Nortel Networks Multiservice Switch

7400/15000/20000

ATM Configuration Management

NN10600-710

Nortel Networks Multiservice Switch 7400/15000/20000

ATM Configuration Management

Publication: NN10600-710
Document status: Standard
Document version: 6.1S2
Document date: November 2004

Publication history

November 2004

6.1S2 Standard
General availability. Contains information on Nortel Networks Multiservice Switch 7400, 15000, and 20000 for the PCR6.1 release.

Contents

List of figures

List of tables

About this document

NN10600-710 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Configuration Management* describes how to configure Nortel Networks Multiservice Switch ATM services for permanent and switched connections.

The following topics are discussed in this section:

Who should read this document and why

This guide is for persons who need to configure Nortel Networks Multiservice Switch ATM services for their network.

What you need to know

This guide requires that you understand Nortel Networks Multiservice Switch network architecture and operation. You should also understand ATM technology and the open systems interconnection (OSI) model.

Use the following documents to help you understand Nortel Networks Multiservice Switch ATM concepts before you use the information in this guide:

- NN10600-700 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Technology Fundamentals*
- NN10600-702 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals*
- NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*
- NN10600-706 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*
- NN10600-707 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Queuing and Scheduling Fundamentals*
- NN10600-708 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM CAC and Bandwidth Fundamentals*

Use the information in NN10600-715 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Fault and Performance Management* to help you monitor and maintain ATM services after you have configured ATM in your network.

How this document is organized

This book contains task-oriented procedures for configuring ATM in a Nortel Networks Multiservice Switch network. Each procedure has a component tree hierarchy so the operator has a quick view of which components and attributes the procedure requires.

This book contains the following sections:

What's new in this document

The following features were added to this document:

- "PNNI local and global rerouting" (page 25)
- "Multiservice Switch ATM SPVC/SPVP call redirection" (page 25)

Other changes made to this document include the following.

- The section "ATM resource management configuration" (page 37) was updated with information about configuring CQC-based FPs and ATM Resource and Frame Resource values.
- The sections "Configuring traffic management for service categories" (page 95), "Optimizing traffic management for virtual channels" (page 123), and "Optimizing traffic management for virtual paths" (page 141) were updated with additional information regarding the forceTagging attribute.
- The section "Configuring point-to-multipoint permanent virtual channels" (page 128) was updated to indicate the maximum number of leaves that can be linked to a single root PVC.
- Procedure sections have been reorganized to comply with Modular Task Based Information (MTBI) standards.
- The terms Passport and PVG have been rebranded in conjunction with the new Nortel Networks' brand simplified naming format. Passport is now referred to as the Nortel Networks Multiservice Switch, and PVG is

now Media Gateway 7480/15000. For more information on the product rebranding, refer to NN10600-000 *Nortel Networks Multiservice Switch 7400/15000/20000 What's New in PCR6.1*.

- Procedure sections have been reorganized to comply with Modular Task Based Information (MTBI) standards.
- For CR Q00804394, the section "Considerations for the DS1 MSA32 FP and the E1 MSA32 FP" (page 423) was updated with information about the allowed total of *connectionPoolCapacity*, *multiCastBranchesCapacity*, *protectedConnectionPoolCapacity* and *protectedMcastBranchesCapacity*.
- For CR Q00857292, the *sourceCalledAddress* attribute was changed to *secondaryCalledAddress* in the following figures: "Configuring call redirection for Atmif vcc component hierarchy" (page 183), "Configuring call redirection for Atmif vpc component hierarchy" (page 185), and "Configuring call redirection for Atmif vpt component hierarchy" (page 187).
- For CR Q01011313, the figure "PNNI trace destination interface component hierarchy" (page 67) was updated to state that the <ifType> is Pnni only.
- For CR Q00857334, the following tasks interchanged positions in the figure "Switched connection provisioning procedures" (page 156): Configuring call redirection for an AtmIf Vpc Src (optional) and Configuring call redirection for an AtmIf Vpt Src (optional). The task Configuring call redirection for an AtmIf Vpt Src (optional) was changed to Configuring call redirection for an AtmIf Vpt Vcc Src (optional).
- For CR Q00904657, the limitations of configuring ATMIFs for a 2pSTM1eCh FP was added to the explanation of command variable <n> in "Configuring an AtmIf" (page 72).
- The Variable definitions table in section "Configuring connection admission control" (page 85) was updated with more information about the *maxVcc* attribute.
- Updated section "Configuring traffic management for service categories" (page 95) to include information about GQM-based FPs.

- removed all mention of the 1-port OC-48/STM-16 POS FP with PEC NTHW39 and card type 1pOC48SmIrPos since it is service discontinued (SDed)

Multiservice Switch ATM SPVC/SPVP call redirection

The following section was updated for this feature: “Switched connection provisioning procedures” (page 155)

The following sections were added for this feature:

- “Configuring call redirection for the ATM interface virtual channel connection” (page 182)
- “Configuring call redirection for the ATM interface virtual path connection” (page 184)
- “Configuring call redirection for the ATM interface virtual path terminator” (page 186)

PNNI local and global rerouting

The following section was updated for this feature:

- “Configuring rerouting on a non-PNNI ATM interface” (page 283)
- “Configuring rerouting on a PNNI ATM interface” (page 287)
- “Configuring a rerouting override for a soft PVC or soft PVP” (page 298)

Text conventions

This document uses the following text conventions:

- `nonproportional spaced plain type`

  Nonproportional spaced plain type represents system generated text or text that appears on your screen.

- `nonproportional spaced bold type`

  Nonproportional spaced bold type represents words that you should type or that you should select on the screen.

- *italics*

  Statements that appear in italics in a procedure explain the results of a particular step and appear immediately following the step.

  Words that appear in italics in text are for naming.

- `[optional_parameter]`

  Words in square brackets represent optional parameters. The command can be entered with or without the words in the square brackets.

- `<general_term>`

  Words in angle brackets represent variables which are to be replaced with specific values.

- UPPERCASE, lowercase

  Nortel Networks Multiservice Switch node commands are not case-sensitive and do not have to match commands and parameters exactly as shown in this document, with the exception of string options values (for example, file and directory names) and string attribute values.

- |

  This symbol separates items from which you may select one; for example, ON|OFF indicates that you may specify ON or OFF. If you do not make a choice, a default ON is assumed.

- ...

  Three dots in a command indicate that the parameter may be repeated more than once in succession.

The term absolute pathname refers to the full specification of a path starting from the root directory. Absolute pathnames always begin with the slash (/) symbol. A relative pathname takes the current directory as its starting point, and starts with any alphanumeric character (other than /).

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 Networks 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 27) or “Provisioning mode” (page 28).
- 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 28).

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 Networks 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 Networks 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 28).

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

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 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`

Related documents

For a list of related industry standards, see the NN10600-700 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Technology Fundamentals*.

See the following documents in the Nortel Networks Multiservice Switch documentation suite for additional information on ATM services and features:

- NN10600-030 *Nortel Networks Multiservice Switch 7400/15000/20000 Overview*
- NN10600-700 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Technology Fundamentals*

- NN10600-702 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals*
- NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*
- NN10600-706 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*
- NN10600-707 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Queuing and Scheduling Fundamentals*
- NN10600-708 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM CAC and Bandwidth Fundamentals*
- NN10600-715 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Fault and Performance Management*
- NN10600-720 *Nortel Networks Multiservice Switch 7400/15000/20000 AAL1 Circuit Emulation Operations*
- NN10600-730 *Nortel Networks Multiservice Switch 7400/15000/20000 Inverse Multiplexing for ATM Operations*
- NN10600-920 *Nortel Networks Multiservice Switch 7400/15000/20000 Operations: Frame Relay to ATM Interworking*
- NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Technology Fundamentals*
- NN10600-801 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Configuration Management*
- NN10600-581 *Nortel Networks Multiservice Switch 7400/15000/20000 VPN Technology Fundamentals*
- NN10600-582 *Nortel Networks Multiservice Switch 7400/15000/20000 VPN Configuration Management*
- NN10600-060 *Nortel Networks Multiservice Switch 7400/15000/20000 Component Reference*
- NN10600-420 *Nortel Networks Multiservice Switch 7400/15000/20000 Operations: Trunking*
- NN10600-560 *Nortel Networks Multiservice Switch 7400/15000/20000 Accounting*

- *Nortel Networks Multiservice Switch Release Notes*

If you are using the paper version of the documentation library, see NN10600-001 *Nortel Networks Multiservice Switch 7400/15000/20000 Basics: Using the Documentation* for a complete list of documents. If you are using an online version of the library, see the *Passport InfoMap*, which provides you with links to other documents.

For information on last minute updates, see the *Nortel Networks Multiservice Switch Release Notes* included with the software release.

How to get more help

For information on training, problem reporting, and technical support, see the “Nortel Networks support services” section in NN10600-030 *Nortel Networks Multiservice Switch 7400/15000/20000 Overview*.

Chapter 1
ATM configuration

This document provides procedures to configure asynchronous transfer mode (ATM) features and connections.

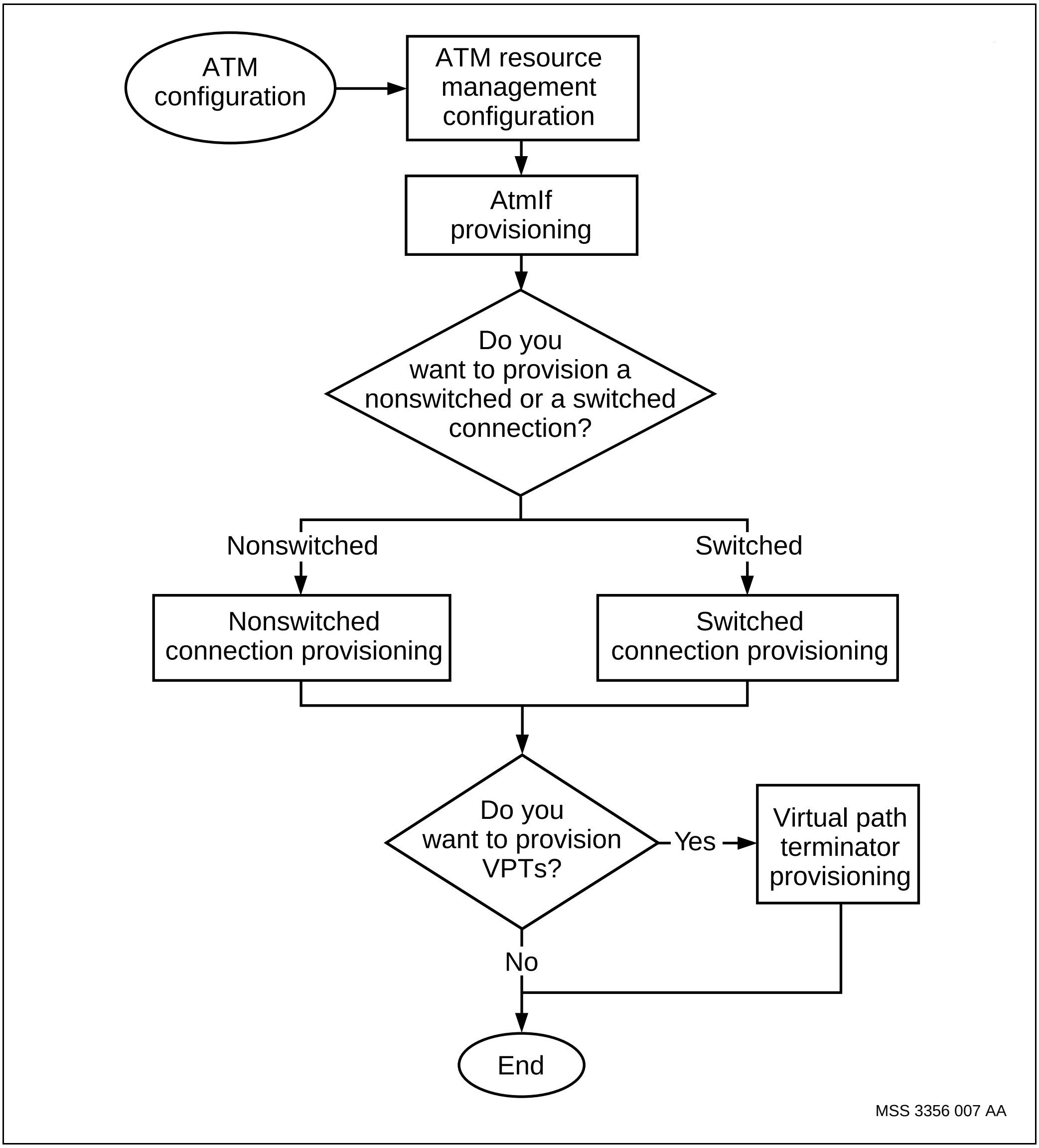
Prerequisites to ATM configuration

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

ATM configuration tasks

This workflow shows you the sequence of tasks you perform to configure ATM resource management. To link to any task, go to “ATM configuration task navigation” (page 35).

Figure 1
ATM configuration tasks



ATM configuration task navigation

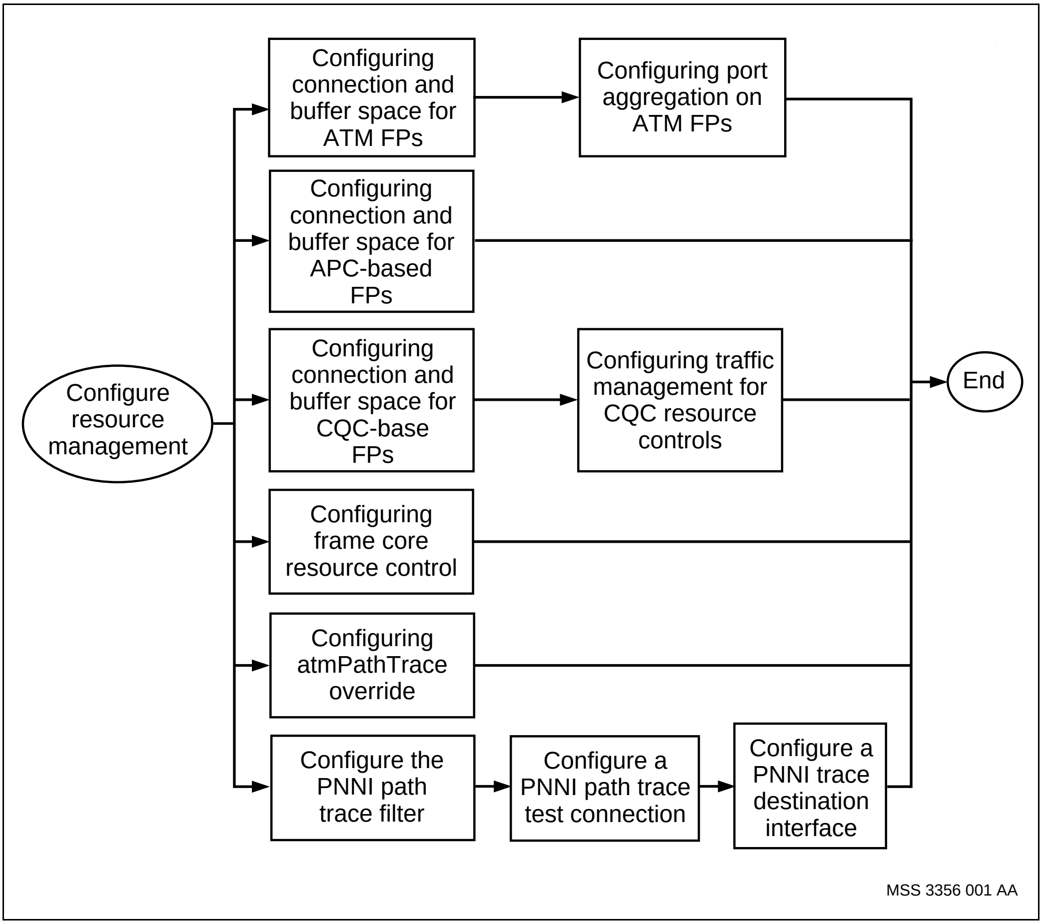
Chapter 2
ATM resource management configuration

Change the default configuration for queue and memory resources to monitor and control common hardware resources on a function processor (FP) support ATM services.

ATM resource management configuration procedures

This task flow shows you the sequence of procedures you perform to configure ATM resource management. To link to any procedure, go to “ATM resource management procedure navigation” (page 38).

Figure 2
ATM resource management configuration procedures



ATM resource management procedure navigation

- “Configuring connection and buffer space for ATM IP FPs” (page 40)
- “Configuring port aggregation on ATM IP FPs” (page 45)
- “Configuring connection and buffer space for APC-based FPs” (page 47)
- “Configuring connection and buffer space for CQC-based FPs” (page 51)

Configuring connection and buffer space for ATM IP FPs

Configure connection (the amount of memory required to support the number of ATM connections) and buffer space (remaining memory not required for connection space which is used for cell or frame queue memory) for ATM IP FPs to set the maximum number of connections or change the sizes of cell or frame PQC memory.

Prerequisites

- See NN10600-707 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Queuing and Scheduling Fundamentals* for more information on queue memory and connections.
- 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.
- Provisioned connection space beyond 29696 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 provisioned connection resources has not been satisfied, determine the number of provisioned 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 provisioned connection resources is greater than 29696 in the *LpEngArcOv* component and the number of available and used connection resources in the *LpEngArc* component is equal to exactly 29696, 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

> **WARNING**
> **This procedure resets the FP**
> Configure the *Arc* component before you activate any ATM service to prevent service interruption.

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. For more details, see “Adjusting resources for hitless services” (page 409).

```
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>

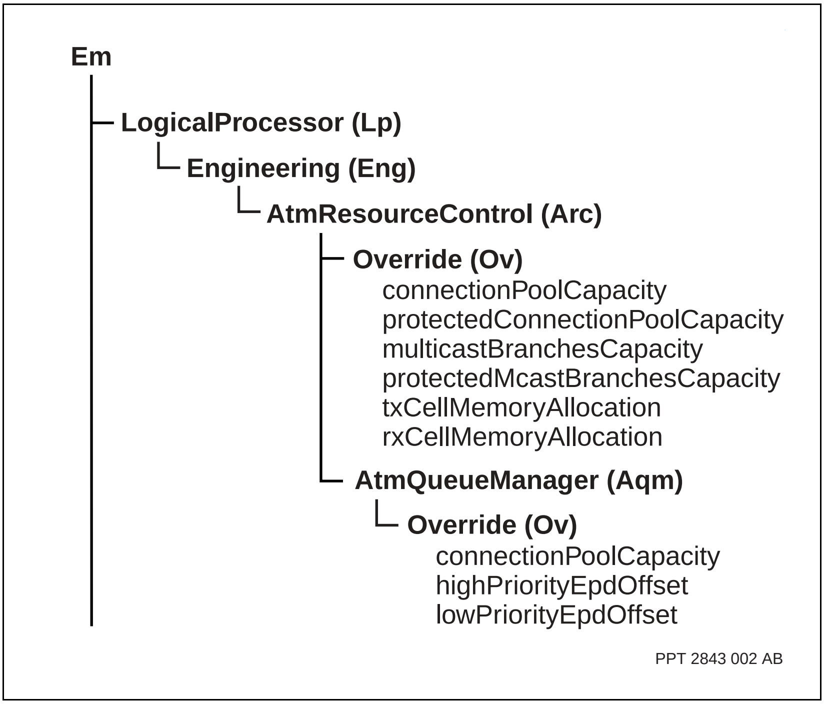
Variable definitions

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.
(Sheet 1 of 2)	

Variable	Value
<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 407). For ATM IP FPs that can provide more than four ATM interfaces, see “Adjusting resources for hitless services” (page 409). When map mode trunks are provisioned 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.
<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*. For ATM IP FPs that can provide more than four ATM interfaces, see “Adjusting resources for hitless services” (page 409)
<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.
(Sheet 2 of 2)	

Procedure job aid

Figure 3
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. Port aggregation on ATM IP FPs applies to all ports, and is configured under the *Arc Aqm/0* component. That is, for ATM IP OC-3 FPs, 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

> **WARNING**
> **This procedure resets the FP**
> Configure the *portCongestionPolicy* component before you activate any ATM service to prevent service interruption.

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.

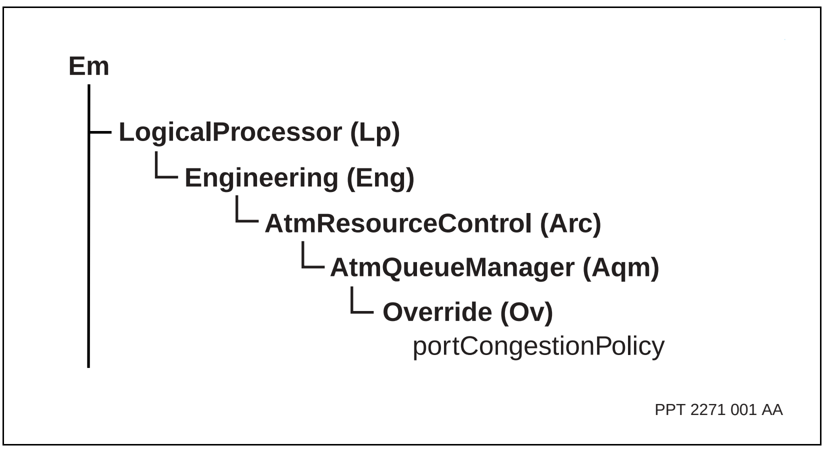
```
set Lp/<lp> Eng Arc Aqm/0 Ov portCongestionPolicy
<policy>
```

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

Figure 4
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.

See NN10600-707 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Queuing and Scheduling Fundamentals* for more detailed information about the relationship between the *Arc* component, the *Apc* component, and connection and buffer space.

Prerequisites

- 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

WARNING
This procedure resets the FP
Configure the *Arc* component before you activate any ATM service to prevent service interruption.

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 to the *EmissionPriorities* in the *APC* instance of the *Lp* component.

```
set Lp/<lp> Eng Arc Apc/<n> Ov classBufferPoolLimit
<cBPL>
```

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.

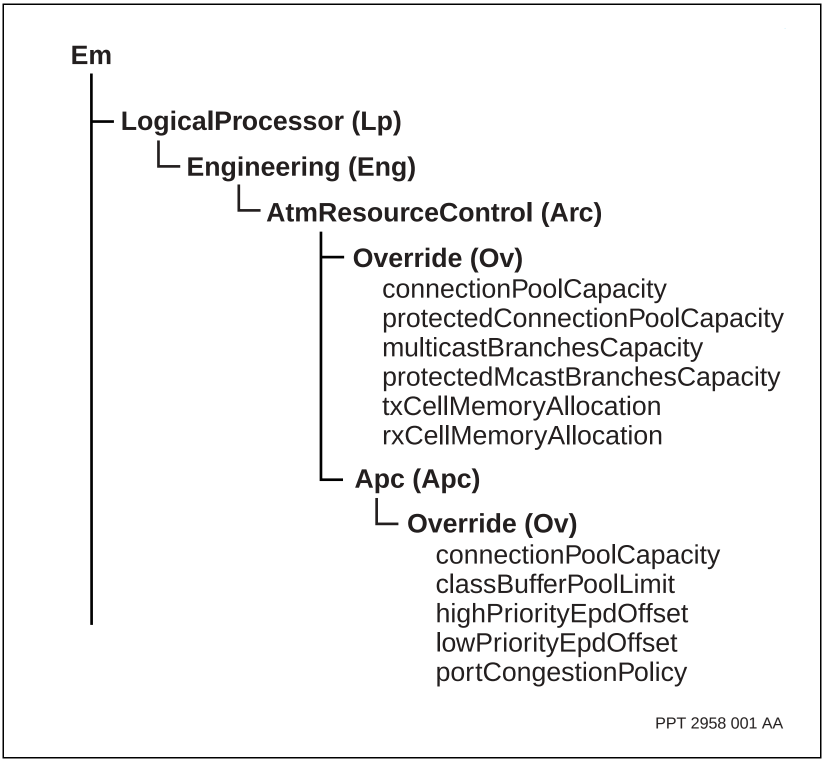
```
set Lp/<lp> Eng Arc Ov rxCellMemoryAllocation <rxCell>
```

Variable definitions

Variable	Value
<cBPL>	is a vector array of eight numeric values from 0 to 100. The *classBufferPoolLimit* is a percentage of the total buffer space at the egress, and must have the same value for the same emission priorities level over different *AtmIf* instances. For more information, see NN10600-060 *Nortel Networks 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

Figure 5
Connection and buffer space for APC-based FPs 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. See “Configuring traffic management for CQC resource controls” (page 55). See NN10600-707 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Queuing and Scheduling Fundamentals* and for more detailed information.

Procedure steps

> **WARNING**
> **This procedure resets the FP**
> Configure the *Arc* component before you activate any ATM service to prevent service interruption.

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>

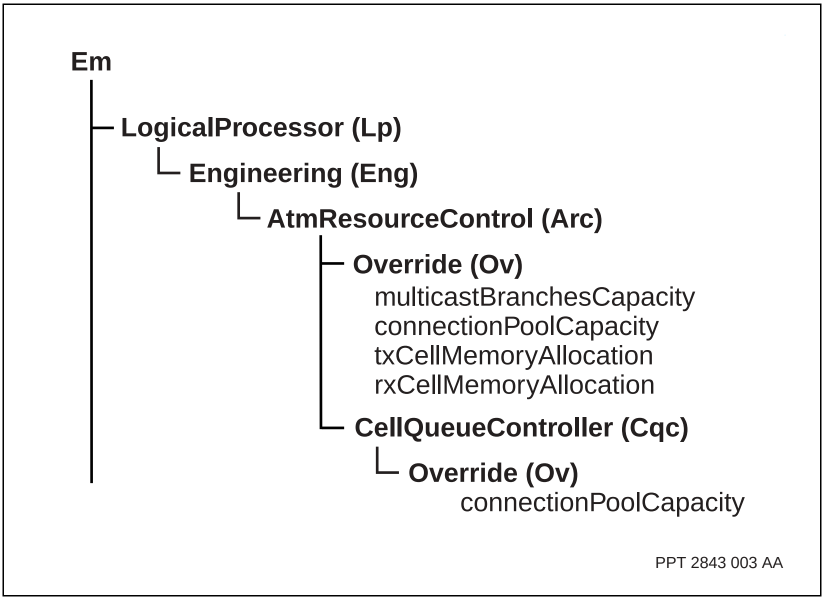
Variable definitions

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.
(Sheet 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 provisioned 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.
(Sheet 2 of 2)	

Procedure job aid

Figure 6
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.

Prerequisites

- For more detailed information about traffic management parameters for CQC-based FPs, see NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals,* NN10600-706 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*, NN10600-707 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Queuing and Scheduling Fundamentals*, and NN10600-708 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM CAC and Bandwidth Fundamentals*.

Procedure steps

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>
```

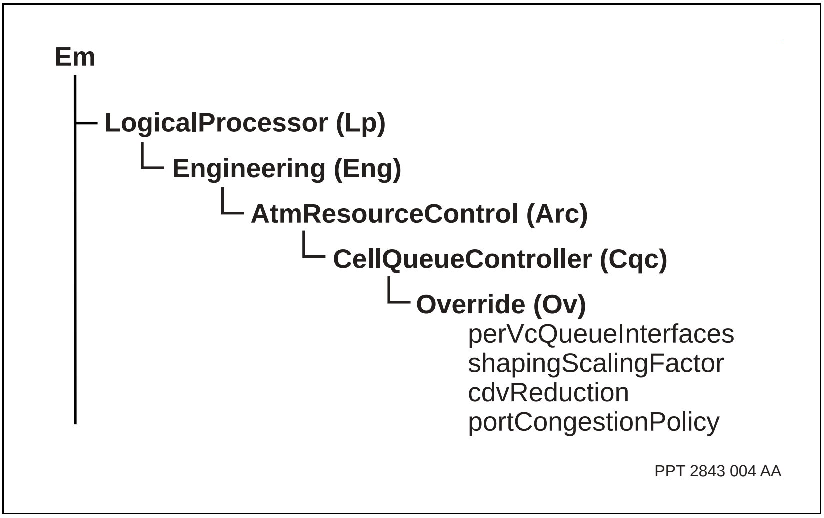
Variable definitions

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.
<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 Networks Multiservice Switch 7400/15000/20000 Inverse Multiplexing for ATM Operations* for details. The fewer the ports provisioned for traffic shaping and per-VC queuing, the greater the number of shaping rates available per port.
(Sheet 1 of 2)	

Variable	Value
<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 provisionable per LP.
(Sheet 2 of 2)	

Procedure job aid

Figure 7
Traffic management for CQC resource controls component hierarchy



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.

When configuring, if all ports are removed from a CQC-based FP and it is reset, after it comes back up the *lp/n eng arc* and *lp/n eng fcrc* attributes values are not correct. To get correct attribute values, at least one port on the FP needs to be provisioned.

Procedure steps

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 FP’s forwarding table through the *ipRoutesPoolCapacity* attribute.

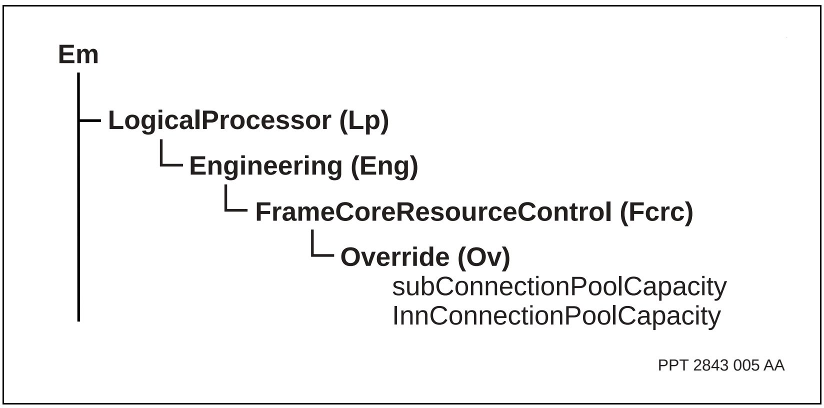
set Lp/<lp> Eng Fcrc Pqc Ov ipRoutesPoolCapacity <ipRtsCap>

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

Figure 8
Frame core resource control component hierarchy



Configuring atmPathTrace override

Configure atmPathTrace override to determine the logical nodes and links that new point-to-point connections traverse.

Prerequisites

- 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

> **WARNING**
> **This procedure resets the FP**
> Configure the *AtmPathTrace* component before you activate any ATM service to prevent service interruption.

1 Add an *AtmPathTraceOverride* component.

```
add Lp/<lp> Eng AtmPathTrace Ov
```

2 Set the maximum trace record attribute or maximum concurrent trace requests.

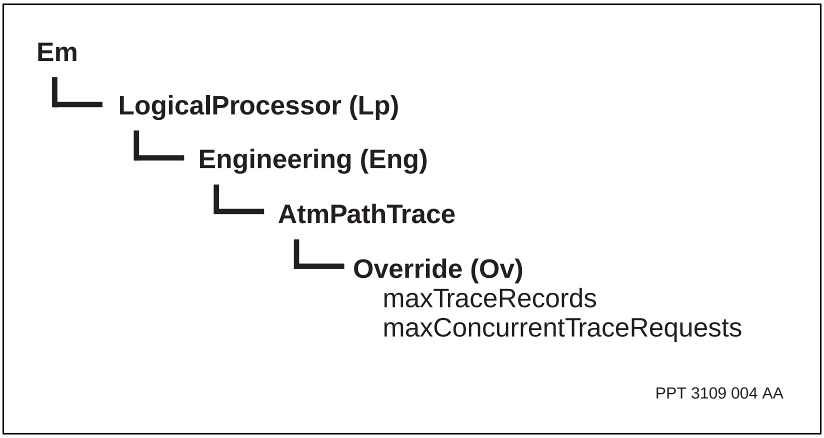
```
set Lp/<lp> Eng AtmPathTrace Ov
<atmPathTraceOverrideAttributes>
```

Variable definitions

Variable	Value
<atmPathTraceOverrideAttributes>	is maxTraceRecords or maxConcurrentTraceRequests.
<lp>	is the number of the LP associated with the ATM FP.

Procedure job aid

Figure 9
AtmPathTrace Override component hierarchy



Configuring the PNNI path trace filter

Configure the PNNI path trace filter to change the default values, if required.

Prerequisites

- Add the *override* component and configure the *maxTraceRecords* and *maxConcurrentTraceRequests* attributes using the procedure “Configuring atmPathTrace override” (page 60).

Procedure steps

1 Use one of the following procedures to define the ATM interface on the source node (the node that originates the call setup request).

- “Configuring a user-to-network connection” (page 208) to define a UNI
- “Configuring an inter-switch protocol interface” (page 213) to define an IISP
- “Configuring an ATM inter-network interface” (page 216) to define an AINI
- “Configuring a PNNI” (page 219) to define a PNNI

2 Add the *PathTraceFilter* component under the ATM signaling interface component.

```
add AtmIf/<n> <IfType> PathTraceFilter
```

3 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
```

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

```
set atmif/<n> <IfType> PathTraceFilter <ptf component>
```

5 Set the control for the *PathTraceFilter* component.

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

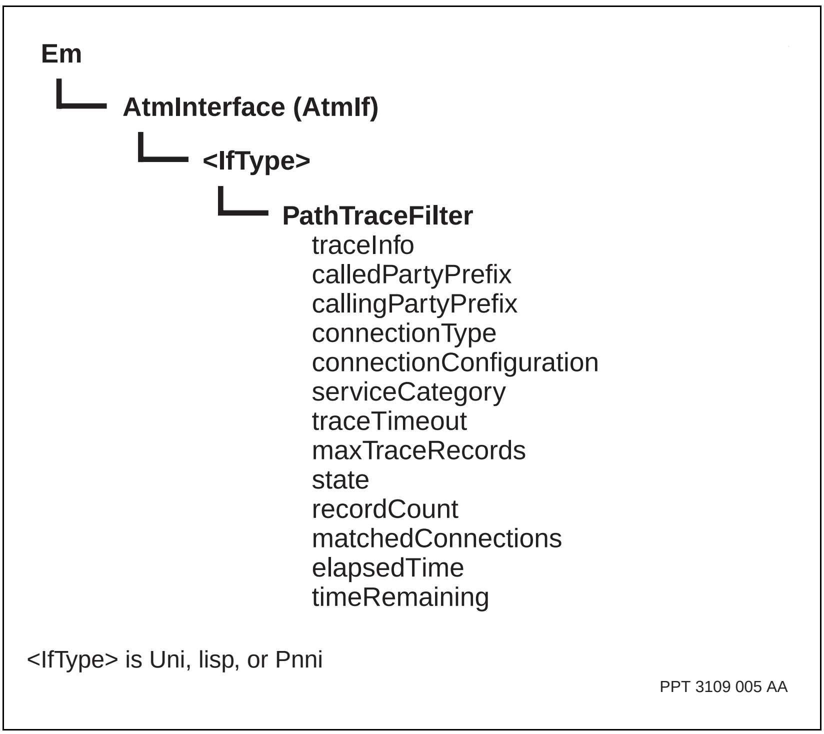
```
set AtmIf/<n> <IfType> PathTraceFilter maxTraceRecords
<m>
```

Variable definitions

Variable	Value
<IfType>	is uni, Iisp, Aini, or Pnni.
<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.
<ptf>	is calledPartyPrefix, callingPartyPrefix, connectionType, connectionConfiguration, or serviceCategory.

Procedure job aid

Figure 10
PNNI path trace filter component hierarchy



Configuring a PNNI path trace test connection

Configure a PNNI path trace test connection to set its attributes.

> ***Note:*** The path trace test connection is operational. Consequently, the user does not need to be in provisioning mode when setting up its attributes.

Procedure steps

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

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

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

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

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

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

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

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

5 Set the crankback flag for the path trace connection.

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

Variable definitions

Variable	Value
<address>	is the destination address.
<calledVpiVci>	is the vpi/vci.
<crankback>	is either yes or no. The default is set to no.
<IfType>	is uni, Iisp, Aini or Pnni.
<n>	is the number of the ATM interface.
(Sheet 1 of 2)	

Variable	Value
<pathTraceTestConnectionAttributes>	is txTrafficDescType, txTrafficDescParm, rxTrafficDescType, rxTrafficDescParm, atmServiceCategory, fwdQosClass, fwdQosParameters, bwdQosClass, bwdQosParameters, bearerClassBbc, transferCapabilityBbc, clippingBbc, or bestEffort.
<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.
(Sheet 2 of 2)	

Procedure job aid

Figure 11
PNNI path trace test connection component hierarchy

- **EM**
  - **AtmIf**
    - **<IfType>**
      - **PathTraceTestConnection**
        - traceCrankback
        - testCalledVpiVci
        - testCalledAddress
        - testSpecifiedPath
        - atmServiceCategory
        - rxTrafficDescType
        - rxTraffficDescParm
        - txTrafficDescType
        - txTrafficDescParm
        - fwdQosClass
        - fwdQosParameters
        - bwdQosClass
        - bwdQosParameters
        - bearerClassBbc
        - transferCapabilityBbc
        - bestEffort
        - clippingBbc

<IfType> is Uni, Iisp, Pnni, or Aini

PPT 3109 007 AA

Configuring a PNNI trace destination interface

Configure a PNNI trace destination interface to change the trace destination interface from the DTL terminator to the PNNI interface, if required. By default, the trace destination interface is the DTL terminator. However, a PNNI interface can be administratively provisioned to be a trace destination interface. In this case, the call trace will collect information starting with the trace source node until the trace destination node (which is the outgoing interface for the call which is provisioned as a trace destination interface).

Procedure steps

1 Use the procedure “Configuring a PNNI” (page 219) to define the ATM interface on the source node (the node that originates the call setup request).

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

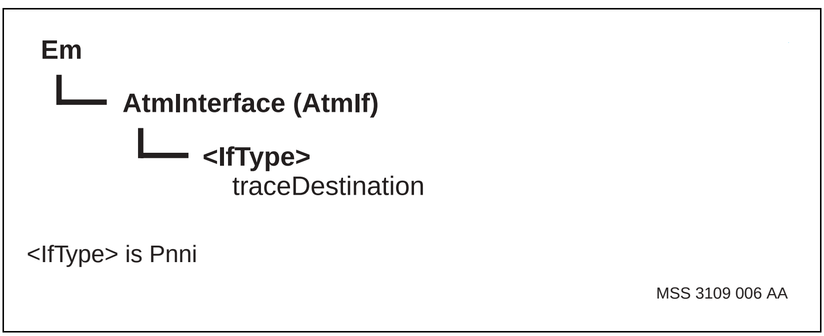
```
set AtmIf/<n> <IfType> TraceDestination <yes|no>
```

Variable definitions

Variable	Value
<IfType>	is pnni.
<n>	is the number of the ATM interface.
<yes|no>	is yes or no. The default value is no.

Procedure job aid

Figure 12
PNNI trace destination interface component hierarchy



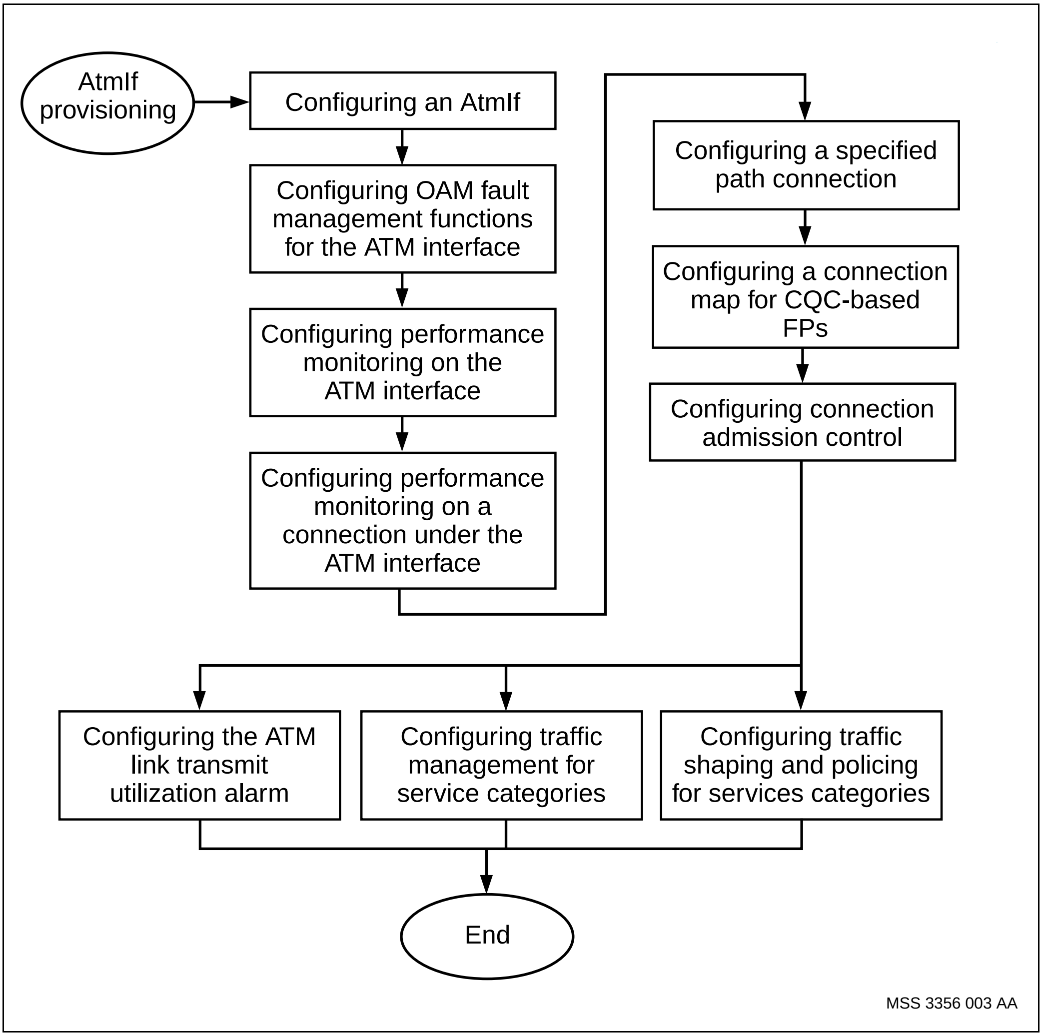
Chapter 3
AtmIf provisioning

Configure an ATM interface (AtmIf) to provide a connection between two network nodes or entities. Each entity may be a terminal, another Nortel Networks Multiservice Switch node, or an external ATM network. Additional optional parameters for the interface can be configured depending on which traffic contract is used.

AtmIf provisioning procedures

This task flow shows you the sequence of procedures you perform to provision AtmIf. To link to any procedure, go to “AtmIf provisioning procedure navigation” (page 70).

Figure 13
AtmIf provisioning procedures



AtmIf provisioning procedure navigation

- “Configuring an AtmIf” (page 72)
- “Configuring OAM fault management functions for the ATM interface” (page 74)

Configuring an AtmIf

Configure an AtmIf to add an ATM interface to a node. You can provision all *AtmInterface* (*AtmIf*) components on the node and perform semantic checks before provisioning the connections. This method ensures that the interfaces and connection mapping components are properly set up before continuing with detailed connection provisioning.

Procedure steps

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 For ATM IP FPs and eight port DS1/E1 CQC-based FPs, set the maximum number of VPI bits that are active for this interface.

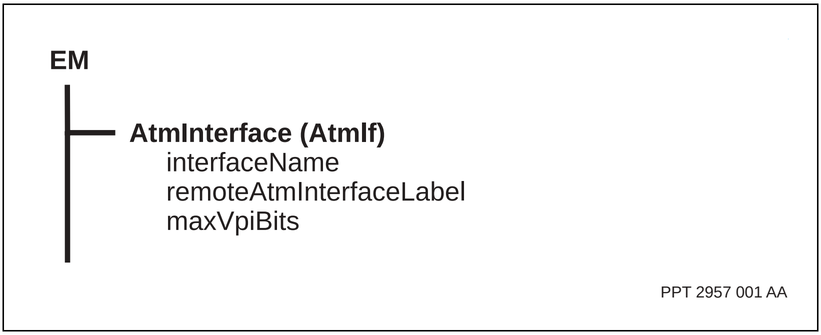
set AtmIf/<n> maxVpiBits <vpiBits>

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 ATMIFs 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 provisioned.
<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

Figure 14
AtmIf 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

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 default loopbacks for the ATM interface. This setting enables or disables the loopback on the ATM interface. The type of loopbacks available for a connection depends on the value of the connection point type associated with the VPC, VPT, or VCC.

```
set AtmIf/<n> remoteAtmInterfaceLabel <remoteAtmIf>
```

```
set AtmIf/<n> segSwitchSideLoopback <segSwLbk>
```

```
set AtmIf/<n> endToEndLoopback <eeLbk>
```

4 Enable or disable cell transfer delay (CTD) calculation.

```
set AtmIf/<n> ctdCalculation on|off
```

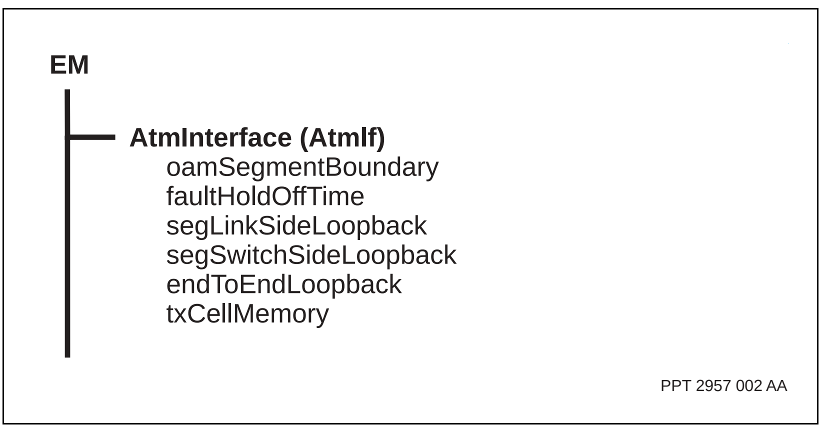
CAUTION
Use of CTD may affect SPVC operation
The CTD feature makes use of the ATM OAM loopback cells. The loss of three consecutive loopback cells will fail the loopback test and take down the SPVC. Possible causes of this include discarded cells due to congestion and traffic management, loss of connectivity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

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 initial 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.
<segLkLbk>	specifies whether link-side segment loopback insertion and termination should be performed on this connection. This value can be on, or off. The default value is off.
<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

Figure 15
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

1 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>
```

2 Choose how the cell loss ratio is measured on the device side of a connection.

```
set atmif/<n> pm switchSideClrType <switchSideClrType>
```

3 To enable auto start mode performance monitoring, set the control mode to autoStart and activate the provisioning view.

```
set atmIf/<n> pm controlMode <mode>
```

```
check prov
```

```
activate prov
```

```
confirm prov
```

4 Stop the measurement process by turning off the AtmIf Pm measurements.

```
set atmIf/<n> pm <segmentSide> <measurement>
```

```
check prov
```

```
activate prov
```

```
confirm prov
```

5 To enable on-demand performance monitoring, set the control mode to onDemand and activate the provisioning view.

```
set atmIf/<n> pm controlMode <mode>
```

```
check prov
```

```
activate prov
```

6 Start measurements for on-demand mode performance monitoring.

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

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

7 When you have taken enough measurements, stop measurements for on-demand mode performance monitoring.

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

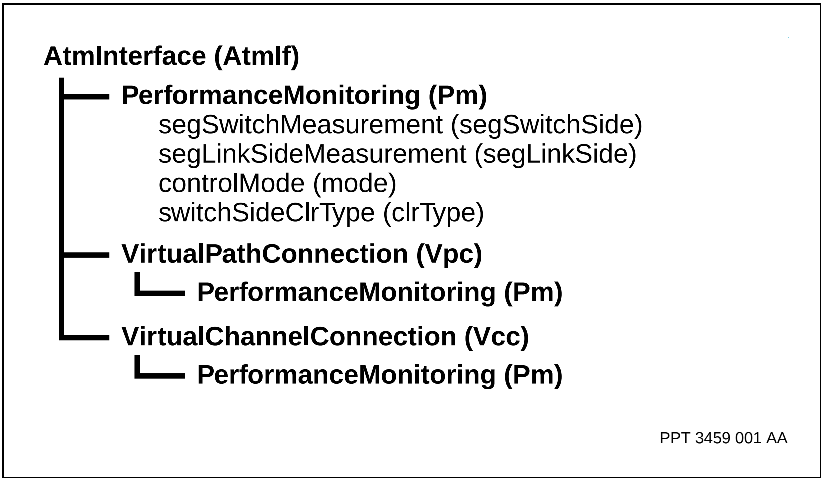
```
stop atmIf/<n> vpc/<x> pm/<side>
```

Variable definitions

Variable	Value
<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.

