Nortel Networks Multiservice Switch 7400/15000/20000

IP Configuration Management

NN10600-801

Document status: Standard
Document issue: 6.1S2
Document date: November 2004
Product release: Release 6.1
Job function: Operations
Type: NTP
Language type: U.S. English

Contents

About this document

NN10600-801 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Configuration Management* describes how to configure virtual routers, the Internet protocol (IP), and other protocols and services related to IP service in Nortel Networks Multiservice Switch systems.

This guide is for anyone who performs the following Multiservice Switch IP service tasks: planning, installation, provisioning, operating, and maintenance.

What's new in this document

The following features were added to this document:

- 4-port 10/100BaseT Ethernet FP (page 9)
- 8-port 10/100 BaseT Ethernet FP (page 9)
- MD5 Authentication (page 10)
- Multi-hop EBGP (page 10)
- Multi-protocol BGP route distribution (page 10)
- Protected default route (page 10)
- Router model enhancements (page 11)
- VIPR on 4-port gigabit Ethernet, 4-port 10/100 BaseT Ethernet, and 8-port 10/100 BaseT Ethernet FPs (page 11)
- VIPR on 16-port OC-3/STM-1 POS and ATM FP (page 11)
- Virtual router redundancy protocol (VRRP) on 4-port Gigabit Ethernet, 4-port 10/100 BaseT Ethernet, and 8-port 10/100 BaseT Ethernet FPs (page 11)
- VR control plane protection (CPP) (page 11)

Other changes made to this document include the following:

- Updated the section Defining protected default route (page 199) with a new value for route preference.

- Moved the following chapters to the NN10600-591 *Nortel Networks Multiservice Switch 7400/15000/20000 Layer 3 Traffic Management Configuration*:
  - — IP CoS
  - — IP DiffServ
  - — IP DiffServ migration
  - — IP flow filters
- 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*.
- Updated sections Configuring an ATM MPE soft PVC source to ATM MPE soft PVC destination connection (page 29), Configuring an ATM MPE soft PVC source to ATM UNI interface connection (page 33) and Configuring an ATM UNI interface soft PVC source to ATM MPE soft PVC destination connection (page 37) to indicate support for soft PVCs between an ATM UNI and ATM MPE.
- Added a new attribute *sHoldOffTime* to procedure Configuring an ATM MPE soft PVC source to ATM MPE soft PVC destination connection (page 29)
- Added section Adding a frame relay DTE interface (page 44).
- Updated the section VIPR configuration (page 18) to remove references to Flat VR and Direct VR.

4-port 10/100BaseT Ethernet FP

The following sections were updated for this feature:

- Prerequisites to VR Ethernet access media configuration (page 74)
- VR Ethernet access media configuration (page 74)
- Linking an Ethernet interface in VLAN mode (page 77)

8-port 10/100 BaseT Ethernet FP

The following sections were either added or updated for this feature:

- Linking an Ethernet interface in port mode (page 76)
- Linking an Ethernet interface in VLAN mode (page 77)
- Monitoring the Ethernet access media configuration (page 256)

The following sections were updated for this feature:

- Prerequisites to VR Ethernet access media configuration (page 74)
- VR Ethernet access media configuration (page 74)
- VRRP configuration (page 218)
- Enabling VRRP on participating Multiservice Switch VRs (page 220)
- Creating a VRRP virtual router (page 221)
- Setting the VRRP advertisement interval (page 230)
- Defining critical IP interfaces (page 232)

MD5 Authentication

The following sections were added for this feature:

- Changing an MD5 key on an authenticated OSPF link (page 139)
- Changing an MD5 key on an authenticated OSPF virtual link (page 148)
- Changing an MD5 key on an authenticated BGP-4 peer (page 163)
- Monitoring the BGP-4 MD5 Authentication (page 283)

The following sections were updated for this feature:

- Configuring an OSPF area (page 133)
- Configuring an OSPF interface (page 135)
- Configuring a BGP-4 peer (page 159)
- Configuring OSPF virtual links (page 145)

Multi-hop EBGP

The following sections were updated for this feature:

- Configuring a BGP-4 peer (page 159)

Multi-protocol BGP route distribution

The following sections were updated for this feature:

- Configuring BGP-4 import policy (page 165)
- Configuring BGP-4 export policy (page 170)

Protected default route

The following sections were updated for this feature:

- VR static route configuration (page 188)

Router model enhancements

The following section was updated for this feature:

- VR PPP access media configuration (page 84)

VIPR on 4-port gigabit Ethernet, 4-port 10/100 BaseT Ethernet, and 8-port 10/100 BaseT Ethernet FPs

The following sections were added for this feature:

- VR Ethernet access media configuration procedures (page 75)
- Linking an Ethernet interface in port mode (page 76)
- Linking an Ethernet interface in VLAN mode (page 77)

VIPR on 16-port OC-3/STM-1 POS and ATM FP

The following sections were updated for this feature:

- Configuring an ATM PVC for an ATM MPE interface (page 26)
- Configuring an ATM MPE soft PVC source to ATM MPE soft PVC destination connection (page 29)

Virtual router redundancy protocol (VRRP) on 4-port Gigabit Ethernet, 4-port 10/100 BaseT Ethernet, and 8-port 10/100 BaseT Ethernet FPs

The following sections were updated for this feature:

- Creating a VRRP virtual router (page 221)
- Example of creating a VRRP virtual router associated with an Ethernet interface in port mode (page 224)
- Example of creating a VRRP virtual router associated with a VLAN on an Ethernet interface (page 227)
- Defining critical IP interfaces (page 232)

VR control plane protection (CPP)

The following sections were added for this feature:

- VR IP features configuration (page 203)
- Configuring IP CPP on the VR (page 234)
- Monitoring IP CPP on the VR (page 305)

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 12) or Provisioning mode (page 12).
- 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 Completing configuration changes (page 13).

Operational mode

Procedures contained within this document can be performed in operational mode or provisioning mode. When you initially log into a Nortel Networks Multiservice Switch node, you are in operational mode. Multiservice Switch nodes 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, use the start Prov command. Only one user can be in provisioning mode at a time. Nortel Networks Multiservice Switch nodes 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 Completing configuration changes (page 13).

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

Completing 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. Follow this procedure in provisioning mode when asked to complete the configuration changes. See the section Provisioning mode (page 12) 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 with portable formats:

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

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

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

4 End the provisioning session:

```
end Prov
```

Request for comments (RFCs)

The following Requests for Comments (RFCs) containing information related to IP are available from numerous sources including Internet Network Information Center (NIC) servers:

- RFC761, *DoD standard Transmission Control Protocol*
- RFC768, *User Datagram Protocol*
- RFC0791, *Internet Protocol*
- RFC792, *Internet Control Message Protocol*
- RFC793, *Transmission Control Protocol*

- RFC815, *IP Datagram Reassembly Algorithms*
- RFC821, *Simple Mail Transfer Protocol*
- RFC826, *An Ethernet Address Resolution Protocol*
- RFC854, *Telnet Protocol Specifications*
- RFC904, *Exterior Gateway Protocol Formal Specification*
- RFC950, *Internet Standard Subnetting Procedure*
- RFC951, *Bootstrap Protocol (BootP)*
- RFC959, *File Transfer Protocol*
- RFC1009, *Requirements for Internet Gateways*
- RFC1038, *Draft Revised IP Security Option*
- RFC1042, *Standard for Transmission of IP Datagrams over IEEE 802 Networks*
- RFC1122, *Requirements for Internet Hosts - Communication Layers*
- RFC1157, *Management Information Base for Network Management of TCP/IP-based Internets*
- RFC1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*
- RFC1253, *OSPF Version 2 Management Information Base*
- RFC1354, *IP Forwarding Table MIB*
- RFC1517, *Applicability Statement For the Implementation of Classless Inter-Domain Routing (CIDR)*
- RFC1518, *An Architecture for IP Address Allocation with CIDR*
- RFC1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*
- RFC1541, *Dynamic Host Configuration Protocol*
- RFC1577, *Classical IP and ARP over ATM*
- RFC1583, *OSPF Version 2*
- RFC1657, *Border Gateway Protocol version 4 (BGP-4) MIB*
- RFC1701, *Generic Routing Encapsulation*
- RFC1702, *Generic Routing Encapsulation over IPv4 networks*
- RFC1723, *RIP Version 2 Carrying Additional Information*
- RFC1724, *RIP Version 2 MIB Extension*
- RFC1745, *BGP4/IDRP for IP-OSFP Interaction*
- RFC1771, *Border Gateway Protocol 4 (BGP-4*)

- RFC1772, *Application of the Border Gateway Protocol in the Internet*
- RFC2003, *IP Encapsulation within IP*
- RFC2334, *Server Cache Synchronization Protocol*
- RFC2474, *DiffServ Field Definition*
- RFC2597, *Assured Forwarding PHB Group*
- RFC3246, *An Expedited Forwarding PHB*

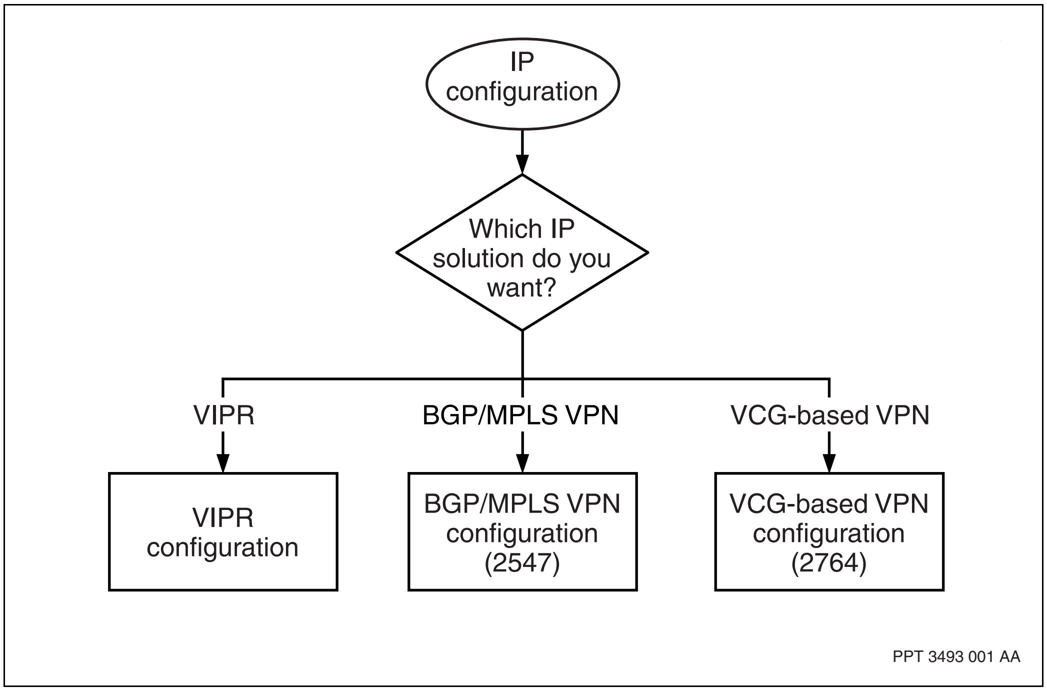
IP configuration

Configure Internet Protocol (IP) to enable Nortel Networks Multiservice Switch nodes to provide IP virtual private network (VPN) capabilities across Multiservice Switch networks. Multiservice Switch nodes use virtual routers (VRs) and routers (RTRs) to provide IP connectivity between nodes.

IP configuration tasks

This work flow shows you the sequence of tasks you perform to configure IP. To link to any task, go to IP configuration task navigation (page 16).

IP configuration tasks


IP configuration
Which IP solution do you want?
VIPR
BGP/MPLS VPN
VCG-based VPN
VIPR configuration
BGP/MPLS VPN configuration (2547)
VCG-based VPN configuration (2764)
PPT 3493 001 AA

IP configuration task navigation

- VIPR configuration (page 18)

- For information on configuring BGP/MPLS VPN, see NN10600-582 *Nortel Networks Multiservice Switch 7400/15000/20000 VPN Configuration Management*
- For information on configuring VCG-based VPN, see NN10600-582 *Nortel Networks Multiservice Switch 7400/15000/20000 VPN Configuration Management*

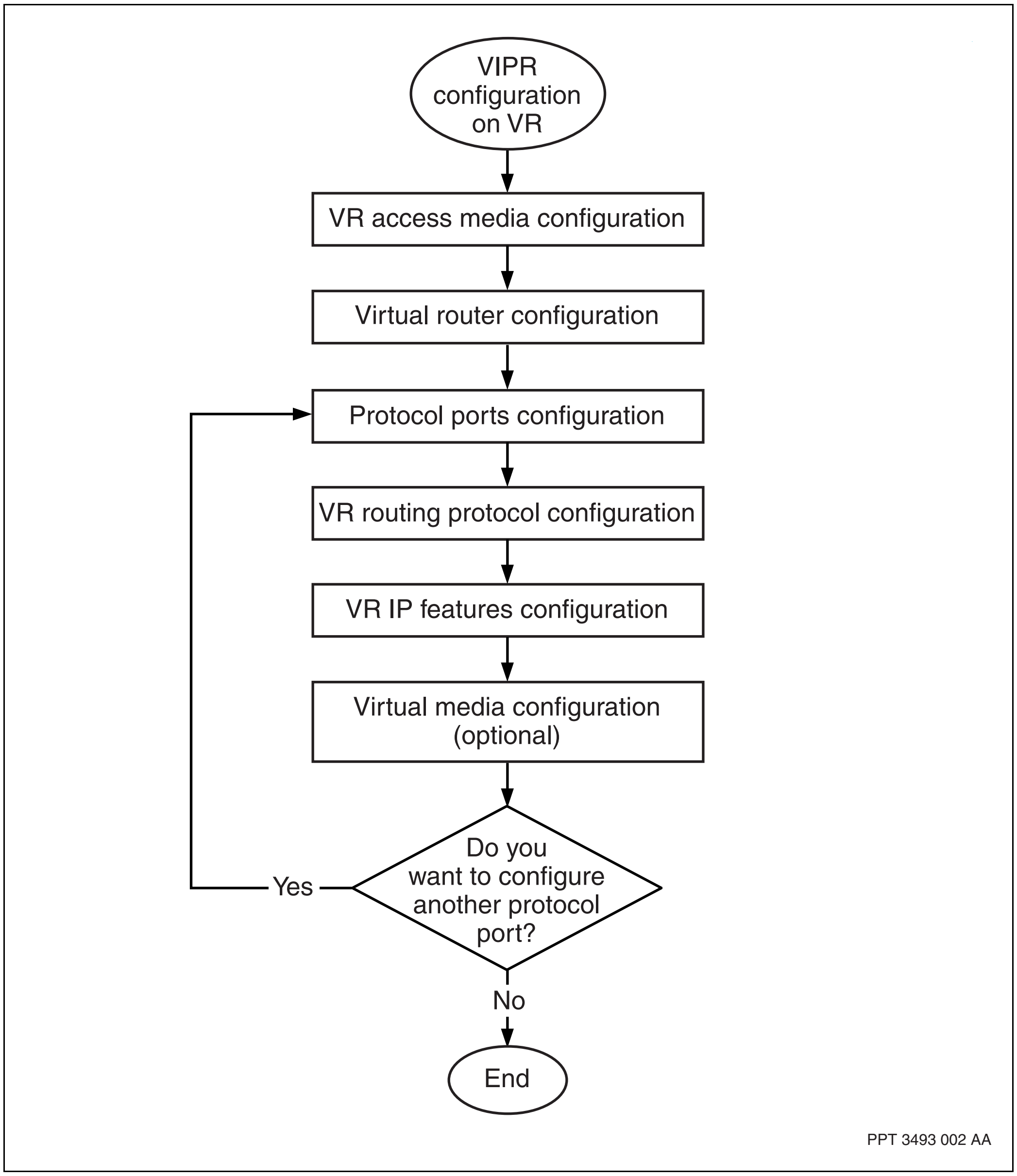
VIPR configuration

Configure the virtual IP router (VIPR) virtual router (VR) to provide a software emulation of physical routers. The two main functions of a VR are: constructing routing tables describing the paths to networks or subnetworks, and forwarding or switching packets to the final destination network or subnetwork.

VIPR configuration on VR tasks

This work flow shows you the sequence of tasks you perform to configure the VR model. To link to any task, go to VIPR configuration on VR task navigation (page 19).

VIPR configuration on VR tasks


VIPR configuration on VR
VR access media configuration
Virtual router configuration
Protocol ports configuration
VR routing protocol configuration
VR IP features configuration
Virtual media configuration (optional)
Do you want to configure another protocol port?
Yes
No
End
PPT 3493 002 AA

VIPR configuration on VR task navigation

- VR access media configuration (page 20)
- Virtual router configuration (page 78)
- Protocol ports configuration (page 97)
- VR routing protocol configuration (page 105)
- VR IP features configuration (page 203)
- Virtual media configuration (page 237)

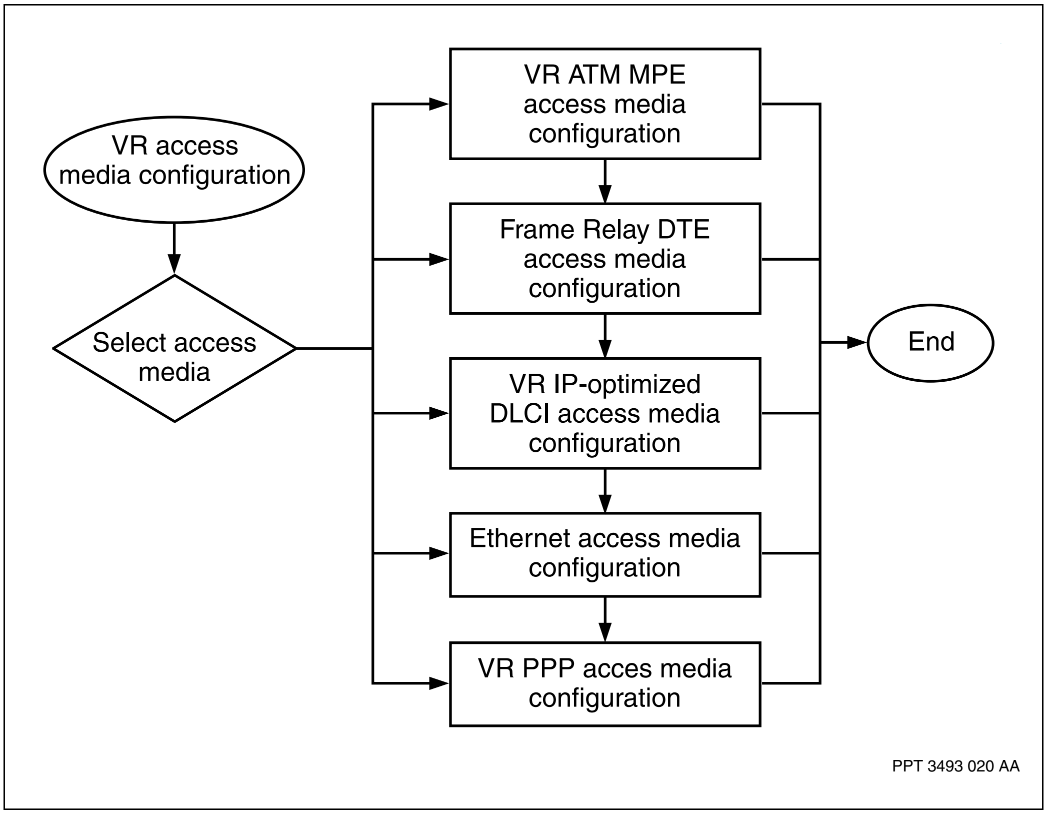
VR access media configuration

Configure VR access media to provide customer access to the network using IP media. For conceptual information about types of access media, see IP media in NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Technology Fundamentals*.

VR access media configuration tasks

This work flow shows you the sequence of tasks you perform to configure access media on the VR model. To link to any task, go to VR access media configuration task navigation (page 21).

VR access media configuration tasks



VR access media configuration
Select access media
VR ATM MPE access media configuration
Frame Relay DTE access media configuration
VR IP-optimized DLCI access media configuration
Ethernet access media configuration
VR PPP acces media configuration
End
PPT 3493 020 AA

VR access media configuration task navigation

- VR ATM MPE access media configuration (page 22)
- Frame relay DTE access media configuration (page 41)
- VR IP-optimized DLCI access media configuration (page 60)
- VR Ethernet access media configuration (page 74)
- VR PPP access media configuration (page 84)

VR ATM MPE access media configuration

Configure VR ATM multiprotocol encapsulation (MPE) access media to allow IP traffic to be transmitted across the ATM network using two types of ATM MPE media: permanent virtual circuits (PVCs) and soft PVCs.

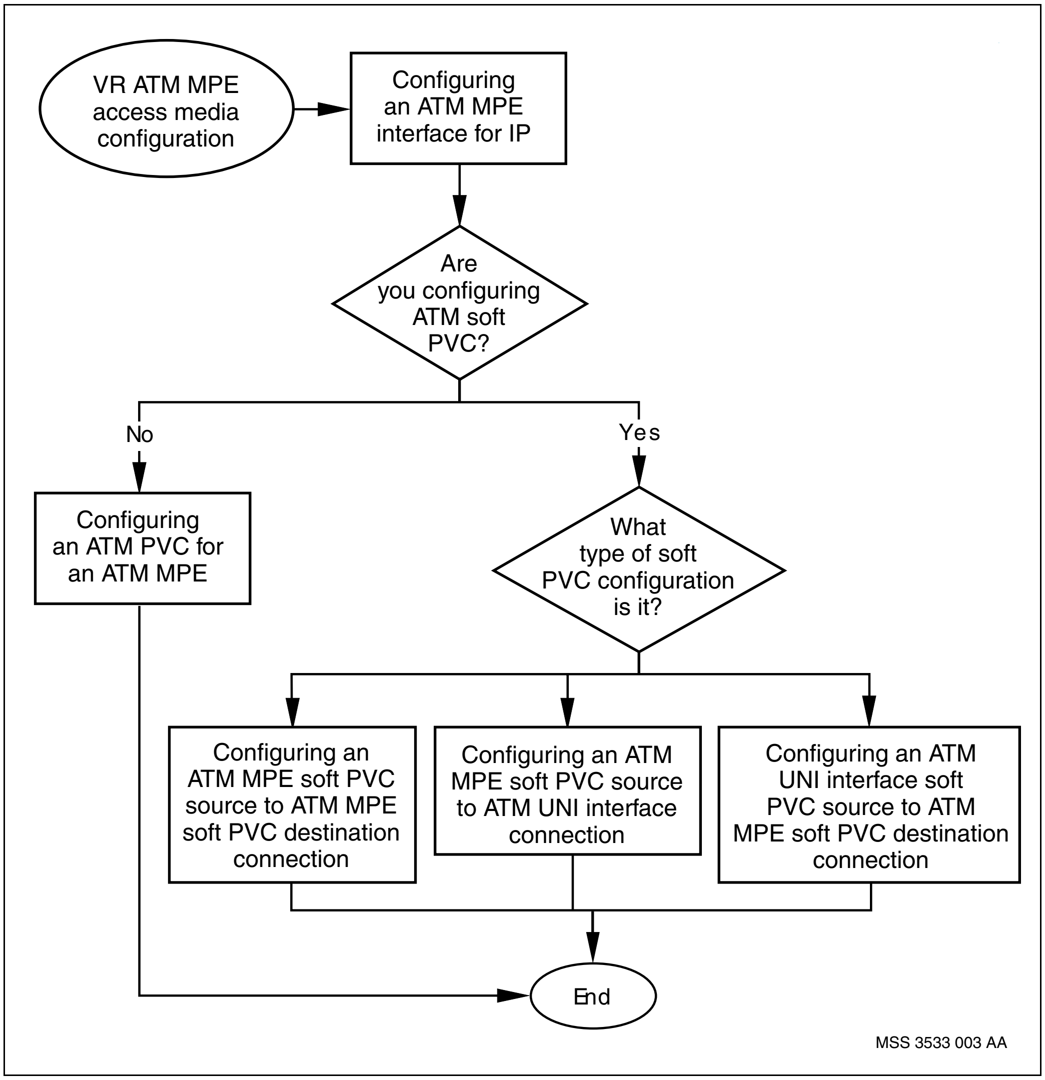
Prerequisites to VR ATM MPE access media configuration

- Configure the required ATM interfaces and connections. See NN10600-710 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Configuration Management*.

VR ATM MPE access media configuration procedures

This task flow shows you the sequence of procedures you perform to configure VR ATM MPE access media. To link to any procedure, go to VR ATM MPE access media configuration procedure navigation (page 23).

VR ATM MPE access media configuration procedures


VR ATM MPE access media configuration
Configuring an ATM MPE interface for IP
Are you configuring ATM soft PVC?
No
Yes
Configuring an ATM PVC for an ATM MPE
What type of soft PVC configuration is it?
Configuring an ATM MPE soft PVC source to ATM MPE soft PVC destination connection
Configuring an ATM MPE soft PVC source to ATM UNI interface connection
Configuring an ATM UNI interface soft PVC source to ATM MPE soft PVC destination connection
End
MSS 3533 003 AA

VR ATM MPE access media configuration procedure navigation

- Configuring an ATM MPE interface for IP traffic (page 24)
- Configuring an ATM PVC for an ATM MPE interface (page 26)
- Configuring an ATM MPE soft PVC source to ATM MPE soft PVC destination connection (page 29)
- Configuring an ATM MPE soft PVC source to ATM UNI interface connection (page 33)
- Configuring an ATM UNI interface soft PVC source to ATM MPE soft PVC destination connection (page 37)

Configuring an ATM MPE interface for IP traffic

Configure an ATM multiprotocol encapsulation (MPE) interface for IP traffic to provide an ATM MPE connection between a Nortel Networks Multiservice Switch node and the IP network.

Procedure steps

Step	Action
1	Create an instance of the ATM MPE service. **`add AtmMpe/<n>`** When you create the *AtmMpe* component, the Multiservice Switch system automatically adds an instance of the *AtmConnection* (*Ac*) component, *Ac/1*, under the *AtmMpe* component.
2	Specify the maximum transmission unit (MTU) size to be used for ATM connections on this interface. **`set AtmMpe/<n> mtu <size>`**
3	Specify the encapsulation type to be used for ATM connections on this interface. **`set AtmMpe/<n> encapType <type>`**

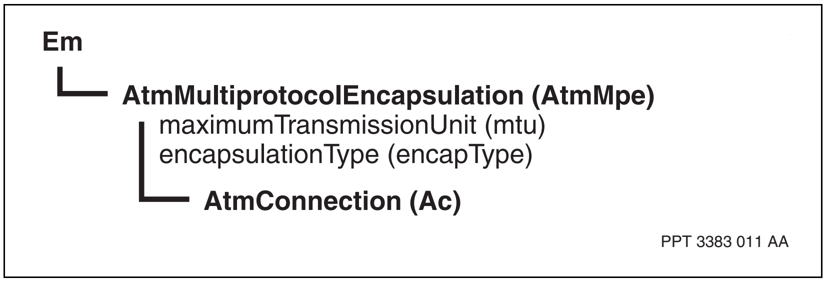
--End--

Variable definitions

Variable	Value
<n>	is the instance number of the ATM MPE interface.
<size>	is the size of the largest datagram that can be sent on the interface.
<type>	is the encapsulation type defined for ATM connections on the interface. If you set the encapsulation type to *ipVcEncap*, your must configure a static ARP entry to ensure IP connectivity across the ATM network. For more information of static ARP, see Configuring VR static ARP (page 184). The *ipVcEncap* attribute value is supported only for ATM MPE PVCs, not for soft PVCs.

Procedure job aid

ATM MPE interface for IP traffic component hierarchy


Em
AtmMultiprotocolEncapsulation (AtmMpe)
maximumTransmissionUnit (mtu)
encapsulationType (encapType)
AtmConnection (Ac)
PPT 3383 011 AA

Configuring an ATM PVC for an ATM MPE interface

Configure an ATM permanent virtual circuit (PVC) for an ATM multiprotocol encapsulation (MPE) interface to support full-mesh connectivity between VRs or virtual connection gateways (VCGs) using nailed-up connection points.

Prerequisites

Attention: ATM MPE over PVCs is supported on CQC, GQM-based, and PQC-based FPs.

- Each *AtmMpe* component must have at least one *AtmConnection* component associated with a VCC that connects to every other Nortel Networks Multiservice Switch node that uses the ATM MPE service.
- Each VCC (VPI.VCI) can be associated with the ATM interface or with a virtual path terminator (VPT). If the VCC is associated with a VPT, the VPI value is the instance of the *Vpt* component and the VCI value is the instance of the *Vcc* component. See NN10600-710 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Configuration Management*.
- If you are using VC-based multiplexing for IP traffic, you must configure a static ARP entry for the ATM VCC to ensure IP connectivity across the ATM network. Since VC-based multiplexing supports traffic for only one protocol type (in this case, IP), ARP packets cannot transmit on the ATM MPE service. See Configuring VR static ARP (page 184).

Procedure steps

Step **Action**

1 Optionally, create an instance of an ATM connection for the ATM MPE service for the benefit of redundancy.

```
add AtmMpe/<n> Ac/<conn>
```

When you create the *AtmMpe* component, the Multiservice Switch system automatically adds an instance of the *AtmConnection* (*Ac*) component, *Ac/1*, under the *AtmMpe* component. Add the *Ac* component only when you need another connection after *Ac/1*.

Do not change the attribute *AtmMpe Ac mplsSig* from its default value of *shared*. Setting this attribute to *dedicated* dedicates the connection to MPLS signalling.

2 Configure a VCC under an ATM interface, if one is not already available. This VCC is directly associated with the ATM MPE service and resides on the same node.

```
add AtmIf/<i> [Vpt/<Vpi>] Vcc/<vc>
```

3 Create a nailed-up endpoint (NEP) for the VCC, if one is not already available.

```
add AtmIf/<i> [Vpt/<Vpi>] Vcc/<vc> Nep
```

4 Link the ATM MPE service to the ATM VCC.

```
set AtmMpe/<n> Ac/<conn> atmConnection AtmIf/<i>
[Vpt/<Vpi>] Vcc/<vc> Nep
```

5 If you are using an ILS forwarder FP, link the ATM MPE service to the ILS forwarder card.

```
set AtmMpe/<n> ilsForwarder Lp/<z> IlsFwdr/<f>
```

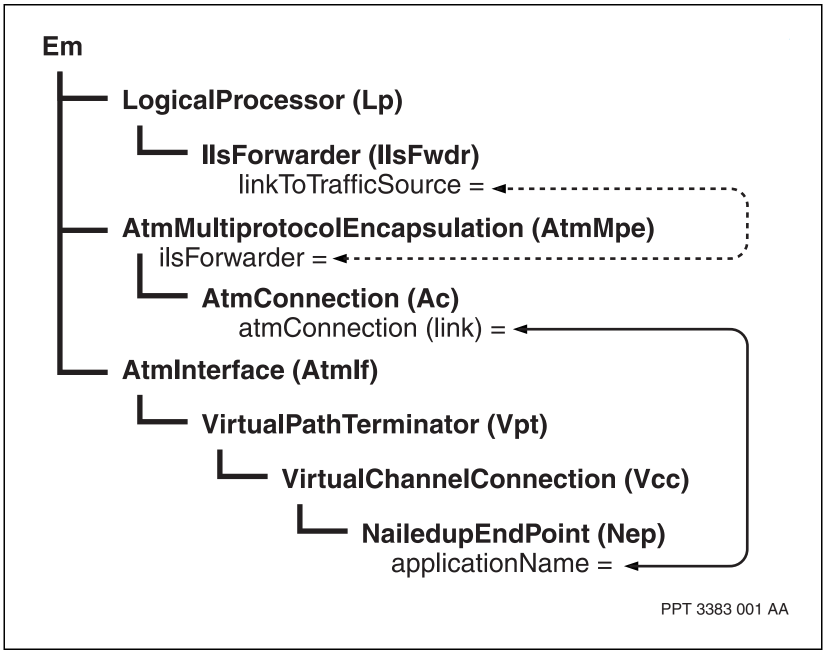
--End--

Variable definitions

Variable	Value
<conn>	is the instance number of the ATM connection on the ATM MPE interface.
<f>	is the instance number of the ILS forwarder component under the logical interface.
<i>	is the instance number of the ATM interface.
<n>	is the instance number of the ATM MPE interface.
<vc>	is the instance value of the VCC. If the virtual channel is associated with a VPT, this value is the VCI value.
[Vpt/<Vpi>]	is the *VirtualPathTerminator* (*Vpt*) component instance.
<z>	is the instance number of the logical processor (LP) interface.

Procedure job aid

ATM PVC for an ATM MPE interface component hierarchy



Configuring an ATM MPE soft PVC source to ATM MPE soft PVC destination connection

Configure an ATM soft permanent virtual circuit (PVC) for an ATM multiprotocol encapsulation (MPE) interface to support full-mesh connectivity between customer virtual routers (VRs) in a PNNI network, or virtual connection gateways (VCGs) across the backbone.

Attention: ATM MPE over soft PVCs is supported on GQM-based and PQC-based FPs.

Attention: ATM MPE over soft PVCs is not supported on CQC-based FPs.

Prerequisites

- Configure the *Pnni* component under the *AtmRouting* and *AtmInterface* components. See NN10600-710 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Configuration Management*.
- When provisioning ATM MPE soft PVCs, the called/destination endpoint should be provisioned first. This allows the default NSAP address of the called/destination endpoint ATM MPE to be determined for the purpose of provisioning the calling/source endpoint.

Procedure steps

Step	Action
1	Optionally, create an instance of an ATM connection for the ATM MPE service for the benefit of redundancy. **`add AtmMpe/<n> Ac/<conn>`** When you create the *AtmMpe* component, the Multiservice Switch system automatically adds an instance of the *AtmConnection* (*Ac*) component (*Ac/1*) under the *AtmMpe* component. Add the *Ac* component only when you need another connection after *Ac/1*. Do not change the attribute *AtmMpe Ac mplsSig* from its default value of *shared*. Setting this attribute to *dedicated* dedicates the connection to MPLS signalling.
2	Create the soft PVC termination point. **`add AtmMpe/<n> Stp`**
3	If this is the calling end of the soft PVC, define the retry period. **`set AtmMpe/<n> Stp retry <period>`**

4 Optionally, if this is the calling end of the soft PVC, define the time value used to wait before an attempt is made to reestablish a connection after an established connection has been cleared/released.

```
set AtmMpe/<n> Stp sHoldOffTime <holdOffTime>
```

5 Optionally, define the local NSAP address if you are not using the default address.

```
set AtmMpe/<n> Stp laddr <nsap_addr>
```

6 Provision the endpoint as the calling or called endpoint of the soft PVC.

```
add AtmMpe/<n> Ac/<conn> SrcPvc
```

or

```
add AtmMpe/<n> Ac/<conn> DstPvc
```

When you provision the endpoint as a source, or calling, endpoint, the Multiservice Switch system automatically adds a *TrafficManagement* (*Tm*) component under the *SrcPvc* component.

7 If this is the calling end of the soft PVC, display the *opLocalAddress* attribute of the *AtmMpe Stp* component at the remote end to determine the NSAP address at the remote end of the connection.

```
d AtmMpe/<n> Ac/<conn> Stp opLocalAddress
```

8 If this is the calling end of the soft PVC, set the remote address of the source to the NSAP address at the remote end of the connection. This address must match the provisioned or default NSAP address at the remote end.

```
set AtmMpe/<n> Ac/<conn> SrcPvc raddr <nsap_addr>
```

9 If this is the calling end of the soft PVC, set the remote connection identifier. The *remoteConnectionIdentifier* is the instance of the *AtmMpe Ac* component at the remote end of the connection.

```
set AtmMpe/<n> Ac/<conn> SrcPvc rci <remote_conn>
```

10 If this is the calling end of the soft PVC, define the ATM service category for the connection.

```
set AtmMpe/<n> Ac/<conn> SrcPvc Tm service <category>
```

11 If this is the calling end of the soft PVC, define the peak cell rate (PCR) for both directions of the connection.

```
set AtmMpe/<n> Ac/<conn> SrcPvc Tm pcr <rate>
```

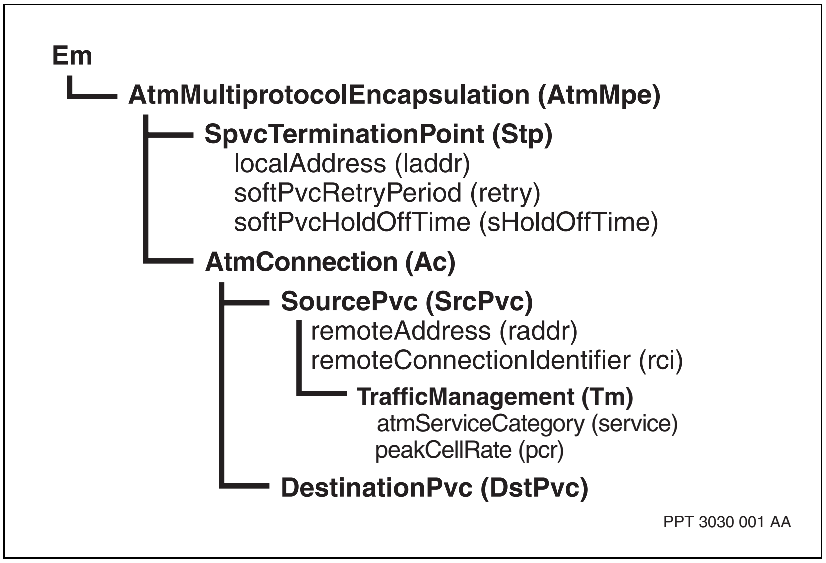
--End--

Variable definitions

Variable	Value
<category>	is the service category of the calling end of the soft PVC. When you define a service category for the soft PVC, the BBC IE parameters are derived from the service category. For more information on ATM traffic management, see NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*.
<conn>	is the instance number of the ATM connection on the ATM MPE interface.
<holdOffTime>	The time, in milliseconds, to hold off before attempting to reestablish a connection after an established soft PVC connection is cleared/released.
<n>	is the instance number of the ATM MPE interface.
<nsap_addr>	is a valid 40-digit NSAP address for the *AtmMpe* component. The default is an empty string. You should normally use the default NSAP address for each *AtmMpe* component instance. This practice ensures that any hierarchical routing in the PNNI network is maintained. If you do not enter an address, the system supplies a default. For more information on NSAP addressing, see NN10600-702 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals*.
<period>	is the value that determines how long the calling end should wait after a failed setup attempt before making the next setup request. If you have several ATM MPE applications provisioned on a Multiservice Switch node, you should provision staggered *retry* timers. This practice ensures that the CP is not stressed by several ATM MPE applications simultaneously trying to reestablish their soft PVCs after failure of a common interface.
<rate>	is the value of the peak cell rate. When you define a PCR for the soft PVC, the PCR0+1 traffic descriptor IE parameter is configured. All other parameters are derived from the service category. For more information on ATM traffic management, see NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*
<remote_conn>	is an integer in the range 1 to 255. The default is 1.

Procedure job aid

ATM MPE soft PVC source to ATM MPE soft PVC destination connection component hierarchy


Em
AtmMultiprotocolEncapsulation (AtmMpe)
SpvcTerminationPoint (Stp)
localAddress (laddr)
softPvcRetryPeriod (retry)
softPvcHoldOffTime (sHoldOffTime)
AtmConnection (Ac)
SourcePvc (SrcPvc)
remoteAddress (raddr)
remoteConnectionIdentifier (rci)
TrafficManagement (Tm)
atmServiceCategory (service)
peakCellRate (pcr)
DestinationPvc (DstPvc)
PPT 3030 001 AA

Configuring an ATM MPE soft PVC source to ATM UNI interface connection

Configure an ATM multiprotocol encapsulation (MPE) interface soft PVC source to ATM UNI soft PVC destination connection over a PNNI network, in the case where only a segment of the end-to-end ATM connection is over a PNNI network, and the soft PVC segment of the ATM connection terminates on an ATM UNI interface.

Attention: ATM MPE over soft PVCs is not supported on CQC-based FPs. In this case, it applies to the ATM MPE source endpoint.

Attention: ATM MPE over soft PVCs is supported on GQM-based and PQC-based FPs.

Prerequisites

- Configure the *Pnni* component under the *AtmRouting* and *AtmInterface* components, and the *Uni* component under the *AtmInterface* component. See NN10600-710 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Configuration Management*.

Procedure steps

Step	Action

1 Optionally, create an instance of an ATM connection for the ATM MPE service on the source node.

```
add AtmMpe/<n> Ac/<conn>
```

When you create the *AtmMpe* component, the Multiservice Switch system automatically adds an instance of the *AtmConnection* (*Ac*) component, *Ac/1*, under the *AtmMpe* component.

Do not change the attribute *AtmMpe Ac mplsSig* from its default value of *shared*. Setting this attribute to *dedicated* dedicates the connection to MPLS signalling.

2 Create the soft PVC termination point on the source node.

```
add AtmMpe/<n> Stp
```

3 Define the retry period on the source node.

```
set AtmMpe/<n> Stp retry <period>
```

4 Optionally, on the source node, define the time value used to wait before an attempt is made to reestablish a connection after an established connection has been cleared/released.

```
set AtmMpe/<n> Stp sHoldOffTime <holdOffTime>
```

5 Provision the endpoint as the calling endpoint of the soft PVC on the source node.

```
add AtmMpe/<n> Ac/<conn> SrcPvc
```

When you provision the endpoint as a source, or calling, endpoint, the Multiservice Switch system automatically adds a *TrafficManagement* (Tm) component under the *SrcPvc* component.

6 Perform the following on the destination node where the remote AtmIf exists to determine the NSAP address at the remote end of the connection.

```
display -o AtmIf/<i> Uni Addr/*,default
```

7 On the source node, set the remote address of the source to the NSAP address of the remote end of the connection. This is the address of the ATM UNI interface (*AtmIf Uni*) where the VCC connection terminates.

```
set AtmMpe/<n> Ac/<conn> SrcPvc raddr <nsap_addr>
```

8 Set the remote connection identification on the source node. The remoteCi is the VCI (Virtual Channel Identifier) of the remote UNI *AtmIf Vcc*. The VPI (Virtual Path Identifier) is always 0 in this case. If an *AtmIf Vcc Dst* component is provisioned, this represents the VCI value of that Vcc. If no *AtmIf Vcc Dst* component is provisioned, the *AtmIf Vcc Dst* component is dynamically created, and the VCI value of the *Vcc* is the remoteCi value.

```
set AtmMpe/<n> Ac/<conn> SrcPvc rci <remote_vci>
```

9 Define the ATM service category for the connection on the source node.

```
set AtmMpe/<n> Ac/<conn> SrcPvc Tm service <category>
```

10 Define the peak cell rate (PCR) for both directions of the connection on the source node.

```
set AtmMpe/<n> Ac/<conn> SrcPvc Tm pcr <rate>
```

11 Optionally, add a VCC destination point at the remote end of the connection on the UNI AtmIf on the destination node.

```
add AtmIf/<i> Vcc/0.<remote_vci> Dst
```

The *AtmIf Vcc Vcd Tm* component on destination node should be configured to match the *AtmMpe Ac SrcPvc Tm* component parameters on the source node. For more information, see NN10600-710 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Configuration Management*.

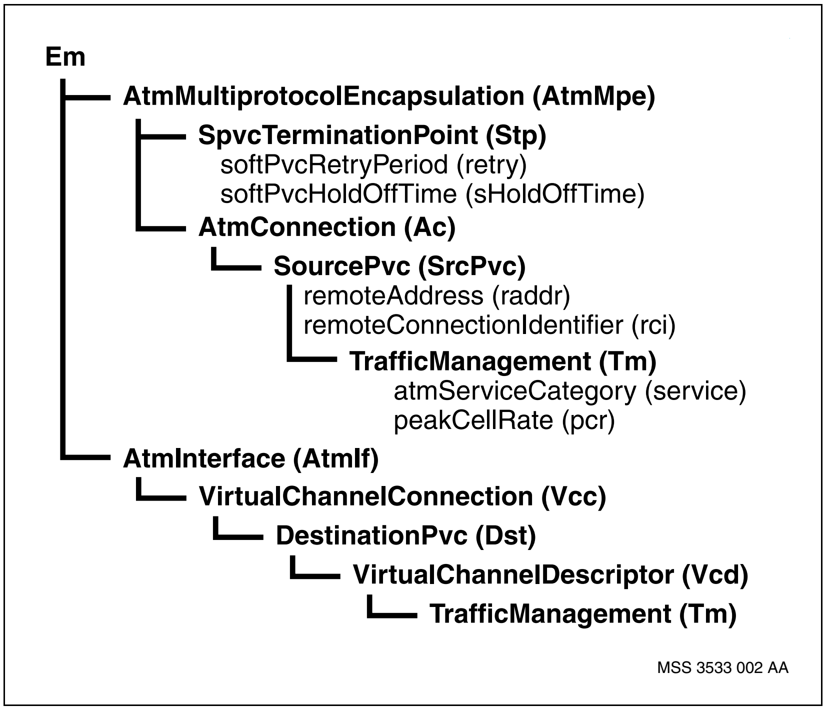
--End--

Variable definitions

Variable	Value
<category>	is the service category of the calling end of the soft PVC. When you define a service category for the soft PVC, the BBC IE parameters are derived from the service category. For more information on ATM traffic management, see NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*.
<conn>	is the instance number of the ATM connection on the ATM MPE interface.
<holdOffTime>	is the time, in milliseconds, to hold off before attempting to reestablish a connection after an established soft PVC connection is cleared/released.
<i>	is the instance number of the remote ATM interface where the ATM connection is to be terminated.
<n>	is the instance number of the ATM MPE interface.
<nsap_addr>	is a valid 40-digit NSAP address for the destination ATM UNI interface.
<period>	is the value that determines how long the calling end should wait after a failed setup attempt before making the next setup request. If you have several ATM MPE applications provisioned on a Multiservice Switch node, you should provision staggered *retry* timers. This practice ensures that the CP is not stressed by several ATM MPE applications simultaneously trying to reestablish their soft PVCs after failure of a common interface.
<rate>	is the value of the peak cell rate. When you define a PCR for the soft PVC, the PCR0+1 traffic descriptor IE parameter is configured. All other parameters are derived from the service category. For more information on ATM traffic management, see NN10600-705 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Traffic Management Fundamentals*
<remote_vci>	the remote VCC VCI instance value in the range of 32 to 255.

