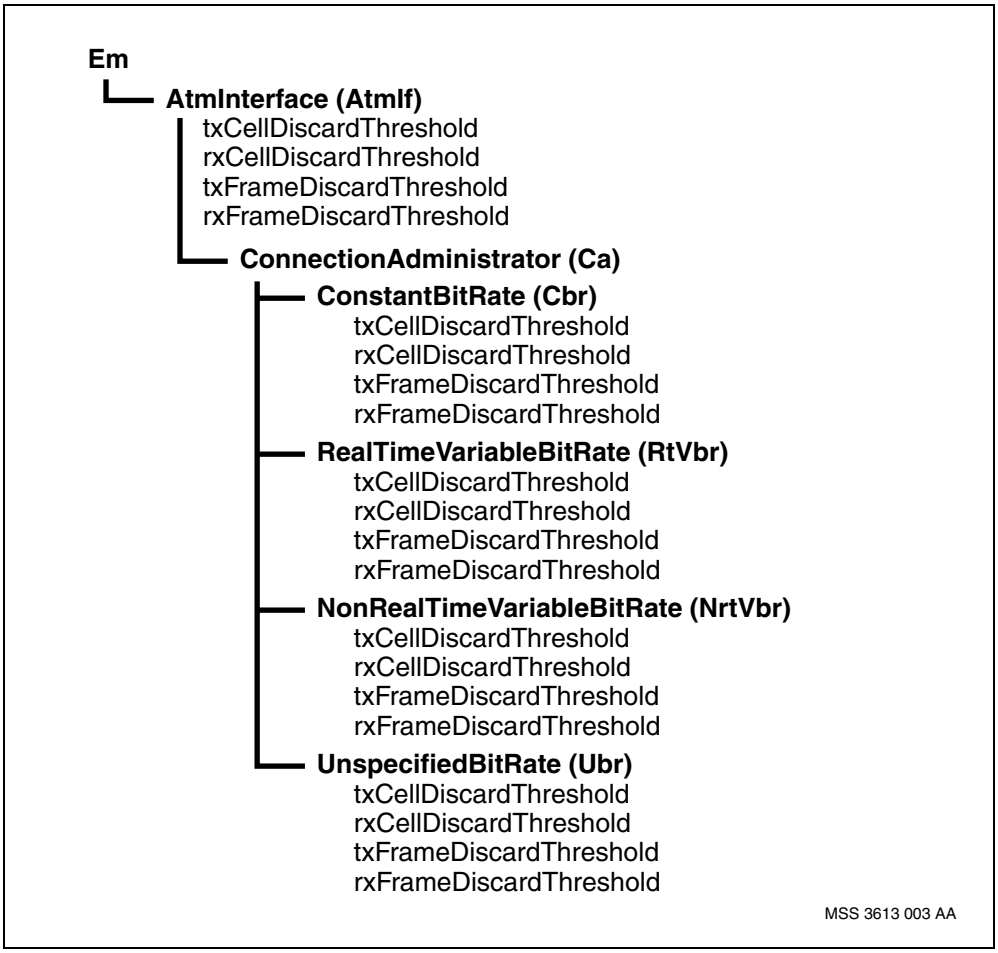
--End--

Variable definitions

Variable	Value
<m>	is the threshold value of the ATM interface and the four service categories discards.
<n>	is the instance value of the *AtmIf* component (any unique value from 1 to 4 095).

Procedure job aid

ATM discard alarm component hierarchy



Virtual path terminator configuration

Configure virtual path terminator to allow Nortel Multiservice Switch nodes to support multiple virtual interfaces on one physical port by associating each virtual interface with a virtual path. This configuration permits you to tunnel SVCs through permanent virtual paths (PVP).

Prerequisites to virtual path terminator configuration

CAUTION

Port test failure for Vpc/0

Do not use an identifier of 0 for a virtual path connection. Setting the *VirtualPathConnection* (*Vpc*) component to 0 results in a port test failure.

WARNING

You must configure VPTs in the VCC space. Each Basic VPT uses two VCCs and each Standard VPT uses three VCCs.

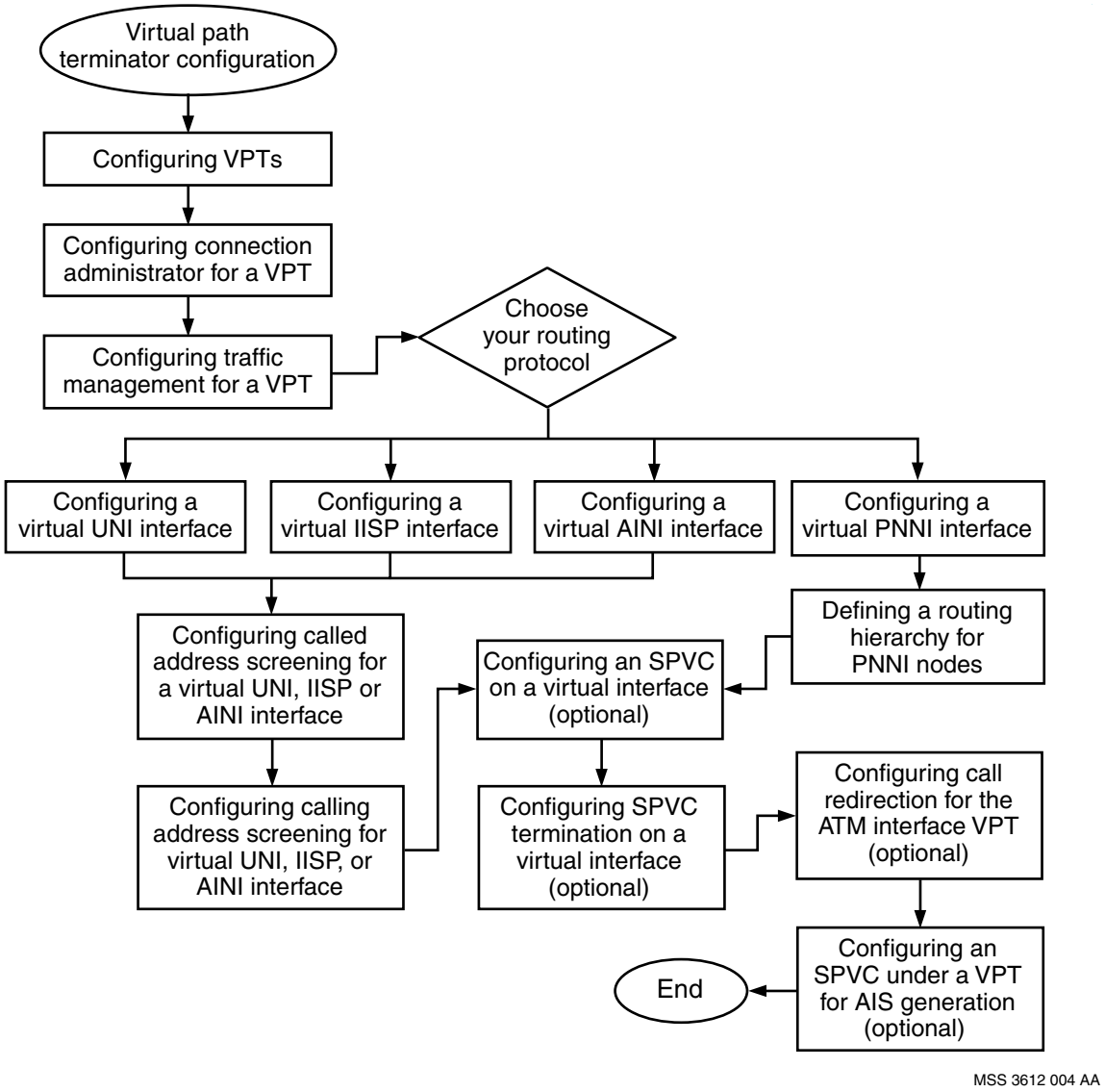
- You must configure *VirtualPathTerminator* components in VCC space. Each Basic VPT uses two VCCs and each Standard VPT uses three VCCs.
- Virtual paths connections and virtual terminators must be configured to support virtual interfaces. A virtual path connection (VPC) or virtual path terminator (VPT) with an identifier of 0 disables the non-associated signaling channel. A semantic check ensures that a *Vpc/0* or *Vpt/0* component cannot exist if there is a *Uni*, *Iisp*, Aini or *Pnni* component.
- In order for connections, including control channels, to be admitted under a Vpt, sufficient bandwidth must be available under that Vpt. The Vpt must be configured with a traffic contract which reserves sufficient bandwidth for the control channels and any dynamic connections setup under the Vpt. A Vpt configured with a UBR service category does not reserve any bandwidth. This can result in connections requiring bandwidth to not be admitted by the Vpt

- A *ConnectionAdministrator* component must exist under a *VirtualPathTerminator* component before you can configure a virtual *Uni*, *Iisp*, Aini or *Pnni* component under that *VirtualPathTerminator* component.

Virtual path terminator configuration procedures

This task flow shows you the sequence of procedures you perform to configure virtual interfaces. To link to any procedure, go to Virtual path terminator configuration procedure navigation (page 68).

Virtual path terminator configuration procedures



Virtual path terminator configuration procedure navigation

Configuring VPTs

Configure VPTs to create a virtual interface.

Prerequisites

- For CQC ATM FPs, the VPI value of the VPT may fall outside the valid connection space for the VPT. If this situation occurs, you must add a connection mapping override component and configure it for the same connection space as that used by VCCs. See Configuring the connection map for CQC-based FPs (page 36) for more information.

Procedure steps

Step	Action
1	Add a virtual path terminator component. **`add AtmIf/<n> Vpt/<Vpi>`**
2	Set the VPT component type. **`set AtmIf/<n> Vpt/<Vpi> Vpd vptType <vptType>`**
3	Set the default loopbacks for the connection. **`set AtmIf/<n> Vpt/<Vpi> Vpd segLinkSideLoopback <segLkLbk>`** **`set AtmIf/<n> Vpt/<Vpi> Vpd endToEndLoopback <eeLbk>`**
4	If required, set the holding priority for the VPT. **`set AtmIf/<n> Vpt/<Vpi> Vpd Tm holdingPriority <hpri>`**

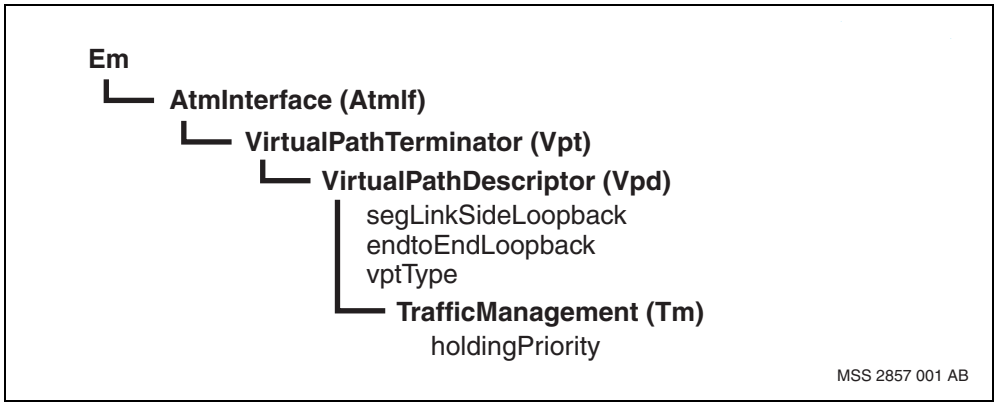
--End--

Variable definitions

Variable	Value
<eeLbk>	specifies if end-to-end loopback insertion and termination should be performed on this connection. This value can be *on*, *off*, or *sameAsInterface*. The default is *sameAsInterface*.
<hpri>	is a decimal from 0 to 4. The default is 2. The decimal 0 indicate highest priority and the decimal 4 indicates lowest priority).
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.

Procedure job aid

VPTs component hierarchy



Configuring connection administrator for a VPT

Configure connection administrator for a VPT to define the maximum number of VCCs under the VPT. The connection administrator is also used to define and configure bandwidth pools for the virtual interface.

Procedure steps

Step	Action
1	Add a connection administrator component for the VPT. **add AtmIf/<n> Vpt/<Vpi> Ca**
2	Set the maximum number of connections for the VPT. This value cannot exceed the total number defined in the *AtmIf/<n> Ca maxVccs* attribute. **set AtmIf/<n> Vpt/<Vpi> Ca maxVccs <vccs>**
3	Set the maximum number of connections for the UBR service category. **set AtmIf/<n> Vpt/<Vpi> Ca Ubr/0 maxVccs <ubrvccs>**
4	Configure the bandwidth connection pool capacity. **set AtmIf/<n> Ca bandwidthPool 1 <percentage1> 2 <percentage2> 3 <percentage3> 4 <percentage4> 5 <percentage5>**
5	Set up the bandwidth connection pool assignments for each applicable service category. **set AtmIf/<n> Vpt/<Vpi> Ca <ServiceCategory>/0 pool <pool>**
6	Set the cell loss ratio (CLR) for each applicable service category. **set AtmIf/<n> Ca <ServiceCategory>/0 cellLossRatio <clr>**
7	Set the cell delay variation tolerance (a parameter that affects the equivalent cell rate) for the CBR service category. **set AtmIf/<n> Ca Cbr/0 cdvt <cdvt>**

--End--

Variable definitions

Variable	Value
<cdvt>	is a numeric value from 1 to 10 000. The default is 250. This value applies to all CBR traffic on the interface. If usage parameter control (UPC) is enabled for the connection, the value in this attribute affects traffic policing configuration. The larger the value, traffic policing is more tolerant of bandwidth use beyond the traffic contract.
<clr>	is a negative logarithmic value (base 10) between 0 and 15. The default is 10 for CBR and RT-VBR service categories, and 7 for the NRT-VBR service category. The value for this attribute must be identical for CBR and RT-VBR service categories because traffic in these service categories have the same discard priority. The value this attribute for the NRT-VBR service category must be less than or equal to the value in the CBR and RT-VBR service categories. In Nortel Multiservice Switch systems, CLR is configured as a power of 10. For example, to configure a CLR of 10^{-10}, the <clrValue> would be 10.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<percentage1>, <percentage2>, <percentage3>, <percentage4>, <percentage5>	are vectors that define the bandwidth pools. These vectors consist of an index entry ranging from 1 to 5 and a decimal entry (<percentage1>, <percentage2>, <percentage3>, <percentage4>, and <percentage5>) that has a value between 0 and 12 800. The decimal entry sets the percentage of link bandwidth allowed in the bandwidth pool defined by the index entry. The default values of this vector are *1 100 2 0 3 0 4 0 5 0*.
<pool>	is pool1, pool2, pool3, pool4, or pool5. The default is pool1 for all service categories.
<ServiceCategory>	is Cbr, RtVbr, or NrtVbr. Only the 0 instance of a service category exists.
<ubrvccs>	is either *sameAsCa* or a decimal in the range of 0 to 16 384. The default is *sameAsCa*. This attribute defines the maximum number of VCCs for the UBR service category.

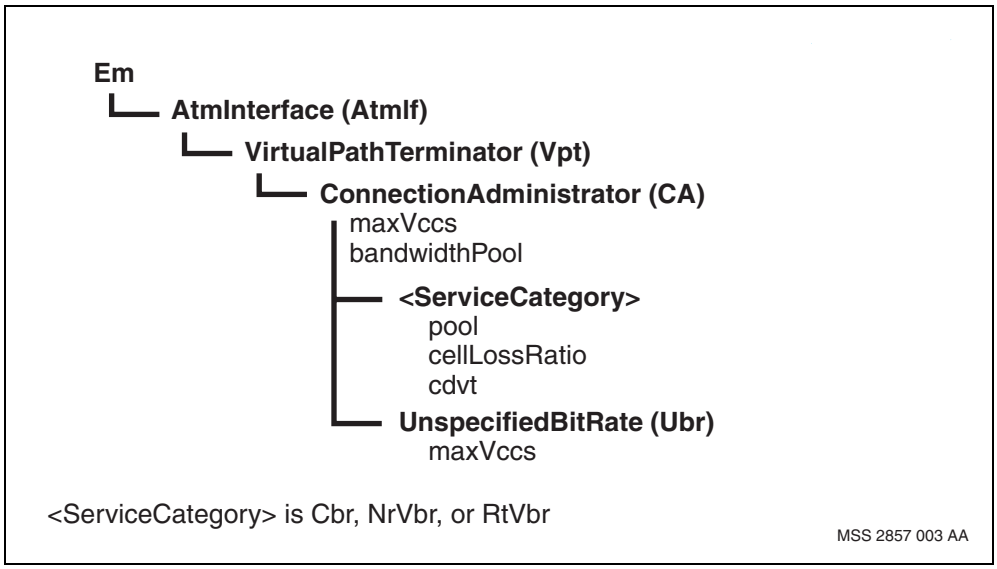
(1 of 2)

Variable	Value
<vccs>	is the maximum number of virtual channel connections that can be configured on this Vpt component. This value is numeric between 0 and 16 384. The default is 255. The sum of all *AtmIf Vpt Ca maxVccs* attributes for an ATM interface must be smaller or equal to the value of the *AtmIf Ca maxVccs* attribute for that ATM interface.
<Vpi>	is the instance value of the *Vpt* component. The value can be 0 - 4 095 for ATM IP cards, 0 - 4094 for 8-port ATM cards, or 0 - 255 for CQC-based cards.

(2 of 2)

Procedure job aid

Connection administrator for a VPT component hierarchy



Configuring traffic management for a VPT

Configure traffic management to smooth traffic bursts by regulating the emission interval of cells or frames in the transmit direction of a virtual path terminator (VPT).

Prerequisites

- For information on traffic shaping and policing concepts, see NN10600-706 *Nortel Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*.

Procedure steps

Step	Action
1	Set the shaping requirements for transmitted traffic to the ATM interface. **set AtmIf/<n> Vpt/<Vpi> Vpd Tm trafficShaping <trShaping>**
2	Set the UPC requirements for received traffic from the ATM interface. **set AtmIf/<n> Vpt/<Vpi> Vpd Tm usageParameterControl <upcValue>**
3	Set the ATM service category for the VPT. **set AtmIf/<n> Vpt/<Vpi> Vpd Tm atmServiceCategory <service>**
4	Set the traffic descriptor type for the transmit direction of the VPT. **set AtmIf/<n> Vpt/<Vpi> Vpd Tm txTrafficDescType <txTdt>**
5	Set the traffic descriptor parameters for the transmit direction of the VPT. **set AtmIf/<n> Vpt/<Vpi> Vpd Tm txTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>] [4 <parm4>] [5 <parm5>]**
6	Set the traffic descriptor type for the receive direction of the VPT. **set AtmIf/<n> Vpt/<Vpi> Vpd Tm rxTrafficDescType <rxTdt>** The default value is usually configured, which uses the traffic descriptor type and traffic descriptor parameters that have been configured for the transmit direction (that is, symmetrical traffic).
7	If you are configuring UPC for standard VPTs, set the traffic descriptor parameters for the receive direction of the VPT. **set AtmIf/<n> Vpt/<Vpi> Vpd Tm rxTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>] [4 <parm4>]** If the value of the *rxTrafficDescType* attribute is *sameAsTx*, the parameters for this attribute are the same as the parameters in the *txTrafficDescParm* attribute.

8 Set per-VC queuing for the VPT.

```
set AtmIf/<n> Vpt/<Vpi> Vpd Tm unshapedTransmitQueueing
<unshapValue>
```

9 Set the holding priority for the VPT.

```
set AtmIf/<n> Vpt/<Vpi> Vpd Tm holdingPriority <hpri>
```

Attention: Set the *holdingPriority* attribute with a different value for each service category to keep higher priority connections during a bandwidth decreases.

10 Configure the fairness weighting for an unshaped VPT.

```
set atmIf/<n> Vpt/<Vpi> Vpd Tm weight <weight>
```

CAUTION
Configuring txQueueLimit may reset the ATM interface
Configuring the *txQueueLimit* attribute is a critical change that causes the ATM interface to reset. Any active call is dropped.

11 Configure the transmit queue limit for the VPT.

```
set atmIf/<n> Vpt/<Vpi> Vpd Tm txQueueLimit <txqlim>
```

Use sameAsCa to derive the queue limit from the configuration for the *ConnectionAdministrator* component. For FPs other than CQC-based FPs, use a decimal to configure a specific limit that is different from the limit set for the *ConnectionAdministrator* component.

--End--

Variable definitions

Variable	Value
<hpri>	is a decimal from 0 to 4. The default is 2. The decimal 0 indicate highest priority and the decimal 4 indicates lowest priority).
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<parm1>	is traffic descriptor parameter 1 (peak cell rate) between 0 and 2 147 483 647. The default is 0.
<parm2>	is traffic descriptor parameter 2 (PCR0 or sustained cell rate) between 0 and 2 147 483 647. The default is 0.
<parm3>	is traffic descriptor parameter 3 (maximum burst size) between 0 and 2 147 483 647. The default is 0.
<parm4>	is traffic descriptor parameter 4 (CDVT) between 0 and 10 000 to 1 200 000. The default is 250. If you are going to enable UPC on the VCC, the traffic descriptor parameters for the receive direction must take UPC into account. When rxTdt is 9 for UBR with MDCR, <parm1> is PCR, <parm2> is CDVT, and <parm3> is MDCR.
<parm5>	is traffic descriptor parameter 5 (requested shaping rate) between 0 and 2 147 483 647. This value does not apply to basic VPTs. The default is 0. When rxTdt is 9 for UBR with a minimum desired cell rate (MDCR), <parm1> is PCR, <parm2> is CDVT, and <parm3> is MDCR.
<rxTdt>	is a value between 1 and 9, or sameAsTx, defining the traffic descriptor type. The default is sameAsTx. Traffic descriptor 9 is to be used exclusively by UBR with MDCR. In addition, UPC is not supported with traffic descriptor 9 because traffic descriptor 9 is only meant to override the minimum cell rate for designated UBR with MDCR connections.
<service>	is Cbr, RtVbr, NrtVbr, or Ubr. The default is Ubr.

(1 of 3)

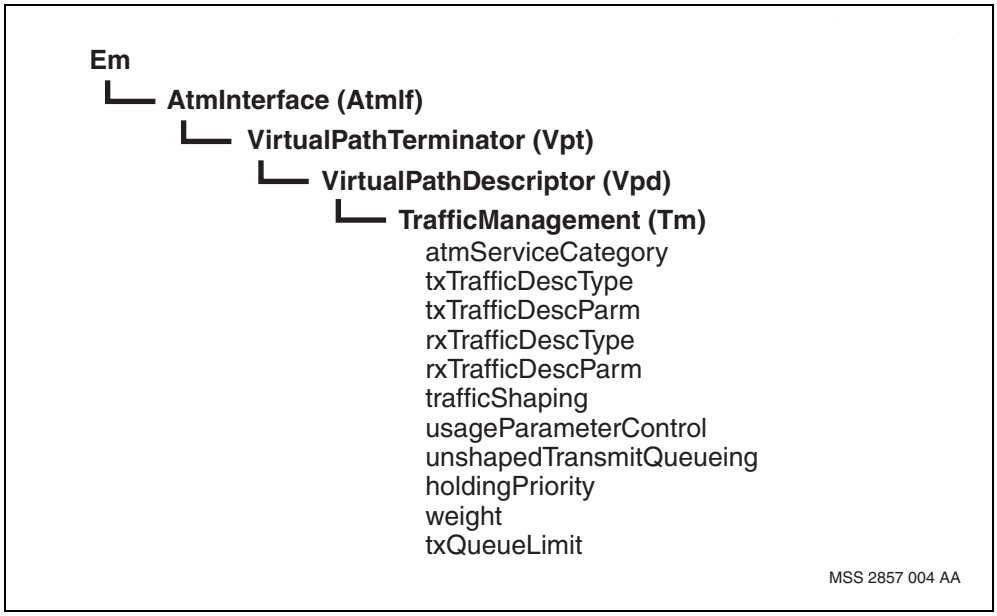
Variable	Value
<trShaping>	is either disabled or *sameAsCa*. The default is *sameAsCa*. Use sameAsCa to derive shaping characteristics from the service category settings under the AtmIf *ConnectionAdministrator* component. If the service category has shaping disabled, software disables shaping for transmitted traffic on VPTs under that service category. The shaping characteristics that software derives from the AtmIf *ConnectionAdministrator* component depends on the hardware configuration. See the table Relationship between the txTrafficDescType and trafficShaping attribute values (page 55) for information on how characteristics apply. Use disabled to disable shaping for transmitted traffic to the ATM interface, regardless of the setting for the service category.
<txqlim>	is either sameAsCa or a decimal from 5 to 63 488 (in cells). Note that for CQC-based FPs, the decimal range is 5 to 30 000 and that for APC-based FPs, the range is 88 to 65 535. The default is sameAsCa.
<unshapValue>	is common or sameAsCa. The default value is sameAsCa. For ATM IP and APC-based FPs, the value must be set to the default value, sameAsCa. Use sameAsCa to apply the default per-VC queuing permissions that you defined for the *AtmIf* component. For FPs other than the ATM IP and APC-based FPs, you can use common to direct to the common queue. This parameter applies to unshaped traffic only. Shaped traffic is always serviced on per-VC queues.
<upcValue>	is e*nforced*, *disabled*, *sameAsCa*, or *monitored*. The default is *sameAsCa*. Use enforced to enable policing for received traffic from the ATM interface regardless of the setting for the service category. Use disabled to disable policing for received traffic from the ATM interface regardless of the setting for the service category. Use *sameAsCa* to derive UPC characteristics from the service category settings under the AtmIf *ConnectionAdministrator* component. If UPC for the service category is disabled, software disables policing for received traffic on connections under that service category. Use monitored to count UPC violations and pass cells unchanged.
(2 of 3)	

Variable	Value
<Vpi>	is the instance value of the *Vpt* component and can be any unique value from 0 to 4 095.
<weight>	is a decimal value from 1 to 4 095, sameAsCa, upToQueueLimit. The default is sameAsCa. For the APC-based FPs, the value, upToQueueLimit, does not apply. Use a decimal value when you need to define relative fairness between a number of connections under the interface. Use sameAsCa to derive the fairness weighting from the configuration for the *ConnectionAdministrator* component. For AQM-based processors, use upToQueueLimit to limit fairness weighting by the transmit queue limit specified in the *txQueueLimit* attribute.

(3 of 3)

Procedure job aid

Traffic management for a VPT component hierarchy



Configuring a virtual UNI interface

Configure a virtual UNI interface to support the UNI protocol.

Procedure steps

Step	Action
1	Add a virtual interface. **`add AtmIf/<n> Vpt/<Vpi> Uni`**

WARNING
Critical change
Changing the value of the *vpci* attribute is a critical change and causes the node to take down all existing SVCs and SPVCs under the VPT you are configuring.

2	Configure the virtual path connection identifier (VPCI) value for the interface. **`set AtmIf/<n> Vpt/<Vpi> Uni vpci <vpci_value>`**
3	Set the interface version. **`set AtmIf/<n> Vpt/<Vpi> Uni version <ver>`**
4	Configure the interface as either the user side or the network side. **`set AtmIf/<n> Vpt/<Vpi> Uni side <ifside>`** Configure the peer interface as the opposite side. If the connection maps for peer interfaces differ, configure the side with the smaller connection map as network. This configuration avoids call setup failures due to mismatching. The connection map for an ATM interface applies to all UNI/IISP/PNNI interfaces (both actual and virtual) under the ATM interface. It may not be possible to adjust the connection map to remove a mismatch condition.
5	Add the required static addresses for the interface. **`set AtmIf/<n> Vpt/<Vpi> Uni Address/ <address>,<addr_type>`** Because there is no default address, you must configure a static address if the virtual interface terminates an SPVC.

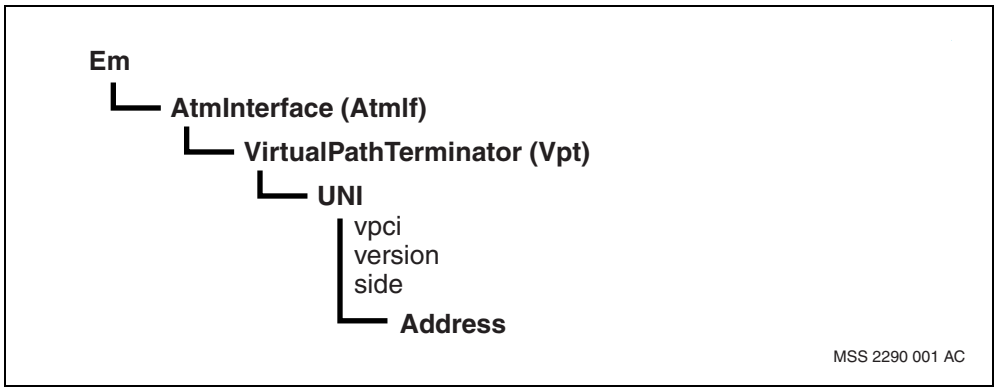
--End--

Variable definitions

Variable	Value
<addr_type>	is either primary or alternate.
<address>	is a static address associated with the interface, consisting of either up to 40 hexadecimal digits or a single wild card character (the? symbol). You can configure multiple static address for each interface, but all address entries must be unique on this interface.
<ifside>	is either user or network. The default is network. The side of the interface must be balanced by the value of the side attribute for the interface at the other end. For example, if you define this interface as network side, define the peer interface as user side.
<n>	is a decimal value from 1 to 4 095, representing the number of the ATM interface.
<ver>	is either atmForum30, atmForum31or atmForum40, if the type is Uni. The default is atmForum40. The setting must be compatible with the version setting on the peer interface. is either iisp10Sig30 or iisp10Sig31, if the type is Iisp. The default is iisp10Sig31. The setting must be compatible with the version setting on the peer interface.
<vpci_value>	is a decimal value from 0 to 255. The default value is 0. The peer interface must have the same VPCI for SVCs or SPVCs to set up across the virtual path.
<Vpi>	is the number of the VPT that you added in step 1 of this procedure. This must be a value between 0-255.

Procedure job aid

Virtual UNI interface component hierarchy



Configuring a virtual IISP interface

Configure a virtual IISP interface to support the IISP protocol.

Procedure steps

Step	Action
1	Add a virtual interface. **`add AtmIf/<n> Vpt/<Vpi> Iisp`**

WARNING
Critical change
Changing the value of the *vpci* attribute is a critical change and causes the node to take down all existing SVCs and SPVCs under the VPT you are configuring.

2	Configure the virtual path connection identifier (VPCI) value for the interface. **`set AtmIf/<n> Vpt/<Vpi> Iisp vpci <vpci_value>`**
3	Set the interface version. **`set AtmIf/<n> Vpt/<Vpi> Iisp version <ver>`**
4	Configure the interface as either the user side or the network side. **`set AtmIf/<n> Vpt/<Vpi> Iisp side <ifside>`**
5	Configure the peer interface as the opposite side. If the connection maps for peer interfaces differ, configure the side with the smaller connection map as network. This configuration avoids call setup failures due to mismatching. The connection map for an ATM interface applies to all UNI/IISP/PNNI interfaces (both actual and virtual) under the ATM interface. It may not be possible to adjust the connection map to remove a mismatch condition.
6	Add the required static addresses for the interface. **`set AtmIf/<n> Vpt/<Vpi> Iisp Address/ <address>,<addr_type>`** Because there is no default address, you must configure a static address if the virtual interface terminates an SPVC.

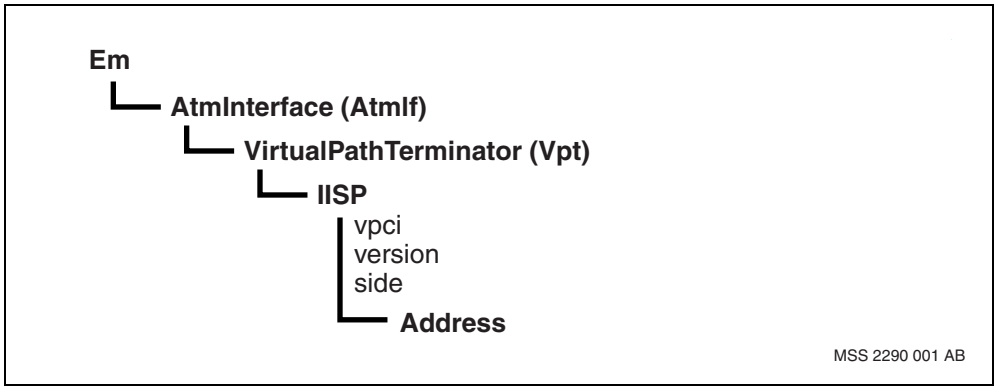
--End--

Variable definitions

Variable	Value
<addr_type>	is either primary or alternate.
<address>	is a static address associated with the interface, consisting of either up to 40 hexadecimal digits or a single wild card character (the? symbol). You can configure multiple static address for each interface, but all address entries must be unique on this interface.
<ifside>	is either user or network. The default is network. The side of the interface must be balanced by the value of the side attribute for the interface at the other end. For example, if you define this interface as network side, define the peer interface as user side.
<IfType>	*is Uni, Iisp,* or *Aini*.
<n>	is a decimal value from 1 to 4 095, representing the number of the ATM interface.
<ver>	is either atmForum30, atmForum31or atmForum40, if the type is Uni. The default is atmForum40. The setting must be compatible with the version setting on the peer interface.
	is either iisp10Sig30 or iisp10Sig31, if the type is Iisp. The default is iisp10Sig31. The setting must be compatible with the version setting on the peer interface.
<vpci_value>	is a decimal value from 0 to 255. The default value is 0. The peer interface must have the same VPCI for SVCs or SPVCs to set up across the virtual path.
<Vpi>	is the number of the VPT that you added in step 1 of this procedure. This must be a value between 0-255.

Procedure job aid

Virtual IISP interface component hierarchy



Configuring a virtual AINI interface

Configure a virtual AINI interface to support the AINI protocol.

Procedure steps

Step	Action
1	Add a virtual interface. **`add AtmIf/<n> Vpt/<Vpi> Aini`**
	 **WARNING** **Critical change** Changing the value of the *vpci* attribute is a critical change and causes the node to take down all existing SVCs and SPVCs under the VPT you are configuring.
2	Configure the virtual path connection identifier (VPCI) value for the interface. **`set AtmIf/<n> Vpt/<Vpi> Aini vpci <vpci_value>`**
3	Set the interface version. **`set AtmIf/<n> Vpt/<Vpi> Aini version <ver>`**
4	Configure the interface as either the user side or the network side. **`set AtmIf/<n> Vpt/<Vpi> Aini side <ifside>`**
5	Configure the peer interface as the opposite side. If the connection maps for peer interfaces differ, configure the side with the smaller connection map as network. This configuration avoids call setup failures due to mismatching. The connection map for an ATM interface applies to all UNI/IISP/PNNI interfaces (both actual and virtual) under the ATM interface. It may not be possible to adjust the connection map to remove a mismatch condition.
6	Add the required static addresses for the interface. **`set AtmIf/<n> Vpt/<Vpi> Aini Address/ <address>,<addr_type>`** Because there is no default address, you must configure a static address if the virtual interface terminates an SPVC.