Procedure job aid

Figure 16
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

1 Add a *Pm* component under the *VCD* or *VPD* component.

```
add AtmIf/<n> <connection_type>/<y> <descriptor_type>
pm
```

2 Choose the measurements to calculate CL, AR, or both, and the side of the segment to test.

```
set atmif/<n> <connection_type>/<y> <descriptor_type>
pm &lt;segment side&gt; <measurement>
```

3 Turn the measurements off for the side of the segment you do not wish to test.

```
set atmif/<n> <connection_type>/<y> <descriptor_type>
pm &lt;segment side&gt; ~<measurement>
```

4 Choose how the cell loss ratio is measured on the device side of a connection.

```
set atmif/<n> vcc/<y> vcd pm switchSideClrType
<switchSideClrType>
```

5 Set the control mode.

```
set atmif/<n> <connection_type>/<y> <descriptor_type>
pm controlMode <mode>
```

6 To start measurements in auto-start mode, activate the provisioning view.

```
check prov
```

```
activate prov
```

7 To start measurements in on-demand mode, activate the provisioning view and then start performance monitoring under the connection.

```
check prov
```

```
activate prov
```

```
start atmif/<n> <connection_type>/<x> pm/<side>
```

8 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>
```

9 When you have collected enough measurements during auto-mode performance monitoring, stop measurements by activating a different provisioning view.

```
check prov
```

```
activate prov
```

Variable definitions

Variable	Value
<connection_type>	is vcc (for a virtual channel connection) or vpc (for 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.

Procedure job aid

Figure 17
Performance monitoring on a connection under the ATM interface component hierarchy

- **AtmInterface (AtmIf)**
  - **VirtualChannelConnection (Vcc)**
    - **PerformanceMonitoring (Pm)**
    - **VirtualChannelDescriptor (Vcd)**
      - **PerformanceMonitoring (Pm)**
        - segSwitchMeasurement (segSwitchSide)
        - segLinkSideMeasurement (segLinkSide)
        - controlMode (mode)
        - switchSideClrType (clrType)
  - **VirtualPathConnection (Vpc)**
    - **PerformanceMonitoring (Pm)**
    - **VirtualPathDescriptor (Vpd)**
      - **PerformanceMonitoring (Pm)**
        - segSwitchMeasurement (segSwitchSide)
        - segLinkSideMeasurement (segLinkSide)
        - controlMode (mode)
        - switchSideClrType (clrType)

PPT 3459 002 AA

Configuring a specified path connection

Configure a specified path connection to manually configure a path with a predetermined sequence of physical nodes and port IDs for an SPVC/SPVP/VPT SPVP call in flat and hierarchical PNNI topologies.

Prerequisites

- For information about attributes that may affect specified path connections, see “Specified path connection attributes” (page 102).

Procedure steps

1 Add the *Mdtl* component under the connection or to the existing SPVC/P connection.

add AtmIf/<n> <connection_type>/<m> src mdtl

2 Assign a primary path to the connection

set AtmIf/<n> <connection_type>/<m> src mdtl primaryPath <primarypathname>

3 Optionally, assign an alternate path to the connection.

set AtmIf/<n> <connection_type>/<m> src mdtl alternatePath <alternatepathname>

4 Optionally, enable/disable the automatic fallback path.

set AtmIf/<n> <connection_type>/<m> src mdtl automaticFallback <option>

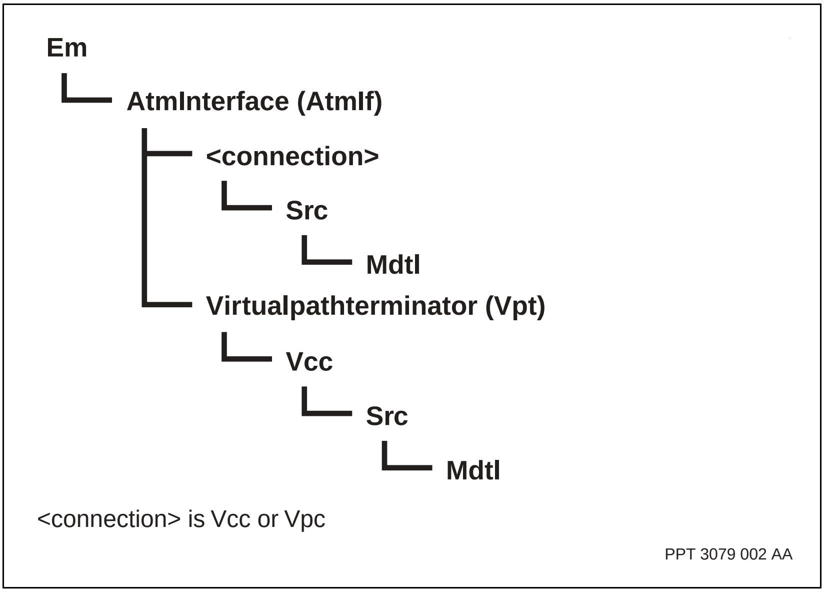
Variable definitions

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 vcc (for virtual channel connection) or vpc (for virtual path connection).
<m>	is the instance value of the *VCC, VPC,* or *VPT* component.
(Sheet 1 of 2)	

Variable	Value
<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.
(Sheet 2 of 2)	

Procedure job aid

Figure 18
Specified path connection component hierarchy



Configuring a connection map for CQC-based FPs

Configure a connection map for CQC-based FPs to define the range of address spaces for connections under the ATM interface. See “Connection map configuration for CQC-based FPs” (page 399).

Configuring connection admission control

Configure connection admission control to define the maximum number of connections for each instance of the ATM interface.

Procedure steps

1 Set the maximum number of connections (VPCs, VPTs, and VCCs) for the interface. See “Configuring connection and buffer space for ATM IP FPs” (page 40). 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 <ServiceCategory2>/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>
```

Variable definitions

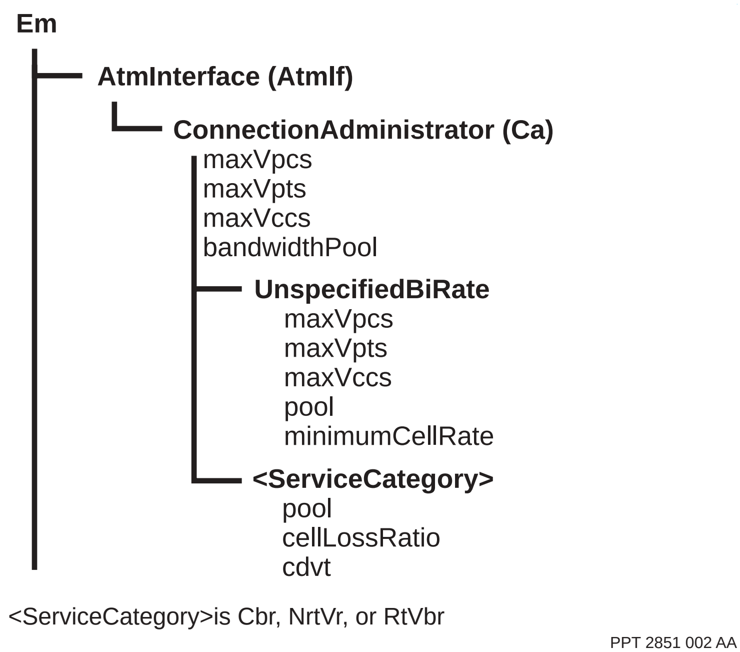
Variable	Value
<cdvt>	is a numeric value from 1 to10 000. The default is 250. This value applies to all CBR traffic on the interface. If usage parameter control (UPC) is enabled for the connection, the value in this attribute affects traffic policing configuration.
<clr>	is a negative logarithmic value (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 provisioned as a power of 10. For example, to provision a CLR of 10^{-10}, the <clrValue> would be 10.
(Sheet 1 of 3)	

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 resource management configuration” (page 37). If the FP is running spared ATM interfaces, see “Adjusting resources for hitless services” (page 409). 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 provisioned, 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 provisioned maxVccs or there is a lack of bandwidth available. Consideration for reserved signalling Vccs for ATM networking must be taken into account when provisioning the maxVccs. If an ATM networking protocol and n permanent Vccs (PVCs and SPVCs) are to be provisioned under an AtmIf, then the maxVccs under AtmIf Ca should be provisioned with a value of n + 2 (where 2 is the maximum number of reserved signalling Vccs for an ATM networking protocol).
(Sheet 2 of 3)	

Variable	Value
<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.
<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.
<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, NrtVbr, or Ubr. Only the 0 instance of a service category exists.
<ServiceCategory2>	is Cbr, RtVbr, or NrtVbr. Only the 0 instance of a service category exists.
(Sheet 3 of 3)	

Procedure job aid

Figure 19
Connection admission control component hierarchy



Configuring traffic shaping and policing for service categories

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

> ***Note:*** 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 95).
- 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 94) when provisioning usage parameter control (UPC) for service categories.
- See NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*.

- See NN10600-706 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*.
- See NN10600-707 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Queuing and Scheduling Fundamentals*.
- See NN10600-708 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM CAC and Bandwidth Fundamentals*.

Procedure steps

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>
```

3 Configure cell delay variation tolerance for the service category.

```
set AtmIf/<n> Ca <ServiceCategory>/0 cdvt <cdvt>
```

Variable definitions

Variable	Value
<cdvt>	is a decimal in the range of 1 to 1 200 000 (in microseconds). The default is 250. If this attribute has a high value, the UPC capability is more tolerant of bandwidth use beyond the specification of the traffic contract. This value also affects the equivalent cell rate for CBR connections.
<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.
(Sheet 1 of 2)	

Variable	Value
<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 93) 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 93) 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.
(Sheet 2 of 2)	

Procedure job aid

Figure 20
Traffic shaping and policing for service categories component hierarchy

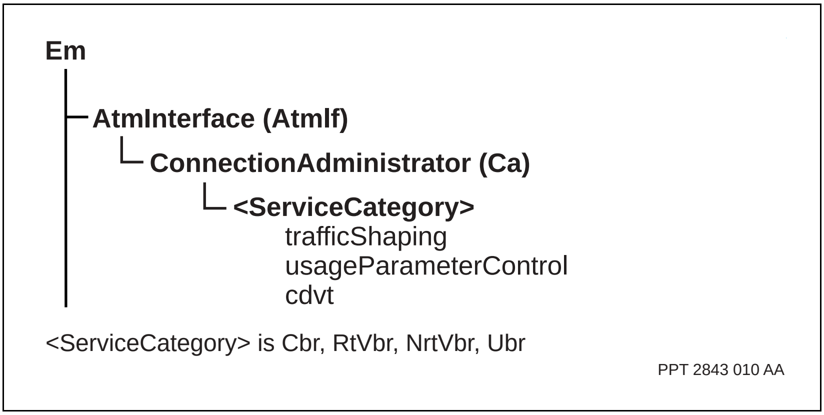


Table 1
Relationship between the txTrafficDescType and trafficShaping attribute values

Type of FP	txTrafficDescType value	trafficShaping value	Type of traffic shaping applied for service categories
AQM-based, CQC-based, GQM-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, GQM-based	3, 4, 5, 6, 7, 8, or 9	enabled disabled	linear traffic shaping enabled for the connection
AQM-based, CQC-based, GQM-based	3, 4, 5, or 9	inverseUpc	linear traffic shaping enabled for the connection
(Sheet 1 of 2)			

Table 1 (continued)
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, GQM-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
(Sheet 2 of 2)			

Table 2
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
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

Configuring traffic management for service categories

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

Prerequisites

- See NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*.
- See NN10600-706 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*.
- See NN10600-707 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Queuing and Scheduling Fundamentals*.
- See NN10600-708 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM CAC and Bandwidth Fundamentals*.
- 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 104).

Procedure steps

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 <unshap>
```

6 Configure the fairness weighting.

```
set AtmIf/<n> Ca <ServiceCategory>/0 weightPolicy
<weight>
```

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

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

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.

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>
```

Variable definitions

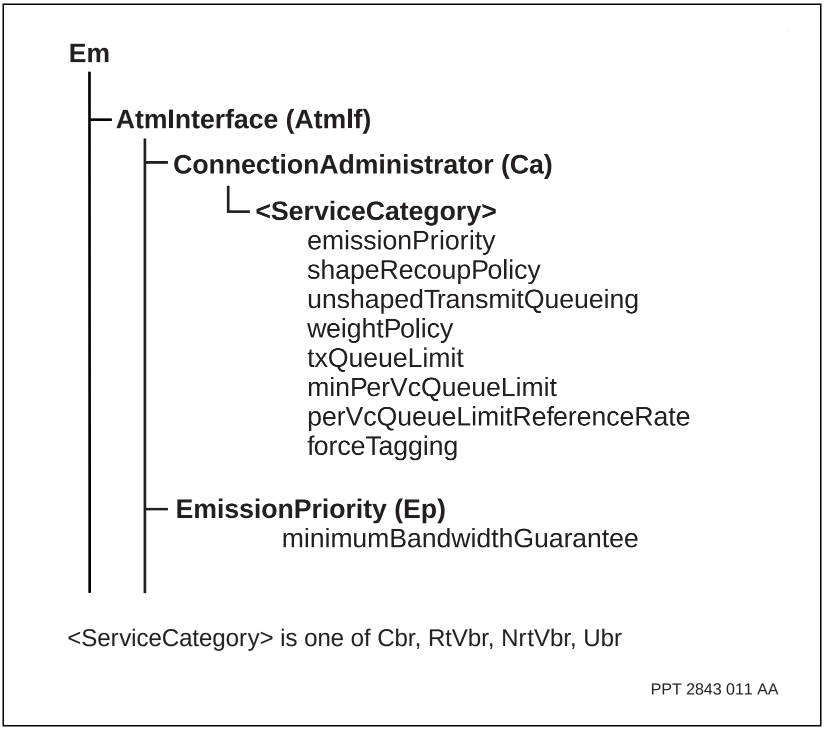
Variable	Value
<ep>	is a decimal value. For AQM-based FPs, the applicable range is 0 to 7 for all ATM service categories. For APC-based FPs, the values can be 0, 2, 3, 4, or 7. CQC-based FPs ignore the *emissionPriority* attribute. AQM default values: CBR 0; rt-VBR 1; nrt-VBR 4; UBR 7 APC default values: 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.
(Sheet 1 of 3)	

Variable	Value
<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 Networks Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*, the section on shaper congestion. Set the value to minimumCdv to turn the recoup policy off for connections under this service category.
<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 Networks 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 Networks 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*.
(Sheet 2 of 3)	

Variable	Value
<unshap>	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 Networks 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 “Interpretation of the value autoConfigure for the unshapedTransmitQueueing attribute” (page 105) 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 provisioned 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.
(Sheet 3 of 3)	

Procedure job aid

Figure 21
Traffic management for service categories component hierarchy



Configuring the ATM link transmit utilization alarm

Configure the ATM link transmit utilization alarm. If there is an outstanding alarm, it should clear immediately.

Procedure steps

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.

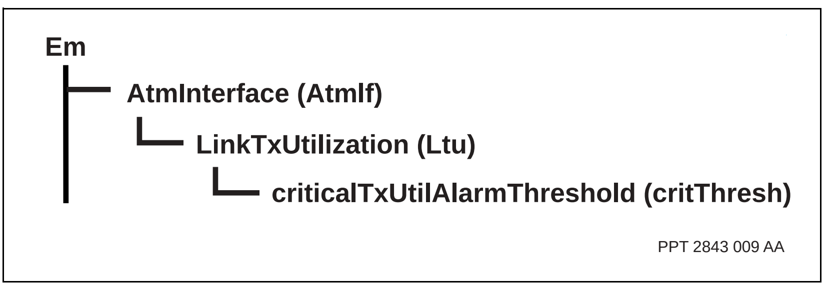
```
set AtmIf/<n> Ltu critThresh disabled
```

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

Figure 22
ATM link transmit utilization alarm component hierarchy



Specified path connection attributes

This section provides information about additional attributes that can impact the provisioning of specified path connections.

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.

> ***Note:*** 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 Networks 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 95)
- shapeRecoupPolicy described in the procedure “Configuring traffic management for service categories” (page 95)
- weightPolicy described in the procedure “Configuring traffic management for service categories” (page 95)

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

1 The EP in each service category must be in the following order: CBR <= rt-VBR <= nrt-VBR <= UBR.

2 Two different service categories can have the same EP value only if they are both shaped.

3 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.

4 If the rt-VBR service category is shaped, it must share the same EP as the CBR service categories.

5 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.

6 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.

7 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 105) when configuring the *unshapedTransmitQueueing* attribute in the procedure “Configuring traffic management for service categories” (page 95).

Table 3
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
(Sheet 1 of 2)		

Table 3 (continued)
Interpretation of the value autoConfigure for the unshapedTransmitQueueing attribute

Type of FP	Condition	Interpretation of value autoConfigure
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
(Sheet 2 of 2)		

Chapter 4
Non-switched connection provisioning

Provision 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 provisioning procedures

This task flow shows you the sequence of procedures you perform to provision non-switched connections. To link to any procedure, go to “Non-switched connection provisioning procedure navigation” (page 109).

Figure 23
Non-switched connection provisioning procedures

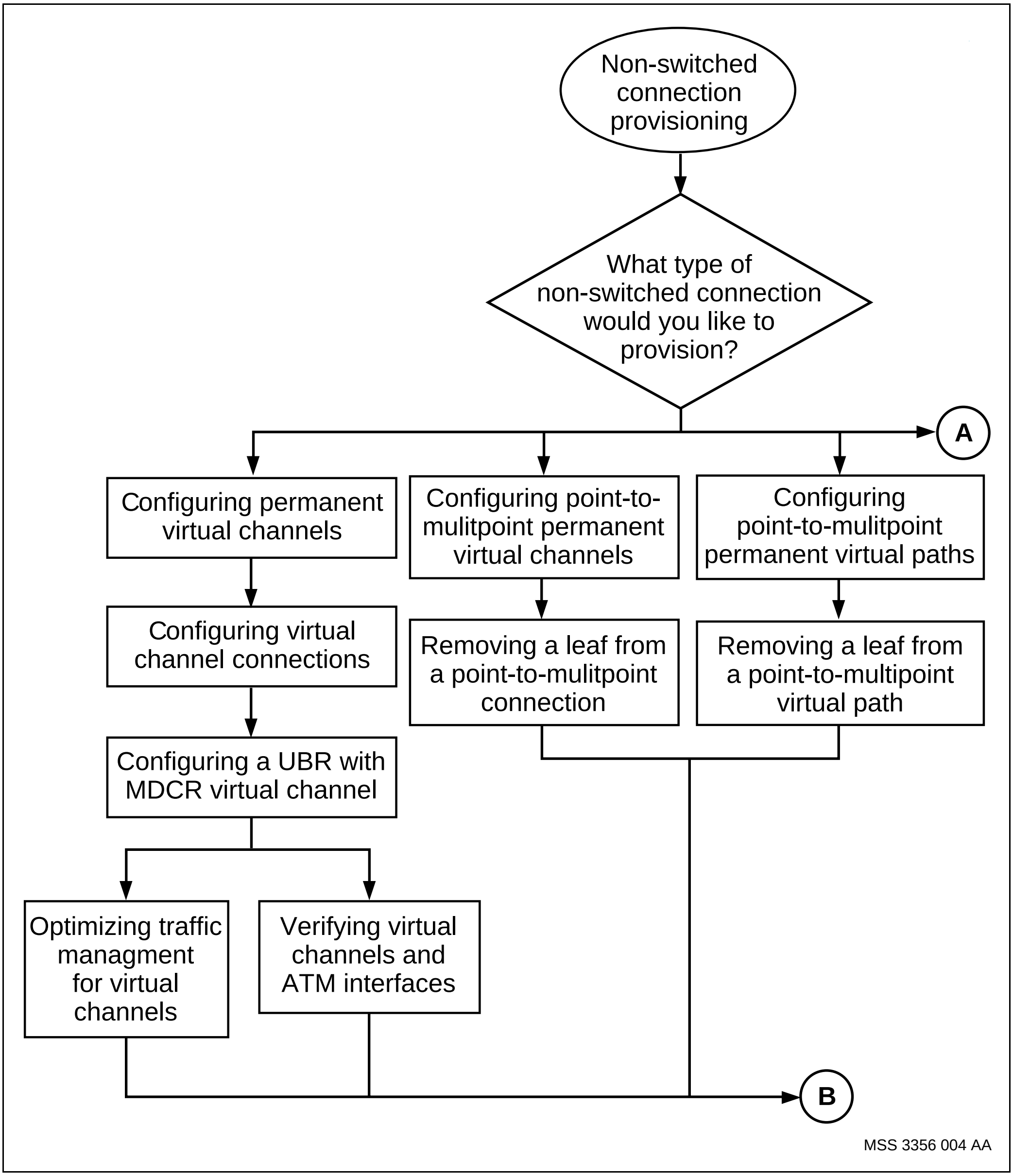
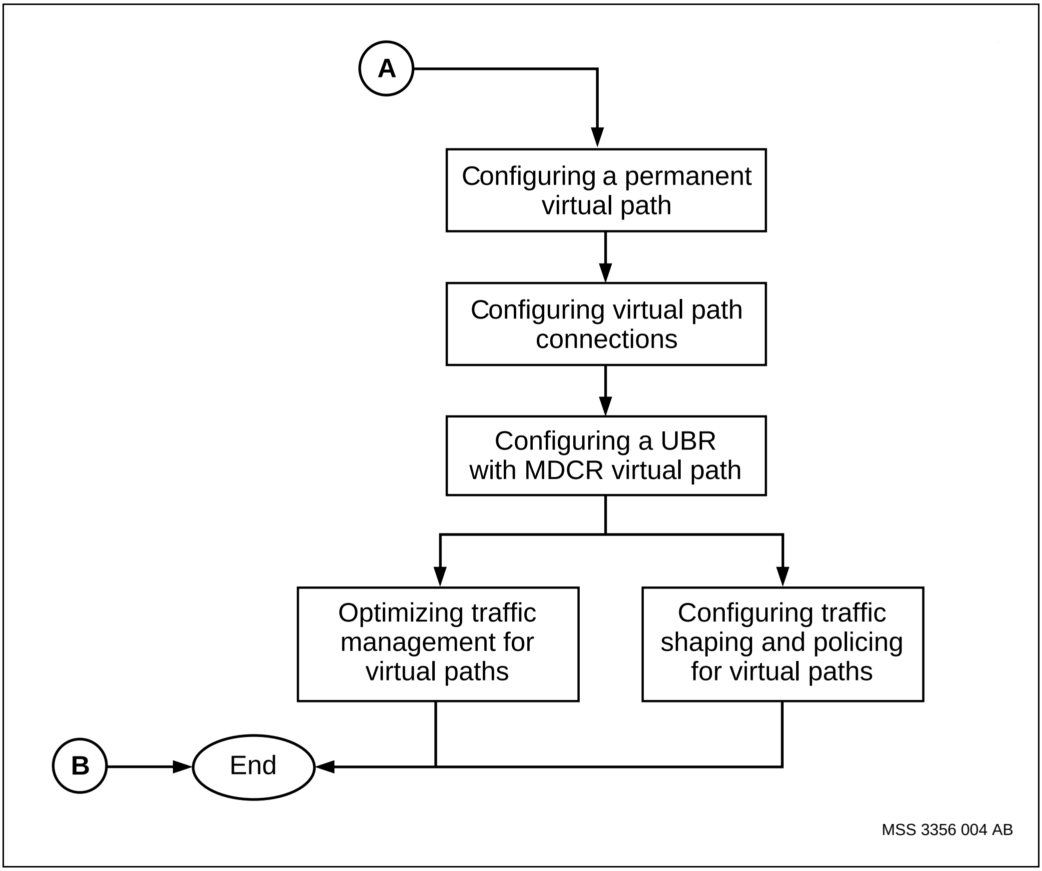


Figure 24
Non-switched connection provisioning procedures (continued)



Non-switched connection provisioning procedure navigation

- "Configuring permanent virtual channels" (page 111)
- "Configuring virtual channel connections" (page 114)
- "Configuring a UBR with MDCR virtual channel" (page 119)
- "Configuring traffic shaping and policing for virtual channels" (page 120)
- "Optimizing traffic management for virtual channels" (page 123)

Configuring permanent virtual channels

Configure permanent virtual channels (PVC) when channels that consist of predefined static routes that remain set up when not in use are required.

Prerequisites

- If you are associating the virtual channels used in this procedure with a virtual path terminator, you must configure a virtual path terminator (VPT) before you can add a virtual connection. See “Virtual path terminator provisioning” (page 321) and “Virtual interface provisioning” (page 341).

Procedure steps

1 Add two virtual channel connections. See “Configuring virtual channel connections” (page 114).

2 Add a *Nrp* component under the each Vcc component.

add AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Nrp

3 Link the two *Nrp* components together.

set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Nrp nextHop AtmIf/<n2> [Vpt/<Vpi>] Vcc/<x2> Nrp

4 Set the OAM segment boundary.

set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Nrp oamSegmentBoundary <sb>

5 Specify that the connection is bandwidth elastic or non-elastic. VPTs do not support bandwidth elastic connections. You must configure all VCCs in the VPT as bandwidth non-elastic connections.

set AtmIf/<n> Vcc/<x> Nrp bandwidthElastic <bwEl>

6 Specify the holding priority of the connection.

set AtmIf/<n> Vcc/<x> Nrp overrideHoldingPriority <ohpri>

7 Optionally, specify the circuit ID value for the PVC.

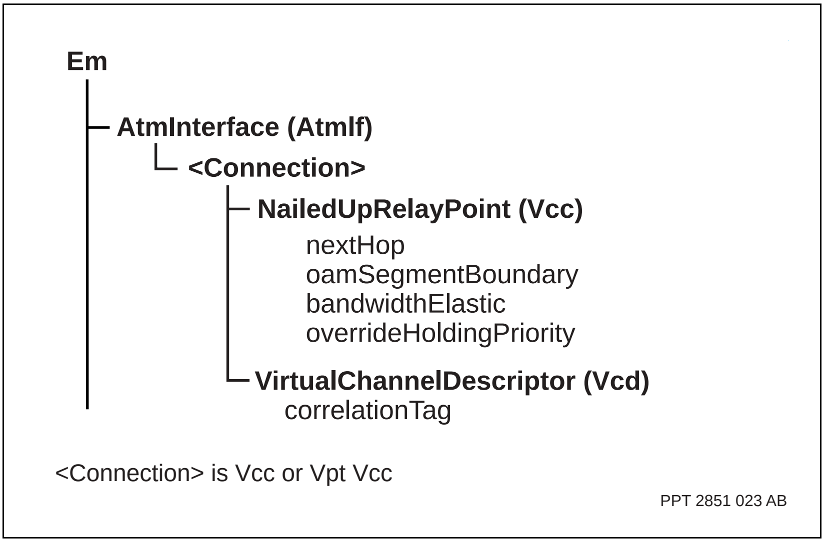
set AtmIf/<n> Vcc/<x> Vcd correlationTag <correlationTag>

Variable definitions

Variable	Value
<bwEl>	is yes or no. The default is no.
<correlationtag>	is the circuit ID for the PVC. This value can be any 64-byte string and is included in the ATM accounting record as the PVC identifier.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<ohpri>	is any number from 0 to 4 inclusive (0 is the highest priority and 4 is the lowest priority), or noOverride. The value provisioned for this attribute overrides the holding priority specified for each *Vcc* or *Vcd* component. If *noOverride* is selected, the holding priority specified for each *Vcc* or *Vcd* component is used.
<sb>	is yes, no, or sameAsInterface. The default is sameAsInterface. If the relay point is on a segment boundary, the value of the *connectionPointType* operation attribute for the connection is *segmentEndPoint*; if the relay point is not on a segment boundary, *connectionPointType* is *connectingPoint*.
<Vpi>	is the instance value of the *Vpt* component and can be any unique value from 0 to 4 095.
[Vpt/<Vpi>]	is the *VirtualPathTerminator* component. Use this parameter if you are associating the virtual connection with a virtual path terminator.
<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.

Procedure job aid

Figure 25
Permanent virtual channels component hierarchy



Configuring virtual channel connections

Configure virtual channel connections (VCC) between points in an ATM network where the adaption layer is accessed.

Prerequisites

- Add an ATM interface. See "Configuring an AtmIf" (page 72) for details. If you are associating the virtual channel with a virtual path terminator, you must configure a virtual path terminator. See the section "Virtual path terminator provisioning" (page 321) for information on configuring virtual path terminators.

Procedure steps

1 Add a virtual channel connection component.

```
add AtmIf/<n> [Vpt/<Vpi>] Vcc/<x>
```

2 Set the default loopbacks for the channel.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd
segLinkSideLoopback <segLkLbk>
```

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd
segSwitchSideLoopback <segSwLbk>
```

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd endToEndLoopback
<eeLbk>
```

3 Set the ATM service category for the channel.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
atmServiceCategory <atmS>
```

4 Set the traffic descriptor type for the transmit direction of the channel.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
txTrafficDescType <txTdt>
```

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

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
txTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>]
[4 <parm4>] [5 <parm5>]
```

6 Set the traffic descriptor type for the receive direction of the channel.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
rxTrafficDescType <rxTdt>
```

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

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
rxTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>]
[4 <parm4>]
```

Variable definitions

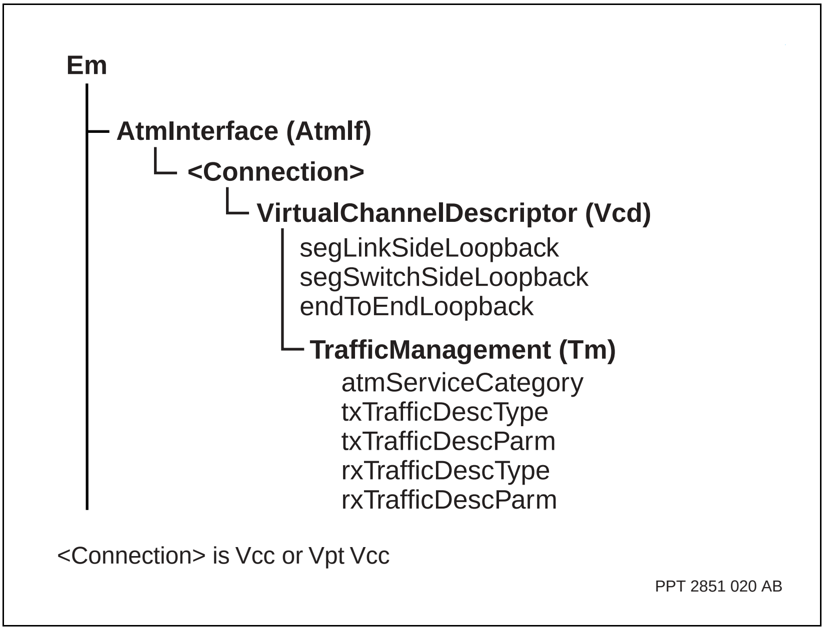
Variable	Value
<atmS>	is Cbr, RtVbr, NrtVbr, Ubr or derivedFromBbc. The default is Ubr. Set the value to derivedFromBbc if this channel is to become an SPVC. See "Optimizing traffic management capabilities for point-to-point SPVCs" (page 193) for further information.
<eeLbk>	specifies whether end-to-end 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.
<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.
(Sheet 1 of 3)	

Variable	Value
<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.
<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.
<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
<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.
(Sheet 2 of 3)	

Variable	Value
<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.
<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
<Vpi>	is the instance value of the *Vpt* component and can be any unique value from 0 to 4 095. This value depends on the value in the *maxVpiBits* attribute and the card type.
[Vpt/<Vpi>]	is the *VirtualPathTerminator* component. Use this parameter if you are associating the virtual connection with a virtual path terminator.
<x>	is the instance of the *Vcc* component. 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, which cannot be 3 or 4. The VCI values 3 and 4 are reserved in VPTs. VCI values below 32 are reserved for use by future standards.
(Sheet 3 of 3)	

Procedure job aid

Figure 26
Virtual channel connections component hierarchy



Configuring a UBR with MDCR virtual channel

Configure a UBR with MDCR virtual channel to use an explicit value for MDCR by using the traffic descriptor type 9 and setting the traffic descriptor parameters as described in "Configuring virtual channel connections" (page 114).

Alternatively, you can configure a UBR with MDCR virtual channel to use the default minimum cell rate for the UBR service category at an interface by using traffic descriptor type 1, 2, or 3 and setting the traffic descriptor parameters as described in "Configuring virtual channel connections" (page 114).

Configuring traffic shaping and policing for virtual channels

Configure traffic shaping and policing for virtual channels to smooth out traffic bursts by regulating the emission interval of cells or frames in the transmit direction.

> ***Note 1:*** These procedures do not apply to basic or standard VPT VCCs.
>
> ***Note 2:*** Traffic shaping is not possible for CBR connections on a CQC-based FP.

Prerequisites

- You must have traffic shaping and policing enabled at the ATM interface. See “Configuring traffic shaping and policing for service categories” (page 90) for information on enabling traffic shaping and policing.
- 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 93) when provisioning 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 93) when provisioning usage parameter control (UPC) for service categories.
- For information on traffic shaping and policing concepts, see NN10600-706 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*.

Procedure steps

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

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
trafficShaping <trshaping>
```

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

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
usageParameterControl <upc>
```

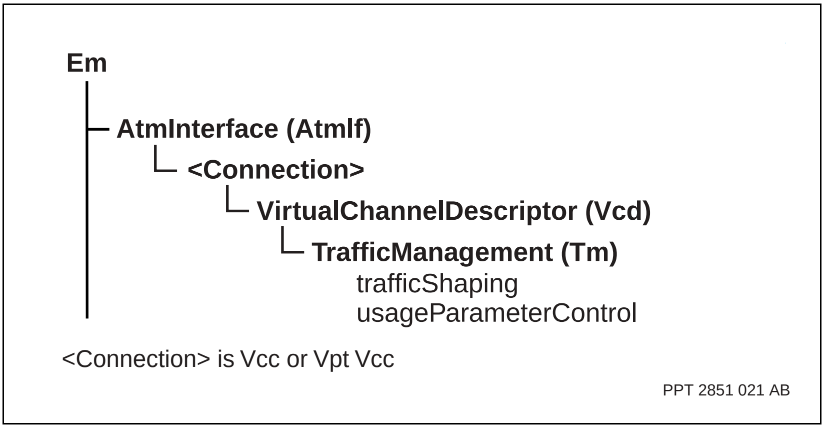
Variable definitions

Variable	Value
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<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 the txTrafficDescType and trafficShaping attribute values” (page 93) 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.
<upc>	is enforced, 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.
<Vpi>	is the instance value of the *Vpt* component and can be any unique value from 0 to 4 095.
(Sheet 1 of 2)	

Variable	Value
[Vpt/<Vpi>]	is the *VirtualPathTerminator* component. Use this parameter if you are configuring the virtual connection with a virtual path terminator.
<x>	is the instance of the virtual channel. 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.
(Sheet 2 of 2)	

Procedure job aid

Figure 27
Traffic shaping and policing for virtual channels component hierarchy



Optimizing traffic management for virtual channels

Optimize traffic management for virtual channels by changing the default traffic management parameters for the connection.

> ***Note:*** This procedure applies to virtual channels at the interface layer and virtual channels associated with a virtual path terminator. It does not apply to basic or standard VPT VCCs.

Prerequisites

- For information on traffic management concepts, see NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*.

Procedure steps

1 Set per-VC queuing for the channel.

set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm unshapedTransmitQueueing <unshap>

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

set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm txPacketWiseDiscard <txpd>

3 Optionally, for PVP and PVC connections on Nortel Networks 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> [Vpt/<Vpi>] Vcc/<x> Vcd Tm txWredMode <mode>

set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm txWredThreshold <threshold>

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

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

set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm rxPacketWiseDiscard <rxpd>

5 Set the holding priority for the channel.

```
set AtmIf/<n> Vcc/<x> Vcd Tm holdingPriority
<hPri>
```

6 Configure the fairness weighting for an unshaped channel.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm weight
<weight>
```

7 Configure the transmit queue limit for the channel.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm txQueueLimit
<txQlim>
```

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.

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

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm forceTagging
<tag>
```

Variable definitions

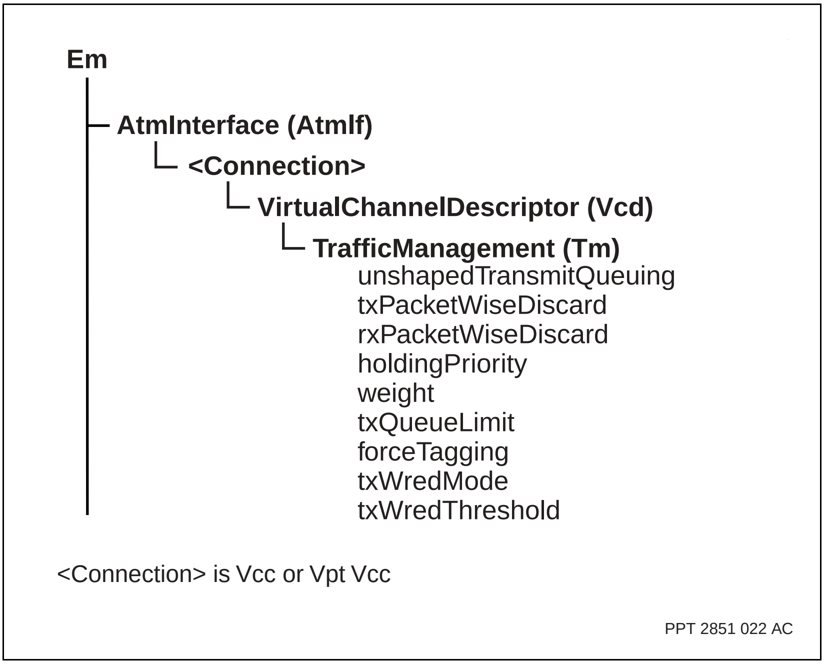
Variable	Value
<hPri>	is a decimal from 0 to 4. The default is 2. The decimal 0 indicate highest priority and the decimal 4 indicates lowest priority).
<mode>	is disabled, enabled, or perFlow. The default value is disabled.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<rxpd>	is disabled 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.
(Sheet 1 of 3)	

Variable	Value
<tag>	is sameAsCa, 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. ***Note:*** 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.
<txpd>	is disabled 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 sameAsCa 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.
<unshap>	is common 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.
<Vpi>	is the instance value of the *Vpt* component and can be any unique value from 0 to 4 095.
(Sheet 2 of 3)	

Variable	Value
[Vpt/<Vpi>]	is the *VirtualPathTerminator* component. Use this parameter if you are associating the virtual channel with a virtual path terminator. This parameter is not applicable to standard VPTs.
<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.
(Sheet 3 of 3)	

Procedure job aid

Figure 28
Traffic management for virtual channels component hierarchy



Configuring point-to-multipoint permanent virtual channels

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

Prerequisites

- You must be familiar with the information provided in "Configuring connection and buffer space for ATM IP FPs" (page 40).
- Add a virtual channel connection for the root side of the connection and one for each leaf connection. See "Configuring virtual channel connections" (page 114).
- For information about configuring point-to-multipoint connections using virtual path connections, see "Configuring point-to-multipoint permanent virtual paths" (page 145).
- For a listing of FPs providing point-to-multipoint connections capability, see NN10600-702 *Nortel Networks Multiservice Switch 7400/15000/ 20000 ATM Routing and Signalling Fundamentals*.
- For supporting information, see "Point-to-multipoint connections" (page 149).

Procedure steps

1 Add an *MNrp* component under the *Vcc* component for the root side of the connection.

```
add AtmIf/<n> Vcc/<x> MNrp
```

2 Set *mCastConnectionType* component under the *vcd* component.

```
set Atmif/<n> Vcc/<x> vcd mCastConnectionType
pointToMultipointRoot
```

3 Add a *Nrp* component under each *Vcc* component that will act as a leaf connection.

```
add AtmIf/<n> Vcc/<x> Nrp
```

4 Set *mCastConnectionType* component under each leaf's *vcd* component.

```
set AtmIf/<n> Vcc/<x> Vcd mCastConnectionType
pointtoMultipointLeaf
```

5 Link the root *MNrp* component to each leaf *Nrp* component.

`set AtmIf/<n> Vcc/<x> MNrp nextHop AtmIf/<n2> Vcc/<x2> Nrp`

6 Repeat step 5 for each leaf component.

7 Set the ATM service category for the connection.

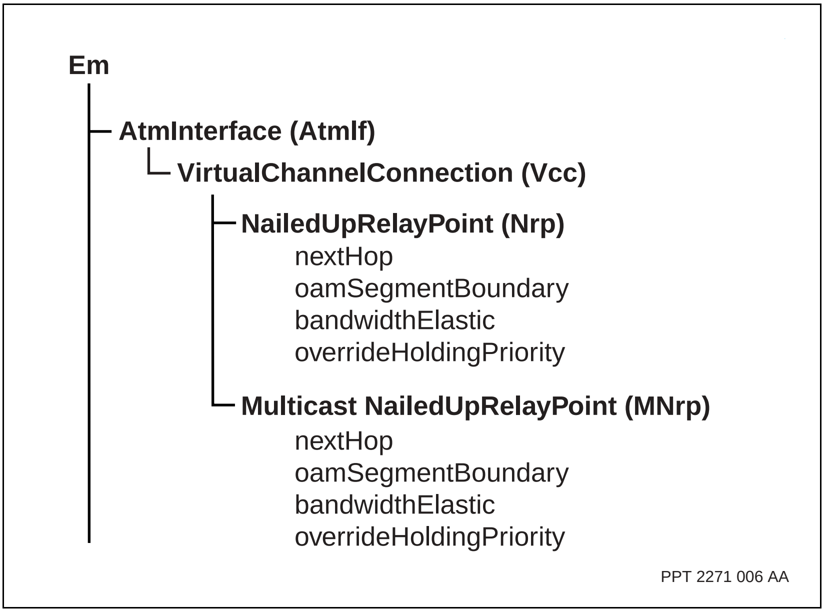
`set AtmIf/<n> Vcc/<r> Vcd Tm atmServiceCategory <atmS>`

Variable definitions

Variable	Value
<atmS>	is one of Cbr, RtVbr, NrtVbr, Ubr or derivedFromBbc. The default is Ubr.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<r>	is the instance of the root *Vcc*.
<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.

Procedure job aid

Figure 29
Point-to-multipoint permanent virtual channels component hierarchy



Removing a leaf from a point-to-multipoint connection

Remove a leaf from a point-to-multipoint connection to terminate one of the connections multiple end points. While using this procedure, none of the connections established between the root point and its leaves will be affected until after the final step is completed.

Procedure steps

1 Indicate that you wish to remove one or more leaf connections from a root point.

set AtmIf/<n> Vcc/<x> Mnrp nextHop !

2 Identify the leaf connections that you wish to remain established with the root point. The link established between the root point and any leaf points not included in this step are also removed.

set atmif/<n> Vcc/<x> MNrp nexthop atmif/<n2> Vcc/<x2> Nrp

3 Repeat step 2 for all leaf points you want to remain linked with the root.

Variable definitions

Variable	Value
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<n2>	is the instance value of the *AtmInterface* component of the leaf connection.
<x>	is the instance of the root *Vcc*. If the virtual channel is associated with an *AtmIf* component, <x> represents the VPI.VCI value.
<x2>	is the instance of the leaf *Vcc*. If the virtual channel is associated with an *AtmIf* component, <x> represents the VPI.VCI value.

Configuring a permanent virtual path

Configure a permanent virtual path (PVP) to establish a virtual path with endpoints which is configured by the network operator and remains in place even when it is not being used.

Procedure steps

1 Add two virtual path connections. See “Configuring virtual path connections” (page 134).

2 Add a *Nrp* component under the each Vpc component.

add AtmIf/<n> Vpc/<Vpi> Nrp

3 Link the two *Nrp* components together.

set AtmIf/<n1> Vpc/<Vpi1> Nrp nextHop AtmIf/<n2> Vpc/<Vpi2> Nrp

4 Set the OAM segment boundary.

set AtmIf/<n> Vpc/<Vpi> Nrp oamSegmentBoundary <sb>

5 Specify that the connection is bandwidth elastic or non-elastic.

set AtmIf/<n> Vpc/<Vpi1> Nrp bandwidthElastic <bwEl>

6 Specify the holding priority of the connection.

set AtmIf/<n> Vpc/<Vpi> Nrp overrideHoldingPriority <ohpri>

7 Optionally, specify the circuit ID value for the PVP.

set AtmIf/<n> Vpc/<Vpi> Vpd correlationTag <correlationtag>

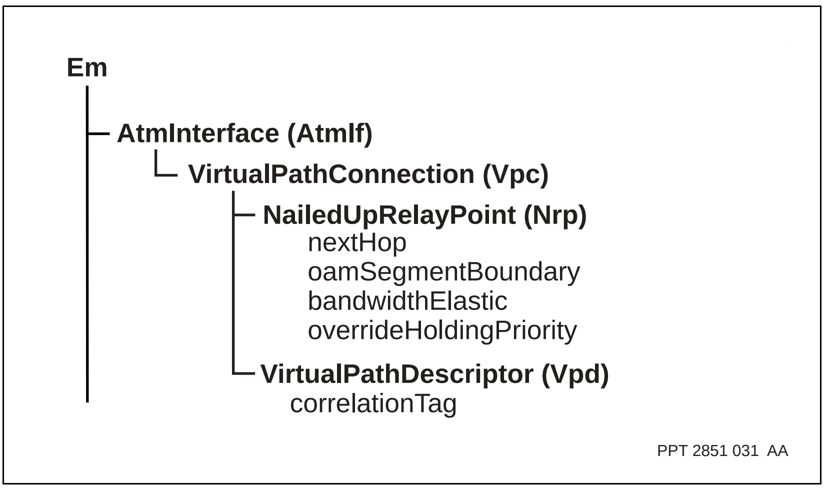
Variable definitions

Variable	Value
<bwEl>	is yes or no. The default is no.
<correlationTag>	is the circuit ID for the PVP. This value can be any 64-byte string and is included in the ATM accounting record as the PVP identifier.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
(Sheet 1 of 2)	

Variable	Value
<ohpri>	is any number from 0 to 4 inclusive (0 is the highest priority and 4 is the lowest priority), or noOverride. The value provisioned for this attribute overrides the holding priority specified for each *Vpd* component. If noOverride is selected, the holding priority specified for each *Vpd* component is used.
<sb>	is yes, no, or sameAsInterface. The default is sameAsInterface. If the relay point is on a segment boundary, the value of the *connectionPointType* operation attribute for the connection is *segmentEndPoint*; if the relay point is not on a segment boundary, *connectionPointType* is *connectingPoint*.
<Vpi>	is the VPI value.
(Sheet 2 of 2)	

Procedure job aid

Figure 30
Point-to-multipoint connection leaf component hierarchy



Configuring virtual path connections

Configure virtual path connections between points in an ATM network where the adaption layer is accessed.