Procedure job aid

ATM MPE soft PVC source to ATM UNI interface connection


Em
AtmMultiprotocolEncapsulation (AtmMpe)
SpvcTerminationPoint (Stp)
softPvcRetryPeriod (retry)
softPvcHoldOffTime (sHoldOffTime)
AtmConnection (Ac)
SourcePvc (SrcPvc)
remoteAddress (raddr)
remoteConnectionIdentifier (rci)
TrafficManagement (Tm)
atmServiceCategory (service)
peakCellRate (pcr)
AtmInterface (AtmIf)
VirtualChannelConnection (Vcc)
DestinationPvc (Dst)
VirtualChannelDescriptor (Vcd)
TrafficManagement (Tm)
MSS 3533 002 AA

Configuring an ATM UNI interface soft PVC source to ATM MPE soft PVC destination connection

Configure an ATM UNI soft PVC source to ATM multiprotocol encapsulation (MPE) interface soft PVC destination connection over a PNNI network, in the case where only a segment of the end-to-end ATM connection is over a PNNI network, and the soft PVC segment of the ATM connection originates from an ATM UNI interface.

Attention: ATM MPE over soft PVCs is not supported on CQC-based FPs. In this case, it applies to the ATM MPE destination endpoint.

Attention: ATM MPE over soft PVCs is supported on GQM-based and PQC-based FPs.

Prerequisites

- Configure the *Pnni* component under the *AtmRouting* and *AtmInterface* components, and the *Uni* component under the *AtmInterface* component. See NN10600-710 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Configuration Management*.

Procedure steps

Step **Action**

1 Optionally, create an instance of an ATM connection for the ATM MPE service on the destination node.

```
add AtmMpe/<n> Ac/<conn>
```

When you create the *AtmMpe* component, the Multiservice Switch system automatically adds an instance of the *AtmConnection* (*Ac*) component, *Ac/1*, under the *AtmMpe* component.

Do not change the attribute *AtmMpe Ac mplsSig* from its default value of *shared*. Setting this attribute to *dedicated* dedicates the connection to MPLS signalling.

2 Create the soft PVC termination point on the destination node.

```
add AtmMpe/<n> Stp
```

3 Optionally, on the destination node, define the local NSAP address if you are not using the default address.

```
set AtmMpe/<n> Stp laddr <nsap_addr>
```

4 Provision the endpoint as the called endpoint of the soft PVC on the destination node.

```
add AtmMpe/<n> Ac/<conn> DstPvc
```

5 Activate the changes on the destination node.

6 Add a VCC soft PVC source connection endpoint on the source node.

```
add AtmIf/<i> [Vpt/<Vpi>] Vcc/<vc> Src
```

7 Obtain the NSAP address of the destination *AtmMpe*.

```
display AtmMpe/<n> Ac/<conn> Stp opLocalAddress
```

8 On the source node, set the called address of the *Src* component to the NSAP address of the *AtmMpe* destination point.

```
set AtmIf/<i> [Vpt/<Vpi>] Vcc/<vc> Src calledAddress
<nsap_addr>
```

9 On the source node, configure the component to connect to the remote *AtmMpe Ac* instance.

```
set AtmIf/<i> [Vpt/<Vpi>] Vcc/<vc> Src calledVpiVci
0.<conn>
```

The *calledVpiVci* attribute is used to do this. The VPI instance is always 0 and the VCI value represents the Ac instance of the *AtmMpe Ac* being connected to.

10 Optionally, on the source node, configure the VCC source traffic management parameters (*AtmIf Vcc Vcd Tm* or *AtmIf Vpt Vcc Vcd Tm*) and other A*tmIf Vcc Src* or *AtmIf Vpt Vcc Src* component attributes. For more information, see NN10600-710 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Configuration Management*.

--End--

Variable definitions

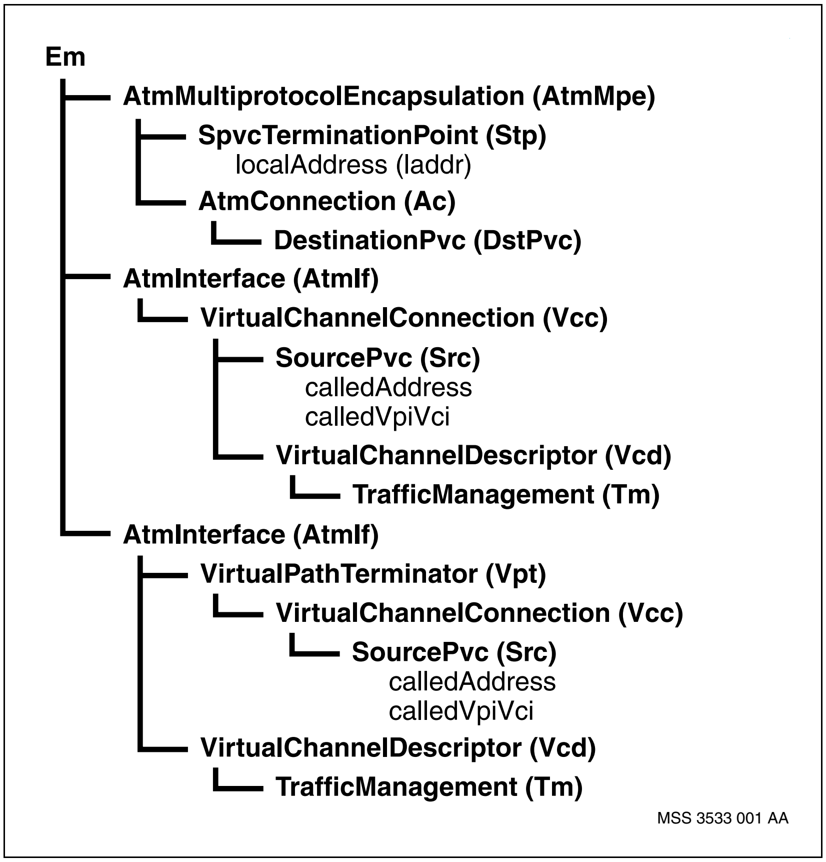
Variable	Value
<conn>	is the instance number of the ATM connection on the ATM MPE interface.
<i>	is the instance number of the ATM interface at the source end.
<n>	is the instance number of the ATM MPE interface.

(1 of 2)

Variable	Value
<nsap_addr>	is a valid 40-digit NSAP address for the *AtmMpe* component. The default is an empty string. You should normally use the default NSAP address for each *AtmMpe* component instance. This practice ensures that any hierarchical routing in the PNNI network is maintained. If you do not enter an address, the system supplies a default. For more information on NSAP addressing, see NN10600-702 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Routing and Signalling Fundamentals*.
<vc>	is the instance value of the VCC. If the virtual channel is associated with a VPT, this value is the VCI value.
[Vpt/<Vpi>]	is the *VirtualPathTerminator* (*Vpt*) component instance.
(2 of 2)	

Procedure job aid

ATM UNI interface soft PVC source to ATM MPE soft PVC destination connection



Frame relay DTE access media configuration

Configure frame relay data terminating device (DTE) to enable Nortel Networks Multiservice Switch nodes to carry IP over frame relay media.

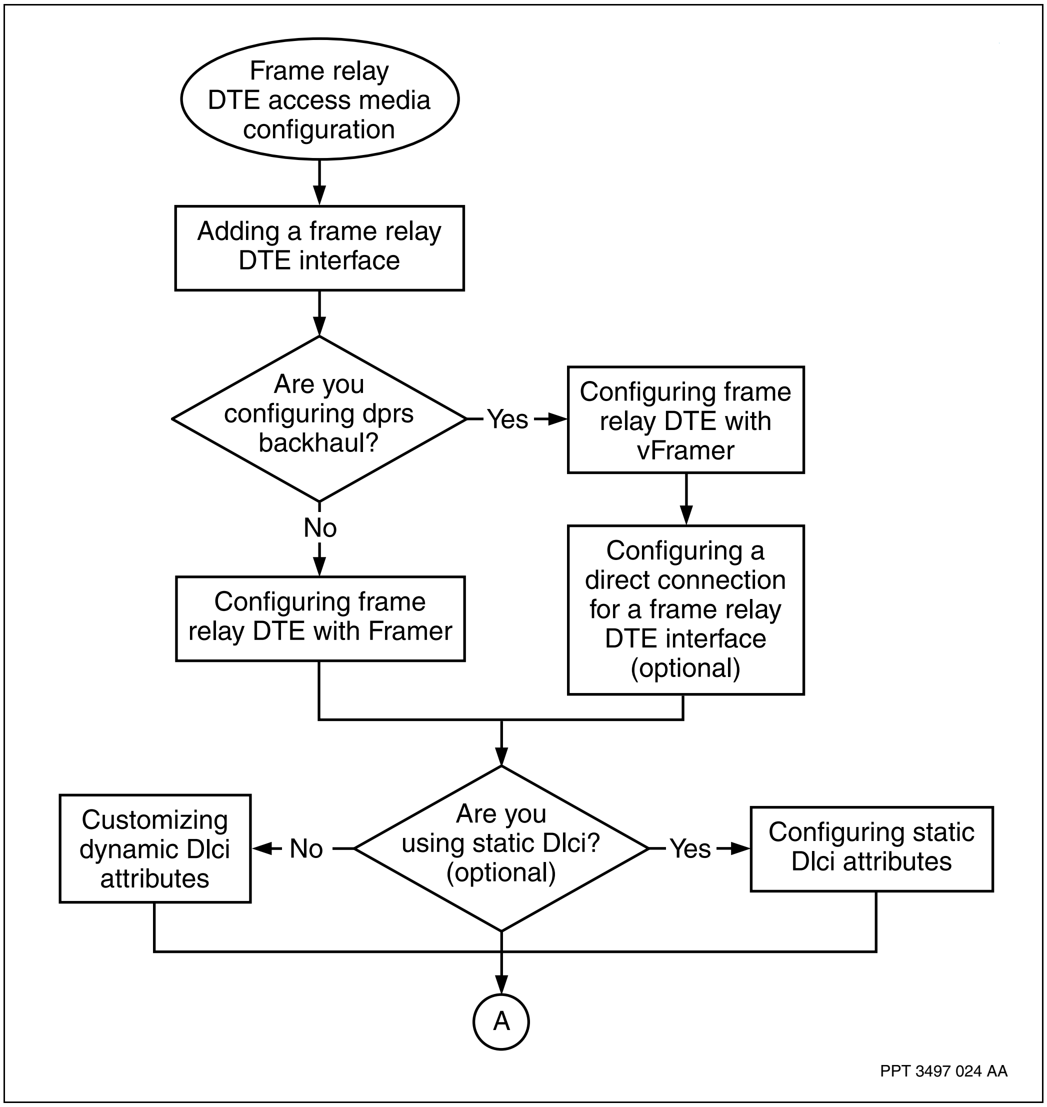
Prerequisites to frame relay DTE access media configuration

- Configure the required frame relay interfaces and connections. See NN10600-901 *Nortel Networks Multiservice Switch 7400/15000/20000 Frame Relay Configuration Management*.

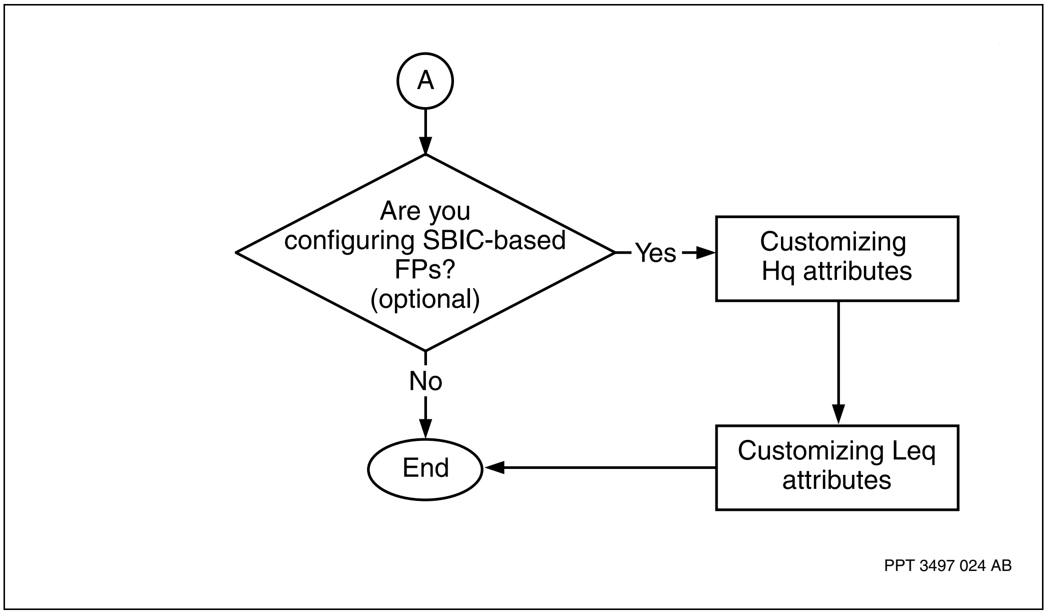
Frame relay DTE access media configuration procedures

This task flow shows you the sequence of procedures you perform to configure frame relay DTE. To link to any procedure, go to Frame relay DTE access media configuration procedure navigation (page 43).

Frame relay DTE access media configuration procedures


Frame relay DTE access media configuration
Adding a frame relay DTE interface
Are you configuring dprs backhaul?
Yes
Configuring frame relay DTE with vFramer
No
Configuring frame relay DTE with Framer
Configuring a direct connection for a frame relay DTE interface (optional)
Are you using static Dlci? (optional)
No
Customizing dynamic Dlci attributes
Yes
Configuring static Dlci attributes
A
PPT 3497 024 AA

Frame relay DTE access media configuration procedures (continued)


A
Are you configuring SBIC-based FPs? (optional)
Yes
No
Customizing Hq attributes
Customizing Leq attributes
End
PPT 3497 024 AB

Frame relay DTE access media configuration procedure navigation

- Adding a frame relay DTE interface (page 44)
- Configuring frame relay DTE with Framer (page 45)
- Configuring frame relay DTE with vFramer (page 47)
- Configuring a direct connection for a frame relay DTE interface (page 50)
- Customizing static Dlci attributes (page 52)
- Customizing dynamic Dlci attributes (page 54)
- Customizing Hq attributes (page 56)
- Customizing Leq attributes (page 58)

Adding a frame relay DTE interface

Configure a frame relay data terminating device (DTE) interface to provide a frame relay connection between a Nortel Networks Multiservice Switch node and the IP network.

Procedure steps

Step	Action
1	Add an FrDte link level protocol interface application. The attributes associated with this new interface have default values assigned automatically. One remote group (Rg/1) is created automatically when the *FrDte* component is added. **`add FrDte/<a>`**

--End--

Variable definitions

Variable	Value
<a>	is the number of the FrDte instance.

Configuring frame relay DTE with Framer

Configure a frame relay data terminating device (DTE) with Framer connection between a frame relay DTE interface and a frame relay UNI interface as an alternative to a logical or direct connection.

Procedure steps

Step **Action**

1 Add a logical processor (Lp).

```
add Lp/<e>
```

2 Add a v.35 port to the logical processor.

```
add Lp/<e> V35/<f>
```

3 Link the FrDte application to the hardware component.

```
set FrDte/<a> Framer interfaceName Lp/<e> V35/<f>
```

4 Add a static DLCI component.

```
add FrDte/<a> StDlci/<d>
```

5 Link the DLCI to a remote group.

```
set FrDte/<a> StDlci/<d> linkToRemoteGroup FrDte/<a> Rg/
<g>
```

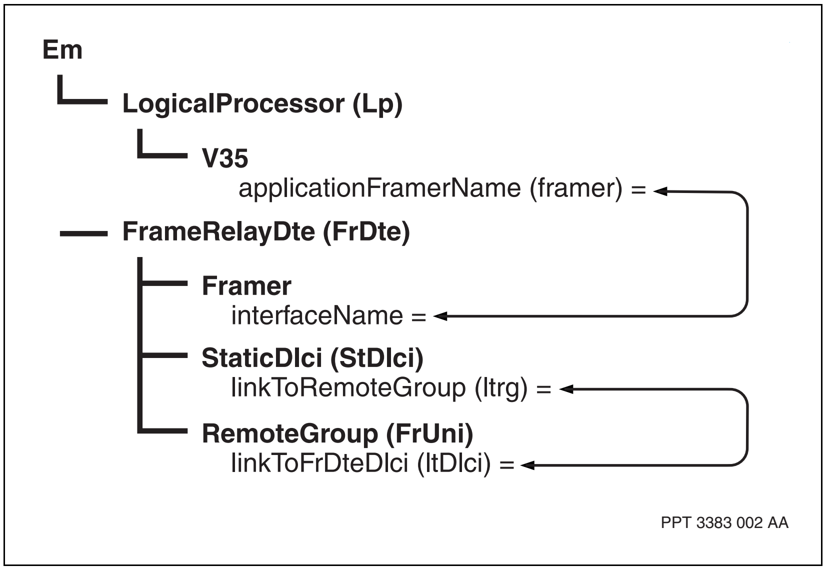
--End--

Variable definitions

Variable	Value
<a>	is the number of the FrDte instance.
<d>	is the number of the static DLCI instance.
<e>	is the number of the Lp instance.
<f>	is the number of the V35 instance.
<g>	is the number of the remote group instance.

Procedure job aidProcedure job aid

Frame relay DTE with Framer component hierarchy



Configuring frame relay DTE with vFramer

Configure frame relay data terminating device (DTE) with vFramer to establish a logical connection between a frame relay DTE interface and a frame relay UNI interface as an alternative to a physical or direct connection. Not using a physical connection frees a port on the card.

Procedure steps

Step **Action**

1 Delete the framer component.

```
delete FrDte/<a> Framer
```

2 Add a virtual framer component to the FrDte.

```
add FrDte/<a> VFramer
```

3 Delete the framer component to FrUni/b.

```
delete FrUni/<b> Framer
```

4 Add a virtual framer subcomponent to FrUni/b.

```
add FrUni/<b> VFramer
```

5 Link the FrDte virtual framer.

```
set FrUni/<b> VFramer otherVirtualFramer FrDte/<a>
VFramer
```

6 Link the FrUni virtual framer.

```
set FrDte/<a> VFramer otherVirtualFramer FrUni/<b>
VFramer
```

If the virtual framer is configured on the 4-port DS3Ch or 1-port STM-1Ch FPs, the pair of virtual framers must be on the same LP.

7 Link the FrDte application to the Lp.

```
set FrDte/<a> VFramer lp Lp/<e>
```

8 Add a static Dlci.

```
add FrDte/<a> StaticDlci/<d>
```

9 Link the static Dlci to a remote group.

```
set FrDte/<a> StaticDlci/<d> linkToRemoteGroup
FrDte/<a> Rg/<g>
```

10 Add DLCI and DirectCall subcomponents to both FrUni components.

```
add FrUni/<b> Dlci/<d>
add FrUni/<c> Dlci/<d>
```

When you add a *Dlci* component, Nortel Networks Multiservice Switch system automatically creates a *DirectCall (Dc)* subcomponent.

11 Set the *type* attribute of FrUni/<b> Dlci/<d> Dc.

```
set FrUni/<b> Dlci/<d> Dc type master
```

12 Set the *remoteDna* attribute of FrUni/<b> Dlci/<d> Dc.

```
set FrUni/<b> Dlci/<d> Dc rdna <dna of Fruni/<c>
```

13 Set the *remoteDlci* attribute of FrUni/<b> Dlci/<d> Dc.

```
set FrUni/<b> Dlci/<d> Dc rdlci <d>
```

14 Set the *type* attribute of FrUni/<c> Dlci/<d> Dc.

```
set FrUni/<c> Dlci/<d> Dc type slave
```

15 Set the *remoteDna* attribute of FrUni/<c> Dlci/<d> Dc.

```
set FrUni/<c> Dlci/<d> Dc rdna <dna of Fruni/<b>>
```

16 Set the *remoteDlci* attribute of FrUni/<c> Dlci/<d> Dc.

```
set FrUni/<c> Dlci/<d> Dc rdlci <d>
```

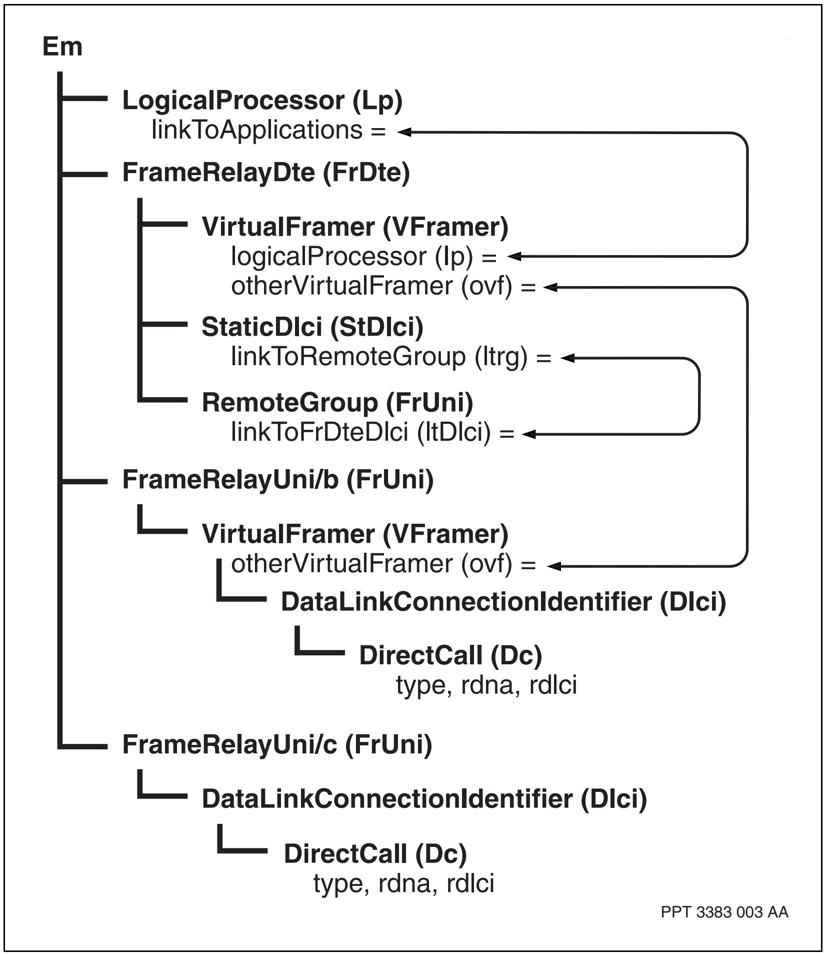
--End--

Variable definitions

Variable	Value
<a>	is the number of the FrDte instance.
<b>	is the number of the FrUni instance.
<c>	is the number of the FrUni instance.
<d>	is the number of the StaticDlci instance.
<e>	is the number of the Lp instance.
<g>	is the number of the remote group instance.

Procedure job aid

Frame relay DTE with vFramer component hierarchy



Configuring a direct connection for a frame relay DTE interface

Configure a direct connection between a frame relay data terminating device (DTE) interface and a frame relay UNI interface as an alternative to a physical or logical connection.

Procedure steps

Step	Action

1	Set the *LocalManagementInterface (LMI)* components on the FrDte and the FrUni to which it interfaces so that no LMI procedures are running.
	`set FrDte/<a> Lmi procedures none`
	`set FrUni/<b> Lmi procedures none`

2	Turn accounting data collection off for the DLCI of the FrUni that interfaces with the FrDte. `set FrUni/<b> Dlci/<d> Sp accounting off`
3	Add a direct connection component to the FrDte. `add FrDte/<a> Dconn`
4	Add a direct connection component to the customer-facing FrUni. `add FrUni/<c> Dconn`
5	Link the direct connection between the FrDte and the customer-facing FrUni. `set FrDte/<a> Dconn directFrUniConnection FrUni/<c> Dconn`

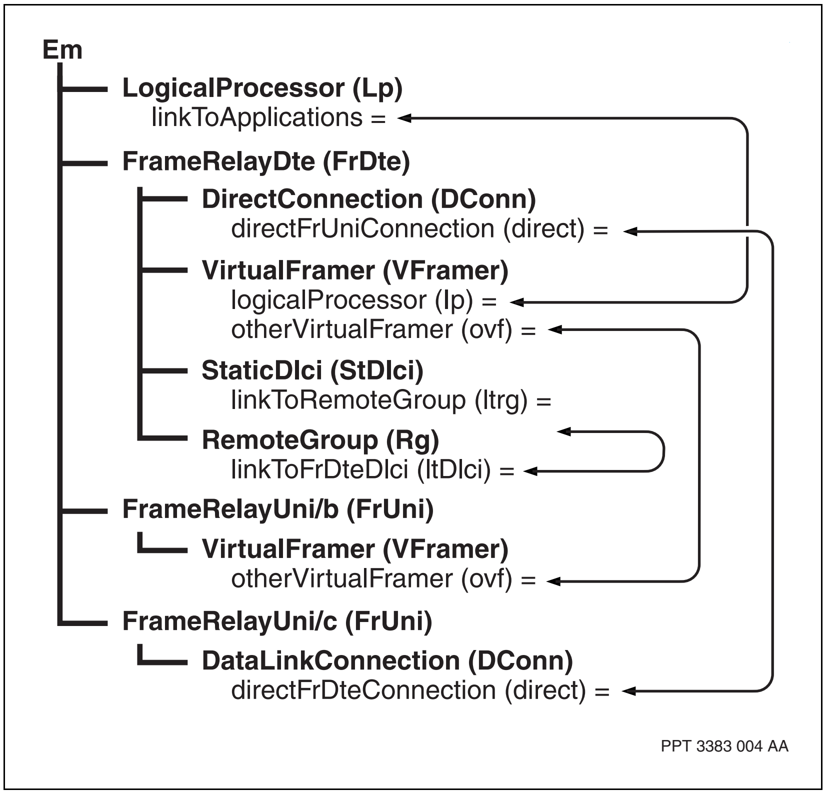
--End--

Variable definitions

Variable	Value
<a>	is the number of the FrDte instance.
<b>	is the number of the FrUni instance.
<c>	is the number of the FrUni instance.
<d>	is the number of the Dlci.

Procedure job aid

Direct connection for a frame relay DTE interface component hierarchy



Customizing static Dlci attributes

Customize the static provisionable data link connection identifier (*Dlci*) attributes to control the traffic characteristics on the link, such as size of traffic, information rate setting, dropped bandwidth and size of burst.

Dual-leaky (CIR and EIR) buckets are used to perform egress rate enforcement. The attribute *committedInformationRate* specifies the drain rate of the CIR bucket. And the attributes *committedBurst* and *excessBurst* specify the capacity of the CIR and EIR bucket respectively.

Procedure steps

Step	Action
1	Set the rate enforcement policy. `set FrDte/<a> StDlci/<b> rateEnforcement <enforcement>`
2	Define the average number of bits to be transferred per second over the *Dlci* component to the *Dce* component. `set FrDte/<a> StDlci/<b> committedInformationRate <rate>`
3	Define the committed burst size (in bits) to which the *Dlci* component wants to subscribe. `set FrDte/<a> StDlci/<b> committedBurst <burst>`
4	Define the excess bursts (in bits) to which the *Dlci* component wants to subscribe. `set FrDte/<a> StDlci/<b> excessBurst <excess>`
5	Specify the action taken when *committedBurst* size has been exceeded on the *Dlci* component but the *excessBurst* size has not. `set FrDte/<a> StDlci/<b> excessBurstAction <action>`

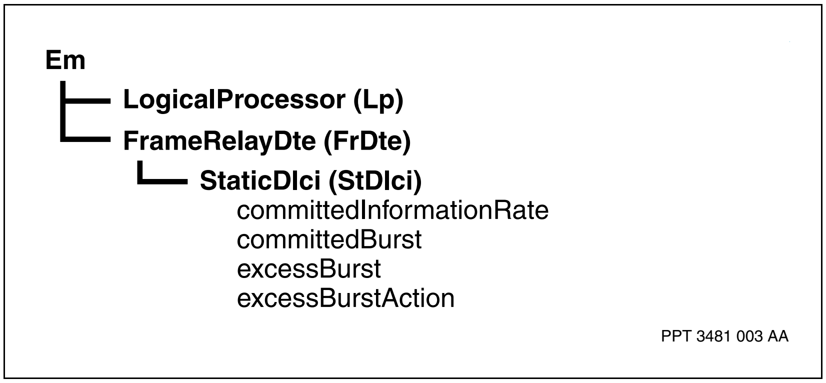
--End--

Variable definitions

Variable	Value
<a>	is the number of the FrDte instance.
<action>	is the action you take when *committedBurst* size has been exceeded on the *Dlci* component but the *excessBurst* size has not.
<b>	is the number of the static DLCI instance.
<burst>	is the value of the committed burst size, in bits, to which the *Dlci* component wants to subscribe.
<enforcement>	is the rate enforcement policy.
<rate>	is the average number of bits to be transferred per second over the *Dlci* to the *Dce*.

Procedure job aid

Static Dlci attributes component hierarchy



Customizing dynamic Dlci attributes

Customize the dynamic provisionable attributes of the *FrDte* data link connection identifier (*Dlci*) subcomponent to provide additional capabilities.

Procedure steps

Step	Action
1	Define the average number of bits to be transferred per second over the *Dlci* to the *Dce*. **`set FrDte/<a> DynDlciDef committedInformationRate <rate>`**
2	Set the rate enforcement policy on the *Dlci* component. **`set FrDte/<a> DynDlciDef rateEnforcement <enforcement>`**
3	Define the committed burst size to which the *Dlci* component wants to subscribe. **`set FrDte/<a> DynDlciDef committedBurst <burst>`**
4	Define the excess bursts (in bits) to which the *Dlci* component wants to subscribe. **`set FrDte/<a> DynDlciDef excessBurst <excess>`**
5	Specify the action taken when *committedBurst* size has been exceeded on the *Dlci* component but the *excessBurst* size has not. **`set FrDte/<a> DynDlciDef excessBurstAction <action>`**

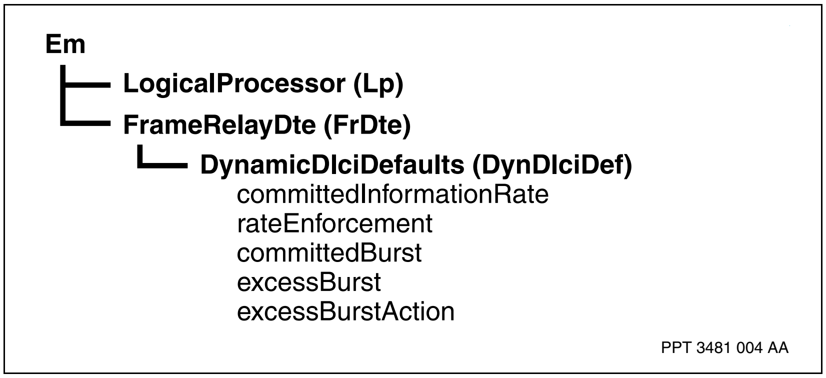
--End--

Variable definitions

Variable	Value
<a>	is the number of the FrDte instance.
<action>	is the action you take when *committedBurst* size has been exceeded on the *Dlci* component but the *excessBurst* size has not.
<burst>	is the value of the committed burst size, in bits, to which the *Dlci* component wants to subscribe.
<d>	is the number of the Dlci.
<enforcement>	is the rate enforcement policy.
<rate>	is the average number of bits to be transferred per second over the *Dlci* to the *Dce*.

Procedure job aid

Dynamic Dlci attributes component hierarchy



Customizing Hq attributes

Customize the hibernation queue (*Hq)* subcomponent under the *StDlci* component to provide the same type of service as the *Leq* component, except that the service is performed on a per DLCI basis. It is supported on SBIC-based FPs only. See Customizing Leq attributes (page 58) for more information.

Procedure steps

Step **Action**

1 Add the *Hq* subcomponent under the *StaticDlci* component.

```
add FrDte/<a> StDlci/<b> Hq
```

2 Define the maximum number of packets allowed in the link emission queue (*Leq*).

```
set FrDte/<a> StDlci/<b> Hq maxPackets <maxPackets>
```

3 Define the maximum amount of data allowed in the queue at any given time.

```
set FrDte/<a> StDlci/<b> Hq maxMsecData <maxMsecPackets>
```

4 Restrict the percentage of multicast packets allowed in the queue.

```
set FrDte/<a> StDlci/<b> Hq maxPercentMulticast
<maxPercentMulticast>
```

5 Ensure old packets on the queue are discarded if they are not transmitted.

```
set FrDte/<a> StDlci/<b> Hq timeToLive <timeToLive>
```

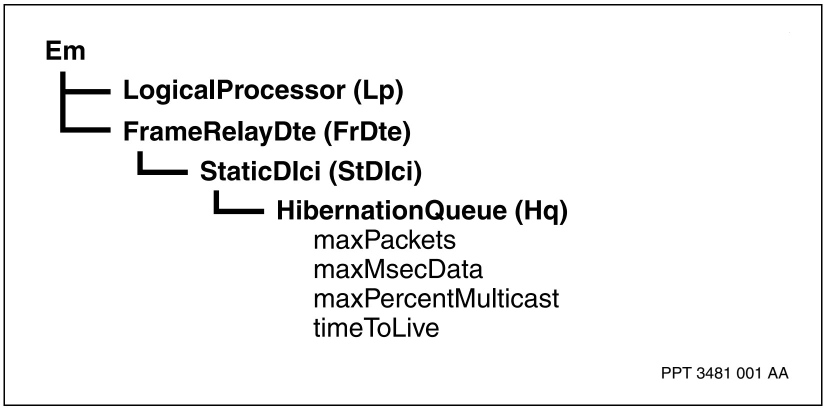
--End--

Variable definitions

Variable	Value
<a>	is the number of the FrDte instance.
<b>	is the number of the static DLCI instance.
<maxMsecPackets>	is the maximum amount of data, in milliseconds, allowed in the queue at any given time.
<maxPackets>	is the maximum number of packets allowed in the link emission queue (*Leq*).
<maxPercentMulticast>	is the percentage of multicast packets allowed in the queue.
<timeToLive>	is the time, in milliseconds, for packets to live in the queue before they are discarded, if not transmitted.

Procedure job aid

Hq attributes component hierarchy



Customizing Leq attributes

Add a link emission queue (*Leq*) subcomponent to the *FrDte* component to provide more elasticity (packet queuing) in the transmit data path, especially for slow frame relay data rates that tend to be overdriven by higher speed LANs. *Leq* also allows prioritization of traffic for certain applications, guaranteeing bandwidth for a particular traffic flow. The *Leq* subcomponent is supported only on Nortel Networks Multiservice Switch 7400 nodes with SBIC-based FPs.

Procedure steps

Step **Action**

1 Add the *Leq* subcomponent under *FrDte* component.

```
add FrDte/<a> Leq
```

2 Define the maximum number of packets allowed in the link emission queue (*Leq*).

```
set FrDte/<a> Leq maxPackets <maxPackets>
```

3 Define the maximum amount of data allowed in the queue at any given time.

```
set FrDte/<a> Leq maxMsecData <maxMsecPackets>
```

4 Restrict the percentage of multicast packets allowed in the queue.

```
set FrDte/<a> Leq maxPercentMulticast
<maxPercentMulticast>
```

5 Ensure old packets on the queue are discarded if they are not transmitted.

```
set FrDte/<a> Leq timeToLive <timeToLive>
```

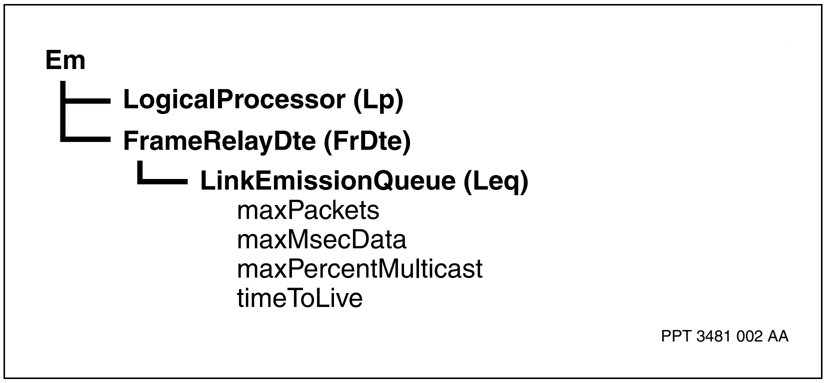
--End--

Variable definitions

Variable	Value
<a>	is the number of the FrDte instance.
<maxMsecPackets>	is the maximum amount of data, in milliseconds, allowed in the queue at any given time.
<maxPackets>	is the maximum number of packets allowed in the link emission queue (*Leq*).
<maxPercentMulticast>	is the percentage of multicast packets allowed in the queue.
<timeToLive>	is the time, in milliseconds, for packets to live in the queue before they are discarded, if not transmitted.

Procedure job aid

Leq attributes component hierarchy



VR IP-optimized DLCI access media configuration

Configure a VR IP-optimized data link connection identifier (DLCI) to enable Nortel Networks Multiservice Switch nodes to carry IP over frame relay using IP-optimized DLCI, which binds directly to a virtual protocol port.

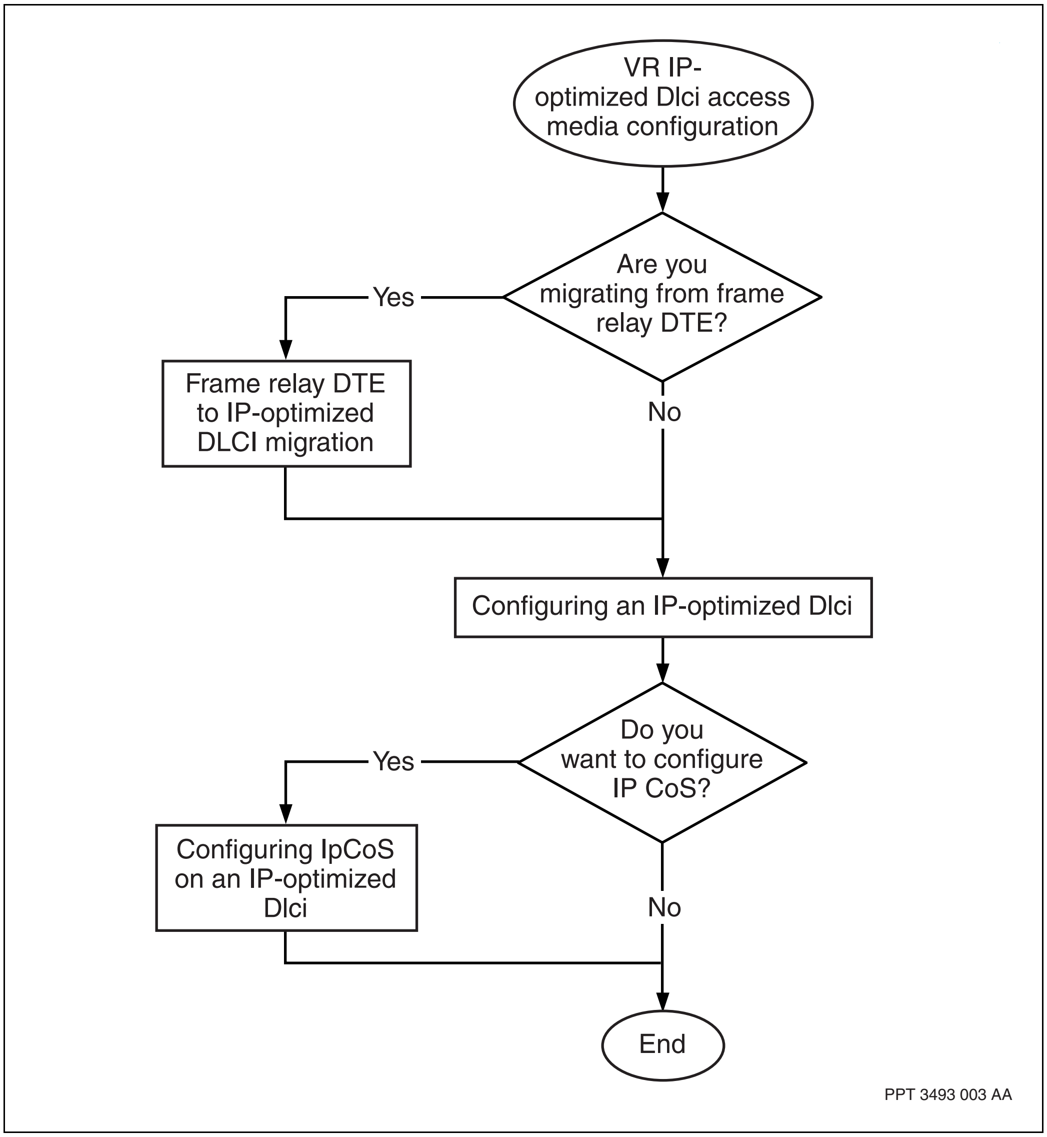
Prerequisites to VR IP-optimized DLCI access media configuration

- Configure the required frame relay interfaces. See NN10600-901 *Nortel Networks Multiservice Switch 7400/15000/20000 Frame Relay Configuration Management*.
- See NN10600-551 *Nortel Networks Multiservice Switch 7400/15000/20000 FP Configuration Reference* for information on which FPs support this service.
- Use the procedures in NN10600-270 *Nortel Networks Multiservice Switch 7400/15000/20000 Software Installation* to load any required features.

VR IP-optimized DLCI access media configuration procedures

This task flow shows you the sequence of procedures you perform to configure VR IP-optimized DLCI. To link to any procedure, go to VR IP-optimized DLCI access media configuration procedure navigation (page 61).

VR IP-optimized DLCI access media configuration procedures


VR IP-optimized Dlci access media configuration
Are you migrating from frame relay DTE?
Yes
Frame relay DTE to IP-optimized DLCI migration
No
Configuring an IP-optimized Dlci
Do you want to configure IP CoS?
Yes
Configuring IpCoS on an IP-optimized Dlci
No
End
PPT 3493 003 AA

VR IP-optimized DLCI access media configuration procedure navigation

- Frame relay DTE to IP-optimized DLCI migration (page 65)
- Configuring an IP-optimized DLCI (page 62)
- Configuring IpCos on an IP-optimized DLCI (page 64)

Configuring an IP-optimized DLCI

Configure an IP-optimized DLCI to create a direct frame relay access to a Nortel Networks Multiservice Switch network.

Prerequisites

- When you are using an IP-optimized DLCI at the edge of the network, verify that the attribute *FrUni Lmi side* is set to *network*. For more information on the *Lmi* component, see NN10600-900 *Nortel Networks Multiservice Switch 7400/15000/20000 Frame Relay Technology Fundamentals*.

Procedure steps

Step **Action**

1 Add the IpDlciGroup.

```
add IpDlciGroup/<n>
```

The subcomponent *IpDlciGroup Frc/1* is automatically created.

2 Add the DLCI and IP connection to the FrUni.

```
add FrUni/<o> Dlci/<p>
```

When you add a *Dlci* component, a *Dc* subcomponent is automatically created.The *Dc* and *IpConnection* subcomponents are mutually exclusive.

3 Delete the *Dc* subcomponent.

```
del FrUni/<o> Dlci/<p> Dc
```

4 Add the IP connection to the FrUni.

```
add FrUni/<o> Dlci/<p> IpConnection
```

5 Link the DLCI to its FrConnection.

```
set FrUni/<o> Dlci/<p> IpConnection linkToIpDlciGrp
IpDlciGroup/<n> Frc/<q>
```

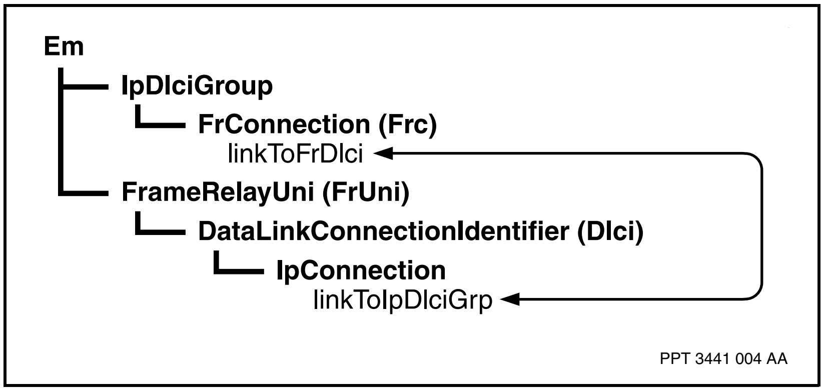
--End--

Variable definitions

Variable	Value
<n>	is the instance value of the interface between the virtual router protocol port and the IP-optimized DLCI.
<o>	is the instance value of the FrUni.
<p>	is the instance value of the DLCI.
<q>	is the instance value of the frame relay connection.

Procedure job aid

IP-optimized DLCI component hierarchy



Configuring IpCos on an IP-optimized DLCI

Configure IpCos on an IP-optimized DLCI to control delay variation when voice is carried across the same interface as data.

Procedure steps

Step	Action
1	Set the CoS index, which is used as the value for the emission priority (EP). `set IpDlciGroup/<n> Frc/<q> ipCoS <cos>`

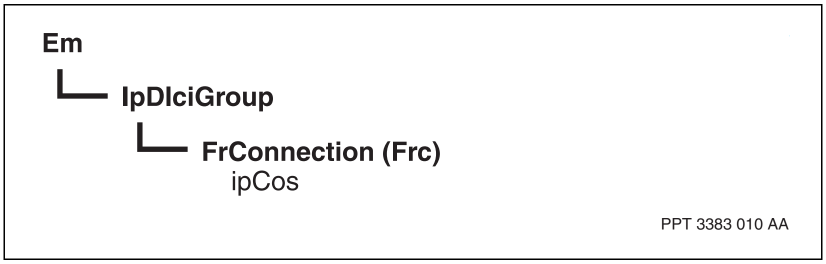
--End--

Variable definitions

Variable	Value
<cos>	is the CoS value associated with the DLCI.
<n>	is the instance value of the interface between the virtual router protocol port and the IP-optimized DLCI.
<q>	is the instance value of the frame relay connection.