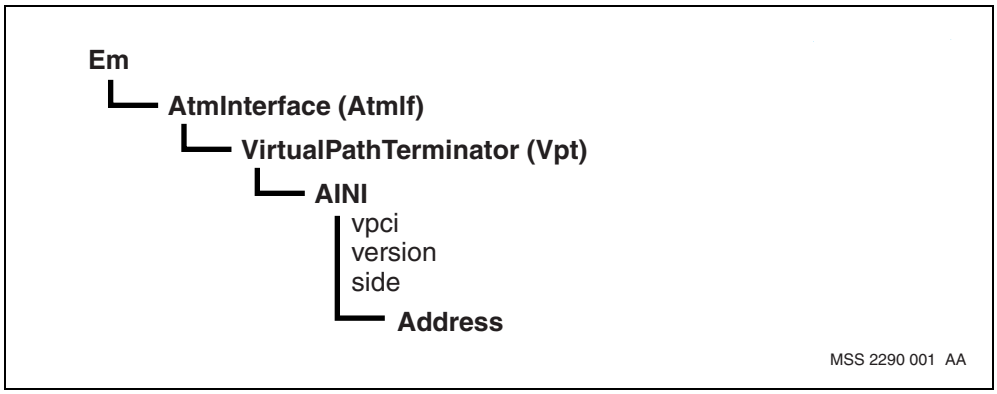
--End--

Variable definitions

Variable	Value
<addr_type>	is either primary or alternate.
<address>	is a static address associated with the interface, consisting of either up to 40 hexadecimal digits or a single wild card character (the? symbol). You can configure multiple static address for each interface, but all address entries must be unique on this interface.
<ifside>	is either user or network. The default is network. The side of the interface must be balanced by the value of the side attribute for the interface at the other end. For example, if you define this interface as network side, define the peer interface as user side.
<n>	is a decimal value from 1 to 4 095, representing the number of the ATM interface.
<ver>	is either atmForum30, atmForum31or atmForum40, if the type is Uni. The default is atmForum40. The setting must be compatible with the version setting on the peer interface. is either iisp10Sig30 or iisp10Sig31, if the type is Iisp. The default is iisp10Sig31. The setting must be compatible with the version setting on the peer interface.
<vpci_value>	is a decimal value from 0 to 255. The default value is 0. The peer interface must have the same VPCI for SVCs or SPVCs to set up across the virtual path.
<Vpi>	is the number of the VPT that you added in step 1 of this procedure. This must be a value between 0-255.

Procedure job aid

Virtual AINI interface component hierarchy



Configuring a virtual PNNI

Configure a virtual PNNI interface to support the PNNI protocol.

Procedure steps

Step	Action
1	Add the component for the virtual interface. **`add AtmIf/<n> Vpt/<Vpi> Pnni`**

WARNING
Critical change
Changing the value of the *vpci* attribute is a critical change and causes the node to take down all existing SVCs and SPVCs under the VPT you are configuring.

2	Configure the VPCI value for the interface. **`set AtmIf/<n> Vpt/<Vpi> Pnni vpci <vpci>`**
3	Add the required static addresses for the interface. Because there is no default address, you must configure a static address if the virtual interface terminates calls. **`set AtmIf/<n> Vpt/<Vpi> Pnni Address/ <address>,<addr_type>`**

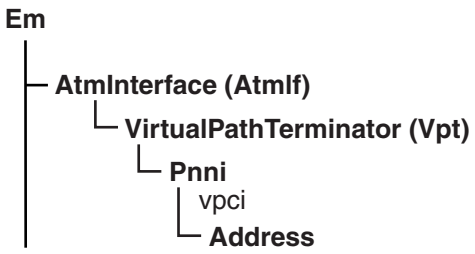
--End--

Variable definitions

Variable	Value
<address>	is a static address associated with the interface, consisting of either, up to 40 hexadecimal digits, or a single wild card character (the? symbol). You can configure multiple static address for each interface, but all address entries must be unique on this interface.
<addr_type>	is primary or alternate.
<n>	i s a decimal value from 1 to 4 095, representing the number of the ATM interface.
<vpci>	is a value from 0 to 255. The default value is 0. The peer interface must have the same VPCI for all SVCs and SPVCs to be able to be set up across the virtual path.
<Vpi>	is the number of the VPT that you added in step 1 of this procedure. The value can be 0 to 4 095 for ATM IP cards, 0 to 4 094 for 8-port ATM cards, or 0 to 255 for CQC cards.

Procedure job aid

Virtual PNNI component hierarchy



Configuring called address screening for a virtual UNI, IISP or AINI interface

Configure called address screening for a virtual UNI, IISP or AINI interface to screen outgoing calls against configured addresses.

By default, called address screening is disabled, which means that calls are not screened. When you enable called address screening, outgoing calls are screened against the addresses that you have configured. The outgoing call is accepted if the called address matches an address that you have configured to be accepted. The outgoing call is rejected if the called address is either not configured, matches an address you have configured to be rejected, or does not match any configured address.

Procedure steps

Step	Action
1	Configure the interface for called address screening. **`add AtmIf/<n> Vpt/<vpi> <IfType> CalledAScr`**
2	Add an ATM address to the screening rules. **`add AtmIf/<n> Vpt/<vpi> <IfType> CalledAScr Address/ <address>,<action>`**
3	Enable called address screening for this ATM interface. **`set AtmIf/<n> Vpt/<vpi> <IfType> CalledAScr adminStatus enabled`**

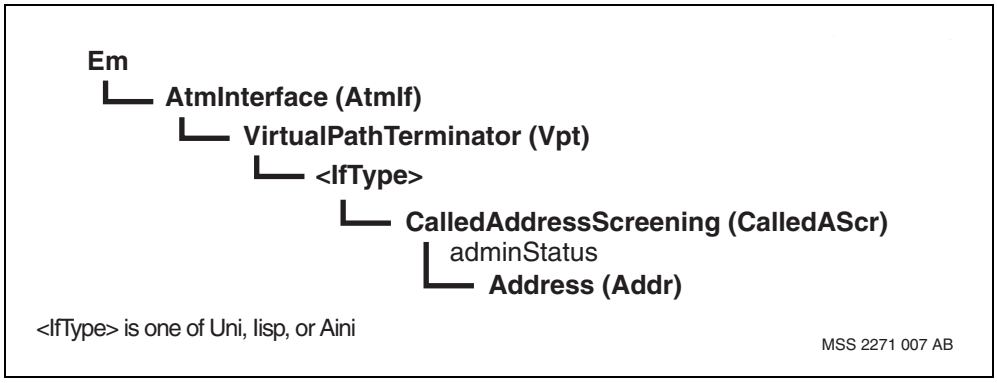
--End--

Variable definitions

Variable	Value
<action>	is accept or reject.
<address>	is a destination ATM address.
<IfType>	is UNI, IISP, or AINI.
<n>	is the number of the ATM interface.
<vpi>	is the number of the VPT.

Procedure job aid

Called address screening for a virtual UNI, IISP or AINI component hierarchy



Configuring calling address screening for a virtual UNI, IISP, or AINI interface

Configure calling address screening for a virtual UNI, IISP, or AINI interface to screen incoming calls against configured addresses.

By default, calling address screening is disabled, which means that calls are not screened.

When you enable calling address screening, incoming calls are screened against the addresses that you have configured. The incoming call is accepted if the calling address matches an address that you have configured to be accepted. The incoming call is rejected if the calling address matches an address you have configured to be rejected, does not match a configured address, or the connection request does not contain a calling address and there is no configured default insertion address.

Procedure steps

Step	Action
1	Configure the interface for calling address screening. **`add AtmIf/<n> Vpt/<vpi> <IfType> CallingAScr`**
2	Specify a default address that is to be inserted into the call connection request if calling address screening is enabled and the connection request does not contain a calling address. **`set AtmIf/<n> Vpt/<vpi> <IfType> CallingAScr defaultInsertionAddress <defInsAddr>`**
3	Add an ATM address to the screening rules. **`add AtmIf/<n> Vpt/<vpi> <IfType> CallingAScr Address/ <address>,<action>`**
4	Enable calling address screening for this ATM interface. **`set AtmIf/<n> Vpt/<vpi> <IfType> CallingAScr adminStatus enabled`**

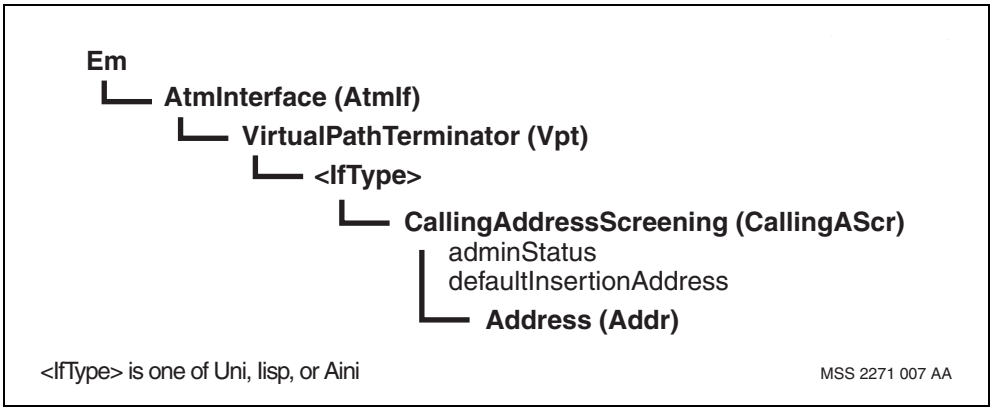
--End--

Variable definitions

Variable	Value
<action>	is accept or reject.
<address>	is a source ATM address.
<defInsAddr>	is the default insertion address that needs to be 0 or 20 bytes (40 hexadecimal digits) in length.
<IfType>	is UNI, IISP, or AINI.
<n>	is the number of the ATM interface.
<vpi>	is the number of the VPT.

Procedure job aid

Calling address screening for virtual UNI, IISP, or AINI component hierarchy



Defining a routing hierarchy for PNNI nodes

Define a routing hierarchy for a PNNI node by changing the default address scope.

Procedure steps

Step	Action
1	Add a PNNI information component so that you can associate information with the ATM address. **add AtmIf/<n> Vpt/<m> <IfType> Address <address>,<addr_type> PnniInfo**
2	Define the scope of the ATM address. **set AtmIf/<n> Vpt/<m> Pnni Address <address>,<addr_type> PnniInfo scope <level>**

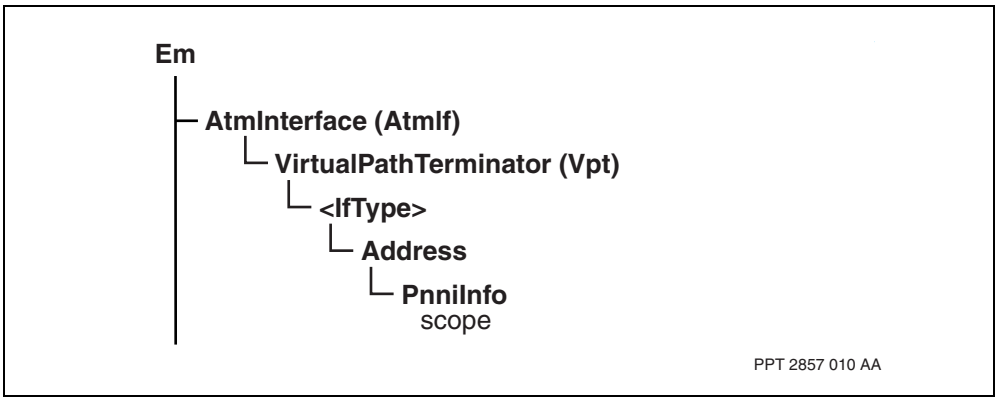
--End--

Variable definitions

Variable	Value
<addr_type>	defines the address type as primary or alternate.
<address>	is a static address associated with the interface, consisting of either, up to 40 hexadecimal digits, or a single wild card character (the? symbol). You can configure multiple static addresses for each interface, but all address entries must be unique on this interface.
<IfType>	is Uni, Iisp, Aini, or Pnni.
<level>	is a value between -1 and 104.
<m>	is the number of the VPT.
<n>	is a decimal value from 1 to 4 095, representing the number of the ATM interface.

Procedure job aid

Routing hierarchy for PNNI nodes component hierarchy



Configuring an SPVC on a virtual interface

Configure an SPVC on a virtual interface to create a switched connection to another interface.

Procedure steps

Step	Action
1	Add the *SourcePvc* component to the connection, if necessary. **add Atmif/<n> vpt/<vpi> vcc/<vci> SourcePvc**
2	Identify the destination end for this SPVC by specifying its address. **set AtmIf/<n> vpt/<vpi> vcc/<vci> SourcePvc calledAddress <calledA>**
3	Identify the VPI.VCI associated with the called (or destination) interface. **set AtmIf/<n> vpt/<vpi> vcc/<vci> SourcePvc calledVpiVci <calledVpiVci>**
4	Identify the source end for this SPVC by specifying its address. **set AtmIf/<n> vpt/<vpi> vcc/<vci> SourcePvc callingAddress <callingA>**

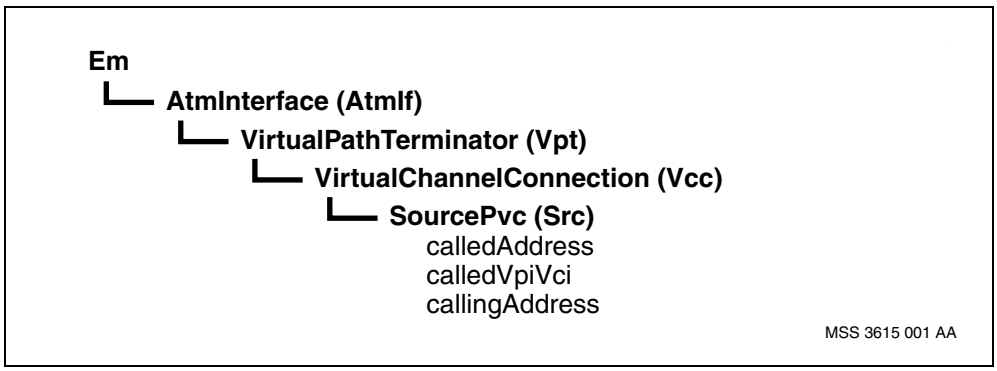
--End--

Variable definitions

Variable	Value
<calledA>	is a default address associated with the destination interface, consisting of 40 hexadecimal digits.
<calledVpiVci>	is the instance value of the virtual channel associated with the remote end-point. The VPI value can be from 0 to 255. The VCI value can be from 32 to 65 535. For Apc based cards (4pOC12 and 16pOC3), the value can be from 32 to 16 383. This VPI.VCI must be within the defined connection map address space on the remote interface.
<callingA>	is the address associated with the source interface, and consists of 40 hexadecimal digits. If you do not specify an address value, the node uses the default address value.
<n>	is the instance value of the Atmif component.
<vci>	is the instance value of the vcc.
<vpi>	is the instance value of the vpt.

Procedure job aid

SPVC on a virtual interface component hierarchy



Configuring SPVC termination on a virtual interface

Configure SPVC termination on a virtual interface to define the node address as a VPT point for an SPVC.

Procedure steps

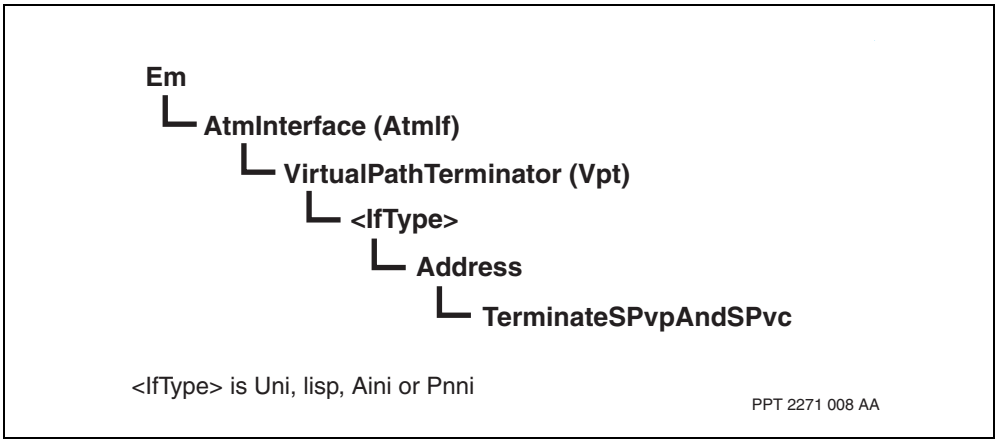
Step	Action
1	Associate a terminate component with the address. **`add AtmIf/<n> Vpt/<m> <IfType> Address <address>,<addr_type> TerminateSPvpAndSPvc`**
	--End--

Variable definitions

Variable	Value
<address>	is a static address associated with the interface, consisting of either, up to 40 hexadecimal digits, or a single wild card character (the? symbol). You can configure multiple static addresses for each interface, but all address entries must be unique on this interface.
<addr_type>	is the address type (either primary or alternate).
<IfType>	is Uni, Iisp, Aini, or Pnni.
<m>	is the number of the VPT that you added.
<n>	is a decimal value from 1 to 4 095, representing the number of the ATM interface.

Procedure job aid

SPVC termination on a virtual interface component hierarchy



Configuring call redirection for the ATM interface VPT

Configure call redirection for the ATM interface VPT to add a secondary destination address to be used if the primary destination connection fails.

After the call redirection for the ATM interface virtual path terminator is configured, the lock and unlock capability of the *CallRedirection* component allows you to include or exclude the use of the primary or secondary destination address during call setup attempts.

Procedure steps

Step	Action
1	Add the *CallRedirection* component. **`add Atmif/<n> vpt/<vpi> vcc/<vci> SourcePvc CallRedirection`**
2	Set the secondary called address. **`set Atmif/<n> vpt/<vpi> vcc/<vci> SourcePvc CallRedirection secondaryCalledAddress <sec_address>`**
3	Set the secondary called vpi. **`set Atmif/<n> vpt/<vpi> vcc/<vci> SourcePvc CallRedirection secondaryCalledVpiVci <sec_vpi.sec_vci>`**

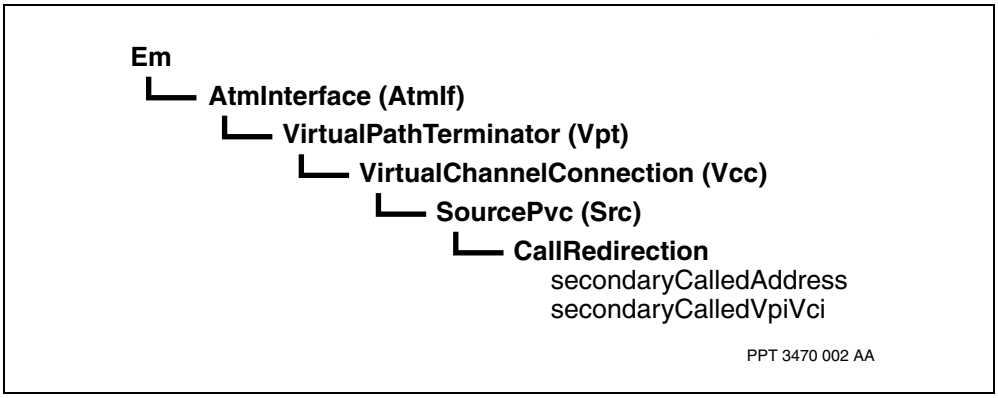
--End--

Variable definitions

Variable	Value
<n>	is the instance value of the Atmif component.
<sec_address>	is the value of the secondary called address attribute.
<sec_vpi.sec_vci>	is the value of the secondary called vpi.vci.
<vci>	is the instance value of the vcc.
<vpi>	is the instance value of the vpt.

Procedure job aid

Configuring call redirection for Atmif vpt component hierarchy



Configuring an SPVC under a VPT for AIS generation

Configure an SPVC under a VPT for AIS generation to enable alarm indication signaling (AIS).

Prerequisites

- To avoid service degradation and potential traffic loss on a given SPVC connection, ensure that the configured traffic management parameters on a configured destination end match the traffic management parameters configured at the source end. All Tx (transmit) parameters at the destination end correspond to the Rx (receive) parameters at the source end. All Tx parameters from the source end must correspond to the Rx parameters at the destination end.
- If you want AIS generation enabled at both ends, both the source and destination must be enabled.
- If you want AIS generation disabled at both ends, both the source and destination must be disabled.

Procedure steps

Step	Action
1	Add a destination SPVC under a Vpt. **add atmif/<n> vpt/<m> vcc/<vci> dst**
2	Enable AIS at the source SPVC under a Vpt. **set atmif/<n> vpt/<m> vcc/<vci> src aisGeneration enable**
3	Enable AIS at the destination SPVC under a Vpt. **set atmif/<n> vpt/<m> vcc/<vci> dst config aisGeneration enable**
4	Disable AIS at the source SPVC under a Vpt **set atmif/<n> vpt/<m> vcc/<vci> src aisGeneration disable**
5	Disable AIS at the destination SPVC under a Vpt. **set atmif/<n> vpt/<m> vcc/<vci> dst config aisGeneration disable**

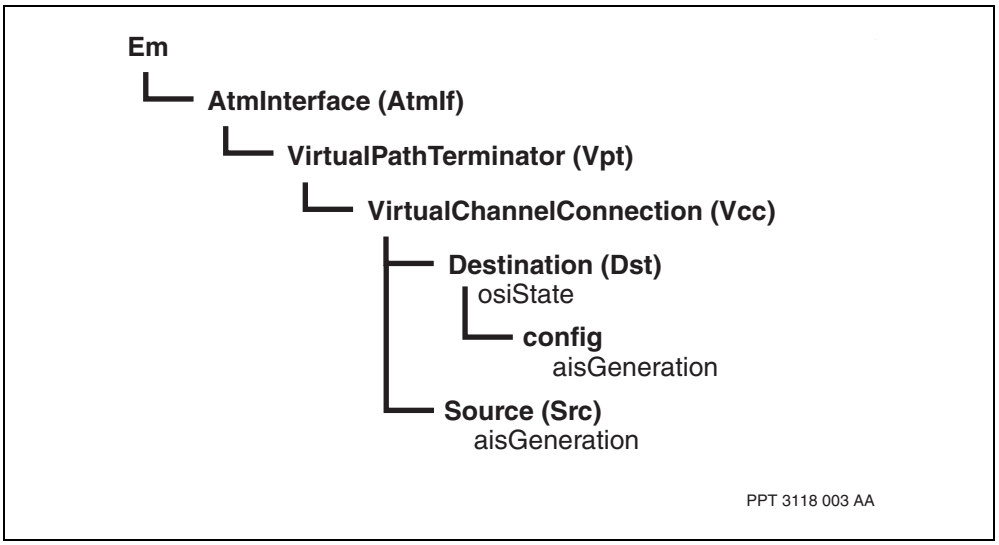
--End--

Variable definitions

Variable	Value
<m>	is the number of the VPT.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<vci>	is the instance value of the Vcc.

Procedure job aid

SPVC under a Vpt for AIS generation component hierarchy



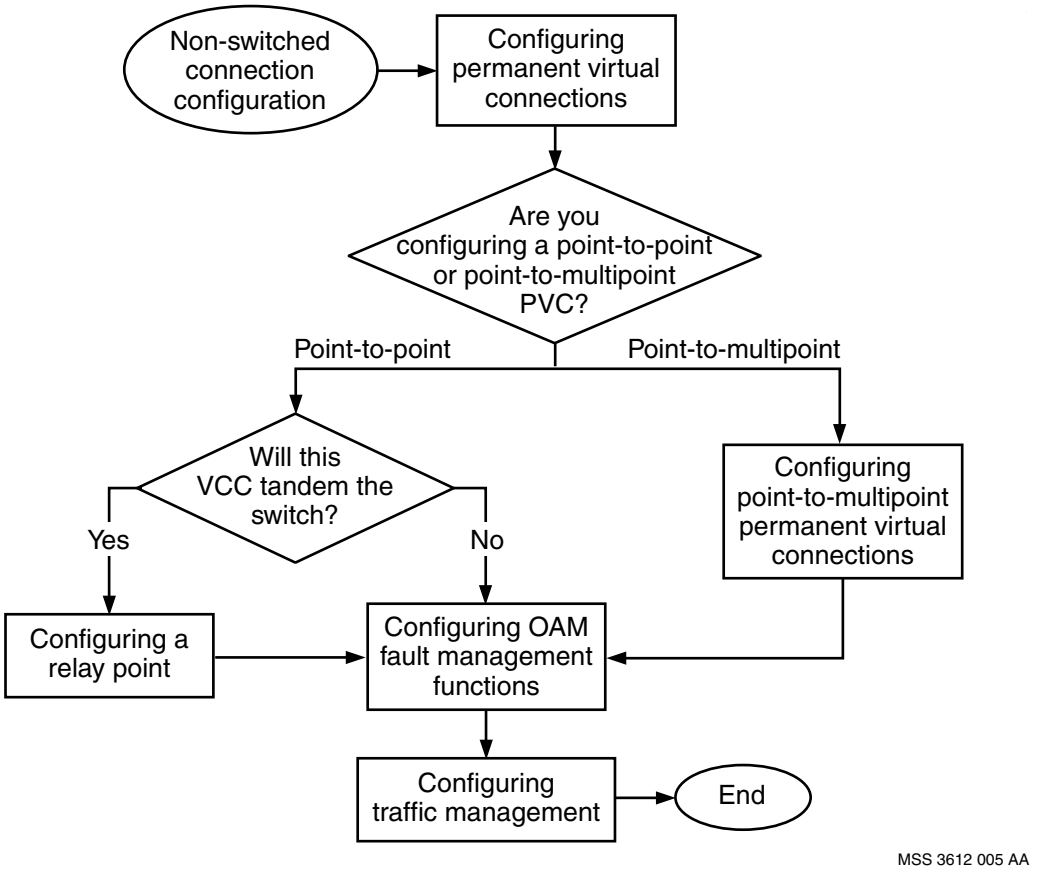
Non-switched connection configuration

Configure non-switched connections to configure virtual connections using a predefined static route to provide a permanently configured connection that remains set up even when not in use.

Non-switched connection configuration procedures

This task flow shows you the sequence of procedures you perform to configure non-switched connections. To link to any procedure, go to Non-switched connection configuration procedure navigation (page 101).

Non-switched connection configuration procedures



Non-switched connection configuration procedure navigation

Configuring permanent virtual connections

Configure permanent virtual connections, PVC or PVP to create an ATM connection between ATM network elements.

Procedure steps

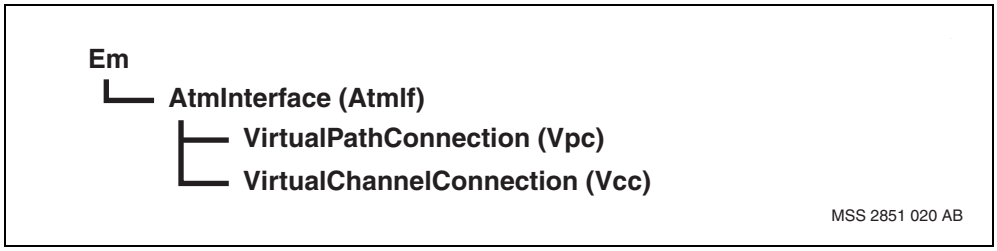
Step	Action
1	Add a virtual channel connection component. **`add AtmIf/<n> <connection_type>/<x>`**
	--End--

Variable definitions

Variable	Value
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<x>	is the instance of the *Vcc* or *Vpc* component. For VCCs, the instance value defines the VPI and VCI values for this connection. For VPCs, the instance value defines the VPI value for this connection.

Procedure job aid

Permanent virtual connections component hierarchy



Configuring a relay point

Configure a relay point to allow a virtual connection to tandem a Nortel Multiservice Switch.

Procedure steps

Step	Action
1	Add a *Nrp* component under the each connection component. **`add AtmIf/<n> <connection_type>/<x> Nrp`**
2	Link the two *Nrp* components together. **`set AtmIf/<n> <connection_type>/<x> Nrp nextHop AtmIf/ <n2> <connection_type>/<x2> Nrp`**

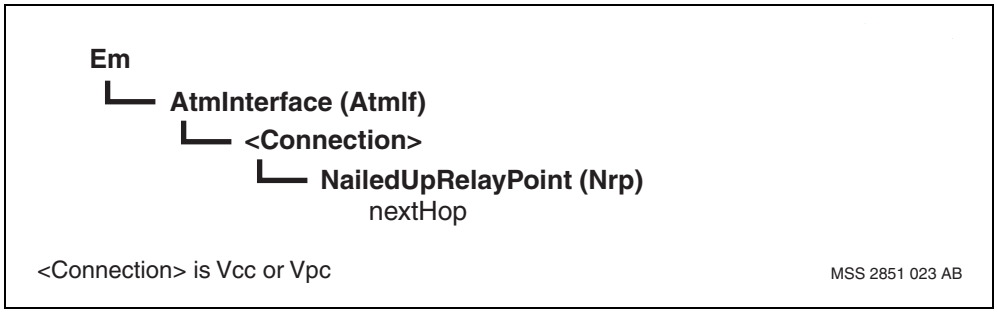
--End--

Variable definitions

Variable	Value
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<n2>	is the second instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<x>	is the instance of the *Vcc* or *Vpc*. For VCCs, the instance value defines the VPI and VCI values for this connection. For VPCs, the instance value defines the VPI value for this connection.
<x2>	is the second instance of the *Vcc* or *Vpc.*

Procedure job aid

Relay point component hierarchy



Configuring point-to-multipoint permanent virtual connections

Configure point-to-multipoint permanent virtual connections (PVC or PVP) to establish a connection with one source and many destinations.

Prerequisites

- For a listing of FPs providing point-to-multipoint connections capability, see NN10600-702 *Nortel Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals*.
- For supporting information, see the figure Point-to-multipoint connections (page 228).

Procedure steps

Step	Action
1	Add an *MNrp* component under the *Vcc* or *Vpc* component for the root side of the connection. **`add AtmIf/<n> <connection_type>/<x> MNrp`**
2	Set *mCastConnectionType* component under the *vcd* component. **`set Atmif/<n> <connection_type>/<x> <descriptor_type> mCastConnectionType pointToMultipointRoot`**
3	Add a *Nrp* component under each connection component that will act as a leaf connection. **`add AtmIf/<n> <connection_type>/<x> Nrp`**
4	Set *mCastConnectionType* component under each leaf's *vcd* component. **`set AtmIf/<n> <connection_type>/<x> <descriptor_type> mCastConnectionType pointtoMultipointLeaf`**
5	Link the root *MNrp* component to each leaf *Nrp* component. **`set AtmIf/<n> <connection_type>/<x> MNrp nextHop AtmIf/ <n2> <connection_type>/<x2> Nrp`**
6	Repeat step 5 for each leaf component.

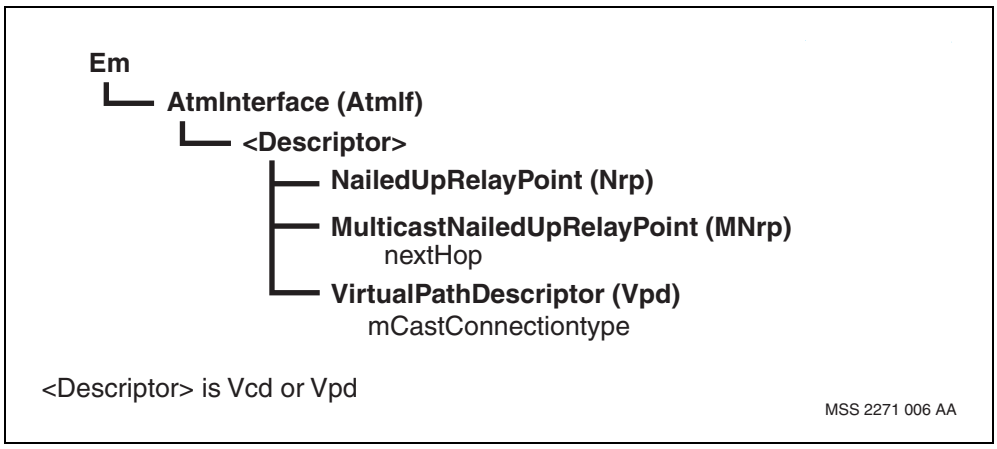
--End--

Variable definitions

Variable	Value
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<descriptor_type>	is either *VirtualPathDescriptor* for Vpcs or VirtualChannelDescriptor for Vccs.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<n2>	is the second instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<x>	is the instance of the *Vcc* or *Vpc*. For VCCs, the instance value defines the VPI and VCI values for this connection. For VPCs, the instance value defines the VPI value for this connection.
<x2>	is the second instance of the *Vcc* or *Vpc.*

Procedure job aid

Point-to-multipoint permanent virtual channels component hierarchy



Configuring OAM fault management functions

Configure OAM fault management functions to allow an operator to determine the status of a network.

Procedure steps

Step	Action
1	Set the *segLinkSideLoopback* for the channel. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> segLinkSideLoopback <segLkLbk>**
2	Set the *segSwitchSideLoopback* for the channel. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> segSwitchSideLoopback <segSwLbk>**
3	Set the *endToEndLoopback* for the channel. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> endToEndLoopback <eeLbk>**
4	Set the OAM segment boundary. **set AtmIf/<n> <connection_type>/<x> Nrp oamSegmentBoundary <sb>**

--End--

Variable definitions

Variable	Value
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<descriptor_type>	is either *VirtualPathDescriptor* for Vpcs or VirtualChannelDescriptor for Vccs.
<eeLbk>	specifies whether end-to-end loopback insertion and termination should be performed on this connection. This value can be on, off, or *sameAsInterface*. The default is *sameAsInterface*.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<sb>	is the OAM segment boundary desired for a Nrp component.

(1 of 2)

Variable	Value
<segLkLbk>	specifies if link-side segment loopback insertion and termination should be performed on this connection. This value can be on, off, or sameAsInterface. The default is sameAsInterface.
<segSwLbk>	specifies if device-side segment loopback insertion and termination should be performed on this connection. This value can be on, off, or sameAsInterface. The default is sameAsInterface.
<x>	is the instance of the *Vcc*. If the virtual channel is associated with an *AtmIf* component, <x> represents the VPI.VCI value. If the virtual channel is associated with a *Vpt* component, <x> represents the VCI value.
(2 of 2)	

Configuring traffic management

Configure traffic management to enable shaping and policing for virtual channel.

Traffic shaping is not possible for CBR connections on a CQC-based FP.

Prerequisites

- You must know the type of FP and value of the *txTrafficDescType* attribute in the connection (*Vpc*, *Vpt*, or *Vcc*). Use the information in the table Relationship between the txTrafficDescType and trafficShaping attribute values (page 55) when configuring traffic shaping for service categories. Inverse UPC traffic shaping is only available on ATM IP FPs with 6, 7, or 8 in the *txTrafficDescType* attribute.
- You must know the value of the *rxTrafficDescType* attribute in the connection (*Vpc*, *Vpt*, or *Vcc*). Use the information in the table Relationship between the txTrafficDescType and trafficShaping attribute values (page 55) when configuring usage parameter control (UPC) for service categories.
- For information on traffic shaping and policing concepts, see NN10600-706 *Nortel Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*.
- For information on traffic management concepts, see NN10600-705 *Nortel Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*.