Procedure steps

1 Add a virtual path component.

add AtmIf/<n> Vpc/<Vpi>

2 Set the default loopbacks for the path.

set AtmIf/<n> Vpc/<Vpi> Vpd segLinkSideLoopback <segLkLbk>

set AtmIf/<n> Vpc/<Vpi> Vpd segSwitchSideLoopback <segSwLbk>

set AtmIf/<n> Vpc/<Vpi> Vpd endToEndLoopback <eeLbk>

3 Set the *ATM service category* for this connection.

set AtmIf/<n> Vpc/<Vpi> Vpd Tm atmServiceCategory <atmS>

4 Set the traffic descriptor type for the transmit direction of the virtual path.

set AtmIf/<n> Vpc/<Vpi> Vpd Tm txTrafficDescType <txTdt>

5 Set the traffic descriptor parameters for the transmit direction of the virtual path.

set AtmIf/<n> Vpc/<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 virtual path.

set AtmIf/<n> Vpc/<Vpi> Vpd Tm rxTrafficDescType <rxTdt>

The default value is usually provisioned, which uses the traffic descriptor type and traffic descriptor parameters that have been provisioned for the transmit direction (that is, symmetrical traffic).

7 Set the traffic descriptor parameters for the receive direction of the virtual path. When rxTdt is 9 for UBR with MDCR, <parm1> is PCR, <parm2> is CDVT, and <parm3> is MDCR. If the setting for *rxTrafficDescType* is sameAsTx, the parameters set for *txTrafficDescParm* are used here.

```
set AtmIf/<n> Vpc/<Vpi> Vpd Tm rxTrafficDescParm [1
<parm1>] [2 <parm2>] [3 <parm3>] [4 <parm4>]
```

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.

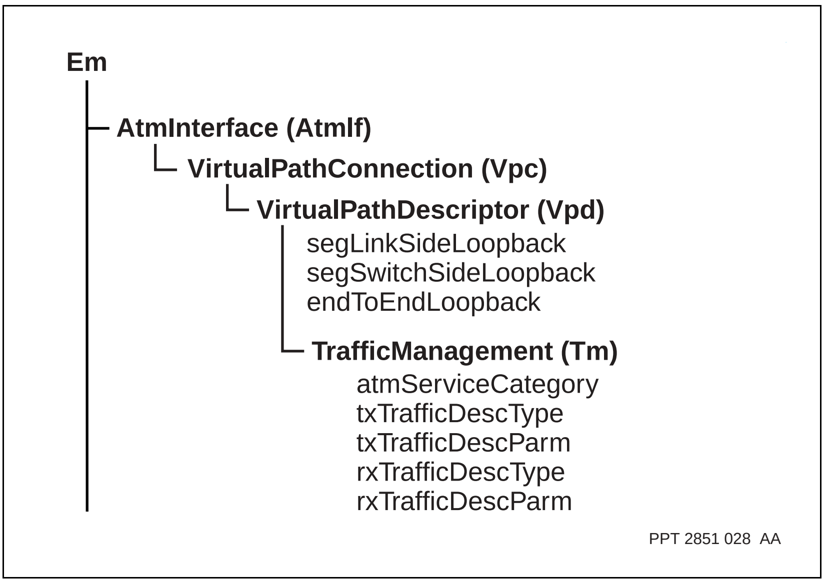
Variable definitions

Variable	Value
<atmS>	is Cbr, RtVbr, NrtVbr, or Ubr. The default is Ubr. Set the value to derivedFromBbc if this channel is to become an SPVP. See “Optimizing traffic management capabilities for point-to-point SPVCs” (page 193) for further information.
<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 *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 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 virtual path, 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. The default is 0.
<rxTdt>	is a value between 1 and 9, defining the traffic descriptor type. The default is 1. Traffic descriptor 9 is to be used exclusively by UBR with MDCR.
<segLkLbk>	specifies whether 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*.
(Sheet 1 of 2)	

Variable	Value
<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*.
<txTdt>	is a value between 1 and 9, defining the traffic descriptor type. The default is 1. Traffic descriptor 9 is to be used exclusively by UBR with MDCR.
<Vpi>	is the instance value of the virtual path. This value can be from 0 to 255.
(Sheet 2 of 2)	

Procedure job aid

Figure 31
Virtual path connections component hierarchy



Configuring a UBR with MDCR virtual path

Configure a UBR with MDCR virtual path to use an explicit value for MDCR by using the traffic descriptor type 9 and setting the traffic descriptor parameters as described in "Configuring virtual path connections" (page 134).

Alternatively, a UBR with MDCR virtual path may be configured to use the default minimum cell rate for the UBR service category at an interface by using traffic descriptor type 1, 2, or 3 and setting the traffic descriptor parameters as described in "Configuring virtual path connections" (page 134).

Configuring traffic shaping and policing for virtual paths

Configure traffic shaping and policing for virtual paths to smooth out traffic bursts by regulating the emission interval of cells or frames in the transmit direction.

Prerequisites

- You must have traffic shaping and policing enabled at the ATM interface. See the section “Configuring traffic shaping and policing for service categories” (page 90) for details.
- For information on traffic shaping and policing concepts, see NN10600-706 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*.

Procedure steps

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

set AtmIf/<n> Vpc/<Vpi> Vpd Tm trafficShaping <trShaping>

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

set AtmIf/<n> Vpc/<Vpi> Vpd Tm usageParameterControl <upc>

Variable definitions

Variable	Value
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<trShaping>	is either 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, software disables shaping for transmitted traffic on paths under that service category. The shaping characteristics that software derives from the ConnectionAdministrator component depends on the hardware configuration. See the table “Relationship between the txTrafficDescType and trafficShaping attribute values” (page 93) 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.
<upc>	is enforced, 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 *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. A value of *monitored* disables UPC since the node cannot monitor UPC violations on a CQC-based FP.
<Vpi>	is the instance value for the path.

Procedure job aid

Figure 32
Traffic shaping and policing for virtual paths component hierarchy

```
Em
 |
 +-- AtmInterface (AtmIf)
 |     └── VirtualPathConnection (Vpc)
 |           └── VirtualPathDescriptor (Vpd)
 |                 └── TrafficManangement (Tm)
 |                       trafficShaping
 |                       usageParameterControl
```

PPT 2851 029 AA

Optimizing traffic management for virtual paths

Optimize traffic management for virtual paths by changing the default traffic management parameters for the connection.

Prerequisites

- Configure a virtual path connection according to the procedure “Configuring virtual path connections” (page 134).
- For information on traffic management concepts, see NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*.

Procedure steps

1 Set per-VC queuing for the path.

set AtmIf/<n> Vpc/<Vpi> Vpd Tm unshapedTransmitQueueing <unshap>

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

set AtmIf/<n> Vpc/<Vpi> Vpd Tm txPacketWiseDiscard <txpd>

3 Optionally, for Nortel Networks 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> Vpc/<Vpi> Vpd Tm txWredMode <mode>

set AtmIf/<n> Vpc/<x> Vpd Tm txWredThreshold <threshold>

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

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

set AtmIf/<n> Vpc/<Vpi> Vpd Tm rxPacketWiseDiscard <rxpd>

5 Set the holding priority for the path.

set AtmIf/<n> Vpc/<Vpi> Vpd Tm holdingPriority <hpri>

6 Configure the fairness weighting for an unshaped path.

set AtmIf/<n> Vpc/<Vpi> Vpd Tm weight <weight>

7 Configure the transmit queue limit for the path.

```
set AtmIf/<n> Vpc/<Vpi> Vpd Tm txQueueLimit <txqlim>
```

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.

8 Set the default value for cell loss priority for cells in the transmit direction.

```
set AtmIf/<n> Vpc/<Vpi> Vpd Tm forceTagging <tag>
```

Variable definitions

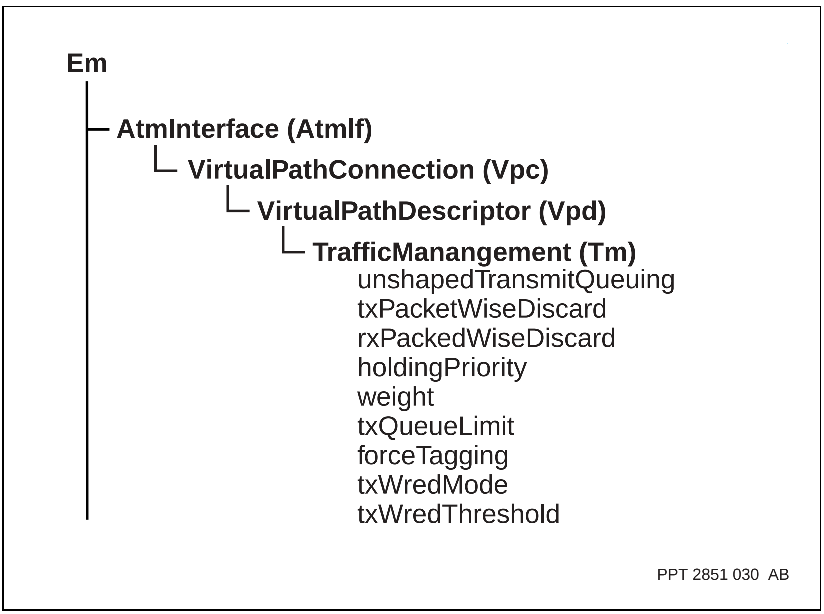
Variable	Value
<hpri>	is a decimal from 0 to 4. The default is 2. The value 0 indicates highest priority and the value 4 indicates lowest priority).
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<mode>	is disabled, enabled, or perFlow. The default value is disabled.
<rxpd>	is disabled or enabled. For GQM-based FPs, rxpd must be *disabled*.
(Sheet 1 of 3)	

Variable	Value
<tag>	is sameAsCa, 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. This value does not apply to interfaces on CQC-based FPs. • 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. ***Note:*** 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.
<txpd>	is disabled or enabled.
<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 92 to 10 240. The default is sameAsCa. 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.
<unshap>	is either *common* 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 IF 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.
(Sheet 2 of 3)	

Variable	Value
<Vpi>	is the instance of the *Vpc*.
<weight>	is a decimal value from 1 to 4 095, or 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.
(Sheet 3 of 3)	

Procedure job aid

Figure 33
Traffic management capabilities for virtual paths component hierarchy



Configuring point-to-multipoint permanent virtual paths

Configure point-to-multipoint permanent virtual paths (PVP) to establish a virtual path with one source and many destinations.

Prerequisites

- See NN10600-702 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals* for a listing of FPs providing point-to-multipoint connections capability.
- Add a virtual path connection for the root side of the connection and one for each leaf connection. See “Configuring a permanent virtual path” (page 132).
- For information about configuring point-to-multipoint over switched virtual connections, see “Configuring point-to-multipoint permanent virtual channels” (page 128).

Procedure steps

1 Add an *MNrp* component under the Vpc component for the root side of the connection.

add AtmIf/<n> Vpc/<Vpi> Mnrp

2 Add a *Nrp* component under each Vpc component that will act as a leaf connection.

add AtmIf/<n> Vpc/<Vpi> Nrp

3 Link the root *MNrp* component to each leaf *Nrp* component.

set AtmIf/<n> Vpc/<Vpi> MNrp nextHop AtmIf/<n2> Vpc/<vpi2> Nrp

4 Repeat step 3 for each leaf component.

5 Set the ATM service category for the connection.

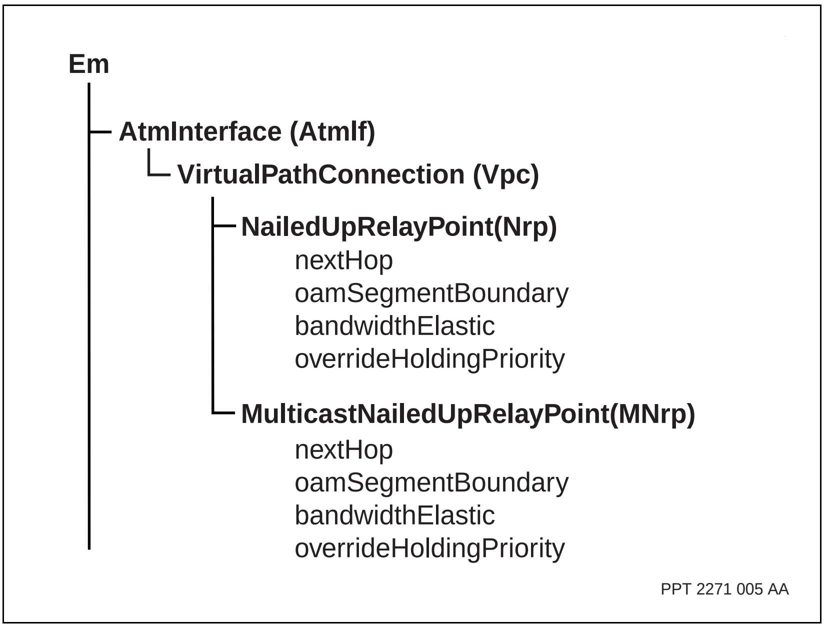
set AtmIf/<n> Vpc/<Vpi> Vpd Tm atmServiceCategory <atmServiceCategory>

Variable definitions

Variable	Value
<atmServiceCategory>	is one of Cbr, RtVbr, NrtVbr, Ubr or derivedFromBbc. The default is Ubr.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<vpi>	is the VPI value.

Procedure job aid

Figure 34
Point-to-multipoint permanent virtual paths component hierarchy



Removing a leaf from a point-to-multipoint virtual path

Remove a leaf from a point-to-multipoint virtual path to terminate one of the paths multiple end points. While using this procedure, none of the paths established between the root point and its leaves will be affected until after the final step is completed.

Procedure steps

1 Indicate that you wish to remove one or more leaf connections from a root point.

```
set AtmIf/<n> Vpc/<vpi> Mnrp nextHop !
```

2 Identify the leaf connections that you wish to remain established with the root point.

```
set atmif/<n> Vpc/<vpi> MNrp nexthop Atmif/<n2> Vpc/
<vpi2> Nrp
```

The link established between the root point and any leaf points not included in this step will be removed.

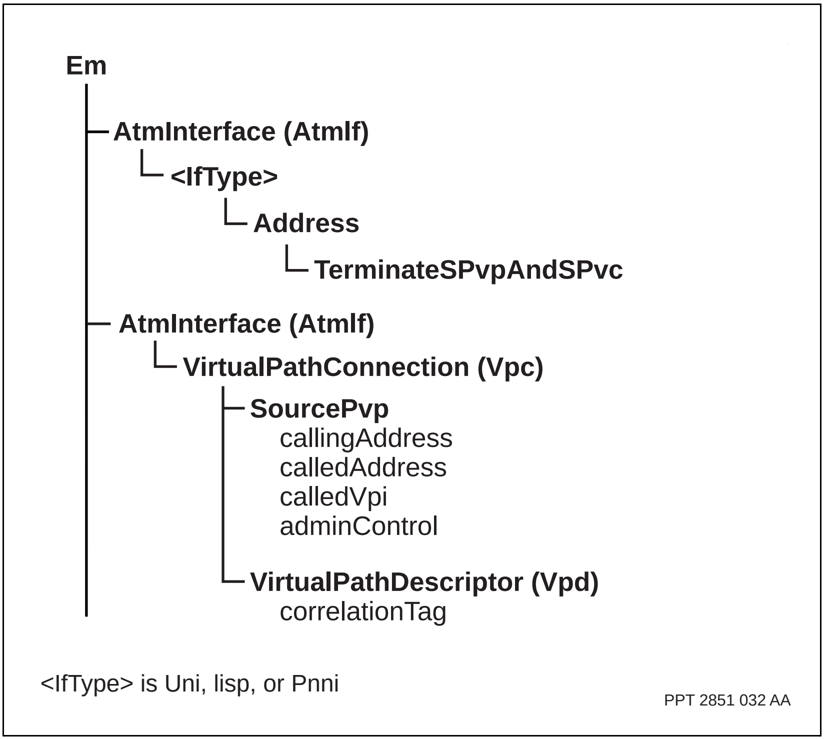
3 Repeat step 2 for all leaf points you want to remain linked with the root.

Variable definitions

Variable	Value
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<n2>	is the instance value of the *AtmInterface* component of the leaf connection.
<vpi>	is the VPI value.
<vpi2>	is the leaf *Vpi* value.

Procedure job aid

Figure 35
Point-to-multipoint virtual path leaf component hierarchy



Point-to-multipoint connections

Point-to-multipoint connections consists of one root connection point represented by the multicast nailed-up relay point (MNRP), which can be linked to multiple nailed-up relay points (NRP) and multiple leaf connection points represented by the multiple branches defined by the *nextHop* attribute.

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 150) for more information about multicast branch calculations.

Example of a multicast branch calculation

Figure “Point-to-multipoint connections” (page 151), displays some point-to-multipoint connections on a Nortel Networks Multiservice Switch 7400 node, and figure “Component tree for point-to-multipoint connections” (page 152) displays the component tree of these point-to-multi point connections.

Figure 36
Point-to-multipoint connections

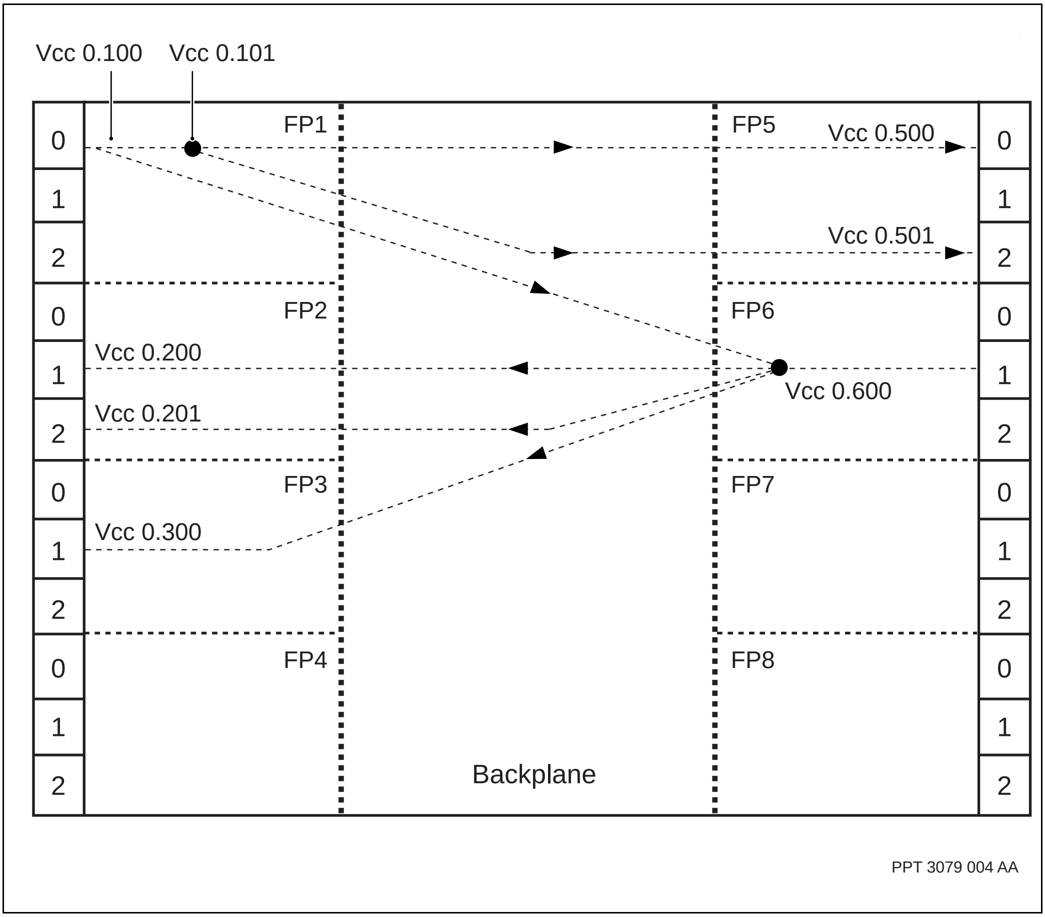
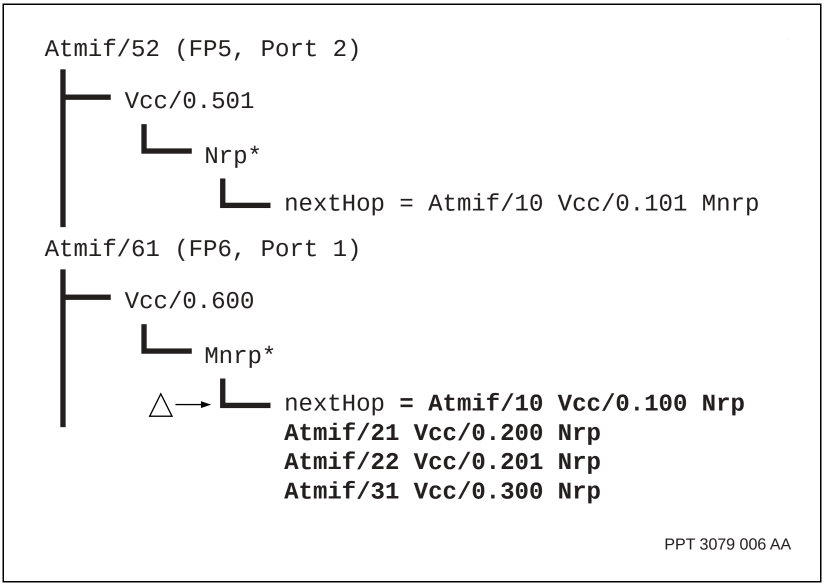


Figure 37
Component tree for point-to-multipoint connections

```
Atmif/10 (FP1, Port 0)
  ├── Vcc/0.101
  │     └── Mnrp*
  │           └── nextHop = Atmif/50 Vcc/0.500 Nrp
  │                         Atmif/52 Vcc/0.501 Nrp
  ├── Vcc/0.101
  │     └── Nrp*
  │           └── nextHop = Atmif/61 Vcc/0.600 Mnrp
Atmif/21 (FP2, Port 1)
  ├── Vcc/0.200
  │     └── Nrp*
  │           └── nextHop = Atmif/61 Vcc/0.600 Mnrp
Atmif/22 (FP2, Port 2)
  ├── Vcc/0.201
  │     └── Nrp*
  │           └── nextHop = Atmif/61 Vcc/0.600 Mnrp
Atmif/31 (FP3, Port 1)
  ├── Vcc/0.300
  │     └── Nrp*
  │           └── nextHop = Atmif/61 Vcc/0.600 Mnrp
Atmif/50 (FP5, Port 0)
  ├── Vcc/0.500
  │     └── Nrp*
  │           └── nextHop = Atmif/10 Vcc/0.101 Mnrp
                   continued...
```

PPT 3079 005 AA

Figure 38
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 AtmIfs, count the entries indicated by a plus (+) sign in figure “Component tree for point-to-multipoint connections” (page 152):

- *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 152):

- *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*)

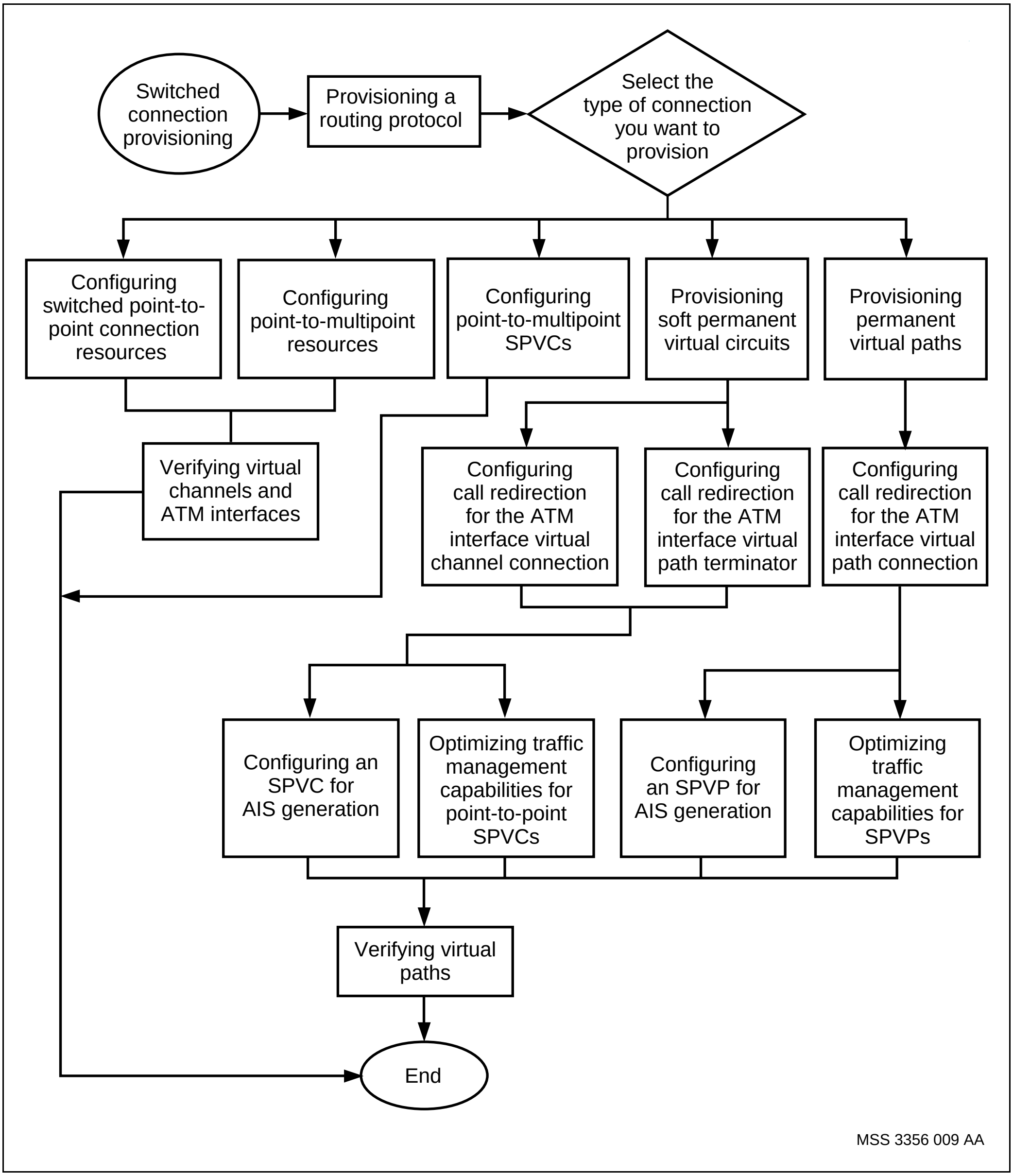
Chapter 5
Switched connection provisioning

Provision virtual connections that are established and torn down by subscriber applications on an as-needed basis. These logical connections remain in place only for the duration of the data transfer.

Switched connection provisioning procedures

This task flow shows you the sequence of procedures you perform to provision switched connections. To link to any procedure, go to "Switched connection provisioning procedure navigation" (page 157).

Figure 39
Switched connection provisioning procedures



Switched connection provisioning procedure navigation

Provisioning a routing protocol

Provision a routing protocol that will be used on the switched connection. See “Routing protocol provisioning” (page 203).

Configuring switched point-to-point connection resources

Configure switched point-to-point connection resources to set the parameters to dynamically establish and terminate connections on a per call basis.

Procedure steps

1 Configure the VPI range for SVPs.

set AtmIf/<n> Ca minAutoSelectedVpi <minVpi>

set AtmIf/<n> Ca maxAutoSelectedVpi <maxVpi>

2 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.

3 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>

4 Set the permitted ATM service categories.

set AtmIf/<n> Ca permittedSvcAtmServiceCategories <ServiceCategory>

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

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

Variable definitions

Variable	Value
<hPri>	is the default holding priority for the specified traffic type, and can be any number from 0 (highest priority) to 4 (lowest priority).
<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.
<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.
<ServiceCategory>	is Ubr, Cbr, RtVbr, or NrtVbr. The default is all service categories.
<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.

Procedure job aid

Figure 40
Switched point-to-point connection resources component hierarchy

- **Em**
  - **AtmInterface (AtmIf)**
    - **ConnectionAdministrator (Ca)**
      - minAutoSelectedVpi
      - minAutoSelectedVciForVpiZero
      - minAutoSelectedVciForNonZeroVpi
      - maxAutoSelectedVpi
      - maxAutoSelectedVciForVpiZero
      - maxAutoSelectedVciForNonZeroVpi
      - permittedAtmServiceCategories
    - **<ServiceCategory>**
      - holdingPriority

<ServiceCategory> is Cbr, RtVbr, NrtVbr, Ubr

PPT 2851 011 AB

Configuring point-to-multipoint resources

Configure point-to-multipoint resources that are necessary for point-to-multipoint connections.

Prerequisites

- Configure point-to-point resources as shown in procedure “Configuring point-to-multipoint SPVCs” (page 165).

Procedure steps

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

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

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

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

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

3 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.

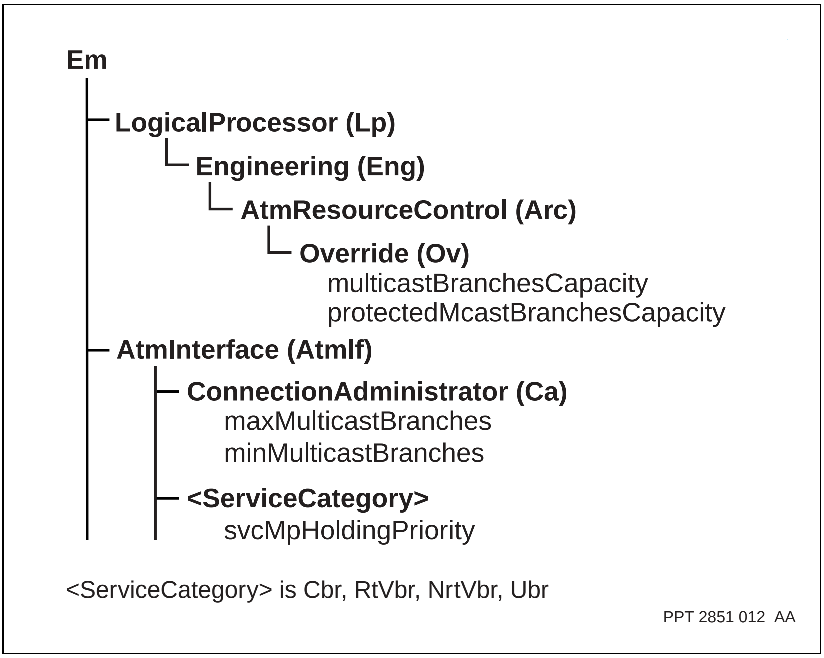
```
set AtmIf/<n> Ca <ServiceCategory>/0
svcMpHoldingPriority <svcMpHrpi>
```

Variable definitions

Variable	Value
<lp>	is the number of the LP associated with the ATM card.
<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).
<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.
<ServiceCategory>	is Cbr, RtVbr, NrtVbr, or Ubr. Only the 0 instance for the 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).

Procedure job aid

Figure 41
Point-to-multipoint resources component hierarchy



Configuring point-to-multipoint SPVCs

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

Procedure steps

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

1. Add a virtual channel connection. See “Configuring virtual channel connections” (page 114).
2. Configure the necessary point-to-multipoint resources. See the procedure “Configuring point-to-multipoint resources” (page 162) for details.
3. Add the *MulticastSourcePvc* component.

```
add AtmIf/<n> vcc/<vcc> msrc
```

4. Set the calling address of the *MulticastSourcePvc* component.

```
set Atmif/<n> vcc/<vcc> msrc callingAddress <addr>
```

5. Set the *mCastConnectionType* attribute for the new vcc.

```
set atmif/x vcc/<vcc> Vcd mCastConnectionType pmpRoot
```

6. Add the *party* subcomponents.

```
add AtmIf/<n> vcc/<vcc> msrc party/<m>
```

7. Set the called address.

```
set atmif/<n> vcc/<vcc> msrc party/<m> calledAddress
<addr_Leaf>
```

8. Set the calledVpiVci.

```
set atmif/<n> vcc/<vcc> msrc party/<m> calledVpiVci
<x1.y1>
```

9. Repeat steps 6, 7, and 8 for each required party. When provisioning multiple parties, no two parties from the same msrc can have an identical combination of *calledaddress* and *calledvpivci* attributes.

10 Set the ATM service category for the channel.

```
set AtmIf/<n> Vcc/<x> Vcd Tm atmServiceCategory <atmS>
```

11 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> Vcc/<x> Vcd Tm txTrafficDescType <txTdt>
```

12 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> Vcc/<x> Vcd Tm txTrafficDescParm [1
<parm1>] [2 <parm2>] [3 <parm3>] [4 <parm4>] [5
<parm5>]
```

13 Set the traffic descriptor type for the receive direction of the channel.

```
set AtmIf/<n> Vcc/<x> Vcd Tm rxTrafficDescType <rxTdt>
```

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

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

15 Set the usage parameter control.

```
set atmif/n vcc/x.y vcd tm upc <upc>
```

16 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 vcc/x.y vcd tm bwdQosClass 0
```

17 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 vcc/x.y vcd tm bwdQosParameters cdv
16777215 clr 255
```

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

```
set atmif/n vcc/x.y vcd tm txPacketWiseDiscard
disabled
```

By default, the txPacketWiseDiscard attribute is disabled.

Variable definitions

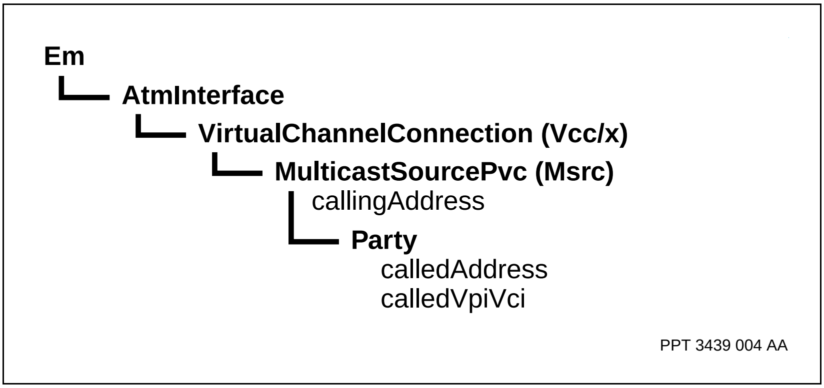
Variable	Value
<addr>	is the address associated with the source interface, and consists of 40 hexadecimal digits. If you do not specify an address value, the node uses the default address value.
<addr_Leaf>	is an address associated with the destination interface, consisting of 40 hexadecimal digits.
<atmS>	is Cbr, Rtvbr, Nrtvbr, or Ubr. The default is Ubr.
<m>	is a value between 1 and 300.
<multicastBranchesCapacity>	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 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.

(Sheet 1 of 2)

Variable	Value
<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.
<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.
<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.
<x2.y2>	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.
(Sheet 2 of 2)	

Procedure job aid

Figure 42
Point-to-multipoint SPVCs component hierarchy



Provisioning soft permanent virtual circuits

Provision soft permanent virtual circuits to configure a switched connection with the same functionality as a permanent virtual connection without the need to manually provision each node along the route.

Prerequisites

- Before completing this procedure, consider the following points:
  - At the source end of the connection, provision 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 Networks 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 provisioning, FP restarts, and software reloads occur.
- Use one of the following procedures to define the ATM interface on the source node (the node that originates the call setup request):
  - "Configuring a user-to-network connection" (page 208) to define a UNI
  - "Configuring an inter-switch protocol interface" (page 213) to define an IISP
  - "Configuring an ATM inter-network interface" (page 216) to define an AINI
  - "Configuring a PNNI" (page 219) to define a PNNI

Procedure steps

1 Add a virtual channel connection. See "Configuring virtual channel connections" (page 114).

2. 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
   ```

3. 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>
   ```

4. Define this Vcc as the source end of an SPVC

   ```
   add AtmIf/<n> [Vpt/<Vpi>] Vcc/<Vpi.Vci> SourcePvc
   ```

5. Identify the source end for this SPVC by specifying its address.

   ```
   set AtmIf/<n> [Vpt/<Vpi>] Vcc/<Vpi.Vci> SourcePvc
   callingAddress <callingA>
   ```

6. Identify the destination end for this SPVC by specifying its address.

   ```
   set AtmIf/<n> [Vpt/<Vpi>] Vcc/<Vpi.Vci> SourcePvc
   calledAddress <calledA>
   ```

7. Identify the VPI.VCI associated with the called (or destination) interface.

   ```
   set AtmIf/<n> [Vpt/<Vpi>] Vcc/<Vpi.Vci> SourcePvc
   calledVpiVci <calledVpiVci>
   ```

8. Set the traffic descriptor type for the transmit direction of the channel.

   ```
   set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
   txTrafficDescType <txTdt>
   ```

9. Set the traffic descriptor parameters for the transmit direction of the channel.

   ```
   set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
   txTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>]
   [4 <parm4>] [5 <parm5>]
   ```

10. Set the traffic descriptor type for the receive direction of the channel.

    ```
    set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
    rxTrafficDescType <rxTdt>
    ```

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

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
rxTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>]
[4 <parm4>]
```

12 Optionally, set the initial state of the connection.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<Vpi.Vci> SourcePvc
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.

13 Optionally, specify the circuit ID value for the SPVC.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<Vpi.Vci> Vcd
correlationTag <correlationTag>
```

Variable definitions

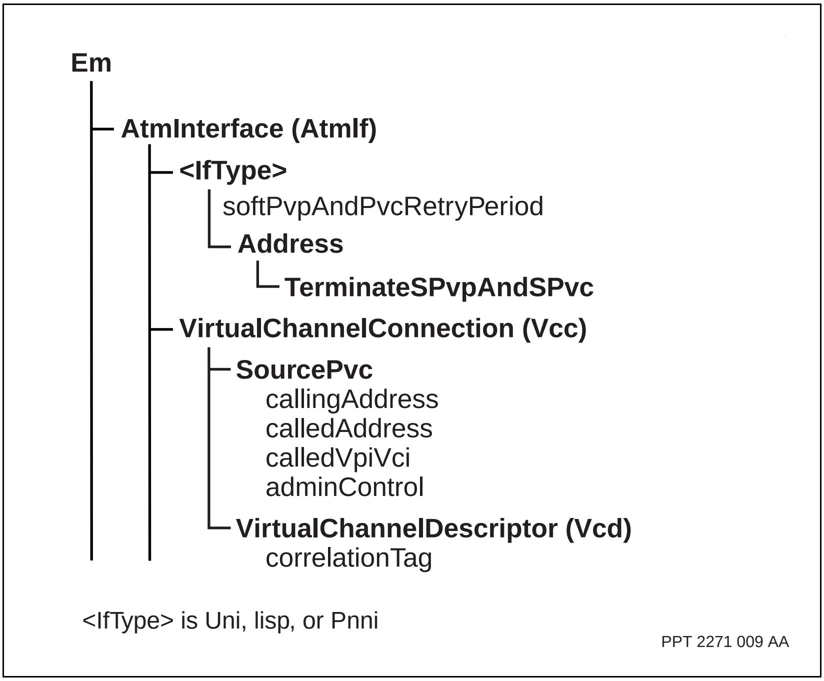
Variable	Value
<addr_type>	is primary or alternate.
<address>	is a static address associated with the interface.
<adminControl>	is the initial state of the connection. The value can be permlocked or unlocked. Permlocked means that the connection is always locked and remains locked even after the FP restarts and the software reloads. Unlocked means that the connection is initially unlocked and remains unlocked even after the FP restarts and the software reloads.
<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.
<correlationtag>	is the circuit ID for the SPVC. This value can be any 64-byte string and is included in the ATM accounting record as the SPVC identifier.
(Sheet 1 of 3)	

Variable	Value
<IfType>	is uni, Iisp, Aini, or Pnni.
<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 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 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.
(Sheet 2 of 3)	

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 provisioned value to 40 s; if you enter 34, the system sets the provisioned value to 30 s. When provisioning a large number of SPVCs on the node, it is important that all of the interfaces have different softPvpAndPvcRetryPeriod attributes. These attributes should be provisioned such that they can not be rounded up or down to the nearest multiple of 10. For example, provision 20, 30, and 70, instead of 21, 22, and 24. This provisioning process enables the timing of the call set-up retries to be spread out to ease congestion on the device.
<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.
<Vpi.Vci>	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.
[Vpt/<Vpi>]	is the *VirtualPathTerminator* component. Use this parameter if you are associating the virtual connection with a virtual path terminator.
(Sheet 3 of 3)	

Procedure job aid

Figure 43
Soft permanent virtual circuits component hierarchy



Provisioning soft permanent virtual paths

Provision soft permanent virtual paths (SPVP) to configure a type of virtual path that allows configured SPVP endpoints in different frame relay or ATM networks to establish a switched connection across network-to-network interfaces.

Prerequisites

- At the source end of the connection, provision a *SourcePvp* component under a *Vpc* component. The *SourcePvp* component specifies the destination's address and its VPI instance in the destination connection map address space. A *Vpd* component defines the connection's traffic requirements.
- Consider the following points before following this procedure:
  - As network routing and signaling establishes the SPVP connection across the network, the nodes along the connection route create dynamic *Vpc* and *RelayPoint* components. If the destination of the SPVP is on a Nortel Networks Multiservice Switch node, the system creates a *DestinationPvp* component under the *Vpc* component on the destination node.
  - The SPVP lock and unlock capability introduces the *adminControl* attribute under the *SourcePVP* component that enables the operator to specify the initial state of the connection after initial provisioning, FP restarts, and software reloads occur.
- Use one of the following procedures to define the ATM interface on the source node (the node that originates the call setup request):
  - "Configuring a user-to-network connection" (page 208) to define a UNI
  - "Configuring an inter-switch protocol interface" (page 213) to define an IISP
  - "Configuring an ATM inter-network interface" (page 216) to define an AINI
  - "Configuring a PNNI" (page 219) to define a PNNI

Procedure steps

1 Add a virtual path connection. See “Configuring virtual path connections” (page 134).

2 On the destination node, add a static address under the appropriate interface and label it as an address that can be used to terminate SPVPs.

```
add AtmIf/<n> <IfType> Address/<address>,<addr_type>
TerminateSpvpAndSpvc
```

3 On the source node, specify how often an SPVP attempts to set up a call after the initial call setup fails.

```
set AtmIf/<n> <IfType> softPvpAndPvcRetryPeriod
<seconds>
```

4 Define this virtual path as the source end of an SPVP

```
add AtmIf/<n> Vpc/<Vpi> SourcePvp
```

5 Identify the source end for this SPVP by specifying its address.

```
set AtmIf/<n> Vpc/<Vpi> SourcePvp callingAddress
<callingA>
```

6 Identify the destination end for this SPVP by specifying its address.

```
set AtmIf/<n> Vpc/<Vpi> SourcePvp calledAddress
<calledA>
```

7 Identify the VPI associated with the called (or destination) interface.

```
set AtmIf/<n> Vpc/<Vpi> SourcePvp calledVpi
<calledVpi>
```

8 Set the traffic descriptor type for the transmit direction of the path.

```
set AtmIf/<n> [Vpc/<Vpi>] Vcc/<x> Vcd Tm
txTrafficDescType <txTdt>
```

9 Set the traffic descriptor parameters for the transmit direction of the path.

```
set AtmIf/<n> [Vpc/<Vpi>] Vcc/<x> Vcd Tm
txTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>]
[4 <parm4>] [5 <parm5>]
```

10 Set the traffic descriptor type for the receive direction of the path.

```
set AtmIf/<n> [Vpc/<Vpi>] Vcc/<x> Vcd Tm
rxTrafficDescType <rxTdt>
```

11 Set the traffic descriptor parameters for the receive direction of the path.

set AtmIf/<n> [Vpc/<Vpi>] Vcc/<x> Vcd Tm rxTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>] [4 <parm4>]

12 Optionally, set the initial state of the connection.

set AtmIf/<n> Vpc/<Vpi> SourcePvc adminControl <admincontrol>

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

13 Optionally, specify the circuit ID value for the SPVP.

set AtmIf/<n> Vpc/<Vpi> Vpd correlationTag <correlationtag>

Variable definitions

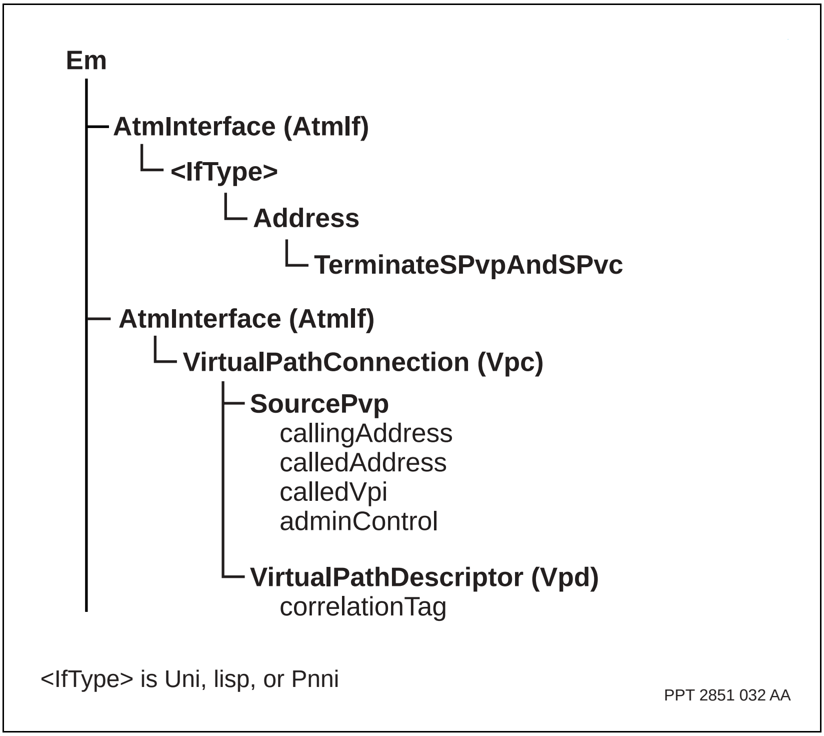
Variable	Value
<addr_type>	is primary or alternate.
<address>	is a static address associated with the interface.
<adminControl>	is the initial state of the connection. The value can be permlocked or unlocked. Permlocked means that the connection is always locked and remains locked even after the FP restarts and the software reloads. Unlocked means that the connection is initially unlocked and remains unlocked even after the FP restarts and the software reloads.
<calledA>	is a default address associated with the destination interface, consisting of 40 hexadecimal digits.
<calledVpi>	is the instance value of the virtual path associated with the remote end-point. The VPI value can be from 0 to 4 095.
<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.
<correlationTag>	is the circuit ID for the SPVP. This value can be any 64-byte string and is included in the ATM accounting record as the SPVP identifier.
<IfType>	is uni, Iisp, Aini, or Pnni.
<parm1>	is traffic descriptor parameter 1 (peak cell rate) between 0 and 2 147 483 647. The default is 0.
(Sheet 1 of 2)	

Variable	Value
<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.
<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 provisioned value to 40 s; if you enter 34, it sets the provisioned value to 30 s.
<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.
<Vpi>	is the instance value of the Vpc. The VPI value can be from 0 to 255.