Procedure job aid

IpCos on an IP-optimized DLCI component hierarchy



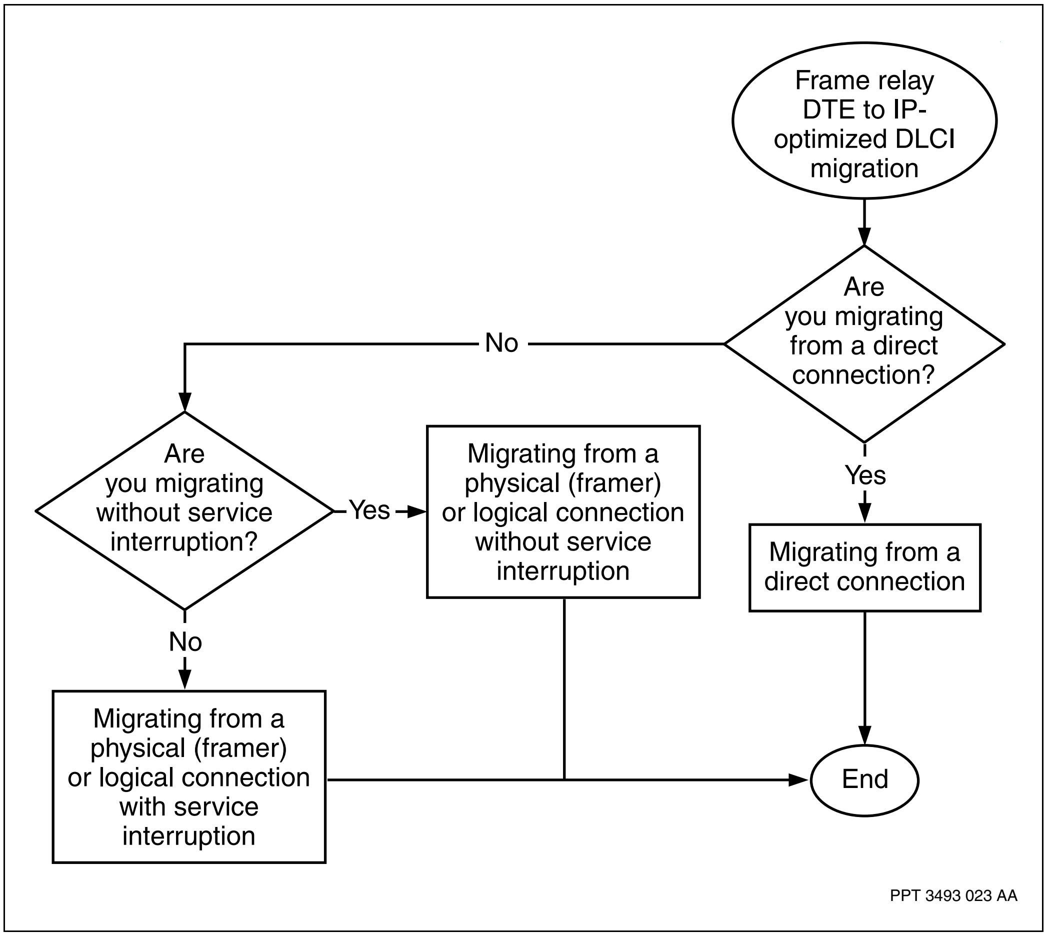
Frame relay DTE to IP-optimized DLCI migration

Migrate from FrDte-based to IP-optimized DLCI IP over frame relay to increase traffic throughput and simplify provisioning for IP over frame relay.

Frame relay DTE to IP-optimized DLCI migration tasks

This task flow shows you the sequence of procedures you perform to migrate frame relay DTE to IP-optimized DLCI. To link to any procedure, go to Frame relay DTE to IP-optimized DLCI migration procedure navigation (page 66).

Frame relay DTE to IP-optimized DLCI migration procedures


Frame relay DTE to IP-optimized DLCI migration
Are you migrating from a direct connection?
No
Yes
Migrating from a direct connection
Are you migrating without service interruption?
Yes
Migrating from a physical (framer) or logical connection without service interruption
No
Migrating from a physical (framer) or logical connection with service interruption
End
PPT 3493 023 AA

Frame relay DTE to IP-optimized DLCI migration procedure navigation

- Migrating from a physical (Framer) or logical connection with service interruption (page 67)
- Migrating from a physical (Framer) or logical connection without service interruption (page 69)
- Migrating from a direct connection (page 72)

Migrating from a physical (Framer) or logical connection with service interruption

Migrate from a physical (Framer) or logical connection with service interruption to increase traffic throughput and simplify provisioning for IP over frame relay by converting the existing DLCI to an IP-optimized DLCI.

Prerequisites

- See Configuring frame relay DTE with Framer (page 45) or Configuring frame relay DTE with vFramer (page 47) for information on the existing connection.

Procedure steps

Step **Action**

1 Remove the components that are no longer required.

```
del FrUni/<c> Dlci/<d> Dc
del FrUni/<b>
del FrDte/<a>
```

2 Add the IP DLCI group.

```
add IpDlciGroup/<e>
```

3 Link the IP DLCI group to the protocol port that was linked to the FrDTE.

```
set IpDlciGroup/<e> linkToProtocolPort Vr/<f> Pp/<g>
```

4 Add the IP connection.

```
add FrUni/<c> Dlci/<d> IpConnection
```

5 Link the DLCI to its FrConnection.

```
set FrUni/<c> Dlci/<d> IpConnection linkTolpDlciGrp
IpDlciGroup/<e> Frc/<h>
```

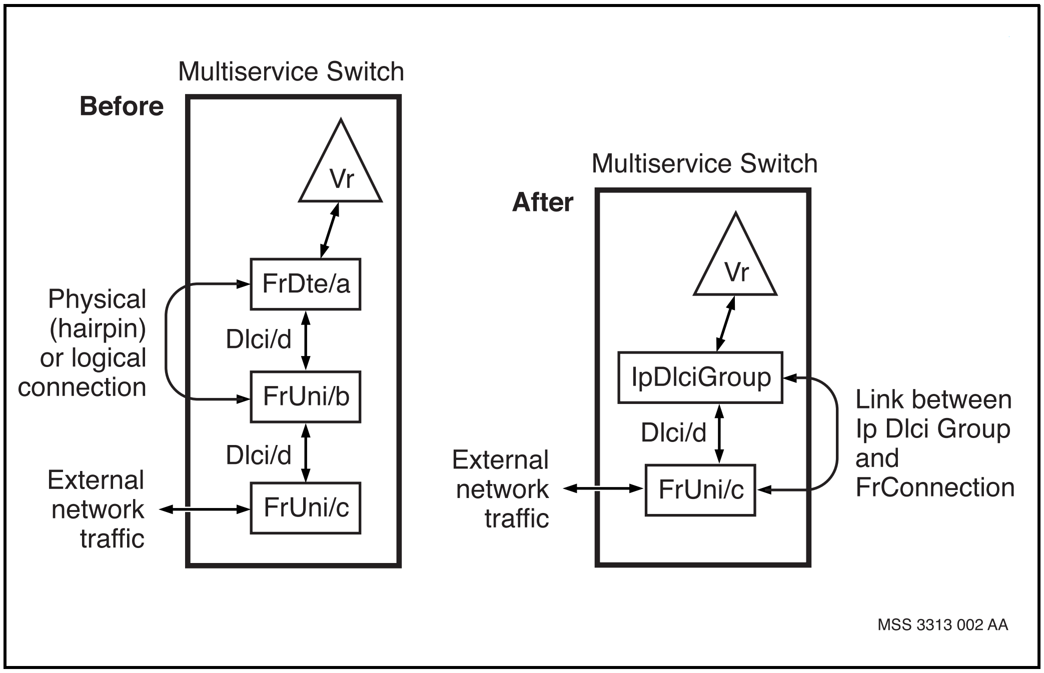
--End--

Variable definitions

Variable	Value
<a>	is the instance value of the FrDTE.
<b>	is the instance value of the FrUni between the FrDTE and the customer-facing FrUni.
<c>	is the instance value of the customer-facing FrUni.
<d>	is the instance value of the DLCI.
<e>	is the instance value of the interface between the virtual router protocol port and the IP-optimized DLCI.
<f>	is the name of the virtual router.
<g>	is the name of the protocol port.
<h>	is the instance value of the frame relay connection.

Procedure job aid

Migration from a physical (Framer) or logical connection with service interruption



Migrating from a physical (Framer) or logical connection without service interruption

Migrate from a physical (Framer) or logical connection without service interruption to increase traffic throughput and simplify provisioning for IP over frame relay.

Prerequisites

- See Configuring frame relay DTE with Framer (page 45) or Configuring frame relay DTE with vFramer (page 47) for information on the existing connection.

Procedure steps

Step **Action**

1 Add the IP DLCI group.

```
add IpDlciGroup/<e>
```

2 Add the DLCI to the FrUni.

```
add FrUni/<c> Dlci/<new>
del FrUni/<c> Dlci/<new> Dc
```

3 Add the IP connection to the FrUni.

```
add FrUni/<c> Dlci/<new> IpConnection
```

4 Link the DLCI to its FrConnection.

```
set FrUni/<c> Dlci/<new> IpConnection linkToIpDlciGrp
IpDlciGroup/<e> Frc/<h>
```

5 Link the IP DLCI group to the protocol port.

```
set IpDlciGroup/<e> linkToProtocolPort Vr/<f> Pp/<g>
```

6 Ensure that you have set up a logical interface and network mask. See Enabling IpPort on a protocol port (page 101).

7 Lock the original DLCI to force traffic on to the IP-optimized DLCI.

```
lock FrUni/<c> Dlci/<old>
```

8 When you are sure the IP-optimized DLCI is working properly, remove the components that are no longer required.

```
del FrUni/<c> Dlci/<old>
del FrUni/<b>
del FrDte/<a>
del Vr/<f> Pp/<i>
```

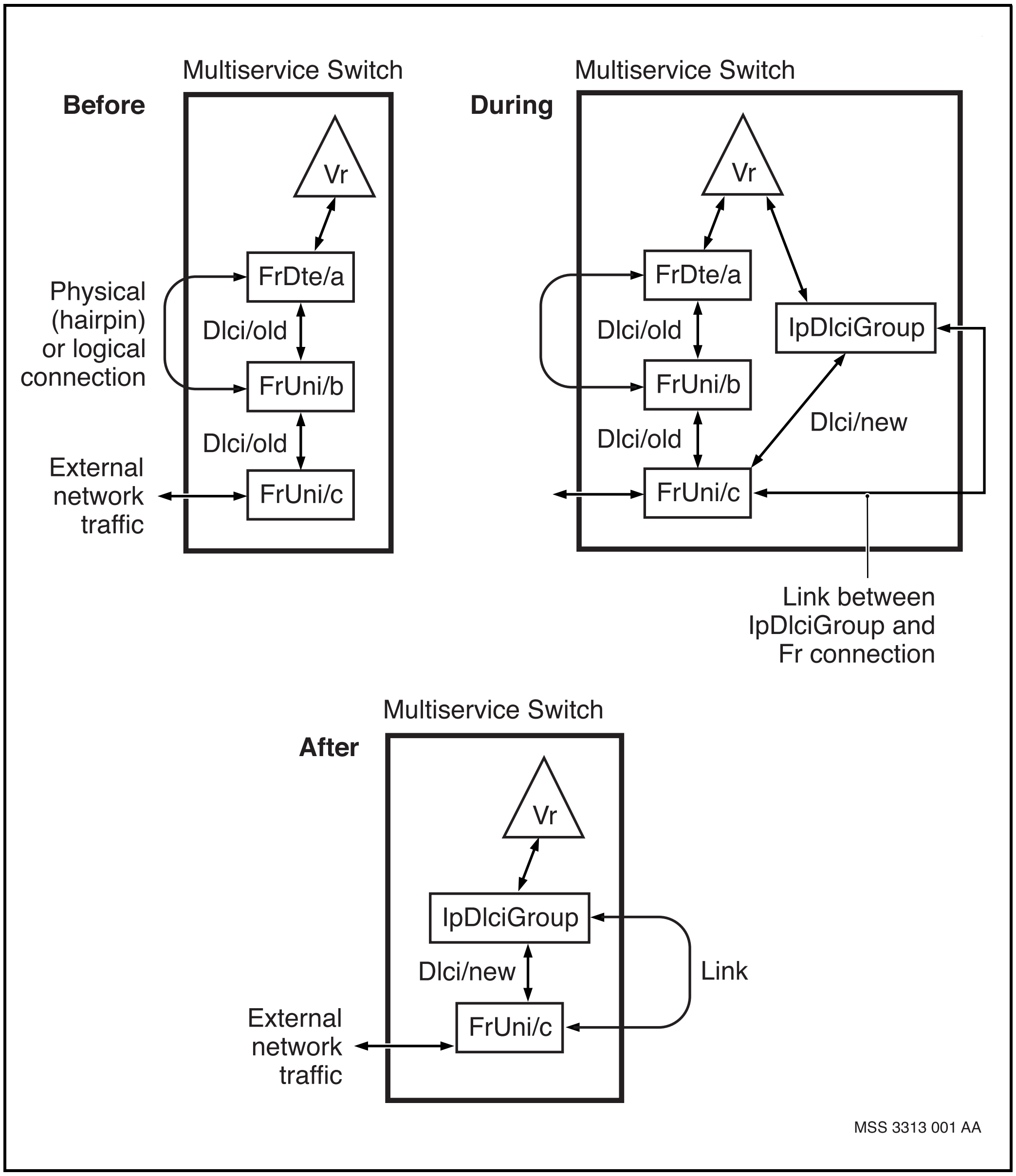
--End--

Variable definitions

Variable	Value
<a>	is the instance value of the FrDTE.
<b>	is the instance value of the FrUni between the FrDTE and the customer-facing FrUni.
<c>	is the instance value of the customer-facing FrUni.
<e>	is the instance value of the interface between the virtual router protocol port and the IP-optimized DLCI.
<f>	is the name of the virtual router.
<g>	is the name of the protocol port for the IP-optimized DLCI.
<h>	is the instance value of the frame relay connection.
<i>	is the name of the protocol port for the frame relay DTE.
<new>	is the instance value of the IP-optimzed DLCI.
<old>	is the instance value of the old DLCI.

Procedure job aid

Migration from a physical (Framer) or logical connection without service interruption



Migrating from a direct connection

Migrate from a direct connection to increase traffic throughput and simplify provisioning for IP over frame relay by converting the existing DLCI to an IP-optimized DLCI.

Prerequisites

- See Configuring frame relay DTE with Framer (page 45) for information on the existing connection.

Procedure steps

Step **Action**

1 Remove the components that are no longer required.

```
del FrUni/<c> Dconn
del FrUni/<c> Dlci/<d> Dc
del FrUni/<b>
del FrDte/<a>
```

2 Add the IP DLCI group.

```
add IpDlciGroup/<e>
```

3 Link the IP DLCI group to the protocol port that was linked to the FrDTE.

```
set IpDlciGroup/<e> linkToProtocolPort Vr/<f> Pp/<g>
```

4 Add the IP connection.

```
add FrUni/<c> Dlci/<d> IpConnection
```

5 Link the DLCI to its FrConnection.

```
set FrUni/<c> Dlci/<d> IpConnection linkTolpDlciGrp
IpDlciGroup/<e> Frc/<h>
```

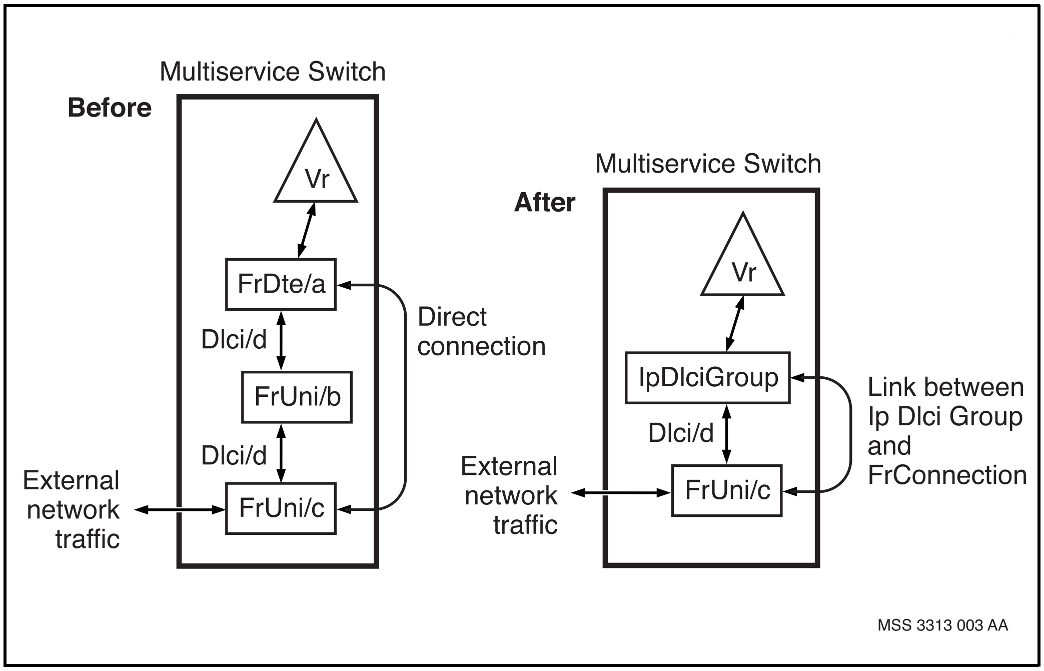
--End--

Variable definitions

Variable	Value
<a>	is the instance value of the FrDTE.
<b>	is the instance value of the FrUni between the FrDTE and the customer-facing FrUni.
<c>	is the instance value of the customer-facing FrUni.
<d>	is the instance value of the DLCI.
<e>	is the instance value of the interface between the virtual router protocol port and the IP-optimized DLCI.
<f>	is the name of the virtual router.
<g>	is the name of the protocol port.
<h>	is the instance value of the frame relay connection.

Procedure job aid

Migration from a direct connection



VR Ethernet access media configuration

Configure VR Ethernet access media to provide an Ethernet connection between a Nortel Networks Multiservice Switch node and the IP network.

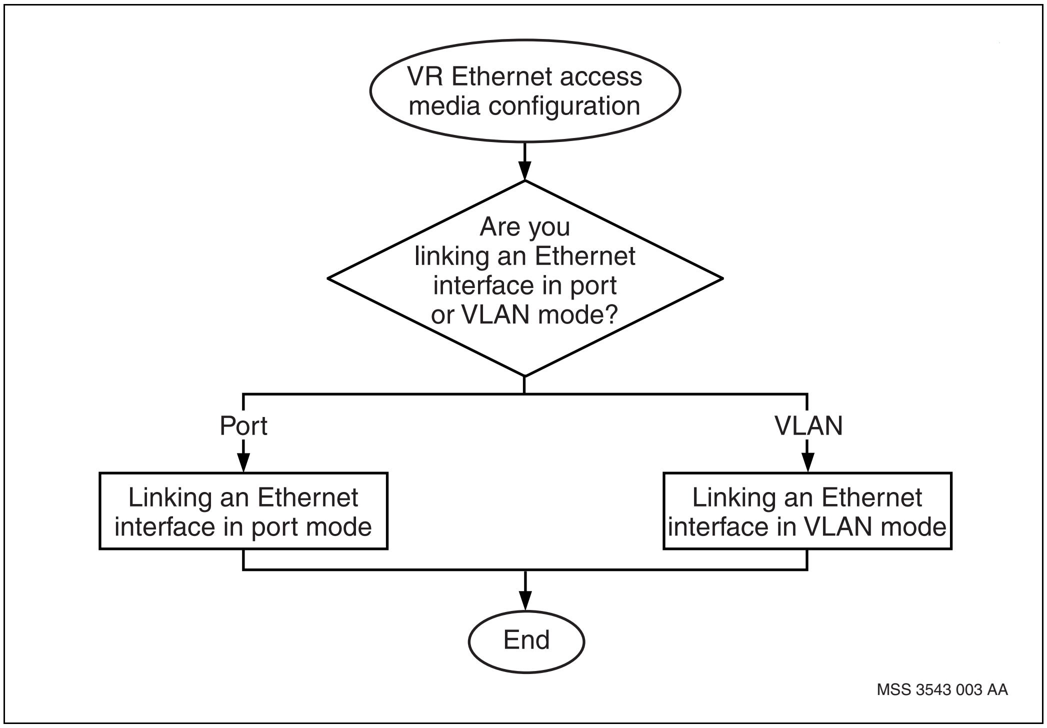
Prerequisites to VR Ethernet access media configuration

- You need to load feature *atmMpe* and *atmMpeSpvc* on the feature list of the 4-port 10/100BaseT Ethernet function processor (FP), 4-port gigabit Ethernet FP, or 8-port 10/100BaseT Ethernet FP in order to forward traffic to ATM MPE media.
- See NN10600-550 *Nortel Networks Multiservice Switch 7400/15000/ 20000 Common Configuration Procedures* for procedures on how to load any required features.
- See NN10600-551 *Nortel Networks Multiservice Switch 7400/15000/ 20000 FP Configuration Reference* for information on which FPs support this service.
- See NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/ 20000 IP Technology Fundamentals* for supporting information, such as application and feature names for IP on Nortel Networks Multiservice Switch nodes.

VR Ethernet access media configuration procedures

This task flow shows you the sequence of procedures to perform in order to configure Ethernet as an access media for IP in a VIPR configuration. To link to any procedure, go to VR Ethernet access media procedure navigation (page 75).

VR Ethernet access media configuration procedures


VR Ethernet access media configuration
Are you linking an Ethernet interface in port or VLAN mode?
Port
VLAN
Linking an Ethernet interface in port mode
Linking an Ethernet interface in VLAN mode
End
MSS 3543 003 AA

VR Ethernet access media procedure navigation

- Linking an Ethernet interface in port mode (page 76)
- Linking an Ethernet interface in VLAN mode (page 77)

Linking an Ethernet interface in port mode

Link an Ethernet interface in port mode to setup port access to an IP service.

Procedure steps

Step	Action
1	Link the *LanApplication* component to the *ProtocolPort* component. **`set La/<x> linkToProtocolPort Vr/<vr_value> Pp/<Pp_value>`**

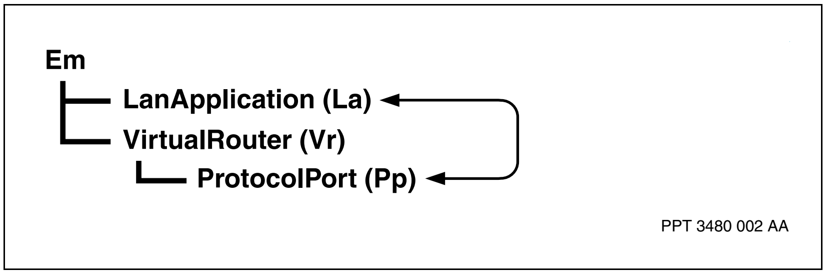
--End--

Variable definitions

Variable	Value
<Pp_value>	is the name of the protocol port.
<vr_value>	is the name of the Vr.
<x>	is the number of the *LanApplication* instance.

Procedure job aid

Ethernet interface in port mode component hierarchy



Linking an Ethernet interface in VLAN mode

Link an Ethernet interface in VLAN mode to provide Ethernet VLAN access to the VIPR solution.

Attention: Ethernet interfaces in VLAN mode are supported on the 4-port 10/100 BaseT Ethernet, 8-port 10/100 BaseT Ethernet, and 4-port gigabit Ethernet function processors (FPs) only.

Procedure steps

Step	Action
1	Link the *Vlan* component to the *LanApplication* component. **`set La/<x> vlan/<y> linkToProtocolPort Vr/<vr_value> Pp/<Pp_value>`**

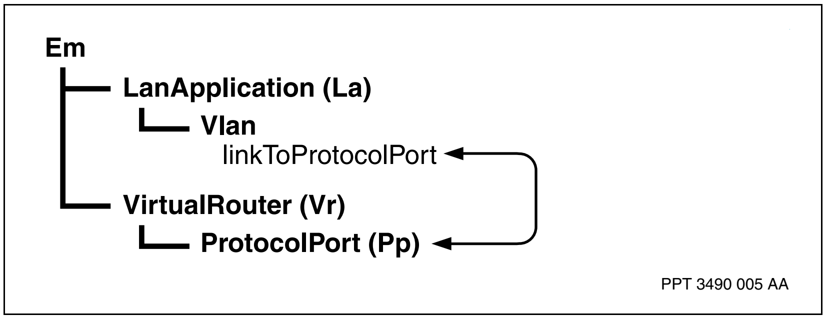
--End--

Variable definitions

Variable	Value
<Pp_value>	is the name of the protocol port.
<vr_value>	is the name of the VR.
<x>	is the number of the LanApplication instance.
<y>	is the number of the vlan instance.

Procedure job aid

Ethernet interface to the VIPR service component hierarchy



Virtual router configuration

Configure the virtual router on Nortel Networks Multiservice Switch nodes to provide a mechanism to segregate router traffic and maintenance activities based on logical separation of ports into separate VRs.

Attention: The 2-port 100baseT FP supports up to two VRs, one per port.

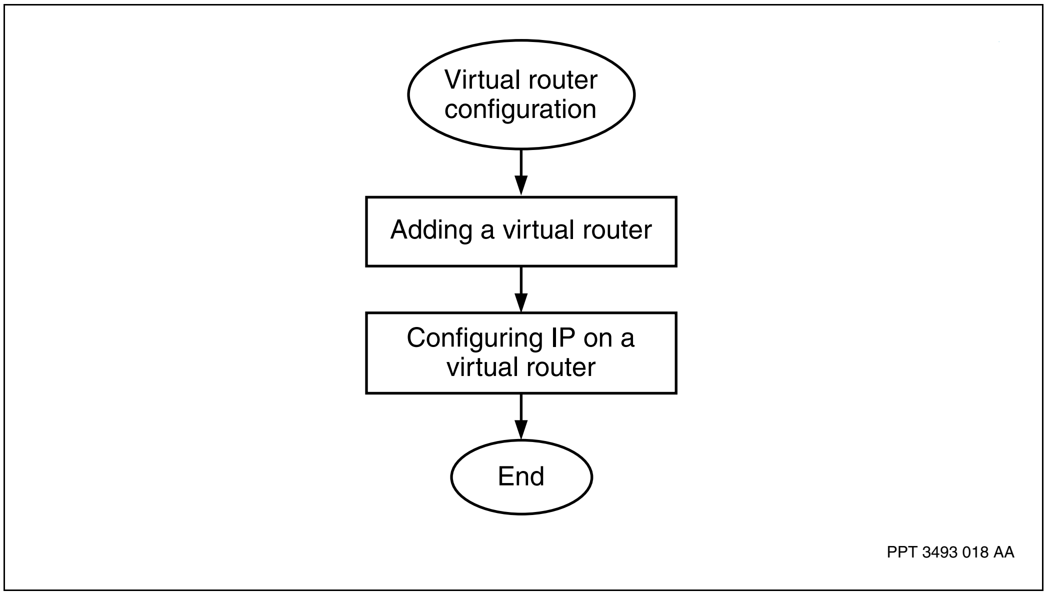
Prerequisites to virtual router configuration

- Download all required software applications. See NN10600-270 *Nortel Networks Multiservice Switch 7400/15000/20000 Software Installation*.
- For information on software applications and their associated feature names for IP, see NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Technology Fundamentals*.

Virtual router configuration procedures

This task flow shows you the sequence of procedures you perform to configure the virtual router on Nortel Networks Multiservice Switch nodes. To link to any procedure, go to Virtual router configuration procedure navigation (page 79).

Virtual router configuration procedures


Virtual router configuration
Adding a virtual router
Configuring IP on a virtual router
End
PPT 3493 018 AA

Virtual router configuration procedure navigation

- Adding a virtual router (page 80)
- Configuring IP on a virtual router (page 82)

Adding a virtual router

Add a virtual router (VR) on Nortel Networks Multiservice Switch nodes to emulate a physical router.

Prerequisites

> **CAUTION**
>
> **Moving, deleting, or locking the management VR**
>
> The first VR that you create on a node is, by default, the management VR. Once you have activated your provisioning (edit) view, you cannot designate any other VR on the node as the management VR. Deleting or locking the management VR once you have activated your provisioning (edit) view results in loss of connectivity to the node.

- In cases where a Multiservice Switch node supports multiple VRs, choose names that easily identify each VR.

Procedure steps

Step	Action
1	Add a *Vr* component. **`add Vr/<vr_name>`**
2	Specify where the virtual router resides. **`set Vr/<vr_name> vrp lp/<vr_lp>`**

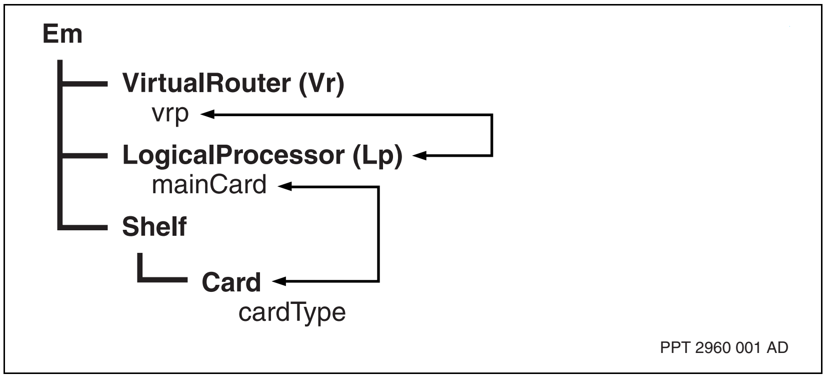
--End--

Variable definitions

Variable	Value
<vr_name>	is the name of the virtual router.
<vr_lp>	is the instance value of the logical processor that is linked to the card on the node where the virtual router resides.

Procedure job aid

Virtual router component hierarchy



Configuring IP on a virtual router

Configure IP on a virtual router to give IP connectivity to a Nortel Networks Multiservice Switch node.

Procedure steps

Step **Action**

1 Add an *Ip* component as a subcomponent of the *Vr* component.

```
add Vr/<vr_name> Ip
```

2 Optionally, change the default cache table size by provisioning a cache table size for a single LP.

```
set Vr/<vr_name> Ip cacheTableSize <lp_id> <cache_size>
```

3 Optionally, set the source route attribute.

```
set Vr/<vr_name> Ip sourceRoute <sr_option>
```

4 Set the *dscpRoutingSource* attribute to configure the DSCP value for locally generated BGP, RIP, and OSPF packets.

```
set Vr/<vr_name> Ip dscpr <dscp_value>
```

5 Set the *dscpGeneralSource* attribute to configure the DSCP value for any locally generated packets other than BGP, RIP, or OSPF.

```
set Vr/<vr_name> Ip dscpg <dscp_value>
```

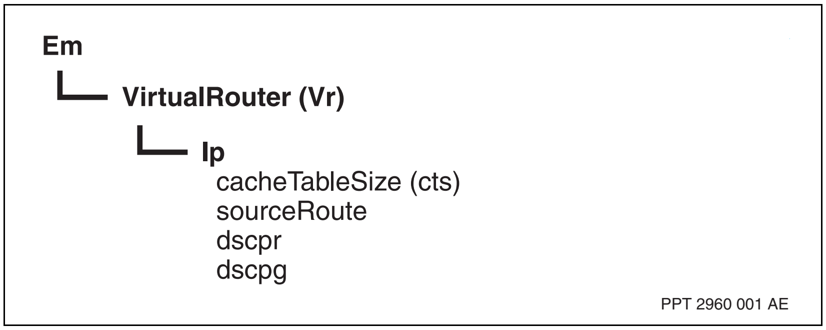
--End--

Variable definitions

Variable	Value
<cache_size>	is the value of the cache size you want to provision for a particular LP. If you choose not to use the default values, you must make cache table size adjustments in multiples of 100. For more information on managing the cache table, see the section on cache table size in NN10600-800 *Nortel Networks Multiservice Switch 7400/ 15000/20000 IP Technology Fundamentals*.
<lp_id>	is the instance value assigned in the IP subcomponent to a particular LP.
<sr_option>	is the provisionable attribute *sourceRoute* under the *Ip* component allows you to enable or disable the processing of input datagrams that have a source route IP option on a VR basis. The default value is disabled. For more information, see NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/ 20000 IP Technology Fundamentals*.
<vr_name>	is the name of the virtual router.
<dscp_value>	is the specific DSCP value you wish to assign for that packet type (Default = 0(df) for *dscpg* and 48 (cs6) for *dscpr*).

Procedure job aid

IP on a virtual router component hierarchy



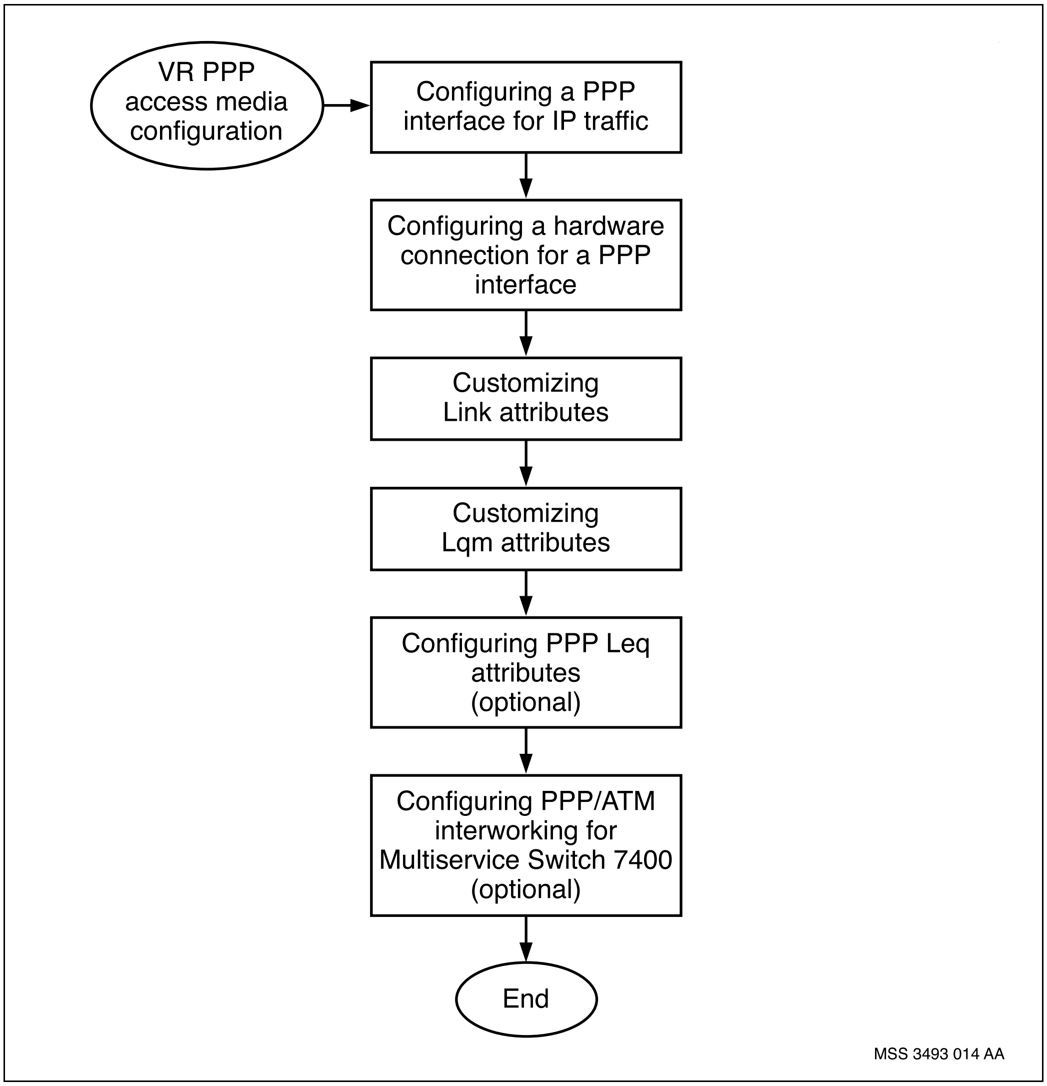
VR PPP access media configuration

Configure VR point-to-point protocol (PPP) access media to enable Nortel Networks Multiservice Switch nodes to carry IP over PPP.

VR PPP access media configuration procedures

This task flow shows you the sequence of procedures you perform to configure point-to-point protocol (PPP). To link to any procedure, go to VR PPP access media configuration procedure navigation (page 85).

VR PPP access media configuration procedures


VR PPP access media configuration
Configuring a PPP interface for IP traffic
Configuring a hardware connection for a PPP interface
Customizing Link attributes
Customizing Lqm attributes
Configuring PPP Leq attributes (optional)
Configuring PPP/ATM interworking for Multiservice Switch 7400 (optional)
End
MSS 3493 014 AA

VR PPP access media configuration procedure navigation

- Configuring a PPP interface for IP traffic (page 86)
- Configuring a hardware connection for a PPP interface (page 87)
- Customizing Link attributes (page 89)
- Customizing Lqm attributes (page 91)
- Customizing PPP Leq attributes (page 92)
- Configuring PPP/ATM interworking for Multiservice Switch 7400 (page 93)

Configuring a PPP interface for IP traffic

Configure a PPP interface for IP traffic to provide a PPP connection between a Nortel Networks Multiservice Switch node and the IP network.

Procedure steps

Step	Action
1	Add a PPP link level protocol interface application to the root component. **`add Ppp/<n>`**
2	If you are linking PPP to a specific protocol, link PPP to Vr Pp by updating the attribute *linkToProtocolPort*.
3	If you are linking PPP to a specific application and did not link PPP to a specific protocol, link Ppp to Rtr Vrf If PppApplication by updating the attribute *linkToApplication*. See NN10600-582 *Nortel Networks Multiservice Switch 7400/15000/20000 VPN Configuration Management*.

--End--

Variable definitions

Variable	Value
<n>	is the number of the PPP instance.

Configuring a hardware connection for a PPP interface

Configure a hardware connection for a PPP interface to route IP packets directly to the link queue of the card where PPP resides without software intervention on the FP. This dramatically improves full duplex packet switching performance over PPP.

Attention: This example uses a Nortel Networks Multiservice Switch 7400 node and MSA32 E1 card.

Procedure steps

Step	Action
1	Add a logical processor (LP). `add Lp/<a>`
2	Add a port to the logical processor. In this example, an E1 port is added. The system automatically adds channel 0 when you add a port. `add Lp/<a> E1/<b>`
3	Link the PPP service to the hardware component. `set Ppp/<n> Framer interfaceName Lp/<a> E1/<b> Chan/0`
4	Configure the timeslots of the hardware connection. `set Lp/<a> E1/<b> Chan/0 timeslots 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31`

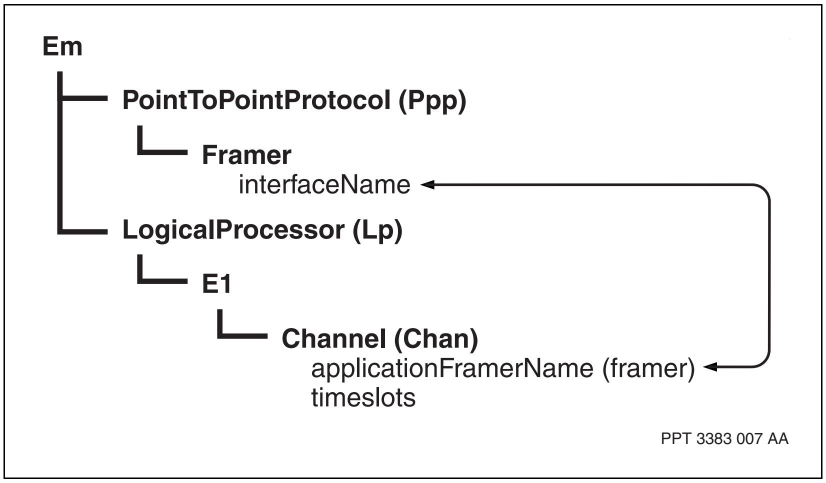
--End--

Variable definitions

Variable	Value
<a>	is the instance number of the Lp.
<b>	is the instance number of the port.
<n>	is the instance number of the PPP.

Procedure job aid

Hardware connection for a PPP interface component hierarchy



Customizing Link attributes

Customize *Link* attributes to provide additional capabilities.

Procedure steps

Step **Action**

1 Change the maximum receive unit (MRU) negotiated with the peer PPP application.

```
set Ppp/<n> Link configInitialMru <mru>
```

2 Use *magic number* negotiation to detect looped back *link* connections.

```
set Ppp/<n> Link configMagicNumber <cmn>
```

3 Set the *link* continuity monitor (LCM).

```
set Ppp/<n> Link continuityMonitor <monitor>
```

4 Customize the elapsed time after which PPP attempts to connect with its peer.

```
set Ppp/<n> Link restartTimer <resTime>
```

5 Customize the number of *Lcp* configuration request retries before entering the stopped state.

```
set Ppp/<n> Link configureRequestTries <tries>
```

6 Customize the number of *Lcp* negative acknowledgements (Naks) that the PPP application sends out when it receives *Lcp* configure requests that are not expected.

```
set Ppp/<n> Link negativeAckTries <negTries>
```

7 Customize the number of *Lcp* terminate request packets a PPP application sends out when it terminates a PPP connection.

```
set Ppp/<n> Link terminateRequestTries <terTries>
```

8 Set the quality threshold to a percentage of good packets required to maintain the *link* connection, when *Lqm* is enabled.

```
set Ppp/1 Link qualityThreshold <quality>
```

9 Customize the quality window during which *Lqm* examines link quality reporting (LQR) samples to determine what the quality has been over the history of the PPP connection.

```
set Ppp/<n> Link qualityWindow <window>
```

--End--

Variable definitions

Variable	Value
<cmn>	is the magic number negotiation variable, which detects looped back *link* connections. Value is enabled or disabled.
<mru>	is the size in bytes of the MRU.
<monitor>	is the attribute that turn the Link Continuity Monitor on or off. Value is enabled or disabled.
<n>	is the number of the PPP instance.
<negTries>	is the number of negative acknowledgements the PPP application sends out before entering the stopped state.
<quality>	is the number of good packets required to maintain the link connection before the link is disabled.
<resTime>	is the elapsed time after which PPP attempts to connect with its peer.
<terTries>	is the number of terminate request packets sent out.
<tries>	is the number of LCP configuration request retries before entering the stopped state.
<window>	is the time in seconds of the quality window.

Customizing Lqm attributes

Customize *Lqm* (link quality monitor) attributes to provide additional capabilities.

Procedure steps

Step	Action
1	Change the status of the *Lqm* component. **`set Ppp/<n> Lqm configStatus <status>`**
2	Customize the *Lqm* reporting period used by the PPP connection. **`set Ppp/<n> Lqm configPeriod <period>`**

--End--

Variable definitions

Variable	Value
<n>	is the number of the PPP instance.
<period>	is the period time in centiseconds.
<status>	is the *Lqm* component status. Value is enabled or disabled.

Customizing PPP Leq attributes

Customize *Ppp Leq* (link emission queue) attributes to provide additional capabilities. The *PPP Leq* subcomponent is supported only on Nortel Networks Multiservice Switch 7400 nodes with SBIC-based FPs.

Procedure steps

Step	Action
1	Add the *Leq* subcomponent. **`add Ppp/<n> Leq`**
2	Define the maximum number of packets allowed in the link emission queue (*Leq*). **`set Ppp/<a> Leq maxPackets <maxPackets>`**
3	Define the maximum amount of data allowed in the queue at any given time. **`set Ppp/<n> Leq maxMsecData <maxMsecPackets>`**
4	Restrict the percentage of multicast packets allowed in the queue. **`set Ppp/<n> Leq maxPercentMulticast <maxPercentMulticast>`**
5	Ensure old packets on the queue are discarded if they are not transmitted. **`set Ppp/<n> Leq timeToLive <timeToLive>`**

--End--

Variable definitions

Variable	Value
<n>	is the number of the PPP instance.
<maxMsecPackets>	is the maximum amount of data, in milliseconds, allowed in the queue at any given time.
<maxPackets>	is the maximum number of packets allowed in the link emission queue (*Leq*).
<maxPercentMulticast>	is the percentage of multicast packets allowed in the queue.
<timeToLive>	is the time, in milliseconds, for packets to live in the queue before they are discarded, if not transmitted.

Configuring PPP/ATM interworking for Multiservice Switch 7400

Configure PPP/ATM interworking on Nortel Networks Multiservice Switch 7400 32-port MSA function processors to enable IP transport between PPP-attached user devices and ATM-attached routers.

Prerequisites

- Load the PppIwf feature onto the ATM FP, which also needs to be configured with PNNI.
- All configuration steps, including the configuration of PNNI or IISP, must be configured in the ATM network to enable the establishment of an SPVC connection. See NN10600-710 *Nortel Networks Multiservice Switch 7400/ 15000/20000 ATM Configuration Management*.
- Configure the required PPP interfaces and connections on the CPE router, connecting to the E1 or DS1 port configured with the *PppIwf* component.
- Connect the core router to a port where the SPVC terminates and with the atmMpe attribute matching the encapType attribute of the *PppIwf* component.
- Configure the static ARP entry on the core router with an IP address that matches the PPP IP address. See Configuring VR static ARP (page 184).

Procedure steps

Step **Action**

1 Update the application version list (AVL) with the selected versions of the software.

```
set Sw avl wanDte_<version>
```

When you update the AVL, you must verify the configuration twice: once before you can activate it, and again when the node is running the new software. For more information about changing the AVL, see NN10600-270 *Nortel Networks Multiservice Switch 7400/15000/20000 Software Installation*.