Procedure steps

Step	Action
1	Set the ATM service category for the channel. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm atmServiceCategory <atmS>**
2	Set the traffic descriptor type for the transmit direction of the channel. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm txTrafficDescType <txTdt>**
3	Set the traffic descriptor parameters for the transmit direction of the channel. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm txTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>] [4 <parm4>] [5 <parm5>]**
4	Set the traffic descriptor type for the receive direction of the channel. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm rxTrafficDescType <rxTdt>**

5 Set the traffic descriptor parameters for the receive direction of the channel.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
rxTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>] [4
<parm4>]
```

6 Set the shaping requirements for transmitted traffic to the ATM interface.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
trafficShaping <trshaping>
```

7 Set the UPC requirements for received traffic from the ATM interface.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
usageParameterControl <upc>
```

8 Set per-VC queuing for the channel.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
unshapedTransmitQueueing <unshap>
```

9 Set packet wise discard for traffic in the transmit direction.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
txPacketWiseDiscard <txpd>
```

10 Optionally, for PVP and PVC connections on Nortel Multiservice Switch 15000 and Multiservice Switch 20000 4-port OC-3, GQM-based, and QRD-based FPs and Multiservice Switch 7400 2-port OC-3 and MSA32mtp FPs, enable weighted random early detection (WRED) and set its parameters.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
txWredMode <mode>
```

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
txWredThreshold <threshold>
```

Even if enabled, WRED is only active for a connection if txPacketWiseDiscard is enabled.

11 Set packet wise discard for traffic in the receive direction.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
rxPacketWiseDiscard <rxpd>
```

12 Set the holding priority for the channel.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
holdingPriority <hPri>
```

Attention: Set the *holdingPriority* attribute with a different value for each service category to keep higher priority connections during a bandwidth decreases.

13 Configure the fairness weighting for an unshaped channel.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
weight <weight>
```

CAUTION
Configuring txQueueLimit may reset the ATM interface
Configuring the *txQueueLimit* attribute is a critical change that causes the ATM interface to reset. Any active call is dropped.

14 Configure the transmit queue limit for the channel.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
txQueueLimit <txQlim>
```

15 Force the value for cell loss priority to equal one for cells in the transmit direction.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm
forceTagging <tag>
```

--End--

Variable definitions

Variable	Value
<atmS>	*is Cbr*, *RtVbr*, *NrtVbr*, *Ubr* or *derivedFromBbc*. The default is *Ubr*. Set the value to *derivedFromBbc* if this channel is to become an SPVC.
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<descriptor_type>	is either *VirtualPathDescriptor* for Vpcs or VirtualChannelDescriptor for Vccs.
<hPri>	is a decimal from 0 to 4. The default is 2. The decimal 0 indicate highest priority and the decimal 4 indicates lowest priority).
<mode>	is d*isabled*, *enabled*, or *perFlow*. The default value is disabled.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<parm1>	is traffic descriptor parameter 1 (peak cell rate) between 0 and 2 147 483 647. The default is 0. The traffic descriptor type and traffic descriptor parameters have default values that for the transmit direction. If the value of the *rxTrafficDescType* attribute is *sameAsTx*, the parameters for this attribute are the same as the parameters in the *txTrafficDescParm* attribute. The encoding value that the node signals for each parameter is three bytes long, which imposes a maximum limit of 16 777 215 for each parameter value. The node truncates larger values.
<parm2>	is traffic descriptor parameter 2 (PCR0 or SCR) between 0 and 2 147 483 647. The default is 0. The traffic descriptor type and traffic descriptor parameters have default values that for the transmit direction. If the value of the *rxTrafficDescType* attribute is *sameAsTx*, the parameters for this attribute are the same as the parameters in the *txTrafficDescParm* attribute. The encoding value that the node signals for each parameter is three bytes long, which imposes a maximum limit of 16 777 215 for each parameter value. The node truncates larger values.
<parm3>	is traffic descriptor parameter 3 (MBS) between 0 and 2 147 483 647. The default is 0. The traffic descriptor type and traffic descriptor parameters have default values that for the transmit direction. If the value of the *rxTrafficDescType* attribute is *sameAsTx*, the parameters for this attribute are the same as the parameters in the *txTrafficDescParm* attribute. The encoding value that the node signals for each parameter is three bytes long, which imposes a maximum limit of 16 777 215 for each parameter value. The node truncates larger values.

(1 of 5)

Variable	Value
<parm4>	is traffic descriptor parameter 4 (CDVT) between 0 and 10 000 to 1 200 000. In provisioning mode, the default value is 0. However, in operational mode the parameter takes on the value of atmIf/x ca ubr/0 cdvt which is 250. If you are going to enable UPC on the Vcc, the traffic descriptor parameters for the receive direction must take UPC into account. The traffic descriptor type and traffic descriptor parameters have default values that for the transmit direction. If the value of the *rxTrafficDescType* attribute is *sameAsTx*, the parameters for this attribute are the same as the parameters in the *txTrafficDescParm* attribute. The encoding value that the node signals for each parameter is three bytes long, which imposes a maximum limit of 16 777 215 for each parameter value. The node truncates larger values.
<parm5>	is traffic descriptor parameter 5 (requested shaping rate) between 0 and 2 147 483 647. This parameter is not applicable to standard VPT VCCs. The default is 0. The traffic descriptor type and traffic descriptor parameters have default values that for the transmit direction. If the value of the *rxTrafficDescType* attribute is *sameAsTx*, the parameters for this attribute are the same as the parameters in the *txTrafficDescParm* attribute. The encoding value that the node signals for each parameter is three bytes long, which imposes a maximum limit of 16 777 215 for each parameter value. The node truncates larger values.
<rxpd>	is d*isabled* or *enabled*. The default is disabled. For soft PVCs, set both the *txPacketWiseDiscard* attribute and the *rxPacketWiseDiscard* attribute to the same value. Partial packet discard (PPD) must be either on or off in both directions for soft PVCs. For GQM-based FPs, rxpd must be disabled.
<rxTdt>	is a value between 1 and 9 and defines the traffic descriptor type. The default is 1. Traffic descriptor 9 is to be used exclusively by UBR with MDCR. When rxTdt is 9 for UBR with a minimum desired cell rate (MDCR), <parm1> is PCR, <parm2> is CDVT, and <parm3> is MDCR

(2 of 5)

Variable	Value
<tag>	is s*ameAsCa*, *enabled*, or *disabled*. The default is *sameAsCa*. This attribute does not apply to interfaces on CQC-based FPs or to Vpts. Use *sameAsCa* to derive the tag setting from the configuration for the *ConnectionAdministrator* component. Use enabled and disabled to reconfigure tagging away from the setting for the *ConnectionAdministrator* component. Use enabled on ATM IP and APC-based FPs to force CLP=1 on all cells in the transmit direction. Use disabled on ATM IP and APC-based FPs to leave the CLP tag unchanged for cells in the transmit direction. Use disabled for interfaces on CQC-based FPs, since forced tagging is not available. The forceTagging feature is only valid for switched connections with a UBR service category and permanent connections of all service categories.
<threshold>	is one of 25, 50, or 75. The default is 25. This value defines a percentage.
<trshaping>	is disabled or *sameAsCa*. The default is *sameAsCa*. Use *sameAsCa* to derive shaping characteristics from the service category settings under the *ConnectionAdministrator* component. If the service category has shaping disabled, the software disables shaping for transmitted traffic on connections under that service category. The shaping characteristics that software derives from the *ConnectionAdministrator* component depends on the hardware configuration. See the table Relationship between rxTrafficDescType and policing action values (page 55) for information on how characteristics apply. On CQC-based FPs and 1-port OC-12 FPs, shaping does not apply when the service category is CBR. Use disabled to disable shaping for transmitted traffic to the ATM interface, regardless of the setting for the service category.
<txpd>	is d*isabled* or *enabled*. The default is disabled. For soft PVCs, set both the *txPacketWiseDiscard* attribute and the *rxPacketWiseDiscard* attribute to the same value. Partial packet discard (PPD) must be either on or off in both directions for soft PVCs.
<txQlim>	is s*ameAsCa* or a decimal from 5 to 512 000. For APC-based FPs, the usable range is 88 to 65 535. The default is *sameAsCa*. For GQM-based FPs, the usable range is 32 to 262 144. Use *sameAsCa* to derive the queue limit from the configuration for the *ConnectionAdministrator* component. For FPs other than CQC-based, use a decimal to configure a specific limit that is different from the limit set for the *ConnectionAdministrator* component.

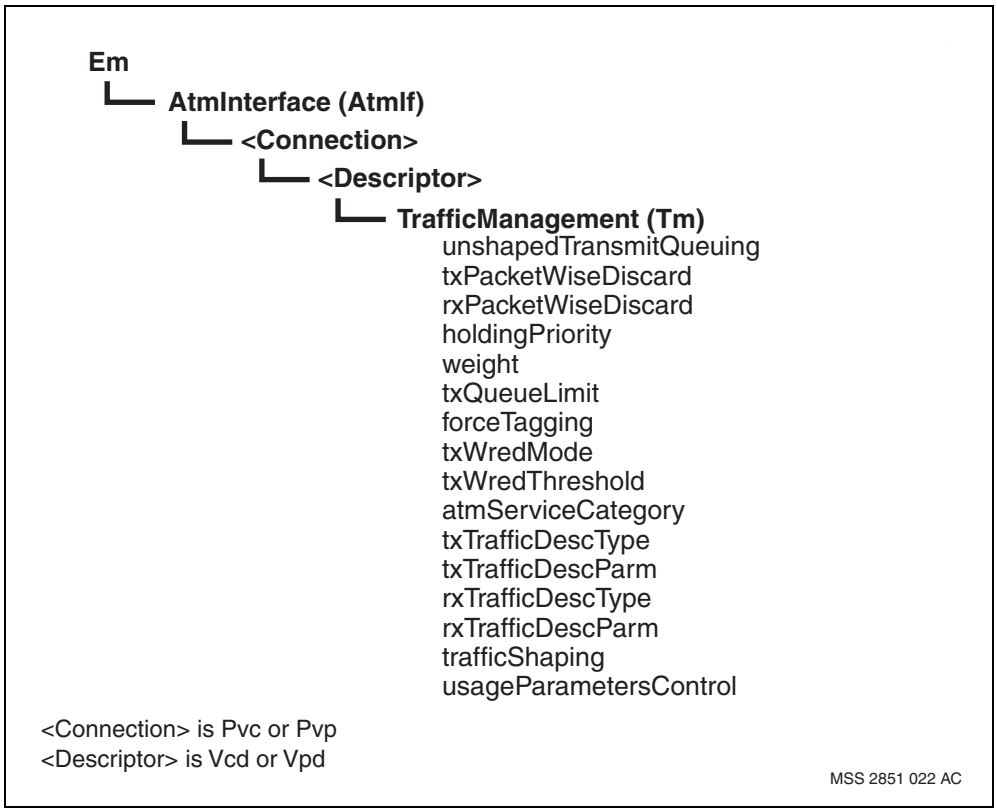
(3 of 5)

Variable	Value
<txTdt>	is a value between 1 and 9 and defines the traffic descriptor type. The default is 1. Traffic descriptor 9 is to be used exclusively by UBR with MDCR. When txTdt is 9 for UBR with a minimum desired cell rate (MDCR), <parm1> is PCR, <parm2> is CDVT, and <parm3> is MDCR
<unshap>	is c*ommon* or *sameAsCa*. The default is value is *sameAsCa*. For APC-based, GQM-based, and ATM IP FPs, the value must be set to *sameAsCa*. Use *sameAsCa* for ATM IP FPs, 1-port OC-12 FPs, CQC-based, and GQM-based FPs to apply the default per-VC queuing permissions that you defined for the *AtmIf* component. Use *common* to direct traffic to the common queue. This attribute applies to unshaped traffic only. Shaped traffic is always serviced on per-VC queues.
<upc>	is e*nforced*, *disabled*, *sameAsCa*, or *monitored*. The default is *sameAsCa*. Set the value to enforced to permit policing for received traffic from the ATM interface regardless of the setting for the service category. Set the value to disabled to turn off policing for received traffic from the ATM interface regardless of the setting for the service category. If UPC for the service category is disabled, the software disables policing for received traffic on connections under that service category. Set the value to *sameAsCa* to derive UPC characteristics from the service category settings under the *ConnectionAdministrator* component. Use monitored to count UPC violations and pass cells unchanged. For CQC-based FPs, a value of monitored disables UPC since the node cannot monitor UPC violations on a CQC-based FP.
(4 of 5)	

Variable	Value
<weight>	is a decimal value from 1 to 4 095, *sameAsCa*, or *upToQueueLimit*. The default is *sameAsCa*. For the APC-based FPs, the value *upToQueueLimit* does not apply. For the GQM-based FPs, the value *upToQueueLimit* must not be used since the hardware expects a finite weight value. Use a decimal value when you need to define relative fairness between a number of connections under the interface. Use *sameAsCa* to derive the fairness weighting from the configuration for the *ConnectionAdministrator* component. For AQM-based processors, use *upToQueueLimit* to limit fairness weighting by the transmit queue limit specified in the *txQueueLimit* attribute. For AQM-based processors, set the *Tm weight* attribute to upToQueueLimit for NRT-VBR and UBR connections. This attribute does not apply to CBR and RT-VBR connections.
<x>	is the instance of the *Vcc*. If the virtual channel is associated with an *AtmIf* component, <x> represents the VPI.VCI value. If the virtual channel is associated with a *Vpt* component, <x> represents the VCI value.
	(5 of 5)

Procedure job aid

Traffic management component hierarchy



Routing protocol configuration

Configure routing protocol to allow switched connections to be created.

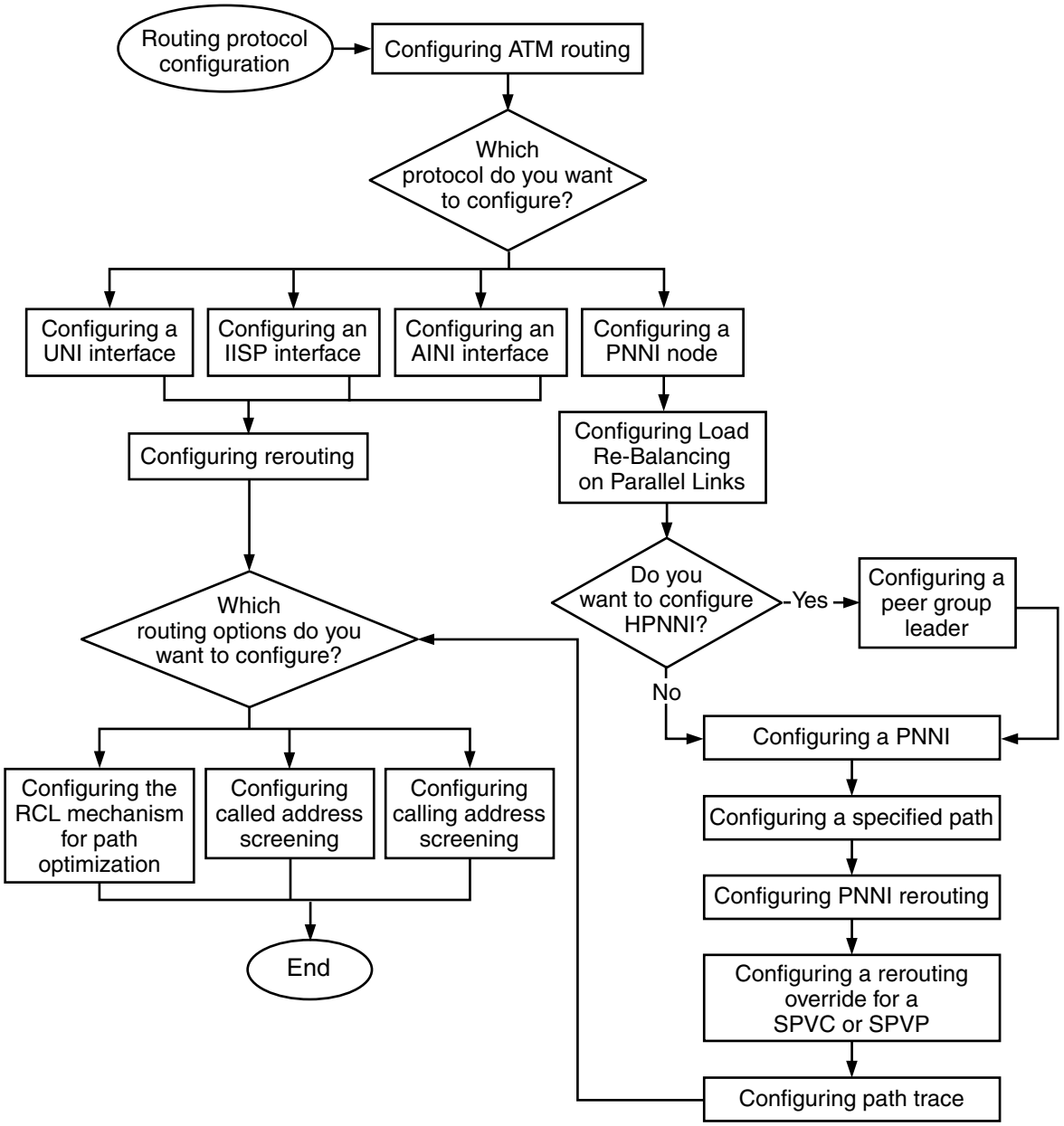
Prerequisites to routing protocol configuration

- To implement ATM networking with UNI, IISP, AINI and PNNI routing and signaling, configure the *AtmRouting* component as a root component. This component provides default addressing information for the host Nortel Multiservice Switch node.
- See NN10600-702 *Nortel Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals* for reference on ATM routing concepts.
- Configure static routing when you configure the signaling interface. Define an address for each signaling interface through the *Address* subcomponent. With the PNNI protocol, the system looks for a static address first. If there is no static address configured, the system routes the call dynamically.

Routing protocol configuration procedures

This taskflow shows you the sequence of procedures you perform to configure routing protocols. To link to any procedure, go to Routing protocol configuration procedure navigation (page 120).

Routing protocol configuration procedures



MSS 3612 007 AB

Routing protocol configuration procedure navigation

Configuring ATM routing

Configure ATM routing of any protocol by adding the necessary software and components.

Prerequisites

WARNING
Critical change
Changing the value of the *nodePrefix* attribute will reboot the Nortel Multiservice Switch node.

- Consult network engineering personnel to determine which LPs support which interfaces. If the atmHpnni feature is being added to the CP feature list, ensure that the atmPnni feature is present on at least one of the FP feature list. If the atmPnni feature is being removed from all the FPs feature lists, ensure that the atmHpnni feature, if present, is also removed from the CP feature list. It is recommended that you configure a software LPT on a per FP basis.

Procedure steps

Step	Action
1	Set the software available to include ATM networking software. **`set Sw avList atmNetworking_<version>`**
2	Specify a node prefix so that this node can be identified in the network. **`set ModuleData nodePrefix <nodePrefix>`**
3	Load the ATM networking feature list software onto one LP on the node. **`set Sw Lpt/ATM featureList <feature>`** Activate configuration changes, see Activating configuration changes (page 238).
4	Add ATM routing. **`add ARtg`**

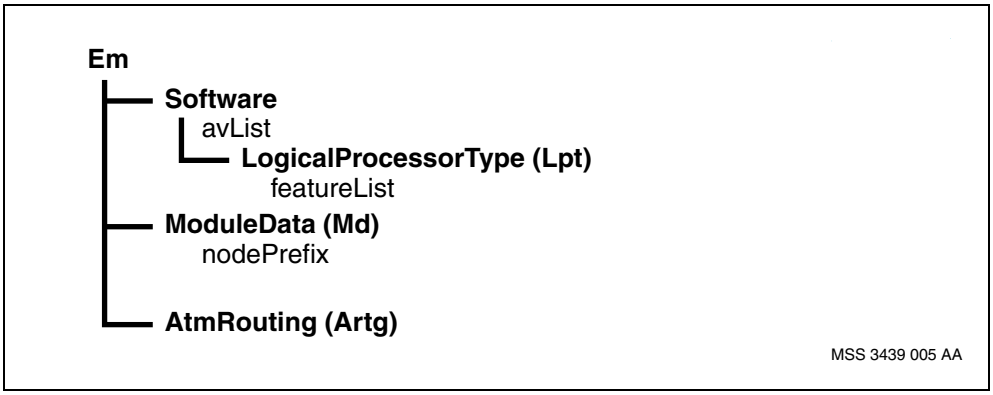
--End--

Variable definitions

Variable	Value
<feature>	is atmUni, atmIisp, atmAini or atmPnni. If an LP supports only PNNIs, load only atmPnni. Do not load atmUni, atmIisp, or atmAini because the LP does not use the software in these features. Loading additional software uses additional memory that would otherwise be used for dynamic connection space. The exception is in a sparing configuration, where you need to share the software LPTs across the FPs.
<nodePrefix>	is a 26-hexadecimal-digit string. The first two digits must be either 39, 45, or 47. If the first two digits are 45, the 18th digit must be an F.
<version>	is the correct version of software for your product.

Procedure job aid

ATM routing component hierarchy



Configuring a UNI interface

Configure a UNI interface to support the UNI protocol.

For APC-based FPs, all *AtmIf* components under each APC must be configured as UNIs.

Procedure steps

Step	Action
1	Define the ATM interface as UNI. **`add AtmIf/<n> Uni`** The system automatically creates the *Signalling* and *Ilmi* subcomponents when you configure the *Uni* component.
2	Set the attributes for the *Uni* component. **`set AtmIf/<n> Uni version <forumVer>, side <ifSide>`**
3	Optionally, configure the UNI addresses. **`add AtmIf/<n> Uni Address/<address>, <addr_type>`**
4	Optionally, configure the UNI address scope for static group addresses. **`add AtmIf/<n> Uni Address/<address>, <addr_type> UniInfo scope <scope> pnniReachability <pnnireach>`**

--End--

Variable definitions

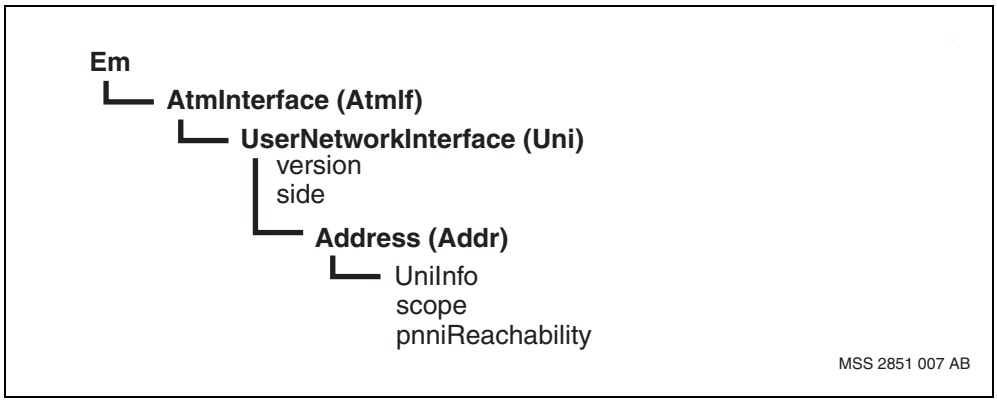
Variable	Value
<address>	is a static address associated with the UNI, consisting of either up to 40 hexadecimal digits or a single wild card character (the ? symbol). You can configure multiple static address for each UNI, but all address entries must be unique on this UNI.
<addr_type>	is primary or alternate. The address is a string entry beginning with 37, 39, 45, or 47. If the address begins with 45, the 18th digit must be the hexadecimal digit F. For complete details on the addressing requirements, see NN10600-702 *Nortel Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals* and *ATM User-to-Network Interface Specification Version 3.1* (ATM Forum Technical Committee).
<forumVer>	is atmForum30, atmForum31, or atmForum40. The default is atmForum40. This attribute defines the version of the ATM Forum specification to which the UNI complies.
<ifSide>	is network or user. The default is network. This attribute defines the side of the UNI interface. The side of the interface must be balanced by the value in the side attribute for the interface at the other end. For example, if this interface is defined as network side, the opposite interface (on the other side of the link) must be defined as user side. If the node is connected to a workstation with an network interface card (NIC), the interface *side* attribute must be configured as network.
<n>	is the number of the ATM interface associated with the UNI.

(1 of 2)

Variable	Value
<pnnireach>	is internal or exterior. The default is internal. This attribute indicates if the address is internal or exterior. The destination of an internal reachable address is directly attached to the logical node that advertises the address. The destination of an exterior reachable address is reachable through, but not located in, a PNNI routing domain. *UniInfo* is not allowed if the PnniInfo component is configured.
<scope>	is one of the following: • useDefaultScope • localNetwork • localPlusOne • localPlusTwo • siteMinusOne • intraSite • intraSitePlusOne • organizationMinusOne • intraOrganization • organizationPlusOne • communityMinusOne • intraCommunity • communityPlusOne • regional • interRegional • global The default is useDefaultScope. The *scope* attribute defines the inclusive routing hierarchy in which the address is known.
	(2 of 2)

Procedure job aid

UNI interface component hierarchy



Configuring an IISP interface

Configure an inter-switch protocol (IISP) to support the IISP protocol.

Procedure steps

Step	Action
1	Define the ATM interface as IISP. **`add AtmIf/<n> Iisp`** The system automatically creates the *Signalling* subcomponent when you configure the *Iisp* component.
2	Set the attributes for the *Iisp* component. **`set AtmIf/<n> Iisp version <forumVer>, side <ifSide>`**
3	Set the IISP interface addresses. **`add AtmIf/<n> Iisp address/<address>,<type>`**

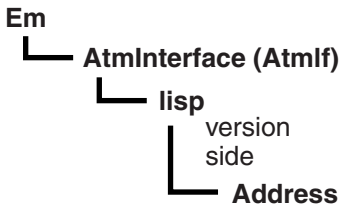
--End--

Variable definitions

Variable	Value
<address>	is a static address associated with the IISP interface, consisting of either, up to 40 hexadecimal digits, or a single wild card character (the ? symbol). You can configure multiple static addresses for each IISP interface, but all address entries must be unique.
<forumVer>	is iisp10Sig30 or iisp10Sig31. The default is iisp10Sig31. This attribute defines the version of the ATM Forum specification to which the IISP interface complies.
<ifSide>	is network or user. The default is network. This attribute defines the side of the interface on which the node resides. The side of the interface must be balanced by the value in the side attribute for the interface at the other end. For example, if this interface is defined as network side, the opposite interface (on the other side of the link) must be defined as user side.
<n>	is the number of the ATM interface.
<type>	is primary or alternate. ATM device-specific addresses are accessible through the ATM interface.

Procedure job aid

IISP interface component hierarchy



MSS 2851 008 AA

Configuring an AINI interface

Configure an ATM inter-network interface (AINI) to support the AINI protocol.

Procedure steps

Step	Action
1	Define the ATM interface as AINI. **`add AtmIf/<n> Aini`** The system automatically creates the *Signalling* subcomponent when you configure the *Aini* component.
2	Set the attributes for the *Aini* component. **`set AtmIf/<n> Aini side <side>`**
3	Set the Aini interface addresses. **`add AtmIf/<n> Aini address/<address>,<type>`** ATM device-specific addresses are accessible through the ATM interface.

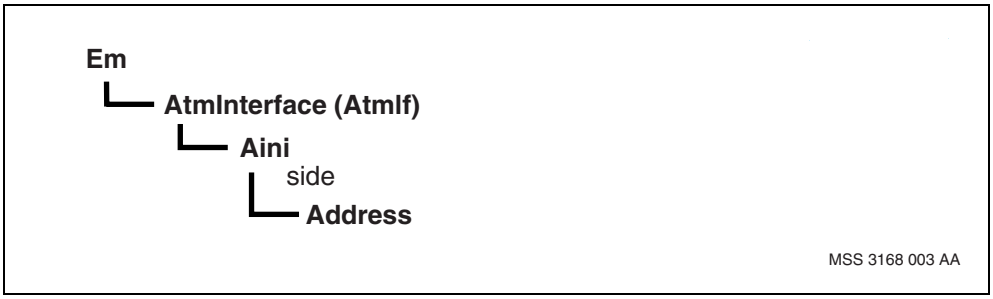
--End--

Variable definitions

Variable	Value
<address>	is a static address associated with the AINI interface, consisting of either, up to 40 hexadecimal digits, or a single wild card character (the ? symbol). You can configure multiple static addresses for each AINI interface, but all address entries must be unique.
<n>	is the number of the ATM interface.
<side>	is networkSide or userSide. The default is networkSide. This attribute defines the side of the interface on which the node resides. The side of the interface must be balanced by the value in the side attribute for the interface at the other end. For example, if this interface is defined as network side, the opposite interface (on the other side of the link) must be defined as user side.
<type>	is primary or alternate.

Procedure job aid

ATM inter-network interface component hierarchy



Configuring a PNNI node

Configure a PNNI node by defining a level of participation for the node in the PNNI networking hierarchy. Ask network engineering personnel for information about determining level assignment.

In a hierarchical PNNI network, Nortel Multiservice Switch systems define level 0 as the top-most level and level 104 as the bottom-most level.

Procedure steps

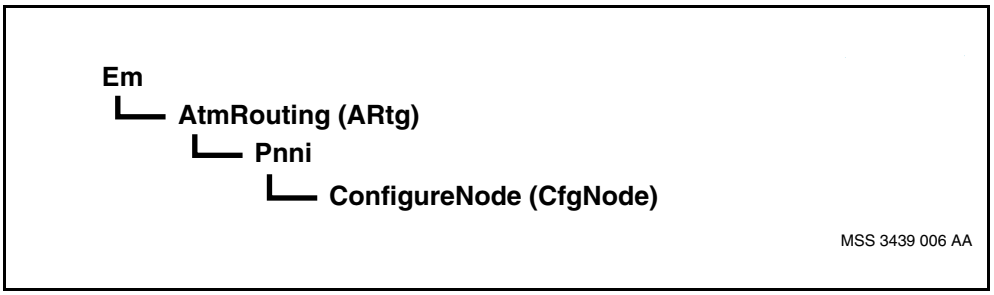
Step	Action
1	Define the node as a PNNI node in the PNNI hierarchy. **`add ARtg Pnni`**
2	Define the node for a single level of participation in the PNNI hierarchy. **`add ARtg Pnni CfgNode/<CfgNode_level>`**
	--End--

Variable definitions

Variable	Value
<CfgNode_level>	is the level of the node in the PNNI networking hierarchy having a value between 0 and 104, inclusive.

Procedure job aid

PNNI dynamic routing component hierarchy



Configuring Load Re-Balancing on Parallel Links

Configure load re-balancing on a PNNI node that requires re-balancing.

Attention: Although the software on a neighbor node must also support load re-balancing, no provisioning is required on the neighbor node.

Procedure steps

Step	Action
1	Add Load Re-Balancing on Parallel Links (LRB) to the *Software LogicalProcessType* feature list. **`set Sw lpt/ATM featureList atmLrb`**
2	Add the *LRBG* component. **`add artg pnni lrbg/1`**
3	Set the *automaticRebalancing* attribute to enable, so that re-balancing is triggered automatically by link up. **`set artg pnni lrbg/1 automaticRebalancing enable`**
4	Optionally, set the *automaticRebalancing* attribute to disable, so that re-balancing is triggered by manual commands. **`set artg pnni lrbg/1 automaticRebalancing disable`**
5	Add the *Link* component. **`add artg pnni lrbg/1 link/<l>`**
6	Set the *interfaceName* attribute which is associated with one of *AtmIf PNNI*. **`set artg pnni lrbg/1 link/<l> interfaceName atmif/<n> pnni`**
7	Repeat step 5 and step 6 for adding more parallel links.

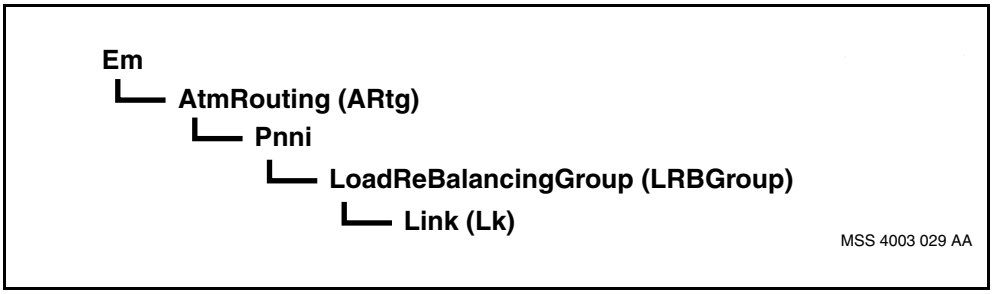
--End--

Variable definitions

Variable	Value
<l>	is the PNNI parallel link number.
<n>	is the *AtmIf PNNI* number.

Procedure job aid

Load Re-Balancing on Parallel Links component hierarchy



Configuring rerouting

Configure rerouting to define the path optimization capability by overriding nodal default rerouting attribute values for the ATM interface level.

Connections that exist before configuring connection subscription for rerouting retain the options determined at call establishment. New call set up requests initiated after configuring connection subscription for rerouting will use the new connection subscription options.

Procedure steps

Step	Action
1	Add the *Reroute* component to the ATM interface. **add AtmIf/<n> <IfType> Reroute**
2	Set the default rerouting protocol used by the interface. **set AtmIf/<n> <IfType> Reroute Override protocol <protocol>**
3	Set the number of path optimizations attempted before declaring a connection optimally routed. **set AtmIf/<n> <IfType> Reroute Override numOptAttempts <numOptAttempts>**
4	Set the connections that will be considered by a module optimization pass. **set AtmIf/<n> <IfType> Reroute Override moduleOptimizationConnections <moduleOptimization>**
5	Set the mechanism used for switchover for the rerouted connection segment. **set AtmIf/<n> <IfType> Reroute Override switchoverMechanism <switchoverMechanism>**
6	Set which connection is requesting subscription to local connection recovery. **set AtmIf/<n> <IfType> Reroute Override localConnectionRecoverySubscr <lConRec>**
7	Set which connection is requesting subscription to local path optimization. **set AtmIf/<n> <IfType> Reroute Override localPathOptimizationSubscr <lPathOpt>**
8	Set which connection is requesting subscription to connection recovery. **set AtmIf/<n> <IfType> Reroute Override globalConnectionRecoverySubscr <gConRec>**
9	Set which connection is requesting subscription to path optimization.

```
set AtmIf/<n> <IfType> Reroute Override
globalPathOptimizationSubscr <gPathOpt>
```

--End--

Variable definitions

Variable	Value
<gConRec>	specifies which connection is requesting subscription to connection recovery. This attribute only applies to connections which request rerouting protocol. The possible values are sameAsARtgPnniReroute (sameAs), sourceSpvcSpvp (source), and transitSpvcSpvp.
<gPathOpt>	specifies which connection is requesting subscription to path optimization.This attribute only applies to connections which request rerouting protocol. The possible values are sameAsARtgPnniReroute (sameAs), sourceSpvcSpvp (source), and transitSpvcSpvp.
<lConRec>	specifies which connection is requesting subscription to local connection recovery. This attribute applies to connection which request local rerouting protocol. The possible values are sameAsARtgPnniReroute (sameAs), svcSvp, and transitSpvcSpvp.
<lfType>	is Uni, Iisp, Aini, or Pnni. The only case in which Reroute is configured on a PNNI interface is when a network-initiated connection (soft PVC or soft PVP) is configured on that interface. The <lfType> component could also be an interface component under a Vpt component.
<lPathOpt>	specifies which connection is requesting subscription to path optimization.This attribute applies to connection which request local rerouting protocols. The possible values are sameAsARtgPnniReroute (sameAs), sourceSpvcSpvp (source), and transitSpvcSpvp.
<moduleOptimization>	specifies the connections that will be considered by a module optimization pass as controlled by Artg Pnni Reroute.This attribute applies to all the rerouting protocols. The possible values are none, recoveredOnly, allSubscribed, and sameAsARtgPnniReroute (sameAs).
<n>	is the instance number of the ATM interface.
<numOptAttempts>	is the number of path optimizations attempted before declaring a connection optimally routed. The possible values is a number from 1 to 15 and sameARtgPnniReroute (sameAs).