(Sheet 2 of 2)

Procedure job aid

Figure 44
Soft permanent virtual paths component hierarchy



Configuring a UBR with MDCR virtual channel

Configure a UBR with MDCR virtual channel to use an explicit value for MDCR by using the traffic descriptor type 9 and setting the traffic descriptor parameters as described in “Configuring virtual channel connections” (page 114).

Alternatively, you can configure a UBR with MDCR virtual channel to use the default minimum cell rate for the UBR service category at an interface by using traffic descriptor type 1, 2, or 3 and setting the traffic descriptor parameters as described in “Configuring virtual channel connections” (page 114).

Configuring call redirection for the ATM interface virtual channel connection

Configure call redirection for the ATM interface virtual channel connection (AtmIf vcc src) to add a secondary destination address to be used if the primary destination connection fails.

After the call redirection for the ATM interface virtual channel 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

1 Add the *SourcePvc* component to the connection, if necessary.

add Atmif/<n> vcc/<vpi.vci> SourcePvc

2 Add the *CallRedirection* component.

add Atmif/<n> vcc/<vpi.vci> SourcePvc CallRedirection

3 Set the secondary called address.

set Atmif/<n> vcc/<vpi.vci> SourcePvc CallRedirection secondaryCalledAddress <sec_address>

4 Set the secondary called vpi.vci.

set Atmif/<n> vcc/<vpi.vci> SourcePvc CallRedirection secondaryCalledVpiVci <sec_vpi.sec_vci>

5 Complete configuration changes. See “Activating configuration changes” (page 28).

Variable definitions

Variable	Value
<n>	is the instance value of the Atmif component.
<sec_address>	is the value of the secondary called address attribute.
(Sheet 1 of 2)	

Variable	Value
<sec_vpi.sec_vci>	is the value of the secondary called vpi.vci.
<vpi.vci>	is the instance value of the vcc.
(Sheet 2 of 2)	

Procedure job aid

Figure 45
Configuring call redirection for Atmif vcc component hierarchy

```
Em
└── AtmInterface (AtmIf)
    └── VirtualChannelConnection (Vcc)
        └── SourcePvc (Src)
            └── CallRedirection
                  secondaryCalledAddress
                  secondaryCalledVpiVci
```

PPT 3470 003 AA

Configuring call redirection for the ATM interface virtual path connection

Configure call redirection for the ATM interface virtual path connection (AtmIf vpc src) 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 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

1 Add the *SourcePvc* component to the connection, if necessary.

```
add Atmif/<n> vpc/<vpi> SourcePvc
```

2 Add the *CallRedirection* component.

```
add Atmif/<n> vpc/<vpi> SourcePvc CallRedirection
```

3 Set the secondary called address.

```
set Atmif/<n> vpc/<vpi> SourcePvc CallRedirection
secondaryCalledAddress <sec_address>
```

4 Set the secondary called vpi.

```
set Atmif/<n> vpc/<vpi> SourcePvc CallRedirection
secondaryCalledVpi <sec_vpi>
```

5 Complete configuration changes. See “Activating configuration changes” (page 28).

Variable definitions

Variable	Value
<n>	is the instance value of the Atmif component.
<sec_address>	is the value of the secondary called address attribute.
(Sheet 1 of 2)	

Variable	Value
<sec_vpi>	is the value of the secondary called vpi.
<vpi>	is the instance value of the vpc.
(Sheet 2 of 2)	

Procedure job aid

Figure 46
Configuring call redirection for Atmif vpc component hierarchy

```
Em
└── AtmInterface (AtmIf)
     └── VirtualPathConnection (Vpc)
          └── SourcePvc (Src)
               └── CallRedirection
                     secondaryCalledAddress
                     secondaryCalledVpi
```

PPT 3470 001 AA

Configuring call redirection for the ATM interface virtual path terminator

Configure call redirection for the ATM interface virtual path terminator (AtmIf vpt vcc src) 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

1 Add the *SourcePvc* component to the connection, if necessary.

```
add Atmif/<n> vpt/<vpi> vcc/<vci> SourcePvc
```

2 Add the *CallRedirection* component.

```
add Atmif/<n> vpt/<vpi> vcc/<vci> SourcePvc
CallRedirection
```

3 Set the secondary called address.

```
set Atmif/<n> vpt/<vpi> vcc/<vci> SourcePvc
CallRedirection secondaryCalledAddress <sec_address>
```

4 Set the secondary called vpi.

```
set Atmif/<n> vpt/<vpi> vcc/<vci> SourcePvc
CallRedirection secondaryCalledVpiVci
<sec_vpi.sec_vci>
```

5 Complete configuration changes. See “Activating configuration changes” (page 28).

Variable definitions

Variable	Value
<n>	is the instance value of the Atmif component.
<sec_address>	is the value of the secondary called address attribute.
(Sheet 1 of 2)	

Variable	Value
<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.
(Sheet 2 of 2)	

Procedure job aid

Figure 47
Configuring call redirection for Atmif vpt component hierarchy

- **Em**
  - **AtmInterface (AtmIf)**
    - **VirtualPathTerminator (Vpt)**
      - **VirtualChannelConnection (Vcc)**
        - **SourcePvc (Src)**
          - **CallRedirection**
            secondaryCalledAddress
            secondaryCalledVpiVci

PPT 3470 002 AA

Optimizing traffic management capabilities for SPVPs

Optimize traffic management capabilities for SPVPs by changing the default traffic management parameters.

Procedure steps

1 Set the ATM service category to be derivedFromBbc for this connection.

```
set AtmIf/<n> Vpc/<Vpi> Vpd Tm atmServiceCategory
derivedFromBbc
```

2 Set the bearer capability for the connection.

```
set AtmIf/<n> Vpc/<Vpi> Vcd Tm bearerClassBBC <bcBBC>
```

3 Set the transfer capability for the connection.

```
set AtmIf/<n> Vpc/<Vpi> Vcd Tm transferCapabilityBBC
<tcBBC>
```

4 Set the clipping susceptibility for the connection.

```
set AtmIf/<n> Vpc/<Vpi> Vcd Tm clippingBBC
<clippingBBC>
```

5 Set the value for the best effort parameter in the ATM traffic descriptor information element.

```
set AtmIf/<n> Vpc/<Vpi> Vcd Tm bestEffort <bestEffort>
```

6 Set the elements that specify the class of service for the transmit direction of the connection.

```
set AtmIf/<n> Vpc/<Vpi> Vpd Tm fwdQosClass <fwdQos>
```

7 Set the elements that specify the quality of service parameters for the transmit direction of the connection.

```
set AtmIf/n> Vpc/<Vpi> Vpd Tm fwdQosParameters cdv
<cdvValue> ctd <ctdValue> clr <clrValue>
```

8 Set the elements that specify the class of service for the receive direction of the connection.

```
set AtmIf/<n> Vpc/<Vpi> Vpd Tm bwdQosClass <bwdQos>
```

9 Set the elements that specify the quality of service elements for the receive direction of the connection.

```
set AtmIf/n> Vpc/<Vpi> Vpd Tm bwdQosParameters cdv
<cdvValue> clr <clrValue>
```

Variable definitions

Variable	Value
<bcBBC>	is a, c, x, or derivedFromServiceCategory. The default is derivedFromServiceCategory.
<bestEffort>	is indicated, notIndicated, or derivedFromServiceCategory. The default is derivedFromServiceCategory.
<bwdQos>	is a value of 0, 1, 2, 3, 4, or sameAsFwd. The default is sameAsFwd.
<cdvValue>	is the acceptable peak-to-peak cell delay variation of CBR and RT-VBR connections.The value is a decimal between 0 and 16 777 215. The default value is 16 777 215.
<clippingBBC>	is yes or no. The default is no.
<clrValue>	is the acceptable cell loss ratio of CBR and RT-VBR connections. The value is a decimal between 1 and 15 or 255. The default value is 255.
<ctdValue>	is the acceptable maximum cell transfer delay of CBR and RT-VBR connections. The value is a decimal between 0 and 16 777 215. The default value is 16 777 215.
<fwdQos>	is a value of 0, 1, 2, 3, or 4. The default is 0.
<tcBBC>	is a value of 0, 1, 2, 5, 8, 9, 10, notApplicable, and derivedFromServiceCategory. The default is derivedFromServiceCategory. This attribute represents the combination of traffic type and timing requirements for the BBC IE.
<Vpi>	is the instance value of the Vpc. The VPI value can be from 0 to 255.

Procedure job aid

Figure 48
Traffic management capabilities for SPVPs component hierarchy

```
Em
 ├─AtmInterface (AtmIf)
 │  ├─VirtualPathConnection (Vpc)
 │  │    └─VirtualPathDescriptor (Vpd)
 │  │         └─TrafficManagement (Tm)
 │  │               atmServiceCategory
 │  │               bearerClassBbc
 │  │               transferCapabilityBbc
 │  │               clippingBbc
 │  │               bestEffort
 │  │               fwdQosClass
 │  │               fwdQosParameters
 │  │               bwdQosClass
 │  │               bwdQosParameters
```

PPT 2851 033 AA

Configuring an SPVP for AIS generation

Configure an SPVP for alarm indication signal (AIS) generation.

Prerequisites

- Before AIS generation can be enabled at the destination, a destination SPVP component must first be configured.
- 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.
- To enable AIS generation at both ends, both the source and destination must be enabled.
- To disable AIS generation at both ends, both the source and destination must be disabled.

Procedure steps

1 Add a destination SPVP.

add atmif/<n> vpc/<vpi> dst

2 Configure the traffic requirements. See the procedure "Optimizing traffic management capabilities for SPVPs" (page 188) for details. For the *trafficShaping* and *usageParameterControl* parameters, these can be independently enabled or disabled on the source and/or destination ends of an SPVC or SPVP.

3 Enable AIS at the source SPVP.

set atmif/<n> vpc/<vpi> src aisGeneration enable

4 Enable AIS at the destination SPVP.

set atmif/<n> vpc/<vpi> dst config aisGeneration enable

5 Disable AIS at the source SPVP.

set atmif/<n> vpc/<vpi> src aisGeneration disable

6 Disable AIS at the destination SPVP.

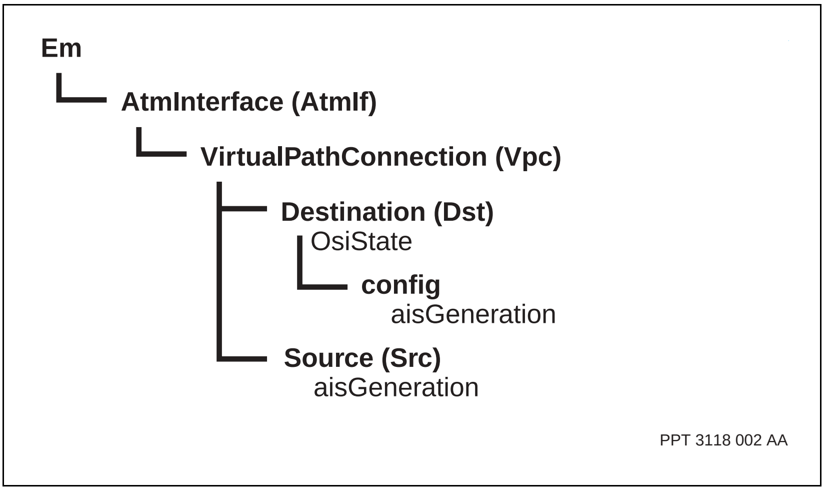
```
set atmif/<n> vpc/<vpi> dst config aisGeneration
disable
```

Variable definitions

Variable	Value
<n>	is the *AtmIf* component instance value.
<Vpi>	is the instance value of the VPC. This value can be from 0 to 255.

Procedure job aid

Figure 49
SPVP for AIS generation component hierarchy



Optimizing traffic management capabilities for point-to-point SPVCs

Optimize traffic management capabilities for point-to-point SPVCs to change the default management parameters for these connections.

Procedure steps

1 Set the ATM service category to be derivedFromBbc for this connection.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm
atmServiceCategory derivedFromBbc
```

2 Set the bearer capability for the connection.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<Vpi.Vci> Vcd Tm
bearerClassBBC <bcBBC>
```

3 Set the transfer capability for the connection.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<Vpi.Vci> Vcd Tm
transferCapabilityBBC <tcBBC>
```

4 Set the clipping susceptibility for the connection.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<Vpi.Vci> Vcd Tm
clippingBBC <clippingBBC>
```

5 Set the value for the best effort parameter in the ATM traffic descriptor information element.

```
set AtmIf/<n> Vcc/<Vpi.Vci> Vcd Tm bestEffort
<bestEffort>
```

6 Set the elements that specify the service parameters for the class of service for the transmit direction of the connection.

```
set AtmIf/<n> Vcc/<Vpi.Vci> Vcd Tm fwdQosClass
<fwdQos>
```

7 Set the elements that specify the service parameters for the quality of service for the transmit direction of the connection.

```
set AtmIf/n> Vcc/<Vpi.Vci> Vcd Tm fwdQosParameters cdv
<cdvValue> ctd <ctdValue> clr <clrValue>
```

8 Set the elements that specify the class of service parameters for the receive direction of the connection.

```
set AtmIf/<n> Vcc/<Vpi.Vci> Vcd Tm bwdQosClass
<bwdQos>
```

9 Set the elements that specify the class of service parameters for the receive direction of the connection.

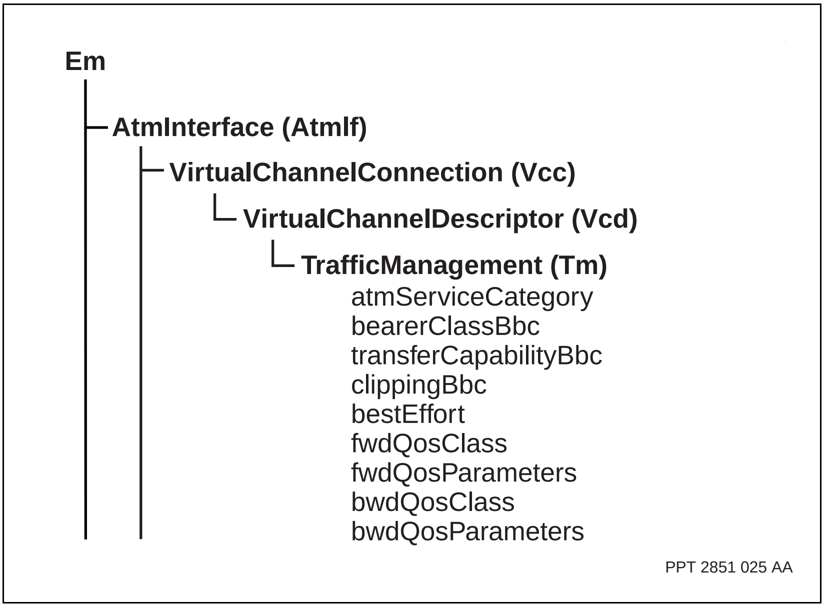
`set AtmIf/n> [Vpt/<Vpi>] Vcc/<Vpi.Vci> Vcd Tm bwdQosParameters cdv <cdvValue> clr <clrValue>`

Variable definitions

Variable	Value
<bcBBC>	is a, c, x, or derivedFromServiceCategory. The default is derivedFromServiceCategory.
<bestEffort>	is indicated, notIndicated, or derivedFromServiceCategory. The default is derivedFromServiceCategory.
<bwdQos>	is a value of 0, 1, 2, 3, 4, or sameAsFwd. The default is sameAsFwd.
<cdvValue>	is the acceptable peak-to-peak cell delay variation of CBR and RT-VBR connections.The value is a decimal between 0 and 16 777 215. The default value is 16 777 215.
<clippingBBC>	is yes or no. The default is no.
<clrValue>	is the acceptable cell loss ratio of CBR and RT-VBR connections. The value is a decimal between 1 and 15 or 255. The default value is 255.
<ctdValue>	is the acceptable maximum cell transfer delay of CBR and RT-VBR connections. The value is a decimal between 0 and 16 777 215. The default value is 16 777 215.
<fwdQos>	is a value of 0, 1, 2, 3, or 4. The default is 0.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<tcBBC>	is a value of 0, 1, 2, 5, 8, 9, 10, notApplicable, and derivedFromServiceCategory. The default is derivedFromServiceCategory. This attribute represents the combination of traffic type and timing requirements for the BBC IE.
[Vpt/<Vpi>]	is the *VirtualPathTerminator* component. Use this parameter if you are associating the virtual connection with a virtual path terminator.
<x>	is the instance of the *Vcc* component. 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.

Procedure job aid

Figure 50
Traffic management capabilities for SPVCs component hierarchy



Configuring an SPVC for AIS generation

Configure an SPVC to generate alarm indication signals.

> ***Note:*** A provisioned Dst PVC can only be used to terminate a point-to-point SPVC.

Prerequisites

- Before AIS generation can be enabled at the destination end, a destination SPVC component must first be configured.
- 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.
- Configure an SPVC. See the procedure “Provisioning soft permanent virtual circuits” (page 170) for details.

Procedure steps

1 Add a destination SPVC.

```
add atmif/<n> vcc/<vpi.vci> dst
```

2 Configure the traffic requirements. See the procedure “Optimizing traffic management capabilities for point-to-point SPVCs” (page 193) for details. For the *trafficShaping* and *usageParameterControl* parameters, these can be independently enabled or disabled on the source and/or destination ends of an SPVC or SPVP.

3 Enable AIS at the source SPVC.

```
set atmif/<n> vcc/<vpi.vci> src aisGeneration enable
```

4 Enable AIS at the destination SPVC.

```
set atmif/<n> vcc/<vpi.vci> dst config aisGeneration
enable
```

5 Disable AIS at the source SPVC.

```
set atmif/<n> vcc/<vpi.vci> src aisGeneration disable
```

6 Disable AIS at the destination SPVC.

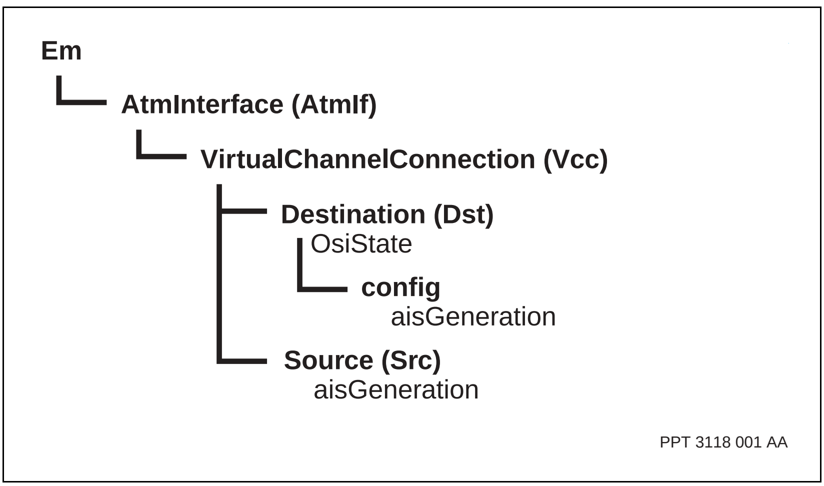
```
set atmif/<n> vcc/<vpi.vci> dst config aisGeneration
disable
```

Variable definitions

Variable	Value
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<Vpi.Vci>	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

Figure 51
SPVC for AIS generation component hierarchy



Verifying virtual paths

Verify virtual paths by using the loop service to verify the integrity of the hardware connections. This is done by transmitting test traffic and returning it to the originating connection component where it is verified against what was originally transmitted.

The *Loop* component configures an ATM connection so that cells received from the ATM interface are retransmitted back to the interface through the connection. The *Loop* component requires exclusive use of its parent *Vpc* component and cannot share its parent with other ATM service components (such as *Nep*, *Nrp*, and *Test*).

Procedure steps

1 Add a virtual path connection. See “Provisioning soft permanent virtual paths” (page 176) or “Configuring virtual path connections” (page 134).

2 Add *Loop* component to the virtual channel.

```
add AtmIf/<n> Vpc/<Vpi> Loop
```

3 As an alternative to provisioning a *Loop* component, you may also add an *NailedUpRelayPoint (Nrp)* component and then set the next hop to itself.

```
add AtmIf/<n> Vpc/<Vpi> Nrp
```

4 Set the next hop to itself.

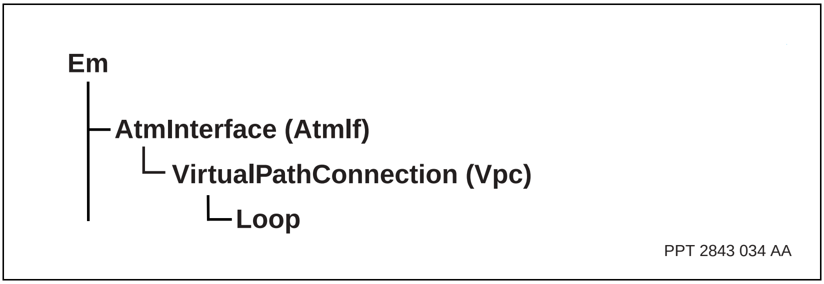
```
set AtmIf/<n> Vpc/<Vpi> Nrp nextHop AtmIf/<n> Vpc/
<Vpi> Nrp
```

Variable definitions

Variable	Value
<n>	is the *AtmIf* component instance value.
<Vpi>	is the instance value of the VPC. This value can be from 0 to 255.

Procedure job aid

Figure 52
Virtual paths component hierarchy



Verifying virtual channels and ATM interfaces

Verify virtual channels and ATM interfaces using the test and loop service to verify the integrity of the hardware connections. This is done by transmitting test traffic and returning it to the originating connection component where it is verified against what was originally transmitted.

The *Test* component allows test traffic to be sent over an ATM connection. It is used to verify the integrity of hardware carrying ATM traffic, and provides test capabilities similar to those provided by non-ATM port and line tests. The *Test* component requires exclusive use of its parent *Vcc* component and cannot share its parent with other ATM service components (such as *Nep*, *Nrp*, and *Loop*).

The *Loop* component configures an ATM connection so that cells received from the ATM interface are retransmitted back to the interface through the connection. The *Loop* component requires exclusive use of its parent *Vcc* component and cannot share its parent with other ATM service components (such as *Nep*, *Nrp*, and *Test*).

This procedure applies to virtual channels at the interface layer and virtual channels associated with a virtual path terminator.

Procedure steps

1 Configure a virtual channel. See the procedure "Configuring virtual channel connections" (page 114) for details.

2 Add the *Test* component.

```
add AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Test
```

3 Specify that the connection is bandwidth elastic or non-elastic.

```
set AtmIf/<n> Vcc/<x> Test bandwidthElastic <bwEl>
```

4 Specify the holding priority of the connection.

```
set AtmIf/<n> Vcc/<x> Test overrideHoldingPriority
<ohpri>
```

5 Add a *Loop* component.

```
add AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Loop
```

6 As an alternative to provisioning a *Loop* component, you may also add an *Nrp* component and then set the next hop to itself. Add the *NailedUpRelayPoint* component.

```
add AtmIf/<n> [Vpt/<Vpi>] Vcc/<VPI.VCI> Nrp
```

7 Set the next hop to itself.

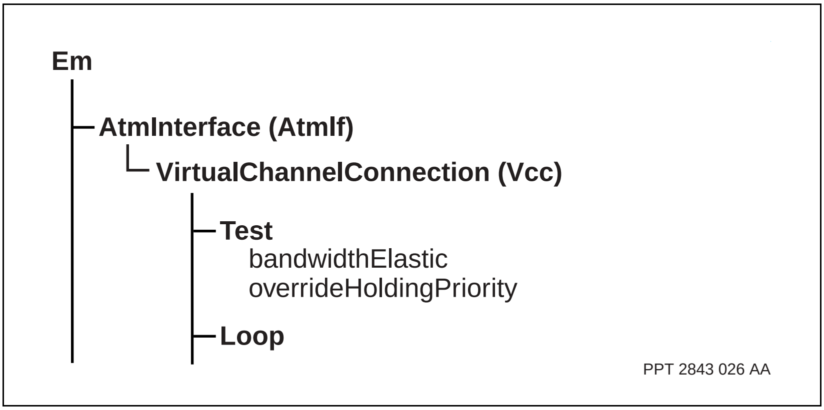
```
set AtmIf/<n1> [Vpt/<Vpi>] Vcc/<x> Nrp nextHop AtmIf/
<n2> [Vpt/<Vpi>] Vcc/<x> Nrp
```

Variable definitions

Variable	Value
<bwEl>	is yes or no. The default is no.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<ohpri>	is any number from 0 to 4 inclusive (0 is the highest priority and 4 is the lowest priority), or noOverride. The value provisioned for this attribute overrides the holding priority specified for each *Vcc* or *Vcd* component. If noOverride is selected, the holding priority specified for each *Vcc* or *Vcd* component is used.
<Vpi>	is the instance value of the *Vpt* component and can be any unique value from 0 to 4 095.
[Vpt/<Vpi>]	is the *VirtualPathTerminator* component. Use this parameter if you are associating the virtual connection with a virtual path terminator.
<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.

Procedure job aid

Figure 53
Virtual channels and ATM interfaces component hierarchy



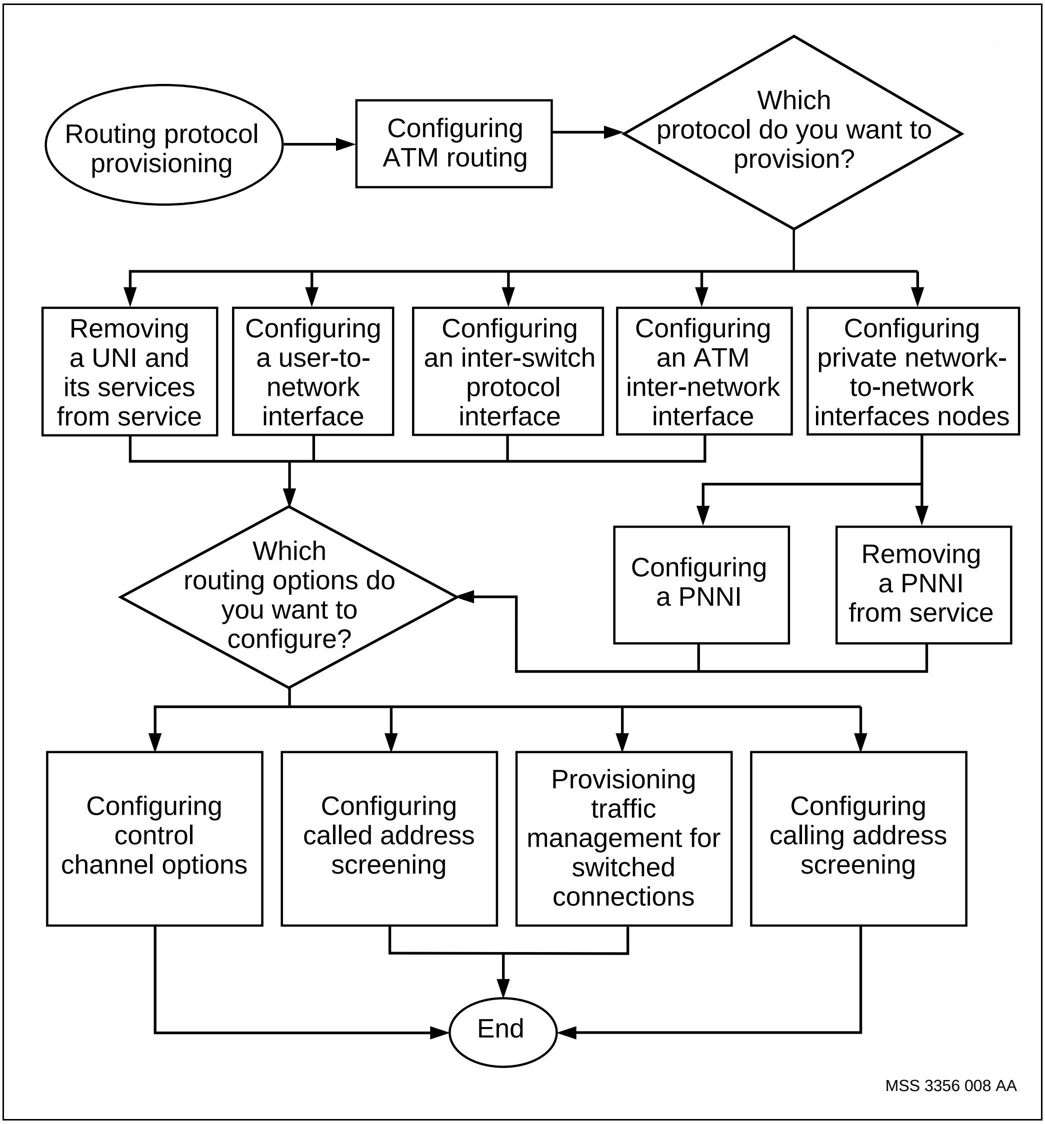
Chapter 6
Routing protocol provisioning

Provision a routing protocol as part of provisioning switched connections. See “Switched connection provisioning” (page 155).

Routing protocol provisioning procedures

This task flow shows you the sequence of procedures you perform to provision routing protocols. To link to any procedure, go to “Routing protocol provisioning procedure navigation” (page 204).

Figure 54
Routing protocol provisioning procedures



Routing protocol provisioning procedure navigation

- “Configuring ATM routing” (page 206)

Configuring ATM routing

Configure ATM routing of any protocol, by adding the ARTG component. If the network involves permanent connections and paths only, you do not need to use this procedure.

Prerequisites

- In Nortel Networks Multiservice Switch releases that support PNNI 1.0, the *AtmCallRouter* component is not used. When changing from a release that does not support PNNI to a PNNI-compliant release, the migration process automatically replaces the *AtmCallRouter* component with the *AtmRouting* component. You must add the *AtmRouting* component as part of the regular configuring process.

Procedure steps

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. 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 provision a software LPT on a per FP basis. For example, add sw lpt/atm1 correlates with shelf card 1.

set Sw Lpt/ATM featureList

4 Add ATM routing.

add ARtg

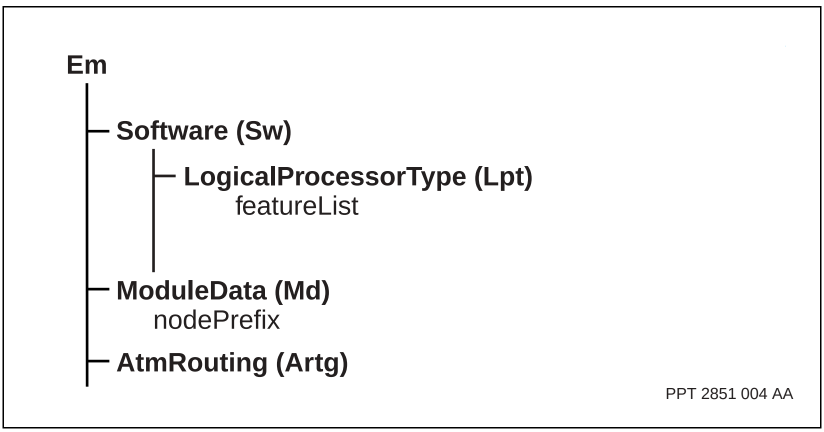
5 Configure the node for static or dynamic routing. See “Configuring PNNI dynamic routing” (page 236) for information on configuring dynamic routing.

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

Figure 55
ATM routing component hierarchy



Configuring a user-to-network connection

Configure a user-to-network connection by adding a user-to-network interface (UNI) to an ATM interface.

> ***Note:*** For APC-based fps, all *AtmIf* components under each APC must be configured as UNIs.

Prerequisites

- For an overview of how this procedure fits into provisioning switched connections, see “Switched connection provisioning” (page 155).

Procedure steps

1 Set the OAM segment boundary to be compatible with a UNI.

```
set AtmIf/<n> oamSegmentBoundary yes
```

2 Define the ATM interface as UNI.

```
add AtmIf/<n> Uni
```

The system automatically creates the *Signalling* and *Ilmi* subcomponents when you provision the *Uni* component.

3 Set the attributes for the *Uni* component.

```
set AtmIf/<n> Uni version <forumVer>, side <ifSide>,
softPvpAndPvcRetryPeriod <seconds> loopPrevention
<loopPrevention>interfaceType <iftype>
```

4 Provision the UNI addresses.

```
add AtmIf/<n> Uni Address/<address>, <addr_type>
```

5 Configure the UNI address scope for static group addresses.

```
add AtmIf/<n> Uni Address/<address>, <addr_type>
UniInfo scope <scope> pnniReachability <pnnireach>
```

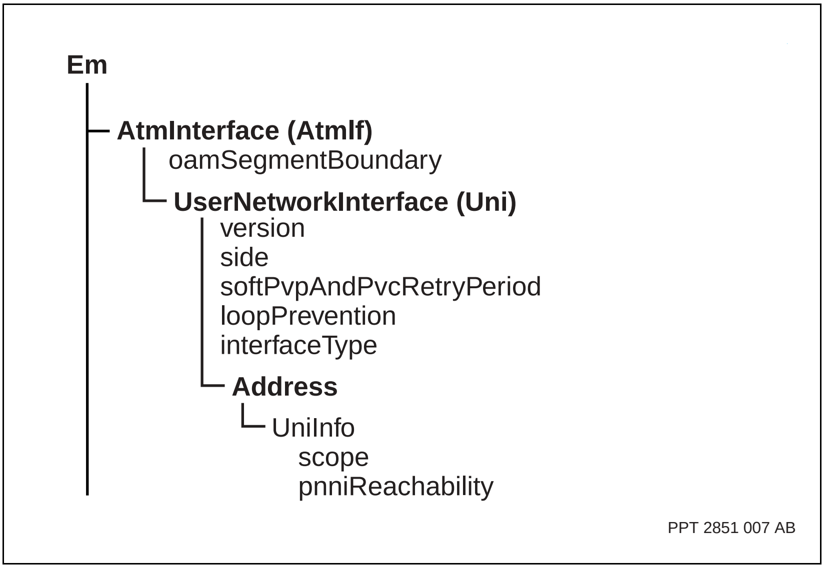
Variable definitions

Variable	Value
<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 Networks Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals* and *ATM User-to-Network Interface Specification Version 3.1* (ATM Forum Technical Committee).
<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 provision multiple static address for each UNI, but all address entries must be unique on this UNI.
<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 provisioned as network.
<iftype>	is public or private. The default is private.
<loopPrevention>	is enabled or disabled. The default is enabled. This attribute defines whether loop prevention is in effect on this interface. When the value is enabled, incoming calls on this interface cannot be routed back out the same interface. When the value is disabled, the routing of incoming calls back out the same interface is permitted. Set the value to disabled only if the routing of incoming calls back out the interface is actually required. Setting the value to enabled prevents only routing loops which involve two network nodes.
<n>	is the number of the ATM interface associated with the UNI.
(Sheet 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.
<seconds>	is a value between 20 and 999 999. The default value is 60. This attribute defines the number of seconds that the application will wait before retrying call set up for SPVCs or SPVPs. Resolution is to the nearest 10 s; for example, if this value is entered as 35, the actual retry time is set to 40 s.
(Sheet 2 of 2)	

Procedure job aid

Figure 56
User-to-network connection component hierarchy



Removing a UNI and its addresses from service

Remove a UNI and its addresses from service in preparation for removing an FP from a shelf. Use this procedure when upgrading FPs to ensure that connections are not improperly removed from service. By removing addresses from a UNI, new connections to those addresses are blocked while existing connections are allowed to remain in service as long as the customer requires.

Procedure steps

1 De-commission the addresses for the UNI under the ATM interface.

```
delete AtmIf/<n> Uni Address/<address>, <addr_type>
```

2 Repeat step 1 for all addresses associated with the UNI for this ATM interface. As addresses are removed, no new connections can be directed to those addresses although existing connections remain in service until terminated by the customer.

3 Wait until all connections are terminated by customers. To confirm, display active connections for the UNI.

```
display AtmIf/<n> Uni Signalling currentConnections
```

4 Repeat step 1 through step 3 on adjacent nodes for all ATM interfaces that point to the addresses you are removing from the local node.

5 When all connections have terminated, delete the ATM interface.

```
delete AtmIf/<n>
```

6 Repeat step 1 through step 5 for each ATM interface on the FP.

7 If there are other linked components associated with the ATM interface, delete these components as required.

8 Remove the FP from the shelf.

Variable definitions

Variable	Value
<addr_type>	is primary or alternate.
<address>	is a static address associated with the UNI.
<n>	is the number of the ATM interface associated with the UNI.

Configuring an inter-switch protocol interface

Configure an inter-switch protocol (IISP) interface by adding it to an ATM interface.

Prerequisites

- For an overview of how this procedure fits into provisioning switched connections, see “Switched connection provisioning” (page 155).

Procedure steps

1 Set the OAM segment boundary to be compatible with an IISP interface.

```
set AtmIf/<n> oamSegmentBoundary no
```

2 Define the ATM interface as IISP.

```
add AtmIf/<n> Iisp
```

The system automatically creates the *Signalling* subcomponent when you provision the *Iisp* component.

3 Set the attributes for the *Iisp* component.

```
set AtmIf/<n> Iisp version <forumVer>, side <ifSide>,
softPvpAndPvcRetryPeriod <seconds>
interDomainCrankback <crank>
```

4 Provision the IISP interface addresses.

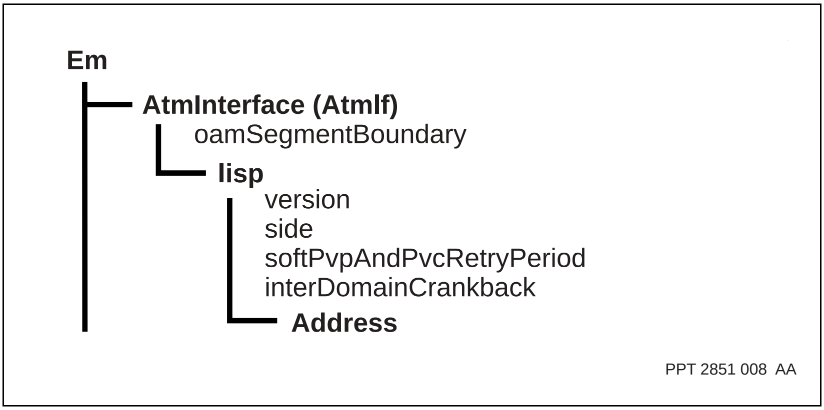
```
add AtmIf/<n> Iisp address/<address>,<type>
```

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 provision multiple static addresses for each IISP interface, but all address entries must be unique.
<crank>	is disabled or enabled. The default value is disabled. This attribute specifies whether the designated transit list (DTL) terminator inserts a crankback information element (IE) when a hop-by-hop routing crankback fails in a neighboring domain. This attribute can be set to enabled to specify to the DTL terminator that the crankback IE should be inserted when hop-by-hop crankback fails. This attribute can also be set to disabled to specify to the DTL terminator not to insert the crankback IE.
<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.
<seconds>	is a value between 20 and 999 999. The default value is 60. This attribute defines the number of seconds that the application will wait before retrying call set up for SPVCs and SPVPs. Resolution is to the nearest 10 s. For example, if you enter this value as 35, the actual retry time is set to 40 s.
<type>	is primary or alternate.
	ATM device-specific addresses are accessible through the ATM interface.

Procedure job aid

Figure 57
Inter-switch protocol interface component hierarchy



Configuring an ATM inter-network interface

Configure an ATM inter-network interface (AINI) by adding it to an ATM interface.

Prerequisites

- For an overview of how this procedure fits into provisioning switched connections, see “Switched connection provisioning” (page 155).

Procedure steps

1 Optionally, set the OAM segment boundary.

```
set AtmIf/<n> oamSegmentBoundary no
```

2 Define the ATM interface as AINI.

```
add AtmIf/<n> Aini
```

The system automatically creates the *Signalling* subcomponent when you provision the *Aini* component.

3 Set the attributes for the *Aini* component.

```
set AtmIf/<n> Aini side <side>,
softPvpAndPvcRetryPeriod <seconds>
softPvpAndPvcHoldOffTime <milliseconds>
passAlongRequest <passalong>
```

4 Provision the Aini interface addresses.

```
add AtmIf/<n> Aini address/<address>,<type>
```

ATM device-specific addresses are accessible through the ATM interface.

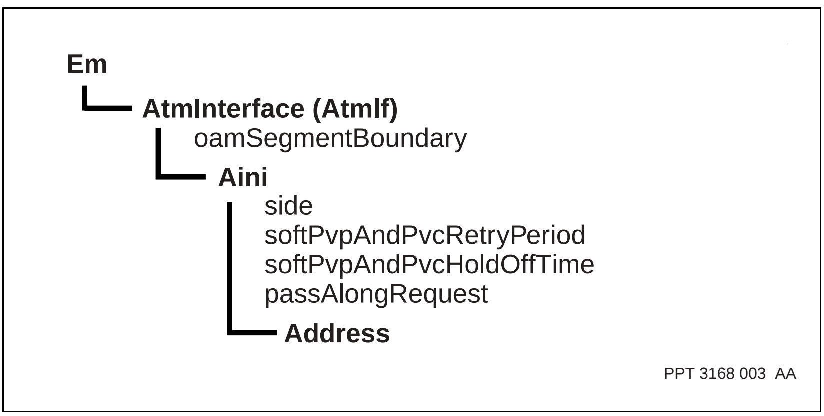
Variable definitions

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 provision multiple static addresses for each AINI interface, but all address entries must be unique.
<milliseconds>	is a value between 0.5 and 20 000. The default value is 0. This attribute defines how long to hold off retries following a release of the connection. The resolution is to the nearest 50 ms.
(Sheet 1 of 2)	

Variable	Value
<n>	is the number of the ATM interface.
<passAlongRequest>	is allowed or discarded. The default value is allowed. This attribute defines whether unrecognized messages and information elements marked as pass along are permitted to cross this signaling interface. A value of allowed enables the message or information element to be passed along without error checking. A value of discarded causes the unrecognized message and information element to be discarded.
<seconds>	is a value between 20 and 999 999. The default value is 60. This attribute defines the number of seconds that the application will wait before retrying call set up for SPVCs and SPVPs. The resolution is to the nearest 10 s. For example, if you enter this value as 35, the actual retry time is set to 40 s.
<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.
(Sheet 2 of 2)	

Procedure job aid

Figure 58
ATM inter-network interface component hierarchy



Configuring private network-to-network interface nodes

Configure private network-to-network interface (PNNI) nodes as required. See “PNNI node configuration” (page 231).

Configuring a PNNI

Configure a private network-to-network interface (PNNI) by adding it to an ATM interface.

Prerequisites

- For an overview of how this procedure fits into provisioning switched connections, see “Switched connection provisioning” (page 155).
- Configure PNNI nodes. See “PNNI node configuration” (page 231).

Procedure steps

1 For PNNI connections, set the cell delay variation requirements for the traffic in each applicable service category.

```
set AtmIf/<n> Ca <ServiceCategory>/0 cdv <cdv>
```

2 For PNNI connections, set the maximum cell transfer delay for the traffic in each applicable service category.

```
set AtmIf/<n> Ca <ServiceCategory>/0 maxCtd
<maxCtdValue>
```

3 Set the OAM segment boundary to be compatible with a PNNI.

```
set AtmIf/<n> oamSegmentBoundary no
```

4 Define the ATM interface as PNNI.

```
add AtmIf/<n> Pnni
```

The system automatically creates the *Signalling* and *Rcc* subcomponents when you provision the *Pnni* component.

5 Define the time-out period for SPVC and SPVP retries.

```
set AtmIf/<n> Pnni softPvpAndPvcRetryPeriod <seconds>
```

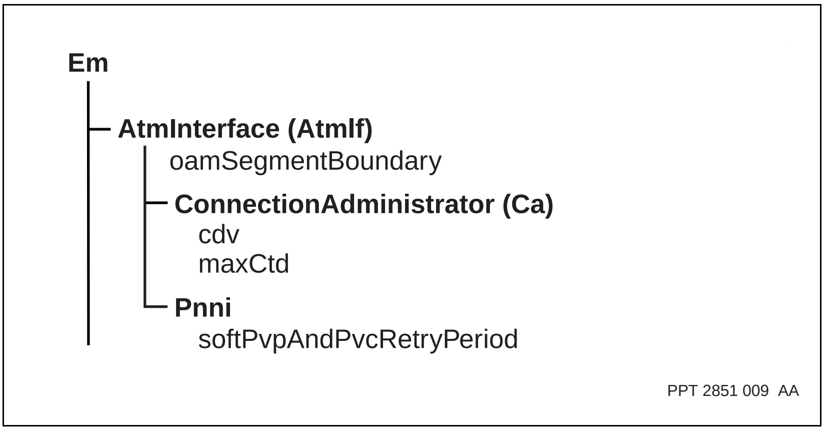
Variable definitions

Variable	Value
<cdv>	is a decimal between 0 and 16 777 215. The default value is 250. Cdv is measured in microseconds.
<maxCtd>	is a decimal between 0 and 16 777 215. The default value is 1 000. MaxCtd is measured in microseconds.
(Sheet 1 of 2)	

Variable	Value
<n>	is the number of the ATM interface that you want to associate with the PNNI.
<seconds>	is a decimal value between 20 and 999 999. The default value is 60. This attribute defines the number of seconds that the application will wait before retrying call set up for SPVCs. Resolution is to the nearest 10 s. For example, if you enter 35, the system sets the provisioned value to 40 s. If you enter 34, it sets the provisioned value to 30 s.
(Sheet 2 of 2)	

Procedure job aid

Figure 59
Private network-to-network interface component hierarchy



Removing a PNNI from service

Remove a PNNI from service to ensure that connections are not improperly removed from service. By removing the PNNI from service before replacing the FP, new connections to the interface are blocked while existing connections are allowed to remain in service as long as the customer requires.