2 Add the logical processor types for the MSA 32 FP. For illustrative purposes only, the name MSA32PIWF is used.

```
add Sw Lpt/<MSA32PIWF>
```

3 Add the MSA32 FP.

```
add Lp/1
set Lp/1 main shelf card/1
set Shelf card/1 cardType <MSA32cardType>
```

4 Set the logical processor types for the MSA 32 FP. For illustrative purposes only, the name MSA32PIWF is used.

```
set Sw Lpt/MSA32PIWF feature pppIwf
```

5 Add a DS1 or E1 port and provision the timeslots for its channels component.

For DS1 function processors:

```
add Lp/<Lp1> DS1/<Ds1>
set Lp/<Lp1> Ds1/<Ds1> Chan/<DS_Chan1> timeslots 1 2 3 4
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
```

For E1 function processors:

```
add Lp/<Lp1> E1/<E1>
set Lp/<Lp1> E1/<E1> Chan/<E1_Chan1> timeslots 1 2 3 4 5
6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31
```

6 Add the *PppIwf* component.

```
add PppIwf/1100
```

7 Link the *Framer* component to the physical TDM port.

```
set PppIwf/1100 int lp/1 ds1/0 chan/0
```

8 Add the *AtmAp* component.

```
add PppIwf/1100 atmap
```

9 Set the *AtmAp addressToCall* attribute to the remote ATM interface.

```
set PppIwf/1100 atmap addresstocall
470000000000000000000000010020480D000B00
```

10 Set the *AtmAp* component called VpiVci to that of the connection required to be set up on the remote ATM interface.

```
set PppIwf/1100 atmap calledvpivci 0.200
```

11 Optionally, set the *atmServiceCategory* attribute for the SPVC:

```
set PppIwf/1100 atm tm atmServiceCategory
<atmServiceCategory>
```

12 If you are using UPC in your ATM PNNI network, you must set the *peakCellRate* attribute:

```
set PppIwf/1100 atm tm peakCellRate <pcrValue>
```

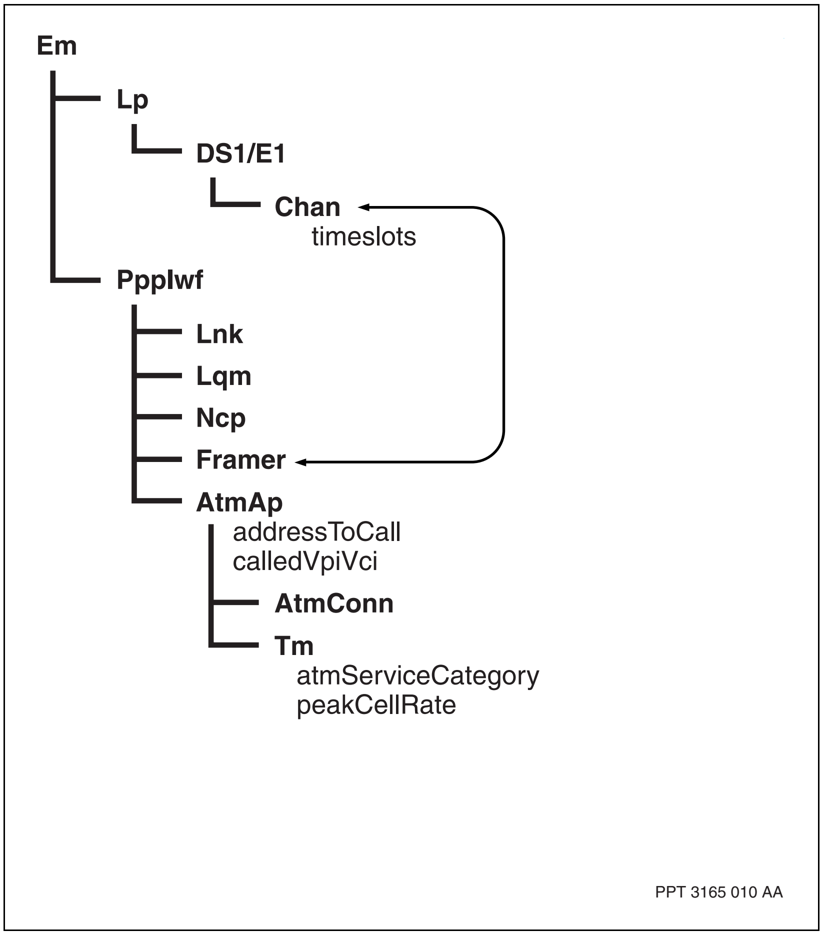
--End--

Variable definitions

Variable	Value
<atmServiceCategory>	is the ATM service category.
<Ds1>	is the instance number of the DS1 tributary, having a value between 0 and 31.
<DS_Chan1>	is the instance number of the DS1 channel, having a value between 0 and 23.
<E1>	is the instance number of the E1 channel, having a value between 0 and 31.
<E1_Chan1>	is the instance number of the E1 channel, having a value between 0 and 31.
<Lp1>	is the instance number of the logical processor associated with the function processor.
<MSA32cardType>	is the MSA32 card type: either 32pDS1MSA32 or 32pE1MSA32.
<pcrValue>	is the peak cell rate value.

Procedure job aid

PPP/ATM interworking for Multiservice Switch 7400 component hierarchy



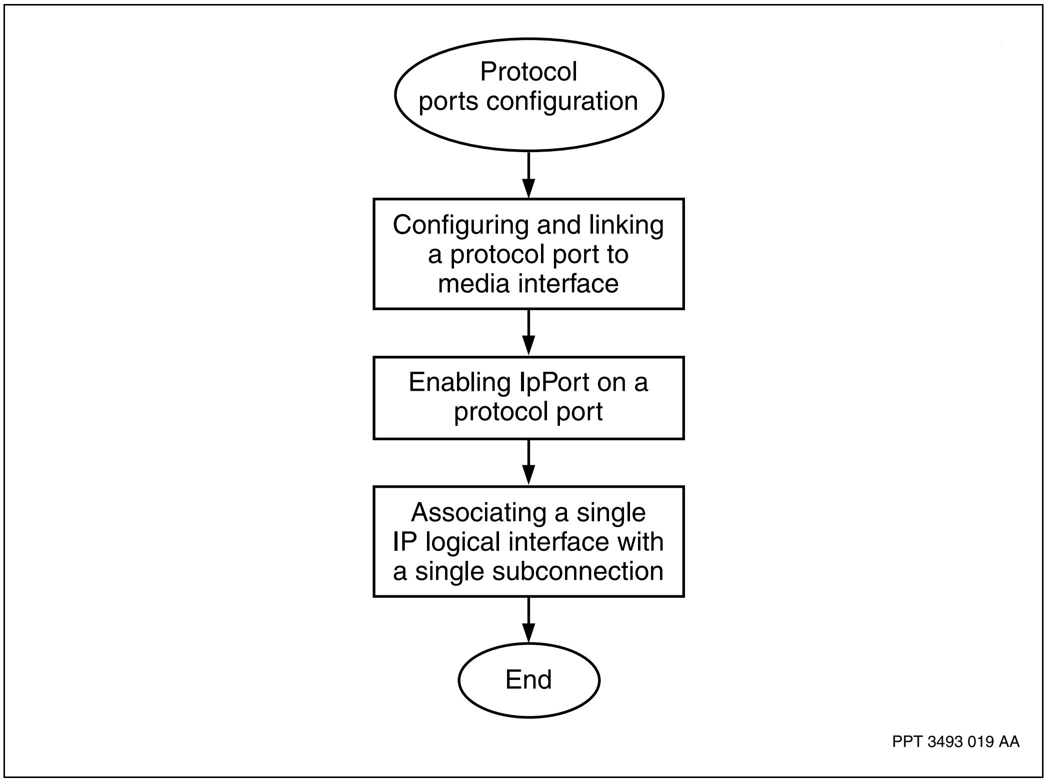
Protocol ports configuration

Configure the protocol ports on Nortel Networks Multiservice Switch nodes to provide a link to the access media that will carry IP traffic.

Protocol ports configuration procedures

This task flow shows you the sequence of procedures you perform to configure the protocol ports on Nortel Networks Multiservice Switch nodes. To link to any procedure, go to Protocol ports configuration procedure navigation (page 97).

Protocol ports configuration procedures


Protocol
ports configuration
Configuring and linking a protocol port to media interface
Enabling IpPort on a protocol port
Associating a single IP logical interface with a single subconnection
End
PPT 3493 019 AA

Protocol ports configuration procedure navigation

- Configuring and linking a protocol port to a media interface (page 99)

- Enabling IpPort on a protocol port (page 101)
- Associating a single IP logical interface with a single subconnection (page 103)

Configuring and linking a protocol port to a media interface

Configure and link a protocol port to a media interface to represent a physical instance of a data link or media protocol. You can configure multiple protocol ports on a virtual router.

Procedure steps

Step	Action
1	Add a *ProtocolPort* component as a subcomponent of the *Vr* component. **add Vr/<vr_name> ProtocolPort/<pp_name>**
2	Configure the association between the media interface and the *ProtocolPort* component by setting the *linkToProtocolPort* attribute: **set <media_interface> linkToProtocolPort Vr/<vr_name> ProtocolPort/<pp_name>**

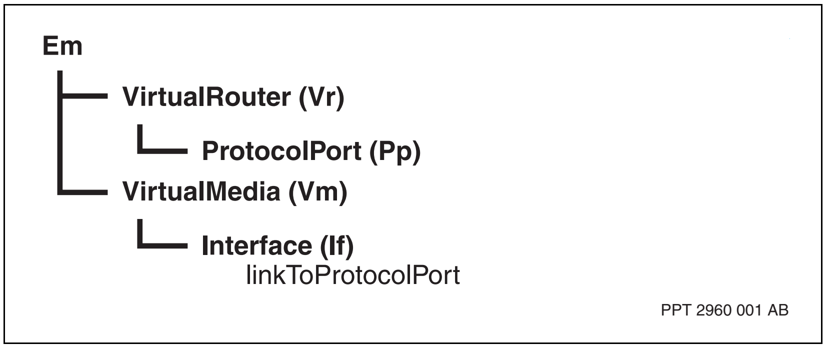
--End--

Variable definitions

Variable	Value
<media_interface>	is the interface name created during the provisioning of a particular medium. Application name values are formatted as a component type/instance value, for example, LA/31. If you are configuring IP over frame relay using frame relay DTE, the media interface is a frame relay remote group, for example, *FrDte/<n> Rg/1*. For more information, see Frame relay DTE access media configuration (page 41). If you are configuring IP over frame relay using an IP-optimized DLCI, the media interface is an IP DLCI group, for example, *IpDlciGroupr/<n>*. For more information, see VR IP-optimized DLCI access media configuration (page 60). If you are configuring an IP tunnel, the media interface is the IP tunnel interface, for example, *Vr/<vr_name> IP Tunnel Sep/ <sep_id>*. For more information, see Configuring PTP tunnels (page 216).
<pp_name>	is the name of the protocol port.
<vr_name>	is the name of the virtual router.

Procedure job aid

Configuring and linking protocol ports to the media interface component hierarchy



Enabling IpPort on a protocol port

Enable IP on a protocol port by adding an *IpPort* component to the *ProtocolPort* component. That enables IP routing on that port.

Procedure steps

Step **Action**

1 Add an *IpPort* component as a subcomponent of the *ProtocolPort* component.

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
```

2 Add an IP logical interface.

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ip_addr>
```

You can configure an IP logical interface against a specific media subconnection, which gives each subconnection its own subnet. See Associating a single IP logical interface with a single subconnection (page 103) to complete this procedure.

3 Provision a network mask for the protocol port. For networks that contain routers running RIP version I, the netmask should be the same for all RIP interfaces.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ip_addr> netMask <netmask>
```

4 Provision a broadcast address for the *ProtocolPort* component.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ip_addr> broadcastAddress <broadcast_addr>
```

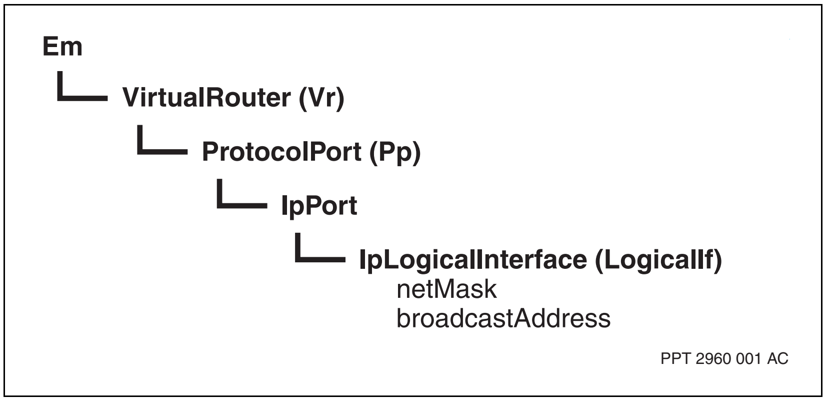
--End--

Variable definitions

Variable	Value
<broadcast_addr>	is the broadcast address of the attached IP network or subnetwork.
<ip_addr>	is the 32-bit IP address assigned to this logical interface.
<netmask>	is the network mask to be used with the IP address.
<pp_name>	is the name of the protocol port.
<vr_name>	is the name of the virtual router.

Procedure job aid

IP on a protocol port component hierarchy



Associating a single IP logical interface with a single subconnection

Associate a single IP logical interface with a single subconnection when you want each subconnection to support a specific IP subnet.

Prerequisites

- The required access media has been provisioned. See one of the following:
  - — VR ATM MPE access media configuration (page 22)
  - — Frame relay DTE access media configuration (page 41)
  - — VR IP-optimized DLCI access media configuration (page 60)

Procedure steps

Step	Action
1	Configure the association between the IP logical interface and the subconnection.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ip_addr> linkToMediaConnection
<media_subconnection>
```

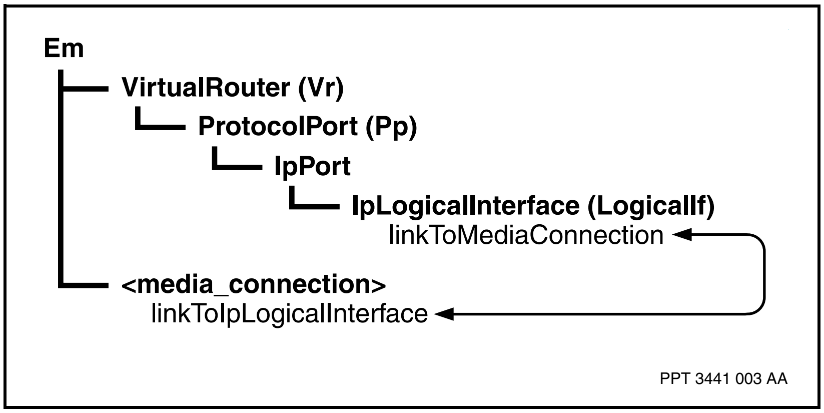
--End--

Variable definitions

Variable	Value
<ip_addr>	is the 32-bit IP address assigned to this logical interface.
<media_subconnection>	is the required media subconnection, which can be • for FrDte, component *FrDte StaticDlci* • for IP-optimized DLCI, component *IpDlciGroup Frc* • for ATM MPE, component *AtmMpe Ac*
<pp_name>	is the name of the protocol port.
<vr_name>	is the name of the virtual router.

Procedure job aid

Single IP logical interface with a single subconnection component hierarchy



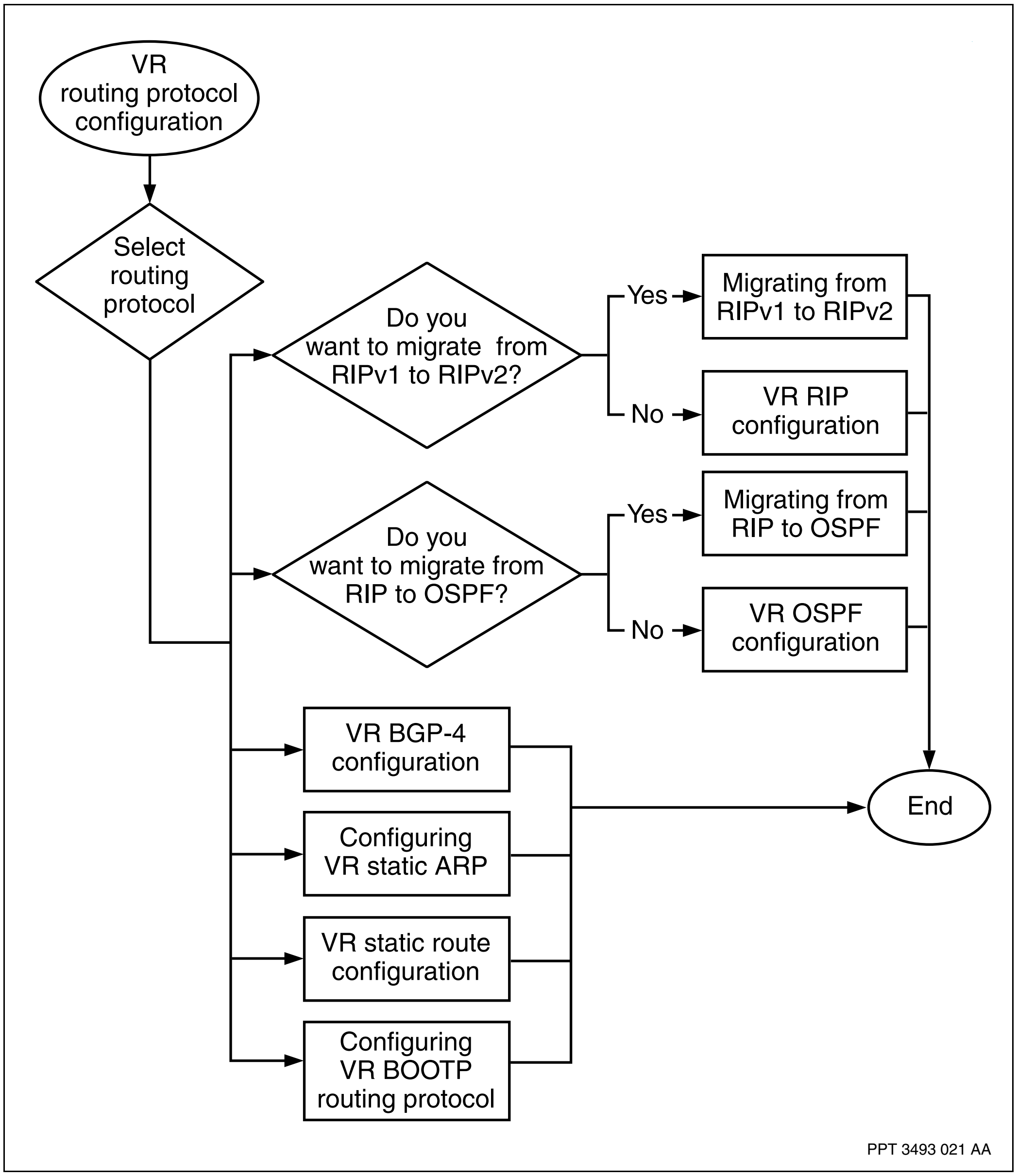
VR routing protocol configuration

Configure VR routing protocols to establish dynamic routing protocols and static routes that enable the exchange of route information.

VR routing protocol configuration tasks

This work flow shows you the sequence of tasks you perform to configure VR routing protocols. To link to any task, go to VR routing protocol configuration task navigation (page 106).

VR routing protocol configuration tasks


VR routing protocol configuration
Select routing protocol
Do you want to migrate from RIPv1 to RIPv2?
Yes
No
Migrating from RIPv1 to RIPv2
VR RIP configuration
Do you want to migrate from RIP to OSPF?
Yes
No
Migrating from RIP to OSPF
VR OSPF configuration
VR BGP-4 configuration
Configuring VR static ARP
VR static route configuration
Configuring VR BOOTP routing protocol
End
PPT 3493 021 AA

VR routing protocol configuration task navigation

- Migrating from RIPv1 to RIPv2 (page 119)
- VR RIP configuration (page 108)
- Migrating from RIP to OSPF (page 124)
- VR OSPF configuration (page 126)
- VR BGP-4 configuration (page 154)
- Configuring VR static ARP (page 184)

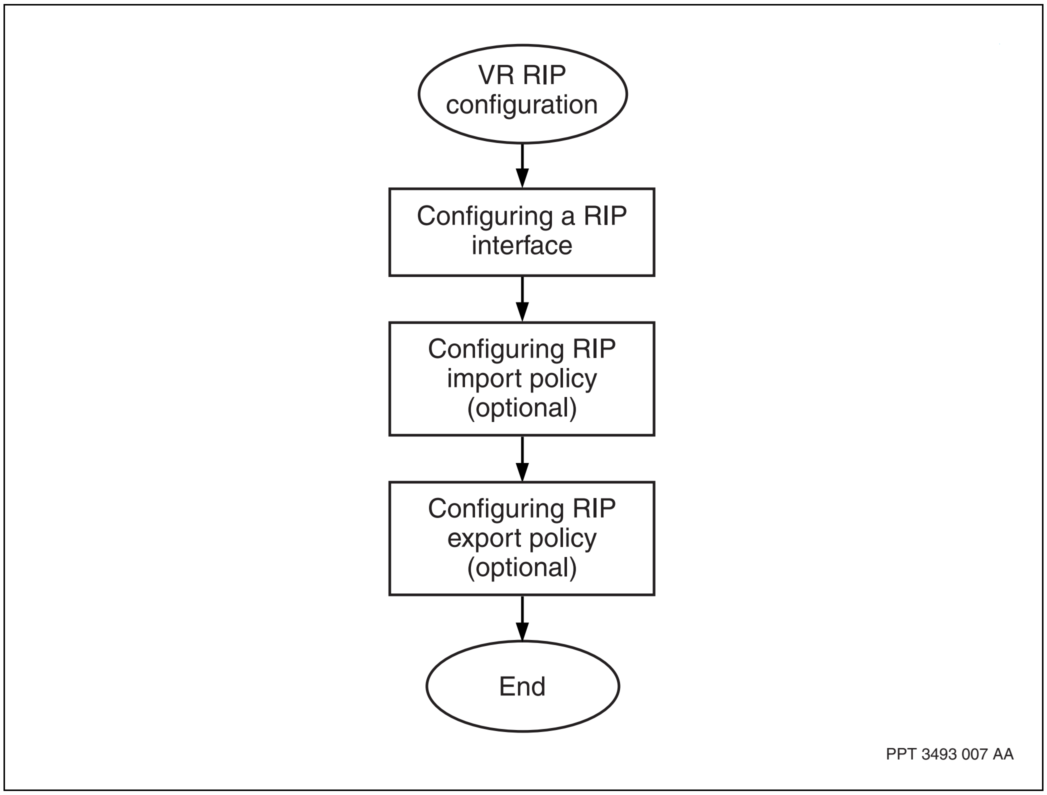
VR RIP configuration

Configure the routing information protocol (RIP) to exchange routing information within a network or between networks using RIP.

VR RIP configuration procedures

This task flow shows you the sequence of procedures you perform to configure RIP. To link to any procedure, go to VR RIP configuration procedure navigation (page 108).

VR RIP configuration procedures


VR RIP configuration
Configuring a RIP interface
Configuring RIP import policy (optional)
Configuring RIP export policy (optional)
End
PPT 3493 007 AA

VR RIP configuration procedure navigation

- Configuring a RIP interface (page 110)

- Configuring RIP import policy (page 113)
- Configuring RIP export policy (page 115)

Configuring a RIP interface

Configure a routing information protocol (RIP) interface to exchange RIP routing information with a RIP neighbor.

Prerequisites

> **CAUTION**
>
> **Changing the value of the ripIf attribute**
>
> Changing the value of some of the *ripIf* component attributes (including *IfConfSend* or *ifConfReceive* attributes) will cause a brief interruption of service on the interface. For information on the behavior of these attributes, see NN10600-060 *Nortel Networks Multiservice Switch 7400/15000/20000 Component Reference*.

- Configure protocol ports for each interface to be included in RIP routing exchanges. See Configuring and linking a protocol port to a media interface (page 99).
- Configure *LogicalIf* components for each interface to be included in RIP routing exchanges. See Enabling IpPort on a protocol port (page 101).

Procedure steps

Step **Action**

1 Add a *Rip* component as a subcomponent of the *Ip* component.

```
add Vr/<vr_name> Ip Rip
```

2 If required, change the route preference.

```
set Vr/<vr_name> Ip Rip defaultRipRtePref <route_pref>
```

3 Add a *RipIf* component to at least one logical interface.

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> RipIf
```

4 Set the version of RIP updates to send from this RIP interface.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> RipIf ifConfSend <tx_value>
```

5 Set the version of RIP updates to receive (accept) on this RIP interface.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> RipIf ifConfReceive <rx_value>
```

All other attributes under the *ripIf* component do not need to be changed from their default values.

6 Optionally, add the *Neighbor* subcomponent. The *Neighbor* subcomponent of the *RipIf* component describes the RIP neighbor for this logical interface. If this protocol port is configured as non-broadcast/multi-access (NBMA), as specified by the *lanModel* attribute under the *IpPort* subcomponent, then the *Neighbor* subcomponent must be provisioned.

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> RipIf Neighbor/<remote_IpAddress>
```

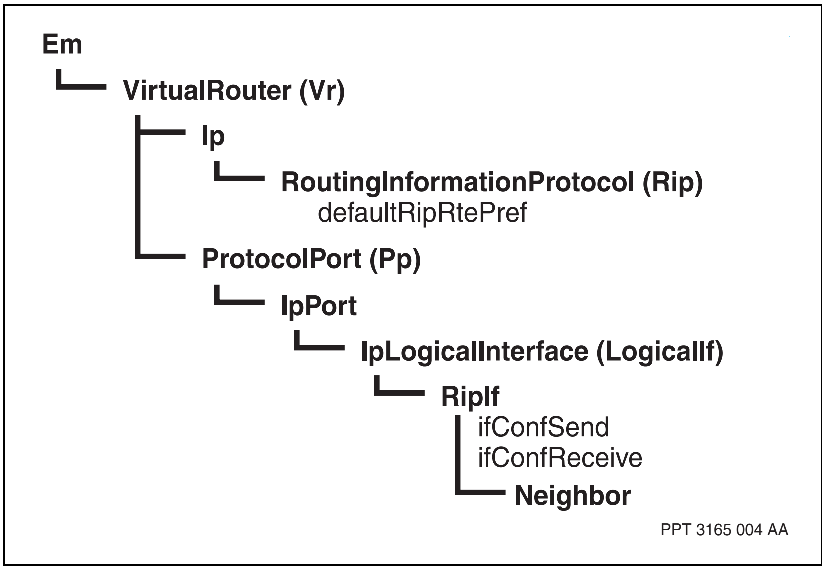
--End--

Variable definitions

Variable	Value
<ipAddress>	is the IP address of the local interface.
<pp_name>	is the name of the protocol port.
<remote_IpAddress>	is the IP address of the remote neighbor interface.
<route_pref>	is the route preference.
<rx_value>	is the value you assign to the *ifConfReceive* attribute. This can be *v1* (allow reception of RIP 1 packets only), *v2* (allow reception of RIP 2 packets only), *both* (allow reception of both RIP 1 and RIP 2 packets), or *reject* (block receipt of RIP packets).
<tx_value>	is the value you assign to the *ifConfSend* attribute. This can be *v1* (send RIP 1 packets only), *v2* (multicast RIP 2 packets only), *v2b* (support RIP 2 packets only), or *silent* (block transmission of RIP packets).
<vr_name>	is the name of the virtual router.

Procedure job aid

RIP interface component hierarchy



Configuring RIP import policy

Configure routing information protocol (RIP) import policy to define which routes learned from RIP neighbors are to be used, and what metrics to assign to them.

Procedure steps

Step	Action
1	Add an *Import* component to the *Rip* component. **`add Vr/<vr_name> Ip Rip Import/<import_policy_number>`**
2	Set the *usageFlag* attribute to indicate whether the routes matching the criteria defined in this import policy are to be used or ignored. **`set Vr/<vr_name> Ip Rip Import/<import_policy_number> usageflag <usageflag_toggle>`**
3	Set the *interface* attribute if you want this import policy to apply to all routes learned from RIP updates on a particular interface. **`set Vr/<vr_name> Ip Rip Import/<import_policy_number> interface <interface_instance>`**
4	Set the *neighbor* attribute if you want this import policy to apply to all routes learned from a particular neighbor. **`set Vr/<vr_name> Ip Rip Import/<import_policy_number> neighbor <interface_instance>`**
5	Set an import metric if you want to overwrite the learned metric of an imported route with a specified metric. **`set Vr/<vr_name> Ip Rip Import/<import_policy_number> importMetric <cost>`**

6	Add one or more *Network <ipaddress> <ipmask>* components if you want to restrict the effects of this import policy to specified network ranges.
	`add Vr/<vr_name> Ip Rip Import/<import_policy_number> Network/<net_instance>`
	`set Vr/<vr_name> Ip Rip Import/<import_policy_number> Network/<net_instance> ipaddress <ip_addr>, ipmask <ip_mask>`

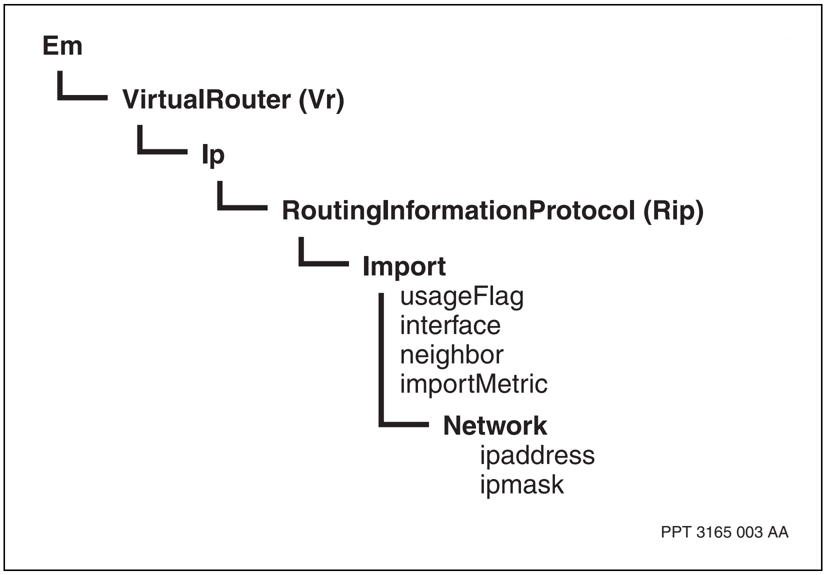
--End--

Variable definitions

Variable	Value
<cost>	is a decimal metric cost to be used for the routes that match the criteria defined in this import policy.
<import_policy_number>	is the instance assigned to the import policy.
<ip_addr>	is the IP address of the network.
<interface_instance>	is the IP address of the interface from which the routing information is learned.
<ip_mask>	is the mask used by the network.
<net_instance>	is the instance assigned to this instance of the *Network* component.
<usageflag_toggle>	specifies whether to use or ignore routes which match the criteria defined in the import policy.
<vr_name>	is the name of the virtual router.

Procedure job aid

RIP import policy component hierarchy



Configuring RIP export policy

Configure routing information protocol (RIP) export policy to define which routing information to advertise to RIP neighbors and what metrics to use.

Procedure steps

Step **Action**

1 Add an *Export* component to the *Rip* component.

```
add Vr/<vr_name> Ip Rip Export/<export_policy_number>
```

2 Set the *advertiseStatus* attribute to indicate whether the routes matching the criteria defined in this export policy are to be sent or blocked.

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
advertiseStatus <advertise_toggle>
```

3 Set the *ripInterface* attribute if you want this export policy to apply to all routes learned from RIP updates on a particular interface.

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
ripinterface <ipAddress>
```

If you specify a RIP interface, set the *protocol* attribute to *rip* when you complete step 4.

4 Optionally, set the *egpAsId* attribute if you want this export policy to apply to all routes learned from EGP which have the specified autonomous system identifier.

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
egpAsId <egpAsId>
```

If you specify an egpAsId, set the protocol to egp in step 6.

5 Optionally, set the ospfTag attribute if you want this export policy to apply to all routes learned from OSPF which have the specified OSPF tag.

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
ospfTag <tag>
```

If you specify an ospfTag, set the protocol to ospfExternal in step 6.

6 Set the protocol attribute if you want this export policy to apply to routes learned from a particular routing protocol

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
protocol <protocol_type>
```

Note that bgpExternal routes are not included when the protocol attribute is set to all.

7 If you want to export bgpInternal routes and you set the *protocol* attribute to all in step 4, set the *redistributeIbgp* attribute as follows.

```
set Vr/<vr_name> Ip Rip redistributeIbgp true
```

If you set the *protocol* attribute to bgpInternal in step 4, you do not need to set the *redistributeIbgp* attribute.

8 Set the *exportMetric* attribute if you want to advertise the routes with a different metric than the one learned. Routes learned from other routing protocols may have metrics larger than the RIP maximum of 15, which results in them being treated as unreachable by RIP.

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
exportMetric <cost>
```

9 Add one or more *Network* components if you want to restrict the effects of this export policy to specified network ranges.

```
add Vr/<vr_name> Ip Rip Export/<export_policy_number>
Network/<net_instance>
```

10 Set the *bgpAsId* attribute if you want this export policy to apply to all routes learned from BGP which have the specified autonomous system identifier.

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
bgpAsId <bgpAsId>
```

11 If you want the routes matching the criteria defined in this export policy to be advertised only on a specific RIP interface, set the *outInterface* attribute.

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
outif <interface_instance>, advertiseStatus block,
protocol rip
```

--End--

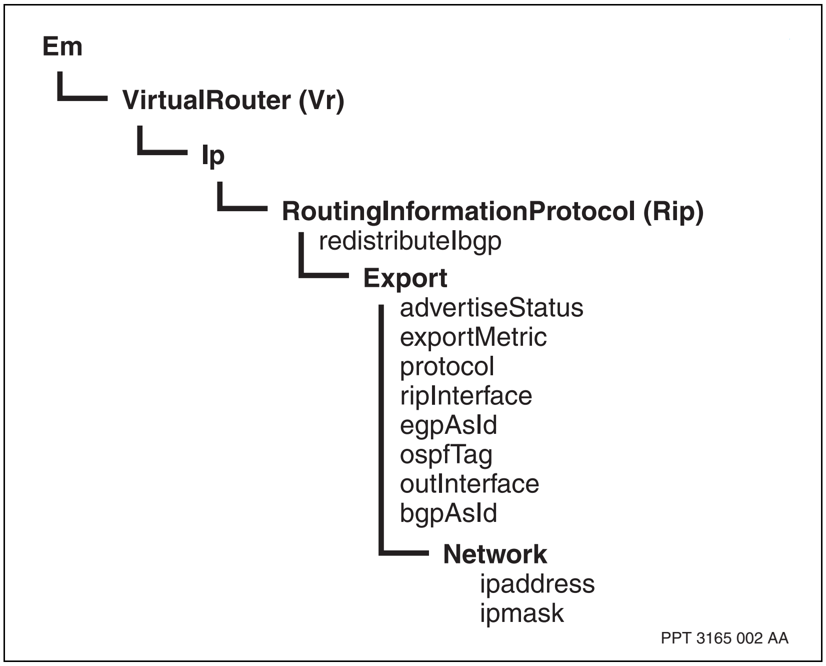
Variable definitions

Variable	Value
<advertise_toggle>	specifies whether to advertise (send) or not advertise (block) routes that match the criteria defined in this export policy.
<cost>	is a decimal metric cost to be used for the routes that match the criteria defined in this export policy.
<bgpAsId>	is a decimal identifier for a BGP autonomous system.
<egpAsId>	is a decimal identifier for an EGP autonomous system.
<export_policy_number>	is the numeric designation assigned to the export policy.
<interface_instance>	is the IP address of the interface from which the routing information is learned.
<ip_addr>	is the IP address of the network.
<ipAddress>	is the IP address of the interface.
<ip_mask>	is the mask used by the network.
(1 of 2)	

Variable	Value
<net_instance>	is the numeric designation assigned to this instance of the *Network* component.
<protocol_type>	identifies the protocol(s) to which the export policy applies. If you set the *protocol* attribute to all, the export policy applies to all routes in the forwarding table except bgpInternal and bgpExternal. If you want to export bgpInternal routes, create an export policy with the *protocol* attribute set to bgpInternal, or see step 7. If you want to export bgpExternal routes, create an export policy with the *protocol* attribute set to bgpExternal.
<tag>	is a decimal number that corresponds to an OSPF tag value.
<vr_name>	is the name of the virtual router.
(2 of 2)	

Procedure job aid

RIP export policy component hierarchy



Migrating from RIPv1 to RIPv2

Migrate from routing information protocol (RIP) from RIPv1 to RIPv2 to take advantage of additional capabilities that provide more informed routing and reduced processing load.

Migrate all Nortel Networks Multiservice Switch nodes on a link-by-link basis, until all the nodes in the network are set to RIPv2. The following steps are executed for each link. They are illustrated in figure Example RIPv1 to RIPv2 migration using two nodes (page 121). The table Example RIPv1 to RIPv2 migration using two nodes (page 121) provides more detail on RIP behavior with different combinations of *ifConfSend* and *ifConfReceive* values.

Prerequisites

- To migrate Nortel Networks Multiservice Switch network nodes from RIPv1 to RIPv2, all nodes must be running a release of software that supports RIPv2 (R5.1 and later). For more information, see NN10600-270 *Nortel Networks Multiservice Switch 7400/15000/20000 Software Installation* and NN10600-272 *Nortel Networks Multiservice Switch 7400/15000/20000 Upgrading Software*.

Procedure steps

Step **Action**

1 Configure the RIP interface on Multiservice Switch 2 to send RIPv2 updates in such a way that RIPv1 can read them.

```
set vr/<vr_name2> pp/<pp_name> ip log/<ip_address> RipIf
ifConfSend rip1Compatible
```

2 Configure the RIP interface on Multiservice Switch 1 to send and accept RIPv2 updates only.

```
set vr/<vr_name1> pp/<pp_name> ip log/<ip_address> RipIf
ifConfReceive ripVersion2 set vr/<vr_name1> pp/<pp_name>
ip log/<ip_address> RipIf ifConfSend ripVersion2
```

3 Configure the RIP interface on Multiservice Switch 2 to send and accept RIPv2 updates only.

```
set vr/<vr_name2> pp/<pp_name> ip log/<ip_address> RipIf
ifConfReceive ripVersion2 set vr/<vr_name2> pp/<pp_name>
ip log/<ip_address> RipIf ifConfSend ripVersion2
```

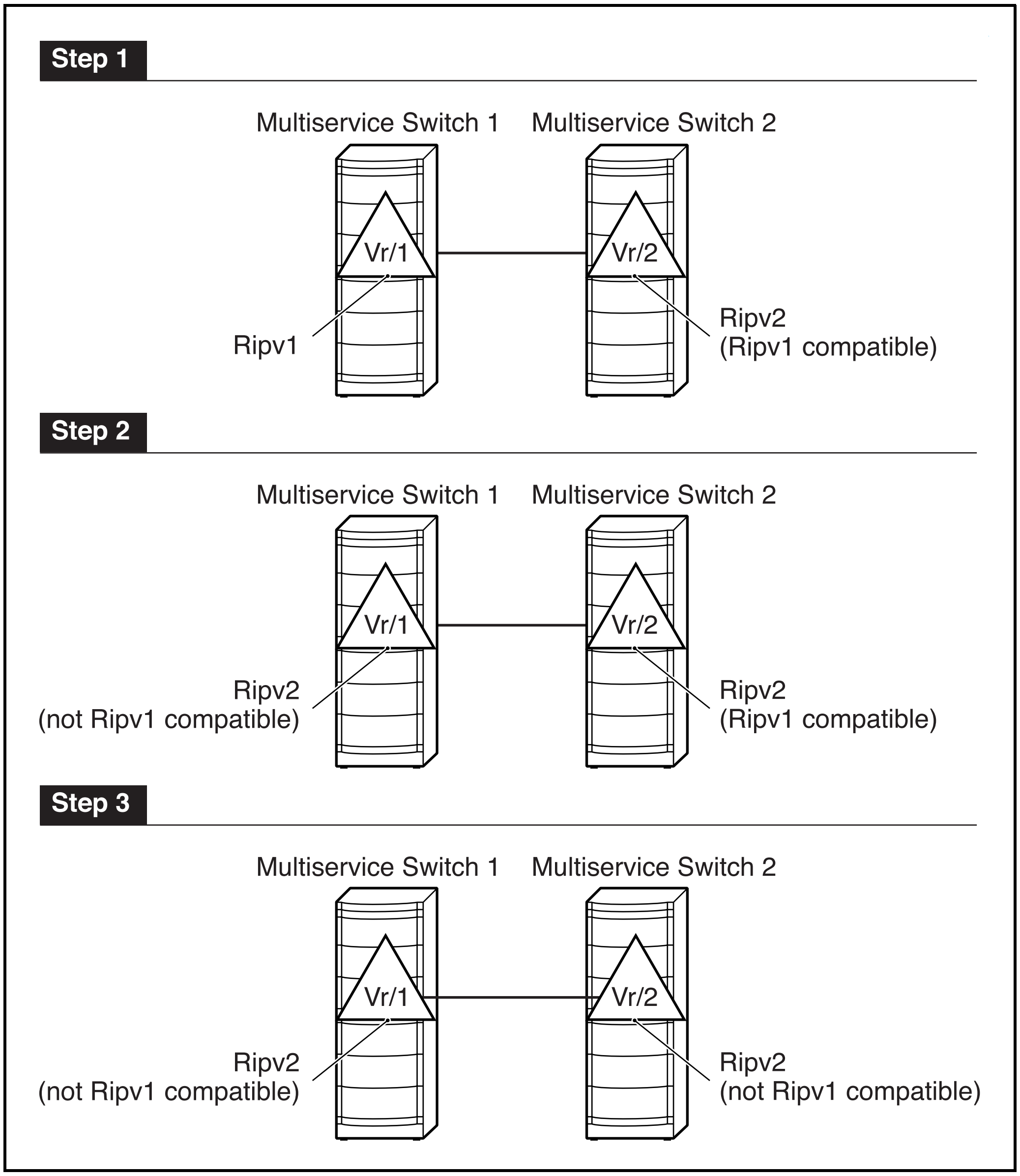
--End--

Variable definitions

Variable	Value
<ip_address>	is the IP address of the local interface.
<pp_name>	is the name of the protocol port.
<vr_name1>	is the name of the first virtual router.
<vr_name2>	is the name of the second virtual router.

Procedure job aid

Example RIPv1 to RIPv2 migration using two nodes



Example migration”RIP behavior of two nodes with different RIP configurations

ifConfSend attribute value on Multiservice Switch 1 (Vr/1) (transmitting)	**ifConfReceive attribute value on Multiservice Switch 2 (Vr/2) (receiving)**			
	***v1* (RIP 1)**	***v2* (RIP 2)**	***both* (RIP 1 or 2)**	***reject* (do not accept)**
silent (do not send)	No transmission	No transmission	No transmission	No transmission/updates are rejected.
v1 (RIP 1)	RIP 1 updates broadcast by Vr/1. RIP 1 updates accepted by Vr/2.	RIP 1 updates broadcast by Vr/1. RIP 1 updates rejected by Vr/2.	RIP 1 updates broadcast by Vr/1. RIP 1 updates accepted by Vr/2. The Vr/2 RIP interface processes the updates as RIP 1 updates.	RIP 1 updates broadcast by Vr/1. RIP 1 updates are rejected by Vr/2.
(1 of 2)				

Example migration”RIP behavior of two nodes with different RIP configurations (continued)

ifConfSend attribute value on Multiservice Switch 1 (Vr/1) (transmitting)	**ifConfReceive attribute value on Multiservice Switch 2 (Vr/2) (receiving)**			
	v1 **(RIP 1)**	***v2*** **(RIP 2)**	***both*** **(RIP 1 or 2)**	***reject*** **(do not accept)**
v2b (RIP 1 compatible)	RIP 2 updates broadcast by Vr/1. RIP 2 updates accepted by Vr/2. The Vr/2 RIP interface processes the RIP 2 updates as RIP 1 updates. (The Vr/2 RIP interface ignores the subnet mask and next hop fields in the RIP 2 update.)	RIP 2 updates broadcast by Vr/1. RIP 2 updates accepted by Vr/2. Vr/2 RIP interface will process the subnet mask and next hop fields in the RIP 2 updates.	RIP 2 updates broadcast by Vr/1. RIP 2 updates accepted by Vr/2. Vr/2 RIP interface will process the subnet mask and next hop fields in the RIP 2 updates.	RIP 2 updates broadcast by Vr/1. RIP 2 updates are rejected by Vr/2.
v2 (RIP 2)	RIP 2 updates are multicast by Vr/1. Because the Vr/2 RIP interface is set for RIP 1 only, Vr/1 will not send RIP 2 updates to Vr/2.	RIP 2 updates are multicast by Vr/1. RIP 2 updates are accepted by Vr/2. Vr/2 RIP interface will process the subnet mask and next hop fields in the RIP 2 updates.	RIP 2 updates are multicast by Vr/1. RIP 2 updates are accepted by Vr/2. Vr/2 RIP interface will process the subnet mask and next hop fields in the RIP 2 updates.	Updates are rejected
(2 of 2)				

Migrating from RIP to OSPF

Migrate from routing information protocol (RIP) to open shortest path first (OSPF) to achieve more sophisticated routing, greater administrative control, and quicker convergence.

Prerequisites

- Add OSPF to all the virtual routers to be migrated. See Adding OSPF to a VR (page 129).

Procedure steps

Step	Action
1	Change the route preference of either RIP or OSPF internal so that RIP routes are preferred over OSPF internal routes. This change must be done on all RIP and OSPF virtual routers in the network/autonomous system at the same time. **`set Vr/<vr_name> Ip Ospf defaultOspfIntRtePreference <rte_pref> or set Vr/<vr_name> Ip Rip defaultRipRtePreference <rte_pref>`**
2	Add OSPF interfaces wherever there are RIP interfaces to be migrated. See Configuring an OSPF interface (page 135). See also chapter VR OSPF configuration (page 126) for more information about configuring OSPF attributes.
3	After OSPF is stable and OSPF routes are as desired, change the route preference modified in step 1 back to the default value. This change must be done on all RIP and OSPF virtual routers in the network/autonomous system at the same time. OSPF will then be more preferred than RIP and the OSPF routes will begin to be used. Changing the route preference does not cause an interruption in service.
4	Delete the RIP interfaces which are no longer required. **`delete Vr/<vr_name> Pp/<pp_name> IpPort LogicalIf/<if_addr> RipIf`**
5	Delete the RIP components which are no longer required.