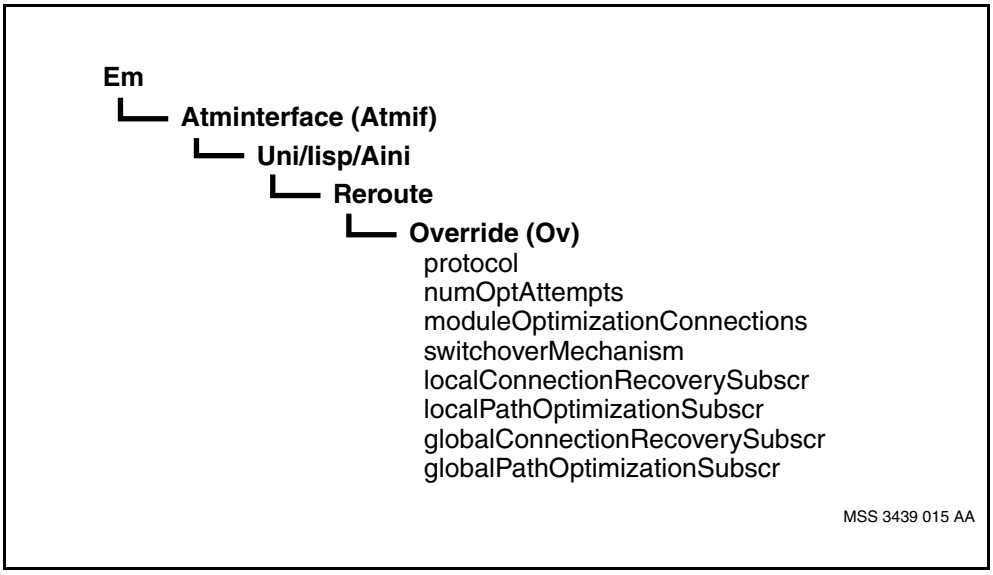
(1 of 2)

Variable	Value
<protocol>	is the default rerouting protocol used by the interface. The possible values are: localGlobal, localOnly, globalOnly, localEbr, ebrOnly, and sameAsARtgPnniReroute (sameAs).
<switchoverMechanism>	specifies the mechanism used for switchover for the rerouted connection segment. The possible values are standard, rcl, and sameAsARtgPnniReroute(sameAs).

(2 of 2)

Procedure job aid

Rerouting component hierarchy



Configuring a peer group leader

Configure a peer group leader (PGL) to allow a node to become eligible for election as the PGL using a top-down approach. If the node's election priority is high enough, the node will be elected as the PGL.

Procedure steps

Step	Action
1	Create another level in the PNNI networking hierarchy. **`add ARtg Pnni CfgNode/<CfgNode_level>`** The new *CfgNode* must have a value that is numerically greater than that configured in Configuring a PNNI node (page 131).
2	Add the PNNI *Pgl* component to the lowest *ConfiguredNode* level. **`add ARtg Pnni CfgNode/<CfgNode_level> Pgl`**
3	Check to see that the next higher PNNI level exists. **`list ARtg Pnni CfgNode/*`**
4	Set the priority for the PGL added in step 2 to a non-zero value. **`set ARtg Pnni CfgNode/<CfgNode_level> Pgl lPrio <lPrio>`**
5	Repeat this procedure as necessary to create new PGLs.

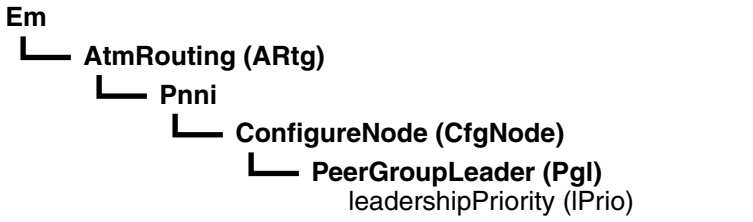
--End--

Variable definitions

Variable	Value
<CfgNode_level>	is the level of the node in the PNNI networking hierarchy having a value that is numerically greater than the node configured in Configuring a PNNI node (page 131). In a hierarchical PNNI network, the Nortel Multiservice Switch system defines level 0 as the top-most level and level 104 as the bottom-most level.
<lPrio>	is a value specifying the leadership priority in the range 1 through 205 inclusive. The default value is 1. If you are using this command to change an existing PGL leadership priority (*lPrio*), take into consideration the fact that the leadership priority value is increased by 50 to ensure stability per the ATM Forum PNNI Specification Version 1.0. If the leadership priority of an existing PGL is modified, and if another node having a higher priority exists, then a PGL re-election is held immediately after the activation of the configured data. If two nodes with the same leadership priority value is in contention for the role of the PGL, the node with the highest node ID is elected.

Procedure job aid

Peer group leader using a top-down approach component hierarchy



Configuring a PNNI

Configure a private network-to-network interface (PNNI) to support the PNNI protocol.

Procedure steps

Step	Action
1	Define the ATM interface as PNNI. **`add AtmIf/<n> Pnni`** The system automatically creates the *Signalling* and *Rcc* subcomponents when you configure the *Pnni* component.
2	Define the administrative weights for each service category in the *AdminWeights* attribute group of the *Pnni* component. **`set AtmIf/<n> Pnni <service_category> <category_value>`**

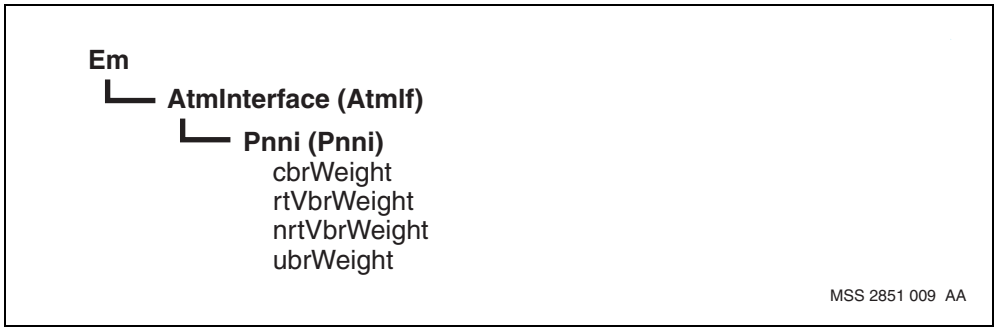
--End--

Variable definitions

Variable	Value
<category_value>	are decimal entries between 0 and 4 294 967 295 for each of the service categories. The default for each attribute is 5040. Multiservice Switch systems use these values to determine the best path if administrative weight is the selected optimization criterion. The higher the value, the less desirable the link is in path selection.
<n>	is the number of the ATM interface that you want to associate with the PNNI.
<service_category>	is one of cbrWeight, rtVbrWeight, nrtVbrWeight, and ubrWeight service categories for the PNNI interface.

Procedure job aid

PNNI component hierarchy



Configuring a specified path

Configure a specific path to override the PNNI selected route.

Procedure steps

Step	Action
1	Add the *nodeName* component. **add ARtg Pnni nodeName/<nodeName>**
2	Assign a node ID to the node name using the lowest level physical node ID. **set ARtg Pnni nodeName/<nodename> nodeId <nodeId>**
3	Add the *MdtlPath* component. **add ARtg Pnni MdtlPath/<MdtlPath>**
4	Add the *Hop* component. **add ARtg Pnni MdtlPath/<MdtlPath> Hop/<Hop>**
5	Link a *nodeName* component to the hop. **set ARtg Pnni MdtlPath/<MdtlPath> Hop/<Hop> nodeName <nodename>**
6	Optionally, specify the port ID of the exit link using the node name translation table. **set ARtg Pnni MdtlPath/<MdtlPath> Hop/<Hop> portId <portId>**

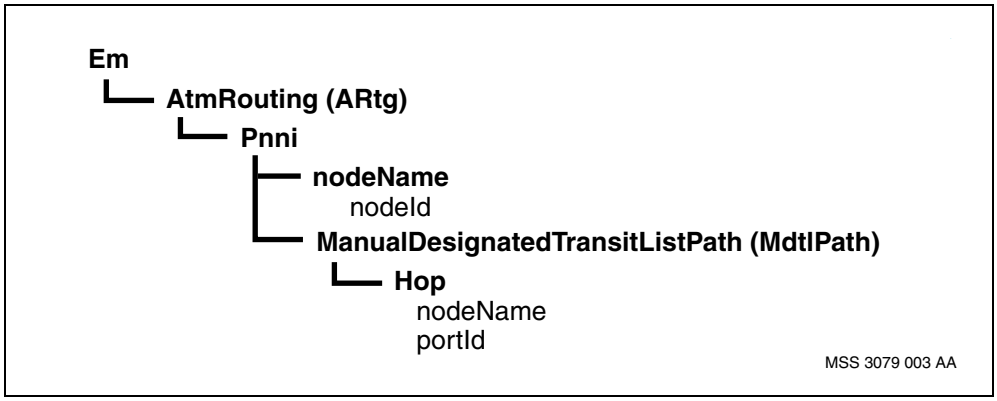
--End--

Variable definitions

Variable	Value
<Hop>	is the instance value of the *Hop* component. The instance value can range from 1 to 255. From 2 to 40 hops can be specified in a specific path. It is recommended that you reserve an instance of 1 for the source node and the highest number for the destination node. The specified path with the lowest instance number is the first hop.
<MdtlPath>	is the name of the *MdtlPath* component. The maximum length for a path name is 32 ASCII characters. The maximum number of paths that can be configured on a source node is 512 paths per shelf.
<nodeId>	is a hexadecimal entry of 22 hexadecimal digits. A logical node ID can be present in the node name translation table. However, when you build a specified path, the path must only contain hops with physical node IDs. Each node must have a unique node ID name. The node ID is derived under the Artg Pnni cfg Node/x opNodeId component.
<nodename>	is the name of the node, which comes from the nodeName in the node translation table.
<portId>	is the value of the port ID. The default value is set to zero when adding a hop. To determine the portId on the given node, issue the following command: Display artg port/*.

Procedure job aid

Specified path component hierarchy



Configuring PNNI rerouting

Configure PNNI rerouting to enable connection recovery and path optimization.

Procedure steps

Step	Action
1	Create a suitable software LPT with the rerouting feature. **set Sw Lpt/<Lpt> featureList atmReroute**
2	Link the LPT to an LP. **set Lp/<n> Lpt Sw Lpt/<Lpt>** <Lpt> is the name you selected in step 1
3	Add the Reroute component under ARtg PNNI. **add ARtg Pnni reroute**
4	Set the default rerouting protocol. **set ARtg Pnni reroute protocol localGlobal**
5	Add the *Reroute* component to the ATM interface. **add AtmIf/<m> Pnni Reroute**
6	Set the method used to identify the type of local domain link. **set AtmIf/<m> Pnni Reroute Ov localDomainLink <domain>**
7	Set the default rerouting protocol used by the interface. **set AtmIf/<m> Pnni Reroute Ov protocol <protocol>**
8	Set the number of path optimizations attempted before declaring a connection optimally routed. **set AtmIf/<m> Pnni Reroute Ov numOptAttempts <numOptAttempts>**
9	Set the connections that will be considered by a module optimization pass as controlled by Artg Pnni Reroute. **set AtmIf/<m> Pnni Reroute Ov moduleOptimizationConnections <moduleOptimization>**
10	Set the mechanism used for switchover for the rerouted connection segment. **set AtmIf/<m> Pnni Reroute Ov switchoverMechanism <switchoverMechanism>**
11	Set which connection is requesting subscription to local connection recovery.

```
set AtmIf/<m> Pnni Reroute Ov
localConnectionRecoverySubscr <lConRec>
```

12 Set which connection is requesting subscription to local path optimization.

```
set AtmIf/<m> Pnni Reroute Ov
localPathOptimizationSubscr <lPathOpt>
```

13 Set which connection is requesting subscription to connection recovery.

```
set AtmIf/<m> Pnni Reroute Ov
globalConnectionRecoverySubscr <gConRec>
```

14 Set which connection is requesting subscription to optimization.

```
set AtmIf/<m> Pnni Reroute Ov
globalPathOptimizationSubscr <gPathOpt>
```

15 Optionally, set the time or times at which a module optimization pass will occur in a 24 hour period. A maximum of 12 module optimization passes can be configured for each 24 hour period. The list of times is sorted internally and does not need to be configured in chronological order.

```
set ARtg Pnni Reroute optimizationTime ! <hour:min>
```

--End--

Variable definitions

Variable	Value
<domain>	specifies the local PNNI domain link type. The possible values are autodetect, intraDomainLink and interDomainLink.
<gConRec>	specifies which connection is requesting subscription to connection recovery. This attribute only applies to connections which request rerouting protocol. The possible values are sameAsARtgPnniReroute (sameAs), sourceSpvcSpvp (source), and transitSpvcSpvp.
<gPathOpt>	specifies which connection is requesting subscription to path optimization.This attribute only applies to connection which request rerouting protocol. The possible values are sameAsARtgPnniReroute (sameAs), sourceSpvcSpvp (source), and transitSpvcSpvp.
<hour:min>	is the time expressed in terms of the 24 hour clock.
<lConRec>	specifies which connection is requesting subscription to local connection recovery. This attribute applies to connection which request local rerouting protocol. The possible values are sameAsARtgPnniReroute (sameAs), svcSvp, and transitSpvcSpvp.
<lPathOpt>	specifies which connection is requesting subscription to path optimization.This attribute applies to connection which request local rerouting protocols. The possible values sameAsARtgPnniReroute (sameAs), sourceSpvcSpvp (source), and transitSpvcSpvp.
<Lpt>	is any name (for example, Reroute).
<m>	is the instance number of the ATM interface.
<moduleOptimization>	specifies the connections that will be considered by a module optimization pass as controlled by Artg Pnni Reroute. This attribute applies to all the rerouting protocols. The possible values are none, recoveredOnly, allSubscribed, and sameAsARtgPnniReroute (sameAs).
<n>	is the number of the LP being configured.
<numOptAttempts>	is the number of path optimizations attempted before declaring a connection optimally routed. The possible values are a number from 1 to 15 and sameARtgPnniReroute (sameAs).

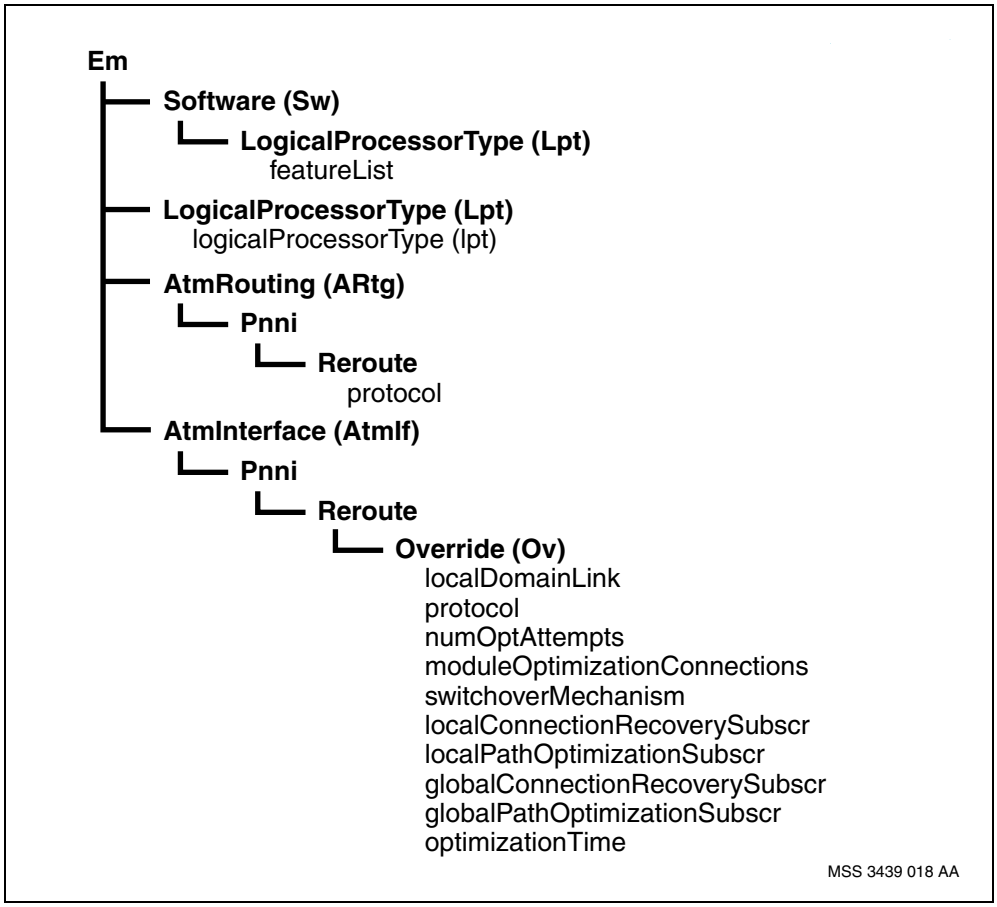
(1 of 2)

Variable	Value
<protocol>	is the default rerouting protocol used by the interface. The possible values are localGlobal, localOnly, globalOnly, localEbr, ebrOnly, and sameAsARtgPnniReroute (sameAs).
<switchoverMechanism>	specifies the mechanism used for switchover for the rerouted connection segment. The possible values are standard, rcl, and sameAsARtgPnniReroute(sameAs).

(2 of 2)

Procedure job aid

PNNI rerouting component hierarchy



Configuring a rerouting override for a SPVC or SPVP

Configure a rerouting override for a SPVC or SPVP to subscribe connections to recovery and/or optimization techniques.

Procedure steps

Step	Action
1	Add the *RerouteOv* component to the VCC. **add AtmIf/<n> <connection_type>/<x> Src RerouteOv**
2	Set which connection is requesting subscription to local connection recovery. **set AtmIf/<n> <connection_type>/<x> Src RerouteOv localConnectionRecoverySubscr yes|no**
3	Set which connection is requesting subscription to local path optimization. **set AtmIf/<n> <connection_type>/<x> Src RerouteOv localPathOptimizationSubscr yes|no**
4	Set which connection is requesting subscription to connection recovery. **set AtmIf/<n> <connection_type>/<x> Src RerouteOv globalConnectionRecoverySubscr yes|no**
5	Set which connection is requesting subscription to path optimization. **set AtmIf/<n> <connection_type>/<x> Src RerouteOv globalPathOptimizationSubscr yes|no**

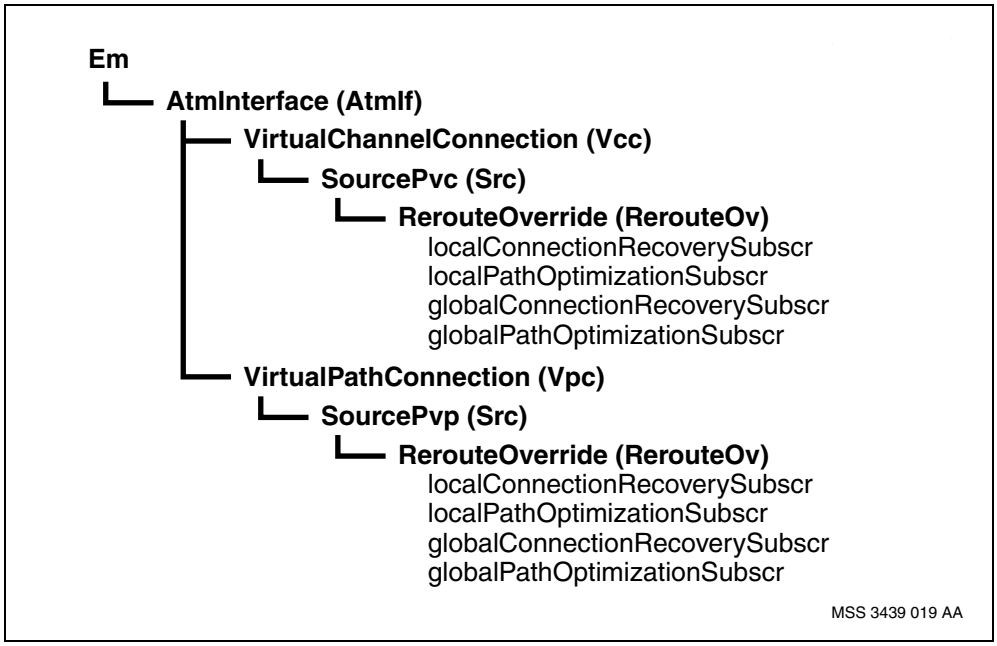
--End--

Variable definitions

Variable	Value
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<n>	is the number of the ATM interface.
<no>	specifies that this soft PVC does not subscribe to connection recovery or path optimization.
<yes>	specifies that this soft PVC subscribes to connection recovery or path optimization (This is the default value.)
<x>	is the instance value of the *Vcc* or Vpc component.

Procedure job aid

Rerouting override for a SPVC or SPVP component hierarchy



Configuring path trace

Configure path trace to set the attributes to run a path trace test, if required. The path trace test connection is operational. Consequently, the user does not need to be in provisioning mode when setting up its attributes.

Prerequisites

WARNING

This procedure resets the FP

Configure the *AtmPathTrace* component before you activate any ATM service to prevent service interruption.

- By default, once the atmUni, atmIisp, atmAini and/or atmPnni features are included in the feature list, the *AtmPathTrace* component dynamically appears under the *Lp Eng* component. The *maxTraceRecords* attribute which specifies the maximum number of records allowed to be stored on this LP has a default value of zero. This means that no filter is allowed to be added below the signaling component, unless *maxTraceRecords* is set to a value between 0 and 200. The *maxConcurrentTraceRequests* attribute which specifies the maximum number of concurrent trace requests on this LP has a default value of 20. Once the maximum number of concurrent trace requests has been reached, the trace transit list information element in the connection setup request is dropped and the setup request proceeds.

Procedure steps

Step	Action
1	Add an *AtmPathTraceOverride* component. **`add Lp/<lp> Eng AtmPathTrace Ov`**
2	Set the maximum trace record attribute or maximum concurrent trace requests. **`set Lp/<lp> Eng AtmPathTrace Ov <atmPathTraceOverrideAttributes>`**
3	Add the *PathTraceFilter* component under the ATM signaling interface component. **`add AtmIf/<n> <IfType> PathTraceFilter`**
4	Set the flags on the path trace filter to specify the desired filter. Otherwise, all flags will take the default values. The default value for the flags is True. **`set AtmIf/<n> <IfType> PathTraceFilter traceInfo crankback connId callRef`**

5 Set the criteria for the *PathTraceFilter* component if necessary. Otherwise, all calls will be traced.

```
set atmif/<n> <IfType> PathTraceFilter <ptf_criteria>
```

6 Set the control for the *PathTraceFilter* component.

```
set AtmIf/<n> <IfType> PathTraceFilter traceTimeout
<minutes>
```

```
set AtmIf/<n> <IfType> PathTraceFilter maxTraceRecords
<m>
```

7 Set the called address for the path trace test connection.

```
set AtmIf/<n> <IfType> PathTraceTestConnection
testCalledAddress <address>
```

8 Set the called vpi.vci for the path trace test connection.

```
set AtmIf/<n> <IfType> PathTraceTestConnection
testCalledVpiVci <calledVpiVci>
```

9 Set the specified path for the path trace test connection.

```
set AtmIf/<n> <IfType> PathTraceTestConnection
testSpecifiedPath <specifiedPath>
```

10 Set the other attributes if desired for the path trace test connection.

```
set AtmIf/<n> <IfType> PathTraceTestConnection
<pathTraceTestConnectionAttributes>
```

11 Set the crankback flag for the path trace connection.

```
set Atmif/<n> <IfType> PathTraceTestConnection
traceCrankback <crankback>
```

12 Set the ATM signaling interface to be a trace destination interface.

```
set AtmIf/<n> <IfType> TraceDestination <yes|no>
```

--End--

Variable definitions

Variable	Value
<address>	is the destination address.
<atmPathTraceOverrideAttributes>	is maxTraceRecords or maxConcurrentTraceRequests.
<calledVpiVci>	is the vpi/vci.
<crankback>	is either yes or no. The default is set to no.
<IfType>	is uni, Iisp, Aini, or Pnni.

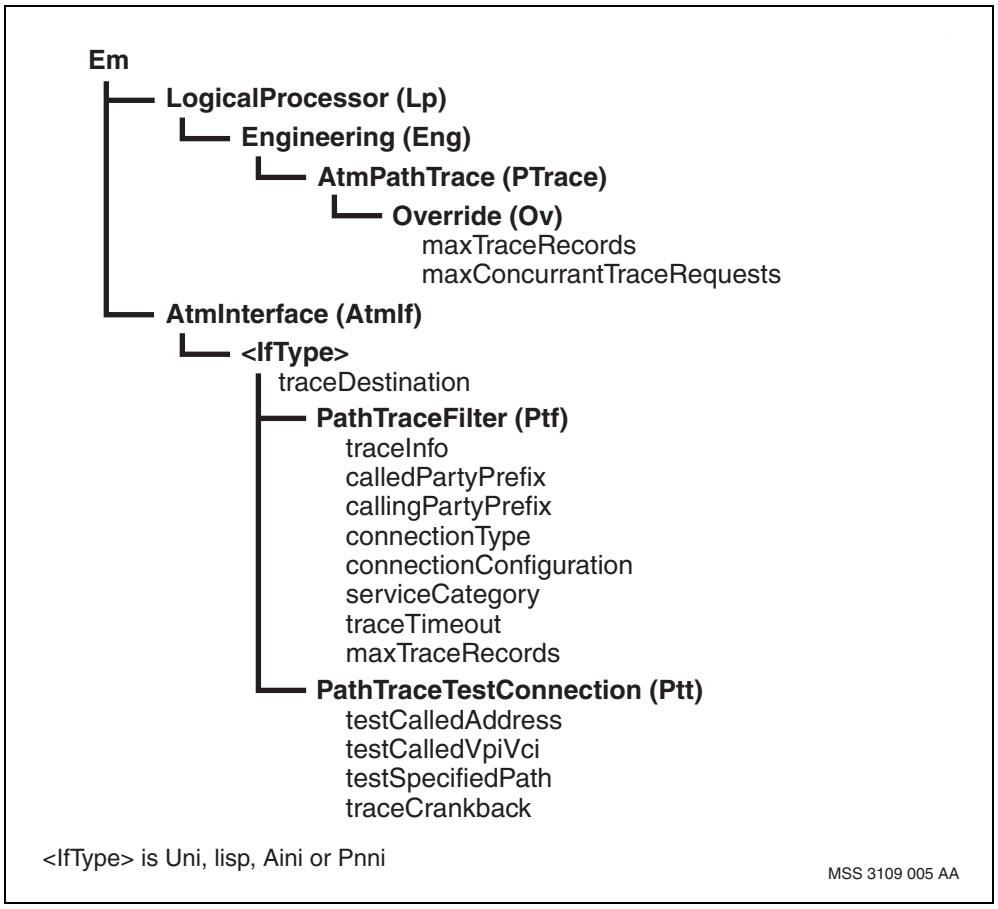
(1 of 2)

Variable	Value
<lp>	is the number of the LP associated with the ATM FP.
<m>	is a number of between 1 and 200. The default value is 2.
<minutes>	is a number between 1 and 70560. The default value is 60.
<n>	is the number of the ATM interface.
<pathTraceTestConnectionAttributes>	is txTrafficDescType, txTrafficDescParm, rxTrafficDescType, rxTrafficDescParm, atmServiceCategory, fwdQosClass, fwdQosParameters, bwdQosClass, bwdQosParameters, bearerClassBbc, transferCapabilityBbc, clippingBbc, or bestEffort.
<ptf_criteria>	is calledPartyPrefix, callingPartyPrefix, connectionType, connectionConfiguration, or serviceCategory.
<specifiedPath>	is an ASCII string containing up to 32 characters. There is no default value. This attribute specifies the path that will be used to get to the destination of this test connection.
<yes|no>	is yes or no. The default value is no.

(2 of 2)

Procedure job aid

Path trace component hierarchy



Configuring the RCL mechanism for path optimization

Configure the reduced cell loss (RCL) mechanism for path optimization to minimize disruption by reducing the amount of cell loss that occurs during a path optimization.

Prerequisites

- Only the Ebr and Global rerouting protocols support the RCL mechanism.

Procedure steps

Step	Action
1	Set the reduced cell loss mechanism. **`set AtmIf/<n> <IfType> reroute ov switchoverMechanism rcl`**
2	If required, revert from the enhanced rerouting capability (where the RCL mechanism is activated) to the standards-based rerouting capability. **`set AtmIf/<n> <IfType> reroute ov switchoverMechanism standard`**

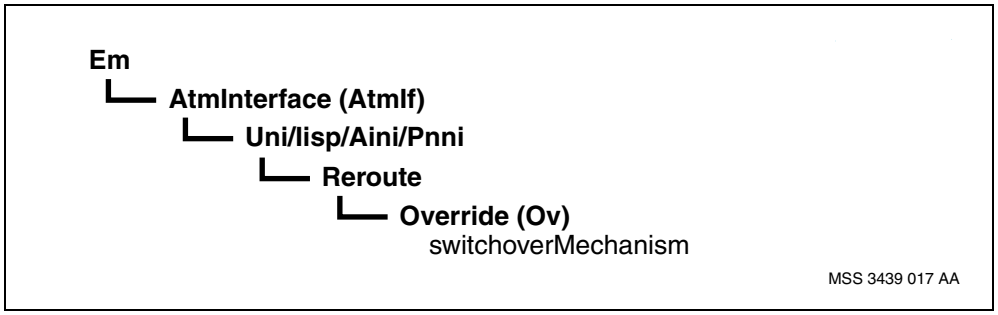
--End--

Variable definitions

Variable	Value
<IfType>	is Uni, Iisp, Aini, or Pnni.
<n>	is the instance number of the ATM interface.

Procedure job aid

RCL mechanism for path optimization component hierarchy



Configuring reachability for PNNI

Configure reachability for PNNI to define the interface addressing requirements for networking under PNNI.

Procedure steps

Step	Action
1	Add a *PnniInfo* component. The *PnniInfo* component contains configured information associated with the ATM address related to PNNI. You add this component only under configured *Address* components. **add AtmIf/<n> <IfType> Address/<address>,<addr_type> PnniInfo**
2	Define the scope and reachability of UNI static addresses for PNNI networking. **set AtmIf/<n> <IfType> Address/<address>,<addr_type> PnniInfo scope <scope> reachability <reach>**

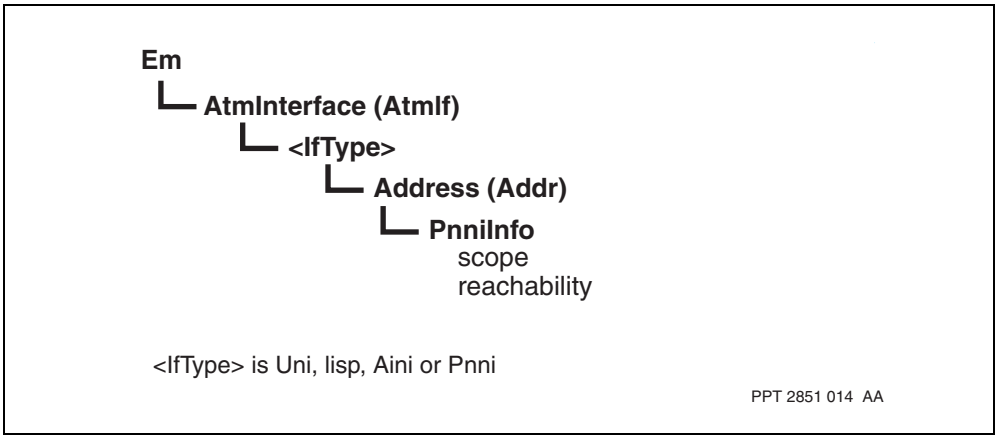
--End--

Variable definitions

Variable	Value
<address>	is a static address associated with the type of interface. The address consists of either up to 40 hexadecimal digits or a single wild card character (the ? symbol). You can configure multiple static addresses for each interface type, but all address entries must be unique.
<addr_type>	is primary or alternate. The default is primary.
<IfType>	is Uni, Iisp, Aini, or Pnni.
<n>	is the number of the ATM interface.
<reach>	is internal or exterior. The default is internal.
<scope>	is a decimal value between 0 and 104. The default is 0. The *scope* attribute defines the scope of the ATM address in a PNNI network. This attribute defines the highest level up to which the node advertises this address in the PNNI hierarchy. Set this attribute to a value that is less than or equal to the level of the lowest level peer group containing this node. A value of 0 (zero) indicates that the node advertises the address globally within the PNNI routing domain.

Procedure job aid

PNNI addressing requirements component hierarchy



Configuring called address screening

Configure called address screening for a UNI, IISP or AINI interface to screen outgoing calls against configured addresses.

By default, called address screening is disabled, which means that calls are not screened. When you enable called address screening, outgoing calls are screened against the addresses that you have configured. The outgoing call is accepted if the called address matches an address that you have configured to be accepted. The outgoing call is rejected if the called address is either not configured, matches an address you have configured to be rejected, or does not match any configured address.