Procedure steps

1 De-commission any static addresses for the PNNI under the ATM interface.

```
delete AtmIf/<n> PNNI Address/<address>
```

2 Repeat step 1 for all static addresses associated with the PNNI for this ATM interface. As addresses are removed, no new connections can be directed to those addresses through this ATM interface.

3 Wait until all connections associated with static addresses are terminated by customers. To confirm, display active connections for the PNNI.

```
display AtmIf/<n> Pnni Signalling currentConnections
```

4 When all connections associated with static addresses have terminated, delete the ATM interface.

```
delete AtmIf/<n>
```

5 Repeat step 1 through step 4 for each ATM interface on the FP.

6 If there are other linked components associated with the ATM interface, delete these components as required.

7 Delete the logical processor.

Variable definitions

Variable	Value
<address>	is a static address associated with the PNNI.
<n>	is the number of the ATM interface associated with the PNNI.

Configuring control channel options

Configure control channel options to change the default signaling control channel parameters.

Prerequisites

Use one of the following procedures to define the ATM interface on the source node (the node that originates the call setup request):

- “Configuring a user-to-network connection” (page 208) to define a UNI
- “Configuring an inter-switch protocol interface” (page 213) to define an IISP
- “Configuring an ATM inter-network interface” (page 216) to define an AINI
- “Configuring a PNNI” (page 219) to define a PNNI

Procedure steps

1 Set up the signaling channel VCI number. The signaling VCI for the interface on the other end of the link must match the value in this attribute.

```
set AtmIf/<n> <IfType> Sig vci <vci>
```

2 Define the address conversion requirements on the outgoing call set-up request protocol data units (PDU) on the signaling channel.

```
set AtmIf/<n> <IfType> Sig addressConversion
<convType>
```

3 For a UNI, you can also enable or disable the UNI signaling channel. Set the operating mode as required.

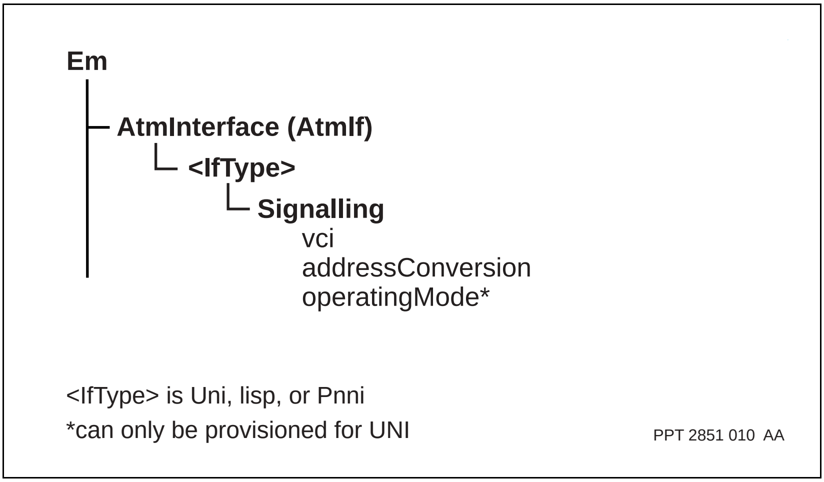
```
set AtmIf/<n> Uni Sig operatingMode <opmode>
```

Variable definitions

Variable	Value
<convType>	is none, nativeE164, or nsap. The default is none. If you use the default value none, nothing is done to the called party number in the outgoing signaling message. If you provision the nativeE164 option, then the interface converts only NSAP encapsulated E.164 addresses to native E.164 format in the outgoing signaling message. If the NSAP address is not in encapsulated E.164 format, the node clears the call. If you provision the nsap option, then the interface converts native E.164 addresses to NSAP E.164.
<IfType>	is Uni, Iisp, Aini, or Pnni.
<n>	is the number of the ATM interface.
<opmode>	is normal or provisionedOnly. The default is normal.
<vci>	is the VCI under VPI=0 for the signaling channel. The default value is 5. It is recommended not to change this default value. This value can be any VCI in the connection map address space for VPI 0.

Procedure job aid

Figure 60
Control channel options component hierarchy



Configuring called address screening

Configure called address screening as part of enabling address screening on the node. By default, called address screening is disabled, which means that calls are not screened.

When you enable called address screening, incoming calls are screened against the addresses that you have provisioned. The incoming call is accepted if the called address matches an address you have provisioned to be accepted. The incoming call is rejected if the called address is either not provisioned, matches an address you have provisioned to be rejected, or does not match any provisioned address.

Procedure steps

1 Provision 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.

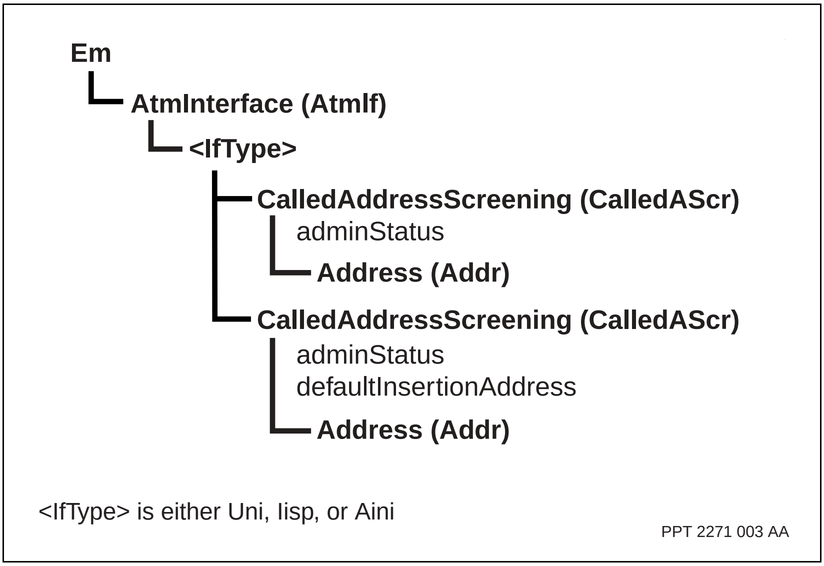
set AtmIf/<n> <IfType> CalledAScr adminStatus enabled

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

Figure 61
Called address screening component hierarchy



Configuring calling address screening

Configure calling address screening as part of enabling address screening on the node. 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 you have provisioned. The incoming call is accepted if the calling address matches an address you have provisioned to be accepted. The incoming call is rejected if the calling address either matches an address that you have provisioned to be rejected, does not match a provisioned address, or the connection request does not contain a calling address and there is no provisioned default insertion address.

Procedure steps

1 Provision 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
```

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.
(Sheet 1 of 2)	

Variable	Value
<IfType>	is Uni, Iisp or Aini
<n>	is the number of the ATM interface.
(Sheet 2 of 2)	

Provisioning switched connection traffic management

Provision switched connection traffic management to optimize switched connection performance. See “Switched connection traffic management provisioning” (page 301).

Chapter 7
PNNI node configuration

As part of provisioning switched connections using PNNI protocol, provision PNNI at the node level. After the nodes are provisioned, individual PNNI connections can be set up.

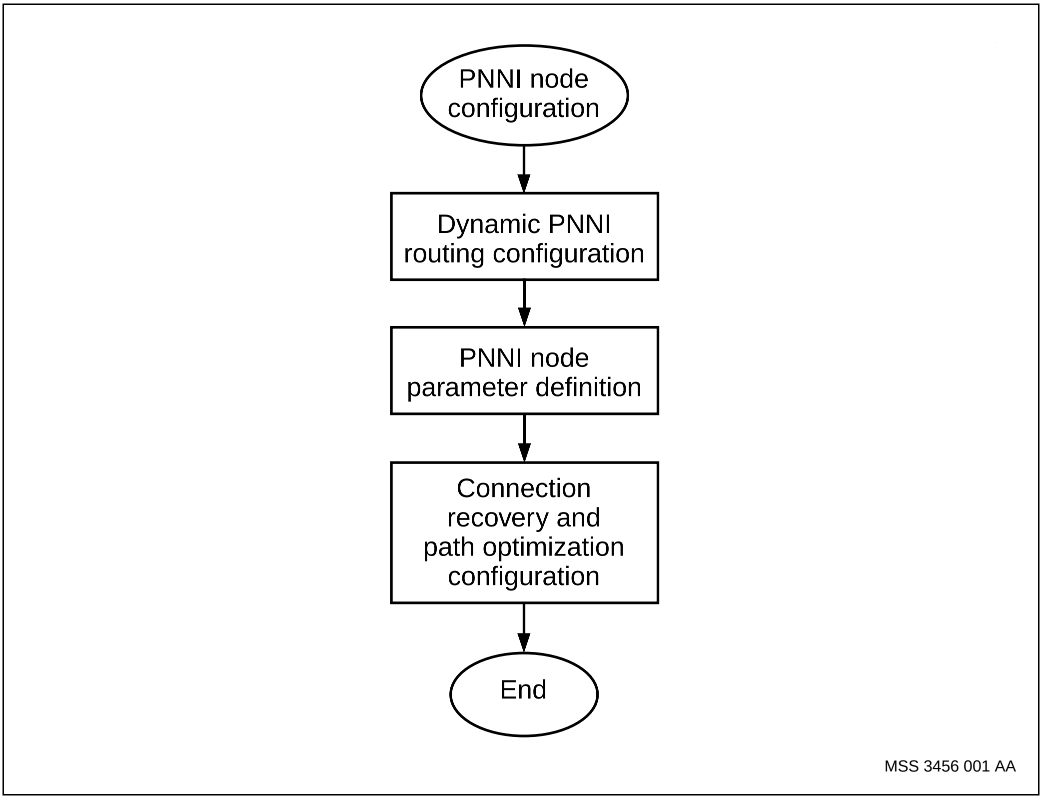
Prerequisites to PNNI node configuration

- 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 Networks Multiservice Switch node.
- See NN10600-702 *Nortel Networks 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.

PNNI node configuration tasks

This work flow shows you the sequence of procedures you perform to provision PNNI nodes. To link to any task, go to “PNNI node configuration task navigation” (page 232).

Figure 62
PNNI node configuration tasks



PNNI node configuration task navigation

- "Dynamic PNNI routing configuration" (page 233)
- "PNNI node parameter definition" (page 249)
- "Connection recovery and path optimization configuration" (page 277)

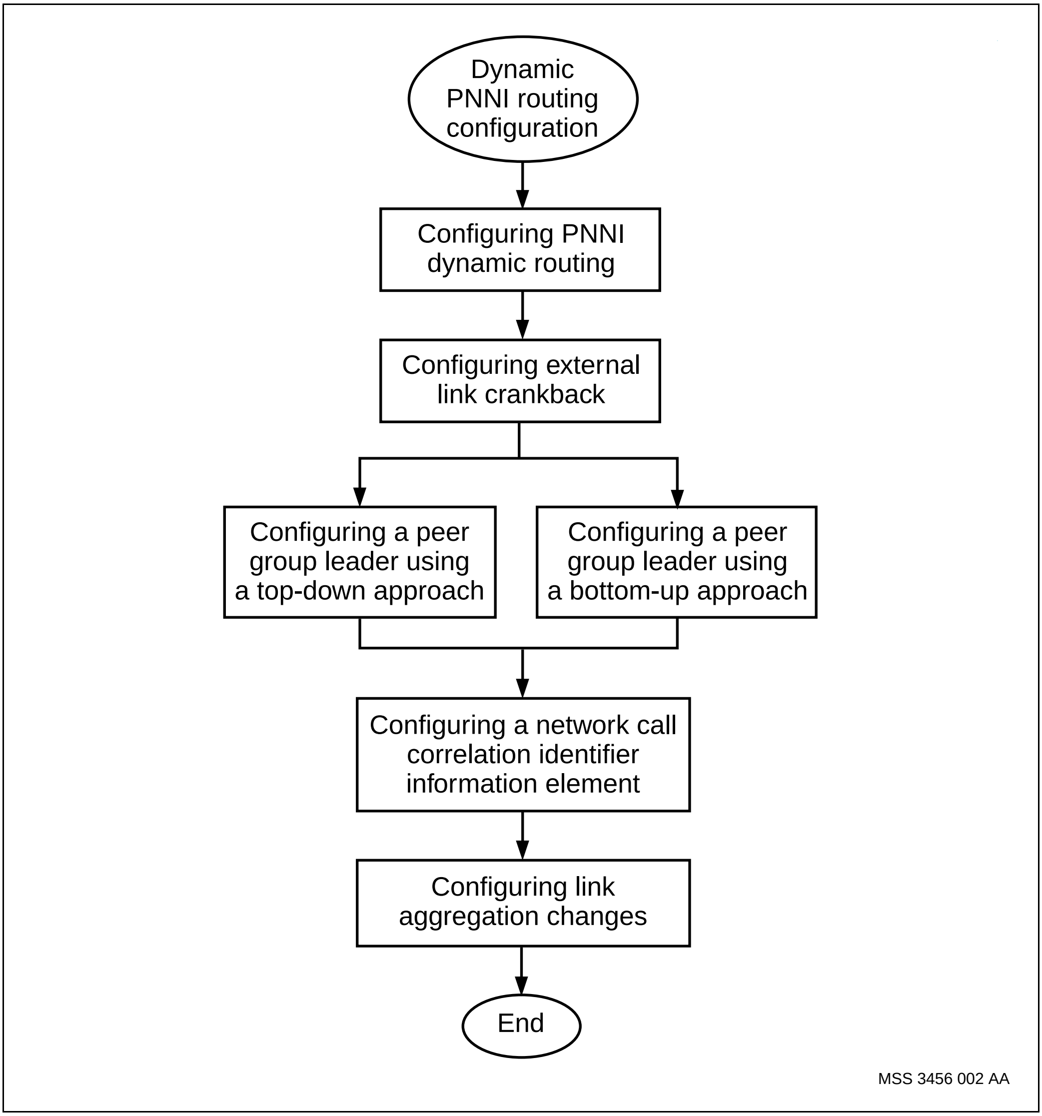
Dynamic PNNI routing configuration

Configure dynamic PNNI routing to establish dynamic routing in your network.

Dynamic PNNI routing configuration procedures

This task flow shows you the sequence of procedures you perform to complete the configuration of PNNI dynamic routing. To link to any procedure, go to “Dynamic PNNI routing configuration procedure navigation” (page 234).

Figure 63
Dynamic PNNI routing configuration procedures



Dynamic PNNI routing configuration procedure navigation

- “Configuring PNNI dynamic routing” (page 236)

Configuring PNNI dynamic routing

Configure PNNI dynamic routing to define a level of participation for the node in the PNNI networking hierarchy. Ask network engineering personnel for information about determining level assignment.

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

Procedure steps

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>
```

3 If necessary, specify the correct peer group ID for the node.

```
set ARtg Pnni CfgNode/<CfgNode_level> peerGroupId
<pgId>
```

4 If the value in the *nodeAddressPrefix* is not appropriate for the network configuration, redefine the default ATM address for this node.

```
set ARtg Pnni nodeAddressPrefix <address>
```

5 Verify that the node identifier is correct. The node advertises this identifier to other nodes in the network.

```
display ARtg Pnni CfgNode/<CfgNode_level> nodeId
```

6 If the value in the *nodeId* attribute is not correct, change the node identifier.

```
set ARtg Pnni CfgNode/<CfgNode_level> nodeId <nodeId>
```

WARNING
This step clears all active calls
Configuring the *nodeId* attribute releases all existing connections.

Variable definitions

Variable	Value
<address>	is a hexadecimal entry of up to 38 hexadecimal digits.
<CfgNode_level>	is the level of the node in the PNNI networking hierarchy having a value between 0 and 104, inclusive.
<nodeId>	is a hexadecimal entry of up to 22 hexadecimal digits.
<pgId>	is a hexadecimal entry of up to 28 hexadecimal digits. The value for the *ConfiguredNode* component is in decimal and the value for the *peerGroupID* attribute is in hexadecimal. That is, when configuring the *peerGroupID* attribute you must translate the decimal value. An example of a peer group ID (14 octets or 28 digits) is 6045613889901111111111111100 In this example, the first octet indicates the level at which the peer group (PG) is defined. In this case, 60 hexadecimal digits are equal to 96 decimal digits. For Nortel Networks Multiservice Switch nodes, you have the ability to configure your peer group ID (PGID) two ways. The first way is to let the ATM node address determine the level (d -p mod nodePrefix). This method requires that all the nodes that you want in the same PG be identical for the same number of octets. In this example, since the level is 96, all nodes in that respective PG have to have the same address for 24 digits (12 octets). The easiest way to figure this out is take the PNNI value, 96 divided by 4 = 24, which is the number of significant digits that must be the same. The second way is to manually provision the PGID (d -p artg pnni cfg/x peerGroupId). In this example, the address is 28 digits long with the cfg level indicator in hexadecimal as the first octet. The *peerGroupId* attribute will already have been set if you configured node addressing through the *ModuleData nodePrefix* attribute. Also, if you configure the *ARtg Pnni nodeAddressPrefix* attribute, the node automatically sets the *peerGroupId* attribute. The *ARtg Pnni nodeAddressPrefix* attribute overrides the value determined by the *ModuleData nodePrefix* attribute.

Procedure job aid

Figure 64
PNNI dynamic routing component hierarchy

- **Em**
  - **AtmRouting (Atrg)**
    - **Pnni**
      - nodeAddressPrefix
      - **ConfigureNode (CfgNode)**
        - peerGroupId (pgId)
        - nodeId

PPT 3439 006 AA

Configuring external link crankback

Configure external link crankback to set up parameters defining when the network will release a connection setup that is in progress if that setup request encounters a failure in the network.

Procedure steps

1 Add ATM routing.

```
add ARtg
```

2 Set the external link PNNI crankback value.

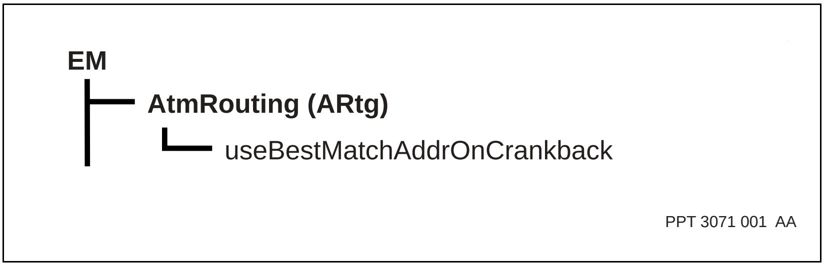
```
set ARtg useBestMatchAddrOnCrankback <crankback>
```

Variable definitions

Variable	Value
<crankback>	specifies when the ATM routing system should be using best match addresses when performing a crankback. By default, when performing a crankback, the ATM routing system only considers addresses of length equal to the length of the address used during the first routing attempt. With this attribute, you may specify that the ATM routing system also considers the best match addresses for a crankback when this node is the DTL originating node, the DTL terminating node, or acts as both the DTL originating and terminating node. The attribute can be set to 4 different values, including none, which is the default value. The following list details the meaning of the values: • A value of none specifies that the DTL originator and terminator are not using the best match address resolution when calculating a route. • A value of origOnly specifies that only the DTL originator uses the best match address resolution when calculating a route. • A value of termOnly specifies that only the DTL terminator uses the best match address resolution when calculating a route. • A value of origTerm specifies that both the DTL originator and terminator uses the best match address resolution when calculating a route.

Procedure job aid

Figure 65
External link crankback component hierarchy



Configuring a peer group leader using a top-down approach

Configure a peer group leader (PGL) using a top-down approach to add a new level to the PNNI hierarchy. You can configure a 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

1 Create another level in the PNNI networking hierarchy that is below that added in “Configuring PNNI dynamic routing” (page 236).

add ARtg Pnni CfgNode/<CfgNode_level>

The new *CfgNode* must have a value that is numerically greater than that configured in “Configuring PNNI dynamic routing” (page 236).

2 Add the new PNNI *Pgl* component to the new node added in step 1.

add ARtg Pnni CfgNode/<CfgNode_level> Pgl

3 Check to see that the next higher PNNI level created in “Configuring PNNI dynamic routing” (page 236) exists.

list ARtg Pnni CfgNode/*

4 If the next higher PNNI level created in “Configuring PNNI dynamic routing” (page 236) does not exist, add it.

add ARtg Pnni CfgNode/<CfgNode_level2>

5 Set the priority for the PGL added in step 2 to a non-zero value.

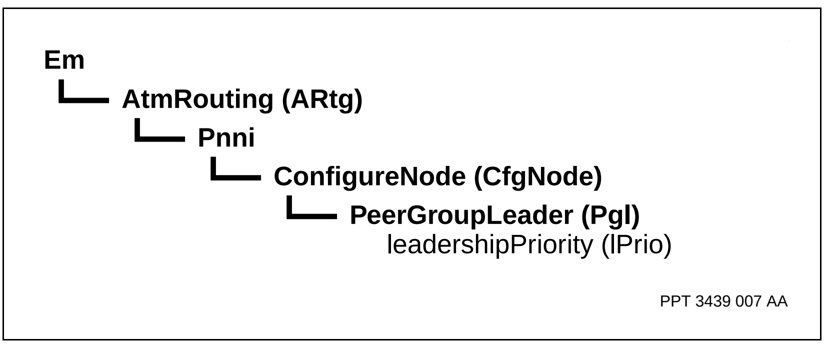
set ARtg Pnni CfgNode/<CfgNode_level> Pgl lPrio <lPrio>

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 PNNI dynamic routing” (page 236). In a hierarchical PNNI network, the Nortel Networks Multiservice Switch system defines level 0 as the top-most level and level 104 as the bottom-most level.
<CfgNode_level2>	is a value that is numerically smaller that configured in step 1.
<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

Figure 66
Peer group leader using a top-down approach component hierarchy



Configuring a peer group leader using a bottom-up approach

Configure a peer group leader (PGL) using a bottom-up approach to allow a node to become eligible for election as the PGL. By configuring some nodes to be PGLs, you add a new level to the PNNI hierarchy. If the node's election priority is high enough, the node will be elected as the PGL.

Procedure steps

1 Add the new PNNI *Pgl* component to the node added in "Configuring PNNI dynamic routing" (page 236).

add ARtg Pnni CfgNode/<CfgNode_level> Pgl

2 Create a level that is higher in the PNNI networking hierarchy than that configured in "Configuring PNNI dynamic routing" (page 236).

add ARtg Pnni CfgNode/<CfgNode_level2>

The new *CfgNode* must have a value that is numerically smaller than that configured in "Configuring PNNI dynamic routing" (page 236).

3 Check to see that the higher PNNI level created in step 2 exists.

list ARtg Pnni CfgNode/*

4 If the higher PNNI level created in step 2 does not exist, add it.

add ARtg Pnni CfgNode/<CfgNode_level>

5 Set the priority for the PGL added in step 1 to a non-zero value.

set ARtg Pnni CfgNode/<CfgNode_level> Pgl lPrio <lPrio>

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.

Variable definitions

Variable	Value
<CfgNode_level>	is the level of the node added in “Configuring PNNI dynamic routing” (page 236).
<CfgNode_level2>	is the level of the node in the PNNI networking hierarchy having a value that is numerically smaller than that configured in “Configuring PNNI dynamic routing” (page 236).
<lPrio>	is a value specifying the leadership priority in the range 1 through 205 inclusive. The default value is 1.

Procedure job aid

Figure 67
Peer group leader using a bottom-up approach component hierarchy

Em
- **AtmRouting (ARtg)**
  - **Pnni**
    - **ConfigureNode (CfgNode)**
      - **PeerGroupLeader (Pgl)**
        leadershipPriority (lPrio)

PPT 3439 007 AA

Configuring the network call correlation identifier information element

Configure the network call correlation identifier information element (NCCI IE) functionality to provide a provisionable attribute *outgoingNcciIeFormat* that can be set to two different values: backwardsCompatible and atmForum.

Prerequisites

- The atmForum value can not be enabled until after all the nodes in the network are upgraded to the software release containing the NCCI IE functionality.

Procedure steps

1 Set the NCCI IE identification option.

```
set Artg Pnni outgoingNcciIeFormat
<outgoingNcciIeFormat>
```

Variable definitions

Variable	Value
<outgoingNcciIeFormat>	is the NCCI IE format in outgoing SETUP messages for all PNNI interfaces on the node. It can be set to backwardsCompatible or atmForum. The backwardsCompatible value is mandatory if any node within the Multiservice Switch network is not running a software release containing NCCI IE functionality. Multiservice Switch systems supply NCCI IE default values in the following two situations: • If the Artg Pnni component is added during initial software loading, the value is backwardsCompatible. • If the Artg Pnni component is added by an operator command, there is no Artg Pnni component during initial software loading, and the value is atmForum.

Procedure job aid

Figure 68
Network call correlation identifier information element component hierarchy

Em
└── **AtmRouting (ARtg)**
  └── **Pnni**
    outgoingNcciIeFormat

PPT 3168 005 AA

Configuring link aggregation changes

Configure link aggregation changes to the aggregation token. Changing the aggregation token changes the way links are aggregated. The link remains available for routing while you are making this change.

> ***Note:*** The aggregation token at each end of the link is used to form the derived aggregation token for the link. Link aggregation is not performed for the unknown type links and at the lowest level node except in the degenerate node situation. The derived aggregation token is not displayed for the link if the link aggregation is not performed for the link.

Procedure steps

1 Set the aggregation token for the ATM interface at both ends of the link.

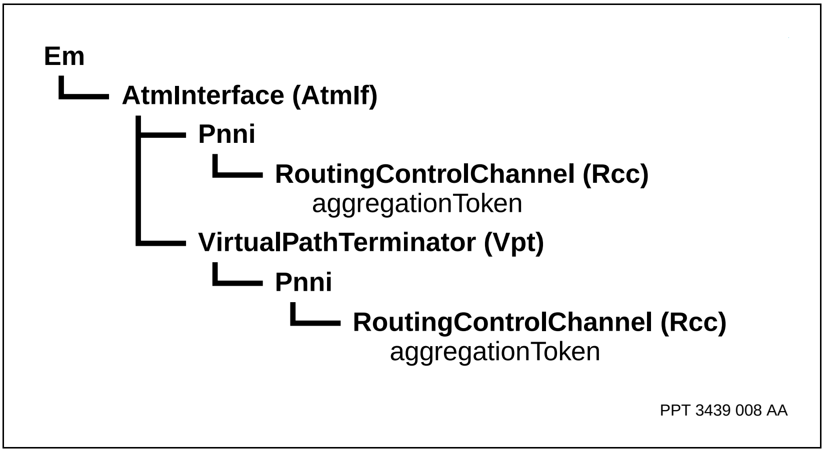
set AtmIf/<AtmIf> Pnni Rcc aggregationToken <aggregationToken>

Variable definitions

Variable	Value
<aggregationToken>	is a value used to form the derived aggregation token for the link.
<AtmIf>	is the instance number of the ATM interface.

Procedure job aid

Figure 69
Link aggregation changes component hierarchy



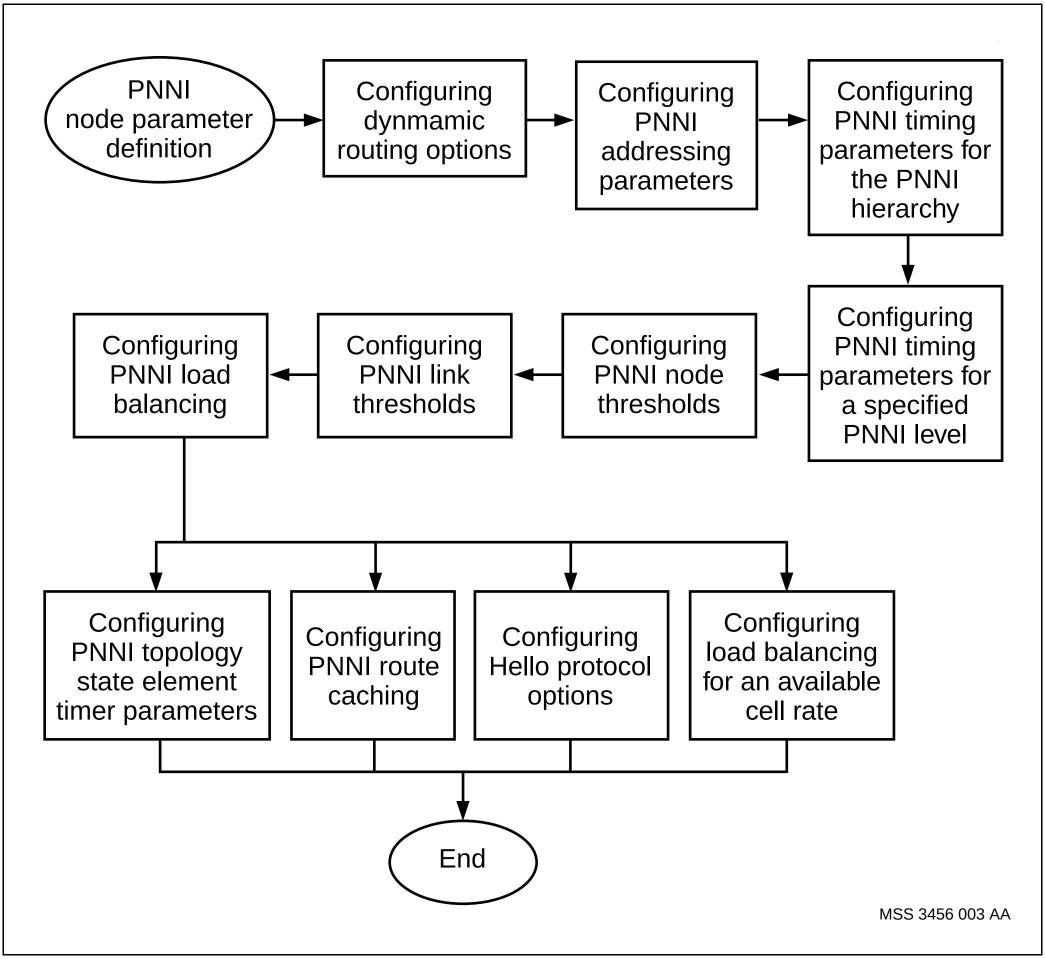
PNNI node parameter definition

Define PNNI node parameters to match the specific requirements of your network.

PNNI node parameter definition procedures

This task flow shows you the sequence of procedures you perform to define PNNI node parameters. To link to any procedure, go to "PNNI node parameter definition procedure navigation" (page 250).

Figure 70
PNNI node parameter definition procedures



PNNI node parameter definition procedure navigation

- "Configuring dynamic routing options" (page 252)
- "Configuring PNNI addressing parameters" (page 255)
- "Configuring PNNI timing parameters for the PNNI hierarchy" (page 257)

Configuring dynamic routing options

Configure dynamic routing options to set parameters that differ from the default routing parameters.

Procedure steps

1 Set the highest level at which this node can advertise or summarize the node address.

```
set ARtg Pnni defaultScope <defaultScope>
```

2 Define the routing domain name.

```
set ARtg Pnni domain <domain>
```

3 Specify if this node can be used as a transit node.

```
set ARtg Pnni CfgNode/<cfgNode_level> restrictTransit
<true|false>
```

4 Specify the number of alternate routing attempts before a call requiring a crankback is rejected.

```
set ARtg Pnni maxAltRtsOnCrankbck
<maxAltRtsOnCrankbck>
```

5 Specify the path constraint that the PNNI routing algorithm can optimize. There is an optimization metric for each ATM service category.

```
set ARtg Pnni optMetric <optMetric_cbr>
<optMetric_rtVbr> <optMetric_nrtVbr> <optMetric_ubr>
```

6 Configure the mapping for UNI scope to PNNI routing level.

```
set ARtg Pnni uniToPnniScopeMap <uniToPnniScopeMap>
```

7 Configure the maximum number of routing paths available for load balancing.

```
set ARtg Pnni Lb/<ATM_service_category> maxPaths
<maxPaths>
```

8 Configure the diversity of the alternate path relative to the optimal path for load balancing.

```
set ARtg Pnni pathDiversity <pathDiversity>
```

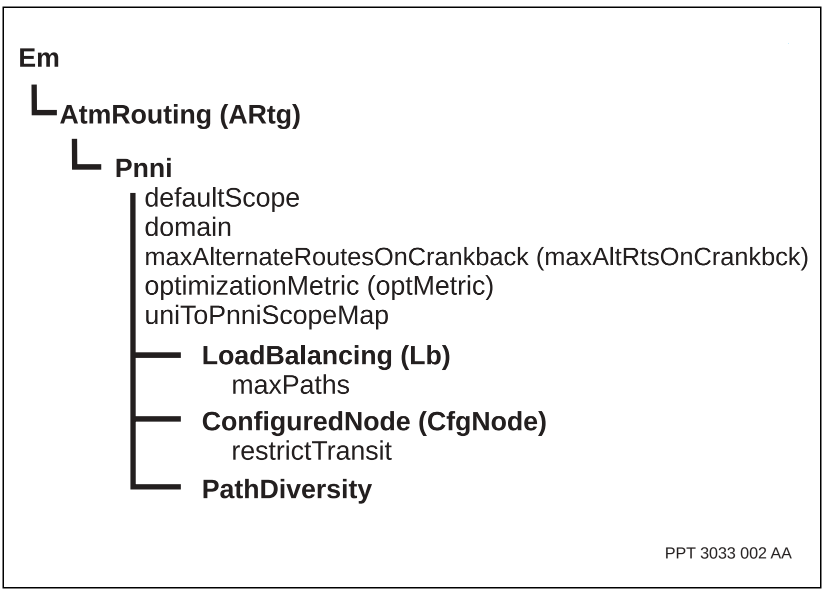
Variable definitions

Variable	Value
<ATM_service_category>	is the ATM service category you want to set.
<CfgNode_level>	is the level of the node in the PNNI networking hierarchy having a value between 0 and 104, inclusive.
<defaultScope>	is a number between 0 and 104. The default is 0. If the value is 0 the node advertises all addresses that do not have a configured scope to all other nodes in the routing domain. If the value is not 0, it must be less than or equal to the level at which this node is configured in the routing hierarchy.
<domain>	is a string entry of up to 32 characters. The value of this attribute is identical for all nodes in the same routing domain.
<maxAltRtsOnCrankbck>	is a decimal between 0 and 20. The default value is 1.
<maxPaths>	is a number between 1 and 20. If method is maxAvCr, the lb maxPaths must be 1.
<optMetric_cbr>, <optMetric_rtVbr>, <optMetric_nrtVbr>, <optMetric_ubr>	is Cdv, maxCtd, or aw for each applicable service category. The default is aw. Set the value to cdv so the routing algorithm computes a route that satisfies the call’s traffic requirements and minimizes the cell delay variation. Set the value to maxCtd so the routing algorithm computes a route that satisfies the call’s maximum cell transfer delay on call setups requiring that service category. Set the value to aw so the routing algorithm computes a route that satisfies the call’s administrative weight on call setups requiring that service category.
<pathDiversity>	is a percentage between 0 and 100.
(Sheet 1 of 2)	

Variable	Value
<true|false>	is true or false. The default is false. Set the value to true to prevent other nodes to use this node as a transit node.
<uniToPnniScopeMap>	is a vector of 15 values, where each value is between 0 and 104. The default is 96,96,96,80,80,72,72,64,64,64,48,48,32,32,0. Values are always in descending order. The first value in the vector corresponds to UNI scope 1 (localNetwork) and the last value corresponds to UNI scope 15 (global). The mapping applies to the membership scope which is passed across the UNI by way of integrated local management interface (ILMI) address registration and is then advertised with the corresponding reachable address as a PNNI routing level. This mapping is also used to map the connection scope, which is passed in setup messages for anycast call requests by way of the connection scope selection information element (CSS-IE).
(Sheet 2 of 2)	

Procedure job aid

Figure 71
Dynamic routing options component hierarchy



Configuring PNNI addressing parameters

Configure PNNI addressing parameters if you want to use values other than the default settings for PNNI addressing.

Procedure steps

1 Add summary address components under the *ConfiguredNode* component.

```
add ARtg Pnni CfgNode/<CfgNode_level> SAddr/<address>,
<prefixLength>, <reachability>
```

2 Suppress summary addresses so the node does not advertise addresses which match the prefix, regardless of scope.

```
set ARtg Pnni CfgNode/<CfgNode_level> SAddr/<address>,
<prefixLength>, <reachability> suppress <true|false>
```

Variable definitions

Variable	Value
<address>	is any address for the node at the current level, and is a hexadecimal entry of 38 hexadecimal digits. Use zeros to pad the address. A summary address is an abbreviation of a set of addresses, represented by an address prefix that all of the summarized addresses have in common.
<prefixLength>	is a decimal between 1 and 152 (no default). This attribute specifies the number of significant bits of the address. These bits are the instance for the *SummaryAddress* component.
<reachability>	is internal or exterior.
<true|false>	is true or false. The default is false.

Procedure job aid

Figure 72
PNNI addressing parameters component hierarchy

```
Em
└── AtmRouting (Atrg)
    └── Pnni
        └── ConfigureNode (CfgNode)
            └── SummaryAddress (SAddr)
                  suppress
```

PPT 3439 009 AA

Configuring PNNI timing parameters for the PNNI hierarchy

Configure PNNI timing parameters to change the default values of the timing parameters for all peer groups at all levels in the PNNI hierarchy.

Procedure steps

1 Specify in seconds how long this node will continue to advertise a horizontal link for which it has not received and processed an LGN Horizontal Link Information Group.

```
set ARtg Pnni horizontalLinkInactivityTime
<horizontalLinkInactivityTime>
```

2 Specify in seconds how long this node will delay advertising its choice of preferred peer group leader. This delay ensures that the topology information propagates across the peer group.

```
set ARtg Pnni pglInitTime <pglInitTime>
```

3 Specify in seconds how long this node will delay being declared the preferred peer group leader by unanimous agreement among its peers.

```
set ARtg Pnni overrideDelay <overrideDelay>
```

4 Specify in seconds how long this node will delay re-starting the process of electing a new peer group leader after it loses its connectivity to the current peer group leader.

```
set ARtg Pnni reElectionInterval <reElectionInterval>
```

5 Specify in seconds how long this node will delay initiating establishment of an SVCC to a neighbor after deciding that a SVCC will be established.

```
set ARtg Pnni svccInitTime <svccInitTime>
```

6 Specify in seconds how long this node will delay re-establishment of an SVCC-based RCC after it is unexpectedly torn down or after all rerouting attempts have failed.

```
set ARtg Pnni svccRetryTime <svccRetryTime>
```

7 Specify in seconds how long this node will wait for the Hello protocol over an SVCC RCC which it originated, to reach a 2-way inside state before giving up and tearing it down.

```
set ARtg Pnni svccCallingIntegrityTime
<svccCallingIntegrityTime>
```

8 Specify in seconds (s) how long this node will wait for the Hello protocol over an SVCC RCC which it has terminated, to reach a 2-way inside state before giving it up and tearing it down.

```
set ARtg Pnni svccCalledIntegrityTime
<svccCalledIntegrityTime>
```

9 Specify in seconds how long this node will wait for a SVCC SETUP message from its neighbor once it knows of the neighbor's existence.

```
set ARtg Pnni svccEstablishmentIntegrityTime
<svccEstablishmentIntegrityTime>
```

Variable definitions

Variable	Value
<horizontalLinkInactivityTime>	is a number of seconds in the range from 30 to 3 600 inclusive. The default is 120.
<overrideDelay>	is a number of seconds in the range 1 to 65 535 inclusive. The default is 30.
<pglInitTime>	is a number of seconds in the range 0 to 65 535 inclusive. The default is 15.
<reElectionInterval>	is a number of seconds in the range 1 and 65 535 inclusive. The default is 15.
<svccCalledIntegrityTime>	is a number of seconds in the range 25 and 3 615 inclusive. The default is 50.
<svccCallingIntegrityTime>	is a number of seconds in the range 10 and 3 600 inclusive. The default is 35.
<svccEstablishmentIntegrityTime>	is a number of seconds in the range 15 and 3 600 inclusive. The default is 60.
<svccInitTime>	is a number of seconds in the range 1 and 120 inclusive. The default is 4.
<svccRetryTime>	is a number of seconds in the range 20 and 60 inclusive. The default is 30.

Procedure job aid

Figure 73
PNNI timing parameters for the PNNI hierarchy component hierarchy

Em
└── **AtmRouting (Atrg)**
    └── **Pnni**
        horizontalLinkInactivityTime
        pgIInitTime
        overrideDelay
        reElectionInterval
        svccInitTime
        svccRetryTime
        svccCallingIntegrityTime
        svccCalledIntegrityTime
        svccEstablishmentIntegrityTime

PPT 3439 010 AA

Configuring PNNI timing parameters for a specified PNNI level

Configure PNNI timing parameters for a specified PNNI level to change the default timing values for all peer groups at a specified level in the PNNI hierarchy.

Procedure steps

1 Specify in seconds how long this node will continue to advertise a horizontal link for which it has not received and processed an LGN Horizontal Link Information Group.

```
set ARtg Pnni CfgNode/<CfgNode_level>
horizontalLinkInactivityTime
<horizontalLinkInactivityTime>
```

2 Specify in seconds how long this node will delay initiating establishment of an SVCC to a neighbor after deciding that a SVCC will be established.

```
set ARtg Pnni CfgNode/<CfgNode_level> svccInitTime
<svccInitTime>
```

3 Specify in seconds how long this node will delay re-establishment of an SVCC-based RCC after it is unexpectedly torn down or after all rerouting attempts have failed.

```
set ARtg Pnni CfgNode/<CfgNode_level> svccRetryTime
<svccRetryTime>
```

4 Specify in seconds how long this node will wait for an SVCC which it has initiated as the calling party, to become fully established before giving up and tearing it down.

```
set ARtg Pnni CfgNode/<CfgNode_level>
svccCallingIntegrityTime <svccCallingIntegrityTime>
```

5 Specify in seconds how long this node will wait for an SVCC which it has accepted as the called party, to become fully established before giving up and tearing it down.

```
set ARtg Pnni CfgNode/<CfgNode_level>
svccCalledIntegrityTime <svccCalledIntegrityTime>
```

6 Specify in seconds how long this node will wait for a SVCC SETUP message from its neighbor once it knows of the neighbor’s existence.

```
set ARtg Pnni CfgNode/<CfgNode_level>
svccEstablishmentIntegrityTime
<svccEstablishmentIntegrityTime>
```

7 Specify in seconds how long this node will delay advertising its choice of preferred peer group leader. This delay ensures that the topology information propagates across the peer group.

```
set ARtg Pnni CfgNode/<CfgNode_level> Pgl initTime
<initTime>
```

8 Specify in seconds how long this node will delay being declared the preferred peer group leader by unanimous agreement among its peers.

```
set ARtg Pnni CfgNode/<CfgNode_level> Pgl
overrideDelay <overrideDelay>
```

9 Specify in seconds how long this node re-starting the process of electing a new peer group leader after it loses its connectivity to the current peer group leader.

```
set ARtg Pnni CfgNode/<CfgNode_level> Pgl
reElectionInterval <reElectionInterval>
```

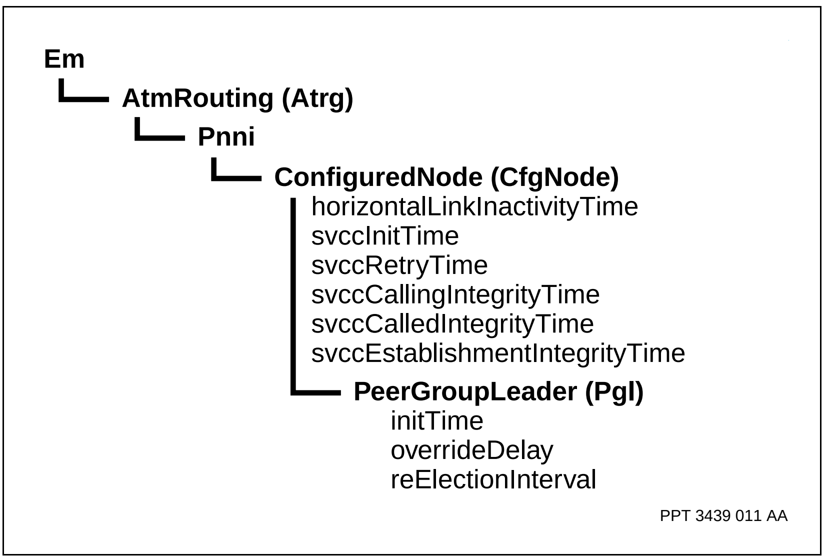
Variable definitions

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 step 1. In a hierarchical PNNI network, The system defines level 0 as the top-most level and level 104 as the bottom-most level.
<horizontalLinkInactivityTime>	is a number of seconds in the range from 30 to 3 600 inclusive. The default is 120.
<overrideDelay>	is a number of seconds in the range 1 to 65 535 inclusive. The default is 30.
<pglInitTime>	is a number of seconds in the range 0 to 65 535 inclusive. The default is 15.
<reElectionInterval>	is a number of seconds in the range 1 and 65 535 inclusive. The default is 15.
<svccCalledIntegrityTime>	is a number of seconds in the range 25 and 3 615 inclusive. The default is 50.
(Sheet 1 of 2)	

Variable	Value
<svccCallingIntegrityTime>	is a number of seconds in the range 10 and 3 600 s inclusive. The default is 35.
<svccEstablishmentIntegrityTime>	is either a number of seconds in the range 15 and 3 600 inclusive, or has the value sameAsPnni. If the value is sameAsPnni, then the actual value is taken from the ARtg Pnni svccEstablishmentIntegrityTime. The default is sameAsPnni.
<svccInitTime>	is a number of seconds in the range 1 and 120 inclusive. The default is 4.
<svccRetryTime>	is a number of seconds in the range 20 and 60 inclusive. The default is 30.
(Sheet 2 of 2)	

Procedure job aid

Figure 74
PNNI timing parameters for a specified PNNI level component hierarchy



Configuring PNNI node thresholds

Configure PNNI node thresholds to change the available cell rate thresholds for the node.

> ***Note:*** It is recommended that you set the nodal thresholds to the minimum value of the link thresholds. This allows the link information to be flooded by the node. In the case of hierarchical PNNI, the available cell rate information for outside links are exchanged by higher level nodes; therefore, the nodal thresholds for the higher level nodes need to be set to the minimum threshold value of all the outside links.

Procedure steps

1 Specify the percentage used to determine the minimum threshold for a significant change in the available cell rate.

```
set ARtg Pnni avcrMt <avcrMt>
```

2 Specify the percentage used to determine a significant change in available cell rate. This value is multiplied by the current available cell rate. If the resulting number is lower than the minimum threshold (the *avcrMt* attribute), the minimum threshold is used.