```
delete Vr/<vr_name> Ip Rip
```

--End--

Variable definitions

Variable	Value
<if_addr>	is the address of the logical interface.
<pp_name>	is the name of the protocol port.
<rte_pref>	is the value assigned to the route preference.
<vr_name>	is the name of the virtual router.

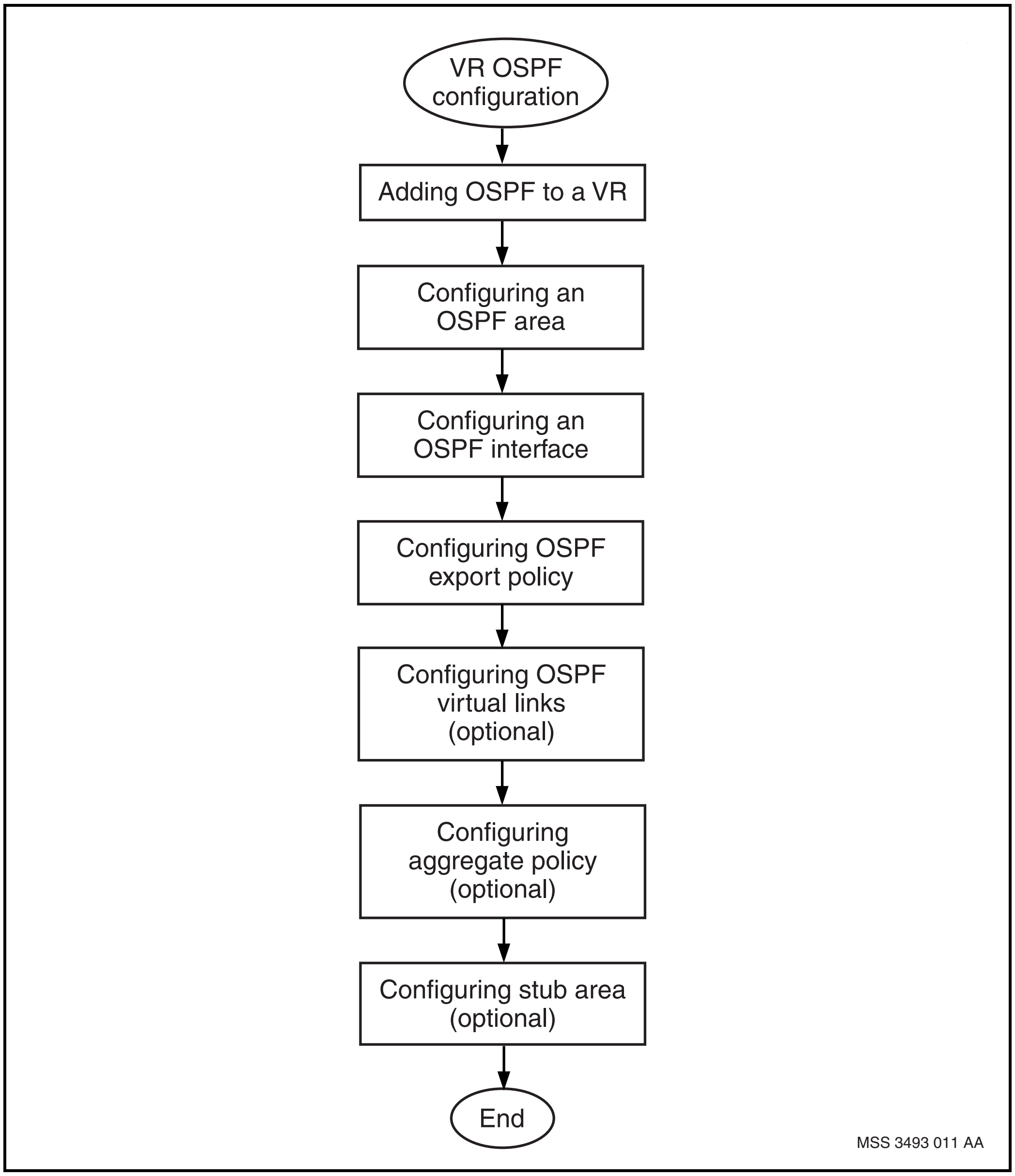
VR OSPF configuration

Configure the open shortest path first (OSPF) routing protocol to exchange routing information within a network or between networks using OSPF.

VR OSPF configuration procedures

This task flow shows you the sequence of procedures you perform to configure OSPF. To link to any procedure, go to VR OSPF configuration procedure navigation (page 127).

VR OSPF configuration procedures


VR OSPF configuration
Adding OSPF to a VR
Configuring an OSPF area
Configuring an OSPF interface
Configuring OSPF export policy
Configuring OSPF virtual links (optional)
Configuring aggregate policy (optional)
Configuring stub area (optional)
End
MSS 3493 011 AA

VR OSPF configuration procedure navigation

- Adding OSPF to a VR (page 129)
- Configuring an OSPF area (page 133)
- Configuring an OSPF interface (page 135)
- Configuring OSPF export policy (page 141)
- Configuring OSPF virtual links (page 145)
- Configuring an aggregate policy (page 150)

- Configuring a stub area (page 152)

Adding OSPF to a VR

Add OSPF to a virtual router (VR) to enable you to connect a Nortel Networks Multiservice Switch node to customer network through the protocol port.

Prerequisites

- If you are going to set attribute *Vr Ip Ospf spareInstance* in step 5 of the procedure, first set attribute *Shelf cpEquipmentProtection* to *hot*. Be aware that setting this attribute to *hot* causes the spare CP to restart.
- After an *Ospf* component is added under *Vr Ip* component, at least one *AreaEntry* has to be added under *Vr Ip Ospf* component, as described in Configuring an OSPF area (page 133).

Procedure steps

Step **Action**

1 Add an *Ospf* component as a subcomponent of the *Ip* component.

```
add Vr/<vr_name> Ip Ospf
```

2 Specify the IP address of the OSPF instance to identify it in the AS.

```
set Vr/<vr_name> Ip Ospf routerId <x.x.x.x>
```

3 Specify the estimated counts if you do not want to use the system defaults. Configuring the value of these attributes to accurate estimates improves performance and does not limit functionality.

```
set Vr/<vr_name> Ip Ospf
estimatedNumberOfInternalOSPFRoutes <int_routes>,
estimatedNumberOfExternalOSPFRoutes <ext_routes>,
estimatedNumberOfInterfacesPerArea <ifs>,
estimatedNumberOfNeighborsPerInterface <nbrs>
```

4 If you want to synchronize the OSPF instance maintained on the standby card, set attribute Shelf *cpEquipmentProtection* to hot. For more information, see NN10600-550 *Nortel Networks Multiservice Switch 7400/15000/20000 Common Configuration Procedures*.

5 Set the spare instance attribute according to whether you want a synchronized OSPF instance maintained on the standby card.

```
set Vr/<vr_name> Ip Ospf spareInstance <sparing_action>
```

6 If required, change the default setting for the alarm generator, which specifies what OSPF alarms are generated.

```
set Vr/<vr_name> Ip Ospf alarmGeneration <alarm>
```

7 If required, change the default for the Dijkstra timer.

```
set Vr/<vr_name> Ip Ospf spfHoldTime <hold_time>
```

8 If required, change the default for the ECMP setting.

```
set Vr/<vr_name> Ip Ospf ecmpStatus <ecmp>
```

9 Optionally, change the route preference attribute for OSPF internal routes.

```
set Vr/<vr_name> Ip Ospf defaultOspfIntRtePref
<route_pref>
```

10 Optionally, change the route preference attribute for OSPF external type 1 routes.

```
set Vr/<vr_name> Ip Ospf defaultOspfExt1RtePref
<route_pref>
```

11 Optionally, change the route preference attribute for OSPF external type 2 routes.

```
set Vr/<vr_name> Ip Ospf defaultOspfExt2RtePref
<route_pref>
```

12 Optionally, if you want the *Ospf* component to behave as an OSPF ASBR, change the default setting of the *AS border router status* attribute.

```
set Vr/<vr_name> Ip Ospf asBdrRtrStatus true
```

Also, at lease one *Export* component needs to be added under *Vr Ip Ospf* component, as described in Configuring OSPF export policy (page 141)".

--End--

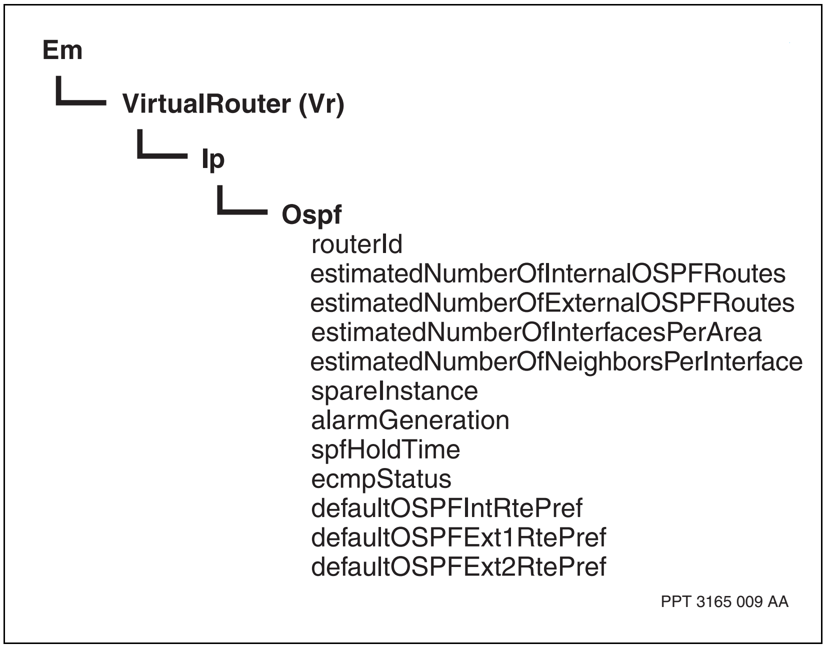
Variable definitions

Variable	Value
<alarm>	is the level of alarm generation for this instance of OSPF.
<ecmp>	is the attribute that specifies if equal cost multi-path (ECMP) is enabled for OSPF learned routes.
<ext_routes>	is the average estimated number of external routes to be exported into OSPF.
<hold_time>	is the time interval, in seconds, between OSPF Dijkstra calculations for this instance of OSPF. This is an important scaling factor during peak load times.
<ifs>	is the average estimated number of interfaces in each area of the AS.
<int_routes>	is the average estimated number of internal OSPF routes to be managed by this OSPF instance.
<nbrs>	is the average estimated number of neighbors for each OSPF interface.
(1 of 2)	

Variable	Value
<route_pref>	is the route preference. The attribute default is 30 for internal routes, 80 for external type 1 routes, and 120 for external type 2 routes.
<sparing_action>	is the spare instance setting. When set to disable, a synchronized OSPF instance is not maintained on the standby card.
<vr_name>	is the name of the virtual router.
<x.x.x.x>	is the 32-bit IP address that uniquely identifies the OSPF router in the AS.
(2 of 2)	

Procedure job aid

OSPF to a VR component hierarchy



Configuring an OSPF area

Configure an OSPF area to specify which area the *Ospf* component is going to participate in routing.

Optionally, set the authentication type for the area to MD5, if you want to use MD5 authentication in the area's OSPF routing exchanges.

Prerequisites

- There must be at least one *AreaEntry* provisioned under a *Vr Ip Ospf* component.

Procedure steps

Step **Action**

1 Add an *AreaEntry* as a subcomponent of the *Ospf* component.

```
add Vr/<vr_name> Ip Ospf AreaEntry/<area_id>
```

If this area 1 is not to be used as a transit area, set attributes *importAsExtern* and *areaSummary* appropriately as described in Configuring a stub area (page 152).

2 Optionally, set the authentication type for the area to MD5, if you want to use MD5 authentication in the area's OSPF routing exchanges.

```
set Vr/<vr_name> Ip Ospf AreaEntry/<area_id> authType
md5
```

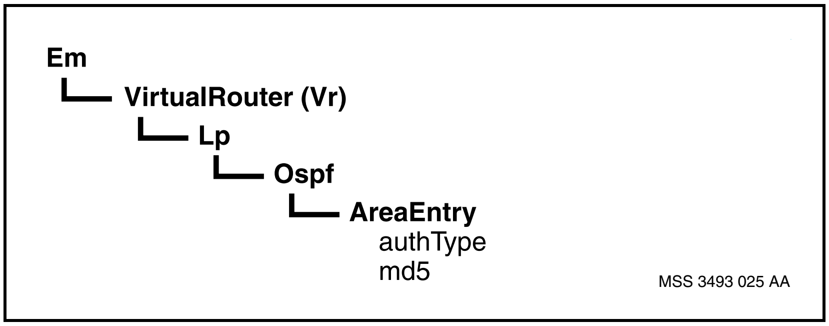
--End--

Variable definitions

Variable	Value
<area_id>	is the OSPF area ID in the format of an IP address.
<auth_type>	is the authentication type specified for the area.
<vr_name>	is the name of the virtual router.

Procedure job aid

OSPF area component hierarchy



Configuring an OSPF interface

Configure an OSPF interface to connect the protocol port on a Nortel Networks Multiservice Switch node to your network or its peers.

Optionally, configure MD5 authentication on an OSPF link to provide protection of OSPF neighbor relationships. Authentication for the OSPF protocol is controlled on a per interface/subnet basis. Change an MD5 key using the procedure Changing an MD5 key on an authenticated OSPF link (page 139).

Prerequisites

- To participate in all OSPF area routing, it is recommended that you provision at least one OSPF interface for each OSPF area, and that you link the OSPF interface to an area.
- Configure a parent component (*ProtocolPort*) for each interface to be included in OSPF routing. See Configuring and linking a protocol port to a media interface (page 99).
- Configure a parent component (*LogicalIf)* for each interface to be included in OSPF routing. See Enabling IpPort on a protocol port (page 101).
- If you are configuring OSPF on IP logical interfaces with links to individual connections (see Associating a single IP logical interface with a single subconnection (page 103)), it is recommended that you do not set attribute *Vr ProtocolPort IpPort LogicalIf OspfIf ifType* to broadcast. See step 3 of the procedure.
- If MD5 authentication is enabled on an *OspfIf*, it must have at least one *KeyId* component and one *md5Key* attribute provisioned.
- It is recommended that different MD5 key values be used on each *OspfIf* subnet.

Attention: As simultaneous provisioning of both the local and remote router is not possible, there will be an interim period during which the local router will have its authentication mode configured as MD5, but the remote router has not yet been configured and is therefore still operating in unauthenticated mode. During this interim period, both routers will raise authentication failure alarms, as their authentication modes do not match.

Procedure steps

Step	Action
1	To enable OSPF on a node, add the *OspfIf* component to each IP logical interface that is taking part in the OSPF process.

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> OspfIf
```

2 Set the *areaId* attribute for the OSPF *LogicalIf* component to define the OSPF area to which the interface connects:

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> OspfIf areaId <area_id>
```

3 Select the OSPF interface type.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> OspfIf ifType <if_type>
```

4 Optionally, configure an MD5 *KeyId* component under the *OspfIf*.

```
add Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/<ip_addr>
OspfIf md5KeyId/<n>
```

5 Set an MD5 key value under the *KeyId* component just added.

```
set Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/<ip_addr>
OspfIf md5KeyId/<n> md5Key <string>
```

6 Select the mode of operation if the interface type is point-to-multipoint.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> OspfIf pointToMultipoint
<mode_of_operation>
```

It is necessary to configure the neighbors for a point-to-multipoint interface in a non-broadcast network.

7 Add a neighbor component for each OSPF neighbor if the interface type is point-to-multipoint non-broadcast or NBMA.

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> OspfIf Neighbor/<ipAddr>
```

For broadcast networks, the OSPF neighbors are discovered automatically by the Hello protocol.

8 Add a *HostEntry* component to the *Ospf* component if you want to advertise a particular host address:

```
add Vr/<vr_name> Ip Ospf HostEntry/<ipAd>,<tos>
```

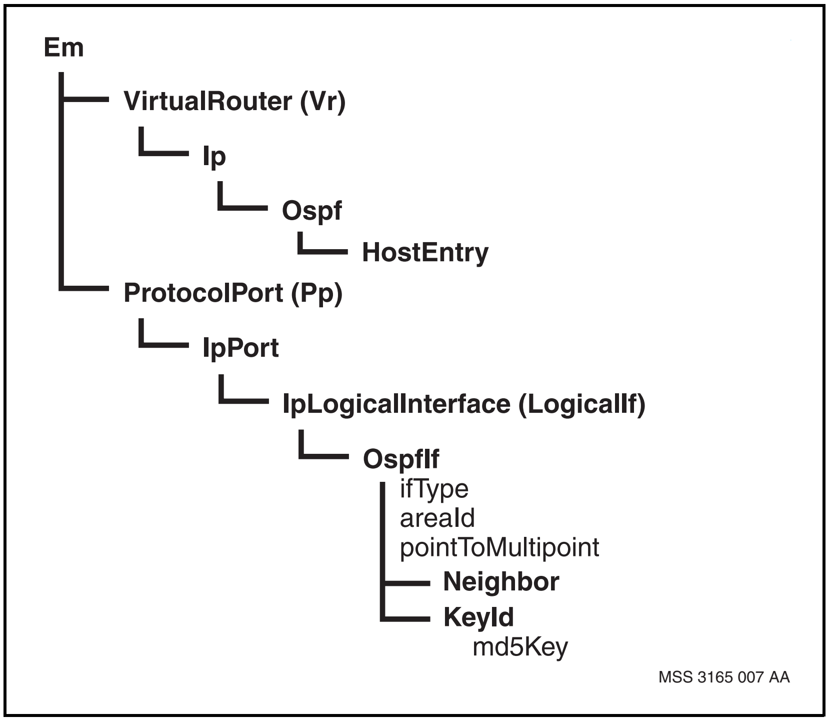
--End--

Variable definitions

Variable	Value
<area_id>	is the 32-bit IP address for the OSPF area in which this router is located. If you do not specify the area ID, the system will assume that the port is in the backbone, area ID 0.0.0.0.
<if_type>	is the type of interface used. If you are configuring OSPF on IP logical interfaces with links to individual connections (see Associating a single IP logical interface with a single subconnection (page 103)), it is recommended that you do not set the attribute *Vr ProtocolPort IpPort LogicalIf OspfIf ifType* to broadcast.
<ipAd>	is the host address to be advertised.
<ipAddr>	is the 32-bit address assigned to the OSPF neighbor.
<ipAddress>	is the 32-bit IP address assigned to this logical interface.
<mode_of_operation>	is the mode of operation value. A value of non-broadcast limits the interface to sending unicast packets whereas a broadcast value enables sending multicast OSPF Hello packets to dynamically discover neighbors.
<n>	is an instance of an MD5 key associated with an OSPF Virtual interface, value of 1-255.
<pp_name>	is the name of the protocol port.
<string>	is 1-16 ASCII characters.
<tos>	specifies the ToS byte value assigned to the packet.
<vr_name>	is the name of the virtual router.

Procedure job aid

OSPF interface component hierarchy



Changing an MD5 key on an authenticated OSPF link

Transition MD5 keys without having to terminate the neighbor relationship for which the key is being changed.

Prerequisites

- To minimize impact on performance, it is important to keep the key transition time to an absolute minimum. The key transition time is the time between configuring the new key information on the local router and configuring it on the remote router. During the key transition time, the local router will be duplicating each packet it transmits (one packet for each key).
- An old key should be deleted when it is no longer in use. Retaining only the newest key makes it obvious which key is in use. There are also performance impacts resulting from retaining the old keys, as incoming packets must be checked against all provisioned keys.

Attention: During the key transition time, the local router will be transmitting packets with both the old key and the new key. An alarm will be raised on the remote router during the key transition time as it is receiving packets authenticated with the new key, even though it has not yet been configured with that key. However, as the remote router is also receiving the same packets authenticated with the old key, the adjacency will not be lost.

Attention: If several obsolete keys have been retained, the user may identify which is the newest key (the key in use) by means of operational attributes.

Procedure steps

Step **Action**

1 Add a new key (*KeyId* <y>) under the *OspfIf* on the local and remote router.

```
add Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/<ip_addr>
OspfIf KeyId/<y>
set Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/<ip_addr>
OspfIf KeyId/<y> md5Key <string>
```

2 Remove the old key (*KeyId* <x>) from the local and remote router.

```
del Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/<ip_addr>
OspfIf KeyId/<x>
```

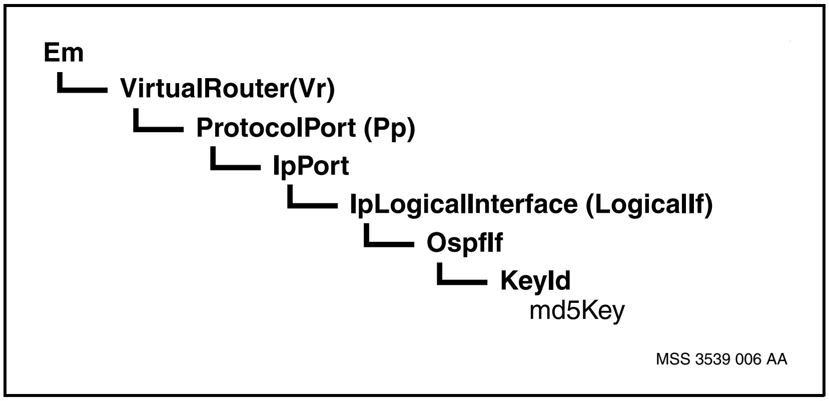
--End--

Variable definitions

Variable	Value
<ip_addr>	is the IP address.
<pp_id>	is the Protocol Port id.
<string>	is 1-16 ASCII characters.
<x>	is the id of the old MD5 key.
<y>	is id of the new MD5 key.
<vr_name>	is the virtual router id.

Procedure job aid

OSPF component hierarchy for changing an MD5 key on an authenticated OSPF link



Configuring OSPF export policy

Configure OSPF export policy to define how routing information learned from other routing protocols is blocked or advertised as OSPF external routes onto the attached OSPF areas.

Procedure steps

Step **Action**

1 Add an *Export* component to the *Ospf* component.

```
add Vr/<vr_name> Ip Ospf Export/<export_policy_number>
```

2 Set the *advertiseStatus* attribute to enable or disable the *Ospf* Export policy.

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
advertiseStatus <advertise_toggle>
```

3 Set the *asBdrRtrStatus* attribute to enable the OSPF export policy. If you set this attribute to false, which is the default, OSPF blocks the export of all non-OSPF learned routes. If you set this attribute to true, the VR being configured will be used as OSPF ASBR.

```
set Vr/<vr_name> Ip Ospf asBdrRtrStatus true
```

4 Optionally, set the *ripInterface* attribute if you want to limit the exported OSPF information to RIP routes learned from a particular RIP interface. This attribute can only be non zero when the protocol is set to RIP.

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
ripInterface <ipAddress>
```

5 Set the *protocol* attribute if you want to specify which routing protocol networks are advertised by this policy.

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
protocol <protocol_type>
```

If you provision the *ripInterface* attribute (see step 4), you must set the *protocol* attribute to all or rip.

6 If you want to export bgpInternal routes and you set the *protocol* attribute to all in step 5, set the *redistributeIbgp* attribute as follows.

```
set Vr/<vr_name> Ip Ospf redistributeIbgp true
```

If you set the *protocol* attribute to bgpInternal in step 5, you do not need to set the *redistributeIbgp* attribute.

7 Set the *ripNeighbor* attribute if you want to limit the exported OSPF information to a specific RIP neighbor. This attribute can only be non zero when the protocol is set to RIP.

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
ripNeighbor <ipAddr>
```

8 Set the *metric* attribute if you want to change the default cost metrics for exported routes.

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
metric <cost>
```

9 Add a *NetworkList* <ipaddress> <ipmask> component if you want to advertise the information defined in this policy about those networks and subnetworks contained in the *NetworkList* component.

```
add Vr/<vr_name> Ip Ospf Export/<export_policy_number>
NetworkList/<net_instance>
```

10 Configure the *NetworkList* <ipaddress> <ipmask> component.

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
NetworkList/<net_instance> ipaddress <ipAd>, ipmask
<ip_mask>
```

11 Set the *egpAsId* attribute if you want to limit the EGP routes to a specific AS.

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
egpAsId <as_id>
```

--End--

Variable definitions

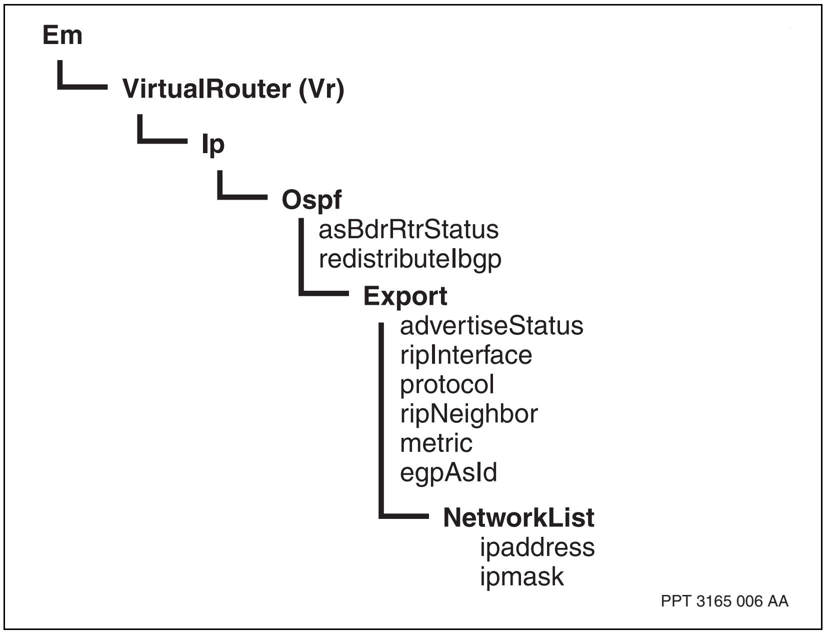
Variable	Value
<advertise_toggle>	‘is set to send (enable) or blocked (disable).
<as_id>	is the autonomous system number of the network routing from which the EGP route was learned.
<cost>	is the OSPF cost that is attached to exported routes.
<export_policy_number>	is the numeric designation assigned to this export policy.
<ipAd>	is the network or subnetwork address of the network.
<ipAddr>	is the 32-bit IP address of the specific router to which the information defined in this policy is advertised.
<ipAddress>	is the local IP address of the RIP interface.
<ip_mask>	is the subnet mask used by the network.
<net_instance>	is the numeric designation assigned to this instance of the *NetworkList* component.

(1 of 2)

Variable	Value
<protocol_type>	is the protocol type. If you set the *protocol* attribute to all, the export policy applies to all routes in the forwarding table except bgpInternal and bgpExternal. If you want to export bgpInternal routes, create an export policy with the *protocol* attribute set to bgpInternal, or see step 6. If you want to export bgpExternal routes, create an export policy with the *protocol* attribute set to bgpExternal.
<vr_name>	is the name of the virtual router.
(2 of 2)	

Procedure job aid

OSPF export policy component hierarchy



Configuring OSPF virtual links

Configure a virtual link to connect an isolated area, through another transit area, to the backbone.

Optionally, configure MD5 authentication on an OSPF virtual link to provide protection of OSPF neighbor relationships. Change an MD5 key by using the procedure Changing an MD5 key on an authenticated OSPF virtual link (page 148).

Prerequisites

- If MD5 authentication is enabled for an *Ospfl virtIfEntry,* it must have at least one *KeyId* component and one *md5Key* attribute provisioned.
- It is recommended that different MD5 key values be used on each virtual link.

Attention: As simultaneous provisioning of both the local and remote router is not possible, there will be an interim period during which the local router will have its authentication mode configured as MD5, but the remote router has not yet been configured and is therefore still operating in unauthenticated mode. During this interim period, both routers will raise authentication failure alarms, as their authentication modes do not match.

Procedure steps

Step **Action**

1 Add a *VirtIfEntry* subcomponent under the *Ospf* subcomponent.

```
add Vr/<vr_name> Ip Ospf VirtIfEntry/
<localAreaIdIpAddress>,<nbrRouterIdIpAddress>
```

2 Optionally, change the frequency of the OSPF hello protocol exchange interval.

```
set Vr/<vr_name> Ip Ospf VirtIfEntry/
<localAreaIdIpAddress>,<nbrRouterIdIpAddress>
helloInterval <interval>
```

3 Optionally, configure the OSPF simple authentication key if you want to provide a higher level of security to OSPF routing exchanges.

```
add Vr/<vr_name> Ip Ospf VirtIfEntry/
<localAreaIdIpAddress>,<nbrRouterIdIpAddress> authKey
<authentication_string>
```

4 Optionally, Configure an MD5 *KeyId* component under the *virtIfEntry.*

```
add Vr/<vr_name> Ip Ospf virtIfEntry/<virtif_id>
md5KeyId/<n>
```

5 Set an MD5 key value under the MD5 *KeyId* component just added.

```
set Vr/<vr_name> Ip Ospf virtIfEntry/<virtif_id>
md5KeyId/<n> md5Key <string>
```

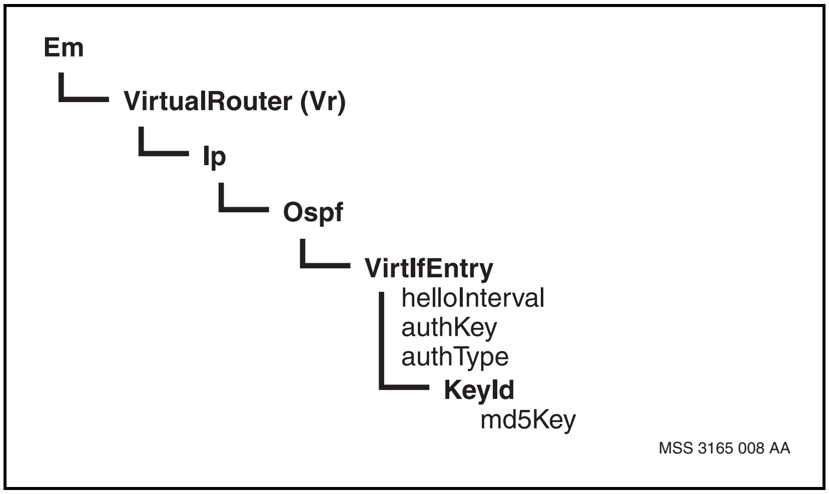
--End--

Variable definitions

Variable	Value
<authentication_string>	is an hexadecimal string of 1 to 8 bytes (Hex representation of ASCII character set, actually 16 Hex digits). Ensure that it matches the *authenticationkey* attribute currently in use in the area, otherwise routing exchanges are discarded.
<interval>	is the value of the hello exchange interval. You can choose to adjust the interval depending upon the media bandwidth and traffic conditions. All routers on that media must use the same hello interval.
<localAreaIdIpAddress>	is the OSPF area ID of the local end of the virtual link.
<n>	is an instance of an *md5Key* associated with an OSPF Interface or an OSPF Virtual interface, value of 1-255.
<nbrRouterIdIpAddress>	is the OSPF Router ID of the remote end of the virtual link.
<string>	is 1-16 ASCII characters.
<virtif_id>	is the *VirtIfEntry* id.
<vr_name>	is the name of the virtual router.

Procedure job aid

OSPF virtual links component hierarchy



Changing an MD5 key on an authenticated OSPF virtual link

Transition MD5 keys without having to terminate the neighbor relationship for which the key is being changed.

Prerequisites

- To minimize impact on performance, it is important to keep the key transition time to an absolute minimum. The key transition time is defined as the time between configuring the new key information on the local router and configuring it on the remote router. During the key transition time, the local router will be duplicating each packet it transmits (one packet for each key).
- An old key should be deleted when it is no longer in use. Retaining only the newest key makes it obvious which key is in use. There are also performance impacts resulting from retaining the old keys, as incoming packets must be checked against all provisioned keys.

Attention: During the key transition time, the local router will be transmitting packets with both the old key and the new key. An alarm will be raised on the remote router during the key transition time as it is receiving packets authenticated with the new key, even though it has not yet been configured with that key. However, as the remote router is also receiving the same packets authenticated with the old key, the adjacency will not be lost.

Attention: If several obsolete keys have been retained, the user may identify which is the newest key (the key in use) by means of operational attributes.

Procedure steps

Step **Action**

1 Add a new key (*KeyId* <y>) under the *virtIfEntry* on the local router.

```
add Vr/<vr_id> Ip Ospf virtIfEntry/<virtif_id> KeyId/<y>
set Vr/<vr_id> Ip Ospf virtIfEntry/<virtif_id> KeyId/<y>
md5Key <string>
```

2 Add the new key to the remote router.

```
add Vr/<vr_id> Ospf virtIfEntry/<virtif_id> KeyId/<y>
set Vr/<vr_id> Ip Ospf virtIfEntry/<virtif_id> KeyId/<y>
md5Key <string>
```

3 Remove the old key (*KeyId* <x>) from the local router.

```
del Vr/<vr_id> Ip Ospf virtIfEntry/<virtif_id> KeyId/<y>
```

4 Remove the old key (*KeyId* <x>) from the remote router.

```
del Vr/<vr_id> Ip Ospf virtIfEntry/<virtif_id> KeyId/<y>
```

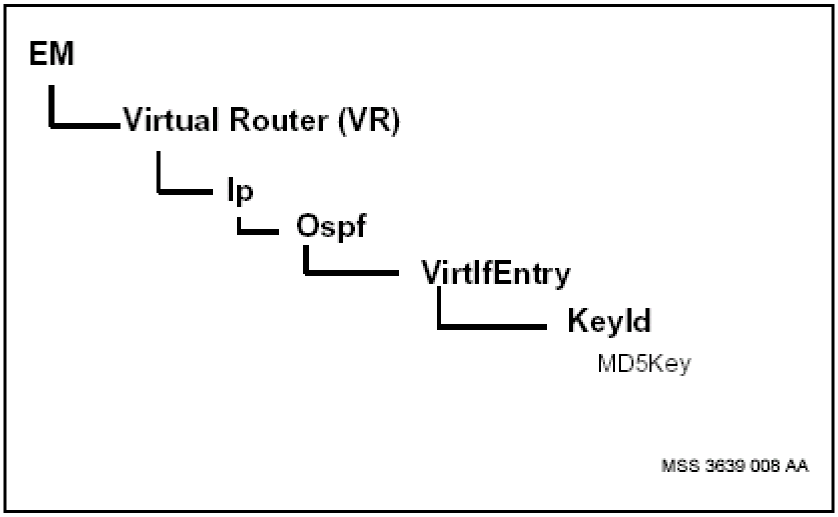
--End--

Variable definitions

Variable	Value
<string>	is 1-16 characters.
<x>	is the id of the old key.
<y>	is the id of the new key.
<vr_id>	is the virtual router id.
<virtif_id>	is the Virtual Entry id.

Procedure job aid

OSPF component hierarchy for changing an MD5 key on an authenticated virtual OSPF link



Configuring an aggregate policy

Configure an aggregate policy to reduce the number of LSAs summarized from one area and generated into another area.

Prerequisites

- If the *AreaAggregateEntry* component is provisioned for a NSSA, the *importAsExtern* of the valid *AreaEntry* has to be set to *importNssa*.

Procedure steps

Step **Action**

1 Add an *AreaAggregateEntry* as a subcomponent of the *Ospf* component.

```
add Vr/<vr_name> Ip Ospf AreaAggregateEntry/
<area_id>,<lsdb_type>,<aggregate_net>,<aggregate_mask>
```

The *AreaAggregateEntry* address must have the same area ID as the *AreaEntry* component that already exists. See Configuring an OSPF area (page 133).

2 Set aggregate policy effect attribute.

```
set Vr/<vr_name> Ip Ospf AreaAggregateEntry/
<area_id>,<lsdb_type>,<aggregate_net>,<aggregate_mask>
effect <advertise_decision>
```

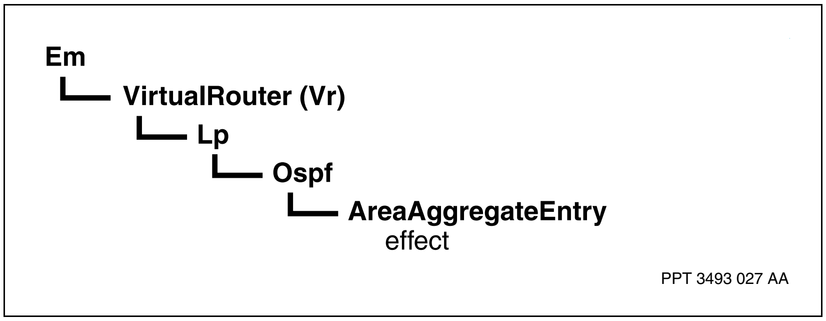
--End--

Variable definitions

Variable	Value
<advertise_decision>	is the value indicating whether to advertise matching routing information or not.
<aggregate_mask>	is the network or subnet mask in the form of IP address that the aggregate policy represents.
<aggregate_net>	is the network or subnet range in the form of IP address that the aggregate policy represents.
<area_id>	is the OSPF area ID in the format of an IP address.
<lsdb_type>	is the type of the LSAs that the aggregate policy represents.
<vr_name>	is the name of the virtual router.

Procedure job aid

Aggregate policy component hierarchy



Configuring a stub area

Configure a stub area to reduce the number of link state advertisements (LSAs) received from outside of the area. It helps reduce the memory requirements for the stub area's internal routers.

Prerequisites

- The attribute *importAsExtern* of the valid *AreaEntry* has to be set to a value other than *importExternal*.

Procedure steps

Step **Action**

1 Add a *StubAreaEntry* as a subcomponent of the Ospf component.

```
add Vr/<vr_name> Ip Ospf StubAreaEntry/<area_id>,<tos>
```

The *StubAreaEntry* address must have the same area ID as the *AreaEntry* component that already exists. See Configuring an OSPF area (page 133).

2 Set the *importAsExtern* attribute.

```
set Vr/<vr_name> Ip Ospf AreaEntry/<area_id>
importAsExtern <import_extern>
```

3 Set the *areaSummary* attribute.

```
Vr/<vr_name> Ip Ospf AreaEntry/<area_id> areaSummary
<area_summary>
```

4 Set the *metricType* attribute.

```
set Vr/<vr_name> Ip Ospf StubAreaEntry/<area_id>,<tos>
metricType <metric_type>
```

The attribute *metric_type* must be set to *ospfMetric* if and only the attribute *import_extern* for the corresponding *AreaEntry* is set to *importNoExternal*.

5 Set the stub area metric if you want to change the default metric value advertised in the default LSAs.

```
set Vr/<vr_name> Ip Ospf StubAreaEntry/<area_id>,<tos>
metric <metric_value>
```

6 Change the *advertiseDefault* attribute if you want to advertise the default routing information or not.

```
set Vr/<vr_name> Ip Ospf StubAreaEntry/<area_id>,<tos>
advertiseDefault <advertise_decision>
```

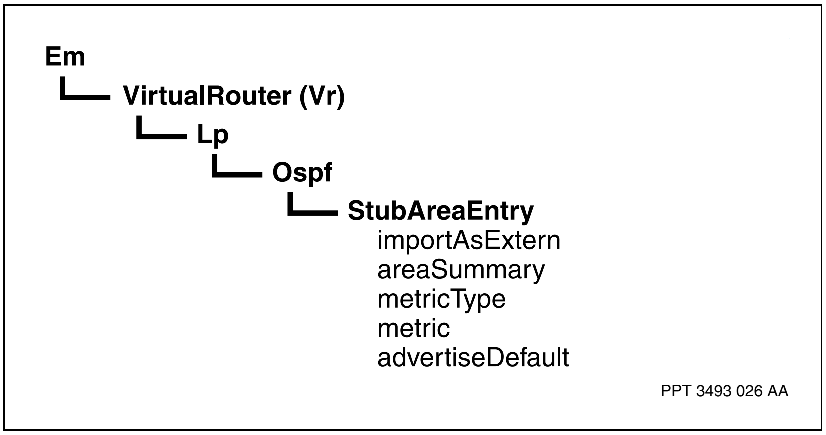
--End--

Variable definitions

Variable	Value
<advertise_decision>	is the value indicating whether to advertise default LSAs or not.
<area_summary>	is the value that controls the import of summary LSAs into stub areas. If you want to configure a stub area, set this attribute to *noAreaSummary*. If you want to configure a summary stub area, set this attribute to *sendAreaSummary*.
<area_id>	is the OSPF area ID in the format of an IP address.
<import_extern>	is the type of service associated with the metric. If you want to configure a stub area, set this attribute to *importNoExternal*. If you want to configure a not-so-stubby-area (NSSA), set this attribute to *importNssa*.
<metric_type>	is the metric type to be advertised in default LSAs.
<metric_value>	is the metric value to be advertised in default LSAs.
<tos>	is the type of service associated with the metric.
<vr_name>	is the name of the virtual router.

Procedure job aidProcedure job aid

Stub area component hierarchy



VR BGP-4 configuration

Configure the virtual router (VR) border gateway protocol 4 (BGP-4) to exchange routing information within a network or between networks using BGP-4, or to populate OSPF or RIP information through an IP tunnel.

Prerequisites to VR BGP-4 configuration

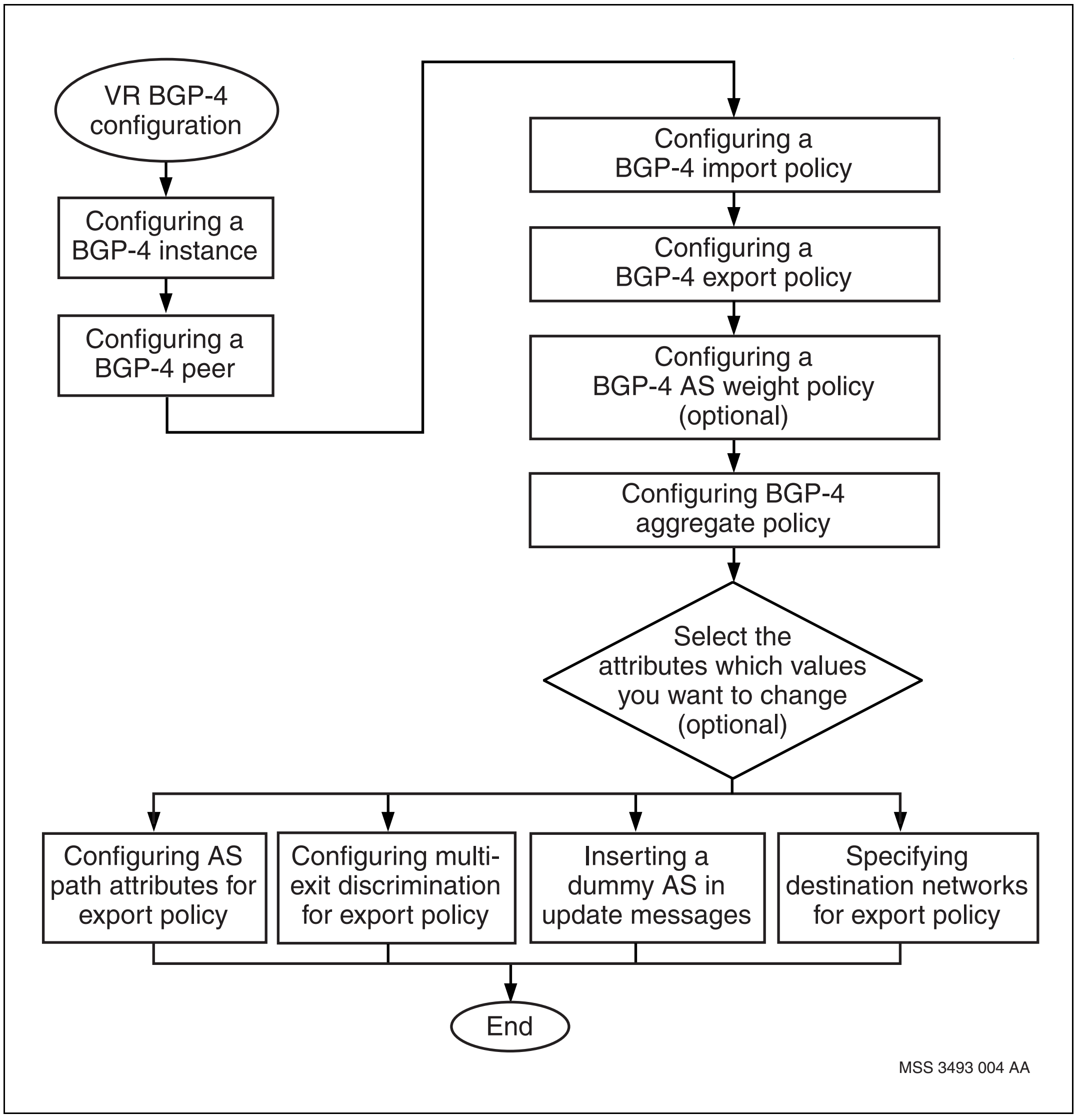
- Install a PM2 function processor and CP2 control processor in Nortel Networks Multiservice Switch nodes. Multiservice Switch nodes support BGP-4 on these cards only. For more information about function processors, see NN10600-551 *Nortel Networks Multiservice Switch 7400/15000/20000 FP Configuration Reference*. For more information about control processors, see NN10600-170 *Nortel Networks Multiservice Switch 7400 Hardware Description* and NN10600-120 *Nortel Networks Multiservice Switch 15000/20000 Hardware Description*.

- Configure one VR for each BGP-4 instance that you intend to configure. You can create only one BGP-4 instance for each VR on the node. See Adding a virtual router (page 80).

VR BGP-4 configuration procedures

This task flow shows you the sequence of procedures you perform to configure BGP-4. To link to any procedure, go to VR BGP-4 configuration procedure navigation (page 155).

VR BGP-4 configuration procedures


VR BGP-4 configuration
Configuring a BGP-4 instance
Configuring a BGP-4 peer
Configuring a BGP-4 import policy
Configuring a BGP-4 export policy
Configuring a BGP-4 AS weight policy (optional)
Configuring BGP-4 aggregate policy
Select the attributes which values you want to change (optional)
Configuring AS path attributes for export policy
Configuring multi-exit discrimination for export policy
Inserting a dummy AS in update messages
Specifying destination networks for export policy
End
MSS 3493 004 AA

VR BGP-4 configuration procedure navigation

- Configuring a BGP-4 instance (page 157)
- Configuring a BGP-4 peer (page 159)
- Configuring BGP-4 import policy (page 165)
- Configuring BGP-4 export policy (page 170)
- Configuring BGP-4 AS weight policy (page 174)
- Configuring BGP-4 aggregate policy (page 175)
- Configuring AS path attributes for export policy (page 178)
- Configuring multi-exit discrimination for export policy (page 180)

- Inserting a dummy AS in update messages (page 181)
- Specifying destination networks for export policy (page 182)

Configuring a BGP-4 instance

Configure a BGP-4 instance if you need to use BGP-4 to communicate between the Nortel Networks Multiservice Switch node and your network, or to populate OSPF or RIP information through an IP tunnel.

Procedure steps

Step **Action**

1 Create an instance of the BGP-4 protocol.