Procedure steps

Step	Action
1	Configure the interface for called address screening. **`add AtmIf/<n> <IfType> CalledAScr`**
2	Add an ATM address to the screening rules. **`add AtmIf/<n> <IfType> CalledAScr Address/ <address>,<action>`**
3	Enable called address screening for this ATM interface. **`set AtmIf/<n> <IfType> CalledAScr adminStatus enabled`**

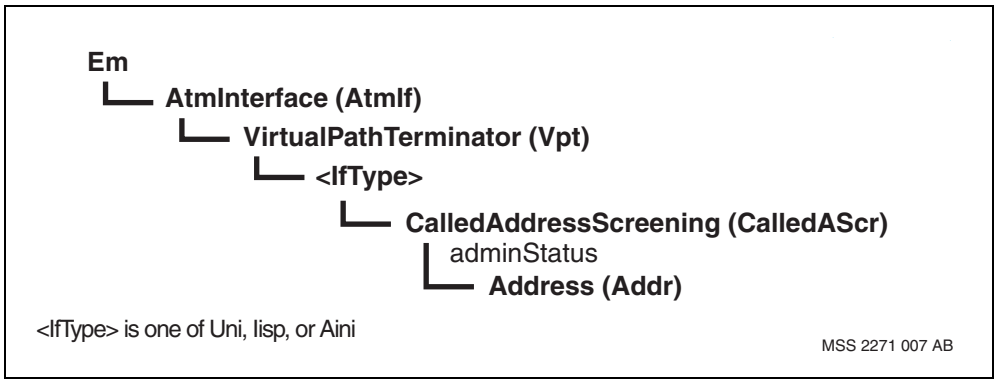
--End--

Variable definitions

Variable	Value
<action>	is accept or reject.
<address>	is a destination ATM address.
<IfType>	is Uni, Iisp, or Aini
<n>	is the number of the ATM interface.

Procedure job aid

Called address screening component hierarchy



Configuring calling address screening

Configure calling address screening for a UNI, IISP, or AINI interface to screen incoming calls against configured addresses.

By default, calling address screening is disabled, which means that calls are not screened.

When you enable calling address screening, incoming calls are screened against the addresses that you have configured. The incoming call is accepted if the calling address matches an address that you have configured to be accepted. The incoming call is rejected if the calling address matches an address you have configured to be rejected, does not match a configured address, or the connection request does not contain a calling address and there is no configured default insertion address.

Procedure steps

Step	Action
1	Configure the interface for calling address screening. **`add AtmIf/<n> <IfType> CallingAScr`**
2	Specify a default address that is to be inserted into the call connection request if calling address screening is enabled and the connection request does not contain a calling address. **`set AtmIf/<n> <IfType> CallingAScr defaultInsertionAddress <defInsAddr>`**
3	Add an ATM address to the screening rules. **`add AtmIf/<n> <IfType> CallingAScr Address/ <address>,<action>`**
4	Enable calling address screening for this ATM interface. **`set AtmIf/<n> <IfType> CallingAScr adminStatus enabled`**

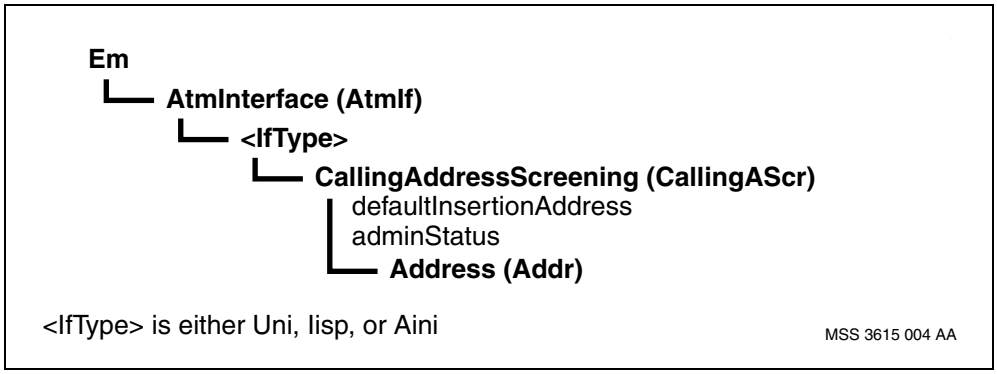
--End--

Variable definitions

Variable	Value
<action>	is accept or reject.
<address>	is a source ATM address.
<defInsAddr>	is the default insertion address that needs to be 0 or 20 bytes (40 hexadecimal digits) in length.
<IfType>	is Uni, Iisp or Aini
<n>	is the number of the ATM interface.

Procedure job aid

Calling address screening component hierarchy



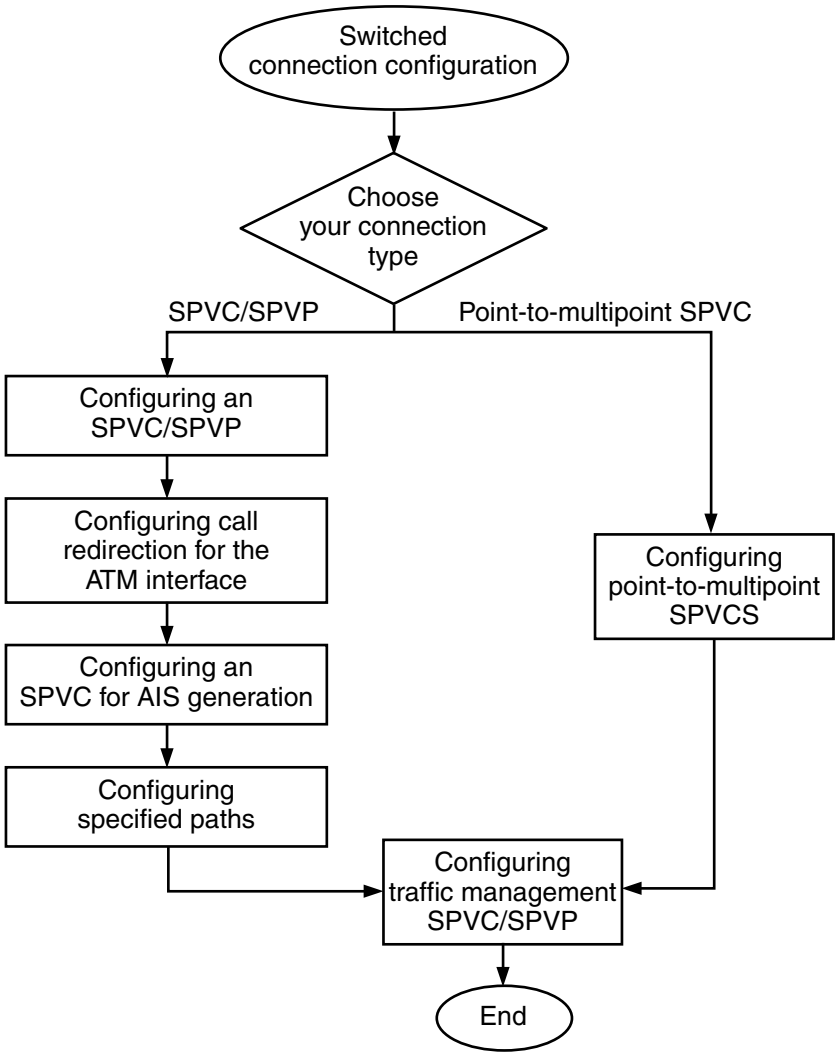
Switched connection configuration

Configure switched connections to enable the dynamic creation of virtual connections using an ATM routing protocol.

Switched connection configuration procedures

This task flow shows you the sequence of procedures you perform to configure switched connections. To link to any procedure, go to Switched connection configuration procedure navigation (page 162).

Switched connection configuration procedures



Switched connection configuration procedure navigation

- Configuring an SPVC/SPVP (page 163)
- Configuring call redirection for the ATM interface (page 167)
- Configuring an SPVC/SPVP for AIS generation (page 169)
- Configuring specified paths (page 171)
- Configuring point-to-multipoint SPVCs (page 173)
- Configuring traffic management for SPVCs and SPVPs (page 175)

Configuring an SPVC/SPVP

Configure an SPVC/SPVP to enable the dynamic creation of switched connections between ATM network elements.

Prerequisites

- Before completing this procedure, consider the following points:
  - — At the source end of the connection, configure a *SourcePvc* component under a *Vcc* component. The *SourcePvc* component defines the destination's address and its VPI.VCI instance in the destination connection map address space. A *Vcd* component defines the connection's traffic requirements.
  - — As network routing and signaling establishes the SPVC connection across the network, the nodes along the connection route create dynamic *Vcc* and *RelayPoint* components. If the destination of the SPVC is on a Nortel Multiservice Switch node, the system creates a *DestinationPvc* component under the *Vcc* component on the destination node.
  - — The SPVC lock and unlock capability introduces the *adminControl* attribute under the *SourcePVC* component that enables the operator to specify the initial state of the connection after initial configuring, FP restarts, and software reloads occur.

Procedure steps

Step	Action
1	On the destination node, add a static address under the appropriate interface and label it as an address that can be used to terminate SPVCs. **add AtmIf/<n> <IfType> Address/<address>,<addr_type> TerminateSpvpAndSpvc**
2	On the source node, specify how often an SPVC attempts to set up a call after the call setup fails and there has been one attempted retry. **set AtmIf/<n> <IfType> softPvpAndPvcRetryPeriod <seconds>**
3	Define this connection as the source end of an SPVC/SPVP **add AtmIf/<n> <connection_type>/<x> Src**
4	Identify the source end for this SPVC/SPVP by specifying its address. **set AtmIf/<n> <connection_type>/<x> Src callingAddress <callingA>**
5	Identify the destination end for this SPVC/SPVP by specifying its address.

```
set AtmIf/<n> <connection_type>/<x> Src calledAddress
<calledA>
```

6 Identify the identifier associated with the called (or destination) interface.

```
set AtmIf/<n> vcc/<x> Src calledVpiVci
<identifier_value>
```

```
set AtmIf/<n> vpc/<x> Src calledVpi <identifier_value>
```

7 Optionally, set the initial state of the connection.

```
set AtmIf/<n> <connection_type>/<x> Src adminControl
<adminControl>
```

If the connection is locked, the VPI and VCI numbers at the source node are reserved without allocating bandwidth to the connection when the connection is configured, regardless of the operational state of the connection.

--End--

Variable definitions

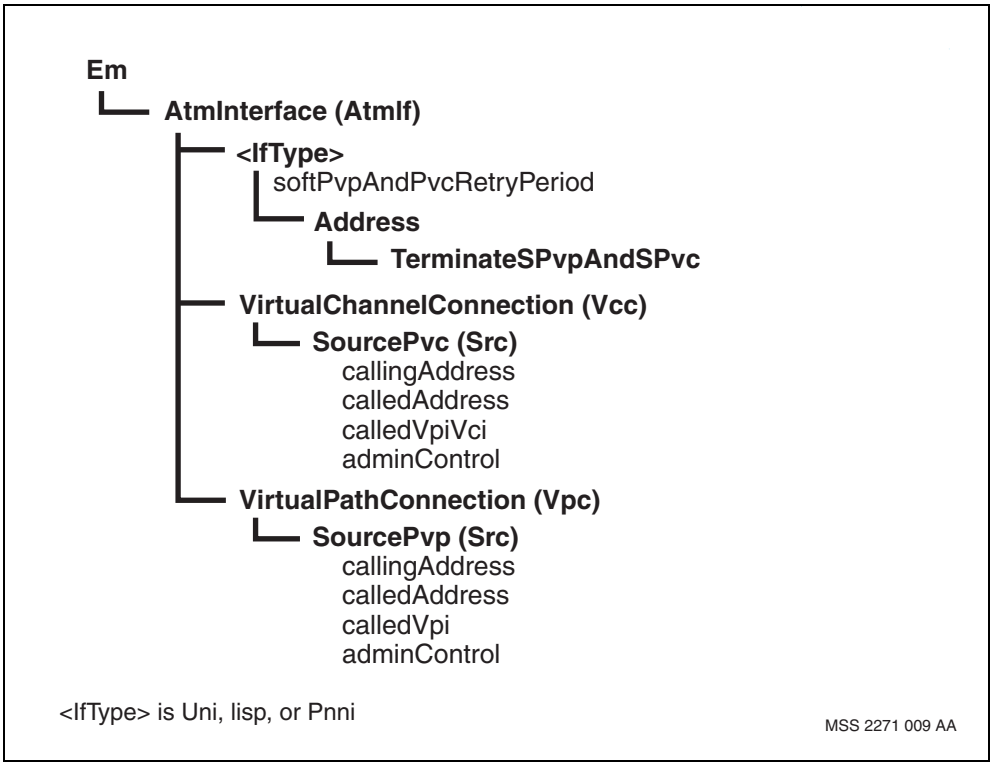
Variable	Value
<address>	is a static address associated with the interface.
<adminControl>	is the initial state of the connection. The value can be permlocked or unlocked. Permlocked means that the connection is always locked and remains locked even after the FP restarts and the software reloads. Unlocked means that the connection is initially unlocked and remains unlocked even after the FP restarts and the software reloads.
<addr_type>	is primary or alternate.
<calledA>	is a default address associated with the destination interface, consisting of 40 hexadecimal digits.
<callingA>	is the address associated with the source interface, and consists of 40 hexadecimal digits. If you do not specify an address value, the node uses the default address value.
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<identifier_value>	is the instance value of the virtual channel associated with the remote end-point.
<IfType>	is uni, Iisp, Aini, or Pnni.
<n>	is the number of the ATM interface.

(1 of 2)

Variable	Value
<seconds>	is a decimal value between 20 and 999 999. The default value is 60. Resolution is to the nearest 10 s. For example, if you enter 35, the system sets the configured value to 40 s; if you enter 34, the system sets the configured value to 30 s. When configuring a large number of SPVCs on the node, it is important that all of the interfaces have different softPvpAndPvcRetryPeriod attributes. These attributes should be configured such that they can not be rounded up or down to the nearest multiple of 10. For example, configure 20, 30, and 70, instead of 21, 22, and 24. This configuring process enables the timing of the call set-up retries to be spread out to ease congestion on the device.
<x>	is the instance value of the vcc or vpc.
	(2 of 2)

Procedure job aid

SPVC/SPVP component hierarchy



Configuring call redirection for the ATM interface

Configure call redirection for the ATM interface to add a secondary destination address to be used if the primary destination connection fails.

After the call redirection for the ATM interface connection is configured, the lock and unlock capability of the *CallRedirection* component allows you to include or exclude the use of the primary or secondary destination address during call setup attempts.

Procedure steps

Step	Action
1	Add the *SourcePvc*/SourcePvp component to the connection, if necessary. **add Atmif/<n> <connection_type>/<x> Src**
2	Add the *CallRedirection* component. **add Atmif/<n> <connection_type>/<x> Src CallRedirection**
3	Set the secondary called address. **set Atmif/<n> <connection_type>/<x> Src CallRedirection secondaryCalledAddress <sec_address>**
4	Set the secondary called vpi.vci for Vccs or secondary called Vpi for Vpts. **set Atmif/<n> vcc/<x> Src CallRedirection secondaryCalledVpiVci <sec_vpi.sec_vci>** **set Atmif/<n> vpc/<x> Src CallRedirection secondaryCalledVpi <sec_vpi>**

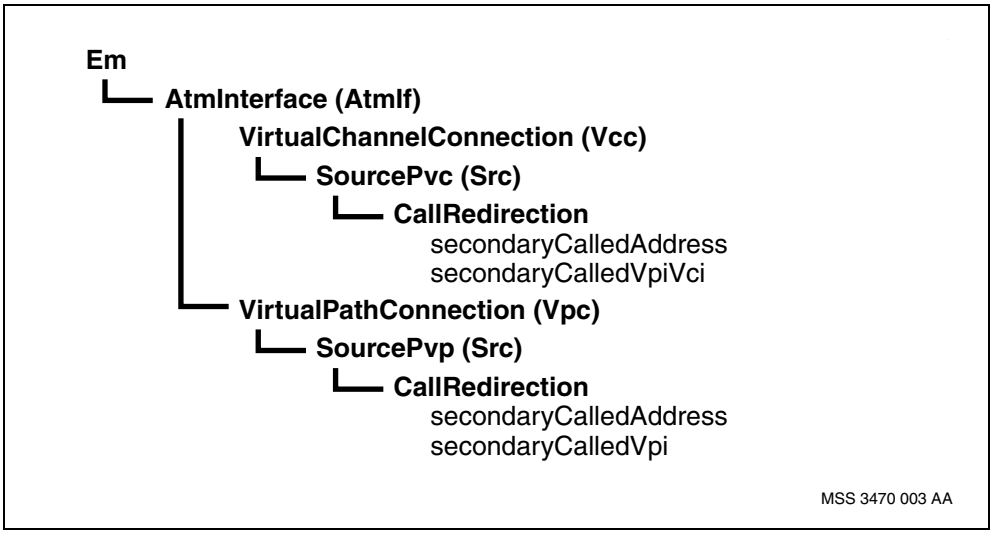
--End--

Variable definitions

Variable	Value
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<n>	is the instance value of the Atmif component.
<sec_address>	is the value of the secondary called address attribute.
<sec_vpi.sec_vci>	is the value of the secondary called vpi.vci.
<sec_vpi>	is the value of the secondary called vpi.
<x>	is the instance value of the vcc or vpc.

Procedure job aid

Call redirection for ATM interface component hierarchy



Configuring an SPVC/SPVP for AIS generation

Configure an SPVC/SPVP for AIS generation to create alarm indication signals.

Prerequisites

- To avoid service degradation and potential traffic loss on a given SPVC connection, ensure that the configured traffic management parameters on a configured destination end match the traffic management parameters configured at the source end. All Tx (transmit) parameters at the destination end correspond to the Rx (receive) parameters at the source end. All Tx parameters from the source end must correspond to the Rx parameters at the destination end.
- If you want AIS generation enabled at both ends, both the source and destination must be enabled.
- If you want AIS generation disabled at both ends, both the source and destination must be disabled.

Procedure steps

Step	Action
1	Add a destination SPVC. **add atmif/<n> <connection_type>/<x> dst**
2	Enable AIS at the source SPVC. **set atmif/<n> <connection_type>/<x> src aisGeneration enable**
3	Enable AIS at the destination SPVC. **set atmif/<n> <connection_type>/<x> dst config aisGeneration enable**
4	Disable AIS at the source SPVC. **set atmif/<n> <connection_type>/<x> src aisGeneration disable**
5	Disable AIS at the destination SPVC. **set atmif/<n> <connection_type>/<x> dst config aisGeneration disable**

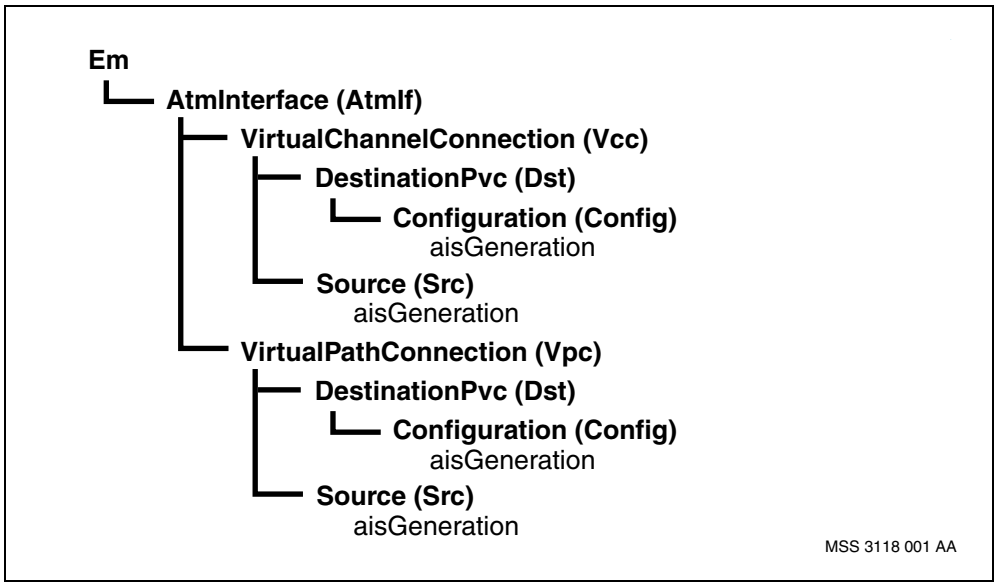
--End--

Variable definitions

Variable	Value
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<x>	is the instance value of the Vcc. The VPI value can be from 0 to 255. The VCI value can be from 32 to 65 535. For Apc based cards (4pOC12 and 16pOC3), the VCI value can be from 32 to 16 383.

Procedure job aid

SPVC/SPVP for AIS generation component hierarchy



Configuring specified paths

Configure a specified path connection to manually configure a path with a predetermined route for a SPVC/SPVP call.

Prerequisites

- For information about attributes that may affect specified path connections, see Specified path connection attributes (page 219).

Procedure steps

Step	Action
1	Add the *Mdtl* component under the connection or to the existing SPVC/P connection. **`add AtmIf/<n> <connection_type>/<x> src mdtl`**
2	Assign a primary path to the connection **`set AtmIf/<n> <connection_type>/<x> src mdtl primaryPath <primarypathname>`**
3	Optionally, assign an alternate path to the connection. **`set AtmIf/<n> <connection_type>/<x> src mdtl alternatePath <alternatepathname>`**
4	Optionally, enable/disable the automatic fallback path. **`set AtmIf/<n> <connection_type>/<x> src mdtl automaticFallback <automaticfallback>`**

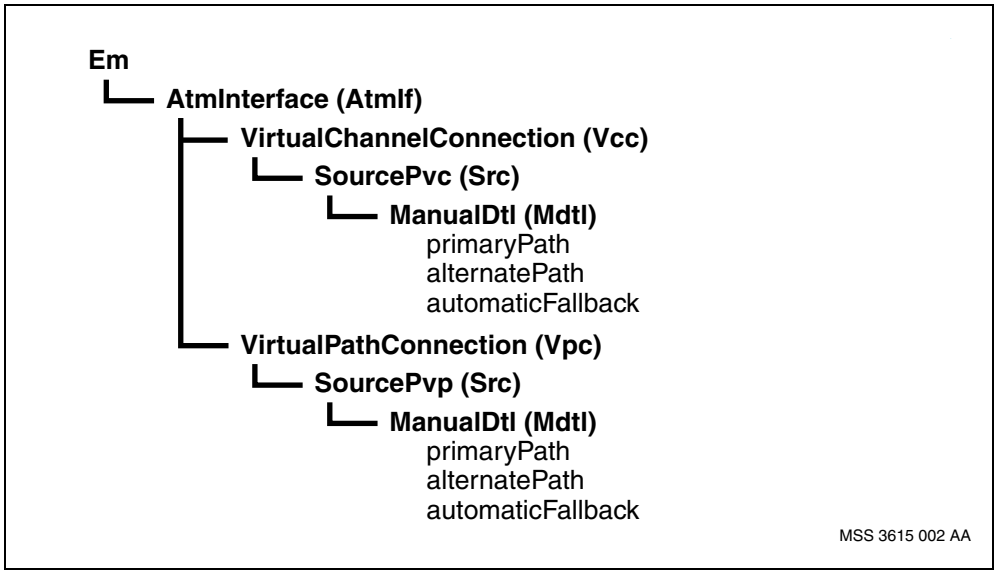
--End--

Variable definitions

Variable	Value
<alternatepathname>	is the name of the alternate path.
<automaticfallback>	is enabled or disabled. By default, the *automaticFallback* attribute is set to enabled.
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095.
<primarypathname>	is the name of the primary path.
<x>	is the instance value of the vcc or vpc.

Procedure job aid

Specified paths component hierarchy



Configuring point-to-multipoint SPVCs

Configure point-to-multipoint SPVCs with a single source and multiple destination points.

Procedure steps

Step	Action
1	Add the *MulticastSourcePvc* component. **add AtmIf/<n> vcc/<vcc> msrc**
2	Set the calling address of the *MulticastSourcePvc* component. **set Atmif/<n> vcc/<vcc> msrc callingAddress <addr>**
3	Set the *mCastConnectionType* attribute for the new vcc. **set atmif/x vcc/<vcc> Vcd mCastConnectionType pmpRoot**
4	Add the *party* subcomponents. **add AtmIf/<n> vcc/<vcc> msrc party/<m>**
5	Set the called address. **set atmif/<n> vcc/<vcc> msrc party/<m> calledAddress <addr_Leaf>**
6	Set the calledVpiVci. **set atmif/<n> vcc/<vcc> msrc party/<m> calledVpiVci <x1.y1>**
7	Repeat steps 6, 7, and 8 for each required party. When configuring multiple parties, no two parties from the same msrc can have an identical combination of *calledaddress* and *calledvpivci* attributes.

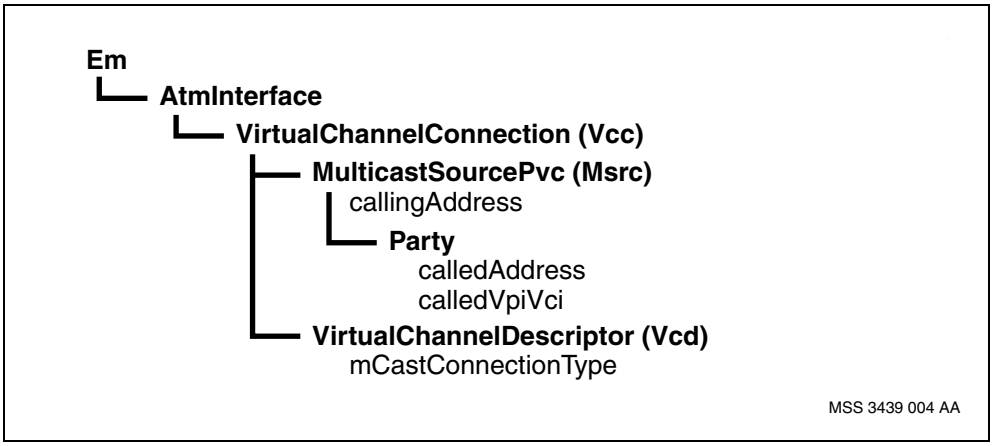
--End--

Variable definitions

Variable	Value
<addr>	is the address associated with the source interface, and consists of 40 hexadecimal digits. If you do not specify an address value, the node uses the default address value.
<addr_Leaf>	is an address associated with the destination interface, consisting of 40 hexadecimal digits.
<m>	is a value between 1 and 300.
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095.
<vcc>	the instance value defines the VPI and VCI values for the connection.
<x1.y1>	is the instance value of the virtual channel associated with the remote end-point. The VPI value can be from 0 to 255. The VCI value can be from 32 to 65 535. For Apc based cards (4pOC12 and 16pOC3), the value can be from 32 to 16 383. This VPI.VCI must be within the defined connection map address space on the remote interface.

Procedure job aid

Point-to-multipoint SPVCs component hierarchy



Configuring traffic management for SPVCs and SPVPs

Configure traffic management for SPVCs and SPVPs to enable shaping and policing for virtual connections.

Procedure steps

Step	Action
1	Set the ATM service category for the connection. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm atmServiceCategory <atmS>**
2	Set the traffic descriptor type for the transmit direction of the channel. *TxTdt* must be 1, 2 or 9. The default is 1. **set AtmIf/<n> <connection_type>/<x> <descriptor_type>d Tm txTrafficDescType <txTdt>**
3	Set the traffic descriptor parameters for the transmit direction of the channel. All Tx descriptor parameters must be set to 0. As a result, if txTdt is set to 9 for PMP SPVCs then the txTrafficDescParms must not be set to any value other than 0. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm txTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>] [4 <parm4>] [5 <parm5>]**
4	Set the traffic descriptor type for the receive direction of the channel. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm rxTrafficDescType <rxTdt>**
5	Set the traffic descriptor parameters for the receive direction of the channel. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm rxTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>] [4 <parm4>]**
6	Set the usage parameter control. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> Tm upc <upc>**
7	Set the backwards Qos class. The backward Qos class must be 0. If it is set to sameAsFwd, and the fwdQosClass attribute is not 0, then set the bwdQosClass attribute to 0. **set AtmIf/<n> <connection_type>/<x> <descriptor_type> tm bwdQosClass 0**
8	Set the backwards Qos parameters. Use the default values for the backward Qos parameters. If you are not using the default values, then set the cdv to 16777215 and the clr to 255.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> tm
bwdQosParameters cdv 16777215 clr 255
```

9 Set the packet wise discard in the transmit direction to disabled.