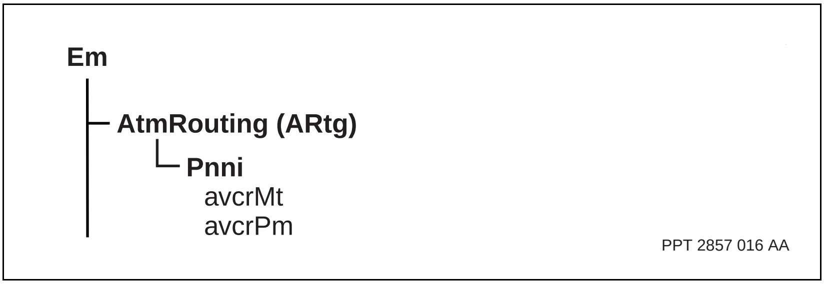
```
set ARtg Pnni avcrPm <avcrPm>
```

Variable definitions

Variable	Value
<avcrMt>	is a percentage from 1 to 99. The default value is 3 percent.
<avcrPm>	is a percentage from 1 to 99. The default value is 50 percent.

Procedure job aid

Figure 75
PNNI node thresholds component hierarchy



Configuring PNNI link thresholds

Configure PNNI link thresholds to set the available cell rate thresholds on a per link basis.

Procedure steps

1 Specify the percentage used to determine the minimum threshold for a significant change in the available cell rate on the link. Set the *avcrMt* component to the same value on both ends of the link.

```
set Atmif/n Pnni avcrMt <avcrMt>
```

2 Specify the percentage used to determine a significant change in available cell rate. This value is multiplied by the current available cell rate. If the resulting number is lower than the minimum threshold (the *avcrMt* attribute multiplied by MaxAvcr), the minimum threshold is used. Set the *avcrPm* component to the same value at both ends of the link.

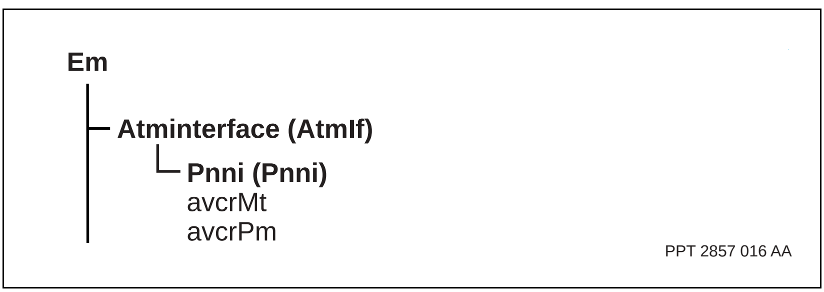
```
set Atmif/n Pnni avcrPm <avcrPm>
```

Variable definitions

Variable	Value
<avcrMt>	is a percentage from 1 to 99. The default value is the same value as the one provisioned under *Artg Pnni*.
<avcrPm>	is a percentage from 1 to 99. The default value is the same value as the one provisioned under *Artg Pnni*.

Procedure job aid

Figure 76
PNNI link thresholds component hierarchy



Configuring PNNI load balancing

Configure PNNI load balancing to set the load balancing method for each ATM service category, thus facilitating a uniform utilization of the network bandwidth.

Procedure steps

1 Set the load balancing method for each ATM service category.

```
set ARtg Pnni Lb/<ATM_service_category> method
<method>
```

2 Set the minimum acceptable load balancing variance for the applicable optimization metrics (aw, cdv and ctd) for each ATM service category.

```
set ARtg Pnni Lb/<ATM_service_category> minVar aw
<minVar_aw>
cdv <minVar_cdv> ctd <minVar_ctd>
```

If both the *minVar* and slopeVar attributes are set to zero for an optimization metric and ATM service category, the multi-path variance functionality is disabled for that optimization metric and ATM service category.

3 Set the slope of the load balancing acceptable variance for the three optimization metrics (aw, cdv and ctd) for each ATM service category.

```
set ARtg Pnni Lb/<ATM_service_category> slopeVar aw
<slopeVar_aw> cdv <slopeVar_cdv> ctd <slopeVar_ctd>
```

4 Set the number of alternate paths that the on-demand route computation algorithm will compute.

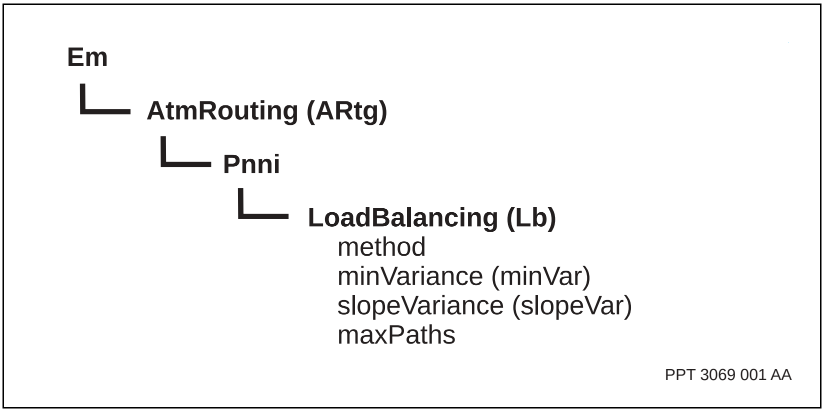
```
set ARtg Pnni Lb/<ATM_service_category> maxPaths
<maxPaths>
```

Variable definitions

Variable	Value
<ATM_service_category>	is the ATM service category you want to set.
<maxPaths>	is the number of alternate paths. If the *Cache* subcomponent is configured, the maxPaths attribute also specifies the number of alternate paths to a destination node stored in the PNNI route cache. By default, the maxPaths attribute is set to the value of 2. When the *maxPaths* attribute is greater than 3, the *method* attribute can only be set to random for each service category. If method is maxAvCr, the lb maxPaths must be 1.
<method>	is the value specifying the method of load balancing under the *LoadBalancing* subcomponent. Values of the method attribute are maxAvCr, random, avCrProb, optMetricProb, and avCrOptMetricProb.For more details on configuring maxAvCr, see <insert x-ref to Configuring load balancing on available cell rate (AvCr).
<minVar_aw>	is a decimal value that represents the minimum variance for the administrative weight optimization metric.
<minVar_cdv>	is a decimal value that represents the minimum variance for the cell delay variation optimization metric.
<minVar_ctd>	is a decimal value that represents the minimum variance for the cell transfer delay optimization metric.
<slopeVar_aw>	is a percentage value that represents the slope of the load balancing acceptable variance for the administrative weight optimization metric.
<slopeVar_cdv>	is a percentage value that represents the slope of the load balancing acceptable variance for the cell delay variation optimization metric.
<slopeVar_ctd>	is a percentage value that represents the slope of the load balancing acceptable variance for the cell transfer delay optimization metric.

Procedure job aid

Figure 77
PNNI load balancing component hierarchy



Configuring PNNI topology state element timer parameters

Configure PNNI topology state element timer parameters to change the defaults setting for this option.

Procedure steps

1. Specify the rate in seconds at which the node sends out PTSE packets.

   `set ARtg Pnni ptseHoldDown <psteH>`

2. Specify the default duration in seconds of the PTSE timer. When this timer expires, the node sends out a PTSE packet to the neighboring node.

   `set ARtg Pnni ptseRefreshInterval <ptseR>`

3. Specify how long the information in the PTSE packet is valid. The number in this attribute multiplied with the value in the *ptseRefreshInterval* attribute is the initial lifetime that this node places into PTSE packets.

   `set ARtg Pnni ptseLifetimeFactor <ptseL>`

4. Specify the duration in seconds between the retransmission of unacknowledged database summary packets, PTSE request packets and PNNI topology state packets (PTSPs). See NN10600-005 *Nortel Networks Multiservice Switch 7400/15000/20000 Terminology* for a description of these terms.

   `set ARtg Pnni requestRxMtInterval <request>`

5. Specify the minimum amount of time in seconds between transmission of delayed PTSE acknowledgement packets.

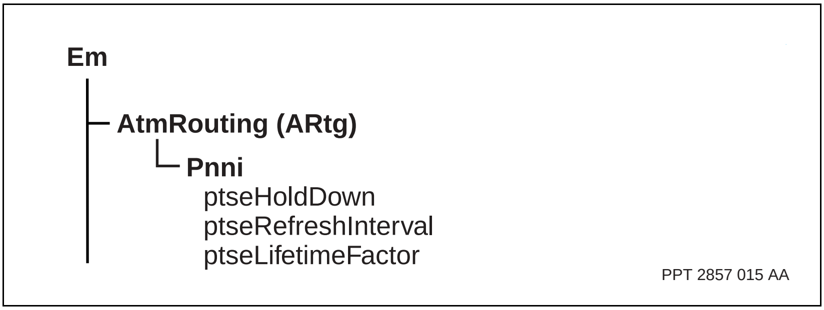
   `set ARtg Pnni peerDelayedAckInterval <peer>`

Variable definitions

Variable	Value
<peer>	is a number of seconds between 0.1 and 65 535. The default is 0.1.
<psteH>	is a number of seconds between 0.1 and 65 535. The default is 1.
<ptseL>	is a number between 101 and 1001. The default value is 200.
<ptseR>	is a number of seconds between 300 and 65 535. The default is 1 800.
<request>	is a number of seconds between 1 and 65 535. The default is 5.

Procedure job aid

Figure 78
PNNI topology state element timer parameters component hierarchy



Configuring PNNI route caching

Configure PNNI route caching to store and maintain multiple alternate routes to several destinations.

Procedure steps

1 Add the *Cache* component.

add ARtg Pnni Cache

2 Set the aging period for the route cache entries.

set ARtg Pnni Cache agingPeriod <agingPeriod>

3 Set the maximum number of entries in the cache.

set ARtg Pnni Cache maxNumEntries <maxNumEntries>

4 Set the maximum number of equal cost paths that are kept to an intermediate node during on-demand route computations.

set ARtg Pnni Cache maxPathToIntermediateNode <maxPathToIntermediateNode>

5 Set whether or not to purge the whole route cache entry when a link becomes invalid.

set ARtg Pnni Cache purgeEntryAtPathInvalid <true/false>

6 Set the number of alternate paths that are stored in the cache entry for each destination node.

set ARtg Pnni Lb/<ATM_service_category> maxPaths <maxPaths>

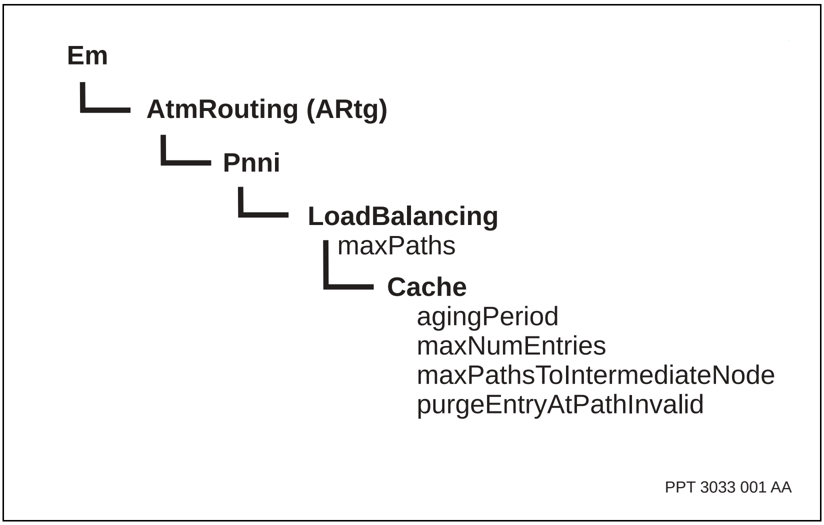
Variable definitions

Variable	Value
<agingPeriod>	is a value specifying the lifetime of a cache entry in seconds.
<maxNumEntries>	is the maximum number of entries in the route cache. The maximum value is 30 000/Lb maxPaths.
<maxPaths>	is the number of alternate paths. The maximum value is 1-20. If the method is maxAvCr, the lb maxPaths must be 1.
(Sheet 1 of 2)	

Variable	Value
<maxPathToIntermediateNode>	is the maximum number of equal cost paths that are kept to an intermediate node during on-demand route computations. The default value is 1.
<true/false>	specifies whether or not to purge the whole route cache entry.
(Sheet 2 of 2)	

Procedure job aid

Figure 79
PNNI route caching component hierarchy



Configuring Hello protocol options

Configure Hello protocol options to change the default values of the Hello parameters at the node level or for each ATM interface. The *AtmRouting Pnni* component specifies the Hello parameters for all the PNNI interfaces. However, each PNNI interface can overwrite the Hello parameters by specifying new values in the *AtmIf Pnni Rcc* component.

Procedure steps

1 Specify the rate in seconds at which the node sends out Hello packets.

```
set ARtg Pnni helloHoldDown <hhd>
```

2 Specify the default duration in seconds of the Hello timer. When this timer expires, the node sends out a Hello message to the neighboring nodes.

```
set ARtg Pnni helloInterval <hli>
```

3 Specify the default number of intervals allowed to pass without receiving a Hello packet from the next node before this node considers the link inactive.

```
set ARtg Pnni helloInactivityFactor <hif>
```

4 Specify the rate in seconds at which the interface sends out Hello packets.

```
set AtmIf/<n> Pnni Rcc helloHoldDown <hhd>
```

The default is 0.0 seconds which means that the system uses the same value as in the *helloHoldDown* attribute of the *AtmRouting Pnni* component.

5 Specify the default duration in seconds of the Hello timer. When this timer expires, the interface sends out a Hello message to the neighboring nodes.

```
set AtmIf/<n> Pnni Rcc helloInterval <hli>
```

The default is 0 seconds which means that the system uses the same value as in the *helloInterval* attribute of the *AtmRouting Pnni* component.

6 Specify the default number of Hello intervals allowed to pass without this interface receiving a Hello packet from the neighboring node.

```
set AtmIf/<n> Pnni Rcc helloInactivityFactor <hif>
```

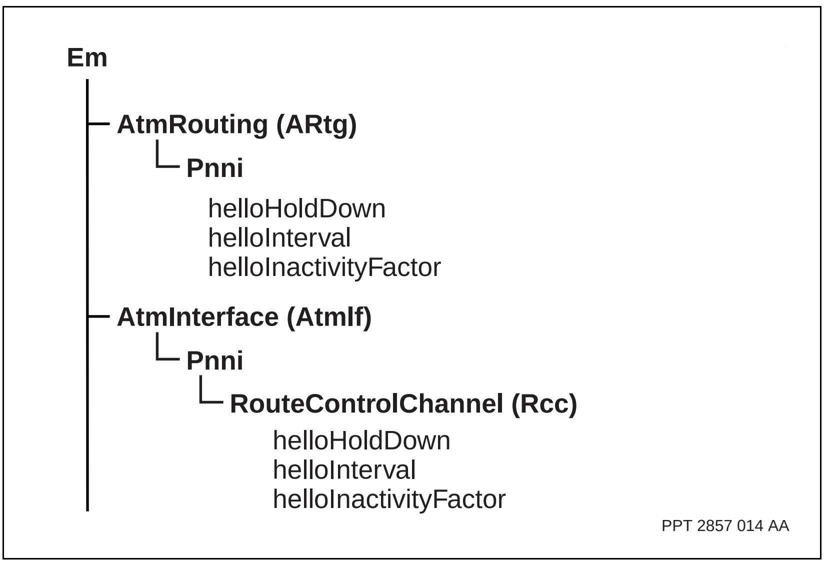
The default is 0 seconds which means that the system uses the same value as in the *helloInterval* attribute of the *AtmRouting Pnni* component.

Variable definitions

Variable	Value
<hhd>	is a number of seconds between 0.1 and 65 535.0 with up to one decimal place allowed. The default is 1.0. When large connection spaces are switched across PNNI links, this parameter must be increased to allow for delays in receiving packets due to signaling congestion.
<hif>	is a number of seconds between 1 and 65 535. The default value is 5. When large connection spaces are switched across PNNI links, this parameter must be increased to allow for delays in receiving packets due to signaling congestion.
<hli>	is a number of seconds between 1 and 65 535. The default is 15. When large connection spaces are switched across PNNI links, this parameter must be increased to allow for delays in receiving packets due to signaling congestion.
<n>	is the instance value of the *AtmIf* component and can be any unique value from 1 to 4 095.

Procedure job aid

Figure 80
Hello protocol options component hierarchy



Configuring load balancing for an available cell rate

Configure load balancing for an available cell rate (AvCr).

Prerequisites

- Review the default values for the *avcrMt* and *avcrPm* attributes for both the nodal and link levels to ensure that the values are suitable for the network. If changes are required, refer to “Configuring PNNI node thresholds” (page 263) and “Configuring PNNI link thresholds” (page 265).

  Note: If the route cache is provisioned and the maxAvCr method is provisioned, then only the best acceptable path is cached. It is not recommend that the route cache be used in combination with the maxAvCr load balancing method.

Procedure steps

1 Set the load balancing method for the available cell rate (maxAvCr).

```
set ARtg Pnni Lb/<ATM_service_category> maxAvCr
```

2 Set the number of alternate paths that the on-demand route computation algorithm will compute.

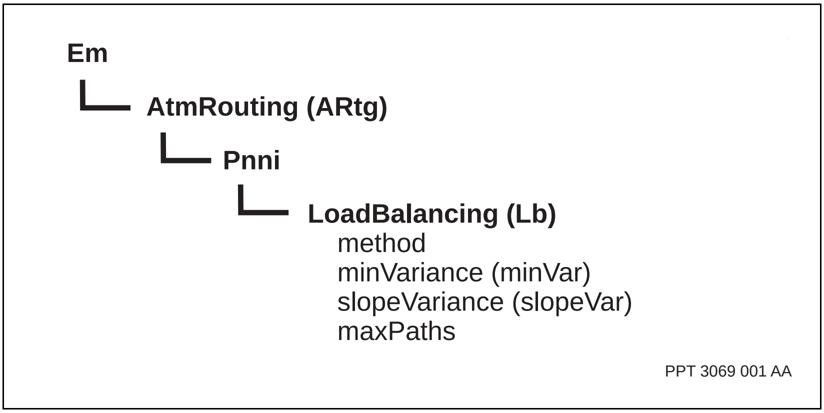
```
set ARtg Pnni Lb maxPaths 1
```

Variable definitions

Variable	Value
<ATM_service_category>	is the ATM service category you want to set.

Procedure job aid

Figure 81
Load balancing for an available cell rate component hierarchy



Connection recovery and path optimization configuration

To configure the connection recovery and path optimization capability, you must

- Install the rerouting feature on each ATM card and node that will act as one of the following nodes:
  - rerouting node (source node)
  - rendezvous node (destination node)
  - local/global rerouting node (entry border node into a local or global rerouting domain that does not include the source node)
  - local/global rendezvous node (exit border node leaving a local or global rerouting domain that does not include the destination node)
- Configure the nodal defaults for all rerouting/rendezvous nodes by
  - selecting the local domain value which will determine the local rerouting domain boundaries
  - defining the default rerouting protocol and rerouting services to be used by connections that are sourced off this node, the switchover mechanism and the type of connections optimized (recovered, all subscribed or none) during a module optimization. Note that depending on the signaling protocol used by a given connection, the rerouting protocol/services subscribed to by that connection may be signalled across the network (use the same as provisioned on the source node, (for example: PNNI) or will change and use the default settings on the first entry border node into another local/global domain, (for example: IISP).
- Add the reroute component to the signaling interface that will source a connection. By default, the signaling interface will use the rerouting attribute values provisioned at the nodal level to determine rerouting protocol and rerouting services available to various connection types and the switchover mechanism. However, the operator has the ability to override all of these settings at the interface level by optionally configuring a *Reroute Ov* component.

- Optionally configure a *RerouteOv* component for a soft PVC or soft PVP. The *RerouteOv* component enables the operator to override the rerouting capabilities defined for the interface on the basis of individual connections.

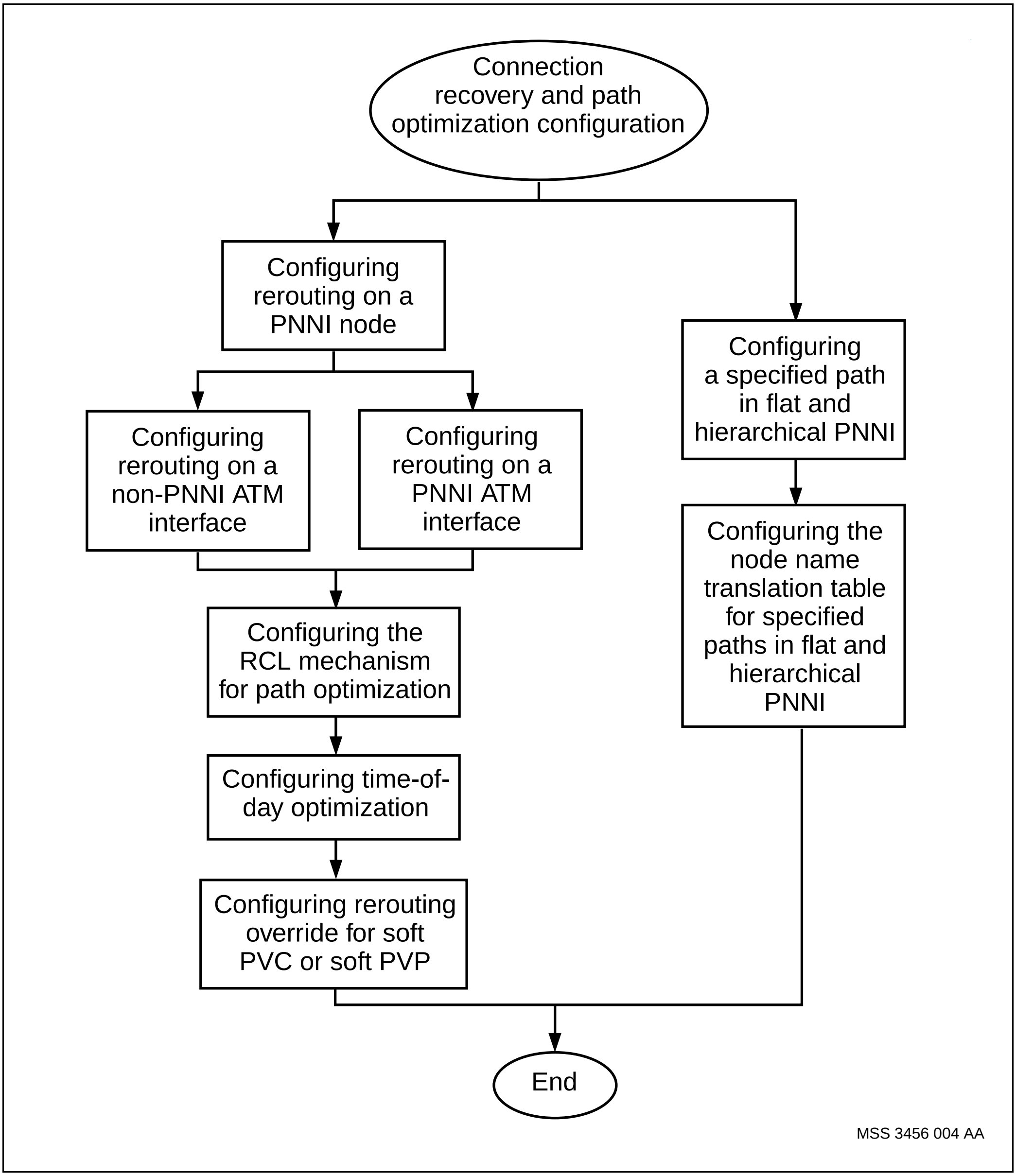
Prerequisites to connection recovery and path optimization configuration

- To configure connection recovery on a signaling interface, you can set the *localConnectionRecovery* or globalConnectionRecovery attribute to turn on connection recovery for configured connections. To configure path optimization, you can set the *localPathOptimization* or globalPathOptimization attribute to turn on path optimization for configured connections. The default is *sameAsARtgPnniReroute* for both attributes.
- When you configure rerouting for a signaling interface, a dynamic *RerouteInfo* component appears under the *Vcc* and *Vpc* components for each affected connection. For a soft PVC or soft PVP, you can optionally configure an *RerouteOv* component. The *RerouteOv* component allows you to override the rerouting capabilities defined for the interface on the basis of individual connections.
- Changes to the configured attributes are not critical to the connections set up under the *AtmIf* component. Existing connections retain the old subscription options and new connections are assigned the new options. However, changes to the *RerouteOv* component are not critical.
- When rerouting capabilities are first provisioned on the node/interface, they will only apply to new connections or connections that are re-established.

Connection recovery and path optimization configuration procedures

This task flow shows you the sequence of procedures you perform to configure connection recovery and path optimization. To link to any procedure, go to “Connection recovery and path optimization configuration procedure navigation” (page 280).

Figure 82
Connection recovery and path optimization configuration procedures



Connection recovery and path optimization configuration procedure navigation

Configuring rerouting on a PNNI node

Configure rerouting on a PNNI node to enable connection recovery and path optimization.

Procedure steps

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 Set the local domain identifier value.

```
set ARtg Pnni reroute localDomain <localDomain>
```

Variable definitions

Variable	Value
<localDomain>	is the value of the local domain being configured (unspecified, intraDomain, or a numeric value from 2-1024).
<Lpt>	is any name (for example, Reroute).
<n>	is the number of the LP being configured.

Procedure job aid

Figure 83
Rerouting on a PNNI node component hierarchy

- **Em**
  - **AtmRouting (Atrg)**
    - **Pnni**
      - **Reroute**
        - protocol
        - localDomain
        - numOptAttempts
        - moduleOptimizationConnections
        - switchoverMechanism
        - localConnectionRecoverySubscr
        - localPathOptimizationSubscr
        - globalConnectionRecoverySubscr
        - globalPathOptimizationSubscr
        - optimizationTime

PPT 3439 012 AA

Configuring rerouting on a non-PNNI ATM interface

Configure rerouting on a non-PNNI ATM interface to define the path optimization capability by overriding nodal default rerouting attribute values on the ATM interface level (including VPT).

> ***Note:*** 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

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 as controlled by Artg Pnni Reroute.

```
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>
```

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.
<IfType>	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 <IfType> 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.
(Sheet 1 of 2)	

Variable	Value
<moduleOptimizationConnections>	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).
<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).
(Sheet 2 of 2)	

Procedure job aid

Figure 84
Rerouting on a non-PNNI ATM interface component hierarchy

- **Em**
  - **AtmRouting (Atrg)**
    - **Uni/Iisp/Aini**
      - **Reroute**
        - **Override (Ov)**
          - protocol
          - numOptAttempts
          - moduleOptimizationConnections
          - switchoverMechanism
          - localConnectionRecoverySubscr
          - localPathOptimizationSubscr
          - globalConnectionRecoverySubscr
          - globalPathOptimizationSubscr
    - **VirtualPathTerminator (Vpt/x)**
      - **Uni/Iisp/Aini**
        - **Reroute**
          - **Override (Ov)**
            - protocol
            - numOptAttempts
            - moduleOptimizationConnections
            - switchoverMechanism
            - localConnectionRecoverySubscr
            - localPathOptimizationSubscr
            - globalConnectionRecoverySubscr
            - globalPathOptimizationSubscr

PPT 3439 015 AA

Configuring rerouting on a PNNI ATM interface

Configure rerouting on a PNNI ATM interface to define the path optimization capability by overriding nodal default rerouting attribute values on the ATM interface level (including VPT).

> ***Note:*** 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

1 Add the *Reroute* component to the ATM interface.

add AtmIf/<n> Pnni Reroute

2 Set the method used to identify the type of local domain link.

set AtmIf/<n> Pnni Reroute Ov localDomainLink <domain>

3 Set the default rerouting protocol used by the interface.

set AtmIf/<n> Pnni Reroute Ov protocol <protocol>

4 Set the number of path optimizations attempted before declaring a connection optimally routed.

set AtmIf/<n> Pnni Reroute Ov numOptAttempts <numOptAttempts>

5 Set the connections that will be considered by a module optimization pass as controlled by Artg Pnni Reroute.

set AtmIf/<n> Pnni Reroute Ov moduleOptimizationConnections <moduleOptimization>

6 Set the mechanism used for switchover for the rerouted connection segment.

set AtmIf/<n> Pnni Reroute Ov switchoverMechanism <switchoverMechanism>

7 Set which connection is requesting subscription to local connection recovery.

set AtmIf/<n> Pnni Reroute Ov localConnectionRecoverySubscr <lConRec>

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

set AtmIf/<n> Pnni Reroute Ov localPathOptimizationSubscr <lPathOpt>

9 Set which connection is requesting subscription to connection recovery.

set AtmIf/<n> Pnni Reroute Ov globalConnectionRecoverySubscr <gConRec>

10 Set which connection is requesting subscription to optimization.

set AtmIf/<n> Pnni Reroute Ov globalPathOptimizationSubscr <gPathOpt>

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.
<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.
<moduleOptimizationConnections>	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.
(Sheet 1 of 2)	

Variable	Value
<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).
<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).
(Sheet 2 of 2)	

Procedure job aid

Figure 85
Rerouting on a PNNI ATM interface component hierarchy

- **Em**
  - **AtmRouting (Atrg)**
    - **Pnni**
      - **Reroute**
        - **Override (Ov)**
          localDomainLink
          protocol
          numOptAttempts
          moduleOptimizationConnections
          switchoverMechanism
          localConnectionRecoverySubscr
          localPathOptimizationSubscr
          globalConnectionRecoverySubscr
          globalPathOptimizationSubscr
    - **VirtualPathTerminator (Vpt/x)**
      - **Pnni**
        - **Reroute**
          - **Override (Ov)**
            localDomainLink
            protocol
            numOptAttempts
            moduleOptimizationConnections
            switchoverMechanism
            localConnectionRecoverySubscr
            localPathOptimizationSubscr
            globalConnectionRecoverySubscr
            globalPathOptimizationSubscr

PPT 3439 016 AA

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 optimization.

Prerequisites

- Rerouting must be configured on all the rerouting and rendezvous nodes. See “Configuring rerouting on a PNNI node” (page 281), “Configuring rerouting on a non-PNNI ATM interface” (page 283) and “Configuring rerouting on a PNNI ATM interface” (page 287).
- Only the Ebr and Global rerouting protocols support the RCL mechanism.

Procedure steps

1 Set the reduced cell loss mechanism.

set AtmIf/<n> <IfType> reroute ov switchoverMechanism rcl

2 Activate the segment OAM boundary at the ATM interface.

set AtmIf/<n> oamSegmentBoundary yes

3 If required, revert from the enhanced rerouting capability (where the RCL mechanism is activated) to the standards-based rerouting capability.

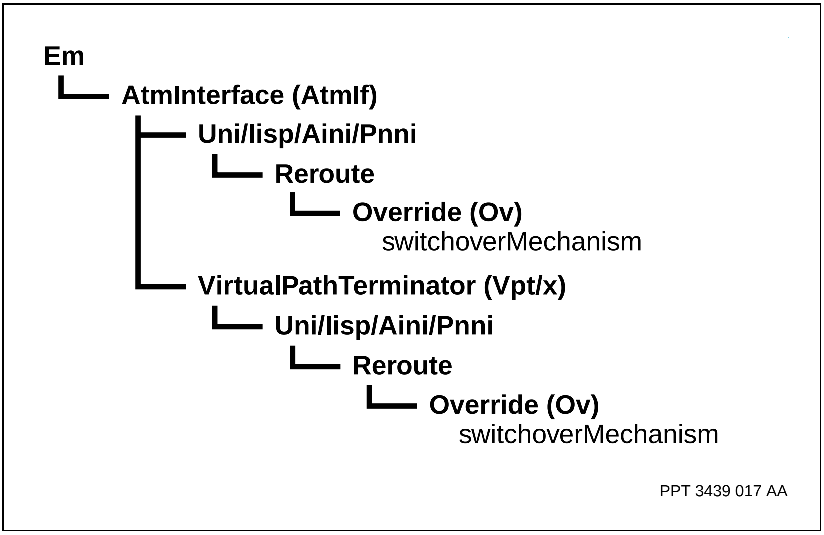
set AtmIf/<n> <IfType> Reroute Override switchoverMechanism standard

Variable definitions

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

Procedure job aid

Figure 86
RCL mechanism for path optimization component hierarchy



Configuring time-of-day optimization

Configure time-of-day optimization to set the times at which a module optimization pass will occur. 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.

For example, if a module optimization lasts for six minutes from start to completion, you can configure the time intervals at less than six minute intervals and some intervals will be skipped. For example, if you configure five minute time intervals, at 10:05, 10:10, and 10:15, the first interval will run until completion and the second interval, 10:10 will be skipped. Therefore, the next scheduled module optimization will begin at 10:15.

Procedure steps

1 Set the time or times at which a module optimization pass will occur in a 24 hour period.

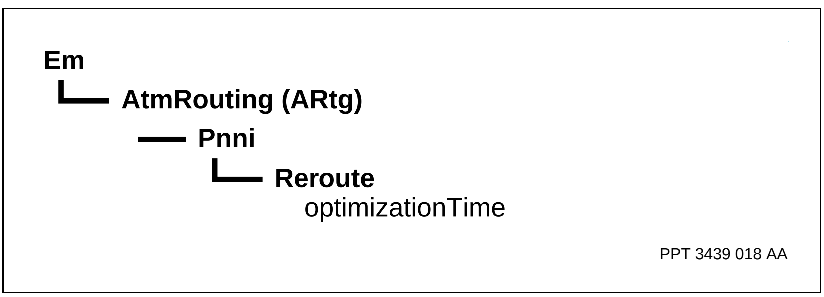
```
set ARtg Pnni Reroute optimizationTime ! <hour:min>
```

Variable definitions

Variable	Value
<hour:min>	is the time expressed in terms of the 24 hour clock.

Procedure job aid

Figure 87
Time-of-day optimization component hierarchy



Configuring a specified path in flat and hierarchical PNNI

Configure a specific path in flat and hierarchical PNNI as part of configuring paths.

Prerequisites

- See “Configuring the node name translation table for specified paths in flat and hierarchical PNNI” (page 296).

Procedure steps

1 Add the *MdtlPath* component.

add ARtg Pnni MdtlPath/<MdtlPath>

2 Add the *Hop* component.

add ARtg Pnni MdtlPath/<MdtlPath> Hop/<Hop>

3 Link a *nodeName* component to the hop.

set ARtg Pnni MdtlPath/<MdtlPath> Hop/<Hop> nodeName/<nodename>

4 Optionally, specify the port ID of the exit link using the node name translation table.

set ARtg Pnni MdtlPath/<MdtlPath> Hop/<Hop> portId/<portId>

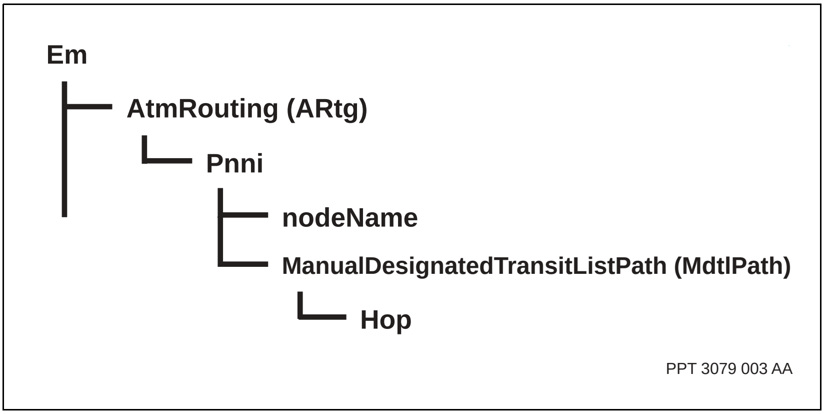
Variable definitions

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 provisioned on a source node is 512 paths per shelf.
(Sheet 1 of 2)	

Variable	Value
<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/*.
(Sheet 2 of 2)	

Procedure job aid

Figure 88
Specified path in flat and hierarchical PNNI component hierarchy



Configuring the node name translation table for specified paths in flat and hierarchical PNNI

Configure the node name translation table for specified paths in flat and hierarchical PNNI as part of configuring paths.

Prerequisites

- Ensure that the following nodes have PCR4.2 or later software loaded:
  - source node
  - any exit border node
  - the twentieth hop node along the given specified path

Procedure steps

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.

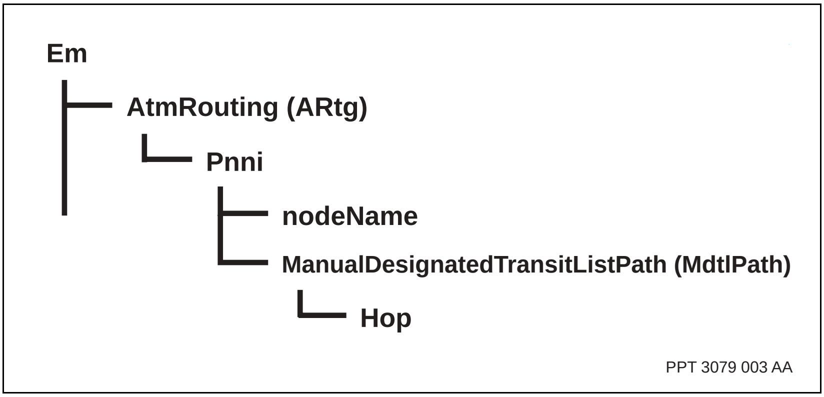
```
set ARtg Pnni nodeName/<nodename> nodeId
```

Variable definitions

Variable	Value
<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. It is recommended to use the nodeName component under the moduleData component.

Procedure job aid

Figure 89
Node name translation table for specified paths in flat and hierarchical PNNI component hierarchy



Configuring a rerouting override for a soft PVC or soft PVP

Configure a rerouting override for a soft PVC or soft PVP on the interface using default routing attributes, if required.

Procedure steps

CAUTION
Perform this procedure during maintenance only
It is recommended that you perform this procedure only during maintenance windows. Configuring the RerouteOv component on a soft PVC or soft PVP is a critical change that causes the connection to restage.

1 Add the *RerouteOv* component to the VCC.

```
add AtmIf/<n> Vcc/<Vpi.Vci> Src RerouteOv
```

2 Set which connection is requesting subscription to local connection recovery.

```
set AtmIf/<n> Vcc/<Vpi.Vci> Src RerouteOv
localConnectionRecoverySubscr yes|no
```

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

```
set AtmIf/<n> Vcc/<Vpi.Vci> Src RerouteOv
localPathOptimizationSubscr yes|no
```

4 Set which connection is requesting subscription to connection recovery.

```
set AtmIf/<n> Vcc/<Vpi.Vci> Src RerouteOv
globalConnectionRecoverySubscr yes|no
```

5 Set which connection is requesting subscription to path optimization.

```
set AtmIf/<n> Vcc/<Vpi.Vci> Src RerouteOv
globalPathOptimizationSubscr yes|no
```

Variable definitions

Variable	Value
<n>	is the number of the ATM interface.
<no>	specifies that this soft PVC does not subscribe to connection recovery or path optimization.
<Vpi.Vci>	is the instance value of the *Vcc* component. The Vpi value can be from 0 to 255. The Vci value can be from 32 to 65 535.
	The VCC could also be a VPC or VPT VCC.
<yes>	specifies that this soft PVC subscribes to connection recovery or path optimization (This is the default value.)

Procedure job aid

Figure 90
Rerouting override for a soft PVC or soft PVP component hierarchy

- **Em**
  - **AtmInterface (AtmIf)**
    - **VirtualChannelConnection (Vcc/x.y)**
      - **SourcePvc (Src)**
        - **RerouteOverride (RerouteOv)**
          protocol
          localConnectionRecoverySubscr
          localPathOptimizationSubscr
          globalConnectionRecoverySubscr
          globalPathOptimizationSubscr
    - **VirtualPathConnection (Vpc/y)**
      - **SourcePvc (Src)**
        - **RerouteOverride (RerouteOv)**
          protocol
          localConnectionRecoverySubscr
          localPathOptimizationSubscr
          globalConnectionRecoverySubscr
          globalPathOptimizationSubscr
    - **VirtualPathTerminator (Vpt/x)**
      - **VirtualChannelConnection (Vcc/y)**
        - **SourcePvc (Src)**
          - **RerouteOverride (RerouteOv)**
            protocol
            localConnectionRecoverySubscr
            localPathOptimizationSubscr
            globalConnectionRecoverySubscr
            globalPathOptimizationSubscr

PPT 3439 019 AA

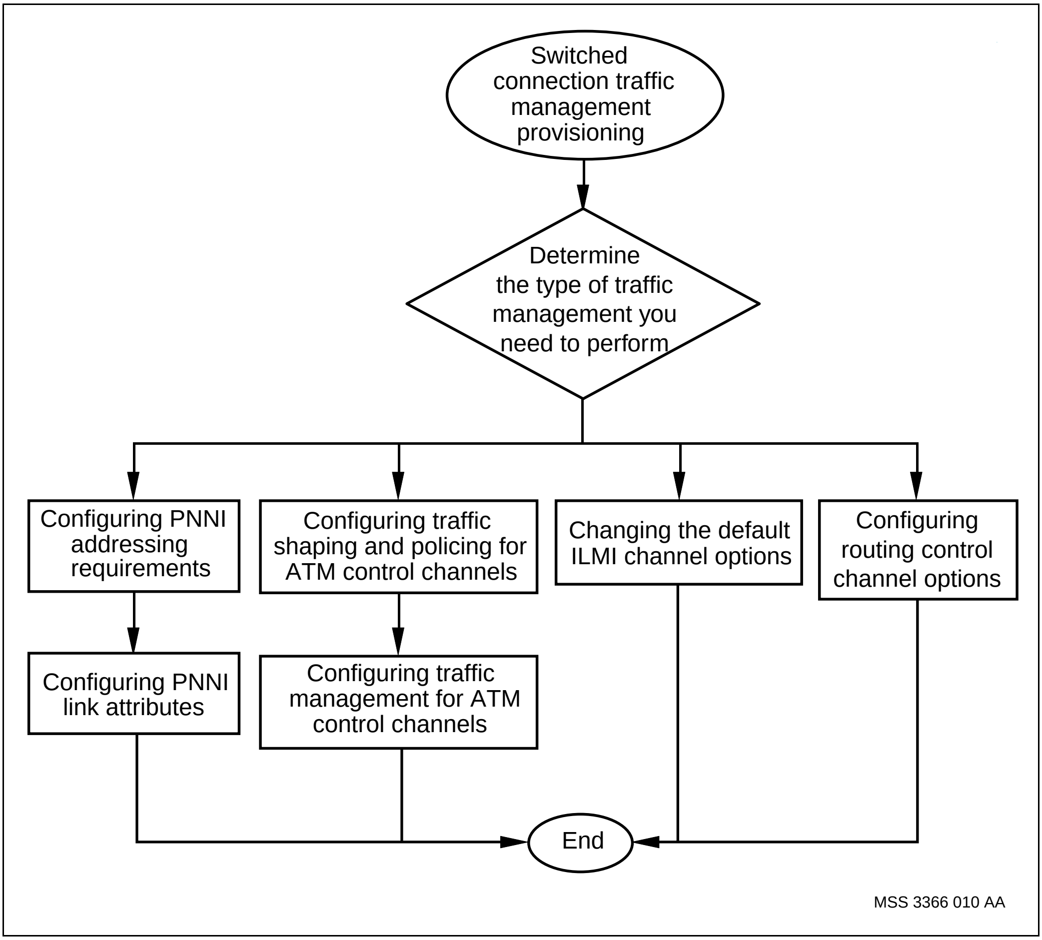
Chapter 8
Switched connection traffic management provisioning

Provision traffic management for switched connections to change node and network configuration for signaling and routing control channels, traffic management, and addressing. The procedures are optional and include changes that you do not need to begin with, but require later as the network expands or as end-user requirements become more demanding.

Switched connection traffic management provisioning procedures

This task flow shows you the sequence of procedures you perform to provision traffic management for switched connections. To link to any procedure, go to “Switched connection traffic management provisioning procedure navigation” (page 302).

Figure 91
Switched connection traffic management provisioning procedures



Switched connection traffic management provisioning procedure navigation

- “Configuring PNNI addressing requirements” (page 304)
- “Configuring PNNI link attributes” (page 306)
- “Configuring traffic shaping and policing for ATM control channels” (page 307)

Configuring PNNI addressing requirements

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

Procedure steps

1 Add a *PnniInfo* component. The *PnniInfo* component contains provisioned information associated with the ATM address related to PNNI. You add this component only under provisioned *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>

Variable definitions

Variable	Value
<addr_type>	is primary or alternate. The default is primary.
<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 provision multiple static addresses for each interface type, but all address entries must be unique.
<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

Figure 92
PNNI addressing requirements component hierarchy

- **Em**
  - **AtmInterface (AtmIf)**
    - **<IfType>**
      - **Address (Addr)**
        - **PnniInfo**
          - scope
          - reachability

<IfType> is Uni, Iisp, Aini or Pnni

PPT 2851 014 AA

Configuring PNNI link attributes

Configure PNNI link attributes to change the default values of these variables.

Procedure steps

1 Define the administrative weights for each service category in the *AdminWeights* attribute group of the *Pnni* component.

set AtmIf/<n> Pnni cbrWeight <cbr_value>

set AtmIf/<n> Pnni rtVbrWeight <rtVbr_value>

set AtmIf/<n> Pnni nrtVbrWeight <nrtVbr_value>

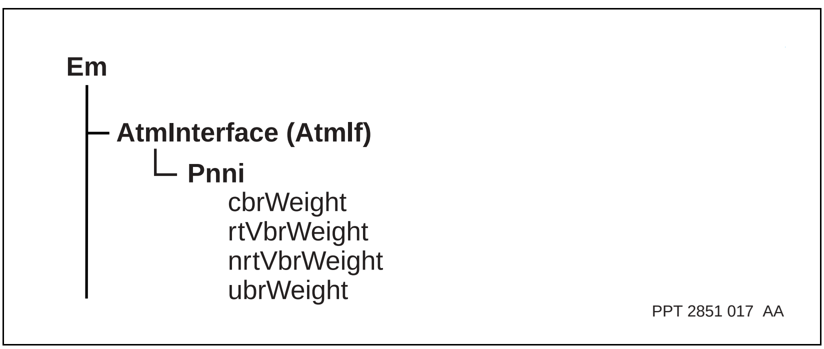
set AtmIf/<n> Pnni ubrWeight <ubr_value>

Variable definitions

Variable	Value
<cbr_value>, <rtVbr_value>, <nrtVbr_value>, and <ubr_value>	are decimal entries between 0 and 4 294 967 295. 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.

Procedure job aid

Figure 93
PNNI link attributes component hierarchy



Configuring traffic shaping and policing for ATM control channels

Configure traffic shaping and policing for ATM control channels to control the traffic flow for these channels. If you enable traffic shaping or policing in a service category, you can disable it for signaling or routing control channels.

Prerequisites

- For APC-based FPs, ensure that all ATM interfaces are configured as either UNIs or PNNIs (all <IfType> must be configured to have the value Uni or Pnni).
- For information on traffic shaping and policing concepts, see NN10600-706 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Shaping and Policing Fundamentals*.