```
add Vr/<vr_name> Ip Bgp
```

2 Specify the autonomous system (AS) number of the BGP-4 instance.

```
set Vr/<vr_name> Ip Bgp localAs <asNo>
```

3 Specify the router identifier for the BGP-4 instance.

```
set Vr/<vr_name> Ip Bgp bgpIdentifier <ipAddress>
```

4 Specify the local preference for routes received from external peers, if you do not want to use the default value.

```
set Vr/<vr_name> Ip Bgp locPrf <loc_pref>
```

5 Specify the MED value for routes sent to external peers, if you do not want to use the default value.

```
set Vr/<vr_name> Ip Bgp med <metric>
```

6 Configure BGP-4 as a route reflector within the AS, if you do not want to use the default value.

The default cluster identifier, as specified in the *routeReflectorCluster* attribute, has the same IP address as the router ID.

```
set Vr/<vr_name> Ip Bgp rr <true_false>
```

7 If required, change the default for the alarm generator, which specifies what BGP alarms are generated.

```
set Vr/<vr_name> Ip Bgp alarmGeneration <alarm>
```

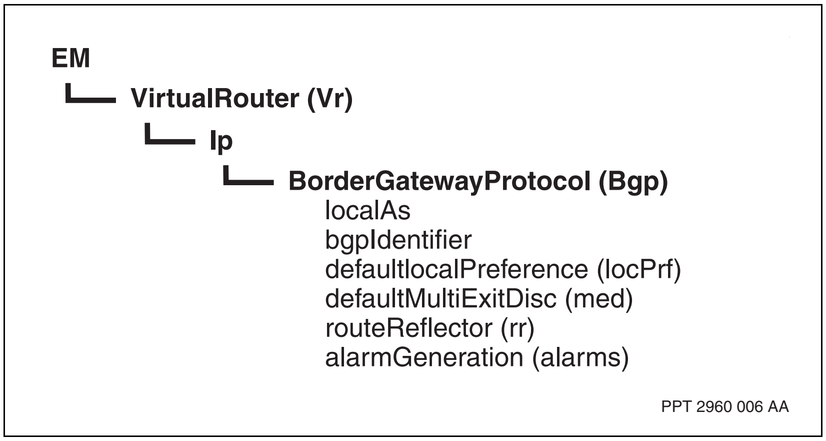
--End--

Variable definitions

Variable	Value
<alarm>	is the level of alarm generation for this instance of BGP.
<asNo>	is the autonomous system (AS) to which the BGP-4 instance belongs.
<ipAddress>	is the 32-bit IP address that identifies the BGP-4 router.
<loc_pref>	is the local preference.
<metric>	is the MED value.
<true_false>	specifies whether the BGP-4 instance behaves as a route reflector within the AS.
<vr_name>	is the name of the virtual router.

Procedure job aid

BGP-4 instance component hierarchy



Configuring a BGP-4 peer

Configure BGP-4 peers to exchange routing information about reachable destinations in different autonomous systems (AS). The BGP-4 peers use this routing information to construct a map of AS connectivity that allows them to eliminate routing loops and enforce policy decisions at the AS level.

Optionally, configure MD5 authentication on a BGP peer connection to provide protection of BGP neighbor relationships. Change an MD5 key on an authenticated BGP peer using the procedure Changing an MD5 key on an authenticated BGP-4 peer (page 163).

Prerequisites

- If configuring MD5 authentication, it is recommended that different MD5 key values be used for each BGP-4 peer.

Attention: A longer MD5 key will be more difficult to break, however, the longer the key used, the greater the impact on performance. A key length of between 12 and 24 characters should be sufficiently secure without having too great an impact on performance.

Procedure steps

Step	Action

1 Create an instance of a BGP-4 peer under the BGP-4 instance.

```
add Vr/<vr_name> Ip Bgp Peer/<peer>
```

The component *Vr Ip Bgp Peer Desc* is automatically created.

2 Specify the AS number for the BGP-4 peer.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc peerAs <asNo>
```

3 Specify the local IP address for the BGP-4 peer.

If you set the *localAddressConfigured* attribute as the local IP address used for BGP peering, you can use either a physical interface address or a virtual interface address (for example, an always-up IP address). In the case of EBGP peering with the *multiHopEbgp* attribute disabled, you must use the address of the directly connected interface to ensure proper exchange of routing information.

If you are using an always-up IP interface, the local IP address of the BGP-4 peer must match the IP address of the virtual router associated virtual media logical interface. If you are using an always-up interface, make sure the *Vm If mode* attribute is set to alwaysUpInterface. See Virtual media configuration (page 237).

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc lac <lac_addr>
```

4 Specify the frequency of keep alive message retransmissions from the BGP-4 peer.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc kac <kac_timer>
```

5 Specify the maximum length of time between BGP-4 keep alive messages from the BGP-4 peer before the BGP-4 instance considers the connection down.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc htc <htc_timer>
```

6 Specify the minimum length of time TCP waits before re-attempting to establish a BGP-4 connection.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc connectRetry
<con_tmr>
```

7 Specify the minimum length of time the BGP-4 peer waits before sending route updates to a neighbor AS.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc minAsOrig
<orig_tmr>
```

8 Specify the length of time after which BGP-4 can re-advertise route information to peers in other ASs.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc minRouteAdv
<adv_tmr>
```

BGP-4 ignores the *minAsOrigTime* and *minRouteAdvTime* attributes for routes that are withdrawn.

9 Specify whether the BGP-4 peer is a route reflector client.

```
set Vr/<vr_name> Ip Bgp peer/<peer> Desc isRrClient
<true_false>
```

10 Enable dynamic default aggregation (DDA) for routes learned from the EBGP peer if you want to prioritize dynamic default routes.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc diaMed <metric>
```

11 Specify whether the private AS number is removed from the AS path attribute of routes sent to the EBGP peer.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc remPrivate
<rem_priv>
```

12 Advertise the route availability to the BGP-4 peer.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc nhs <nhs>
```

13 Enable the multihop EBGP route distribution on a peer if you want to allow the distribution of routes across multiple hops between BGP peers that belong to different autonomous systems (AS).

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc multiHopEbgp
<multihop_ebgp>
```

14 Optionally, enable MD5 authentication by setting the MD5 key value under the peer’s *Descriptor*.

```
set Vr/<vr_id> Ip Bgp Peer/<ip_addr> Desc md5Key
<ASCII_string>
```

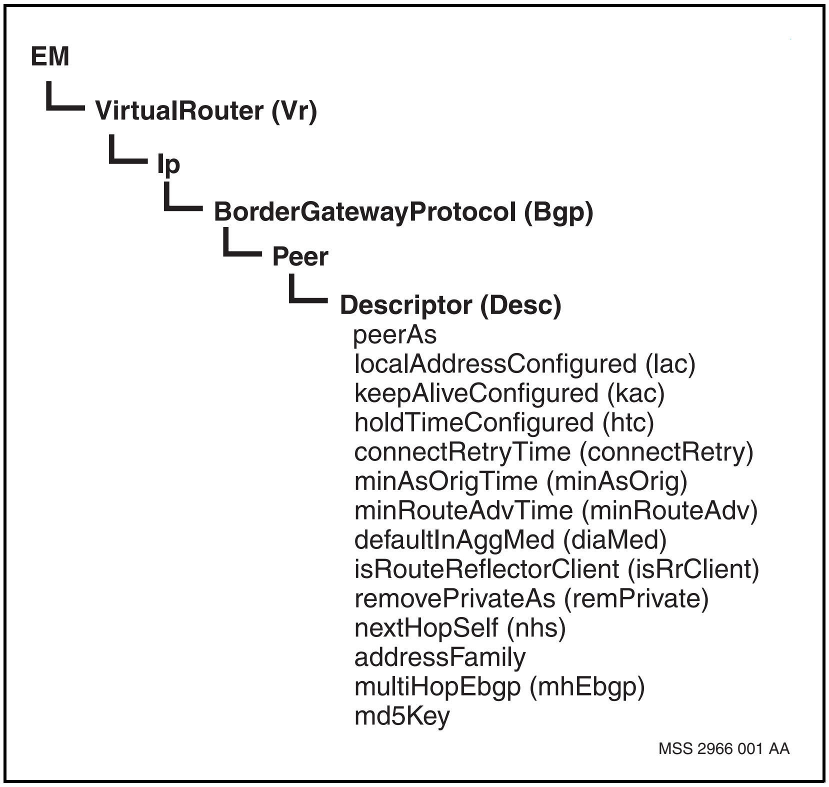
--End--

Variable definitions

Variable	Value
<adv_tmr>	is the time interval in seconds.
<ASCII_string>	is 1-255 ASCII characters.
<asNo>	is the AS to which the BGP-4 peer belongs.
<con_tmr>	is the time interval in seconds.
<htc_timer>	is the time interval in seconds.
<kac_timer>	is the time interval in seconds. When you set this attribute to 0, BGP-4 does not send keep alive messages.
<lac_addr>	is the local IP address for this BGP-4 peer connection.
<metric>	is the MED path attribute value assigned to the DDA route. Specifying a MED metric other than the default value enables DDA for the BGP-4 peer. BGP-4 sets the MED path attribute of the default route to the value you configure in the *defaultInAggMed* (*diaMed*) attribute.
<multihop_ebgp>	is identifying whether the multihop EBGP route distribution capability is disabled or enabled on a EBGP peer. If the *multiHopEbgp* attribute is enabled, then the *localAs* attribute and *peerAs* attribute cannot be the same.
<nhs>	indicates if the next-hop-self is enabled or disabled when a route is sent to the peer. For details on the syntax for this attribute, see NN10600-060 *Nortel Networks Multiservice Switch 7400/15000/20000 Component Reference*.
<orig_tmr>	is the time interval in seconds.
<peer>	is the IP address of the BGP-4 peer. To use an always-up IP interface, the IP address of the BGP-4 peer must be the same as the associated virtual media logical interface. See Configuring an always-up interface (page 241).
<rem_priv>	indicates whether the private AS number removal is enabled or disabled when the route is advertised to another AS.
<true_false>	specifies whether the BGP-4 peer is a route reflector client. You must configure the BGP-4 instance as a route reflector if you set the *isRouteReflectorClient* attribute to true.
<vr_name>	is the name of the virtual router.

Procedure job aid

BGP-4 peer component hierarchy


EM
VirtualRouter (Vr)
Ip
BorderGatewayProtocol (Bgp)
Peer
Descriptor (Desc)
peerAs
localAddressConfigured (lac)
keepAliveConfigured (kac)
holdTimeConfigured (htc)
connectRetryTime (connectRetry)
minAsOrigTime (minAsOrig)
minRouteAdvTime (minRouteAdv)
defaultInAggMed (diaMed)
isRouteReflectorClient (isRrClient)
removePrivateAs (remPrivate)
nextHopSelf (nhs)
addressFamily
multiHopEbgp (mhEbgp)
md5Key
MSS 2966 001 AA

Changing an MD5 key on an authenticated BGP-4 peer

Transition MD5 keys without having to terminate the BGP-4 neighbor session for which the key is being changed.

Prerequisites

- It is recommended that different MD5 key values be used for each BGP-4 peer.
- When changing keys on a BGP session between a Multiservice Switch node and a non-Multiservice Switch, the key should be changed on the Multiservice Switch node, to minimize the shortage time.

Attention: A longer MD5 key will be more difficult to break, however, the longer the key used, the greater the impact on performance. A key length of between 12 and 24 characters should be sufficiently secure without having too great an impact on performance.

Procedure steps

Step **Action**

1 Set the *keyTransitionDelay* attribute on the local and remote router, for all BGP peers.

```
set Vr/<vr_id> Ip Bgp keyTransitionDelay <n>
```

2 Alternatively, set the *keyTransitionDelay* attribute on the local and remote routers, for a specific peer's *Descriptor*.

```
set Vr/<vr_id> Ip Bgp Peer/<ip_addr> Desc
keyTransitionDelay <n>
```

3 Set the new MD5 key value on the local and remote router.

```
set Vr/<vr_id> Ip Bgp Peer/<ip_addr> Desc md5Key
<ASCII_string>
```

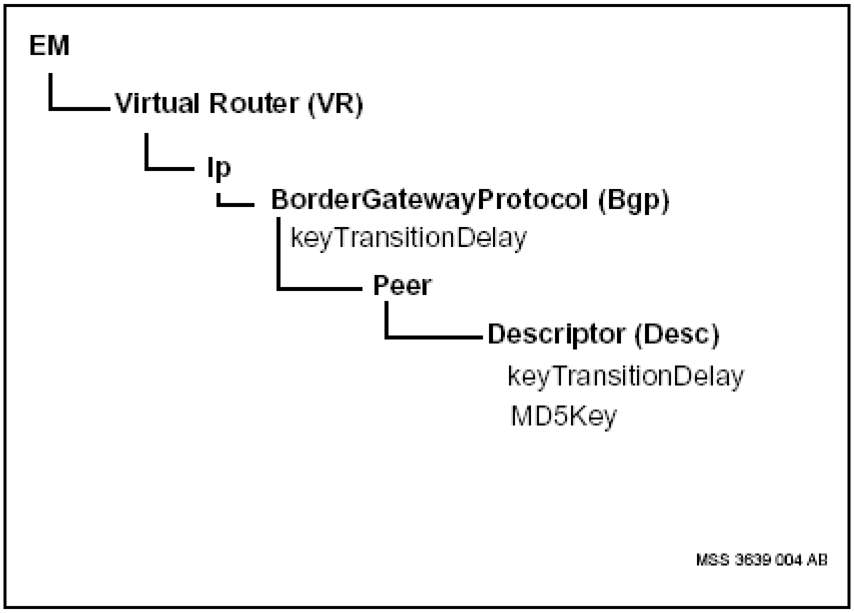
--End--

Variable definitions

Variable	Value
<ASCII_string>	is 1-255 ASCII characters.
<ip_addr>	is the IP address.
<n>	is the amount of time the operator has to provision both the local and remote ends with the new key, without losing service. for the *Bgp* component, the value is 1 to 20160 (minutes), default is 10 minutes. for the *Descriptor* component, the value is “sameAsBgp, 1 to 20160 (minutes)”, default is “sameAsBgp”.
<vr_id>	is the virtual router id.

Procedure job aid

BGP-4 component hierarchy for changing an MD5 key



Configuring BGP-4 import policy

Configure BGP-4 import policy to specify what routing information BGP-4 allows into or blocks from the IP routing database. Nortel Networks Multiservice Switch node BGP-4 import policies can be customized through configurable attributes.

Procedure steps

Step **Action**

1 Create an instance of an import policy. The instance number only identifies the policy and is not related to policy preference.

```
add Vr/<vr_name> Ip Bgp Import/<im_plcy>
```

2 Specify the AS number of the BGP-4 peer from which routes are learned.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> peerAs <asNo>
```

3 Specify the IP address of the BGP-4 peer from which routes are learned.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> peerIpAddress
<addr>
```

4 Specify the AS that originated the routes learned over the BGP-4 peer.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> originAs
<or_asNo>
```

5 Specify a regular expression that identifies AS paths from which BGP-4 accepts route updates if you do not want to use the default value.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> asExpr
<path_expr>
```

6 Specify a regular expression that identifies community paths from which BGP-4 accepts route updates if you do not want to use a default value.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> comExpr
<com_expr>
```

7 If you have configured an AS path and community path expression for the import policy, specify a preference for the policy.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> exprPref <pref>
```

When the expression attributes of two import policies match the same AS or community, BGP-4 uses the preference metric to select a preferred policy. A higher value indicates a higher preference.

8 Specify the protocol that originated the routes learned over the BGP-4 peer.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> originProtocol
<protocol>
```

9 Specify whether BGP-4 uses or ignores information in routing updates if it meets the criteria specified in the policy:

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> usage <flag>
```

10 If required, change the route preference attribute *defaultIbgpRtePref* for BGP internal routes.

```
set Vr/<vr_name> Ip Bgp defaultIbgpRtePref
<ibgp_route_pref>
```

11 If required, override the route preference by changing attribute *ibgpRtePref* for BGP internal routes.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy>
ibgpRtePreference <ibgp_override>
```

12 If required, change the route preference attribute *defaultEbgpRtePref* for BGP external routes.

```
set Vr/<vr_name> Ip Bgp defaultEbgpRtePref
<ebgp_route_pref>
```

13 If required, override the route preference by changing attribute *ebgpRtePref* for BGP external routes.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy>
ebgpRtePreference <override>
```

14 Specify a preference for routes that match the import policy:

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> locPrf
<loc_pref>
```

If you do not set this value, BGP-4 applies the local preference configured under the BGP-4 instance in the *defaultLocalPreference* attribute to routes that meet the import policy criteria.

15 Specify the community number that BGP-4 inserts in the community path attribute for routes that match the criteria of this import policy.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> appCom <com_no>
```

16 Specify a destination network for this BGP-4 import policy if you want to restrict the destination networks.

```
add Vr/<vr_name> Ip Bgp Import/<imp_plcy> Net/<net_no>
```

You must create an instance of the *Network* (*Net*) subcomponent for each destination network associated with the import policy. If you do not specify any destination networks, BGP-4 applies the import policy to all networks.

17 Specify the network prefix for the destination network associated with this BGP-4 import policy.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> Net/<net_no>
prefix <prefix>
```

18 Specify the network prefix length for the destination network associated with this BGP-4 import policy.

```
set Vr/<vr_name> Ip Bgp Import/<imp_plcy> Net/<net_no>
length <length>
```

19 Specify the address family of the import policy.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> addressFamily
<address_family> ipv4Unicast
```

MBGP uses the same import BGP policy at the VCG level as BGP. For information about MBGP, see the chapter “Configuring multi-protocol BGP,” in NN10600-582 *Nortel Networks Multiservice Switch 7400/15000/20000 VPN Configuration Management*.

--End--

Variable definitions

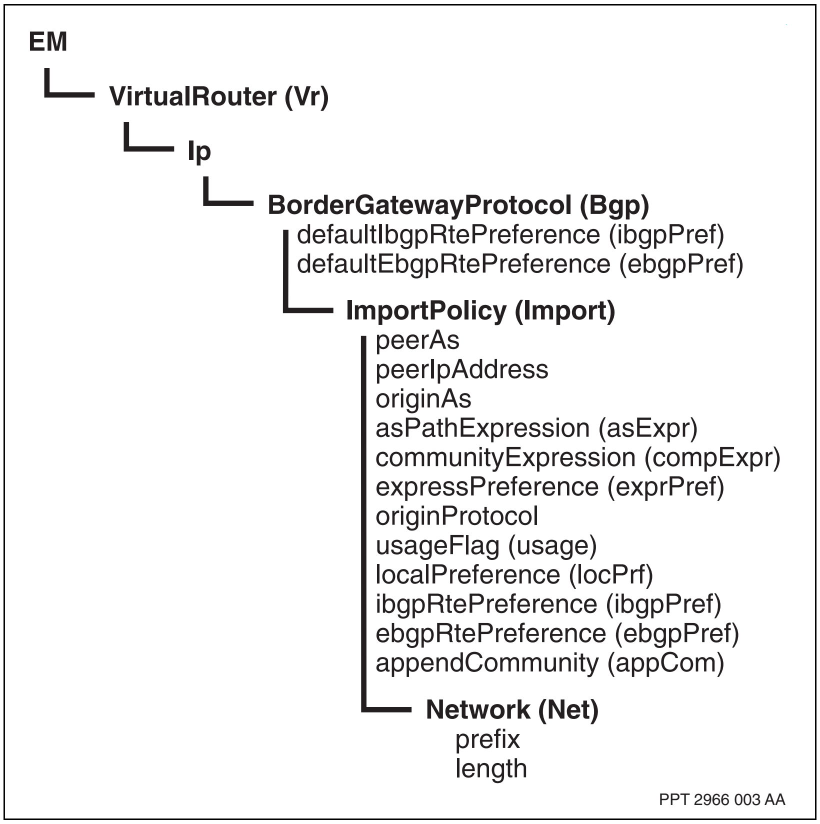
Variable	Value
<addr>	is the IP address of the BGP-4 peer. If you set this value to 0.0.0.0, the policy matches any IP address.
<address_family>	defines the multi-protocol extension address family of the routes to which the policy applies. The addressFamily attribute is used to distinguish the policy for MBGP applications compared to BGP applications. A policy may be selected to apply only to MBGP (mbgpVpn), only to BGP (ipv4Unicast), or both. The default is both.
<asNo>	is the AS to which the BGP-4 peer belongs. If you set this value to 0, the policy matches any AS number.
<com_expr>	a regular expression identifying community paths to match.
<com_no>	is the community number added to the community path attribute.
<ebgp_override>	is the override route preference. Attribute default is sameAsBgp, which means use the value of attribute *defaultEbgpRtePref* for the route preference. To prefer BGP external routes over OSPF internal routes, the recommended setting for *ebgpRtePreference* is 6.
<ebgp_route_pref>	is the route preference.
<flag>	indicates whether BGP-4 uses or ignores received routing updates.
<ibgp_override>	is the override route preference. Attribute default is sameAsBgp, which means use the value of attribute *defaultIbgpRtePref* for the route preference. To prefer BGP internal routes over OSPF internal routes, the recommended setting for *ibgpRtePreference* is 6.
<ibgp_route_pref>	is the route preference.
<im_plcy>	is the instance number of the import policy.
<length>	specifies the length of the network prefix.
<loc_pref>	is the relative preference for routes that match the import policy's criteria
<net_no>	identifies the destination network associated with the import policy.
<or_asNo>	is the number of the AS that originated the learned routes. If you set this value to 0, the policy matches any AS number.
<path_expr>	a regular expression identifying AS paths to match.

(1 of 2)

Variable	Value
<pref>	is the relative preference of a path-based policy.
<prefix>	is the network prefix, in the form of an IP address.
<protocol>	identifies the protocol that originated the routes.
<vr_name>	is the name of the virtual router.
	(2 of 2)

Procedure job aid

BGP-4 import policy component hierarchy


EM
VirtualRouter (Vr)
Ip
BorderGatewayProtocol (Bgp)
defaultIbgpRtePreference (ibgpPref)
defaultEbgpRtePreference (ebgpPref)
ImportPolicy (Import)
peerAs
peerIpAddress
originAs
asPathExpression (asExpr)
communityExpression (compExpr)
expressPreference (exprPref)
originProtocol
usageFlag (usage)
localPreference (locPrf)
ibgpRtePreference (ibgpPref)
ebgpRtePreference (ebgpPref)
appendCommunity (appCom)
Network (Net)
prefix
length
PPT 2966 003 AA

Configuring BGP-4 export policy

Configure BGP-4 export policy to specify what IP routing information BGP-4 distributes to other BGP-4 peers. This is also the only way to allow routing information from other protocols to be distributed into BGP-4. Nortel Networks Multiservice Switch node BGP-4 export policies can be customized through configurable attributes.

Procedure steps

Step **Action**

1 Create an instance of an export policy. The instance number only identifies the policy and is not related to policy preference.

```
add Vr/<vr_name> Ip Bgp Export/<ex_plcy>
```

2 Specify the AS number of the BGP-4 peer to which routes are advertised.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> peerAs <asNo>
```

3 Specify the IP address of the peer AS to which routes are advertised.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> peerIpAddress
<addr>
```

4 Specify the protocol to which the export policy applies.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> protocol
<prtcl>
```

5 If you specified EGP in the *protocol* attribute, specify the number of the EGP AS to which the export policy applies.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> egpAs <egp_id>
```

If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or EGP.

6 If you specified BGP in the *protocol* attribute, specify the number of the BGP-4 AS to which the export policy applies.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> bgpAs <bgp_id>
```

If you set this attribute to a non-zero value, the *protocol* attribute must be set to all, bgpInternal or bgpExternal.

7 If you specified OSPF in the *protocol* attribute, specify the OSPF tag for the OSPF routes to which the export policy applies.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> ospfTag
<ospf_tag>
```

If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or ospfExternal.

8 If you specified RIP in the *protocol* attribute, specify the local RIP interface from which RIP routes were learned. If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or RIP.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> ripIf <rip_if>
```

The IP address of the RIP interface corresponds to the IP address of a *IplogicalIf* component configured under the IP port. For more information, see Enabling IpPort on a protocol port (page 101).

9 If you specified RIP in the *protocol* attribute, specify the RIP neighbor from which RIP routes were learned.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> ripNbr
<rip_nbr>
```

If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or RIP.

10 Specify a preference for routes that match the export policy.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy) locPref
<loc_pref>
```

11 Specify whether BGP-4 advertises or blocks routes that meet the export policy’s criteria.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> advertise
<send_block>
```

12 Specify the address family of the export policy.

```
set Vr/<vr_name> Ip Bgp Export/<im_plcy> addressFamily
<address_family> ipv4Unicast
```

MBGP uses the same export BGP policy at the VCG level as BGP. For information about MBGP, see the chapter “Configuring multi-protocol BGP,” in NN10600-582 *Nortel Networks Multiservice Switch 7400/15000/20000 VPN Configuration Management*.

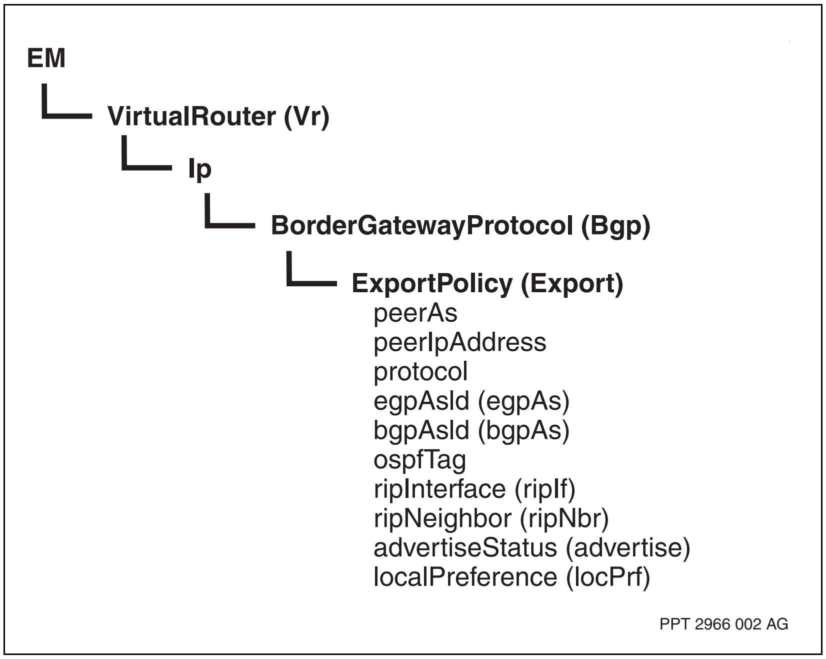
--End--

Variable definitions

Variable	Value
<addr>	is the IP address of the BGP-4 peer. If you set this value to 0.0.0.0, the policy matches any IP address.
<address_family>	defines the multi-protocol extension address family of the routes to which the policy applies. The addressFamily attribute is used to distinguish the policy for MBGP applications compared to BGP applications. A policy may be selected to apply only to MBGP (mbgpVpn), only to BGP (ipv4Unicast), or both. The default is both.
<asNo>	identifies the AS to which the BGP-4 peer belongs. If you set this value to 0, BGP-4 advertises to all peer ASs.
<bgp_id>	is the number of the BGP AS.
<egp_id>	is the number of the EGP AS.
<ex_plcy>	is the instance number of the export policy.
<loc_pref>	is the preference for routes that match the criteria of the export policy.
<ospf_tag>	is the OSPF tag stored in OSPF external routes.
<prtcl>	identifies the protocol type for matching routes.
<rip_if>	is the IP address of the RIP interface from which RIP routes were learned.
<rip_nbr>	is the IP address of the RIP neighbor from which RIP routes were learned.
<send_block>	indicates BGP-4 behavior for routes that match the export policy.
<vr_name>	is the name of the virtual router.

Procedure job aid

BGP-4 export policy component hierarchy



Configuring BGP-4 AS weight policy

Configure BGP-4 autonomous systems (AS) weight policy to set a preference for one autonomous system and discriminate against other autonomous systems.

Procedure steps

Step **Action**

1 Create an instance of an AS weight policy. The instance number only identifies the policy and is not related to policy preference.

```
add Vr/<vr_name> Ip Bgp As/<as_no>
```

2 Specify the weight for the AS of the BGP-4 instance. The lowest weight is preferred in route selection.

```
set Vr/<vr_name> Ip Bgp As/<as_no> weight <as_wgt>
```

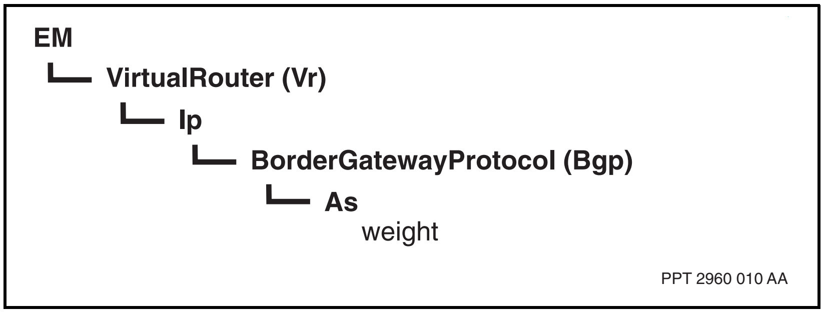
--End--

Variable definitions

Variable	Value
<as_no>	is the instance number of the AS.
<as_wgt>	is the weight assigned to the AS.
<vr_name>	is the name of the virtual router.

Procedure job aid

BGP-4 AS weight policy component hierarchy



Configuring BGP-4 aggregate policy

Configure BGP-4 aggregate policy to enable BGP-4 to combine the characteristics of different routes and advertise the combination as a single route. Aggregation reduces the data a BGP-4 speaker stores and exchanges with other BGP-4 speakers.

Procedure steps

Step	Action
1	Create an instance of an BGP-4 aggregate policy. `add Vr/<vr_name> Ip Bgp Aggregate/<aggr>`
2	Define a set of routes if you want to specify the routes that BGP-4 aggregates or advertises with the aggregated route policy. `add Vr/<vr_name> Ip Bgp Aggregate/<aggr> Net/<net_no>`
3	Specify the network prefix for the destination network associated with this BGP-4 aggregate policy. `set Vr/<vr_name> Ip Bgp Aggregate/<aggr> Net/<net_no> prefix <prefix>`
4	Specify the network prefix length for the destination network associated with this BGP-4 aggregate policy. `set Vr/<vr_name> Ip Bgp Aggregate/<aggr> Net/<net_no> length <length>`
5	Specify the protocol to which the aggregate policy applies: `set Vr/<vr_name> Ip Bgp Aggregate/<aggr> protocol <prtcl>`
6	If you specified EGP in the *protocol* attribute (see step 5), specify the number of the EGP AS to which the aggregate policy applies. `set Vr/<vr_name> Ip Bgp Aggregate/<aggr> egpAs <egp_id>`
7	If you specified BGP in the *protocol* attribute, specify the number of the BGP AS to which the aggregate policy applies. `set Vr/<vr_name> Ip Bgp Aggregate/<aggr> bgpAs <bgp_id>`
8	If you specified OSPF in the *protocol* attribute, specify the OSPF tag for the OSPF routes to which the aggregate policy applies. `set Vr/<vr_name> Ip Bgp Aggregate/<aggr> ospfTag <ospf_tag>`
9	If you specified RIP in the *protocol* attribute, specify the local RIP interface from which RIP routes were learned. `set Vr/<vr_name> Ip Bgp Aggregate/<aggr> ripIf <rip_if>`

10 Specify whether BGP-4 aggregates or advertises routes that meet the aggregate policy's criteria.

```
set Vr/<vr_name> Ip Bgp Aggregate/<aggr> action
<agg_adv>
```

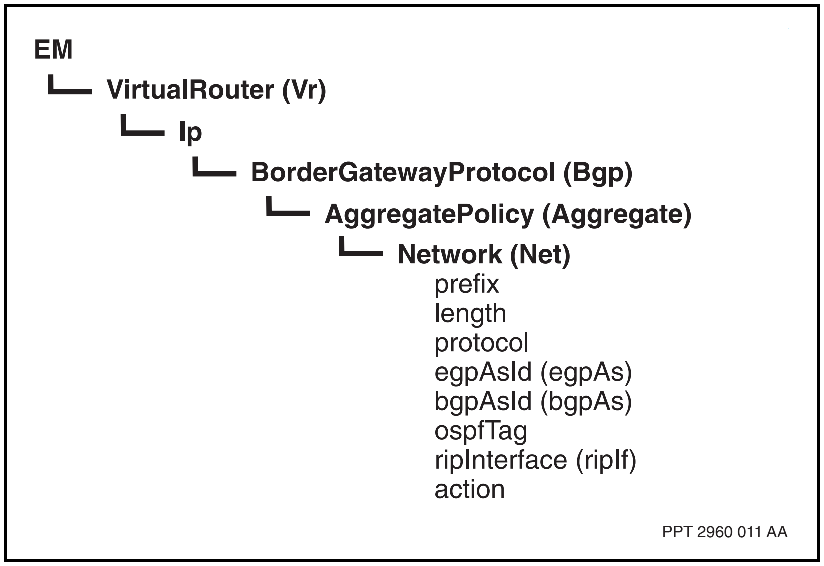
--End--

Variable definitions

Variable	Value
<agg_adv>	indicates BGP-4 behavior, aggregate or advertise, for routes that match the aggregate policy.
<aggr>	is the aggregate policy, in the format <prefix, length>.
<bgp_id>	is the number of the BGP AS. If you set this attribute to a non-zero value, the *protocol* attribute must be set to all, bgpInternal or bgpExternal.
<egp_id>	is the number of the EGP AS. If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or EGP.
<length>	is the length of the network prefix.
<net_no>	is the routes associated with the aggregate policy.
<ospf_tag>	is the OSPF tag stored in OSPF external routes. If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or ospfExternal.
<prefix>	is the network prefix, in the form of an IP address.
<prtcl>	is the protocol type for matching routes.
<rip_if>	is the IP address of the RIP interface from which RIP routes were learned. If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or RIP. The IP address of the RIP interface corresponds to the IP address of a *IpLogicalIf* component configured under the IP port. For more information, see Enabling IpPort on a protocol port (page 101).
<vr_name>	is the name of the virtual router.

Procedure job aid

BGP-4 aggregate policy component hierarchy



Configuring AS path attributes for export policy

Configure autonomous systems (AS) path attributes for export policy to provide detailed information about an advertised route. Routers can use path attribute information when making policy decisions.

Procedure steps

Step **Action**

1 Specify a regular expression that identifies AS paths to which BGP-4 advertises route updates if you do not want to use the default value.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> asExpr
<path_expr>
```

2 Specify a regular expression that identifies community paths to which BGP-4 advertises route updates if you do not want to use the default value.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> comExpr
<com_expr>
```

3 If you have configured an AS path and community path expression for the export policy, specify a preference for the policy.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> exprPref <pref>
```

When the expression attributes of two export policies match the same AS or community, BGP-4 uses the preference metric to select a preferred policy. A higher value indicates a higher preference.

4 Specify the community number that BGP-4 inserts in the community path attribute before advertising routes identified by this policy.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> sendCom
<com_no>
```

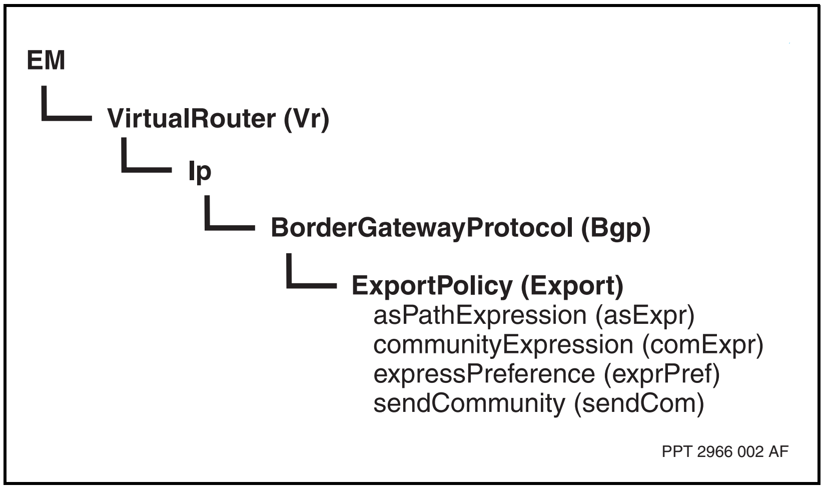
--End--

Variable definitions

Variable	Value
<com_expr>	a regular expression identifying community paths to match.
<com_no>	is the community number added to the community path attribute.
<path_expr>	a regular expression identifying AS paths to match.
<pref>	is the relative preference of a path-based policy.
<vr_name>	is the name of the virtual router.

Procedure job aid

AS path attributes for export policy component hierarchy


EM
VirtualRouter (Vr)
Ip
BorderGatewayProtocol (Bgp)
ExportPolicy (Export)
asPathExpression (asExpr)
communityExpression (comExpr)
expressPreference (exprPref)
sendCommunity (sendCom)
PPT 2966 002 AF

Configuring multi-exit discrimination for export policy

Configure multi-exit discrimination for export policy to include the preferred entry point to the autonomous system in updates to external peers.

Procedure steps

Step	Action
1	Specify the metric that BGP-4 uses for this export policy to discriminate between multiple exit points to an adjacent AS. `set Vr/<vr_name> Ip Bgp Export/<ex_plcy> med <med_value>`
2	Specify whether BGP-4 includes the MED value in updates to EBGP peers. `set Vr/<vr_name> Ip Bgp Export/<ex_plcy> sendMed <true_false>`

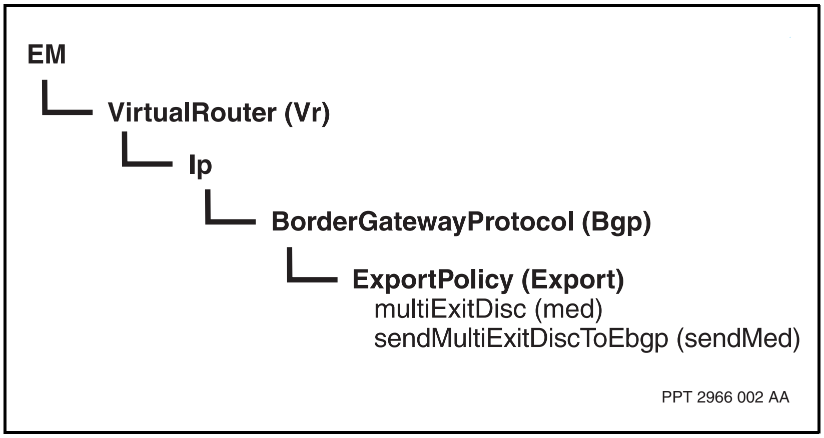
--End--

Variable definitions

Variable	Value
<med_value>	is the multi-exit discrimination (MED) metric.
<true_false>	indicates whether BGP-4 includes the MED metric in updates.
<vr_name>	is the name of the virtual router.

Procedure job aid

Multi-exit discrimination for export policy component hierarchy



Inserting a dummy AS in update messages

Insert a dummy autonomous systems (AS) in update messages to alter the AS path of the outgoing route.

Procedure steps

Step	Action
1	Specify a sequence of AS numbers to be inserted before the local AS number in the AS path attribute if you do not want to use the default value. **`set Vr/<vr_name> Ip Bgp Export/<ex_plcy> insertAs <AS_seq>`**

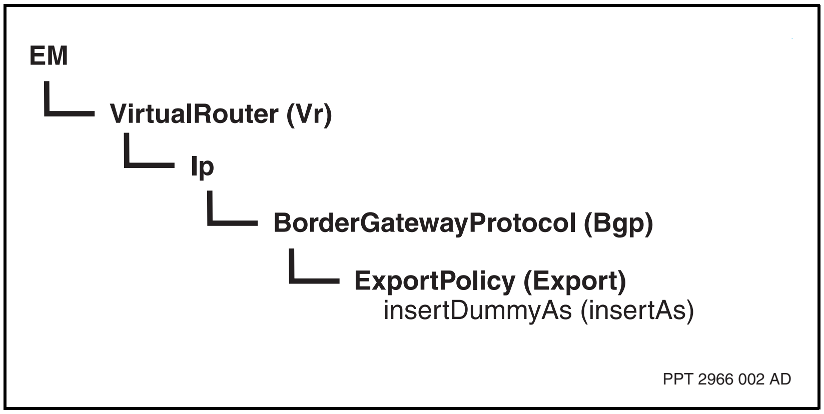
--End--

Variable definitions

Variable	Value
<AS_seq>	a sequence of AS numbers, separated by a period.
<vr_name>	is the name of the virtual router.

Procedure job aid

Inserting a dummy AS in update messages component hierarchy



Specifying destination networks for export policy

Specify destination networks for export policy to define which networks to export BGP-4 routing information to.

Procedure steps

Step **Action**

1 Specify a destination network for this BGP-4 export policy if you want to restrict the destination networks.

```
add Vr/<vr_name> Ip Bgp Export/<ex_plcy> Net/<net_no>
```

You must create an instance of the *Network* (*Net*) subcomponent for each destination network associated with the export policy. If you do not specify any destination networks, BGP-4 applies the export policy to all networks.

2 Specify the network prefix for the destination network associated with this BGP-4 export policy.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> Net/<net_no>
prefix <prefix>
```

3 Specify the network prefix length for the destination network associated with this BGP-4 export policy.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> Net/<net_no>
length <length>
```

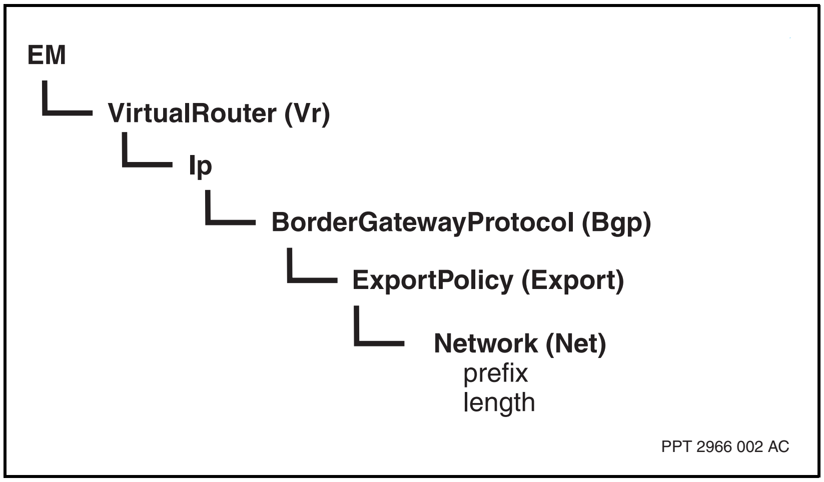
--End--

Variable definitions

Variable	Value
<length>	is the length of the network prefix.
<net_no>	is the destination network associated with the export policy.
<prefix>	is the network prefix, in the form of an IP address.
<vr_name>	is the name of the virtual router.

Procedure job aid

Specifying destination networks for export policy component hierarchy



Configuring VR static ARP

Configure VR static address resolution protocol (ARP) for IP traffic to control ARP-related attributes. ARP maps 32-bit IP addresses to physical hardware addresses.

To eliminate the need to ARP a particular host, configure the *HostEntry* subcomponent of the *Ip* component to define static ARP entries. Static host entries take precedence over dynamic entries learned through the ARP process.

You also configure static ARP host entries to ensure IP connectivity across the ATM network when you are using VC-based multiplexing for IP traffic.

Prerequisites

- Configure a media interface for IP traffic. See:
  - — Configuring an ATM MPE interface for IP traffic (page 24). Specify the VC-based multiplexing as the encapsulation type to be used on the ATM MPE interface.
  - — Adding a frame relay DTE interface (page 44)
- Associate the media interface with a connection. See
  - — Configuring an ATM PVC for an ATM MPE interface (page 26). Associate the ATM MPE interface with an ATM VCC.
  - — Configuring frame relay DTE with Framer (page 45)
  - — Configuring frame relay DTE with vFramer (page 47)
  - — Configuring a direct connection for a frame relay DTE interface (page 50)
- Create an IP port under the protocol port associated with the media interface. See Enabling IpPort on a protocol port (page 101).
- Do not configure static and dynamic ARP entries for the same IP address on the same VCC. You must configure both ends of the connection to be static. An ATM interface configured with a static ARP entry for an IP address will not respond to an inverse ARP request from a dynamic ARP.

for more ARP information, see NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Technology Fundamentals*.

Procedure steps

Step	Action
1	Create a static ARP entry for IP routing to ensure IP connectivity across the network. **`add Vr/<vr_name> Ip Arp HostEntry/<hostAddress>,<cos>`**
2	Configure the physical address of the host entry. If the media type is IP, use the command: **`set Vr/<vr_name> Ip Arp HostEntry/<hostAddress>,<cos> physAddress <MAC_address>`** If the media type is ATM or frame relay, use the command **`set Vr/<vr_name> Ip Arp HostEntry/<hostAddress>,<cos> permanentVirtualCircuitNumber <pvc_number>`**
3	If required, set the *maximumTransmissionUnit* (MTU) attribute. **`set Vr/<vr_name> Ip Arp HostEntry/<hostAddress>,<cos> maxTxUnit <mtu_size>`**
4	If required, set the *encapsulationType* attribute to a non-default value. The encapsulation default auto causes the correct encapsulation type to be chosen based on the media application (for example, Ethernet or IEEE802.3). **`set Vr/<vr_name> Ip Arp HostEntry/<hostAddress>,<cos> encap <encapsulation_type>`** This attribute can only be set for Ethernet media. All other media must use auto.