```
set AtmIf/<n> <connection_type>/<x> <descriptor_type> tm
txPacketWiseDiscard disabled
```

By default, the txPacketWiseDiscard attribute is disabled.

--End--

Variable definitions

Variable	Value
<atmS>	is Cbr, Rtvbr, Nrtvbr, or Ubr. The default is Ubr.
<connection_type>	is either the *Vcc* component, if you are configuring a virtual channel connection, or the *Vpc* component, if you are configuring a virtual path connection.
<descriptor_type>	is vcd (for a virtual channel connection) or vpd (for a virtual path connection).
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095.
<parm1>	is traffic descriptor parameter 1 (peak cell rate) between 0 and 2 147 483 647. The default is 0.
<parm2>	is traffic descriptor parameter 2 (PCR0 or SCR) between 0 and 2 147 483 647. The default is 0.
<parm3>	is traffic descriptor parameter 3 (MBS) between 0 and 2 147 483 647. The default is 0.
<parm4>	is traffic descriptor parameter 4 (CDVT) between 0 and 10 000 to 1 200 000. In provisioning mode, the default value is 0. However, in operational mode the parameter takes on the value of atmIf/x ca ubr/0 cdvt which is 250. If you are going to enable UPC on the Vcc, the traffic descriptor parameters for the receive direction must take UPC into account.
<parm5>	is traffic descriptor parameter 5 (requested shaping rate) between 0 and 2 147 483 647. This parameter is not applicable to standard VPT VCCs. The default is 0.
<rxTdt>	is a value between 1 and 9 that defines the traffic descriptor type. The default is 1. Traffic descriptor 9 is to be used exclusively by UBR with MDCR. When rxTdt is 9 for UBR with MDCR, <parm1> is PCR, <parm2> is CDVT, and <parm3> is MDCR. If the value of the *rxTrafficDescType* attribute is *sameAsTx*, the parameters for this attribute are the same as the parameters in the *txTrafficDescParm* attribute. The encoding value that the node signals for each parameter is three bytes long, which imposes a maximum limit of 16 777 215 for each parameter value. The node truncates larger values.

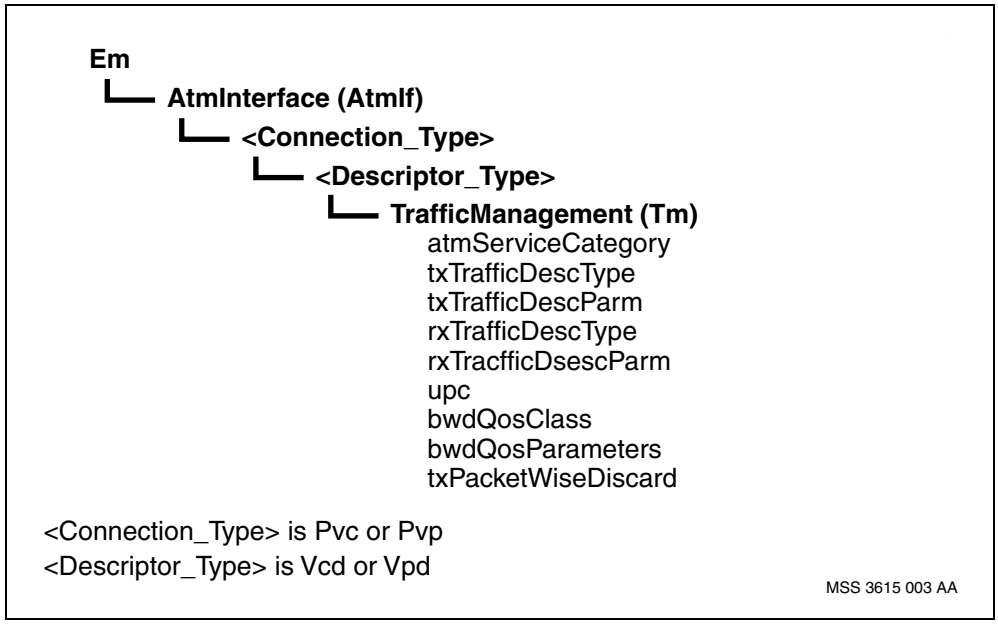
(1 of 2)

Variable	Value
<txTdt>	is a value between 1 and 9 that defines the traffic descriptor type. The default is 1. The traffic descriptor type and traffic descriptor parameters have default values that for the transmit direction. When txTdt is 9 for UBR with a minimum desired cell rate (MDCR), <parm1> is PCR, <parm2> is CDVT, and <parm3> is MDCR. Traffic descriptor 9 is to be used exclusively by UBR with MDCR.
<upc>	is enforced, disabled, sameAs, or monitored.
<x>	is the instance value of the virtual channel associated with the remote end-point. The VPI value can be from 0 to 255. The VCI value can be from 32 to 65 535. For Apc based cards (4pOC12 and 16pOC3), the value can be from 32 to 16 383. This VPI.VCI must be within the defined connection map address space on the remote interface.

(2 of 2)

Procedure job aid

Traffic management for SPVCs and SPVPs component hierarchy



ATM accounting configuration

Configure ATM accounting to set the nodes to collect accounting statistics for ATM connections.

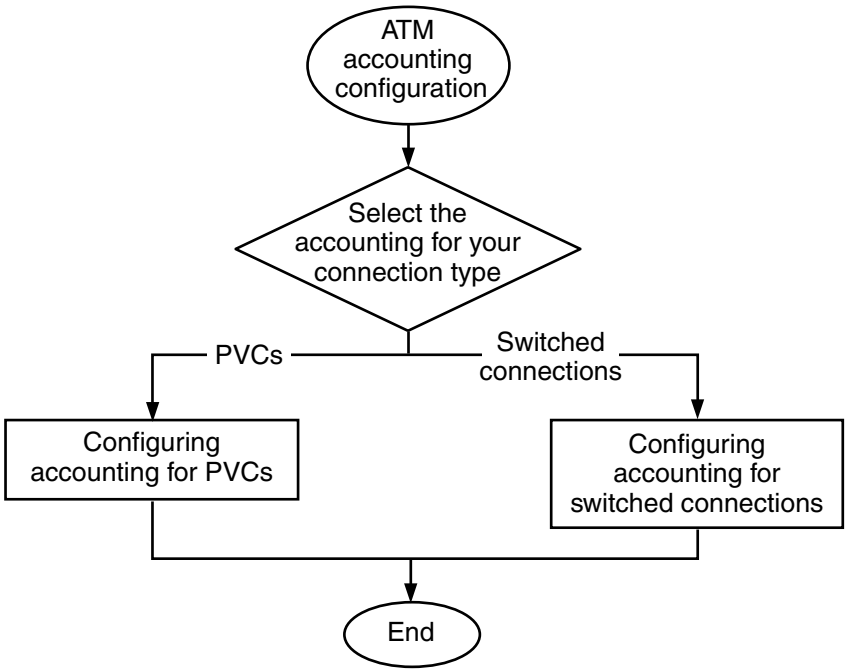
Prerequisites to ATM accounting configuration

- Configure basic accounting by following the procedures in NN10600-560 *Nortel Multiservice Switch 7400/15000/20000 Accounting*.
- For information on ATM accounting, see ATM accounting (page 231).

ATM accounting configuration procedures

This task flow shows you the sequence of procedures you perform to configure ATM accounting. To link to any procedure, go to ATM accounting configuration procedure navigation (page 180).

ATM accounting configuration procedures



ATM accounting configuration procedure navigation

- Configuring accounting for PVCs (page 181)
- Configuring accounting for switched connections (page 184)

Configuring accounting for PVCs

Configure accounting for PVCs to collect accounting statistics on ATM permanent virtual connections.

Procedure steps

Step	Action
1	Add the *NailedUpAccounting* component. **add AtmIf/<n> NAcct**
2	Enable or disable accounting at the interface level. **set AtmIf/<n> NAcct accountCollection <acReason>**
3	Set the correlation tag for this end of the PVC. You must configure the same correlation tag value at each end of the PVC connection. For a virtual path termination (*Vpt*), use the following command: **set AtmIf/<n> Vpt/<x> Vpd correlationTag <tag>** For a virtual channel connection (*Vcc*), use the following command: **set AtmIf/<n> Vcc/<x> Vcd correlationTag <tag>** For a virtual path connection (*Vpc*), use the following command: **set AtmIf/<n> Vpc/<x> Vcd correlationTag <tag>**
4	Set the accounting class for tagging purposes. **set AtmIf/<n> NAcct accountClass <class>**
5	Set the data service exchange for tagging purposes. **set AtmIf/<n> NAcct serviceExchange <sE>**
6	Specify whether the accounting record corresponds to an originating, terminating or intermediate connection point. **set AtmIf/<n> NAcct accountConnectionType <acType>**

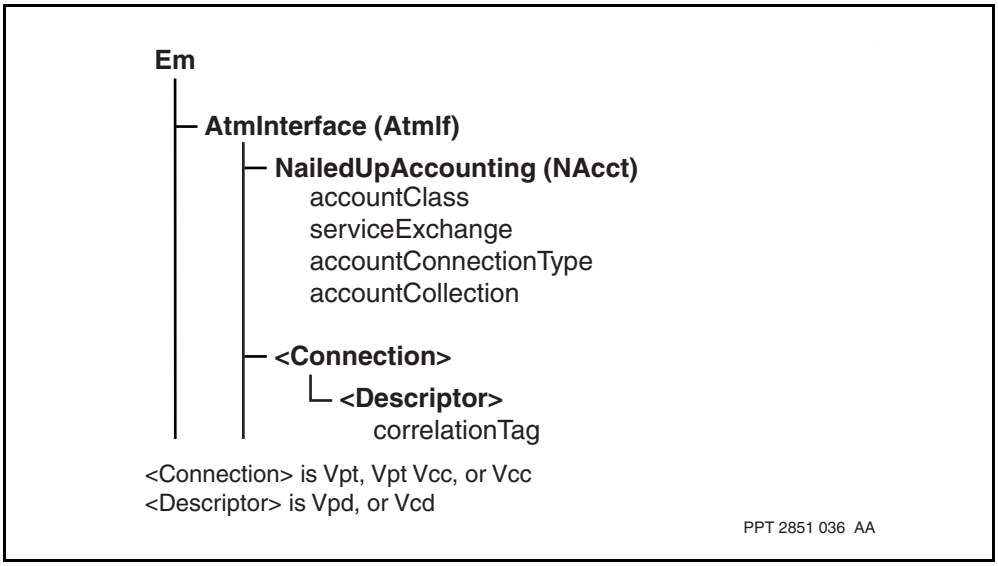
--End--

Variable definitions

Variable	Definition
<acReason>	is one or more of bill, test, study, audit or force to enable accounting. The accountCollection attribute has no default value. If you do not set a value for the accountCollection attribute, Multiservice Switch systems do not collect accounting statistics for this interface component.
<acType>	*is origTerm* or *intermediate*. Use *origTerm* for an interface on an edge node. Use *intermediate* for an interface on an inner node. The default is *origTerm* for Uni interfaces, and *intermediate* for Iisp and Aini interfaces.
<class>	is the value of the accounting class, determined by the network operator (0 to 255). The default value is 0.
<n>	is the instance value of the *AtmInterface* component (any unique value from 1 to 4 095).
<sE>	is the value of the service exchange, determined by the network operator (0 to 255). The default value is 0.
<tag>	is the correlation tag for the PVC (any 64-byte string).
<x>	is the instance of the connection. If the type of connection is a Vpc or a Vpt, <x> represents the VPI value. If the type of connection is a Vcc, <x> represents the VCI value.

Procedure job aid

Accounting for permanent virtual connections component hierarchy



Configuring accounting for switched connections

Configure accounting for switched connections to allow the collection of accounting statistics on ATM switched connections.

Procedure steps

Step	Action
1	Set the accounting class for tagging purposes. **`set AtmIf/<n> <IfType> accountClass <class>`**
2	Set the data service exchange for tagging purposes. **`set AtmIf/<n> <IfType> serviceExchange <sE>`**
3	Set the accountConnectionType only if the accounting record corresponds to an originating, terminating or intermediate connection point. **`set AtmIf/<n> <IfType> accountConnectionType <acType>`**
4	Enable or disable accounting at the interface level. **`set AtmIf/<n> <IfType> accountCollection <acReason>`**

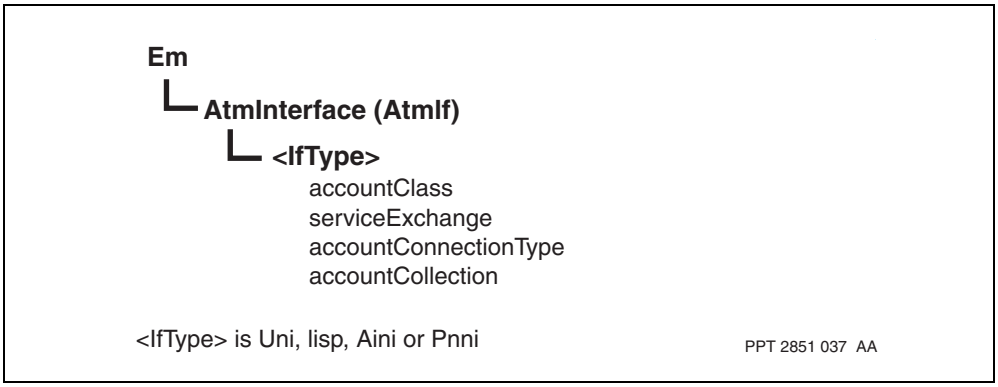
--End--

Variable definitions

Variable	Definition
<acReason>	is one or more of bill, test, study, audit or force to enable accounting. The accountCollection attribute has no default value. If you do not set a value for the accountCollection attribute, Nortel Multiservice Switch systems do not collect accounting statistics for this interface component.
<acType>	*is origTerm* or *intermediate*. The default is *origTerm* for UNIs, and *intermediate* for IISP interfaces, AINIs, and PNNIs. The defined value must be correct even if accounting is disabled. Use *origTerm w*hen a Multiservice Switch node is connected to a non-Multiservice Switch device.
<class>	is the accounting class, determined by the network operator (0 to 255). The default value is 0.
<ifType>	is Uni, Iisp, Aini, or Pnni.
<n>	is the instance value of the *AtmInterface* component (any value from 1 to 4 095).
<sE>	is the service exchange, determined by the network operator (0 to 255). The default value is 0.

Procedure job aid

Accounting for switched connections component hierarchy



Hitless ATM services on FPs

Use the following sections to learn more about hitless ATM services on FPs.

Navigation

Providing hitless ATM services on FPs with optical interfaces

Provide hitless ATM services on FPs with optical interfaces to enable hitless ATM services on Nortel Multiservice Switch nodes. See NN10600-700 *Nortel Multiservice Switch 7400/15000/20000 ATM Technology Fundamentals* for a description of which ATM services can be hitless.

Procedure steps

Step	Action
1	Ensure that the main and spare FPs have the same card type and support dual-FP line APS. See NN10600-120 *Nortel Multiservice Switch 15000/20000 Hardware Description* to see which FPs support dual-FP line APS.
2	Ensure that the main and spare FPs are in the correct predetermined card slots. See NN10600-130 *Nortel Multiservice Switch 15000/20000 Hardware Installation, Maintenance, and Upgrade* for directions on choosing the card slots.
3	Configure one LP for the working line optical interface. See NN10600-550 *Nortel Multiservice Switch 7400/15000/20000 Common Configuration Procedures* for the procedure. Make sure that the LP does not mix any cold standby applications with hot or warm standby applications.
4	Configure a second LP for the protection optical interface. See NN10600-550 *Nortel Multiservice Switch 7400/15000/20000 Common Configuration Procedures* for the procedure. Make sure that the LP does not mix any cold standby applications with hot or warm standby applications.
5	Configure line APS for dual-FP protection.

For a new service, this means configuring a *Laps* component to link the corresponding ports of both LPs. See NN10600-550 *Nortel Multiservice Switch 7400/15000/20000 Common Configuration Procedures* for the procedure.

For an existing service with single-FP line APS, this means converting single-FP line APS to dual-FP line APS, see NN10600-551 *Nortel Multiservice Switch 7400/15000/20000 FP Configuration Reference* for the procedure.

6 Configure the ATM service interface.

For an existing ATM service, adjust the FP resources to account for hitless services. See Adjusting resources for hitless services (page 189) for the procedure.

7 Link the resulting *AtmIf* component from step 6 to the *Sts* or *Vc4* subcomponent of the *Laps* component from step 2.

The ATM interface now provides hitless services. The services are hitless during equipment protection switchovers, line APS switchovers and software migrations.

--End--

Providing hitless ATM services on FPs with electrical interfaces

Provide hitless ATM services on FPs with electrical interfaces to enable hitless ATM services on Nortel Multiservice Switch nodes. See NN10600-700 *Nortel Multiservice Switch 7400/15000/20000 ATM Technology Fundamentals* for a description of which ATM services can be hitless.

Procedure steps

Step	Action
1	Ensure that the main and spare FPs have the same card type and support one-for-one sparing. See NN10600-120 *Nortel Multiservice Switch 15000/20000 Hardware Description* to see which FPs support one-for-one sparing.
2	Ensure that the sparing panel has been installed and the main and spare FPs are connected to it. See NN10600-130 *Nortel Multiservice Switch 15000/20000 Hardware Installation, Maintenance, and Upgrade* for directions on installing the sparing panel.
3	Configure the main FP for one-for-one equipment sparing using an LP that does not mix any cold standby applications with hot or warm standby applications. See NN10600-550 *Nortel Multiservice Switch 7400/15000/20000 Common Configuration Procedures*.
4	Configure the spare FP for one-for-one equipment sparing. See NN10600-550 *Nortel Multiservice Switch 7400/15000/20000 Common Configuration Procedures* for the procedure.
5	Configure the ATM service interface. For an existing ATM service, adjust the FP resources to account for hitless services. See Adjusting resources for hitless services (page 189) for the procedure.
6	Link the resulting *AtmIf* component from step 5 to the appropriate *Ds3* or *E3* subcomponent of the LP from step 2. The ATM interface now provides hitless services. The services are hitless during equipment protection switchovers and software migrations.

--End--

Adjusting resources for hitless services

Adjust resources for hitless services to alter sparing status of an existing ATM interface changes from spared to unspared, or unspared to spared. If the sparing status of many ATM interfaces are changing, repeat this procedure for each ATM interface that is affected.

When the sparing status of an existing ATM interface changes, the value of some attributes of the *AtmResourceControl Override (Arc OV)* component must be adjusted. This ensures that there are sufficient connection resources allocated for both the spared and unspared connections.

The sparing status of an existing ATM interface can change because:

- electrical equipment protection is introduced
- optical dual-FP line APS is introduced
- single-FP line automatic protection switching (line APS) is converted to dual-FP line APS
- the *Software Avl* component is changed so that cold standby applications or features are mixed with hot or warm standby applications or features

Attention: More than one step in this process can cause an FP to reboot. For example, introducing electrical equipment protection and changing the attribute values of the Arc OV component both cause the FP to reboot. All the steps of this process should be coordinated to minimize service interruptions.

Prerequisites

WARNING
This procedure resets the function processor
Configure the *Arc Ov* component during periods of low traffic volume to minimize service interruptions.

- See NN10600-550 *Nortel Multiservice Switch 7400/15000/20000 Common Configuration Procedures* for a description of:
  - hitless services
  - single-FP and dual-FP line APS
  - hot, warm and cold standby applications and features
  - one-for-one sparing
- Ensure that all the preparatory steps for hitless services have been done as described in Providing hitless ATM services on FPs with optical

interfaces (page 186) or Providing hitless ATM services on FPs with electrical interfaces (page 188). This includes the following:

— All required hardware is installed.

— Dual-FP line APS or one-for-one electrical equipment sparing is configured with LPs. Although you can create an LP that mixes cold standby features with hot standby or warm standby features, Nortel Networks does not recommend this action.

— The ATM interface is configured to provide services.

Procedure steps

Step	Action
1	Ensure that you are in operational mode.
2	Determine how many connections are currently supported on the ATM interface. **display -p AtmIf/<n> Ca maxVccs, maxVpcs, maxVpts, minMulticastBranches** The settings for the ATM interface appear. maxVccs = 255 maxVpcs = 128 maxVpts = 128 minMulticastBranches = 25
3	If the maxVccs, maxVpcs, and maxVpts are set to autoConfigure, you may view the actual values. **display Atmif/<n> Ca actualMaxVpcs, actualMaxVpts, actualMaxVccs**
4	Display the existing attribute values for the protected connection pools. **display -p Lp/<lp> Eng Arc Ov** The settings for the protected connection pools for ATM interface appear. connectionPoolCapacity = 6511 protectedConnectionPoolCapacity = 0 multicastBranchesCapacity = 100 protectedMcastBranchesCapacity= 25 txCellMemoryAllocation = 50% rxCellMemoryAllocation = 20%
5	Enter provisioning mode. **start prov**
6	The value of the *protectedConnectionPoolCapacity* attribute must be large enough to accommodate the additional protected VCC and VPC connections. If required, increase value of the

protectedConnectionPoolCapacity attribute by the total of *maxVccs* and *maxVpcs* from step 1.

```
set Lp/<lp> Eng Arc Ov protectedConnectionPoolCapacity
383
```

7 Reduce the value of the *connectionPoolCapacity* attribute by the same amount as in step 6 to release cell queue memory (CQM) for frame and cell buffering.

```
set Lp/<lp> Eng Arc Ov connectionPoolCapacity 6128
```

8 The value of the *protectedMcastBranchesCapacity* attribute must be large enough to accommodate the number of additional multicast branches for spared VCC and VPC ATM interfaces. If required, increase value of the *protectedMcastBranchesCapacity* attribute by the value of the *minMulticastBranches* attribute from step 1.

```
set Lp/<lp> Eng Arc Ov protectedMcastBranchesCapacity 25
```

9 Reduce the value of the *multicastBranchesCapacity* attribute by the same amount as in step 6 to release CQM for frame and cell buffering.

```
set Lp/<lp> Eng Arc Ov multicastBranchesCapacity 75
```

10 Check, activate and confirm the provisioning changes. See Adjusting resources for hitless services (page 189).

--End--

Variable definitions

Variable	Value
<lp>	is the number of the LP on which the interface is configured.
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4095.

Configuration examples

This section describes high-level examples of configuring an ATM bearer service. Information appears in the following sub-sections:

Each of the examples provided in this section include the following information:

- a general statement of the type of connections or network topology
- an illustration of the connection of network topology
- a configuration example

Sample values for components and attributes are used for all examples.

Navigation

Example 1: Logical ABS with NRPs using VCCs

This example shows how to configure a 64 kb/s virtual channel connection (VCC) with a real-time variable bit rate (rt-VBR) ATM service category. See the figure Configuring logical ABS service with Nrps using VCCs (page 194).

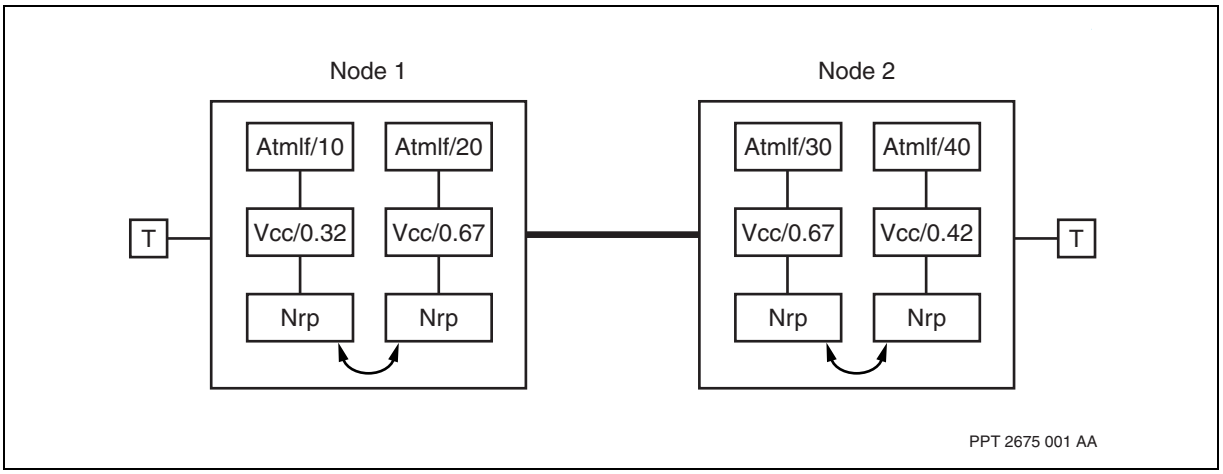
Procedure steps

Step	Action
1	On node 1, create the ATM interface component *AtmIf/10* and link it to the port. **add AtmIf/10** **set AtmIf/10 oamSegmentBoundary yes** **set AtmIf/10 interfaceName lp/1 Sdh/0 path/0**
2	Configure the connection map as required, or use the default.
3	Create a *Vcc* component and define the traffic and *atmServiceCategory* parameters by setting the attributes of *Vcd* component. **add AtmIf/10 Vcc/0.32** **set AtmIf/10 Vcc/0.32 Vcd Tm txTrafficDescType 3** **set AtmIf/10 Vcc/0.32 Vcd Tm txTrafficDescParam 1 167** **set AtmIf/10 Vcc/0.32 Vcd Tm atmServiceCategory rtVariableBitRate**
4	Create an *Nrp* component. **add AtmIf/10 Vcc/0.32 Nrp**
5	Add *AtmIf/20, Vcc/0.67* and another *Nrp* component in the same way.
6	Then link the two *Nrp* components. **set AtmIf/10 Vcc/0.32 Nrp nexthop AtmIf/20 Vcc/0.67 Nrp**
7	Create *AtmIf/30, AtmIf/40, Vcc/0.67, Vcc/0.42*, and two *Nrp* components for Node 2. Link the *Nrp* components and verify and activate provisioning.

--End--

Procedure job aid

Configuring logical ABS service with Nrps using VCCs



Example 2: Logical ABS with NRPs using VPCs

This example shows how to configure a 64 kb/s virtual path connection (VPC) with a rt-VBR ATM service category. See the figure Configuring logical ABS service with Nrps using VPCs (page 195).

Procedure steps

Step	Action
1	On node 1, create the ATM interface component *AtmIf/10* and link it to the port. **add AtmIf/10** **set AtmIf/10 oamSegmentBoundary yes** **set AtmIf/10 interfaceName lp/1 Sdh/0 path/0**
2	Configure the connection map as required, or use the default.
3	Create a *Vpc* component and define the traffic and *atmServiceCategory* parameters by setting the attributes of *Vcd* component. **add AtmIf/10 Vpc/1** **set AtmIf/10 Vpc/1 Vpd Tm txTrafficDescType 3** **set AtmIf/10 Vpc/1 Vpd Tm txTrafficDescParam 1 167** **set AtmIf/10 Vpc/1 Vpd Tm atmServiceCategory rtVariableBitRate**
4	Create an *Nrp* component. **add AtmIf/10 Vpc/1 Nrp** Add *AtmIf/20, Vpc/7* and another *Nrp* in the same way.

5 Then link the two *Nrp* components.

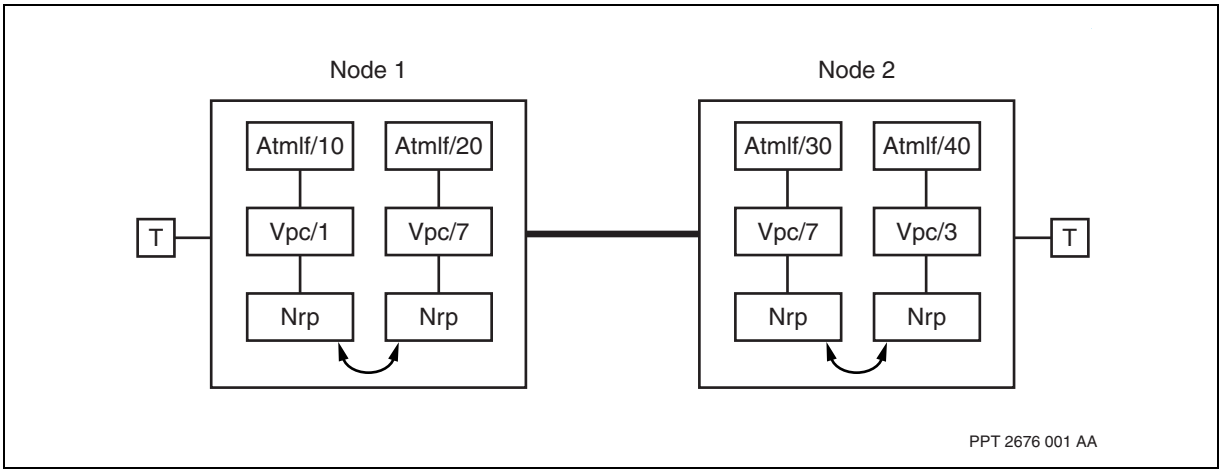
```
set AtmIf/10 Vpc/1 Nrp nexthop AtmIf/20 Vpc/7 Nrp
```

6 Create *AtmIf/30, AtmIf/40, Vpc/7, Vcc/3*, and two *Nrp* components for Node 2. Link the *Nrp* components and verify and activate provisioning.

--End--

Procedure job aid

Configuring logical ABS service with Nrps using VPCs



Example 3: Test and loop configuration using VCCs

This example shows how to configure a 64 kb/s VCC for a test loopback. See the figure Configuring test and loop using VCCs (page 196).

Procedure steps

Step	Action
1	On node 1, create the ATM interface component *AtmIf/20* and link it to the port.

```
add AtmIf/20
set AtmIf/20 oamSegmentBoundary yes
set AtmIf/20 interfaceName lp/1 Sdh/0 path/0
```

2 Configure the connection map as required, or use the default.

3 Create a *Vcc* component and define the traffic and *atmServiceCategory* parameters by setting the attributes of *Vcd* component.

```
add AtmIf/20 Vcc/0.32
set AtmIf/20 Vcc/0.32 Vcd Tm txTrafficDescType 3
set AtmIf/20 Vcc/0.32 Vcd Tm txTrafficDescParam 1 167
set AtmIf/20 Vcc/0.32 Vcd Tm atmServiceCategory RtVbr
```

4 Create a *Test* component.

```
add AtmIf/20 Vcc/0.32 Test
```

5 On Node 2, add *AtmIf/30, Vcc/0.32* and a *Loop* component in the same way.

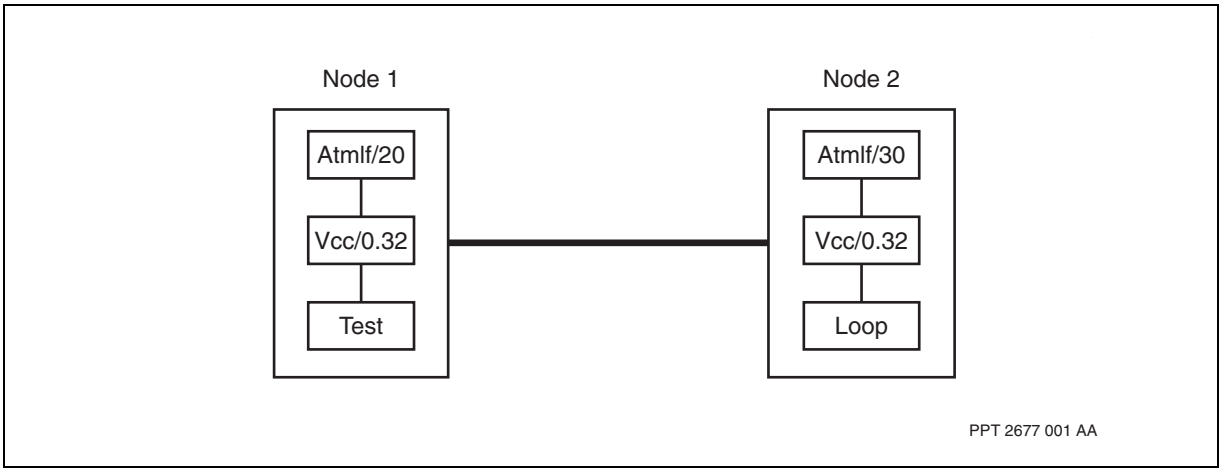
```
add AtmIf/30 Vcc/0.32 Loop
```

6 Configure and activate the Test in order to test the two ends of the VCC.

--End--

Procedure job aid

Configuring test and loop using VCCs



Example 4: Static routing in a small Multiservice Switch-only network

This example shows how to configure static routing in a small Nortel Multiservice Switch-only network. The network configuration is shown in the figure Static routing in a small Multiservice Switch-only network (page 200).

This example shows how to route a call from Node A to Node C directly, and how to route a call from Node A to Node C through Node B. The example shows two primary routes in a loadsharing configuration, in which each route has the same capacity for best-matching address.

The following procedure focuses on setting up networking and static routing requirements. Note the following assumptions:

- you have added the appropriate software feature.
- ATM resource control is configured (optional, but recommended)
- the connection map is established
- you refine traffic management outside of these steps

The SPVC set up in this example uses a customer-defined address. For this reason, the *TerminateSpvc* optional subcomponent is configured.

Procedure steps

Step	Action
1	Set up ATM routing for Node A, specifying the node prefix. The objective of the first series of configuration steps is to set up the node prefix and the ATM interfaces on Node A. See the figure Static routing in a small Multiservice Switch-only network (page 200). **add AtmRouting** **set ModuleData nodePrefix 47139100000000000000000000**
2	Set up an ATM interface on FP #1. Do not configure the *side* attribute since you want to use the default value of *network* for this interface. **add AtmIf/10** **add AtmIf/10 Uni**
3	Set up a VCC on the ATM interface, and configure the SPVC destination address to Node C. **add AtmIf/10 Vcc/0.100** **add AtmIf/10 Vcc/0.100 SourcePvc** **set AtmIf/10 Vcc/0.100 SourcePvc calledAddress 47139310000000000000000000000000000000000000** This address must be 40 digits long. The first part (47139310) is the same as terminating address on Node C, AtmIf/10 (see step 12 of this procedure). **set atmIf/10 Vcc/0.100 Src calledVpiVci 0.100** Ensure that VPI.VCI 0.100 is not in use on the destination interface on Node C.
4	Set up a VPC on the ATM interface, and configure the SPVP to Node C. **add AtmIf/10 Vpc/3** **add AtmIf/10 Vpc/3 SourcePvp** **set AtmIf/10 Vpc/3 SourcePvp calledAddress 47139310000000000000000000000000000000000000** This address must be 40 digits long. The first part (47139310) is the same as terminating address on Node C, AtmIf/10 (see step 12 of this procedure). **set atmIf/10 Vpc/3 Src calledVpi 3** Ensure that VPI 3 is not in use on the destination interface on Node C.
5	Set up an ATM interface on FP #2. This interface links to an ATM interface on Node C.

```
add AtmIf/21

add AtmIf/21 Iisp
```

Do not configure the *side* attribute, since you want to use the default value of *networkSide* for this interface.

```
add AtmIf/21 Iisp Address/471393,primary
```

This address definition permits routing to Node C, and the address is included in the primary list. The partial address that you specify here corresponds to the first part of the node address that you configure under the *ModuleData* component for Node C.

6 To enable the call to go through Node B to Node C, set up a second ATM interface on FP #2. This interface links to an interface on Node B.

```
add AtmIf/22

add AtmIf/22 Iisp
```

Do not configure the *side* attribute, since you want to use the default value of *networkSide* for this interface.

```
add AtmIf/22 Iisp Address/471392,primary
```

This address definition permits routing to Node B, and the address appears in the primary list. The partial address that you specify here corresponds to the first part of the node address that you configure under the *ModuleData* component for Node B.

7 Set up ATM routing for Node B. The objective of the second series of configuration steps is to set up the node prefix and the ATM interfaces on Node B. See the figure Static routing in a small Multiservice Switch-only network (page 200).

```
add AtmRouting

set ModuleData nodePrefix 47139200000000000000000000
```

8 Set up an ATM interface. This ATM interface links to AtmIf/22 on Node A.

```
add AtmIf/30

add AtmIf/30 Iisp

set AtmIf/30 Iisp side userSide

add AtmIf/30 Iisp Address/471391,primary
```

This address definition permits routing to Node A, and the address appears in the primary list. The partial address that you specify here corresponds to the first part of the node address that you configure under the *ModuleData* component for Node A.

9 Set up a seconded ATM interface on the FP. This interface links to an interface on Node C.

```
add AtmIf/31

add AtmIf/31 Iisp
```

```
set AtmIf/31 Iisp side userSide

add AtmIf/31 Iisp Address/471393,primary
```

10 Set up ATM routing for Node C, specifying the node address.

```
add AtmRouting

set ModuleData nodePrefix 47139300000000000000000000
```

11 Set up an ATM interface on the first FP.

```
add AtmIf/10

add AtmIf/10 Uni
```

Do not configure the *side* attribute since you want to use the default value of *network* for this interface.

12 Define the interface as the termination point for the SPVC from Node A.

```
add -s AtmIf/10 Uni Address/47139310,primary
terminateSpvc
```

The address is in the format *47nnnnii*, where *nnnn* is the node identifier and *ii* is the ATM interface instance number. The string `47` indicates that the address in ICD format.

13 Set up an ATM interface on FP #2. This ATM interface links to AtmIf/21 on Node A.

```
add AtmIf/41

add AtmIf/41 Iisp

set AtmIf/41 Iisp side user

add AtmIf/41 Iisp Address/471391,primary
```

This address definition permits routing to Node A, and the address appears in the primary list. The partial address that you specify here corresponds to the first part of the node address that you configure under the *ModuleData* component for Node A.

14 Set up an ATM interface on FP #2. This interface links to AtmIf/31 on Node B.

```
add AtmIf/42

add AtmIf/42 Iisp

add AtmIf/42 Iisp Address/471392,primary
```

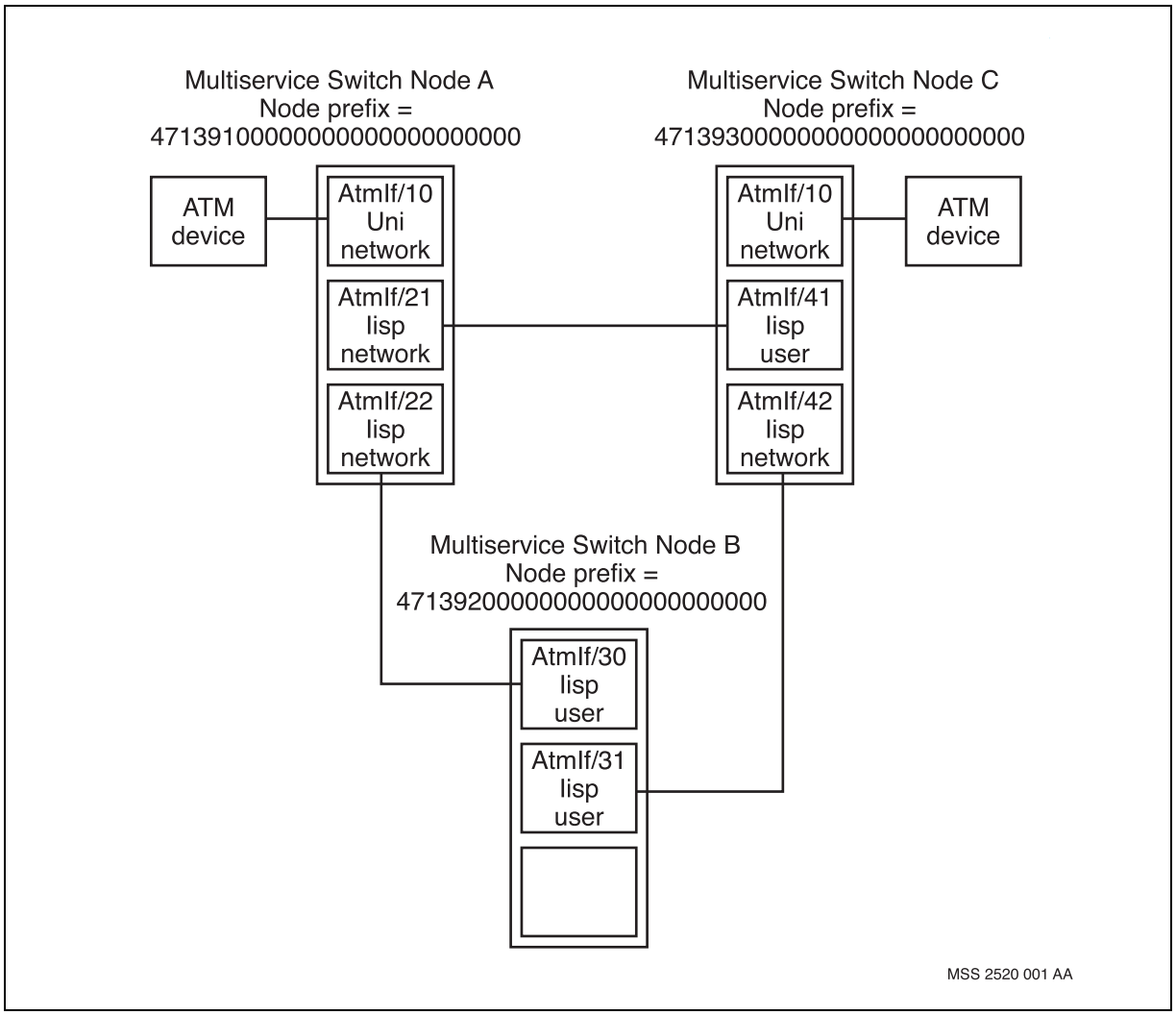
Do not configure the *side* attribute, since you want to use the default value of *networkSide* for this interface.

This address definition permits routing to Node B, and the address appears in the primary list. The partial address that you specify here corresponds to the first part of the node address that you configure under the *ModuleData* component for Node B.

--End--

Procedure job aid

Static routing in a small Multiservice Switch-only network



Example 5: Static routing using wild cards

This example shows how you can configure a wild card address specification to support static routing in a small Nortel Multiservice Switch-only network. The network configuration is shown in the figure Static routing using wild cards (page 204).

This configuration is similar to that illustrated in Example 3: Test and loop configuration using VCCs (page 195). In this example, however, you define a wild card address on *AtmIf/22* on Node A and on *AtmIf/31* on Node B.

Use wild card addresses with caution. In this example, if you configure several additional interfaces on Node B (not shown in the figure Static routing using wild cards (page 204)), all nodes accessible through those interfaces are accessible through *AtmIf/22* on Node A. Similarly, if you configure several additional interfaces on Node C, all nodes accessible through those interfaces are accessible through *AtmIf/31* on Node B.

You must ensure that you do not introduce infinite loops into the network. For example, if *AtmIf/21* on Node A links to *AtmIf/41* on Node C which you also configure with a wild card, call set up originating on any of Nodes A, B, or C can loop until the resources on the ATM interfaces are exhausted.

Procedure steps

Step	Action
1	Set up ATM routing for Node A, specifying the node prefix. The objective of the first series of configuration steps is to set up the node prefix and the ATM interfaces on Node A. See the figure Static routing using wild cards (page 204). **add AtmRouting** **set ModuleData nodePrefix 47139100000000000000000000end**
2	Set up an UNI on FP #1. Do not configure the *side* attribute since you want to use the default value of *network* for this interface. **add AtmIf/10** **add AtmIf/10 Uni**
3	Set up a VCC on the ATM interface, and configure the SPVC to Node C. **add AtmIf/10 Vcc/0.100** **add AtmIf/10 Vcc/0.100 SourcePvc** **set AtmIf/10 Vcc/0.100 SourcePvc calledAddress 47139310000000000000000000000000000000000**

This address must be 40 digits long. The first part (47139310) is the same as terminating address on Node C, AtmIf/10 (see step 11).