Procedure steps

1 Add the *VirutalChannelDescriptor* component to the signaling channel.

```
add AtmIf/<n> <IfType> <Channel> Vcd
```

2 Specify if this connection uses traffic shaping when it transmits traffic to the ATM interface.

```
set AtmIf/<n> <IfType> <Channel> Vcd trafficShaping
<trShaping>
```

3 Specify if this connection enforces usage parameter control (UPC) for when it receives traffic from the ATM interface.

```
set AtmIf/<n> <IfType> <Channel> Vcd
usageParameterControl <upc>
```

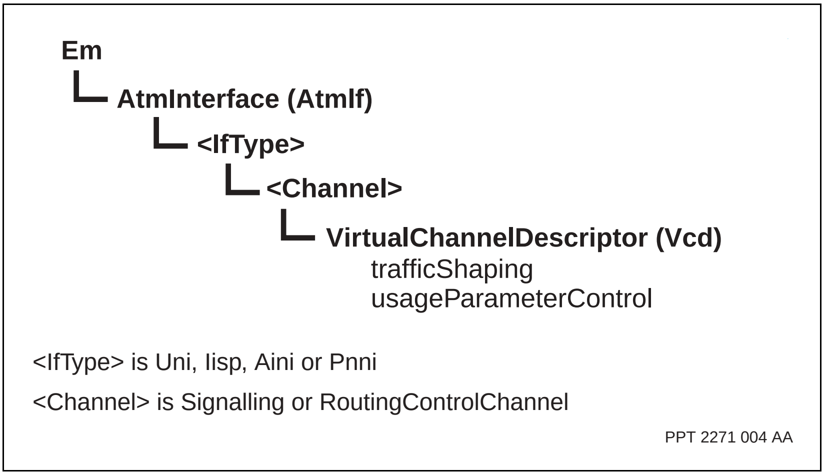
Variable definitions

Variable	Value
<Channel>	is Sig or Rcc. Use the *Sig* component to set the traffic shaping for the signaling channel. Use the *Rcc* component to set traffic shaping for the PNNI routing channel.
<IfType>	is Uni, Iisp, Aini, or Pnni. If the interface is Pnni, you can enable traffic shaping for the signaling or routing control channels (RCC).
(Sheet 1 of 2)	

Variable	Value
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095.
<trShaping>	is sameAsCa or disabled. The default is disabled. Set the value to sameAsCa so this signaling channel uses the value from the *trafficShaping* attribute of the applicable service category for the connection. If you have enabled traffic shaping for the service category at the interface, then traffic shaping is enabled for this connection. Use the default value to turn off traffic shaping for this signaling channel. Enabling traffic shaping has no effect in some instances. When the *atmServiceCategory* attribute is constantBitRate, traffic shaping does not apply regardless of whether you enable or disable it through provisioning.
<upc>	is disabled or sameAsCa. The default is disabled. Set the value to sameAsCa to base UPC on the value in the *usageParameterControl* attribute of the applicable service category for the connection. If you have UPC enabled for the service category at the interface, then UPC is enabled for this connection. 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.
(Sheet 2 of 2)	

Procedure job aid

Figure 94
Traffic shaping and policing for ATM control channels component hierarchy



Configuring traffic management for ATM control channels

Configure traffic management for ATM control channels to optimize traffic management for ATM signaling and routing.

Prerequisites

- For information on traffic management concepts, see NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*.
- See table “Traffic descriptor types and parameters” (page 314) when configuring the *trafficDescType* and *trafficDescParm* attributes. This table shows the relationship between the traffic descriptor type and parameters. Use the information in this table.

Procedure steps

1 Add the *VirutalChannelDescriptor* component to the signaling channel.

add AtmIf/<n> <IfType> <Channel> Vcd

2 Define the traffic descriptor type.

set AtmIf/<n> <IfType> <Channel> Vcd trafficDescType <tdt>

3 For APC-based FPs, define the traffic descriptor parameters.

set AtmIf/<n> <IfType> <Channel> Vcd trafficDescParm 2 <tdp2> 4 <tdp4>

4 For FPs other than APC-based FPs, define the traffic descriptor parameters.

set AtmIf/<n> <IfType> <Channel> Vcd trafficDescParm 1 <tdp1> 2 <tdp2> 3 <tdp3> 4 <tdp4> 5 <tdp5>

5 Define the ATM service category for the control channel.

set AtmIf/<n> <IfType> <Channel> Vcd atmServiceCategory <service>

6 Define transmit queuing method if the channel has traffic shaping disabled.

set AtmIf/<n> <IfType> Sig Vcd unshapedTransmitQueueing <method>

7 Specify the quality of service class for the channel in both the transmit and receive directions.

```
set AtmIf/<n> <IfType> <Channel> Vcd qosClass <class>
```

8 Configure the fairness weighting for the channel.

```
set AtmIf/<n> [Vpt/<Vpi>] Vcc/<x> Vcd Tm weight
<weight>
```

Variable definitions

Variable	Value
<Channel>	is Sig or Rcc. Use the *Sig* component to change the default traffic management parameters for the signaling channel. Use the *Rcc* component to change the traffic management parameters for the PNNI routing channel.
<class>	is 0, 1, 2, 3, or 4. The default is 2. Enter 0 for unspecified bit rate with no performance parameters. Enter 1 for Class A performance requirements (circuit emulation and constant bit rate video) Enter 2 for Class B performance requirements (VBR audio and video). Enter 3 for Class C performance requirements (connection-oriented data transfer). Enter 4 for Class D performance requirements (connectionless data transfer)
<IfType>	is Uni, Iisp, Aini, or Pnni. If the interface is Pnni, you can change the default traffic management parameters for the signaling or routing control channels (RCC).
<method>	is sameAsCa or common. The default is sameAsCa. This attribute specifies queuing if you disable traffic shaping on the signaling channel. When you set this attribute to sameAsCa, the system uses the value from the *AtmIf* component. When you set this attribute to common, the system directs transmit traffic to the common queue.
<n>	is the instance value of the *AtmIf* component, and can be any unique value from 1 to 4 095.
<service>	is constantBitRate, rtVariableBitRate, nrtVariableBitRate, or unspecifiedBitRate. The default is rtVariableBitRate.
(Sheet 1 of 3)	

Variable	Value
<tdp1>, <tdp2>, <tdp3>, <tdp4>, <tdp5>	are vectors of five traffic parameters whose meanings are defined by the *trafficDescType* attribute. For more information, see the table “Traffic descriptor types and parameters” (page 314). These vectors consist of an index entry ranging from 1 to 5 and a decimal entry (<tdp1>, <tdp2>, <tdp3>, <tdp4>, and <tdp5>) with a value between 0 and 2147483647. The decimal entry sets the values for PCR, SCR, MBS or CDVT depending on the type of traffic management defined by the *trafficDescType* attribute. The index entry indicates the traffic descriptor parameter with which the PCR, SCR, MBS or CDVT value is associated. The default values of this vector are *1 0 2 0 3 0 4 0 5 0*. Multiservice Switch nodes use the values of PCR, SCR, MBS and CDVT for connection admission control (CAC). CDVT is used for CAC on CBR connections (*atmServiceCategory* is constantBitRate). For all other values of *atmServiceCategory*, Multiservice Switch nodes in effect do not use CDVT capability on the signaling channel. Equipment from other vendors may use the CDVT parameter. The values of PCR, SCR and requested shaping rate determine the actual shaping rate on connections with traffic shaping enabled. A non-zero value in parameter 5 overrides any value in parameter 1. The system uses this result for the PCR. Parameters 1, 2 and 3 must be non-zero. Parameter 1 must be greater than or equal to parameter 2. Parameter 5, must either be zero (unused) or greater than or equal to parameter 2. Whenever it is valid to specify a PCR, you can also use parameter 5 to specify a requested shaping rate. A non-zero value in parameter 5 overrides the value in parameter 1 and is used as the peak cell rate in calculations of CAC and shaping rate. On CQC cards, when the *trafficDescType* attribute is 3, the Multiservice Switch system shapes traffic at the next rate less than the PCR. When the *trafficDescType* attribute is 6, 7 or 8, the system shapes traffic at the highest available rate that is between PCR and SCR. However, if there is no available shaping rate between PCR and SCR, the system shapes traffic at the next rate above the PCR.
(Sheet 2 of 3)	

Variable	Value
<tdt>	is 3, 6, 7, or 8. The default is 6. This attribute defines the type of traffic management that the system applies to the transmit and receive directions of this connection as defined in the ATM Forum standards. The *trafficDescType* attribute determines the number and meaning of the parameters in the *trafficDescParm* attribute.
<weight>	is sameAsCa, upToQueueLimit, or a decimal from 1 to 4 095. The default is sameAsCa. Use sameAsCa to derive the fairness weighting from the configuration for the *ConnectionAdministrator* component. Use upToQueueLimit to limit fairness weighting by the transmit queue limit specified in the *txQueueLimit* attribute. Use a decimal when you need to define relative fairness between a number of connections under the interface.
(Sheet 3 of 3)	

Procedure job aid

Figure 95
Traffic management for ATM control channels component hierarchy

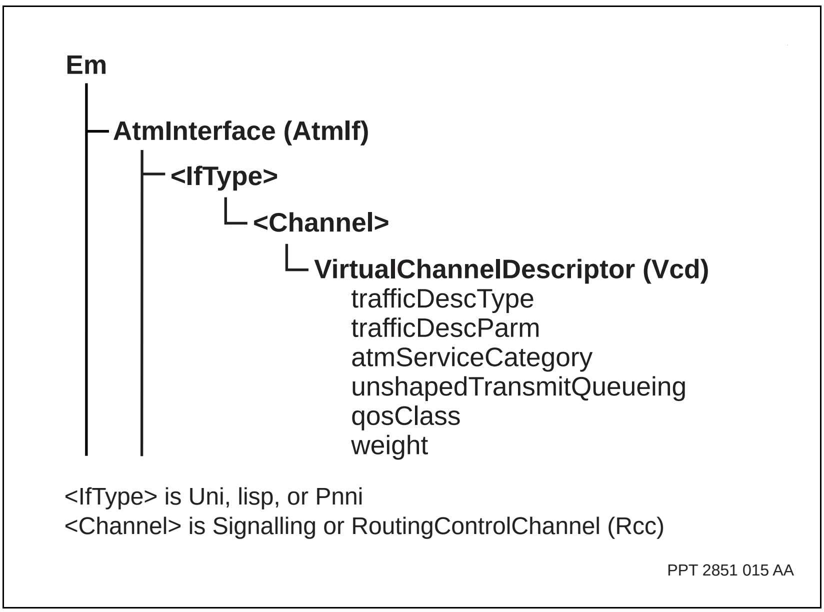


Table 4
Traffic descriptor types and parameters

Traffic descriptor type	**Traffic descriptor parameters**				
	#1 (cells per second)	**#2 (cells per second)**	**#3 (cells)**	**#4 (micro-seconds)**	**#5**
1 or 2 (no traffic management)	n/a	n/a	n/a	n/a	n/a
3	PCR CLP=0+1	n/a	n/a	CDVT (optional parameter)	requested shaping rate
4	PCR CLP=0+1	PCR CLP=0	n/a	CDVT (optional parameter)	requested shaping rate
5	PCR CLP=0+1	PCR CLP=0 with tagging	n/a	CDVT (optional parameter)	requested shaping rate
6	PCR CLP=0+1	SCR CLP=0+1	MBS CLP=0+1	CDVT (optional parameter)	requested shaping rate
7	PCR CLP=0+1	SCR CLP=0	MBS CLP=0	CDVT (optional parameter)	requested shaping rate
8	PCR CLP=0+1	SCR CLP=0 with tagging	MBS CLP=0	CDVT (optional parameter)	requested shaping rate
9	PCR CLP=0+1	CDVT CLP=0+1	MDCR CLP=0+1	n/a	n/a

Changing the default ILMI channel options

Change the default integrated local management interface (ILMI) channel options to use variable definitions other than the default values.

Procedure steps

1 Set up the ILMI channel VCI number.

set AtmIf/<n> Uni Ilmi vci <vci>

2 Disable the ILMI protocol.

set AtmIf/<n> Uni Ilmi operatingMode <opMode>

3 Register address prefixes for the ILMI.

set AtmIf/<n> Uni Ilmi prefixToRegister <prefix>

4 Add an instance of the Esi component.

add AtmIf/<n> Uni Ilmi Esi/<esi>

5 Set the scope for the ESI.

set AtmIf/<n> Uni Ilmi Esi/<esi> uniScope <scope

Variable definitions

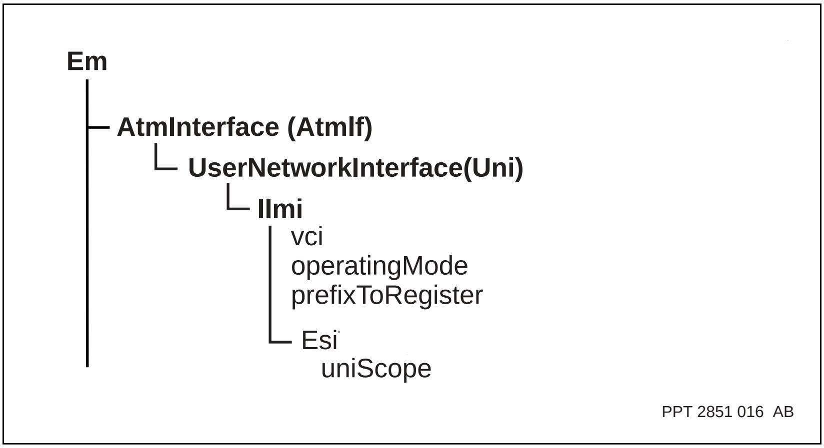
Variable	Value
<esi>	is a string of exactly 12 hexadecimal digits. The interface can register up to 20 address ESIs for the ILMI, but all ESI entries must be unique. This component is used to compose the user party of ATM addresses that is registered with the network side. If the side attribute in the *Uni* component is set to network then do not configure the *Esi* component. If the *side* attribute in the *Uni* component is set to user, then the instance of the *Esi* component is an end system indicator (ESI).
<n>	is the number of the ATM interface.
<opMode>	is addressRegEnabled, addressRegDisabled, or ilmiDisabled. The default is addressRegEnabled which means that the ILMI address registration protocol is enabled. The value addressRegDisabled indicates that the ILMI address registration protocol is disabled. The value ilmiDisabled indicates that all ILMI protocol functionality is disabled.
(Sheet 1 of 3)	

Variable	Value
<prefix>	is an address of exactly 26 hexadecimal digits. The interface can register up to 10 address prefixes for the ILMI under a single UNI, but all prefix entries must be unique. If the *side* attribute in the *Uni* component is set to user, then this attribute must be empty (no registered prefixes). You can provision multiple prefixes under this attribute. The node sends these prefixes to the user side as part of the address registration procedure. If you do not enter one or more prefixes, then the system uses the *nodePrefix* attribute under the *ModuleData* component for ILMI address registration. To allow group addresses, the ATM group authority and format identifier (AFI) must be configured on the UNI network side under the *prefixToRegister* attribute. If the interface is operating under UNI version 3.x, group addresses are not allowed. If you delete a prefix at any time, the system restarts the ILMI channel.
(Sheet 2 of 3)	

Variable	Value
<uniScope>	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 *uniScope* attribute defines the UNI address membership scope for all addresses created through this ESI. UNI scope specifies the inclusive routing hierarchy in which the address is known. Scope is used during ILMI dynamic address registration procedures. The value useDefaultScope maps to localNetwork for ATM group addresses and to global for individual addresses. For interfaces under UNI 3.x, the uniScope attribute is ignored.
<vci>	is the VCI for the ILMI channel under VPI=0. The default value is 16. This value can be any VCI in the connection map space. The ILMI VCI for the interface on the other end of the link must match the value provisioned here.
(Sheet 3 of 3)	

Procedure job aid

Figure 96
Default ILMI channel options component hierarchy



Configuring routing control channel options

Configure routing control channel options to change the variables to setting other than the default routing control channel (RCC) options.

Procedure steps

1 Set up the RCC VCI number.

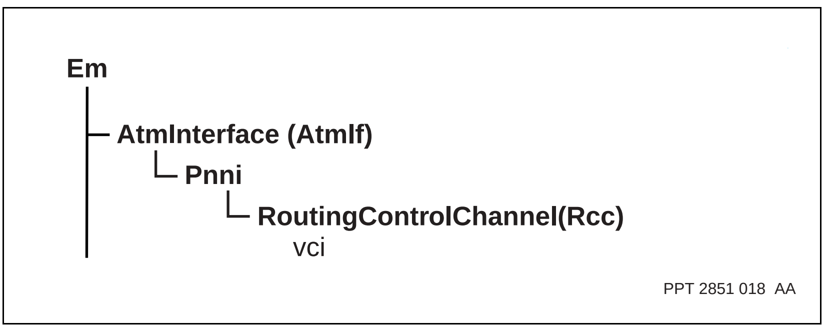
```
set AtmIf/<n> Pnni Rcc vci <vciNumber>
```

Variable definitions

Variable	Value
<n>	is the number of the ATM interface.
<vciNumber>	is the VCI that will be used on VPI=0 for the signaling channel. The default is 18. This value can be any VCI in the connection map address space for VPI 0. The RCC VCI for the interface on the other end of the link must match the value provisioned here.

Procedure job aid

Figure 97
Routing control channel options component hierarchy



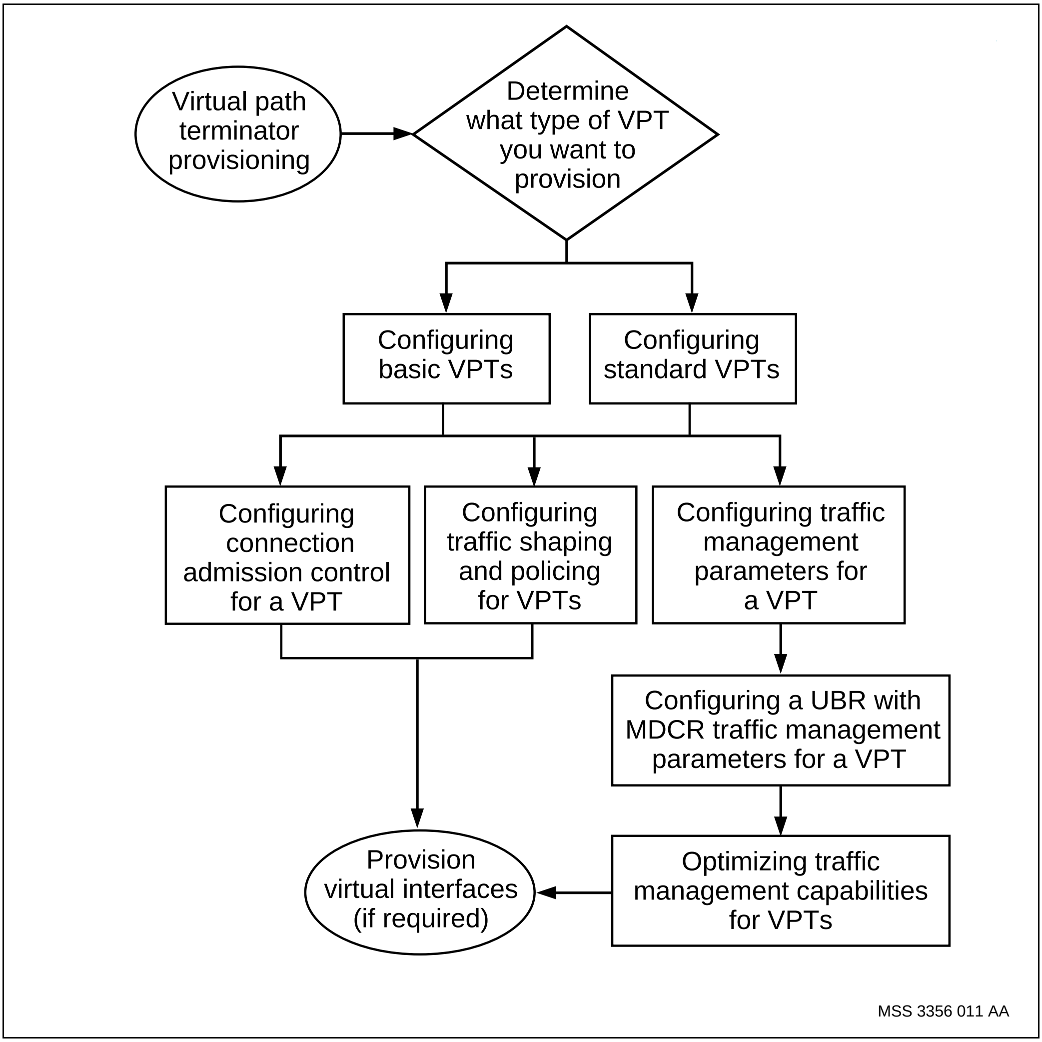
Chapter 9
Virtual path terminator provisioning

Provision virtual path terminators (VPT) to apply traffic management at the point where a VPC is unbundled into its VCC elements.

Virtual path terminator provisioning procedures

This task flow shows you the sequence of procedures you perform to provision virtual path terminators (VPTs). To link to any procedure, go to “Virtual path terminator provisioning procedure navigation” (page 322).

Figure 98
Virtual path terminator provisioning procedures



Virtual path terminator provisioning procedure navigation

- “Configuring basic VPTs” (page 324)
- “Configuring standard VPTs” (page 326)
- “Configuring connection admission control for a VPT” (page 328)

Configuring basic VPTs

Configure basic virtual path terminators (VPT) if you want the VPT to provide traffic management at the VC level only.

Prerequisites

- You must define one of the following:
  - UNI (see “Configuring a user-to-network connection” (page 208)
  - IISP (see “Configuring an inter-switch protocol interface” (page 213)
  - AINI (see “Configuring an ATM inter-network interface” (page 216)
  - PNNI (see “Configuring a PNNI” (page 219) ATM interface
- See the procedure “Configuring an AtmIf” (page 72). When following this procedure, ensure that the *faultHoldOffTime* attribute is set appropriately.
- 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 provision it for the same connection space as that used by VCCs. See “Configuring the connection map” (page 400) for more information.

Procedure steps

1 Add a virtual path terminator component.

add AtmIf/<n> Vpt/<Vpi>

2 Set the default loopbacks for the connection.

set AtmIf/<n> Vpt/<Vpi> Vpd segLinkSideLoopback <segLkLbk>

set AtmIf/<n> Vpt/<Vpi> Vpd endToEndLoopback <eeLbk>

3 If required, set the holding priority for the VPT.

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

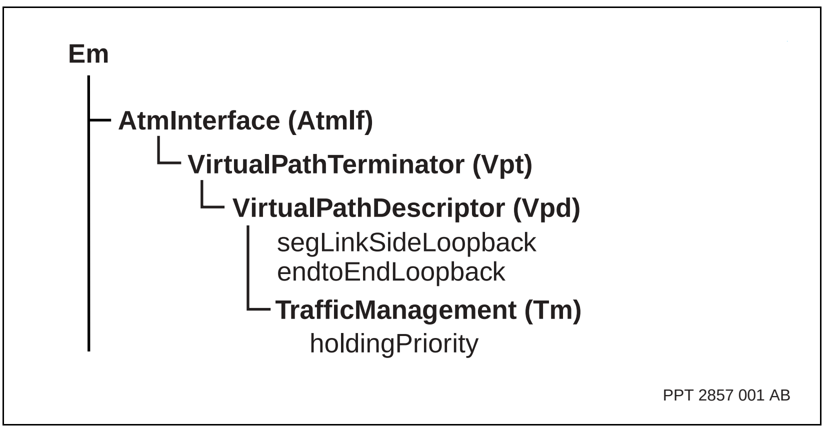
4 Add the virtual channel connections that are associated with the VPT. See the procedure “Configuring virtual channel connections” (page 114) for more information about adding VCCs.

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, 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.
<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*.
<Vpi>	is the instance value of the *Vpt* component. The value can be 0 - 4 095 for ATM IP cards, 0 - 4 094 for 8-port ATM cards, or 0 - 255 for CQC-based cards.

Procedure job aid

Figure 99
Basic VPTs component hierarchy



Configuring standard VPTs

Configure standard virtual path terminators (VPT) if you want to allow simultaneous traffic management at both the VP and VC level. Standard VPTs can also be configured to allow VPs and VCs on the same interface to dynamically share bandwidth.

> ***Note:*** Only AQM-based FPs support standard VPTs.

Prerequisites

- You must define one of the following ATM interfaces:
  - UNI (see “Configuring a user-to-network connection” (page 208)
  - IISP (see “Configuring an inter-switch protocol interface” (page 213)
  - AINI (see “Configuring an ATM inter-network interface” (page 216)
  - PNNI (see “Configuring a PNNI” (page 219)

Procedure steps

1 Add a virtual path terminator (VPT) component.

```
add AtmIf/<n> Vpt/<Vpi>
```

2 Set the VPT component type to standard.

```
set AtmIf/<n> Vpt/<Vpi> Vpd vptType standard
```

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 Configure a connection admission control (CAC) for the VPT. See the procedure “Configuring connection admission control for a VPT” (page 328).

5 Configure the desired traffic shaping and management capabilities for the VPT using the procedures “Configuring traffic shaping and policing for VPTs” (page 331) and “Configuring traffic management parameters for a VPT” (page 334).

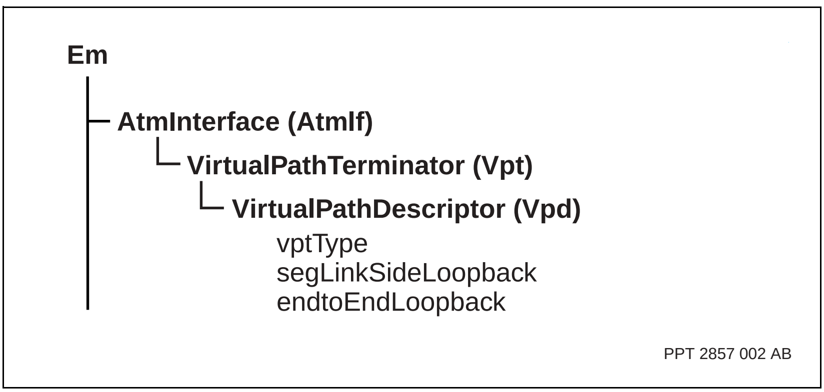
6 Add the virtual channel connections that are to reside on the VPT. See the procedure “Configuring virtual channel connections” (page 114) for more information about adding VCCs.

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*.
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<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*.
<Vpi>	is the instance value of the *Vpt* component. The value can be 0 - 4 095 for ATM IP cards, 0 - 4 094 for 8-port ATM cards.

Procedure job aid

Figure 100
Standard VPTs component hierarchy



Configuring connection admission control for a VPT

Configure connection admission control (CAC) for a virtual path terminator (VPT) if you want CAC techniques applied to the virtual channel connections (VCC) under a VPT.

Procedure steps

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>
```

8 Configure the desired traffic management parameters for the VPT using the steps in the procedure “Configuring traffic management parameters for a VPT” (page 334).

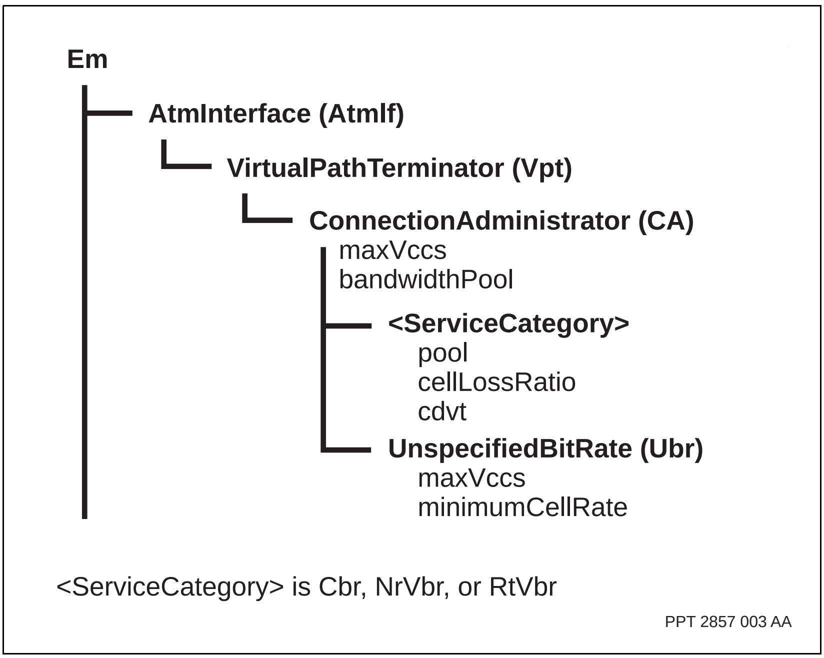
Variable definitions

Variable	Value
<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*.
<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 Networks Multiservice Switch systems, CLR is provisioned as a power of 10. For example, to provision 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.
<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.
(Sheet 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.
(Sheet 2 of 2)	

Procedure job aid

Figure 101
Connection admission control for a VPT component hierarchy



Configuring traffic shaping and policing for VPTs

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

Note: This procedure applies only to standard VPTs.

Prerequisites

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

Procedure steps

1 Enable traffic shaping and policing for the service category associated with the standard VPT. See “Configuring traffic management for service categories” (page 95) for information on enabling traffic shaping and policing.

2 Add a standard VPT. See the procedure “Configuring standard VPTs” (page 326).

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

```
set AtmIf/<n> Vpt/<Vpi> Vpd Tm trafficShaping
<trShaping>
```

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

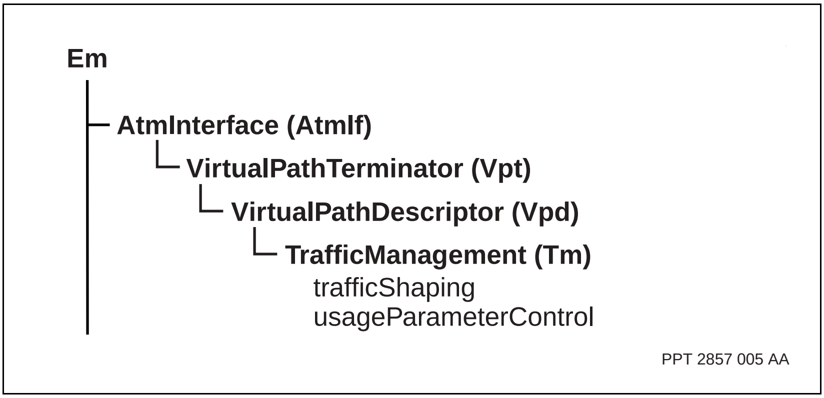
```
set AtmIf/<n> Vpt/<Vpi> Vpd Tm usageParameterControl
<upcValue>
```

Variable definitions

Variable	Value
<n>	is the instance value of the *AtmInterface* component and can be any unique value from 1 to 4 095.
<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 93) 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.
<upcValue>	is enforced, 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.
<Vpi>	is the instance value for the VPT.

Procedure job aid

Figure 102
Traffic shaping and policing for VPTs component hierarchy



Configuring traffic management parameters for a VPT

Configure traffic management parameters for a VPT to optimize traffic management.

Note: For GQM-based FPs this procedure only applies to basic VPTs.

Prerequisites

- For information on traffic management concepts, see NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*.

Procedure steps

1 Set the ATM service category for the VPT.

set AtmIf/<n> Vpt/<Vpt> Vpd Tm atmServiceCategory <service>

2 Set the traffic descriptor type for the transmit direction of the VPT.

set AtmIf/<n> Vpt/<Vpt> Vpd Tm txTrafficDescType <txTdt>

3 Set the traffic descriptor parameters for the transmit direction of the VPT.

set AtmIf/<n> Vpt/<Vpt> Vpd Tm txTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>] [4 <parm4>] [5 <parm5>]

4 Set the traffic descriptor type for the receive direction of the VPT.

set AtmIf/<n> Vpt/<Vpt> Vpd Tm rxTrafficDescType <rxTdt>

The default value is usually provisioned, which uses the traffic descriptor type and traffic descriptor parameters that have been provisioned for the transmit direction (that is, symmetrical traffic).

5 If you are configuring UPC for standard VPTs, set the traffic descriptor parameters for the receive direction of the VPT.

set AtmIf/<n> Vpt/<Vpt> Vpd Tm rxTrafficDescParm [1 <parm1>] [2 <parm2>] [3 <parm3>] [4 <parm4>]

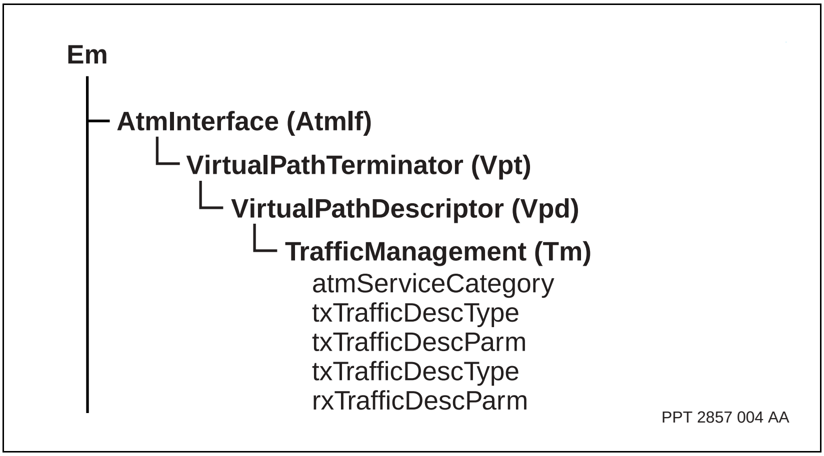
If the value of the *rxTrafficDescType* attribute is *sameAsTx*, the parameters for this attribute are the same as the parameters in the *txTrafficDescParm* attribute.

Variable definitions

Variable	Value
<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.
<txTdt>	is a value between 1 and 9, defining the traffic descriptor type. The default is 1. Traffic descriptor 9 is to be used exclusively by UBR with MDCR.
<Vpt>	is the instance value of the *Vpt* component and can be any unique value from 0 to 4 095.

Procedure job aid

Figure 103
Traffic management parameters for a VPT component hierarchy



Configuring a UBR with MDCR traffic management parameters for a VPT

Configure a UBR with MDCR traffic management for a VPT to use an explicit value for MDCR by using the traffic descriptor type 9 and setting the traffic descriptor parameters.

Alternatively, configure a UBR with MDCR VPT to use the default minimum cell rate for the UBR service category at an interface by using traffic descriptor type 1, 2, or 3 and setting the traffic descriptor parameters.

For more information about traffic descriptor parameters, see “Configuring traffic management parameters for a VPT” (page 334).

Optimizing traffic management capabilities for VPTs

Optimize traffic management capabilities for virtual path terminators (VPT) to refine VPT traffic management settings.

Prerequisites

- For information on traffic management concepts, see NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*.

Procedure steps

1 Set per-VC queuing for the VPT.

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

2 Set the holding priority for the VPT.

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

3 Configure the fairness weighting for an unshaped VPT.

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

4 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.

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.

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.
<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.
<Vpi>	is the instance of the *Vpt*.
<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.

Chapter 10
Virtual interface provisioning

Provision virtual interfaces to allow Nortel Networks 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) or through a virtual path multiplexer.

After completing the procedures in this section, you can complete the following for each virtual interface:

- refinements to operation of the signaling channel, see “Configuring routing control channel options” (page 319)
- addressing requirements for PNNI networking, see “Configuring PNNI addressing parameters” (page 255)
- refinements to the integrated local management interface (ILMI) channel, see “Changing the default ILMI channel options” (page 315)
- refinements to the VCD for the UNI signaling channel, see “Configuring traffic management for ATM control channels” (page 310)

Prerequisites to virtual interfaces provisioning

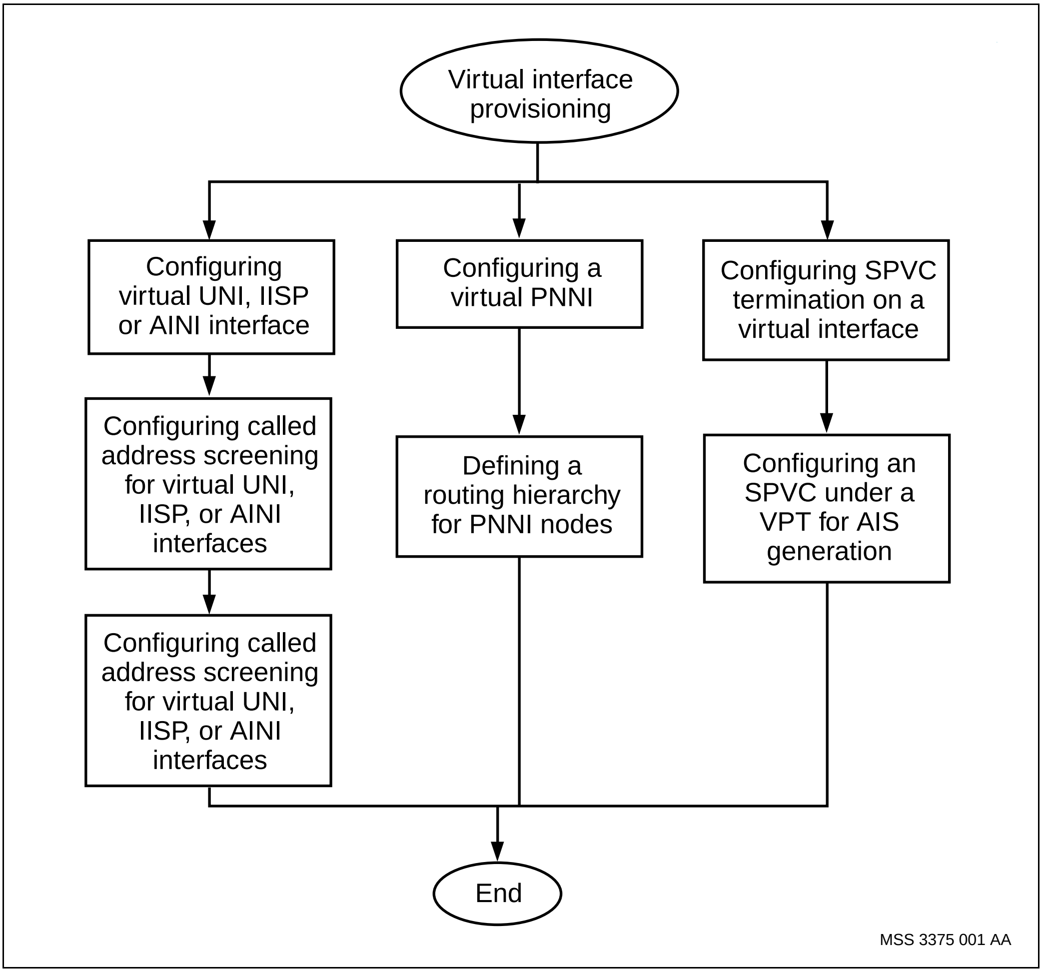
- Virtual paths connections and virtual terminators must be provisioned to support virtual interfaces. A virtual path connection or virtual path terminator (VPT) with an identifier of 0 disables the non-associated signaling channel. A semantic check ensures that a *Vpc/0* or *Vpt/0* component cannot 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 provisioned 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 interface provisioning procedures

This task flow shows you the sequence of procedures you perform to provision virtual interfaces. To link to any procedure, go to “Virtual interface provisioning procedure navigation” (page 343).

Figure 104
Virtual interface provisioning procedures



Virtual interface provisioning procedure navigation

- "Configuring a virtual UNI, IISP, or AINI interface" (page 345)
- "Configuring called address screening for a virtual UNI, IISP or AINI interface" (page 348)
- "Configuring calling address screening for a virtual UNI, IISP, or AINI interface" (page 350)

Configuring a virtual UNI, IISP, or AINI interface

Configure a virtual UNI, IISP, or AINI interface to support the UNI, IISP, or AINI protocol.

Procedure steps

1 Add a virtual interface.

```
add AtmIf/<n> Vpt/<Vpi> <IfType>
```

2 Configure the virtual path connection identifier (VPCI) value for the interface.

```
set AtmIf/<n> Vpt/<Vpi> <IfType> vpci <vpci_value>
```

> **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.

3 Set the interface version.

```
set AtmIf/<n> Vpt/<Vpi> <IfType> version <ver>
```

4 Configure the interface as either the user side or the network side.

```
set AtmIf/<n> Vpt/<Vpi> <IfType> 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 Optionally, modify the default parameters for the signaling channel. See “Configuring routing control channel options” (page 319) for information on how to change these parameters.

7 Optionally, add a signaling virtual channel descriptor (VCD). See “Configuring traffic management for ATM control channels” (page 310) for information on how to add the signaling VCD.

8 Optionally, change the operating parameters for the signaling VCD. See “Configuring traffic management for ATM control channels” (page 310) for information on how to change these parameters.

9 Add the required static addresses for the interface.

```
set AtmIf/<n> Vpt/<Vpi> <IfType> Address/
<address>,<addr_type>
```

Because there is no default address, you must configure a static address if the virtual interface terminates an SPVC.

10 Configure address screening for called or calling addresses. See “Configuring called address screening for a virtual UNI, IISP or AINI interface” (page 348) or “Configuring calling address screening for a virtual UNI, IISP, or AINI interface” (page 350) for information.

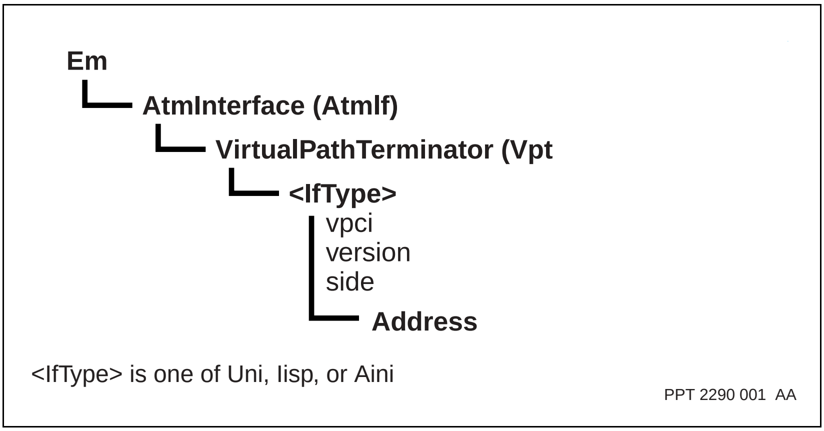
Variable definitions

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.
(Sheet 1 of 2)	

Variable	Value
<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.
(Sheet 2 of 2)	

Procedure job aid

Figure 105
Virtual UNI, IISP, AINI interface 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.

Procedure steps

1 Provision 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

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

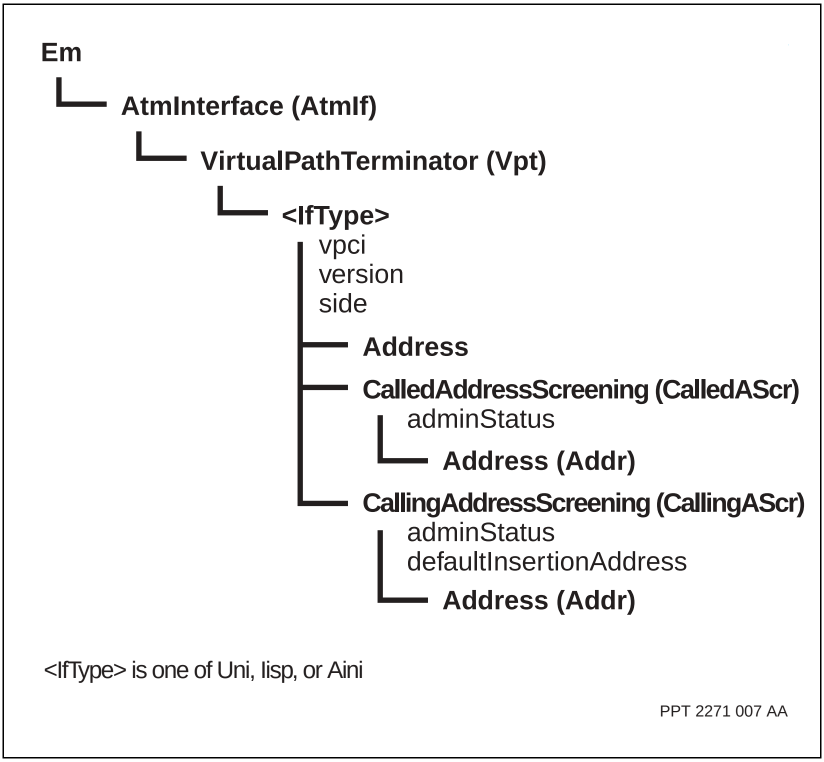
When you enable called address screening, incoming calls are screened against the addresses that you have provisioned. The incoming call is accepted if the called address matches an address that you have provisioned to be accepted. The incoming call is rejected if the called address is either not provisioned, matches an address you have provisioned to be rejected, or does not match any provisioned address.

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

Figure 106
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.

Procedure steps

1 Provision 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
```

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 provisioned. The incoming call is accepted if the calling address matches an address that you have provisioned to be accepted. The incoming call is rejected if the calling address matches an address you have provisioned to be rejected, does not match a provisioned address, or the connection request does not contain a calling address and there is no provisioned default insertion address.

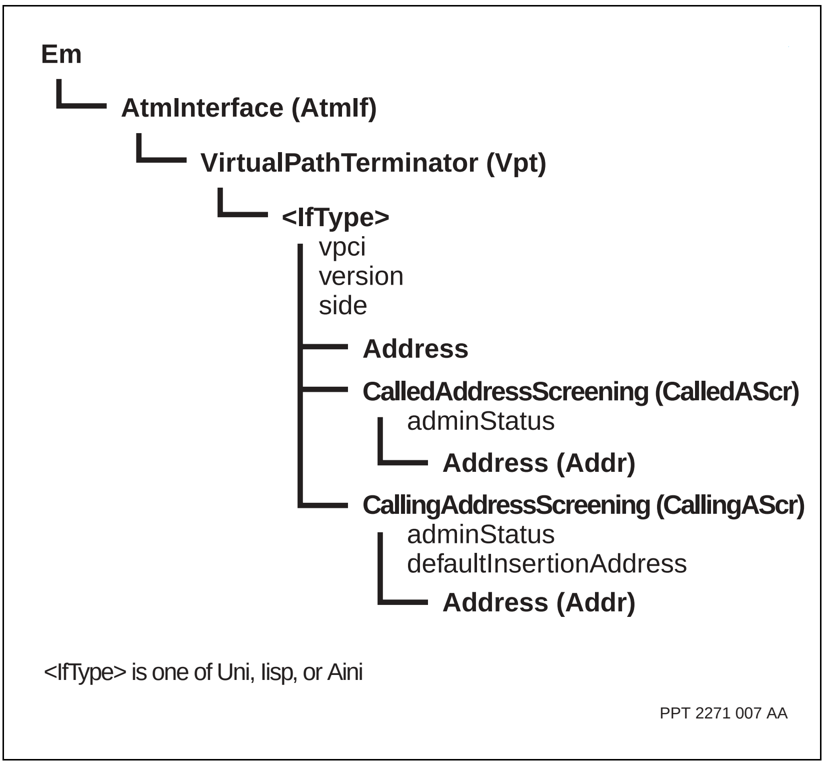
Variable definitions

Variable	Value
<action>	is accept or reject.
<address>	is a source ATM address.
(Sheet 1 of 2)	

Variable	Value
<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.
(Sheet 2 of 2)	

Procedure job aid

Figure 107
Calling address screening for virtual UNI, IISP, or AINI component hierarchy



Configuring a virtual PNNI

Configure a virtual PNNI interface to set up a virtual interface using PNNI protocol.

Procedure steps

1 Add the component for the virtual interface.