--End--

Variable definitions

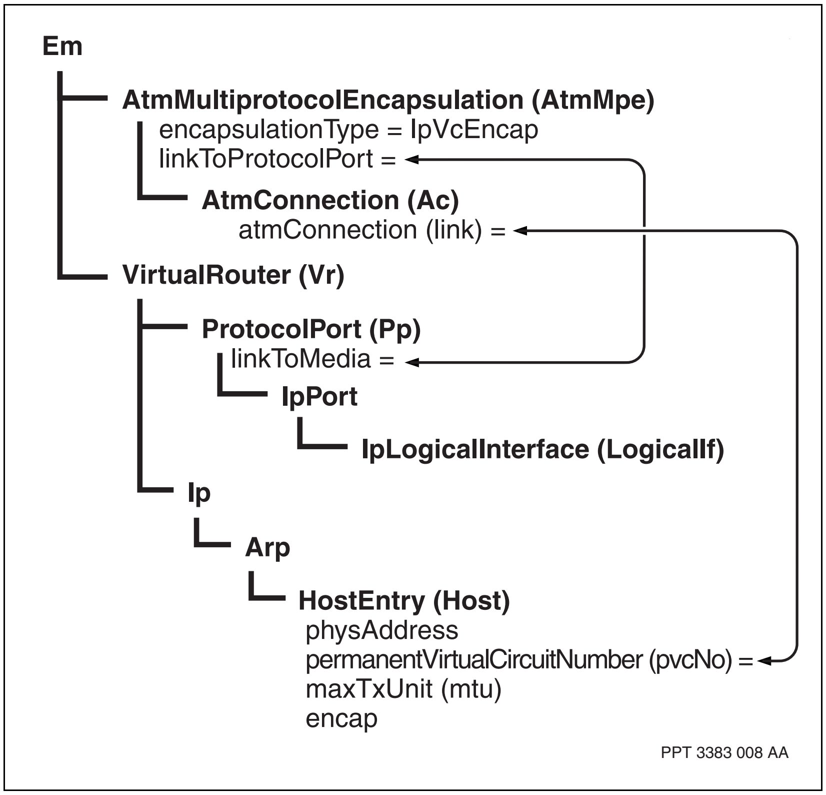
Variable	Value
<cos>	is the CoS index assigned to the packet. If you are using a VC-based media (FrDte, AtmMpe), the CoS index must be 0, 1, 2, or 3. For all other media types, use na.
<encapsulation_type>	one of auto, ieee8023, or Ethernet
<hostAddress>	is the IP address of the static host being defined

(1 of 2)

Variable	Value
<MAC_address>	is the 48-bit MAC address of the host being defined. It is formatted as zero to eight pairs of hex digits separated by dashes. The default address is 00-00-00-00-00-00-00-00.
<mtu_size>	is the size in bytes of the maximum transmission unit, or largest datagram, that the host can accept. The MTU must fall within the valid range for the media on which the host is located.
<pvc_number>	is the PVC for the static host entry. If the media type is frame relay, then this value is the frame relay data link connection identifier. If the media type is ATM, then this value is the instance number of the *AtmConnection* component on the ATM MPE interface.
<vr_name>	is the name of the virtual router.
(2 of 2)	

Procedure job aid

VR static ARP component hierarchy



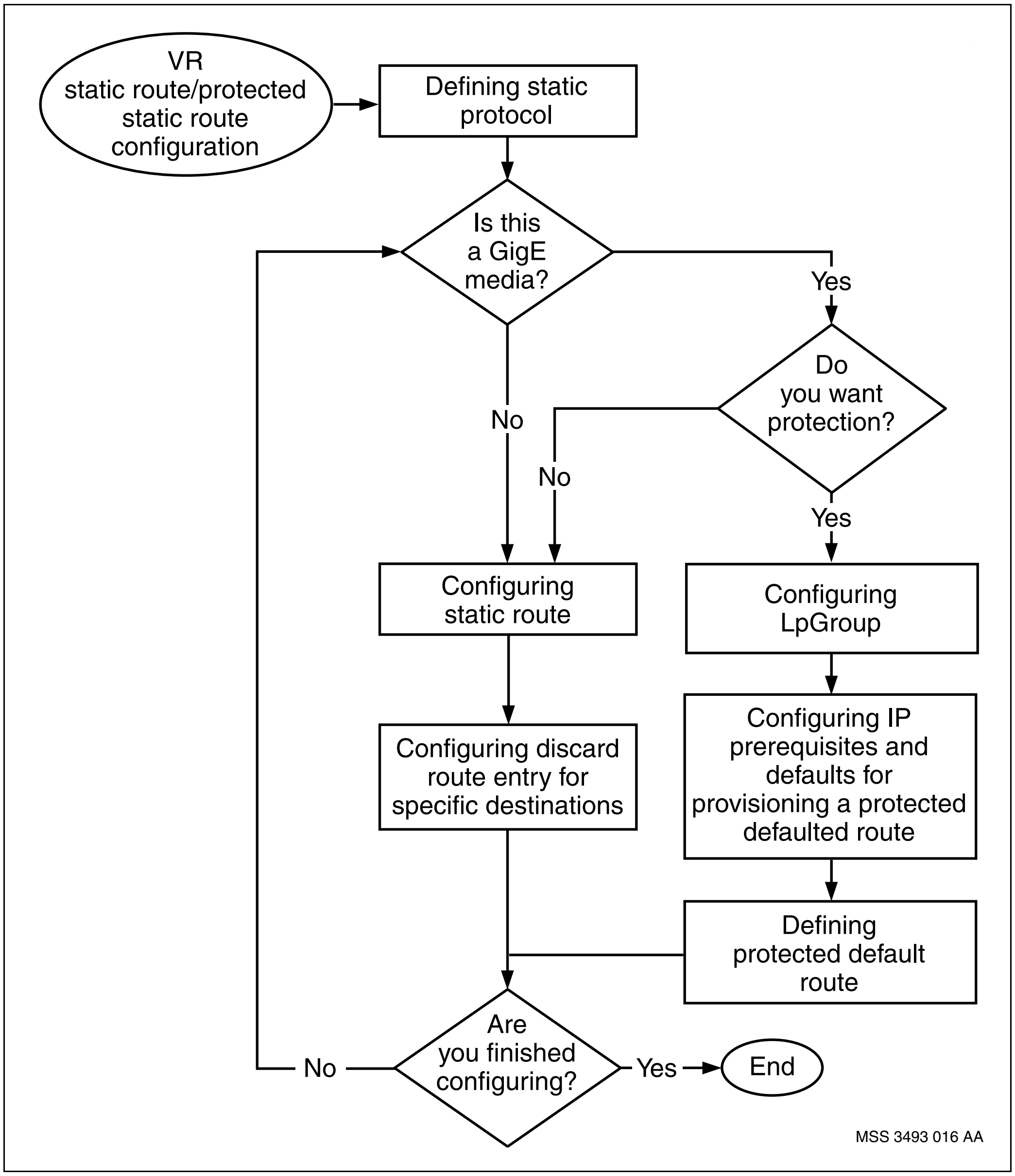
VR static route configuration

Configure VR static routes and a protected default route to allow Nortel Networks Multiservice Switch nodes to specifically identify routes to remote IP networks or hosts.

VR static route configuration

This task flow shows you the sequence of procedures to perform in order to configure static routes and a protected default route. To link to any procedure, go to VR static route configuration procedures (page 189).

VR static route configuration procedures


VR static route/protected static route configuration
Defining static protocol
Is this a GigE media?
Yes
Do you want protection?
No
No
Yes
Configuring static route
Configuring LpGroup
Configuring discard route entry for specific destinations
Configuring IP prerequisites and defaults for provisioning a protected defaulted route
Defining protected default route
Are you finished configuring?
No
Yes
End
MSS 3493 016 AA

VR static route configuration procedure navigation

- Defining static protocol (page 191)
- Configuring static route (page 192)
- Configuring discard route entry for specific destinations (page 194)
- Configuring LpGroup (page 195)
- Configuring IP prerequisites and defaults for provisioning a protected default route (page 197)

- Defining protected default route (page 199)

Defining static protocol

Define static protocol to add the *Static* component.

Procedure steps

Step	Action
1	Add a *Static* component as a subcomponent of the *Ip* component. **`add Vr/<vr_name> Ip Static`**

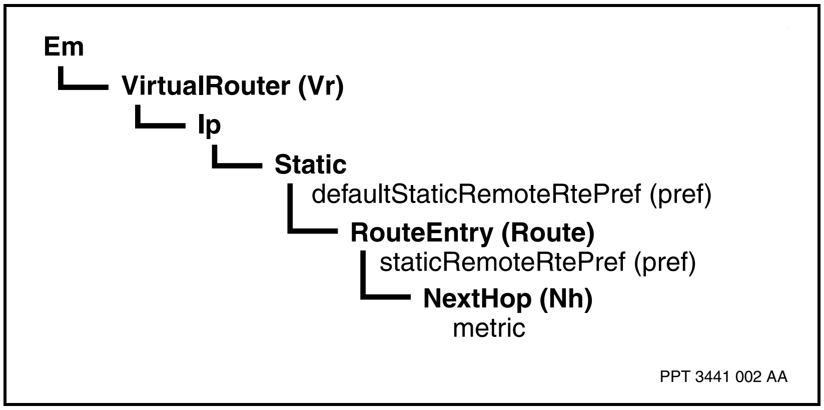
--End--

Variable definitions

Variable	Value
<vr_name>	is the name of the virtual router.

Procedure job aid

Static protocol component hierarchy



Configuring static route

Configure static routes to allow Nortel Networks Multiservice Switch system to identify specific routes to remote IP networks or hosts. The definition includes a destination address, address mask, and one or more next hop addresses (gateways).

Procedure steps

Step	Action
1	Add static routes to the route table. **add Vr/<vr_name> Ip Static RouteEntry/ <ipAddress>,<destMask>,<tos>** If ipAddress specifies a host, then provision destMask as 255.255.255.255. Provision locally attached hosts as *Arp HostEntry* components instead of a *Static* component entry.
2	Provision at least one *NextHop* component for each defined static route. The NextHop_ipAddress parameter must denote a locally provisioned logical interface, for example, the *NextHop_ipAddress* must belong to one of the subnets provisioned under an IpPort. **add Vr/<vr_name> Ip Static RouteEntry/ <ipAddress>,<destMask>,<tos> NextHop/ <nextHop_ipAddress>**
3	If required, set the metric for the route. **set Vr/<vr_name> Ip Static RouteEntry/ <ipAddress>,<destMask>,<tos> NextHop/<nextHop_ipAddress> metric <cost>**
4	If required, change the route preference attribute *defaultStaticRemoteRtePref* for static remote routes. **set Vr/<vr_name> Ip Static defaultStaticRemoteRtePref <route_pref>** When you change the attribute *defaultStaticRemoteRtePref*, services related to that route are disrupted during activation.
5	If required, override the route preference attribute staticRemoteRtePreference. **set Vr/<vr_name> Ip Static RouteEntry/ <ip_address>,<destMask>,<tos> staticRemoteRtePreference <override>**

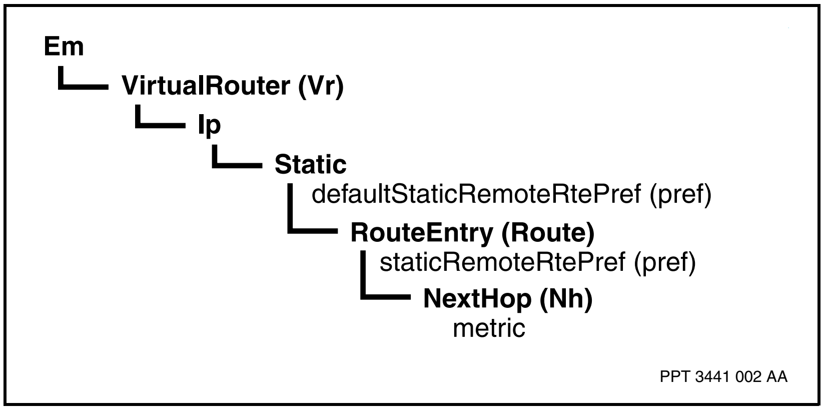
--End--

Variable definitions

Variable	Value
<cost>	is a relative metric value assigned to the static route.
<destMask>	is the subnetwork mask of the remote node used with the IP address.
<ipAddress>	is the IP address of the remote node. It can refer either to a specific node or to a network.
<nextHop_ipAddress>	is the IP address to use to reach the next router in the path to the destination.
<override>	is the override route preference. Attribute default is sameAsStatic, which means use the value of attribute *defaultStaticRemoteRtePref* for the route preference. To prefer static remote routes over OSPF internal routes, the recommended setting for *staticRemoteRtePreference* is 5.
<route_pref>	is the route preference.
<tos>	is the type of service. Currently, only the default value of 0 is supported.
<vr_name>	is the name of the virtual router.

Procedure job aid

Static route component hierarchy



Configuring discard route entry for specific destinations

Configure discard route entry for specific destinations to identify destination networks and nodes that do not receive packets through IP, and to discard packets addressed to these destinations.

Procedure steps

Step **Action**

1 Add a *DiscardRoute* component for each route that is not to receive packets through this router.

```
add Vr/<vr_name> Ip Static DiscardRouteEntry/
<destAddress>,<destMask>
```

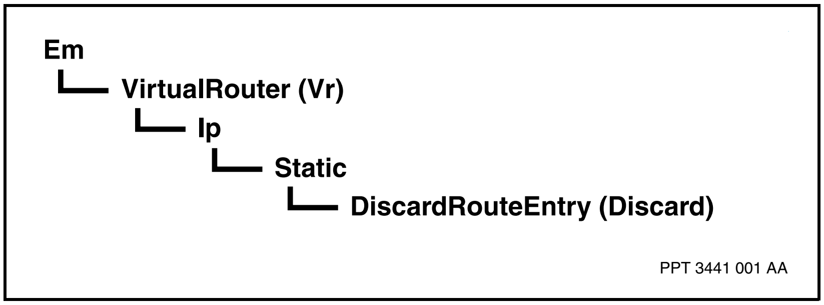
--End--

Variable definitions

Variable	Value
<destAddress>	is the IP address of the host or route whose packets are discarded.
<destMask>	is the subnetwork mask associated with the destination address.
<vr_name>	is the name of the virtual router.

Procedure job aid

Discard route entry for specific destinations component hierarchy



Configuring LpGroup

Configure the LpGroup to group 4pGe FPs together, and to specify the software migration behavior for each FP.

Procedure steps

Step **Action**

1 Add the *LpGroup* component.

```
add LpGroup/<lp_group>
```

2 Add the 4pGe LPs to the *LpGroup*.

```
add LpGroup/<lp_group> Lp/<lp_number>
```

3 Set the software migration behavior for each 4pGe LP in the *LpGroup*.

```
set LpGroup/<lp_group> Lp/<lp_number>
migrationBehaviour <mig_behaviour_value>
```

4 Optionally, repeat step 2 and step 3 to add one or two more LPs.

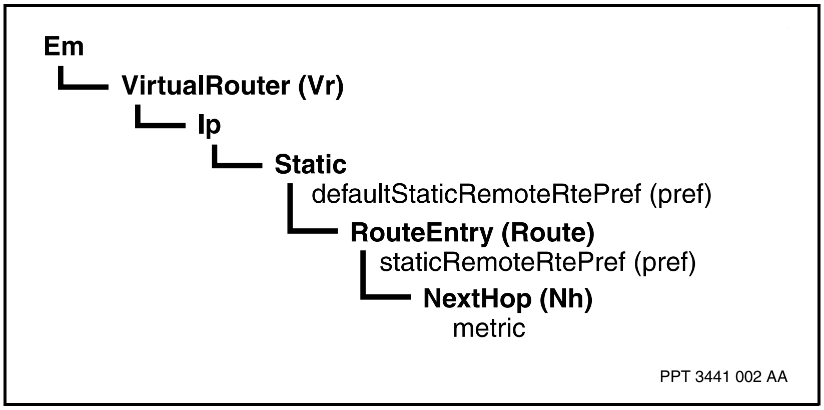
--End--

Variable definitions

Variable	Value
<lp_group>	is the LpGroup for the 4pGe LPs.
<lp_number>	is the number of the Lp.
<mig_behaviour_value>	is the migration behaviour value: stayInServiceShelf or moveToMigrationShelf.

Procedure job aid

LpGroup component hierarchy



Configuring IP prerequisites and defaults for provisioning a protected default route

Configure equal cost multi path (ECMP) as disabled for both Static and OSPF protocols to allow a protected default route (PDR) to be provisioned. PDR is not supported with ECMP enabled.

Attention: If an OSPF default route (DR) is introduced into the PDR VR space then an outage of more than one second is expected. In order to maintain outages of less than one second with PDR present, this configuration is not recommended.

Prerequisites

- Ensure that attribute *autoNegotiation* of the *Ethernet* component is turned on. For information, see NN10600-551 *Nortel Networks Multiservice Switch 7400/15000/20000 FP Configuration Reference*.
- Ensure that attribute *cpEquipementProtection* of the *Shelf* component is set to hot. See procedure Changing the CP equipment protection mode in NN10600-550 *Nortel Networks Multiservice Switch 7400/15000/20000 Common Configuration Procedures*.

Procedure steps

Step **Action**

1 Optionally, set the *preConfigFwdPath* attribute to avoid ‘first packet exceptions’ when installing more specific static or OSPF routes.

```
set Vr/1 Ip preConfigFwdPath enabled
```

Attention: Turning on this attribute should only be done if the number of nextHop routers, across all VRs of this switch, is less than 1000; otherwise resource exhaustion of IP forwarding records on the FP may occur.

2 Disable ECMP support in the static protocol.

```
set Vr/<vr_name> Ip Static maxEcmpNextHops 1
```

3 Disable ECMP support in the OSPF protocol.

```
set Vr/<vr_name> Ip Ospf ecmpStatus disable
```

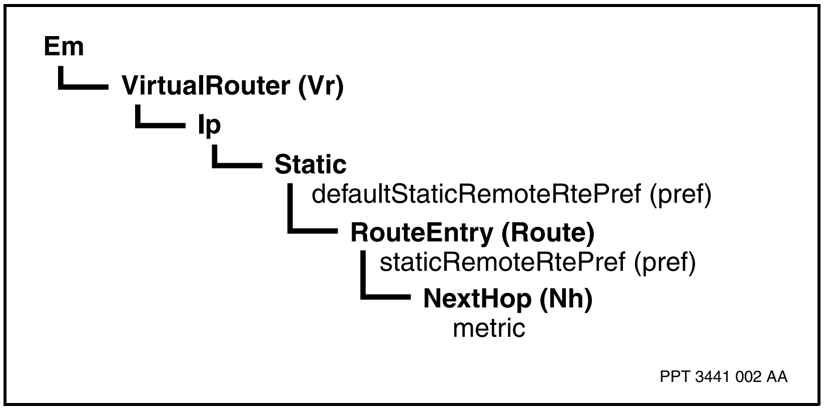
--End--

Variable definitions

Variable	Value
<vr_name>	is the name of the virtual router.

Procedure job aid

IP static protocols component hierarchy



Defining protected default route

Define protected default route to enable hitless (less than one second) IP forwarding support on the IP default route, for 4pGe line failures, card failures, and software migrations.

Procedure steps

Step	Action
1	Add an IP default route. `add Vr/<vr_name> Ip Static Route/0.0.0.0,0.0.0.0,0`
2	Set IP default route as protected. `add Vr/<vr_name> Ip Static Route/0.0.0.0,0.0.0.0,0 protected yes`
3	Set the route preference to most preferred for the protected default route. `add Vr/<vr_name> Ip Static Route/0.0.0.0,0.0.0.0,0 staticRemoteRtePreference 1`
4	Add a nextHop for the protected default route. The nextHop must be attached to a 4pGe protocol port. `add Vr/<vr_name> Ip Static Route/0.0.0.0,0.0.0.0,0 nextHop/<nextHop_address>`
5	Repeat step 4 to add one or more nextHops.

Attention: A protected default route must have at least 2 4pGe nextHops over unique 4pGe protocol ports.

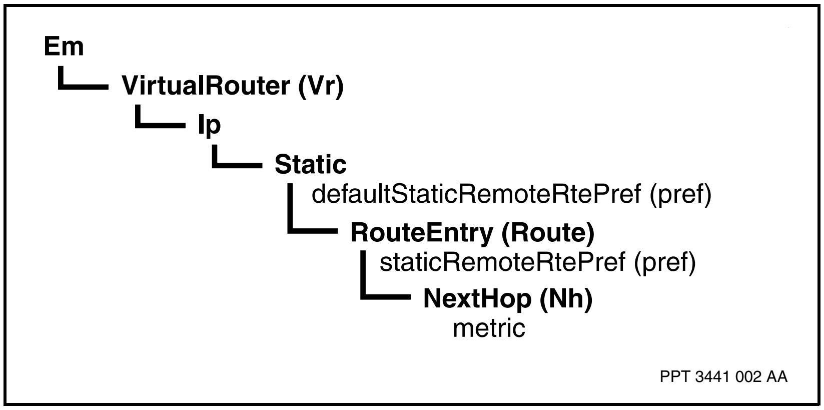
--End--

Variable definitions

Variable	Value
<nextHop_address>	is the IP address of the nextHop router.
<vr_name>	is the name of the virtual router.

Procedure job aid

Protected static route component hierarchy



Configuring VR BOOTP routing protocol

Configure bootstrap protocol (BOOTP) to allow dynamic configuration of a booting host. Nortel Networks Multiservice Switch systems support the BOOTP relay agent functionality described in RFC951 and RFC1542

Procedure steps

Step **Action**

1 Create the BOOTP component, as a subcomponent of IP, so that the IP/UDP accepts packets destined for the BOOTP server port.

```
add Vr/<vr_name> Ip BootpRelayAgent
```

2 Add a *BootpPort* subcomponent to all IP interfaces on the *Vr* component.

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort BootpPort
```

3 Set the *relayForwardStatus* attribute to control how BOOTP request packets, received on another port, will be flooded out this port.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort BootpPort
relayForwardStatus <status_value>
```

4 Identify the logical interface whose address is to be the GIADDR of this BOOTP port.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort BootpPort
bootpLogicalInterface <portIf__value>
```

5 Identify a list of relay addresses to receive BOOTP request messages when they are received on this port. BOOTP requests are also broadcast out all other BOOTP configured ports according to the *relayForwardStatus* attribute of the outgoing port.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort BootpPort
relayAddress <addr__value>
```

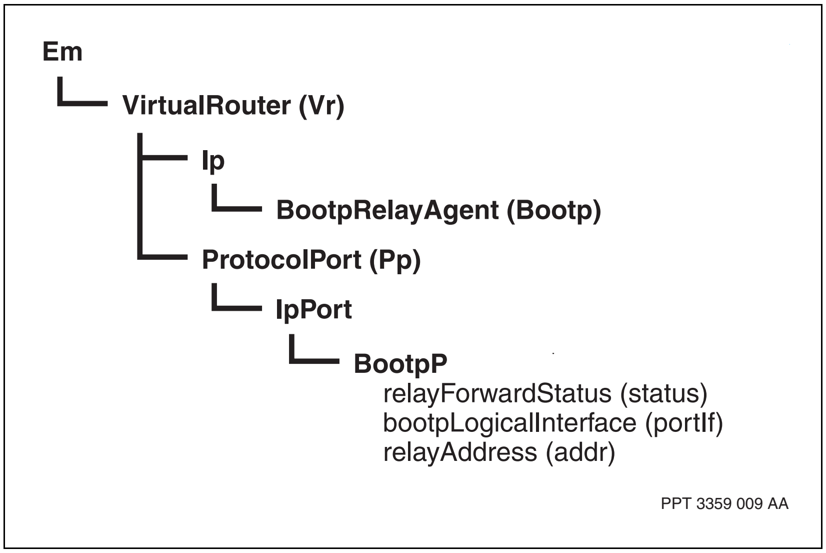
--End--

Variable definitions

Variable	Value
<addr_value>	is the value for the *relayAddress* attribute. There is no default value.
<portIf_value>	is the value for the *bootpLogicalInteface* attribute. The default IP address is 0.0.0.0.
<pp_name>	is the name of the protocol port.
<status_value>	is the value for the *relayForwardStatus* attribute. The default value is disabled.
<vr_name>	is the name of the virtual router.

Procedure job aid

Bootstrap protocol component hierarchy



VR IP features configuration

Configure VR IP features to provide additional IP capabilities that enhance Nortel Networks Multiservice Switch networks.

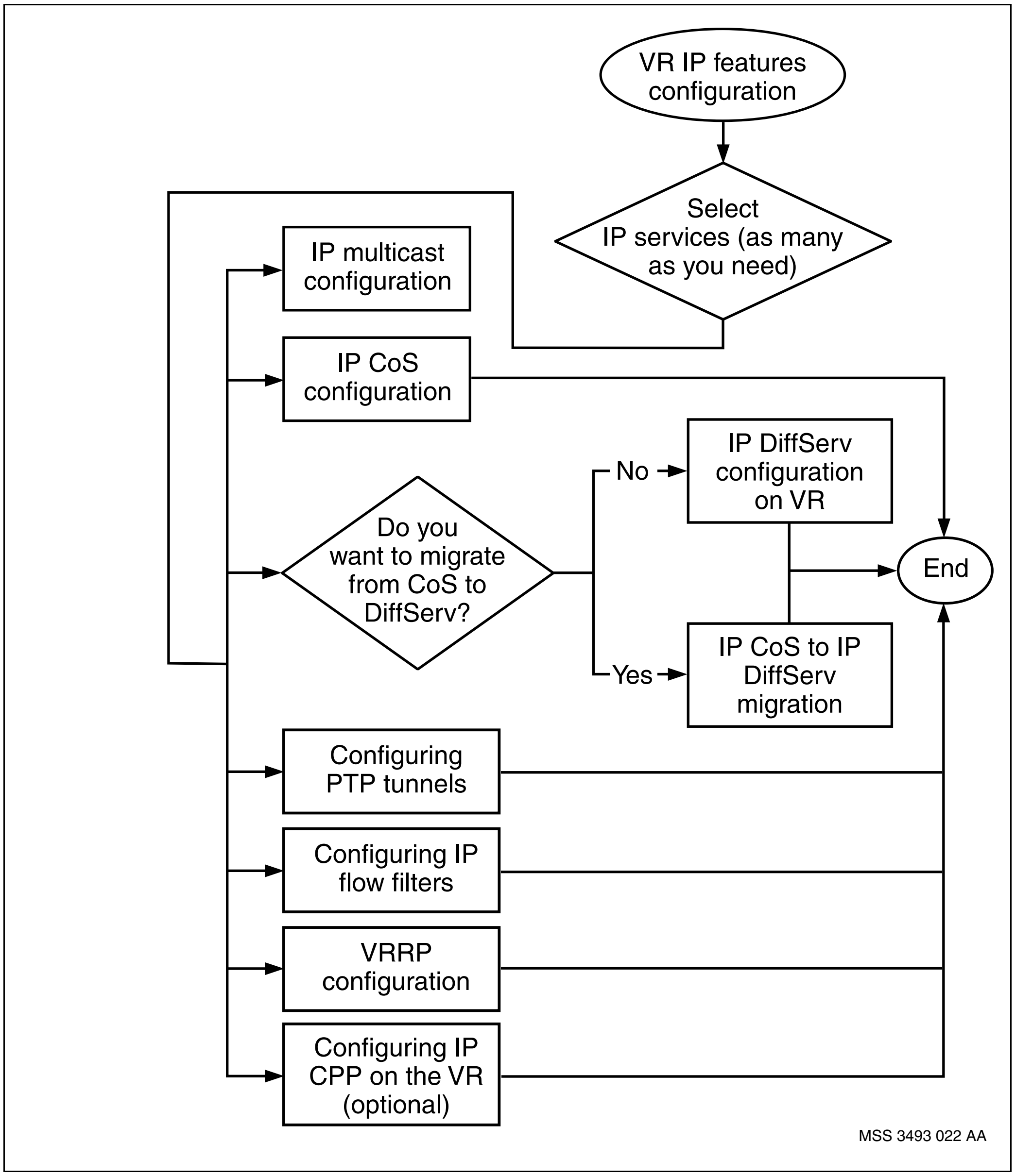
Prerequisites to VR IP features configuration

- For a list of services supported on specific cards, see section “Applications and services supported by function processor” in NN10600-551 *Nortel Networks Multiservice Switch 7400/15000/20000 FP Configuration Reference*.

VR IP features configuration tasks

This work flow shows you the sequence of tasks you perform to configure VR IP services. To link to any task, go to VR IP features configuration task navigation (page 204) following the task flow.

VR IP features configuration task


VR IP features configuration
Select IP services (as many as you need)
IP multicast configuration
IP CoS configuration
Do you want to migrate from CoS to DiffServ?
No
Yes
IP DiffServ configuration on VR
IP CoS to IP DiffServ migration
End
Configuring PTP tunnels
Configuring IP flow filters
VRRP configuration
Configuring IP CPP on the VR (optional)
MSS 3493 022 AA

VR IP features configuration task navigation

- IP multicast configuration (page 206)
- IP CoS configuration. See NN10600-591 *Nortel Networks Multiservice Switch 7400/15000/20000 Layer 3 Traffic Management Configuration*
- IP DiffServ configuration on VR. See NN10600-591 *Nortel Networks Multiservice Switch 7400/15000/20000 Layer 3 Traffic Management Configuration*

- IP CoS to DiffServ migration. See NN10600-591 *Nortel Networks Multiservice Switch 7400/15000/20000 Layer 3 Traffic Management Configuration*
- Configuring PTP tunnels (page 216)
- Configuring VR IP flow filters. See NN10600-591 *Nortel Networks Multiservice Switch 7400/15000/20000 Layer 3 Traffic Management Configuration*
- VRRP configuration (page 218)
- Configuring IP CPP on the VR (page 234)

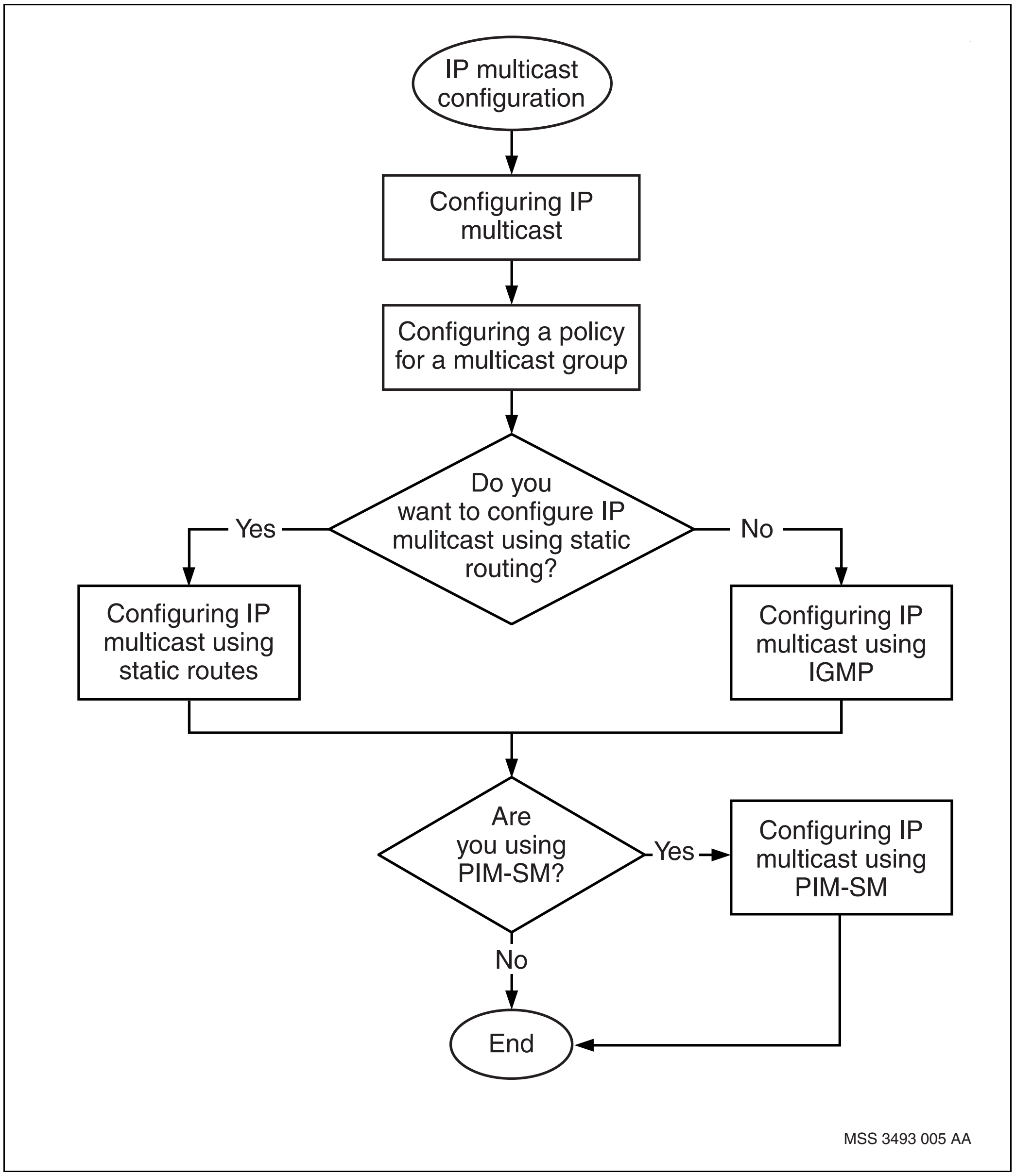
IP multicast configuration

Configure IP multicast totransmit multicast data packets to multiple receivers.

IP multicast configuration procedures

This task flow shows you the sequence of procedures you perform to configure IP multicast. To link to any procedure, go to IP multicast configuration procedure navigation (page 207).

IP multicast configuration procedures


IP multicast configuration
Configuring IP multicast
Configuring a policy for a multicast group
Do you want to configure IP mulitcast using static routing?
Yes
No
Configuring IP multicast using static routes
Configuring IP multicast using IGMP
Are you using PIM-SM?
Yes
Configuring IP multicast using PIM-SM
No
End
MSS 3493 005 AA

IP multicast configuration procedure navigation

- Configuring IP multicast (page 208)
- Configuring a policy for a multicast group (page 209)
- Configuring IP multicast using static routes (page 211)
- Configuring IP multicast using IGMP (page 213)
- Configuring IP multicast using PIM-SM (page 214)

Configuring IP multicast

Configure IP multicast on a VR to add the *Mcast* component and provide support for IP multicast.

Prerequisites

- Configure IP.

Procedure steps

Step	Action
1	Add the *multicast* component to the VR. **`add Vr/<vr_name> Ip Mcast`**

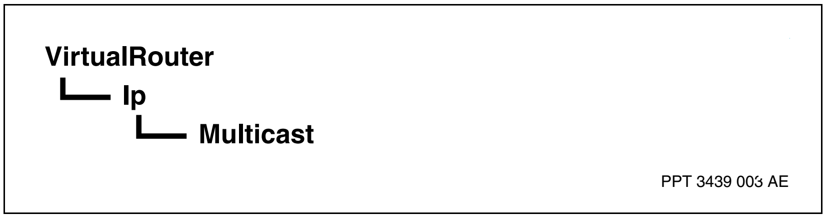
--End--

Variable definitions

Variable	Value
<vr_name>	is the name of the virtual router.

Procedure job aid

IP multicast component hierarchy



Configuring a policy for a multicast group

Configure a policy for a multicast group to control multicast forwarding on specified interfaces. Policy groups can be configured to include a range of multicast group addresses and an action attribute which is set to allow or deny the forwarding of the specified groups. Policy groups may be linked to one or more interfaces.

Attention: A new policy will not take effect if the multicast group addressed by the policy is already in the multicast forwarding table and if one of the OIFs for the group in the multicast forwarding table is one of the *linkToPolicyUser* ports.

Procedure steps

Step **Action**

1 Add a *PolicyGroup* component.

```
add Vr/<vr_name> Ip Mcast PolicyGroup/
<policy_group_name>
```

2 Add a *Group* component.

```
add Vr/<vr_name> Ip Mcast PolicyGroup/
<policy_group_name> group/<ipAddress>,<netmask>
```

3 Set the action performed by the policy.

```
set Vr/<vr_name> Ip Mcast PolicyGroup/
<policy_group_name> action <action>
```

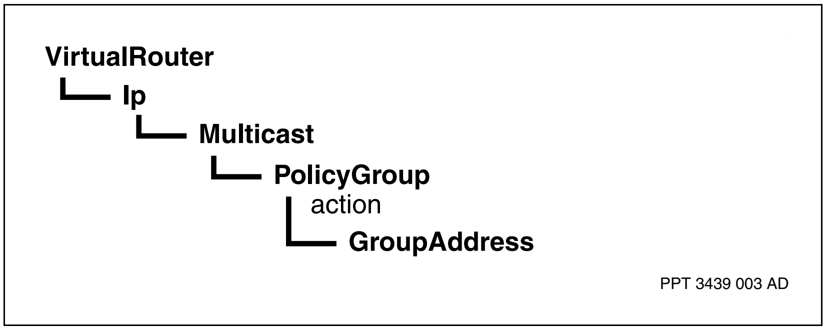
--End--

Variable definitions

Variable	Value
<action>	specifies the action to be taken if a join request is received for any of the multicast group addresses configured as part of this policy.
<ip_address>	is the 32-bit class D multicast group IP address for a local subnet.
<netmask>	is the 32-bit network mask you assign to this IP address.
<policy_group_name>	is the descriptive name you assign to identify a particular policy group.
<vr_name>	is the name of the virtual router.

Procedure job aid

Policy for a multicast group component hierarchy



Configuring IP multicast using static routes

Configure IP multicast using static routes to enable the forwarding of multicast traffic without the use of multicast routing protocols. Static routes may be used alone or in combination with routing protocols. For example, PIM-SM will initiate joins for multicast groups specified by static entries as well as those learned through the use of IGMP.

Procedure steps

Step	Action
1	Add a *Static Route* component. `add Vr/<vr_name> Ip Mcast Static`
2	Configure a static route for a multicast group. `add Vr/<vr_name> Ip Mcast Static Route/<ipAddress>,<domain>`
3	Configure an out interface to a neighboring multicast router for a static route. `set Vr/<vr_name> Ip Mcast Static Route/<ipaddress>,<domain> outInterfaces Vr/<vr_name> Pp/<pp_id> Ipp logicalIf/<ipAddress>`

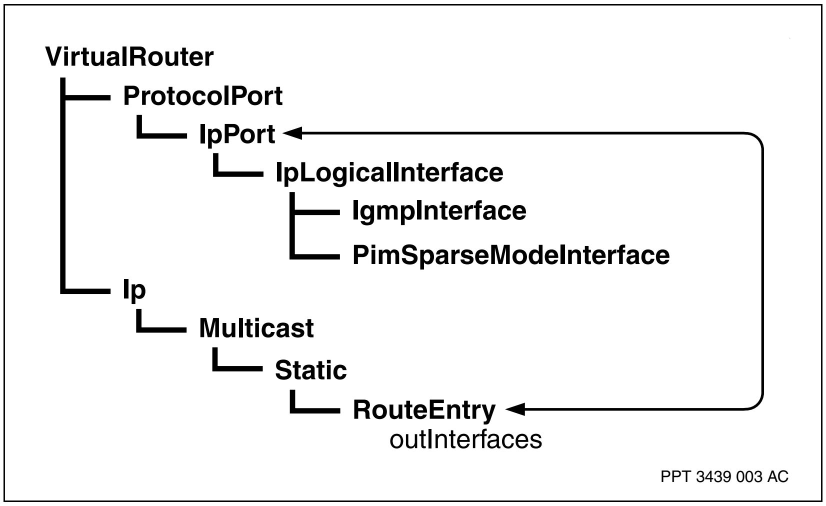
--End--

Variable definitions

Variable	Value
<domain>	is the 32-bit network mask you assign to this IP address.
<ip_address>	is the-32 bit class D multicast group IP address for a local subnet.
<pp_id>	is the instance name you assign to this protocol port.
<vr_name>	is the name of the virtual router.

Procedure job aid

IP multicast using static routes component hierarchy



Configuring IP multicast using IGMP

Configure IP multicast to run the internet group management protocol (IGMP) protocol on a virtual router.

Procedure steps

Step **Action**

1 Enable the IGMP protocol on the VR by adding the *Igmp* component.

```
add Vr/<vr_name> Ip Mcast Igmp
```

2 Enable IGMP on an interface that will communicate with IGMP hosts by adding an *IgmpIf* component.

```
add Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/<ipAddress>
IgmpIf
```

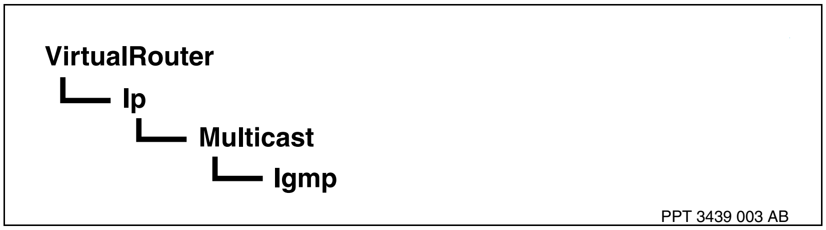
--End--

Variable definitions

Variable	Value
<ip_address>	is the 32-bit address you assign to this logical interface.
<pp_id>	is the instance name you assign to this protocol port.
<vr_name>	is the name of the virtual router.

Procedure job aid

IP multicast using IGMP component hierarchy



Configuring IP multicast using PIM-SM

Configure IP multicast with PIM-SM to route multicast traffic to sparsely populated receivers.

Procedure steps

Step **Action**

1 Enable PIM-SM by adding the *PimSm* component.

```
add Vr/<vr_name> Ip Mcast PimSm
```

2 Configure a multicast domain on the VR by adding a domain identifier.

```
add vr/<vr_name> Ip Mcast Domain/<n>
```

3 Configure a PIM-SM multicast domain on the VR by adding a *Domain* component.

```
add Vr/<vr_name> Ip Mcast PimSm Domain/<n>
```

4 Configure a VR as a candidate RP router in a domain by adding the *CandidateRp (CRp)* component.

```
add Vr/<vr_name> Ip Mcast PimSm Domain/<n> CandidateRp
```

5 Set the range of class D multicast addresses for the candidate RP router using the *CRp* component.

```
add Vr/<vr_name> Ip Mcast PimSm Domain/<n> CRp Group/
<group_address>, <mask>
```

6 Configure a VR as a candidate BSR router in a domain by adding the *CandidateBsr (CBsr)* component.

```
add Vr/<vr_name> Ip Mcast PimSm Domain/<n> CandidateBsr
```

7 Enable the PIM-SM protocol on the IP interface of the applicable protocol port by adding the *PimSmIf* component.

```
add Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/<ipAddress>
PimSmIf
```

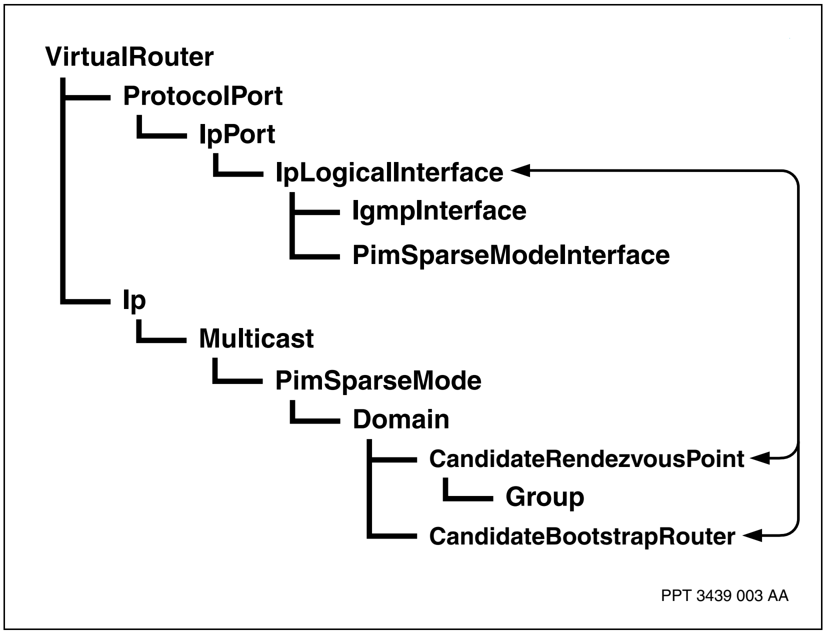
--End--

Variable definitions

Variable	Value
<group_address>	is the instance value of the group address.
<ip_address>	is the 32-bit address you assign to this logical interface.
<mask>	is the instance value of the mask.
<n>	is the number assigned to this instance of the *Domain* component.
<pp_id>	is the instance name you assign to this protocol port.
<vr_name>	is the name of the virtual router.

Procedure job aid

IP multicast using PIM-SM component hierarchy



Configuring PTP tunnels

Configure a point-to-point (PTP) IP tunnel instance to connect two physically separate networks, which share the same address space through an IP network of a different address space.

Prerequisites

- Configuring and linking a protocol port to a media interface (page 99)
- Enabling IpPort on a protocol port (page 101)
- Configure a RIP or OSPF interface so that the tunnel endpoints can receive information about remote subnetworks. See
  - — Configuring a RIP interface (page 110)
  - — Configuring an OSPF interface (page 135)

Procedure steps

Step **Action**

1 Create an IP tunnel instance on the virtual router.

```
add Vr/<vr_name> Ip Tunnel
```

2 Configure the end point for a point-to-point IP tunnel.

```
add Vr/<vr_name> Ip Tunnel Sep/<sep_id>
```

3 Specify the source address of the IP tunnel.

```
set Vr/<vr_name> Ip Tunnel Sep/<sep_id> src <src_addr>
```

4 Specify the tunnel destination address.

```
set Vr/<vr_name> Ip Tunnel Sep/<sep_id> dst <dst_addr>
```

5 Specify the tunnel encapsulation type.

```
set Vr/<vr_name> Ip Tunnel Sep/<sep_id> encapType <type>
```

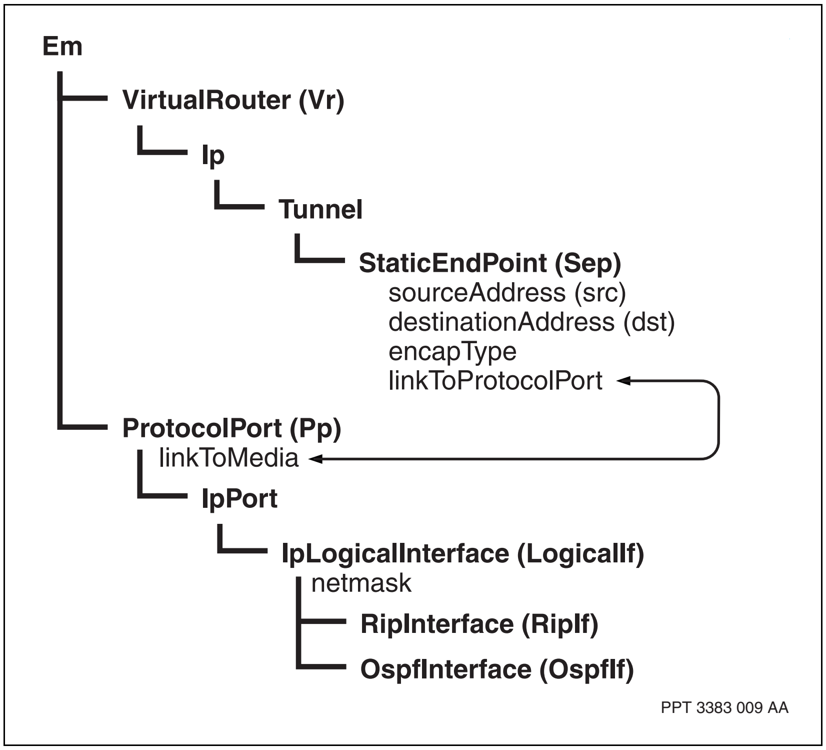
--End--

Variable definitions

Variable	Value
<dst_addr>	is the IP address at the destination end of the tunnel.
<sep_id>	is the instance number of the tunnel end point.
<src_addr>	is the IP address at the source end of the tunnel.
<type>	is the encapsulation type: IP in IP.
<vr_name>	is the name of the virtual router.