```
set AtmIf/10 Vcc/0.100 SourcePvc calledVpiVci 0.100
```

Ensure that VPI.VCI 0.100 is not in use on the destination interface on Node C.

4 Set up a VPC on the ATM interface, and configure the SPVP to Node C.

```
add AtmIf/10 Vpc/3

add AtmIf/10 Vpc/3 SourcePvp

set AtmIf/10 Vpc/3 SourcePvp calledAddress
47139310000000000000000000000000000000000
```

This address must be 40 digits long. The first part (47139310) is the same as terminating address on Node C, AtmIf/10 (see step 11).

```
set AtmIf/10 Vpc/3 SourcePvp calledVpi 3
```

Ensure that VPI 3 is not in use on the destination interface on Node C.

5 Set up an ATM interface on FP #2. This interface link to an interface on Node B, and you define the address as a wild card. Do not configure the *side* attribute, since you want to use the default value of *networkSide* for this interface.

```
add AtmIf/22

add AtmIf/22 Iisp

add AtmIf/22 Iisp Address/?,primary
```

6 Set up ATM routing for Node B. The objective of the second series of configuration steps is to set up the node prefix and the ATM interfaces on Node B (see the figure Static routing using wild cards (page 204)).

```
add AtmRouting

set ModuleData nodePrefix 47139200000000000000000000
```

7 Set up an ATM interface. This ATM interface links to AtmIf/22 on Node A.

```
add AtmIf/30

add AtmIf/30 Iisp

set AtmIf/30 Iisp side userSide

add AtmIf/30 Iisp Address/471391,primary
```

This address definition permits routing to Node A, and the address is included in the primary list. The partial address that you specify here corresponds to the first part of the node address that you configure under the *ModuleData* component for Node A.

8 Set up a second ATM interface on the FP. This interface links to an interface on Node C.

```
add AtmIf/31
```

```
add AtmIf/31 Iisp

set AtmIf/31 Iisp side userSide

add AtmIf/31 Iisp Address/?,primary
```

9 Set up ATM routing for Node C, specifying the node address. The objective of the third and last series of configuration steps is to set up the node prefix and the ATM interfaces on Node C. See the figure Static routing using wild cards (page 204).

```
add AtmRouting

set ModuleData nodePrefix 47139300000000000000000000
```

10 Set up an ATM interface on the first FP. Do not configure the *side* attribute since you want to use the default value of *network* for this interface.

```
add AtmIf/10

add AtmIf/10 Uni
```

11 Define the interface as the termination point for the SPVC from Node A.

```
add AtmIf/10 Uni Address 47139310,primary

add AtmIf/10 Uni Address 47139310,primary terminateSpvc
```

The address is in the format *47nnnnii*, where *nnnn* is the node identifier and *ii* is the ATM interface instance number. The string `47` indicates that the address in ICD format.

12 Set up an ATM interface on FP #2. This interface links to AtmIf/21 on Node B. Do not configure the *side* attribute, since you want to use the default value of *networkSide* for this interface.

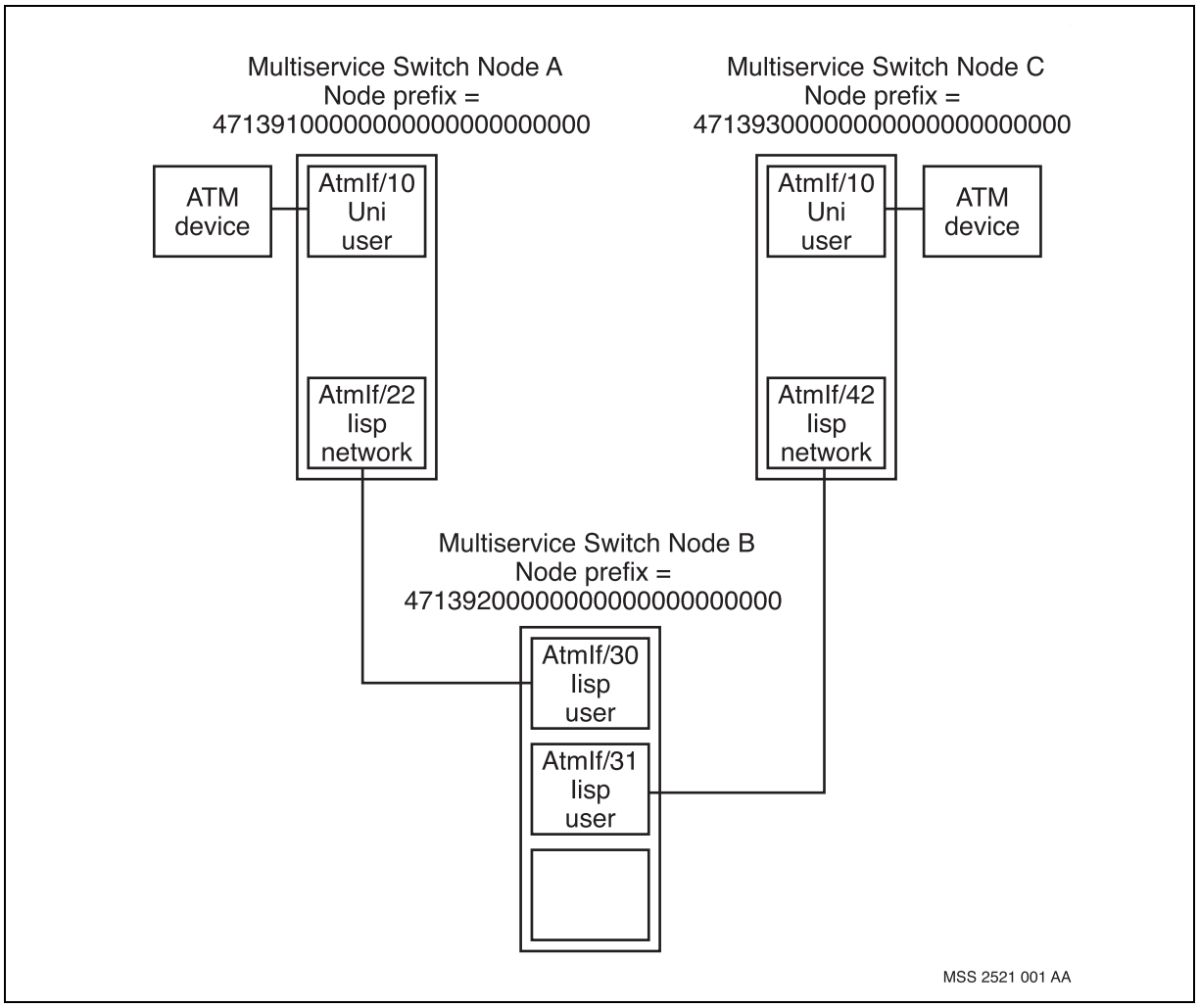
```
add AtmIf/42

add AtmIf/42 Iisp

add AtmIf/42 Iisp address/471392,primary
```

This address definition permits routing to Node B, and the address is included in the primary list. The partial address that you specify here corresponds to the first part of the node address that you configure under the *ModuleData* component for Node B.

--End--

Procedure job aid

Static routing using wild cards



Example 6: Small Multiservice Switch-only network under PNNI

This example shows how to configure a small Nortel Multiservice Switch-only network using PNNI. The figure Small Multiservice Switch-only network under PNNI (page 208) shows the network configuration.

This example shows how to provide a double-hop SPVC over PNNI.

The following procedure focuses on setting up a dynamic SPVC networking under PNNI. Note the following assumptions:

- you have added the software feature.
- ATM resource control is configured
- connection map is established
- you refine traffic management outside of these steps

The SPVC set up in this example uses a customer-defined address. For this reason, the *TerminateSpvc* optional subcomponent is configured.

Procedure steps

Step	Action
1	Set up ATM routing and PNNI for Node A. The objective of the first series of configuration steps is to set up the node prefix and the ATM interfaces on Node A. See the figure Small Multiservice Switch-only network under PNNI (page 208). **`add AtmRouting`** **`add AtmRouting Pnni`**
2	Set the node prefix for Node A. **`set ModuleData nodePrefix 4713911111111111111111111A0`** If not explicitly configured, the node derives the peer group ID for Node A from the value in the *nodePrefix* attribute. For this reason, the first <level> bits must be the same on all three modules in this example, since all modules belong to the same peer group. This example network uses level 96 (that is, 12 octets) for the peer group ID, which is the default value that the ATM Forum defines.
3	Set the routing domain name. **`set AtmRouting Pnni domain MyDomain`** All nodes in the same routing domain must have the same domain name.
4	Set the node to participate at the chosen level in the PNNI hierarchy. **`add AtmRouting Pnni ConfiguredNode/96`**

5 Set up an UNI on FP #1. Do not configure the *side* attribute since you want to use the default value of *network* for this interface.

```
add AtmIf/10

add AtmIf/10 Uni
```

6 Set up a VCC on the ATM interface, and configure the SPVC to Node C.

```
add AtmIf/10 Vcc/0.100

add AtmIf/10 Vcc/0.100 SourcePvc

set AtmIf/10 Vcc/0.100 SourcePvc calledAddress
47139111111111111111111C001000000000000
```

This address must be 40 digits long.

```
set AtmIf/10 Vcc/0.100 SourcePvc calledVpiVci 0.100
```

Ensure that VPI.VCI 0.100 is not in use on the destination interface on Node C.

7 Set up a PNNI on FP #2.

```
add AtmIf/22

add AtmIf/22 Pnni
```

8 Set up ATM routing and PNNI for Node B. The objective of the second series of configuration steps is to set up the node prefix and the ATM interfaces on Node B. See the figure Small Multiservice Switch-only network under PNNI (page 208).

```
add AtmRouting

add AtmRouting Pnni
```

9 Set the node prefix for Node B.

```
set ModuleData nodePrefix 47139111111111111111111B0
```

10 Set the routing domain name.

```
set AtmRouting Pnni domain MyDomain
```

All nodes in the same routing domain must have the same domain name.

11 Set the node to participate at the chosen level in the PNNI hierarchy.

```
add AtmRouting Pnni ConfiguredNode/96
```

12 Set up a PNNI on FPs #1 and #2.

```
add AtmIf/30

add AtmIf/30 Pnni

add AtmIf/31

add AtmIf/31 Pnni
```

13 Set up ATM routing and PNNI for Node C. The objective of the third and last series of configuration steps is to set up the node prefix and the ATM

interfaces on Node C. See the figure Small Multiservice Switch-only network under PNNI (page 208).

```
add AtmRouting
```

```
add AtmRouting Pnni
```

14 Set the node prefix for Node C.

```
set ModuleData nodePrefix 47139111111111111111111C0
```

15 Set the routing domain name.

```
set AtmRouting Pnni domain MyDomain
```

All nodes in the same routing domain must have the same domain name.

16 Set the node to participate at the chosen level in the PNNI hierarchy.

```
add AtmRouting Pnni ConfiguredNode/96
```

17 Set up an UNI on FP #1.

```
add AtmIf/10
```

```
add AtmIf/10 Uni
```

18 Define the interface as the termination point for the SPVC from Node A.

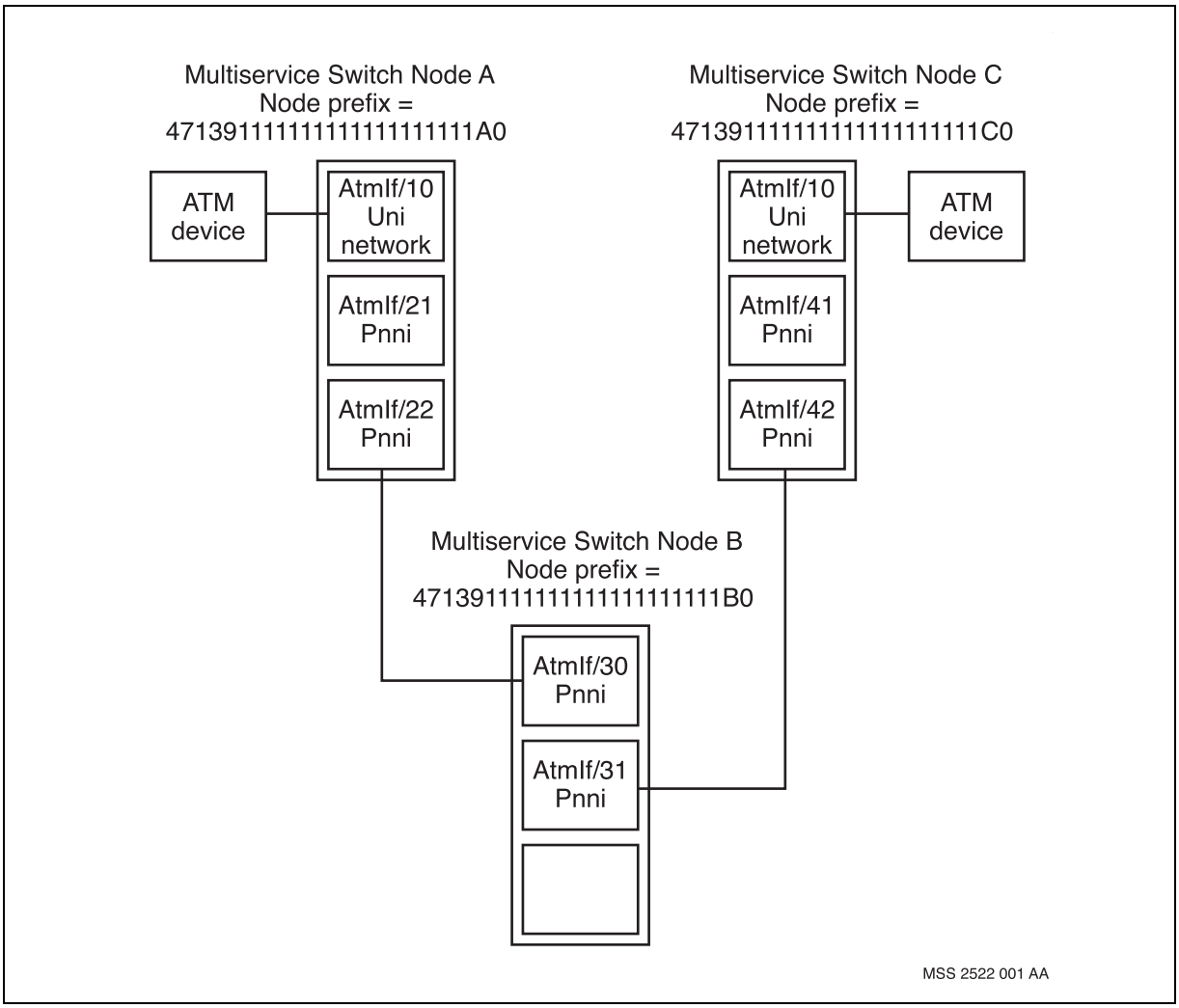
```
add AtmIf/10 Uni Address/
47139111111111111111111C0010000000000,primary
terminateSpvc
```

```
add AtmIf/42 Pnni
```

--End--

Procedure job aid

Small Multiservice Switch-only network under PNNI



Resource adjustments for different function processors

This section describes the relationship between two sets of attributes. One set deals with function processor resources. They are:

- the *connectionPoolCapacity* attribute of the *Lp Eng Arc Ov* component
- the *protectedConnectionPoolCapacity* attribute of the *Lp Eng Arc Ov* component
- the *connectionPoolCapacity* attribute of the *Lp Eng Arc Aqm Ov* component

The second set of attributes deals with the ATM interfaces that use function processor resources. They are:

- the *maxVpcs* attribute of the *AtmIf Ca* component
- the *maxVpts* attribute of the *AtmIf Ca* component
- the *maxVccs* attribute of the *AtmIf Ca* component

These attributes initially have a default setting. However, these default settings are not satisfactory if the FP supports more than four configurable ATM port, for example: the 12-port DS3 FP. Furthermore, on all FPs, the default settings are probably not satisfactory if the connection usage is unequally distributed between ports. To do the proper adjustments, you must first calculate how many additional resources are required for the additional ATM interfaces.

Navigation

- Calculating the value of the connectionPoolCapacity attribute (page 210)
- Calculating the value of the protectedConnectionPoolCapacity attribute (page 211)
- Calculating the value of the connectionPoolCapacity attribute for AQM based ATM FPs (page 212)
- Considerations for the 12-port DS3 ATM FP and the 12-port E3 ATM FP for Multiservice Switch 15000 and Multiservice Switch 20000 nodes (page 212)

- Considerations for the 4-port OC-3 ATM FP (page 214)
- Considerations for the 2-port OC-3 ATM and 2-port STM-1 electrical ATM FPs (page 215)
- Considerations for the 2-port OC-3 ATM and 2-port STM-1 electrical ATM FPs (page 215)
- Considerations for the 2-port OC-3 ATM and 2-port STM-1 electrical ATM FPs (page 215)
- Considerations for the 3-port OC-3 ATM FP (page 217)
- Considerations for the DS1 MSA32 FP and the E1 MSA32 FP (page 217)
- Considerations for the DS1 MSA8 FP and the E1 MSA8 FP (page 218)

Calculating the value of the connectionPoolCapacity attribute

This section deals with the *connectionPoolCapacity* attribute of the *Lp Eng Arc Ov* component.

The value of *connectionPoolCapacity* sets the resources reserved by the FP for the VPTs, VPCs and VCCs connections for all unspared ATM interfaces. Resources for spared ATM interfaces are set in Calculating the value of the protectedConnectionPoolCapacity attribute (page 211).

The value of *connectionPoolCapacity* is calculated from values for the *maxVpcs*, *maxVpts* and *maxVccs* attributes for each *AtmIf CA* component.

Use the following formula to calculate the number of resources for each unspared ATM interface:

```
<num_unspared_cons>= (maxVpcs+maxVccs+[maxVpts x 3])+1
```

The value of connectionPoolCapacity is the sum of all <num_unspared_cons> for a particular FP.

The sum of the *Arc Ov protectedConnectionPoolCapacity* attribute and the *Arc Ov connectionPoolCapacity* attribute should not exceed the practical limit of the number of connection resources for the FP. This limit varies depending on the configuration. Please contact your Nortel Networks' representative to determine the connection resource limit for the FP.

Special considerations are described in:

- Considerations for the 12-port DS3 ATM FP and the 12-port E3 ATM FP for Multiservice Switch 15000 and Multiservice Switch 20000 nodes (page 212)
- Considerations for the 4-port OC-3 ATM FP (page 214)
- Considerations for the 2-port OC-3 ATM and 2-port STM-1 electrical ATM FPs (page 215)

- Considerations for the 3-port OC-3 ATM FP (page 217)
- Considerations for the DS1 MSA32 FP and the E1 MSA32 FP (page 217)
- Considerations for the DS1 MSA8 FP and the E1 MSA8 FP (page 218)

If necessary, adjust the value of the *protectedConnectionPoolCapacity* attribute, or the *maxVpcs*, *maxVpts* and *maxVccs* attributes as required.

To adjust the value of the *protectedConnectionPoolCapacity* attribute see Configuring connection and buffer space for ATM IP FPs (page 22).

To adjust the value of the *maxVpcs*, *maxVpts* and *maxVccs* attributes, see Configuring connection administrator (page 47).

Calculating the value of the protectedConnectionPoolCapacity attribute

This section deals with the *protectedConnectionPoolCapacity* attribute of the *Lp Eng Arc Ov* component.

The value of *protectedConnectionPoolCapacity* sets the resources reserved by the FP for VPTs, VPCs and VCCs for all spared ATM interfaces. Resources for unspared ATM interfaces are set in Calculating the value of the connectionPoolCapacity attribute (page 210).

The value of *protectedConnectionPoolCapacity* is calculated from values for the *maxVpcs*, *maxVpts* and *maxVccs* attributes for each *AtmIf CA* component.

Use the following formula to calculate the number of protected resources for each spared ATM interface:

```
<num_pro_cons>= (maxVpcs+maxVccs+[maxVpts x 3])+1
```

The value of *protectedConnectionPoolCapacity* is the sum of all <num_pro_cons> for a particular FP.

The sum of the *Arc Ov protectedConnectionPoolCapacity* attribute and the *Arc Ov connectionPoolCapacity* attribute cannot exceed the practical limit of the number of connection resources for the FP. The practical limit for most FPs is described in Configuring connection and buffer space for ATM IP FPs (page 22).

Special considerations are described in:

- Considerations for the 12-port DS3 ATM FP and the 12-port E3 ATM FP for Multiservice Switch 15000 and Multiservice Switch 20000 nodes (page 212)
- Considerations for the 4-port OC-3 ATM FP (page 214)

- Considerations for the 2-port OC-3 ATM and 2-port STM-1 electrical ATM FPs (page 215)
- Considerations for the 3-port OC-3 ATM FP (page 217)
- Considerations for the DS1 MSA32 FP and the E1 MSA32 FP (page 217)
- Considerations for the DS1 MSA8 FP and the E1 MSA8 FP (page 218)

If necessary, adjust the value of the *connectionPoolCapacity* attribute, or the *maxVpcs*, *maxVpts* and *maxVccs* attributes as required.

To adjust the value of the *connectionPoolCapacity* attribute see Configuring connection and buffer space for ATM IP FPs (page 22).

To adjust the value of the *maxVpcs*, *maxVpts* and *maxVccs* attributes, see Configuring connection administrator (page 47).

Calculating the value of the connectionPoolCapacity attribute for AQM based ATM FPs

By default, the value of the connectionPoolCapacity is divided equally among all AQMs.

Unless the distribution of the connections is unequal between ports, the default value of the *Arc Aqm Ov connectionPoolCapacity* attribute is generally sufficient for the 4-port OC-3 FPs. However, this is generally not the case for the 12-port E3 ATM FP and the 12-port DS1 ATM FP.

See Considerations for the 12-port DS3 ATM FP and the 12-port E3 ATM FP for Multiservice Switch 15000 and Multiservice Switch 20000 nodes (page 212).

Considerations for the 12-port DS3 ATM FP and the 12-port E3 ATM FP for Multiservice Switch 15000 and Multiservice Switch 20000 nodes

For the 12-port DS3 ATM FP and the 12-port E3 ATM FP, there are four instances of the *Aqm* component. *Aqm/0* represents ports 0, 1 and 2. *Aqm/1* represents ports 3, 4 an 5. *Aqm/2* represents ports 6, 7 and 8. *Aqm/3* represents ports 9, 10 and 11.

The default value of the *Aqm Ov connectionPoolCapacity* attribute is derivedFromArc, where the resources set by the *Arc Ov connectionPoolCapacity* attribute are divided equally among the instances of the *Aqm* component.

If you configure more than one port on a single AQM, you must adjust the value of the *Aqm Ov connectionPoolCapacity* attribute for that AQM to provide sufficient resources for the additional ports.

Use the following formula to calculate the number of resources required for each additional port:

<num_aqm_cons>= (maxVpcs+maxVccs+[maxVpts x 3])+1

The value of the *Aqm Ov connectionPoolCapacity* attribute for each AQM is the sum of <num_aqm_cons> for each port using that AQM.

The value of all *Aqm Ov connectionPoolCapacity* attributes cannot exceed the sum of the *Arc Ov protectedConnectionPoolCapacity* attribute and the *Arc Ov connectionPoolCapacity* attribute.

Alternatively, you can:

- adjust the value of the *maxVpcs*, *maxVpts* and *maxVccs* attributes for the ATM interfaces running on the ports using that AQM. See Configuring connection administrator (page 47).
- configure the ATM interface to run on another port using a different AQM

For example, to calculate the *connectionPoolCapacity* of a new *AtmIf* on port/3 under Aqm/1 using default values where one AtmIf is configured on ports 0, 1, and 2:

- maxVccs = 255
- maxVpcs = 128
- maxVpts = 128

CPC(Aqm/0) ≥ (3 x 255) + (3 x 128) + (3 x 128 x3) + 1 x 3
≥ 765 + 384 +1152 +3
≥ 2304
CPC = 768 x 4
CPC = 3072

The minimum total value of the *connectionPoolCapacity* attribute under

- *lp/x Eng Arc Ov* is 3072
- *lp/x Eng Arc Aqm/0 Ov*
- *lp/x Eng Arc Aqm/1 Ov* is 768

For example, to calculate the *connectionPoolCapacity* where a larger value is assigned to the *connectionPoolCapacity* and the connections are distributed unequally among the following three ports:

For port/0:

- maxVccs = 511
- maxVpcs = 128
- maxVpts = 128

For port/1:

- maxVccs = 700
- maxVpcs = 15
- maxVpts = 5

For port/2:

- maxVccs = 500
- maxVpcs = 323
- maxVpts = 100

Use the following formula to calculate the number of resources required for each additional port:

```
<num_aqm_cons>= (maxVpcs(all ports)+ maxVccs(all
ports)+[maxVpts(all ports)X 3])+1 X(number of ports)
```

CPC(Aqm/0) ≥ (511 + 700 + 500) + (128+15 +323) + ((128 +5 +100) x 3) +1 x 3

≥ 1711 + 466 + 699 +3

CPC ≥ 2879

The minimum total value of the *connectionPoolCapacity* is 3072. The value of *connectionPoolCapacity* attribute under

- *lp/x Eng Arc Ov* connectionPoolCapcity is 2879
- *connectionPoolCapcity is 2879lp/x Eng Arc Aqm/0 Ov*

Considerations for the 4-port OC-3 ATM FP

For the 4-port OC-3 ATM FP, there are four instances of the *Aqm* component. *Aqm/0* represents port 0. *Aqm/1* represents port 1. *Aqm/2* represents port 2. *Aqm/3* represents port 3.

Use the following formula to calculate the number of resources required for each additional port:

```
<num_aqm_cons>= (maxVpcs+maxVccs+[maxVpts x 3])+1
```

For example, to calculate the *connectionPoolCapacity* where one AtmIf is configured on each port where:

- maxVccs = 255

- maxVpcs = 128
- maxVpts = 128

CPC(Aqm/0) ≥ 255 + 128 + (3 x 128) + 1
≥ 768
CPC ≥ 768 x 4
CPC ≥ 3072

The minimum total value of the *connectionPoolCapacity* is 3072.

For example, to calculate the *connectionPoolCapacity* where one AtmIf is configured on each port where:

- maxVccs = 511
- maxVpcs = 128
- maxVpts = 128

CPC(Aqm/0) ≥ 511 + 128 + (3 x128) + 1
≥ 1024
CPC ≥ 1024 + 768 + 768 + 768 (768 is the default value of connections under each port)
CPC ≥ 1024 + 2304
≥ 3328

The minimum total value of the *connectionPoolCapacity* is 3328. The value of *connectionPoolCapacity* attribute under *lp/x Eng Arc Ov* can be set to 3328 and the value of *connectionPoolCapacity* attribute under *lp/x Eng Arc Aqm/0 Ov* can be set to 1024.

Considerations for the 2-port OC-3 ATM and 2-port STM-1 electrical ATM FPs

For the 2-port OC-3 ATM FP, the value of the connectionPoolCapacity is divided between 2 ports.

For the 2-port OC-3 ATM FP and the 2-port STM-1e ATM FP, the value of the connectionPoolCapacity is divided between 2 ports.

Use the following formula to calculate the number of resources required for each additional port:

```
<num_aqm_cons>= (maxVpcs(all ports)+ maxVccs(all
ports)+[maxVpts(all ports)X 3])+1 X(number of ports)
```

CPC= Connection capacity for port 0 +Connection capacity for port 1

By using the default values of maxVccs, maxVpcs, and maxVpts,

CPC(Aqm/0) ≥ [255 + 128) + (3 x128) + 1] x 2
≥ 768 + 768
≥ 1536

If you change the value of the connection capacity to a value lower than 1536, then you have to change the values of the maxVccs, maxVpcs, and maxVpts for both ports. The value of the connectionPoolCapacity can be lowered to 12.

Considerations for the 2-port STM-1 electrical channelized CES/ATM/ IMA FPs

When configuring the *Lp Eng Arc Ov* and *Lp Eng Arc Aqm/n Ov* components, the user needs to follow the following guidelines to ensure proper resource reservation.

- For *Arc Ov*, *connectionPoolCapacity* + *protectedConnectionPoolCapacity* should be less than, or equal to16000.
- For *Arc Aqm*/n *Ov* (where n = 0-3), *connectionPoolCapacity* should be less than, or equal to 5000.

In addition, each CES connection configured on the 2pSTM1eCh FP requires 1 ATM connection resource internally.

- *connectionPoolCapacity* >= maximum desired unprotected ATM connections + maximum desired protected CES connections.
- *protectedConnectionPoolCapacity* >= maximum desired protected ATM connections + maximum desired protected CES connections
- *Lp Eng Arc Aqm*/1 (*Aqm*/3) Ov *connectionCapacity* >= maximum desired CES connections on Sdh/0 (Sdh/1) + maximum desired ATM connections on this AQM.

Considerations for the 2-port STM-1 optical channelized CES/ATM/ IMA FPs

Follow these guidelines to ensure proper resource reservation when configuring the *Lp Eng Arc Ov* and *Lp Eng Arc Aqm/n Ov* components.

- For *Lp Eng Arc Ov*, the connectionPoolCapacity + protectedConnectionPoolCapacity should be less than, or equal to16000.
- For *Lp Arc Aqm/n Ov* (where n = 0-3), the connectionPoolCapacity should be less than 6000.

Since each CES connection requires one ATM connection, the following resource considerations are required.

- connectionPoolCapacity should be equal to or greater than the maximum unprotected ATM connections + the maximum unprotected CES connections.
- protectedConnectionPoolCapacity should be equal to or greater than the maximum protected ATM connections + the maximum protected CES connections.
- For *Lp Eng Arc Aqm/1 Ov* the connectionCapacity should be equal to or greater than the maximum CES connections on Sdh/0 + maximum ATM connection on this Aqm/1. There are no CES connections on Aqm/0.
- For *Lp Eng Arc Aqm/3 Ov* the connectionCapacity should be equal to or greater than the maximum CES connections on Sdh/1 + maximum ATM connection on this Aqm/3. Note that there are no CES connections on Aqm/2.

Considerations for the 3-port OC-3 ATM FP

For the 3-port OC-3 ATM FP, the value of the connectionPoolCapacity is divided among 3 ports.

Use the following formula to calculate the number of resources required for each additional port:

```
<num_aqm_cons>= (maxVpcs(all ports)+ maxVccs(all
ports)+[maxVpts(all ports)X 3])+1 X(number of ports)
```

CPC= Connection capacity for port 0 +Connection capacity for port 1 +Connection capacity for port 2

By using the default values of maxVccs, maxVpcs, and maxVpts,

CPC(Aqm/0) ≥ 255 + 128 + 384 + 1
≥ 768
≥ 2304 +768
≥ 3072

If you decide to use connection capacity values greater than 3072, then it has to be in multiples of 768.

Considerations for the DS1 MSA32 FP and the E1 MSA32 FP

If more than 24 ATM interfaces are running on a DS1 MSA32 FP or an E1 MSA32 FP, then you must either:

- reduce the values of *maxVpcs*, *maxVccs* and *maxVpts* so the maximum number of connections for all ATM interfaces on the MSA32 FPs' electrical ports is equal to or less than 4 096.

or

- configure the *Arc Override* and *Arc Aqm/0 Override* components and increase the connection capacity.

The sum of *connectionPoolCapacity*, *multiCastBranchesCapacity*, *protectedConnectionPoolCapacity* and *protectedMcastBranchesCapacity* must be less than or equal to 8 000 for MSA32 FPs. For MSA32 FPs with integrated optical trunks the sum of *connectionPoolCapacity*, *multiCastBranchesCapacity*, *protectedConnectionPoolCapacity* and *protectedMcastBranchesCapacity* must be less than or equal to 8 000 connections for aqm/0 and less than or equal to 16 000 connections for aqm/1.

Considerations for the DS1 MSA8 FP and the E1 MSA8 FP

If more than 24 ATM interfaces are running on a DS1 MSA8 FP or an E1 MSA8 FP, then you must either:

- reduce the values of *maxVpcs*, *maxVccs* and *maxVpts* so the maximum number of connections for all ATM interfaces on the MSA8 FPs' electrical ports is equal to or less than 4 096.

or

- configure the *Arc Override* and *Arc Aqm/0 Override* components and increase the connection capacity.

The sum of *connectionPoolCapacity*, *multiCastBranchesCapacity*, *protectedConnectionPoolCapacity* and *protectedMcastBranchesCapacity* must be less than or equal to 8 000 for MSA8 FPs.

ATM configuration overview

Use the following sections to learn more about configuration specifications.

Navigation

Specified path connection attributes

This section provides information about additional attributes that can impact the configuration of specified path connections. See the following sections for more information:

RemoteAtmIf considerations

Network management applications can use the configuration information stored in this attribute to discover the network's ATM topology. The value for this attribute may include the component identifier, address, or other numerical identifiers. The syntax of the value must follow the convention of your network management platform. If there is no such platform or associated convention, leave the value empty (the default value). The value is only stored by the module. It is not validated and is not used to perform any validation for remote-connectivity.

OamSegmentBoundary considerations

Set the value to yes to define the interface as an OAM segment boundary. Some connections passing through the interface terminate OAM segment loopback cells, and others pass through transparently, as indicated by the *connectionType* operational attribute for the associated connections. The *connectionType* attribute is set as follows:

- for *AtmIf Vcc/Vpc Rp* components, the *connectionType* attribute is set to *segmentEndPoint*
- for *AtmIf Vcc/Vpc Nrp* components, it depends on the value of the connection component's *oamSegmentBoundary* attribute: if the *Nrp oamSegmentBoundary* is yes, or sameAsInterface, the *connectionType* attribute is set to *segmentEndPoint*; if the *Nrp oamSegmentBoundary* is no, the *connectionType* attribute is set to *connectingPoint*
- for *AtmIf Vcc/Vpc Nep* or *Ep* components, the *connectionType* attribute is set to *connectionEndPoint*

Set the value to no to indicate that the interface is not an OAM segment boundary. Some connections passing through the interface terminate OAM segment loopback cells, and others pass through transparently, as indicated by the *connectionType* operational attribute for the associated connections. The *connectionType* attribute is set as follows:

- for *AtmIf Vcc/Vpc Rp* components, the *connectionType* attribute is set to *connectingPoint*
- for *AtmIf Vcc/Vpc Nrp* components, it depends on the value of the connection component's *oamSegmentBoundary* attribute: if the *Nrp oamSegmentBoundary* is no, or sameAsInterface, the *connectionType* attribute is set to *connectingPoint*; if the *Nrp oamSegmentBoundary* is yes, the *connectionType* attribute is set to *segmentEndPoint*
- for *AtmIf Vcc/Vpc Nep* or *Ep* components, the *connectionType* attribute is set to *connectionEndPoint*

Cell transfer delay calculation considerations

Setting the attribute to on causes the CTD to be calculated when a segment device side loopback is enabled on an SPVC or SPVP connection. You can then display CTD statistics as operational attributes of the Vcc and Vpc components of every associated source SPVC or SPVP on the interface. Setting this attribute also causes CTD measurements to be collected in the ATM accounting record. The accounting record for source SPVC and SPVP connections includes values for the maximum, minimum, and average CTD calculations, and the number of CTD measurements taken during the last accounting interval. (The values are collected only at the source of the SPVC or SPVP connection. They are not visible at the destination end of the connection.)

Setting the attribute to off disables CTD calculation.

Only connections configured as segment end points generate CTD measurements. To configure segment end points, set the interface *oamSegmentBoundary*.

You must also enable accounting before the CTD statistics can be maintained. If you do not enable accounting, the interface displays the current CTD, but none of the other attribute values.

Attention: CTD is only supported on SPVCs and SPVPs. Since CTD cells are processed in software, there is a delay in reading and writing the time stamps. This delay causes the CTD value on a Nortel Multiservice Switch node to be slightly greater than the same CTD measured using external test equipment, which does the stamping in hardware.

ArtgPnniMdtlPath considerations

This path name needs to exist under the *ARtgPnniMdtlPath* component. Otherwise, the check prov command will fail.

Attributes for traffic management on AQM and APC-based FPs

The following attributes are applicable to AQM-based and APC-based FPs only:

- *emissionPriority* described in the procedure Configuring traffic management for service categories (page 57)
- *shapeRecoupPolicy* described in the procedure Configuring traffic management for service categories (page 57)
- *weightPolicy* described in the procedure Configuring traffic management for service categories (page 57)

The value of the *emissionPriority* attribute in each service category must comply with all of the following restrictions:

- The EP in each service category must be in the following order: CBR <= rt-VBR <= nrt-VBR <= UBR.
- Two different service categories can have the same EP value only if they are both shaped.
- For non-APC-based FPs, the CBR and rt-VBR service categories can have EP values of 0 to 7. In the case of APC-based FPs, CBR and rt-VBR service categories can have EP values of 0, 2, 3, 4, and 7.
- If the rt-VBR service category is shaped, it must share the same EP as the CBR service categories.

- The ABR service category is always shaped. If the nrt-VBR service category is also shaped, it must have the same EP value as the ABR service category. If the nrt-VBR service category is not shaped, it must have a higher EP value than the ABR service category.
- If the UBR service category is shaped, it must have the same EP as the ABR service category. If the UBR service category is not shaped, it must have a lower EP than the ABR service category.
- For non-APC-based FPs, if the value in the *unshapedTransmitQueueing* attribute is common, then each service category has specific EPs. For the CBR service category, the EP is 0 or 2. For the rt-VBR service category, the EP is 1 or 3. For nrt-VBR and UBR service categories, the EP is 4, 5, 6,or 7.

On AQM FPs, if the value in the *unshapedTransmitQueueing* attribute is common, you cannot modify either of the following attributes:

- the *txQueueLimit* attribute in the service categories
- the *weight* attribute at the connection level

Use the table Interpretation of the value autoConfigure for the unshapedTransmitQueueing attribute (page 222) when configuring the *unshapedTransmitQueueing* attribute in the procedure Configuring traffic management for service categories (page 57).

Interpretation of the value autoConfigure for the unshapedTransmitQueueing attribute

Type of FP	Condition	Interpretation of value autoConfigure
AQM-based	traffic shaping is disabled in service category	perVc
AQM-based	traffic shaping enabled or inverseUpc in service category	perVc
APC-based	traffic shaping is disabled in service category or traffic shaping enabled using the linear shaper	perVc
1-port OC12 FPs, 1-port OC48c FPs	traffic shaping is disabled	common
CQC-based	traffic shaping not enabled for interface	common
CQC-based	traffic shaping enabled for interface	perVc

Supporting information for configuring the connection map for CQC-based FPs

Use the following sections to learn more about configuring the connection map for CQC-based FPs.

- Relationship between the resource controls and the interface for CQC-based FPs (page 223)
- Connection map templates for CQC-based FPs (page 223)

Relationship between the resource controls and the interface for CQC-based FPs

The resource control components control the following ATM resources for CQC-based FPs (excluding the eight-port DS1/E1 FPs):

- connection space—the amount of memory required to support the number of ATM connections
- buffer space—used for cell queue memory. Buffer space is the remaining memory not required for connection space.
- traffic management for the FP and ports

After the number of connections is reserved, the connection map defines how the connection pool is partitioned across the VCI and VPI address space. The connection map defines which VPI spaces are reserved for VCCs (the remaining VPI are available for the VPCs). The connection map dimensions cannot exceed the number of connection points reserved through the resource control components.

The resource control components reserve memory and connection space. The *ConnMap* component configures the connection space. The values for the *AtmIf* attributes *maxVpcs, maxVpts*, and *maxVccs* reserve FP memory (not cell memory) and limits the number of connections. The connection map limits the values of these attributes.

Connection map templates for CQC-based FPs

The tables in this section represent standard configurations which cover the most common network requirements involving CQC-based FPs (excluding the eight-port DS1/E1 FPs). For additional information on connection map options, see Nortel Multiservice Switch Release Notes.

Connection mapping sample configuration values for CQC-based FPs (excluding eight-port DS1/E1 FPs) (page 224) provides sample configuration values for configuring the available VCC space through the connection map. Connection mapping configurations and values (excluding eight-port DS1/E1

FPs) (page 225) presents the application of the sample configuration values. Use the following considerations when applying the information from the tables:

- All table entries represent the actual values used to configure the address space.
- Configuration 1 is used when VPI/0 is used as a VPC. In this case, no VCCs can be supported under this ATM interface.
- The value provided for *totalCap* in each configuration is for the entire LP; the value for *connCap* in each configuration is for the individual port being configured.
- The number of VCCs always takes into consideration the reserved VPI.VCI=0.0-31.
- The first VPI in the VCC space under non-zero VPIs is represented by *x* and can be any acceptable value for *firstVpi*, where 16 <= x <= 240 and is a multiple of 16.
- Configuration 12 is a special case for the maximum number of VPIs allowed on port 2, since it has a lower maximum number of connections.

Connection mapping sample configuration values for CQC-based FPs (excluding eight-port DS1/E1 FPs)

Component	Configurations											
Attribute	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
ConnMap												
• nZVccs	0	768	256	256	256	256	3840	1792	1792	256	256	256
• nVpis	0	0	15	16	3	4	0	1	1	55	56	32
• firstVpi	1	1	1	*x*	1	*x*	1	16	1	1	16	16
• nVccs	0	64	64	64	256	256	64	2048	1024	64	64	64
Arc												
• totalCap	3072	3072	4608	4608	4608	4608	0	0	0	0	0	0
• connCap	0	0	0	0	0	0	4096	4096	4096	4096	4096	2560

Connection mapping configurations and values (excluding eight-port DS1/E1 FPs)

Config #	Application	Usable VCC range	Maximum VCCs	Maximum VPCs	Notes
1	All connections as VPCs	none	0	256	no VCCs under this interface
2	Large number of VCCs under VPI 0 (default configuration)	0.1 - 0.767	767	255	• the range of usable VCIs can be increased to 0.1279 by setting "numberOfVccsForVpiZero (nZVccs)" to 1280 • no non-zero VPIs
3	Limited number of VCCs under many non-zero VPIs (starting at VPI = 1)	0.1 - 0.255 1.0 - 1.63 through 15.0 - 15.63	1215	240	• 255 VCCs under VPI=0 • 64 VCCs under each non-zero VPI
4	Limited number of VCCs under many non-zero VPIs (starting at a VPI value other than 1)	0.1 - 0.255 x.0 - x.63 through (x+15).0 - (x+15).63	1279	239	• 255 VCCs under VPI=0 • 64 VCCs under each non-zero VPI • 1 < x < 256, in a multiple of 16
5	Larger number of VCCs under a few non-zero VPIs (starting at VPI = 1)	0.1 - 0.255 through 3.0 - 3.255	1023	252	• 255 VCCs under VPI=0 • 256 VCCs under each non-zero VPI
6	Larger number of VCCs under a few non-zero VPIs (starting at a VPI value other than 1)	0.1 - 0.255 x.0 - x.255 through (x+3).0 - (x+3).255	1279	251	• 255 VCCs under VPI=0 • 256 VCCs under each non-zero VPI • 1 < x < 256, in a multiple of 16
7	Larger number of VCCs under VPI=0 with the maximum possible number of VPCs	0.1 - 0.3839	3839	255	• 3839 VCCs under VPI=0 • no non-zero VPIs

(1 of 2)

Connection mapping configurations and values (excluding eight-port DS1/E1 FPs) (continued)

Config #	Application	Usable VCC range	Maximum VCCs	Maximum VPCs	Notes
8	Maximum number of VCCs under a non-zero VPI in the range 16 to 255	0.1 - 0.1791 16.0 - 16.2047	3839	254	• 1791 VCCs under VPI=0 • 2048 VCCs under the single non-zero VPI
9	Maximum number of VCCs under VPI=1	0.1 - 0.1791 1.0 - 1.1023	2815	254	• 1791 VCCs under VPI=0 • 1024 VCCs under the single non-zero VPI
10	Maximum number of VPIs for VCCs (starting at VPI=1)	0.1 - 0.255 1.0 - 1.63 through 55.0 - 55.63	3775	200	• 255 VCCs under VPI=0 • 64 VCCs under each non-zero VPI
11	Maximum number of VPIs for VCCs (starting at VPI=16)	0.1 - 0.255 16.0 - 16.63 through 72.0 - 72.63	3839	199	• 255 VCCs under VPI=0 • 64 VCCs under each non-zero VPI
12	Maximum number of VPIs on port 2 (starts at VPI=16)	0.1 - 0.255 16.0 - 16.63 through 48.0 - 48.63	2303	223	• 255 VCCs under VPI=0 • 64 VCCs under each non-zero VPI

(2 of 2)

Connection and buffer space configuration for permanent point-to-multipoint connections

The root connection points are taken from the standard connection pool associated with the *lp/x eng arc ov connectionPoolCapacity* parameter and should also be counted when setting the value of *AtmIf Ca maxVcc* or *AtmIf Ca maxVpc*. Each connection leaf is taken from a different pool, the size of which is defined by the *lp/x eng arc ov multicastBranchesCapacity* parameter.

To decide what value to assign to *lp/x eng arc ov multicastBranchesCapacity*, count the total number of connection leaves defined on the FP. The *lp/x eng arc ov multicastBranchesCapacity* parameter should be set to a value at least equal to that count. Note, that the connection pool for multicast leaf connection points is not split between AQMs. As a result, no adjustment is required at the AQM level.

The components and attributes under *AtmIf/x Ca* and *Lp/x Eng Arc Ov* are associated with multicast branches.

The calculations are summarized as follows:

- Each entity of a *Vcc/n.m Nrp*, *Vpc/n Nrp*, *Vcc/n.m Mnrp* or *Vpc/n Mnrp* component counts as one entity for *lp/x eng arc ov connectionPoolCapacity*.
- Each entry in the *nextHop* attribute of a *Vcc/n.m Mnrp* or *Vpc/n Mnrp* component counts as one entity for *lp/x eng arc ov multicastBranchesCapacity*.
- On each root connection, we count one for *maxVccs* or *maxVpcs* on the FP where the root physically exists.
- On each leaf connection, we count the number of leaves for *maxVccs* or *maxVpcs* on the FP where the leaves are physically located.

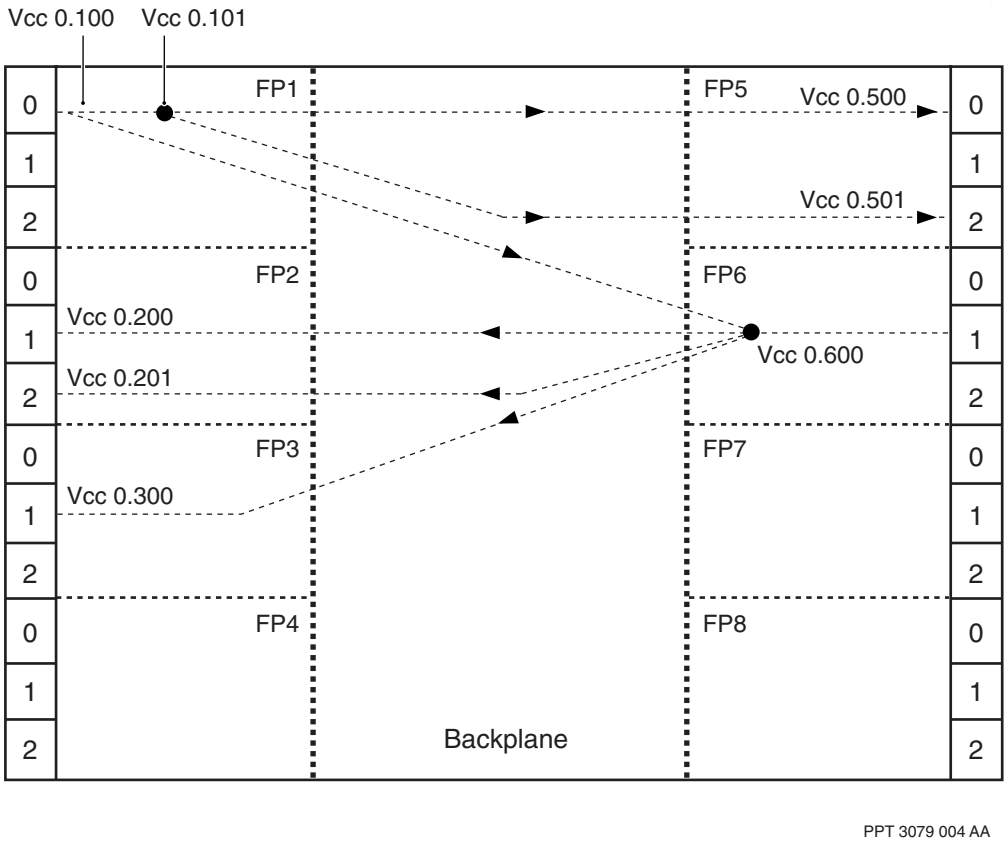
Note that the entry in the *nextHop* attribute of a *Vcc/n.m Nrp* or *Vpc/n Nrp* component is not counted in either pool.

See Example of a multicast branch calculation (page 227) for more information about multicast branch calculations.

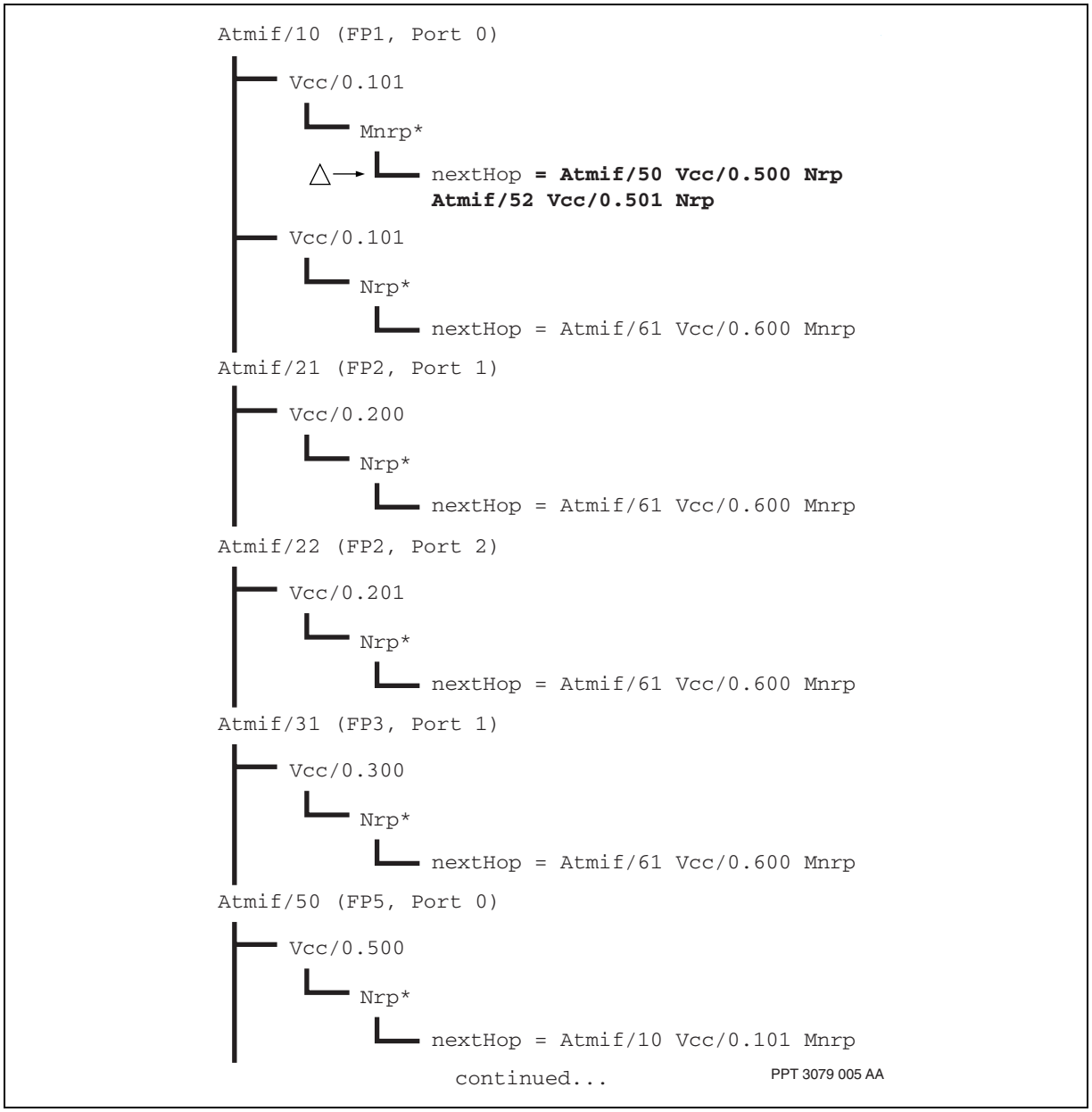
Example of a multicast branch calculation

Figure Point-to-multipoint connections (page 228), displays some point-to-multipoint connections on a Nortel Multiservice Switch 7400 node, and figure Component tree for point-to-multipoint connections (page 229) displays the component tree of these point-to-multi point connections.

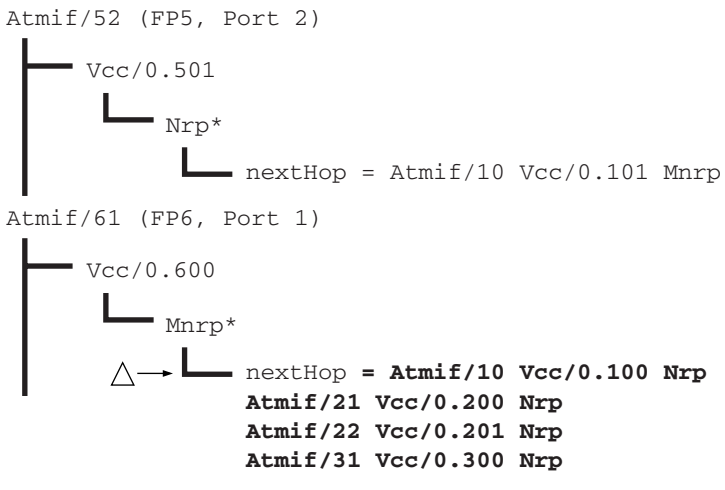
Point-to-multipoint connections



Component tree for point-to-multipoint connections



Continued from previous figure (Component tree for point-to-multipoint connections)



Each entity of a *Vcc/n.m Nrp*, *Vpc/n Nrp*, *Vcc/n.m Mnrp* or *Vpc/n Mnrp* component counts as one in the *lp/x eng arc ov connectionPoolCapacity*. For ATM interfaces, count the entries indicated by a plus (+) sign in figure Component tree for point-to-multipoint connections (page 229):

- *Atmif/10*, count 2 for *Lp/1 Eng Arc Ov connectionPoolCapacity*
- *Atmif/21* and *Atmif/22*, count 2 for *Lp/2 Eng Arc Ov connectionPoolCapacity*
- *Atmif/31*, count 1 for *Lp/3 Eng Arc Ov connectionPoolCapacity*
- *Atmif/50* and Atmif/52, count 2 for *Lp/5 Eng Arc Ov connectionPoolCapacity*
- *Atmif/61*, count 1 for *Lp/6 Eng Arc Ov connectionPoolCapacity*

Each entry in the *nextHop* attribute of a *Vcc/n.m Mnrp* or *Vpc/n Mnrp* component counts as one in the *lp/x eng arc ov multicastBranchesCapacity*. For Atmifs count the entries indicated by a triangle in figure Component tree for point-to-multipoint connections (page 229):

- *Atmif/10*, count 2 in *Lp/1 Eng Arc Ov multicastBranchesCapacity*
- *Atmif/61*, count 4 in *Lp/6 Eng Arc Ov multicastBranchesCapacity*

On Each root connection, we count one for *maxVccs* or *maxVpcs* on the FP where the root physically exists. There are two root connections:

- One counted for *maxVccs* under *Atmif/10 Ca* on FP1. (*Atmif/10 Vcc/0.101*)

- One counted for *maxVccs* under *Atmif/61 Ca* on FP6. (*Atmif/61 Vcc/0.600*)

On each leaf connection, we count one for *maxVccs* or *maxVpcs* on the FP where the leaves are physically located. There are a number of leaves scattered among the FPs

- For FP1, count one more for *maxVccs* under *Atmif/10 Ca*. (*Atmif/10 Vcc/0.100*)
- For FP2, count one for *maxVccs* under *Atmif/21 Ca* (*Atmif/21 Vcc/0.200*) and count one for *maxVccs* under *Atmif/22 Ca* (*Atmif/22 Vcc/0.201*)
- For FP3, count one for *maxVccs* under *Atmif/31 Ca* (*Atmif/31 Vcc/0.300*
- For FP5, count one for *maxVccs* under *Atmif/50 Ca* (*Atmif/50 Vcc/0.500*) and count one for *maxVccs* under *Atmif/52 Ca* (*Atmif/52 Vcc/0.500*)

ATM accounting

For SVCs and SPVCs, accounting configuration is governed by the attributes in the *accountingOptions* group under the *AtmIf Uni*, *AtmIf Iisp*, AtmIf Aini, AtmIf Pnni and their counterparts under Vpt. For PVCs, configuration is governed by the attributes in the *Provisioned* (*Prov*) group under the *AtmIf NailedUpAccounting* (*NAcct*) component. Both the *accountingOptions* group and the *Prov* group contain the following attributes:

- The *accountCollection* attribute turns accounting on or off at the ATM interface. When accounting is turned on, only connections set up from then on will be accounted. Thus, if accounting is required for an ATM interface, turning accounting on using this attribute should be done early during configuration.
- The *accountConnectionType* attribute specifies whether, from the point of view of accounting, this interface is one or the other of
  - an originating/terminating interface (for example an edge interface, connecting to outside the network)
  - an intermediate interface (connecting nodes inside the network)

The rules for configuring the *accountConnectionType* attribute are the following:

- interfaces connecting to outside the network should have the attribute set to *origTerm*. This value is the default for SVC and SPVC Uni interfaces and all PVC basic interfaces. For Pnni, Iisp, and Aini interfaces that connect to outside the network, the default must be changed.
- interfaces connecting nodes inside the network should have the attribute set to *intermediate*. This value is the default for Pnni, Iisp, and Aini interfaces. For Uni or basic interfaces connecting nodes inside the network, the default must be changed.

It is very important to set this attribute to the correct value, even for interfaces and nodes on which accounting is turned off.

The currently configured value of attributes *accountClass* and *serviceExchange* are recorded in the accounting record. These attributes do not affect the internal operation of ATM accounting.

For more information about configuring ATM accounting, see the following sections:

- Accounting system defaults (page 232)
- Accurate record generation (page 233)
- Accounting for permanent virtual connections (page 234)
- Accounting for switched connections (page 235)

Accounting system defaults

ATM accounting system attributes have predetermined defaults. See the following section for information about default settings and how to change the settings:

- The default settings (page 232)
- Changing the default settings (page 232)

The default settings

The default settings are as follows:

- accounting is disabled at *Uni*, *Iisp, and Aini* interfaces, as well as at basic interfaces
- when accounting is enabled, *Uni* and basic interfaces generate originating or terminating accounting records, while *Iisp* and Aini interfaces generate intermediate records

Changing the default settings

When accounting is enabled for a typical configuration the default values of attribute *accountConnectionType* do not need to be changed.

In typical situations, the defaults are such that, if someone enables accounting at any kind of interface, the accounting record generated there will be marked correctly as from an originating, intermediate, or terminating interface and the billing system will not get confused.

However, the default values do need to be changed under the following circumstances:

- when accounting is enabled at a basic interface on an inner node, it should be configured to generate intermediate accounting records

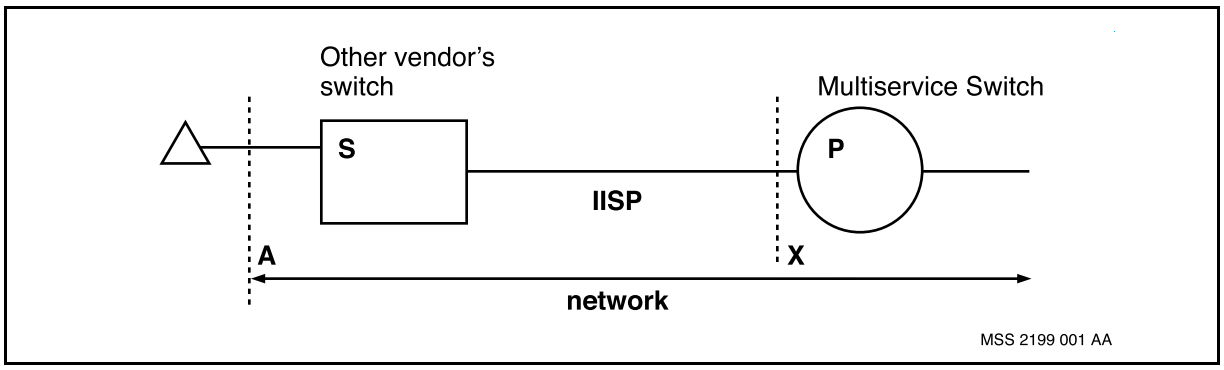
- when an *Iisp* interface or an Aini is connected to outside the network, it must be configured to generate originating or terminating records
- when a *Uni* connects two nodes inside the network, it must be configured to generate intermediate records

Another situation where defaults may need to be changed is shown in the figure, Situation when default settings for accounting might need to be changed (page 233). The edge node S is another vendor's device and does not support accounting. In this case accounting should be enabled on interface X of node P. Even though it is an *Iisp*, this interface should mark the accounting records as originating/terminating rather than intermediate.

Incorrect setting of attribute *accountConnectionType* might result in calls being rejected by the other vendor's device, even when accounting is turned off at interface X.

In all cases, configuring must be done consistently across the network in order to avoid, for any given call, generating more than one originating-point accounting record, as well as more than one terminating-point accounting record (from two different nodes). In this way, the billing system will not have problems correlating the accounting records.

Situation when default settings for accounting might need to be changed



Accurate record generation

Some support in configuring is necessary to ensure accurate record generation because a node

- cannot automatically determine whether it is an edge or an inner node in the network
- does not know if accounting is to be on or off for a given interface

Accounting record generation is controlled by the configuration data in *AtmIf* (attribute *accountCollection* in subcomponent *Uni*, *Iisp, or Aini* for SVCs and SPVCs, or *NAcct* for PVCs). Also, the determination that an interface is on the edge of the network or not is performed by a provisionable attribute (*accountConnectionType*).

The attribute *accountConnectionType* must be correctly configured at each interface, whether or not accounting is enabled.

The attributes, accountCollection and accountConnectionType need to be configured such that

- interfaces that connect to points outside the network have accounting turned on and attribute *accountConnectionType* set to *origTerm*, so that the generated accounting records are marked as *originating* or *terminating,* depending on whether the call request originates or terminates with the interface
- interfaces that connect to points inside the network either have accounting turned off (recommended) or have accounting turned on (for example, for test purposes). Either way, attribute *accountConnectionType* must be set to *intermediate* so that the accounting records are marked as *intermediate*

Accounting for permanent virtual connections

Configuring the *accountCollection* and the *correlationTag* attributes enables and disables accounting. The specified accounting reasons can be used by the customer to tag the accounting records for internal purposes (for their own downstream processing system).

Configuring the *accountConnectionType* attribute determines the value of the field *connectionPointType* in the accounting record. This information can be used by downstream processing systems to distinguish between records in case there are two or more interfaces (points) along the connection that generate accounting records.

With attributes *accountClass* and *serviceExchange*, additional record tagging can be done to include information configured at the interface level in the accounting record. If you do not configure *accountClass* and *serviceExchange*, Nortel Multiservice Switch systems use default value zero in the accounting records.

Changes to the configured attributes take effect on the next calls that are set up by the interface (calls in progress are not affected). The exception to this rule is that the change takes place immediately when accounting is disabled (account collection is assigned the value <empty set>).

Accounting for switched connections

This section provides the components, attributes, and configuration procedures that you require for ATM accounting on SPVCs, SVCs, SPVPs, and SVPs.

The *accountingOptions* group includes the provisionable attributes that support accounting.

Attribute *accountCollection* is used to enable and disable accounting. The specified accounting reasons can be used by the customer to tag the accounting records for internal purposes (for their own downstream processing system).

The configuring of attribute *accountConnectionType* determines the value of the field *connectionPointType* in the accounting record. This information can be used by downstream processing systems to distinguish between records in case there are two or more interfaces (points) along the connection that generate accounting records.

This attribute must be configured correctly for all interfaces, including interfaces for which accounting is turned off. Correct configuring is necessary because the signaling information element (IE) containing the call correlation tag is handled differently depending on whether the interface is an edge (originating/terminating) or intermediate.

When a Multiservice Switch node is connected to a non-Multiservice Switch device using an IISP interface, set the attribute *accountConnectionType* to the value *origterm*. The attribute *accountConnectionType* should not be set to the value *intermediate* (default value for *Iisp*). If the value is set to *intermediate*, the non-Multiservice Switch device receiving the call can reject it with an invalid IE, 7F.

With attributes *accountClass* and *serviceExchange*, additional record tagging can be done to include information configured at the interface level in the accounting record.

Multiservice Switch systems use a default value of zero in the accounting records if you do not configure the *accountClass* and *serviceExchange* attributes.

To specify the type of the interface, configure one of the *Pnni*, *Uni, Iisp*, or *Aini* components under an ATM interface. If you do not define a *Pnni*, *Uni, Iisp*, or *Aini* component under the ATM interface, the system defines the interface as basic. A basic interface supports NPVCs and NPVPs, for which you configure accounting through the *NAcct* component. The correlation tag for PVCs is configured under the *Vcc/Vcd* component. The correlation tag for PVPs is configured under the *Vpc/Vpd* component.

Changes to the configured attributes take effect on the next calls that are set up by the interface (calls in progress are not affected). The exception to this rule is that the change takes place immediately when accounting is disabled (account collection is assigned the value <empty set>).

Procedure conventions

This document uses the following procedure conventions:

- You can enter commands using full component and attribute names, or you can abbreviate them. The commands used in the procedures contain the full component and attribute names in the first instance. In the second instance, the component and attribute names are abbreviated. For more information on abbreviating component and attribute names, see NN10600-060 *Nortel Multiservice Switch 7400/15000/20000 Component Reference*. All component and attribute names are formatted in italics.
- The introduction of every procedure states whether you must perform the procedure in operational mode or provisioning mode. For more information on these modes, see Operational mode (page 237) or Provisioning mode (page 238).
- When you complete a procedure, you can verify your changes and then activate them as the new node configuration. For more information on completing configuration changes and exiting provisioning mode, see Activating configuration changes (page 238).

Operational mode

Procedures contained within this document can either be performed in operational mode or provisioning mode. When you initially log into a node, you are in operational mode. Nortel Multiservice Switch systems use the following command prompt when you are in operational mode:

```
#>
```

where:
is the current command number

In operational mode, you work with operational components and attributes. In operational mode, you can

- list operational components and display operational attributes to determine the current operating parameters for the node
- control the state of parts of the node by locking and unlocking components

- set certain operational attributes and enter commands to perform diagnostic tests

Provisioning mode

To change from operational mode to provisioning mode, type the following command at the operator prompt:

```
start Prov
```

Only one user can be in provisioning mode at a time. Nortel Multiservice Switch systems use the following command prompt whenever you are in provisioning mode:

```
PROV #>
```

where:

is the current command number

In provisioning mode, you work with the provisionable components and attributes that contain the current and future configurations of the node. You can add and delete components, and display and set provisionable attributes. For information on completing the configuration changes, exiting provisioning mode, and returning to operational mode see Activating configuration changes (page 238).

For information on operational and provisionable attributes, see NN10600-060 *Nortel Multiservice Switch 7400/15000/20000 Component Reference*.

Activating configuration changes

Several procedures in this document ask that you complete the configuration changes. When you complete the configuration changes, you are activating the configuration changes, confirming that you want to activate them, and saving the changes. You are instructed to complete the configuration changes only at the end of procedures that you perform in provisioning mode.

CAUTION

Activating a provisioning view can affect service

Activating a provisioning view can result in a CP reload or restart, causing all services on the node to fail. See NN10600-050 *Nortel Multiservice Switch 7400/15000/20000 Command Reference*, for more information.

CAUTION
Risk of service failure
When you activate the provisioning changes (see step 3), you have 20 minutes to confirm these changes. If you do not confirm these changes within 20 minutes, the shelf resets and all services on the node fail.

1 Verify that the provisioning changes you have made are acceptable.

```
check Prov
```

Correct any errors and then verify the provisioning changes again.

2 If you want to store the provisioning changes in a file, save the provisioning view.

```
save -f(<filename>) Prov
```

3 If you want these changes as well as other changes made in the edit view to take effect immediately, activate, confirm, and commit the provisioning changes.

```
activate Prov
confirm Prov
commit Prov
```

4 End the provisioning session.

```
end Prov
```

Nortel Multiservice Switch 7400/15000/20000

ATM Configuration Management

Publication: NN10600-710
Document status: Standard
Document issue: 7.2S1
Document date: March 2006
Product release: PCR7.2 and up
Job function: Operations
Type: NTP
Language type: U.S. English