```
add AtmIf/<n> Vpt/<Vpi> Pnni
```

2 Configure the VPCI value for the interface.

```
set AtmIf/<n> Vpt/<Vpi> Pnni vpci <vpci>
```

> **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.

3 Optionally, modify the default parameters for the signaling channel. See “Configuring routing control channel options” (page 319) for information on how to change these parameters.

4 Optionally, add a signaling virtual channel descriptor (VCD). See “Configuring traffic management for ATM control channels” (page 310) for information on how to add the signaling VCD.

5 Optionally, change the operating parameters for the signaling VCD. See “Configuring traffic management for ATM control channels” (page 310) for information on how to change these parameters.

6 Optionally, modify the default parameters for the routing control channel (RCC). See “Configuring traffic management for ATM control channels” (page 310) for information on how to change these parameters.

7 Optionally, add an RCC VCD. See “Configuring traffic management for ATM control channels” (page 310) for information on how to add the RCC VCD.

8 Optionally, change the operating parameters for the RCC VCD. See “Configuring traffic management for ATM control channels” (page 310) for information on how to change these parameters.

9 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.

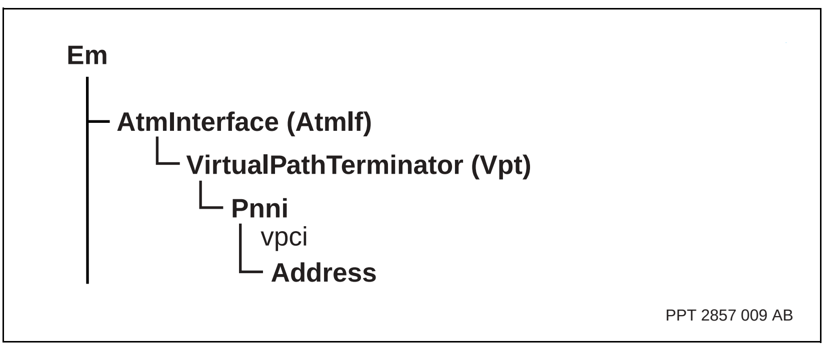
```
set AtmIf/<n> Vpt/<Vpi> Pnni Address/
<address>,<addr_type>
```

Variable definitions

Variable	Value
<addr_type>	is 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.
<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

Figure 108
Virtual PNNI component hierarchy



Defining a routing hierarchy for PNNI nodes

Define a routing hierarchy for PNNI nodes that have been configured as PNNI virtual interfaces.

Use this procedure if the node participates in a PNNI network hierarchy and the scope of an address is different from the scope of the default address for that node.

Procedure steps

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.

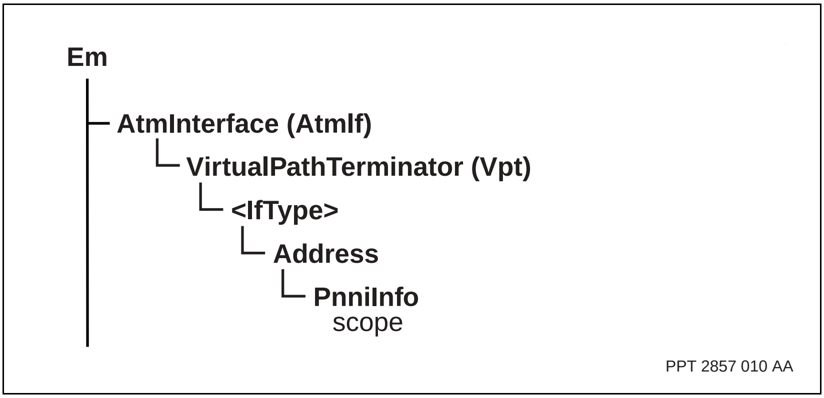
```
set AtmIf/<n> Vpt/<m> Pnni Address
<address>,<addr_type> PnniInfo scope <level>
```

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

Figure 109
Routing hierarchy for PNNI nodes 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.

Prerequisites

- You can configure an SPVC under a VPT only if the VPT has a virtual interface subcomponent (one of *Uni*, *Iisp*, Aini, or *Pnni*).
- Add a virtual interface. See “Configuring a virtual UNI, IISP, or AINI interface” (page 345) and “Configuring a virtual PNNI” (page 352) for procedures on how to add and configure virtual interfaces.

Procedure steps

1 Associate a terminate component with the address.

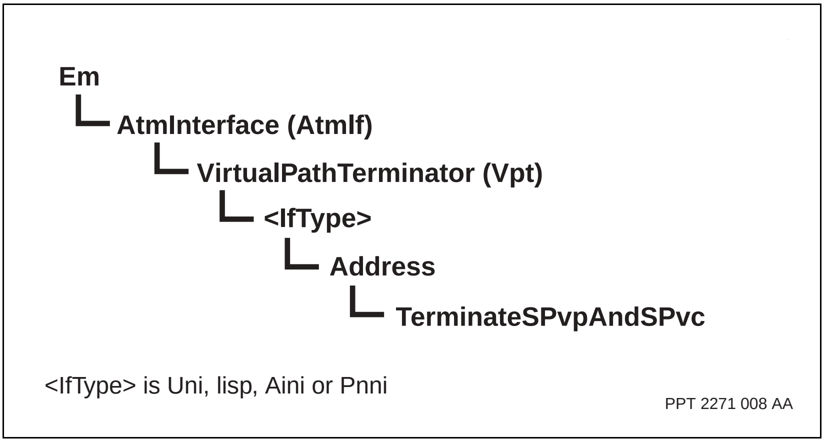
```
add AtmIf/<n> Vpt/<m> <IfType> Address
<address>,<addr_type> TerminateSPvpAndSPvc
```

Variable definitions

Variable	Value
<addr_type>	is the address type (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 addresses for each interface, but all address entries must be unique on this interface.
<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

Figure 110
SPVC termination on a virtual interface component hierarchy



Configuring an SPVC under a VPT for AIS generation

Configure an SPVC under a VPT for AIS generation to enable AIS.

Prerequisites

- Before AIS generation can be enabled at the destination, a destination SPVC under a Vpt must first be configured.
- 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

1 Add a destination SPVC under a Vpt.

add atmif/<n> vpt/<m> vcc/<vci> dst

2 Configure the traffic requirements. See the procedure “Configuring traffic management parameters for a VPT” (page 334) for details. For the *trafficShaping* and *usageParameterControl* parameters, these can be independently enabled or disabled on the source and/or destination ends of an SPVC or SPVP.

3 Enable AIS at the source SPVC under a Vpt.

set atmif/<n> vpt/<m> vcc/<vci> src aisGeneration enable

4 Enable AIS at the destination SPVC under a Vpt.

set atmif/<n> vpt/<m> vcc/<vci> dst config aisGeneration enable

5 Disable AIS at the source SPVC under a Vpt

set atmif/<n> vpt/<m> vcc/<vci> src aisGeneration disable

6 Disable AIS at the destination SPVC under a Vpt.

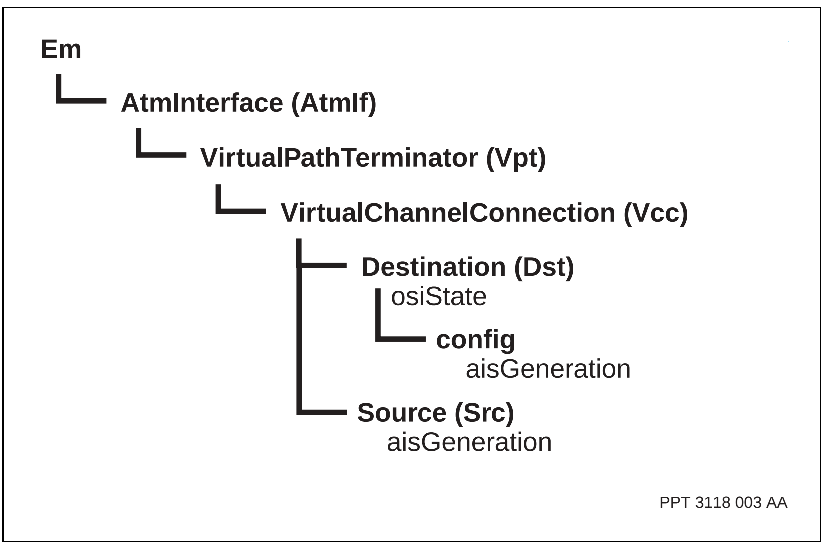
```
set atmif/<n> vpt/<m> vcc/<vci> dst config
aisGeneration disable
```

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.
<Vpi.Vci>	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

Figure 111
SPVC under a Vpt for AIS generation component hierarchy



Chapter 11
ATM accounting configuration

Configure Nortel Networks Multiservice Switch nodes to collect accounting statistics for ATM connections.

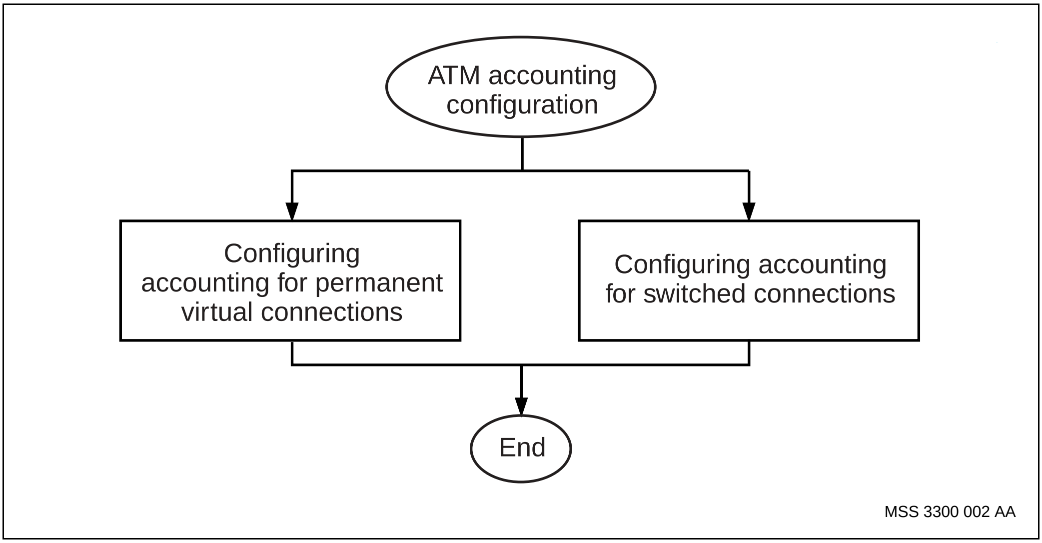
Prerequisites to ATM accounting configuration

- Configure basic accounting by following the procedures in NN10600-560 *Nortel Networks Multiservice Switch 7400/15000/20000 Accounting*.
- For information on ATM accounting, see “ATM accounting” (page 368).
- Configure the following ATM elements:
  - — an ATM interface. See “Configuring an AtmIf” (page 72).
  - — a UNI (if configuring the service on a UNI). See “Configuring a user-to-network connection” (page 208).
  - — an IISP interface (if configuring the service on a IISP interface). See “Configuring an inter-switch protocol interface” (page 213).
  - — an AINI interface (if configuring the service on an AINI interface). See “Configuring an ATM inter-network interface” (page 216).
  - — a PNNI (if configuring the service on a PNNI). See “Configuring a PNNI” (page 219).

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 362).

Figure 112
ATM accounting configuration procedures



ATM accounting configuration procedure navigation

- “Configuring accounting for permanent virtual connections” (page 363)
- “Configuring accounting for switched connections” (page 366)

Configuring accounting for permanent virtual connections

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

Procedure steps

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 connection (*Vpc*) or a virtual path termination (*Vpt*), use the following command:

```
set AtmIf/<n> <connection_type>/<x> Vpd correlationTag
<tag>
```

For a virtual channel connection (*Vcc*), use the following command:

```
set AtmIf/<n> Vcc/<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.

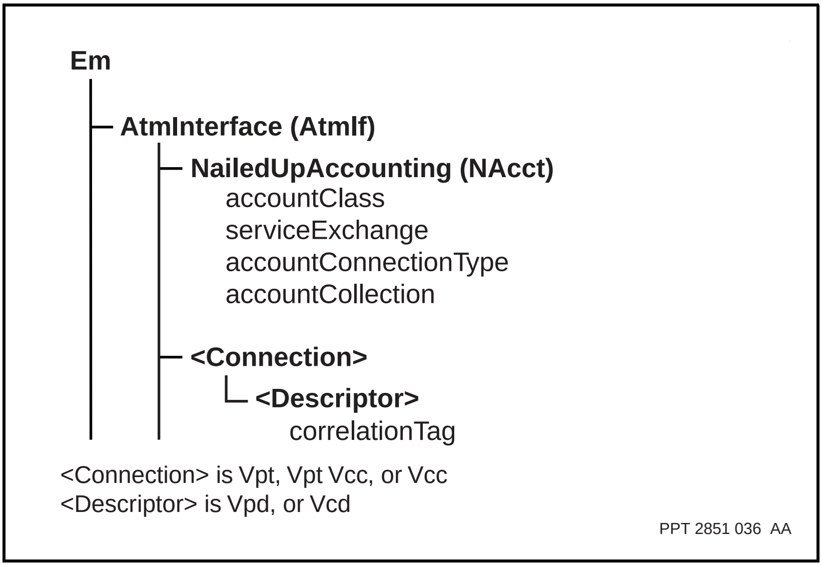
```
set AtmIf/<n> NAcct accountConnectionType <acType>
```

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.
<connection_type>	is Vpc or Vpt. This specifies if the connection is a virtual path connection (*Vpc*) or a virtual path termination (*Vpt*).
<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

Figure 113
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

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>

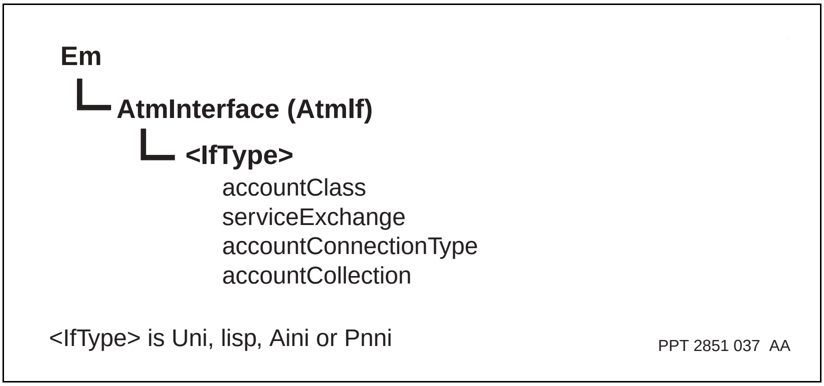
Variable definitions

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 Networks 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.
(Sheet 1 of 2)	

Variable	Definition
<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.
(Sheet 2 of 2)	

Procedure job aid

Figure 114
Accounting for switched connections component hierarchy



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.

  Note: 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 369)
- “Accurate record generation” (page 371)
- “Accounting for permanent virtual connections” (page 371)
- “Accounting for switched connections” (page 372)

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 369)
- “Changing the default settings” (page 369)

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:

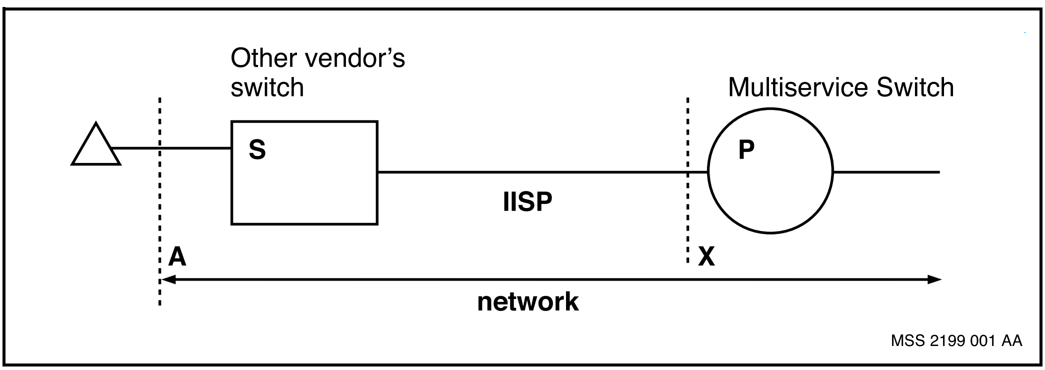
- when accounting is enabled at a basic interface on an inner node, it should be configured to generate intermediate accounting records

- 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 370). 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.

> ***Note 1:*** 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.
>
> ***Note 2:*** 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.

Figure 115
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 provisioning 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*).

> ***Note:*** 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 Networks 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 provisioned 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>).

Chapter 12
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:

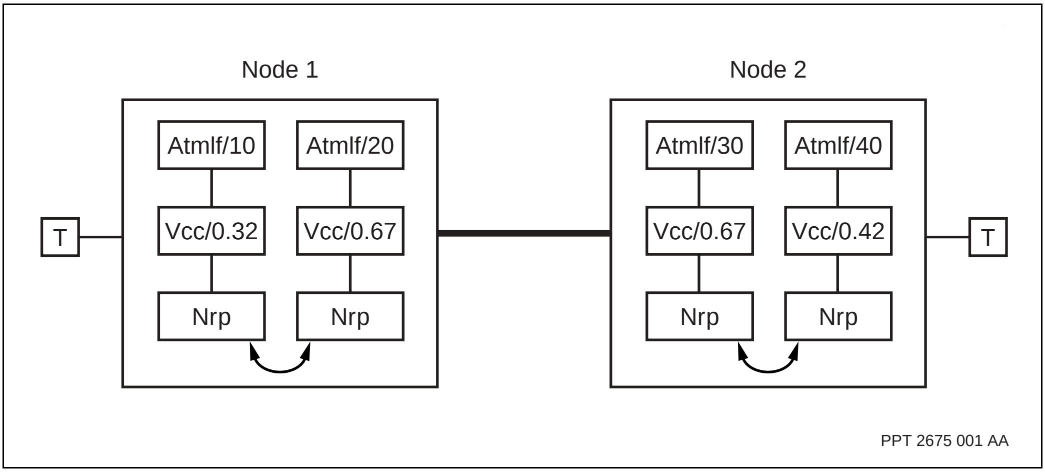
- a general statement of the type of connections or network topology
- an illustration of the connection of network topology
- a provisioning example

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

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 376).

Figure 116
Configuring logical ABS service with NRPS using VCCs



Configuration example for logical ABS with NRPs using VCCs

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.

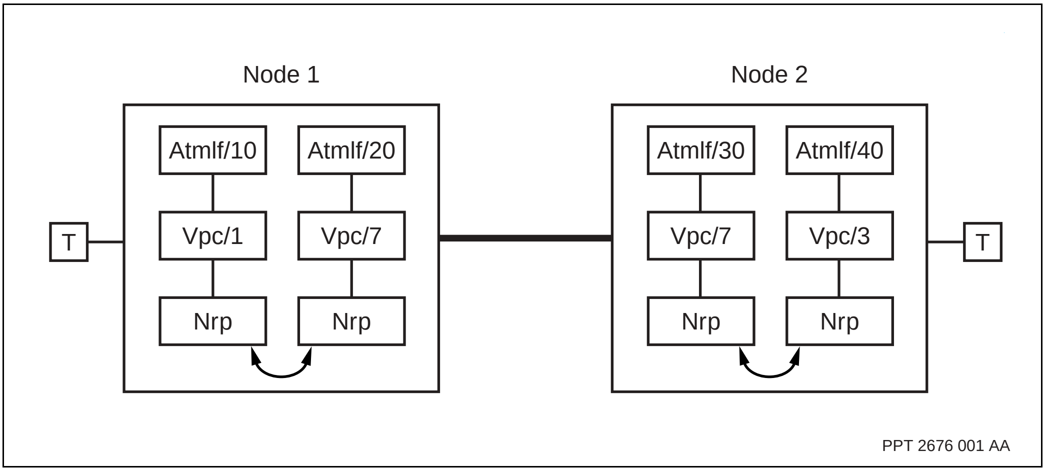
set AtmIf/10 Vcc/0.32 Nrp nexthop AtmIf/20 Vcc/0.67 Nrp

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.

Example 2: Logical ABS with NRPs using VPCs

This example shows how to provision 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 378).

Figure 117
Configuring logical ABS service with Nrps using VPCs



Configuration example for logical ABS with NRPs using VPCs

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.

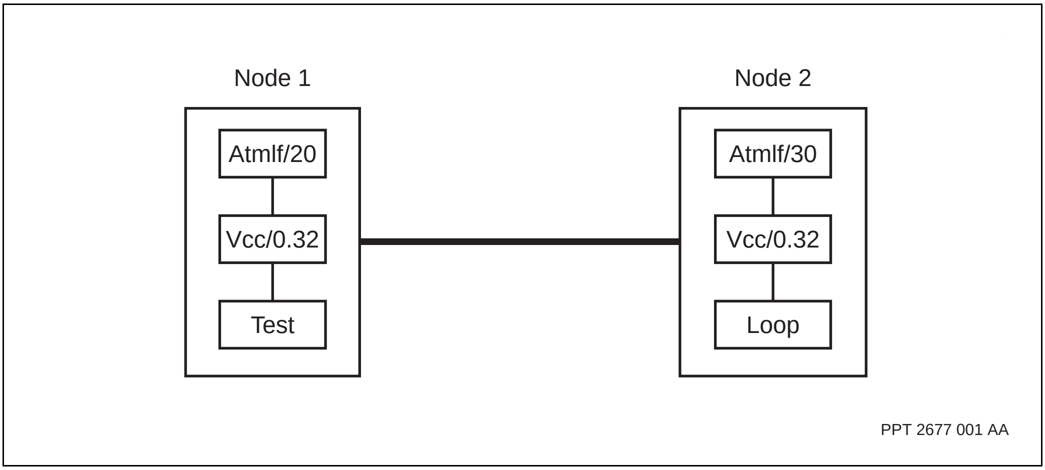
set AtmIf/10 Vpc/1 Nrp nexthop AtmIf/20 Vpc/7 Nrp

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.

Example 3: Test and loop configuration using VCCs

This example shows how to provision a 64 kb/s VCC for a test loopback. See the figure “Configuring test and loop using VCCs” (page 380).

Figure 118
Configuring test and loop using VCCs



Configuring test and loop using VCCs

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.

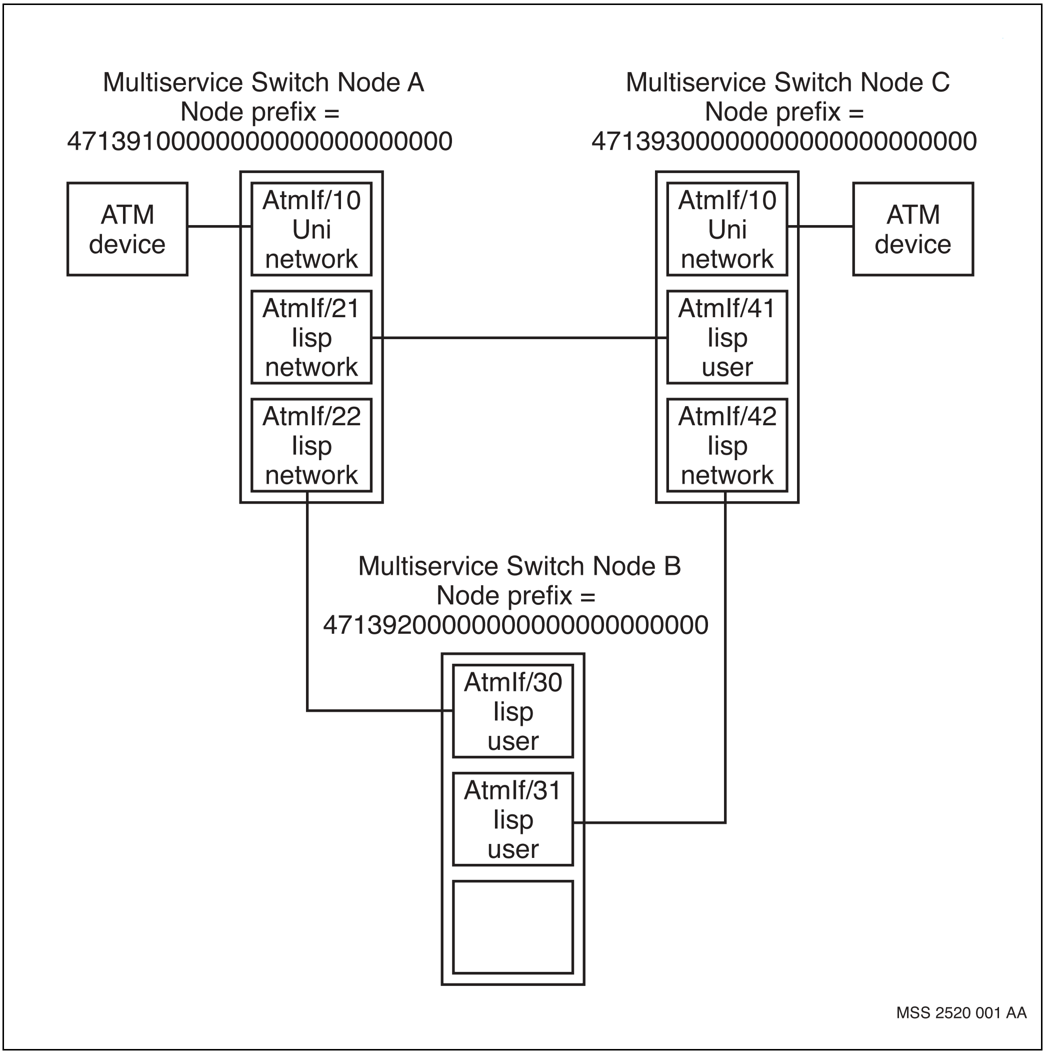
add AtmIf/30 Vcc/0.32 Loop

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

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

This example shows how to provision static routing in a small Nortel Networks Multiservice Switch-only network. The network configuration is shown in the figure “Provisioning static routing in a small Multiservice Switch-only network” (page 383).

Figure 119
Provisioning static routing in a small Multiservice Switch-only network



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 provisioned (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 provisioned.

Configuring static routing in a small Multiservice Switch-only network

1 Set up ATM routing for Node A, specifying the node prefix. The objective of the first series of provisioning steps is to set up the node prefix and the ATM interfaces on Node A. See the figure “Provisioning static routing in a small Multiservice Switch-only network” (page 383).

```
add AtmRouting
```

```
set ModuleData nodePrefix 47139100000000000000000000
```

2 Set up an ATM interface on FP #1. Do not provision the *side* attribute provisioned 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 provision 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
4713931000000000000000000000000000000000
```

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 provision the SPVP to Node C.

add AtmIf/10 Vpc/3

add AtmIf/10 Vpc/3 SourcePvp

set AtmIf/10 Vpc/3 SourcePvp calledAddress 4713931000000000000000000000000000000000

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 provision 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 provision 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 provision 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 provision under the *ModuleData* component for Node B.

7 Set up ATM routing for Node B. The objective of the second series of provisioning steps is to set up the node prefix and the ATM interfaces on Node B. See the figure “Provisioning static routing in a small Multiservice Switch-only network” (page 383).

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 provision 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 provision the *side* attribute provisioned 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 provision 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
```

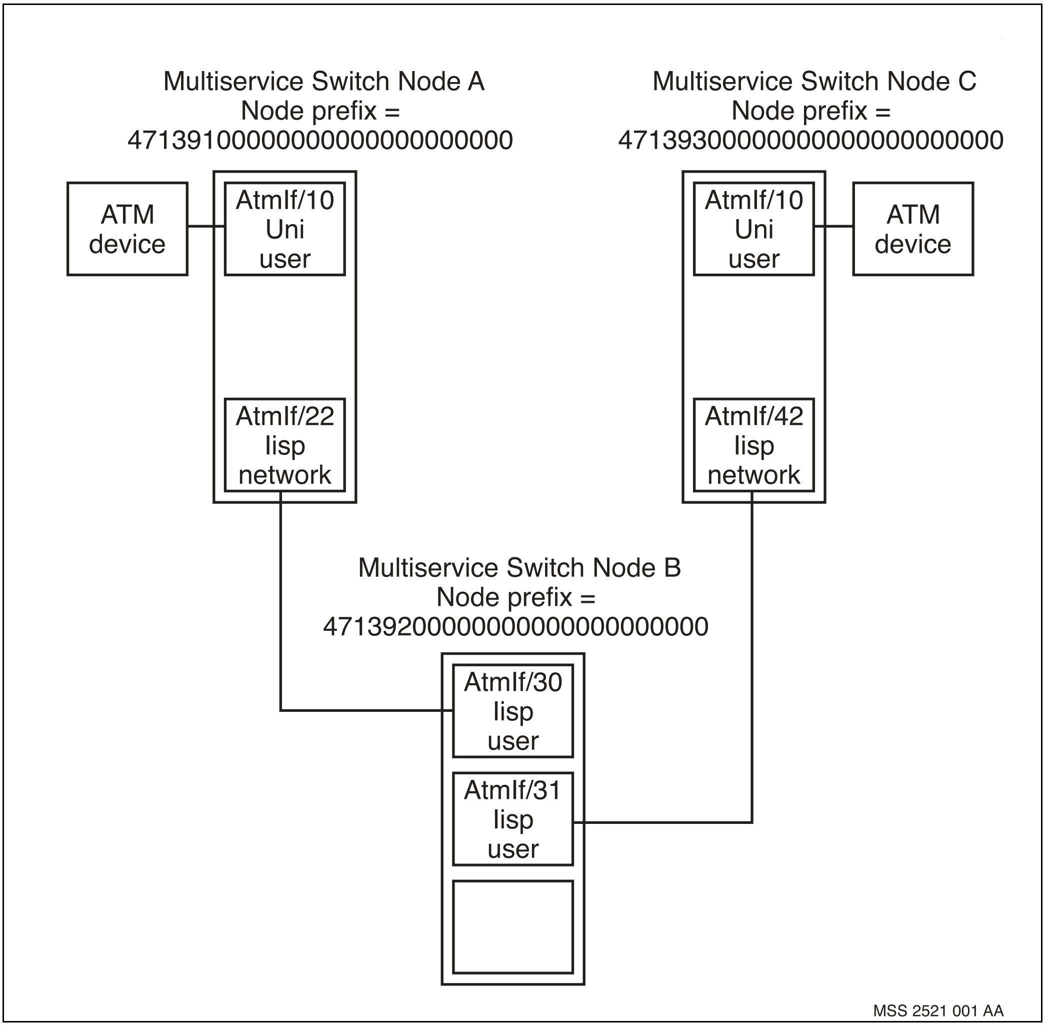
Do not provision the *side* attribute, since you want to use the default value of *networkSide* for this interface.

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 provision under the *ModuleData* component for Node B.

Example 5: Static routing using wild cards

This example shows how you can provision a wild card address specification to support static routing in a small Nortel Networks Multiservice Switch-only network. The network configuration is shown in the figure “Provisioning static routing using wild cards” (page 389).

Figure 120
Provisioning static routing using wild cards



This configuration is similar to that illustrated in “Example 3: Test and loop configuration using VCCs” (page 380). 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 provision several additional interfaces on Node B (not shown in the figure "Provisioning static routing using wild cards" (page 389)), all nodes accessible through those interfaces are accessible through *AtmIf/22* on Node A. Similarly, if you provision 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 provision 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.

Configuring static routing in a Multiservice Switch-only network using wild cards

1 Set up ATM routing for Node A, specifying the node prefix. The objective of the first series of provisioning steps is to set up the node prefix and the ATM interfaces on Node A. See the figure "Provisioning static routing using wild cards" (page 389).

add AtmRouting

set ModuleData nodePrefix 4713910000000000000000000

2 Set up an UNI on FP #1. Do not provision the *side* attribute provisioned 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 provision 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 4713931000000000000000000000000000000000

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 provision the SPVP to Node C.

`add AtmIf/10 Vpc/3`

`add AtmIf/10 Vpc/3 SourcePvp`

`set AtmIf/10 Vpc/3 SourcePvp calledAddress`
`4713931000000000000000000000000000000000`

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 provision 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 provisioning steps is to set up the node prefix and the ATM interfaces on Node B (see the figure "Provisioning static routing using wild cards" (page 389)).

`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 provision 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 provisioning steps is to set up the node prefix and the ATM interfaces on Node C. See the figure “Provisioning static routing using wild cards” (page 389).

add AtmRouting

set ModuleData nodePrefix 4713930000000000000000000

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

add AtmIf/42

add AtmIf/42 Iisp

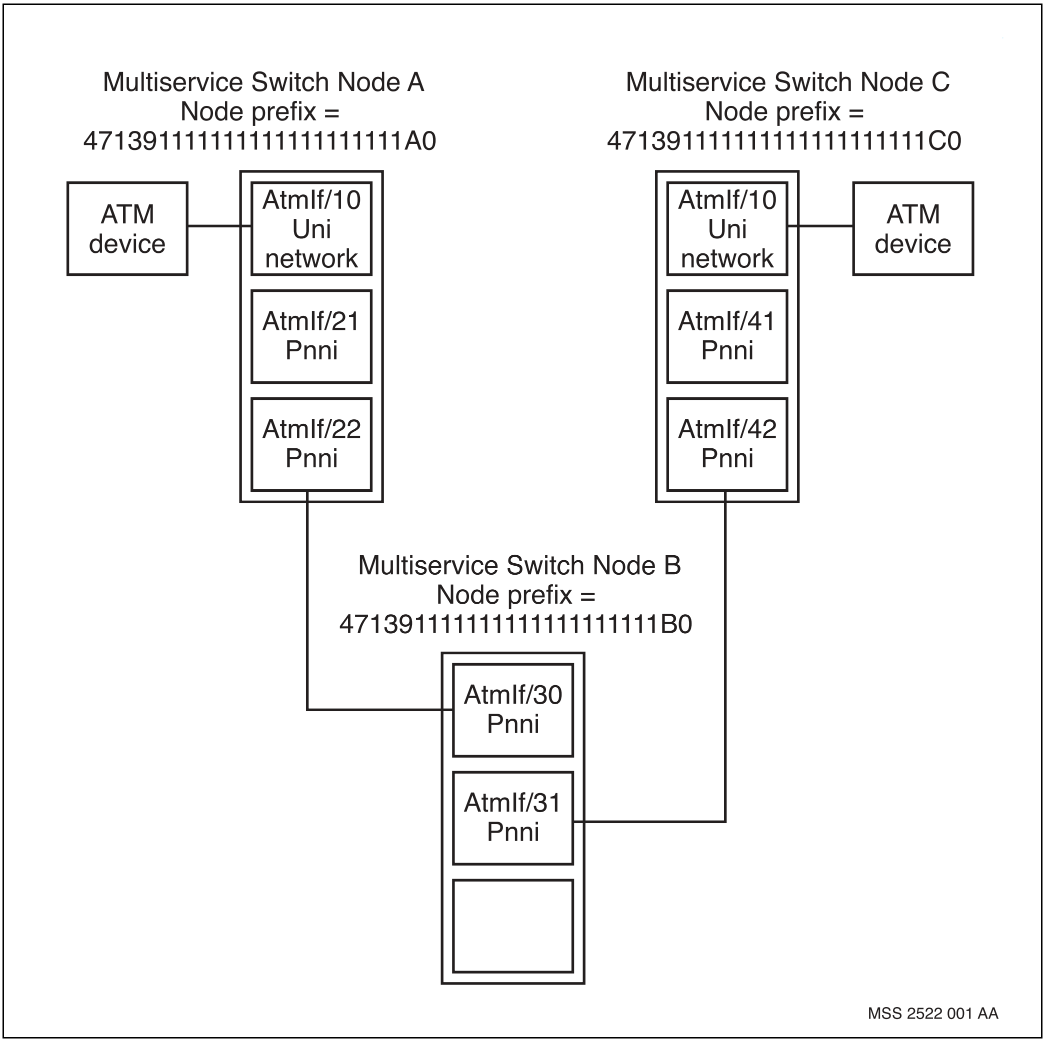
add AtmIf/42 Iisp address/471392,primary

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 provision under the *ModuleData* component for Node B.

Example 6: Small Multiservice Switch-only network under PNNI

This example shows how to provision a small Nortel Networks Multiservice Switch-only network using PNNI. The figure “Provisioning a small Multiservice Switch-only network under PNNI” (page 394) shows the network configuration.

Figure 121
Provisioning a small Multiservice Switch-only network under PNNI



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 provisioned
- 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 provisioned.

Configuring a small Multiservice Switch-only network for dynamic networking

1 Set up ATM routing and PNNI for Node A. The objective of the first series of provisioning steps is to set up the node prefix and the ATM interfaces on Node A. See the figure “Provisioning a small Multiservice Switch-only network under PNNI” (page 394).

add AtmRouting

add AtmRouting Pnni

2 Set the node prefix for Node A.

set ModuleData nodePrefix 4713911111111111111111A0

If not explicitly provisioned, 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 provision 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 provision 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 4713911111111111111111111C001000000000000

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 provisioning steps is to set up the node prefix and the ATM interfaces on Node B. See the figure “Provisioning a small Multiservice Switch-only network under PNNI” (page 394).

add AtmRouting

add AtmRouting Pnni

9 Set the node prefix for Node B.

set ModuleData nodePrefix 4713911111111111111111111B0

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 provisioning steps is to set up the node prefix and the ATM interfaces on Node C. See the figure “Provisioning a small Multiservice Switch-only network under PNNI” (page 394).

add AtmRouting

add AtmRouting Pnni

14 Set the node prefix for Node C.

set ModuleData nodePrefix 471391111111111111111111C0

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/
471391111111111111111111C0010000000000,primary
terminateSpvc

add AtmIf/42 Pnni

Appendix A
Connection map configuration 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.

> ***Note:*** The information in this section does not apply to eight-port DS1/E1 CQC-based or the APC-based FPs.

Prerequisites to connection map configuration for CQC-based FPs

- For a detailed description of how the connection map works, see NN10600-702 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals*.For detailed engineering considerations and constraints, see Nortel Networks 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 Networks 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>

- You must provision *VirtualPathTerminator* components in VCC space. Each Basic VPT uses two VCCs and each Standard VPT uses three VCCs.
- 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 402).
- For some sample values you can use to configure the connection map, see “Connection map templates for CQC-based FPs” (page 403).
- Add an ATM interface before configuring the connection map. See “Configuring an AtmIf” (page 72).

Configuring the connection map

Configure the connection map to define the range of address spaces for ATM connections.

Procedure steps

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

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.

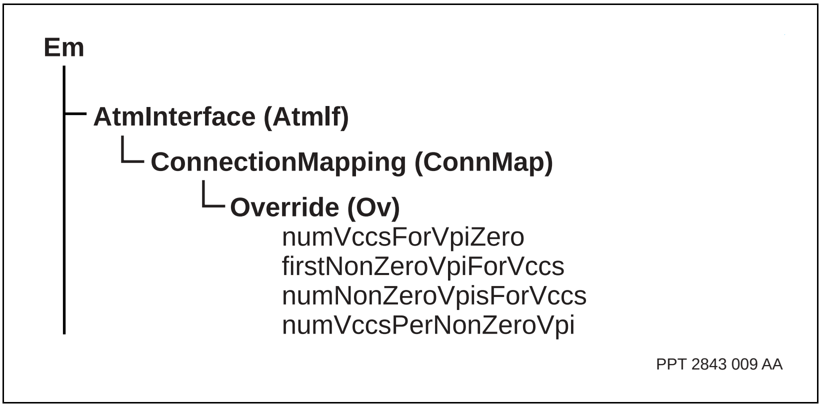
```
set AtmIf/<n> ConnMap Ov numVccsPerNonZeroVpi <nVccs>
```

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.
<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.
<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.
<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

Figure 122
Configuring the connection map component hierarchy



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 Networks Multiservice Switch Release Notes.

“Connection mapping sample configuration values for CQC-based FPs (excluding eight-port DS1/E1 FPs)” (page 404) provides sample configuration values for provisioning the available VCC space through the connection map. “Connection mapping configurations and values (excluding eight-port DS1/E1 FPs)” (page 404) 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 provision 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 provisioned.
- 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.

Table 5
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

Table 6
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
(Sheet 1 of 3)					

Table 6 (continued)
Connection mapping configurations and values (excluding eight-port DS1/E1 FPs)

Config #	Application	Usable VCC range	Maximum VCCs	Maximum VPCs	Notes
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
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

(Sheet 2 of 3)

Table 6 (continued)
Connection mapping configurations and values (excluding eight-port DS1/E1 FPs)

Config #	Application	Usable VCC range	Maximum VCCs	Maximum VPCs	Notes
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
(Sheet 3 of 3)					

Appendix B
Hitless ATM services on FPs

This section contains information on hitless services on FPs. The following information is provided:

Providing hitless ATM services on FPs with optical interfaces

This section lists the main steps that must be performed to provide hitless ATM services on Nortel Networks Multiservice Switch nodes using FPs with optical interfaces. See NN10600-700 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Technology Fundamentals* for a description of which ATM services can be hitless.

Procedure steps

1 Ensure that the main and spare FPs have the same card type and support dual-FP line APS. See NN10600-120 *Nortel Networks 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 Networks 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 Networks 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 Networks 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 Networks 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 Networks 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 409) 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.

Providing hitless ATM services on FPs with electrical interfaces

This section lists the main steps that must be done to provide hitless ATM services on Nortel Networks Multiservice Switch nodes using FPs with electrical interfaces. See NN10600-700 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Technology Fundamentals* for a description of which ATM services can be hitless.

1 Ensure that The main and spare FPs have the same card type and support one-for-one sparing. See NN10600-120 *Nortel Networks*

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 Networks 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 Networks Multiservice Switch 7400/15000/20000 Common Configuration Procedures*.

4 Configure the spare FP for one-for-one equipment sparing. See NN10600-550 *Nortel Networks 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 409) 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.

Adjusting resources for hitless services

The information in this section applies to Nortel Networks Multiservice Switch systems when the sparing status of an existing ATM interface changes from spared to unspared, or unspared to spared.

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

  Note: 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.

See NN10600-550 *Nortel Networks 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

Prerequisites

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 407) or "Providing hitless ATM services on FPs with electrical interfaces" (page 408). 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

This procedure shows the provisioning changes that are required when the sparing status of a single ATM interface changes from unspared to spared. Similar changes are required when the sparing status changes from spared to unspared. If the sparing status of many ATM interfaces are changing, repeat this procedure for each ATM interface that is affected.

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

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.

Variable definitions

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

Appendix C
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. For details, see

- “Calculating the value of the connectionPoolCapacity attribute” (page 414)

- “Calculating the value of the protectedConnectionPoolCapacity attribute” (page 416)
- “Calculating the value of the connectionPoolCapacity attribute for AQM based ATM FPs” (page 417)

In addition, special considerations are associated with some function processors. See for details, 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 417)
- “Considerations for the 4-port OC-3 ATM FP” (page 420)
- “Considerations for the 2-port OC-3 ATM and 2-port STM-1 electrical ATM FPs” (page 421)
- “Considerations for the 2-port OC-3 ATM and 2-port STM-1 electrical ATM FPs” (page 421)
- “Considerations for the 2-port OC-3 ATM and 2-port STM-1 electrical ATM FPs” (page 421)
- “Considerations for the 3-port OC-3 ATM FP” (page 422)
- “Considerations for the DS1 MSA32 FP and the E1 MSA32 FP” (page 423)

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 416).

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 417)
- "Considerations for the 4-port OC-3 ATM FP" (page 420)
- "Considerations for the 2-port OC-3 ATM and 2-port STM-1 electrical ATM FPs" (page 421)
- "Considerations for the 3-port OC-3 ATM FP" (page 422)
- "Considerations for the DS1 MSA32 FP and the E1 MSA32 FP" (page 423)

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 40).

To adjust the value of the *maxVpcs*, *maxVpts* and *maxVccs* attributes, see "Configuring connection admission control" (page 85).

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 414).

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 40).

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 417)
- “Considerations for the 4-port OC-3 ATM FP” (page 420)
- “Considerations for the 2-port OC-3 ATM and 2-port STM-1 electrical ATM FPs” (page 421)
- “Considerations for the 3-port OC-3 ATM FP” (page 422)
- “Considerations for the DS1 MSA32 FP and the E1 MSA32 FP” (page 423)

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 40).

To adjust the value of the *maxVpcs*, *maxVpts* and *maxVccs* attributes, see "Configuring connection admission control" (page 85).

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 417).

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 provision 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 admission control" (page 85).
- provision 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 provisioned on ports 0, 1, and 2:

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

$$\begin{aligned} CPC(Aqm/0) &\geq (3 \times 255) + (3 \times 128) + (3 \times 128 \times 3) + 1 \times 3 \\ &\geq 765 + 384 + 1152 + 3 \\ &\geq 2304 \\ CPC &= 768 \times 4 \\ CPC &= 3072 \end{aligned}$$

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

  Note: You need to provision Aqms and adjust their values prior to performing a check prov for this card.

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 provisioned on each port where:

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

$$
\begin{aligned}
CPC(Aqm/0) &\geq 255 + 128 + (3 \times 128) + 1 \\
&\geq 768 \\
CPC &\geq 768 \times 4 \\
CPC &\geq 3072
\end{aligned}
$$

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

> ***Note:*** There are situations where you want to have an unequal distribution of connection capacity between ports. In this case you have to correctly set the value of the *connectionPoolCapacity* attribute under the *Lp/x Eng Arc Aqm/n Ov* component of each AQM. You need to add an Aqm component under the *Lp/x Eng Arc* component if it is not already provisioned.

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

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

CPC(Aqm/0) ≥ 511 + 128 + (3 × 128) + 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,

$$\begin{aligned} CPC(Aqm/0) &\geq [255 + 128) + (3 \times 128) + 1] \times 2 \\ &\geq 768 + 768 \\ &\geq 1536 \end{aligned}$$

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 provisioning 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 provisioned 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.

  Note: There are no CES connections on AQM 0 and AQM 2, therefore users do not need to take CES connections into consideration on these two AQMs.

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

- provision 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.

Nortel Networks Multiservice Switch 7400/15000/20000

ATM Configuration Management

Release 6.1

Publication: NN10600-710
Document status: Standard
Document version: 6.1S2
Document date: November 2004
Printed in Canada

NORTEL NETWORKS