Procedure job aid

PTP tunnels component hierarchy



VRRP configuration

Configure the virtual router redundancy protocol (VRRP) to enable router redundancy and availability to IP routing.

The Nortel Networks Multiservice Switch system supports the VRRP functionality described in RFC 2338.

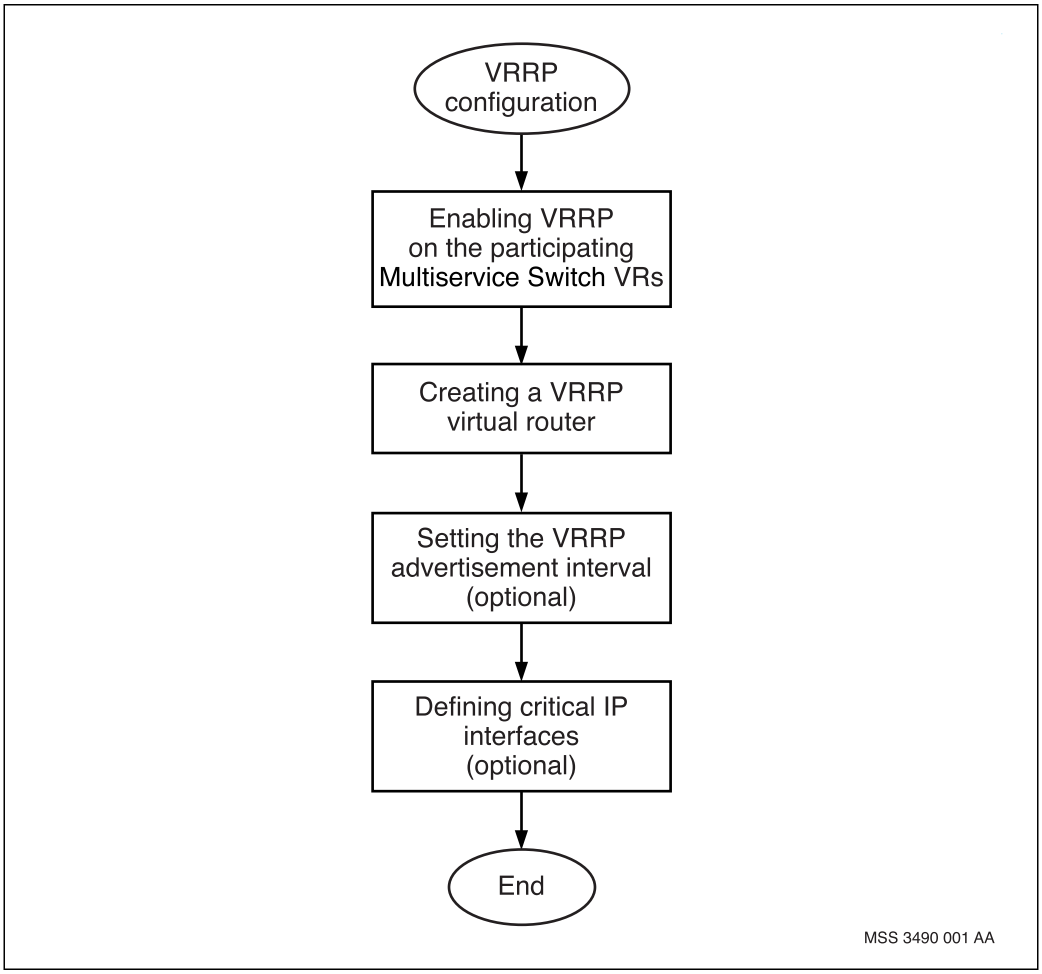
Prerequisites to VRRP configuration

- See the section on VRRP in the NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Technology Fundamentals*.
- The task flow and procedures in this section describe configuring and monitoring VRRP software and services only. Basic configuration at the node level (in this case, creating an instance of a logical processor type (LPT), and adding the *ip and ipVrrp* services to the *featureList* attribute) must be performed first. Use the tasks and procedures in NN10600-550 *Nortel Networks Multiservice Switch 7400/15000/20000 Common Configuration Procedures* if you require supporting information or need to provision or reconfigure any node or nodal elements to support VRRP features.

VRRP configuration procedures

This task flow displays the sequence of procedures you must perform to configure VRRP. To link to any procedure, go to VRRP configuration procedure navigation (page 219).

VRRP configuration procedures


VRRP configuration
Enabling VRRP on the participating Multiservice Switch VRs
Creating a VRRP virtual router
Setting the VRRP advertisement interval (optional)
Defining critical IP interfaces (optional)
End
MSS 3490 001 AA

VRRP configuration procedure navigation

- Enabling VRRP on participating Multiservice Switch VRs (page 220)
- Creating a VRRP virtual router (page 221)
- Setting the VRRP advertisement interval (page 230)
- Defining critical IP interfaces (page 232)

Enabling VRRP on participating Multiservice Switch VRs

Enable VRRP on Nortel Networks Multiservice Switch node VRs to enable the VRRP process on participating VRs. Typically, VRs are on different Multiservice Switch nodes and are connected over an Ethernet LAN segment. Alternatively, any of the following configurations can be used:

- A single Multiservice Switch that has two VRs with the same IP address reachability. Each VR has a VRRP VR instance. One VRRP VR instance is the master, and the other is the backup. Each VRRP VR instance is on a different FP.
- A single Multiservice Switch that has two VRs with the same IP address reachability. Each VR has a VRRP VR instance. One VRRP VR instance is the master, and the other is the backup. Each VRRP VR instance is on a different port on the same FP.

Also, VRRP on one Multiservice Switch VR can interwork with an external router implementing RFC 2338-compliant VRRP on the same Ethernet LAN/VLAN segment.

Procedure steps

Step **Action**

1 Add the *Ip* and *Vrrp* components on a designated master router.

```
add Vr/<router_A> Ip Vrrp
```

2 Add the *Ip* and *Vrrp* components on a second Multiservice Switch router.

```
add Vr/<router_B> Ip Vrrp
```

3 Repeat step 2 for each additional Multiservice Switch VR participating in the VRRP virtual router.

4 Activate the changes. See Completing configuration changes (page 13).

--End--

Variable definitions

Variable	Value
<router_A>	is the instance name of a Multiservice Switch VR on node A.
<router_B>	is the instance name of a Multiservice Switch VR on node B.

Creating a VRRP virtual router

Create a VRRP virtual router to provide router redundancy.

You can create a VRRP virtual router on an Ethernet protocol port but not on another media protocol port.

On the 4-port 10/100 BaseT Ethernet, 4-port Gigabit Ethernet, and 8-port 10/100 BaseT Ethernet function processors (FPs), only one instance of VRRP per protocol port is supported. Other interior gateway protocols on the same protocol port must be in passive mode.

Prerequisites

CAUTION
Risk of service interruption
Activation of provisioning changes to the *VrrpPort* component could result in a brief service interruption if the *VrrpPort* component has a current *virtualRouterState* attribute of master and does not own an IP address (priority not equal to 255).

Procedure steps

Step **Action**

1 Add the *VrrpPort* component to the Nortel Networks Multiservice Switch VR acting as the master VRRP VR.

```
add Vr/<router_A> ProtocolPort/<ethernet> IpPort
VrrpPort/<VRid>
```

2 Set the *ipAddresses* attribute for the VRRP VR with which it is associated.

```
set Vr/<router_A> ProtocolPort/<ethernet> IpPort
VrrpPort/<VRid> ipAdresses <ipaddress>
```

3 Set the *priority* attribute of the master VRRP VR.

```
set Vr/<router_A> ProtocolPort/<ethernet> IpPort
VrrpPort/<VRid> priority <priority_value>
```

4 Add the *VrrpPort* component to the Multiservice Switch VR acting as the backup VRRP VR.

```
add Vr/<router_B> ProtocolPort/<ethernet> IpPort
VrrpPort/<VRid>
```

Attention: The VR acting as the backup VRRP VR can be the same as the VR acting as the master VRRP VR.

5 Optionally, set the *ipAddresses* attribute of VRRP VR.

```
set Vr/<backup> ProtocolPort/<ethernet> IpPort VrrpPort/
<VRid> ipAdresses <ipaddress>
```

6 Set the *priority* attribute of the backup VRRP VR.

```
set Vr/<backup> ProtocolPort/<ethernet> IpPort VrrpPort/
<VRid> priority <priority_value>
```

When you have multiple backup VRRP VRs, it is very important to set the priority because it defines which backup VRRP VR is active when the master VRRP VR has a failure.

7 For the 2-port 10/100 BaseT Ethernet function processor only, repeat step 4 to step 6 for each additional backup VRRP VR.

8 Activate the changes. See Completing configuration changes (page 13).

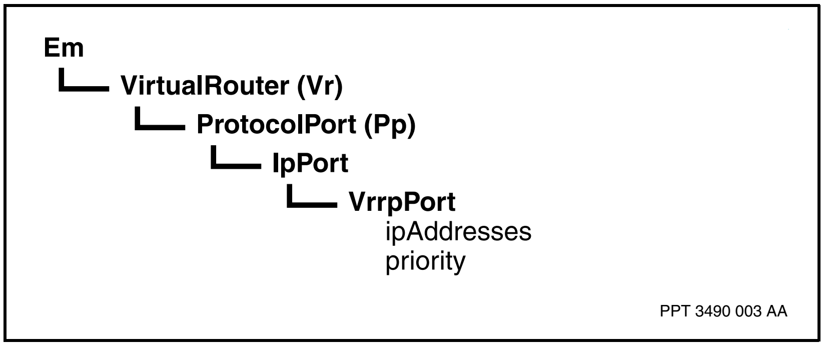
--End--

Variable definitions

Variable	Value
<backup>	is the name of the Multiservice Switch VR designated as a backup VRRP VR.
<ethernet>	is the instance of the Ethernet protocol port.
<ipaddress>	specifies one or more IP addresses associated with the *VrrpPort* instance. Typically, this IP address is same as the one for the master VRRP VR
<priority_value>	is a decimal value between 1 and 255. The higher priority is for the master VRRP VR.
<router_A>	is an instance name of a Multiservice Switch VR on node A.
<router_B>	is an instance name of a Multiservice Switch VR on node B.
<VRid>	is the decimal VRRP virtual router identifier between 1 and 255. Only one VRRP/VRid instance is supported per physical interface.

Procedure job aid

VRRP virtual router component hierarchy



Example of creating a VRRP virtual router associated with an Ethernet interface in port mode

Prior to starting this example, ensure that Nodes 1 and 2 have the *LanApplication* component in port mode. For more information about Configuring an Ethernet interface in port mode, see NN10600-580 *Nortel Networks Multiservice Switch 7400/15000/20000 Ethernet Service Operations*.

Procedure steps

Step	Action

1 Set the *featureList* attribute to include ip and ipvrrp.

```
set sw Lpt/ethernet featureList ip ipvrrp
```

2 Activate the changes without ending your provisioning session. See Completing configuration changes (page 13).

3 For Node 1, complete steps 4 through 7.

4 Add the *Vrrp* component under the *Ip* component to enable that feature on the virtual router (VR).

```
add Vr/1 Ip Vrrp
```

5 Add the *VrrpPort 1* component to the protocol port on the master VRRP VR.

```
add Vr/1 ProtocolPort/ethernet1 IpPort VrrpPort/1
```

6 Set the *ipAddresses* attribute that the VRRP VR is associated with.

```
set Vr/1 ProtocolPort/ethernet1 IpPort VrrpPort/1
ipAddresses <ipaddress_1B>
```

Attention: The <ipaddress_1B> is the logical interface IP address defined under, Vr/1 Pp/<ethernet1> IpPort logicalInterface/<ipaddress_1B>.

7 This VRRP VR is the master because it owns the IP address set in step 5. Set the *priority* attribute of the master VRRP VR to 255.

```
set Vr/1 ProtocolPort/ethernet1 IpPort VrrpPort/1
priority 255
```

8 For Node 2, complete steps 9 through 13.

9 Add the *Vrrp* component under the *Ip* component to enable that feature on the VR.

```
add Vr/1 Ip Vrrp
```

10 Add the *VrrpPort 1* component to the protocol port on the backup VRRP VR.

```
add Vr/1 ProtocolPort/ethernet1 IpPort VrrpPort/1
```

11 Set the *ipAddresses* attribute that the VRRP virtual router is associated with.

```
set Vr/1 ProtocolPort/ethernet1 IpPort VrrpPort/1
ipAddresses <ipaddress_1B>
```

12 This VRRP VR does not own the IP address set in step 11 because it is the backup VRRP VR. Set the *priority* attribute of the backup VRRP VR to a value less than 255.

```
set Vr/1 ProtocolPort/ethernet1 IpPort VrrpPort/1
priority <priority_value>
```

13 Activate configuration changes. See Completing configuration changes (page 13).

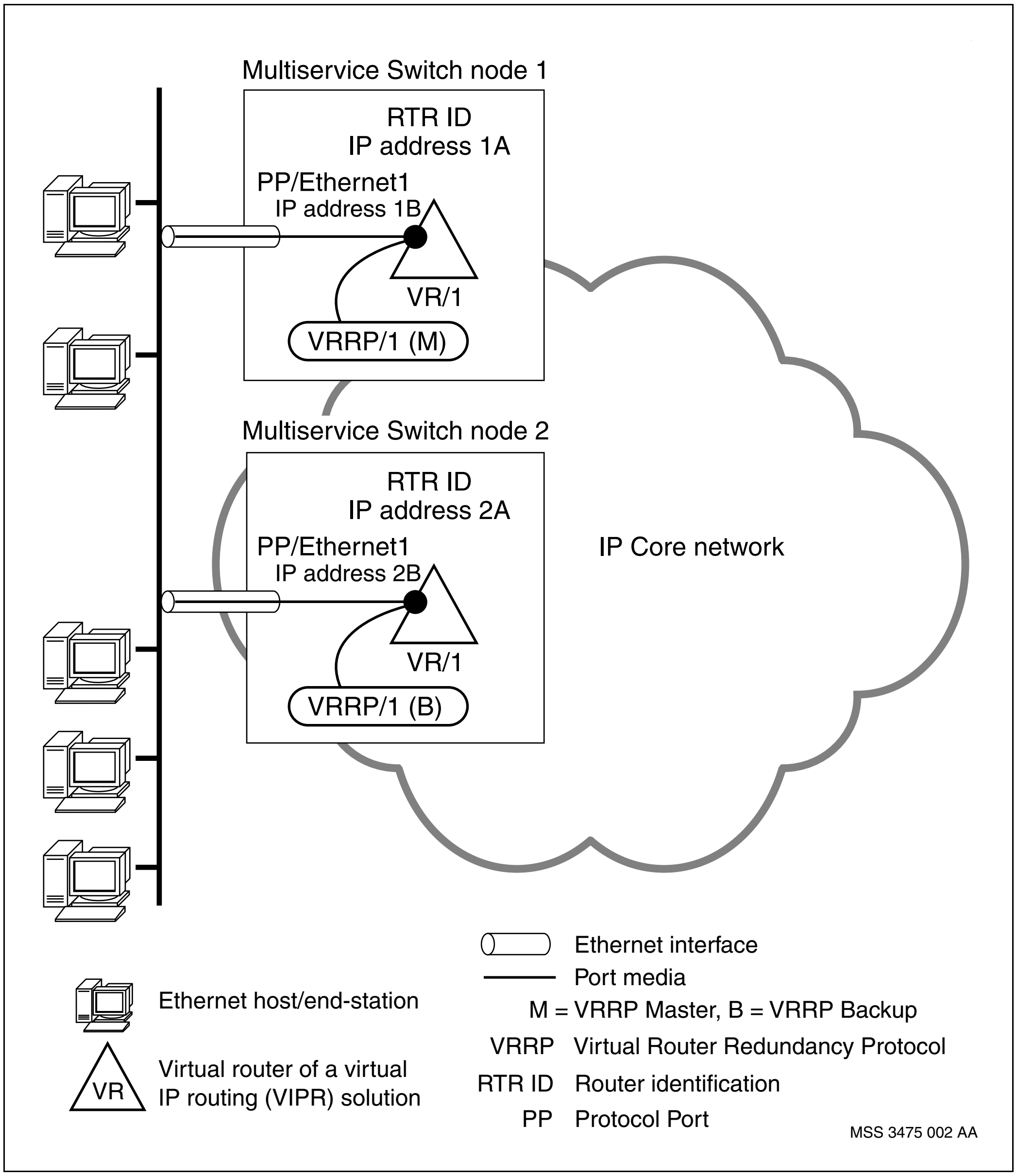
--End--

Variable definitions

Variable	Value
<ipaddress_1B>	is the IP address of the master VRRP VR.
<priority_value>	is a decimal value between 1 and 255. The higher priority is for the master VRRP VR.

Procedure job aid

Example of creating a VRRP virtual router associated with an Ethernet interface in port mode


Multiservice Switch node 1
RTR ID
IP address 1A
PP/Ethernet1
IP address 1B
VR/1
VRRP/1 (M)
Multiservice Switch node 2
RTR ID
IP address 2A
PP/Ethernet1
IP address 2B
VR/1
VRRP/1 (B)
IP Core network
Ethernet interface
Port media
M = VRRP Master, B = VRRP Backup
VRRP Virtual Router Redundancy Protocol
RTR ID Router identification
PP Protocol Port
Ethernet host/end-station
VR
Virtual router of a virtual IP routing (VIPR) solution
MSS 3475 002 AA

Example of creating a VRRP virtual router associated with a VLAN on an Ethernet interface

Prior to starting this example, ensure that Nodes 3 and 4 have the *LanApplication* component in VLAN mode. For more information about Configuring an Ethernet interface in VLAN mode, see NN10600-580 *Nortel Networks Multiservice Switch 7400/15000/20000 Ethernet Service Operations*

Procedure steps

Step **Action**

1 For Node 3, complete steps 2 through 6.

2 Add the *Vrrp* component under the *Ip* component to enable that feature on the virtual router.

```
add Vr/1 Ip Vrrp
```

3 Add the *VrrpPort 4* component to the protocol port on the master VRRP VR.

```
add Vr/1 ProtocolPort/ethernet1 IpPort VrrpPort/4
```

4 Set the *ipAddresses* attribute that the VRRP CR is associated with.

```
set Vr/1 ProtocolPort/ethernet1 IpPort VrrpPort/4
ipAddresses <ipAddress_3B>
```

5 This VRRP VR is the master because it owns the IP address set in step 4. Set the *priority* attribute of the master VRRP virtual router to 255.

```
set Vr/1 ProtocolPort/ethernet1 IpPort VrrpPort/4
priority 255
```

6 Activate the changes without ending your provisioning session. See Completing configuration changes (page 13).

7 For Node 4, complete steps 8 through 12.

8 Add the *Vrrp* component under the *Ip* component to enable that feature on the virtual router.

```
add Vr/1 Ip Vrrp
```

9 Add the *VrrpPort 4* component to the protocol port on the backup VRRP VR.

```
add Vr/1 ProtocolPort/ethernet2 IpPort VrrpPort/4
```

10 Set the *ipAddresses* attribute that the VRRP virtual router is associated with.

```
set Vr/1 ProtocolPort/ethernet2 IpPort VrrpPort/4
ipAddresses <ipAddress_3B>
```

11 This VRRP VR does not own the IP address set in the step 10 because it is the backup VRRP VR. Set the *priority* attribute of the backup VRRP VR to a value less than 255.

```
set Vr/1 ProtocolPort/ethernet2 IpPort VrrpPort/4
priority <priority_value>
```

12 Activate configuration changes. See Completing configuration changes (page 13).

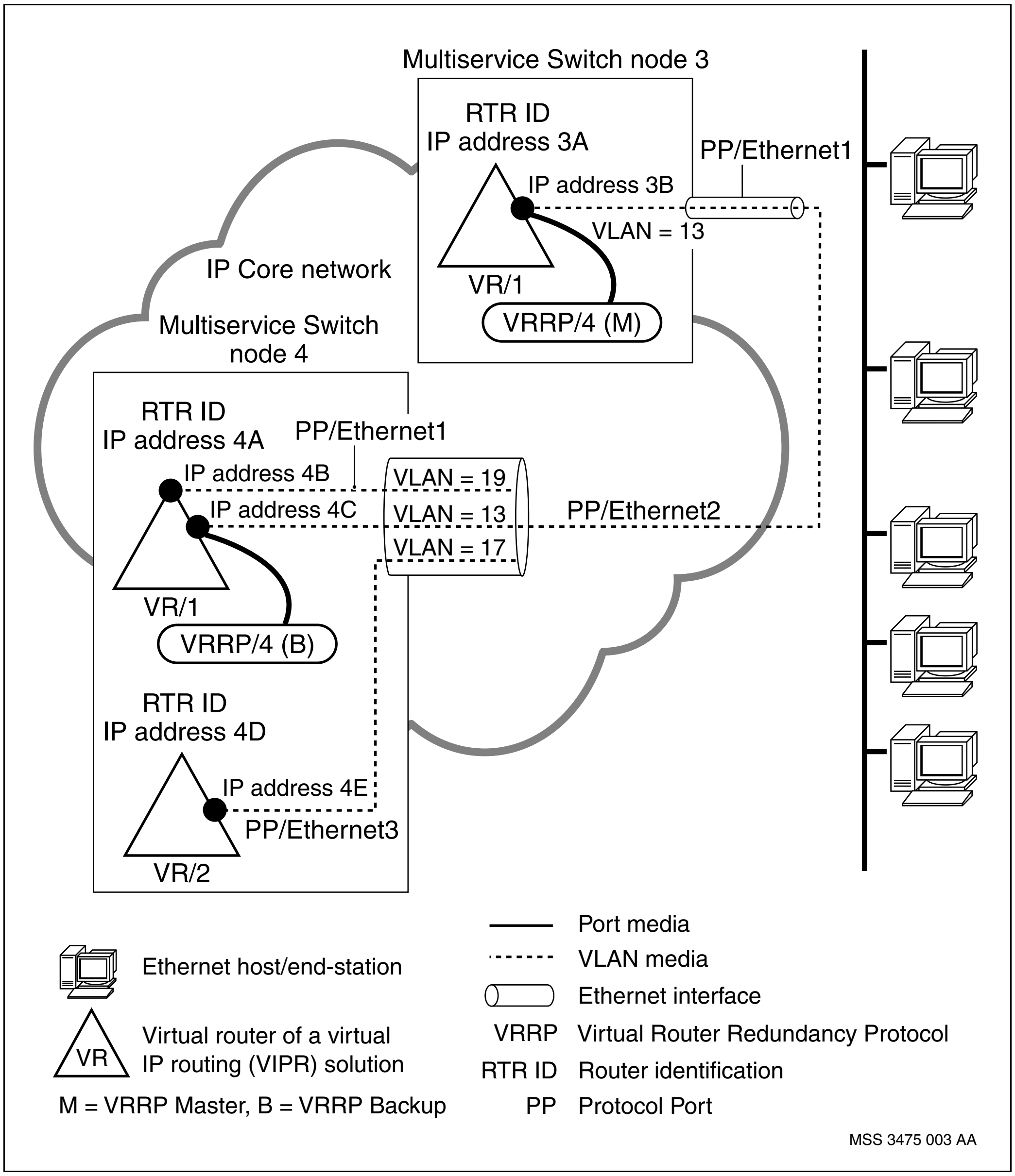
--End--

Variable definitions

Variable	Value
<ipaddress_3B>	is the IP address of the master VRRP VR.
<priority_value>	is a decimal value between 1 and 255. The higher priority is for the master VRRP VR.

Procedure job aid

Example of creating a VRRP virtual router associated with a VLAN on an Ethernet interface



Setting the VRRP advertisement interval

Optionally, set the VRRP advertisement interval on both the master and backup VRRP virtual routers (VRs) through the *advertisementInterval* attribute. The master and backup VRRP VRs require the same value.

Procedure steps

Step **Action**

1 Set the *advertisementInterval* attribute on the master VRRP VR.

```
set Vr/<router_A> ProtocolPort/<ethernet> IpPort
VrrpPort/<VRid> advertisementInterval <ad_inter>
```

2 Set the *advertisementInterval* attribute on the backup VRRP VR.

```
set Vr/<router_B> ProtocolPort/<ethernet> IpPort
VrrpPort/<VRid> advertisementInterval <ad_inter>
```

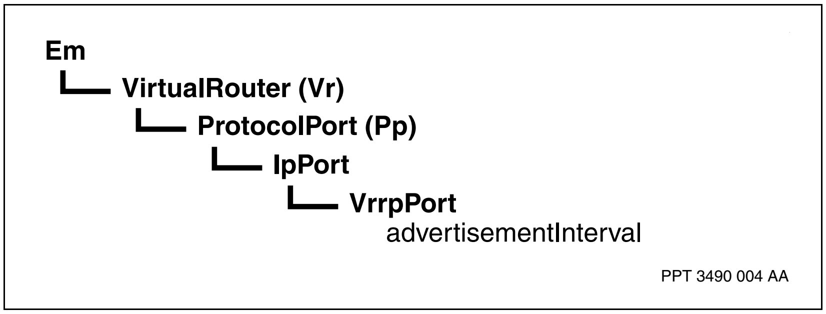
--End--

Variable definitions

Variable	Value
<ad_inter>	is the advertisement interval time in seconds
<ethernet>	is the instance of the Ethernet protocol port
<router_A>	is an instance name of a Multiservice Switch VR on node A.
<router_B>	is an instance name of a Multiservice Switch VR on node B.
<VRid>	is the decimal VRRP virtual router identifier between 1 and 255. Only one VRRP/VRid instance is supported per physical interface.

Procedure job aid

VRRP advertisement interval component hierarchy



Defining critical IP interfaces

Define an IP interface on the local router as critical to cause a role change to that VRRP virtual router.

Attention: Critical IP interfaces are supported on the 2-port 100 BaseT Ethernet, 4-port 10/100 BaseT Ethernet, 8-port 10/100 BaseT Ethernet, and 4-port Gigabit Ethernet function processors (FPs) only.

Procedure steps

Step	Action
1	Add a *CriticalIp* component. **`add Vr/<vr_name> ProtocolPort/<pp_id> IpPort CriticalIp/<cip_id>`**

2	Link the critical IP interface to the VRRP virtual router.
	`set Vr/<master> ProtocolPort/<enet> IpPort VrrpPort/<VRid> linktoCriticalIP Vr/<vr_name> ProtocolPort/<pp_id> IpPort CriticalIp/<vr_name>`
	or
	`set Vr/<vr_name> ProtocolPort/<pp_id> IpPort CriticalIp/<cip_id> linktoVrrp Vr/<master> ProtocolPort/<enet> IpPort VrrpPort/<vr_name>`

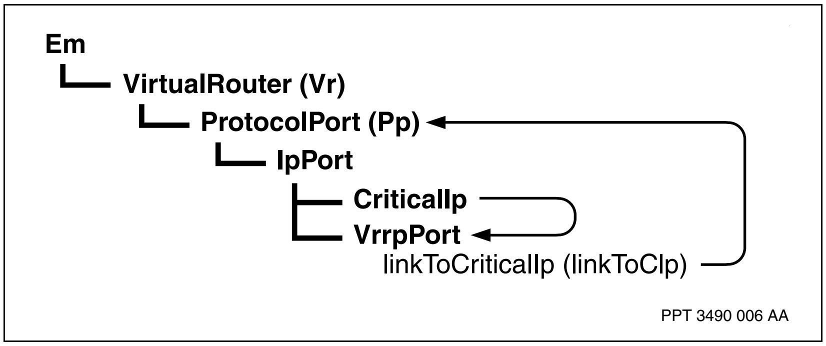
--End--

Variable definitions

Variable	Value
<cip_id>	is the instance (0..255) of the critical IP interface.
<enet>	is the instance of the Ethernet protocol port.
<master>	is the name of the Multiservice Switch VR acting as the master VRRP VR.
<pp_id>	is the instance value assigned to the protocol port running the IP traffic.
<VRid>	is the decimal VRRP virtual router identifier between 1 and 255.
<vr_name>	is the name of the Multiservice Switch VR owning the interface.

Procedure job aid

Critical IP interfaces component hierarchy



Configuring IP CPP on the VR

Configure IP control plane protection (CPP) on the VR to protect the Nortel Networks Multiservice Switch against certain denial of service (DoS) attacks on the control plane by monitoring the flow rate of IP packets destined for local IP destination addresses (DAs).

Attention: CPP is supported on PQC-based FPs only.

Prerequisites

- See the section on CPP in the NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Technology Fundamentals*.
- The procedure in this section describes configuring CPP software and services only. Basic configuration at the node level (in this case, creating an instance of a logical processor type (LPT), and adding the *ip and ipCpp* services to the *featureList* attribute) must be performed first. Use the tasks and procedures in NN10600-550 *Nortel Networks Multiservice Switch 7400/15000/20000 Common Configuration Procedures* if you require supporting information or need to provision or reconfigure any node or nodal elements to support the CPP feature.
- For monitoring IP CPP on the VR procedures, see Monitoring IP CPP on the VR (page 305).

Procedure steps

Step **Action**

1 Add a *Cpp* component as a subcomponent of the *Ip* component.

```
add Vr/<vr_name> ip cpp
```

2 Configure the number of packets per second before discard occurs.

```
set Vr/<vr_name> ip cpp packetsPerSeconds
<packets_per_second>
```

3 Configure the isolation time period.

```
set Vr/<vr_name> ip cpp isolationTime <isolation_time>
```

4 Configure the grace period.

```
set Vr/<vr_name> ip cpp gracePeriod <grace_period>
```

5 Configure the CPP mode.

```
set Vr/<vr_name> ip cpp mode <cpp_mode>
```

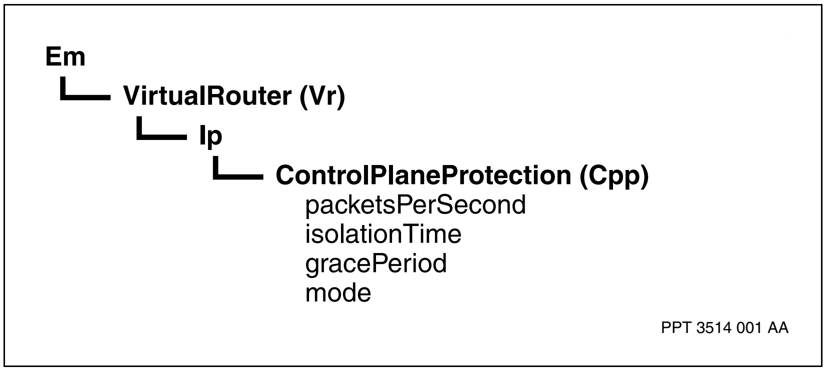
--End--

Variable definitions

Variable	Value
<cpp_mode>	is the CPP operational mode: study, protect or disabled. Use study to determine an acceptable traffic rate. Once you learn the rate created and have the appropriate configuration, use protect. Use disabled to pre-configure the feature without enabling the monitoring process.
<grace_period>	is the period over which the average flow rate is measured to ensure that the exceeded traffic flow rate still exceeds the maximum allowed rate. A value of zero in protect mode means that isolation occurs immediately after an excessive flow is detected.
<isolation_time>	is the amount of time over which traffic will be discarded once isolation has begun. A value of zero indicates to permanently discard traffic until the card is cleared by an operator.
<packets_per_second>	is the flow rate, in packets per second, for the VR's DA, that must be exceeded on a single DA before discard processing occurs.
<vr_name>	is the name of the virtual router.

Procedure job aid

VR CPP component hierarchy



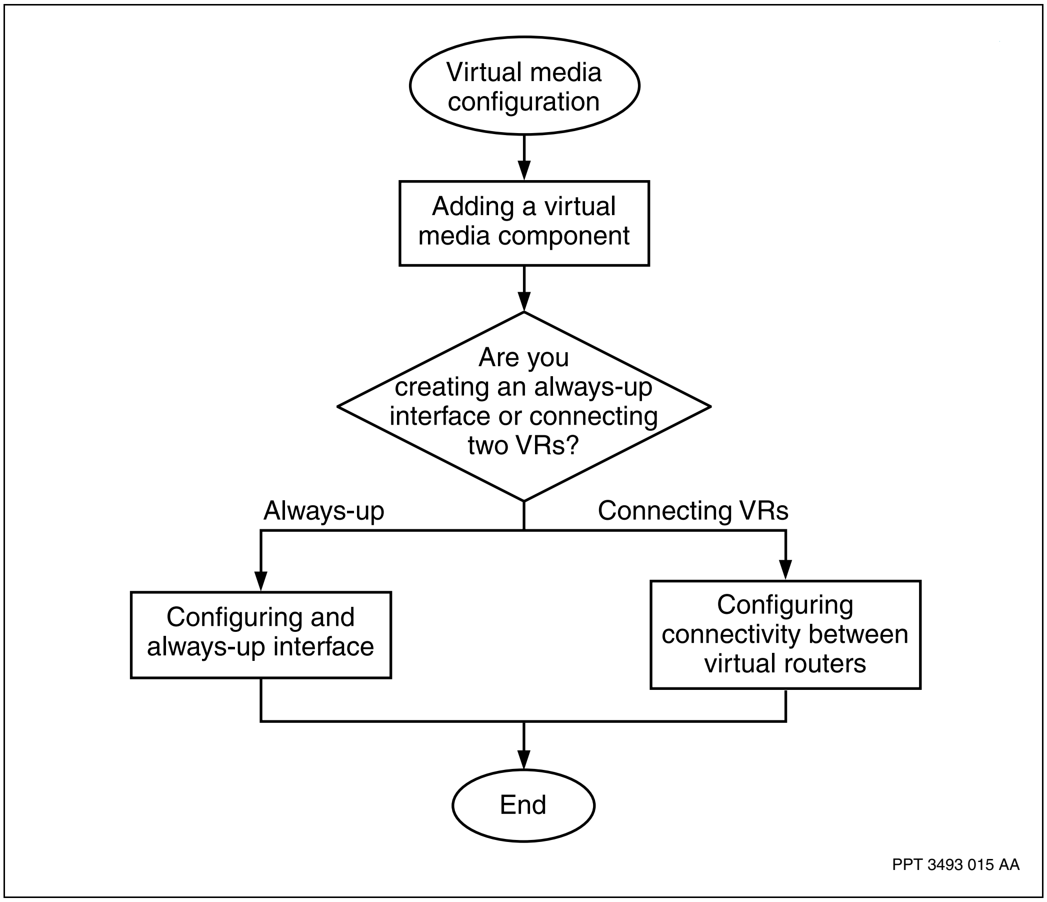
Virtual media configuration

Configure virtual media to connect two or more virtual routers on the same Nortel Networks Multiservice Switch node, or to create an always-up interface.

Virtual media configuration procedures

This task flow shows you the sequence of procedures you perform to configure virtual media on Nortel Networks Multiservice Switch nodes. To link to any procedure, go to Virtual media configuration procedure navigation (page 238).

Virtual media configuration procedures


Virtual media configuration
Adding a virtual media component
Are you creating an always-up interface or connecting two VRs?
Always-up
Connecting VRs
Configuring and always-up interface
Configuring connectivity between virtual routers
End
PPT 3493 015 AA

Virtual media configuration procedure navigation

- Adding a virtual media component (page 239)
- Configuring an always-up interface (page 241)
- Configuring connectivity between virtual routers (page 243)

Adding a virtual media component

Add a virtual media to provide virtual, rather than physical, next-hop connectivity between VRs, or if you intend to configure an always-up IP interface for RIP, OSPF, or BGP-4.

Procedure steps

Step	Action
1	Add one or more *Vm* component(s). **`add Vm/<m>`**
2	If required, add another *If* component as a subcomponent of the *Vm* component to interconnect virtual routers. **`add Vm/<m> If/<n>`**

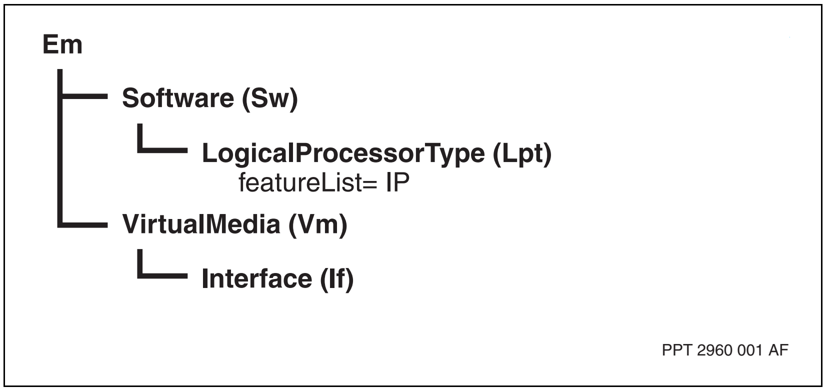
--End--

Variable definitions

Variable	Value
<m>	is the instance value of the *Vm* component. When you add a *Vm* component, the system automatically creates an *If* subcomponent, and assigns it an instance value of 0. For example, if you add *Vm*/0, the system automatically creates *If*/0 (*Vm*/0 *If*/0).
<n>	is the number you assign to the new instance of the *If* component. You can add additional *If* components (up to a maximum of 16, including *If*/0) to the *Vm* component.

Procedure job aid

Virtual media component hierarchy



Configuring an always-up interface

Configure an always-up interface to act as a loopback for routing protocols.

Procedure steps

Step	Action
1	Configure the *mode* attribute to be always up. **`set Vm/<m> if/<n> mode alwaysUpInterface`**
2	Link the virtual media interface to the protocol port. **`set Vm/<m> If/<n> LinkToProtocolPort vr/<vr_name> pp/<pp_name>`**
3	If you are configuring an always-up RIP interface, add the *RipIf* component. **`add vr/<vr_name> pp/<pp_name> Ip Log/<address> RipIf`**

4	If you are configuring an always-up OSPF interface, add the *OspfIf* component.
	`add vr/<vr_name> pp/<pp_name> Ip Log/<address> OspfIf`
	`set vr/<vr_name> ip ospf RouterId <address>`

5	If you are configuring an always-up BGP interface, configure the *Bgp* component. **`set vr/<vr_name> ip Bgp RouterId <address>`**

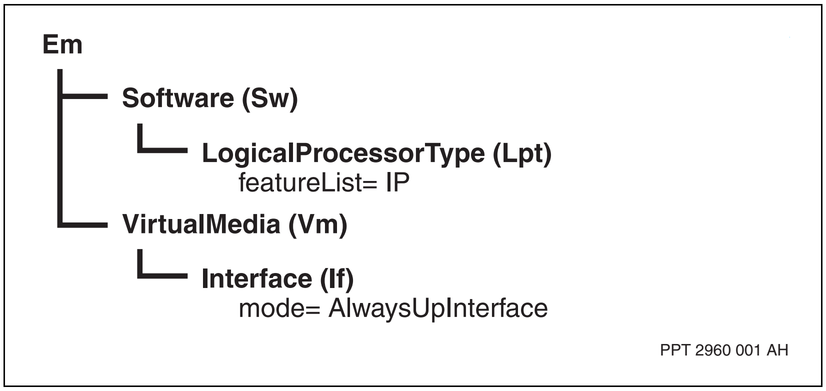
--End--

Variable definitions

Variable	Value
<address>	is the address of the IP address of the virtual media protocol port.
<m>	is the instance value of the *Vm* component. When you add a *Vm* component, the system automatically creates an *If* subcomponent, and assigns it an instance value of 0. For example, if you add *Vm*/0, the system automatically creates *If*/0 (*Vm*/0 *If*/0).
<n>	is the number you assign to the new instance of the *If* component. You can add additional *If* components (up to a maximum of 16, including *If*/0) to the *Vm* component.
<pp_name>	is the name of the protocol port.
<vr_name>	is the name of the virtual router.

Procedure job aid

Always-up interface component hierarchy



Configuring connectivity between virtual routers

Configure connectivity between two different VRs on the same Nortel Networks Multiservice Switch node to allow them to communicate.

Prerequisites

- Two virtual routers and their protocol ports have been provisioned. See Virtual router configuration (page 78) and Protocol ports configuration (page 97).

Procedure steps

Step **Action**

1 Link one *Vm If* component to the protocol port of a VR.

```
set Vm/<m> If/<n> linktoProtocolPort Vr/<vr_name1> Pp/
<pp_name1>
```

To enable connectivity between different VRs, the *Vm* component instance must be the same for the VRs you want to connect, but the *Vm If* component instances must be different.

2 Link the next *Vm If* component to the protocol port of a VR.

```
set Vm/<m> If/<n+1> linktoProtocolPort Vr/<vr_name2> Pp/
<pp_name2>
```

3 Set the *mode* attribute for both *Vm If* components to enable inter-VR connectivity.

```
set Vm/<m> If/<n> mode interVrConnection
set Vm/<m> If/<n+1> mode interVrConnection
```

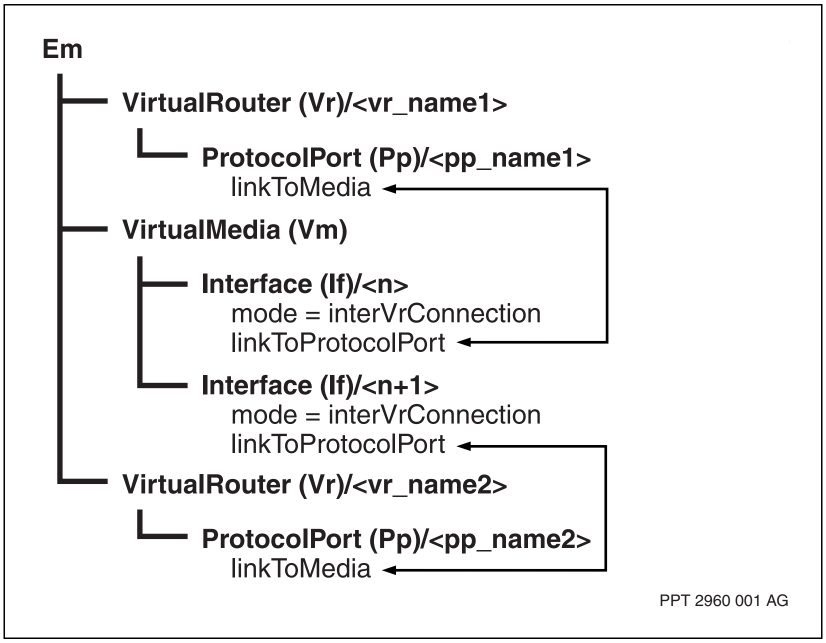
--End--

Variable definitions

Variable	Value
<m>	is the instance value of the virtual media component.
<n>	is the instance value of the interface component.
<pp_name1>	is the name of the protocol port on the first virtual router.
<pp_name2>	is the name of the protocol port on the second virtual router.
<vr_name1>	is the name of the first virtual router.
<vr_name2>	is the name of the second virtual router.

Procedure job aid

Connectivity between virtual routers component hierarchy



IP monitoring and testing

This section contains information about the following:

Monitoring the ATM MPE configuration

This section contains the information you need to monitor and maintain the IP over ATM MPE configuration. Issue all commands in operational mode. See Operational mode (page 12).

For information on specific components and protocols, see the following sections:

- ATM MPE component states (page 246)
- ATM MPE soft PVC component states (page 246)
- Monitoring the AtmMpe component (page 247)
- Monitoring the AtmConnection subcomponent (page 247)
- Monitoring the IlsFwdr component (page 248)
- Testing ATM MPE soft PVC connectivity (page 248)
- Testing ATM MPE soft PVC data flow (page 249)

ATM MPE component states

The table ATM MPE component states (page 246) lists the operational states reported by the ATM MPE service.

ATM MPE component states

Condition	States reported
ifAdminStatus is provisioned as *down* or *testing*	operational: enabled usage: idle snmpOperStatus: down or testing
ifAdminStatus is provisioned as *up* and the component is locked	operational: enabled usage: idle snmpOperStatus: down
ifAdminStatus is provisioned as *up*, the component is unlocked	operational: enabled usage: active snmpOperStatus: up

ATM MPE soft PVC component states

ATM connection status is independent of the IP datapath status. If you lock the *AtmMpe* or *AtmMpe Ac* component, the soft PVC will not be torn down, but the IP forwarding will be disabled for the affected connections.

The table AtmConnection component states (page 247) shows the status of IP forwarding for the ATM MPE media in relation to the soft PVC and *AtmMpe Ac* component status.

AtmConnection component states

Soft PVC state (AtmMpe Ac SrcPvc/ DstPvc)	AtmMpe Ac		IP forwarding status
	OSI admin state	operationalState	
inactive	locked	disabled	disabled
inactive	unlocked	disabled	disabled
active	locked	disabled	disabled
active	unlocked	enabled	enabled

IP forwarding is not enabled immediately after the soft PVC becomes active, but the active state triggers IP forwarding to become enabled.

Monitoring the AtmMpe component

The following section describes how to display configuration information and operational statistics for the *AtmMpe* component.

Monitoring the AtmMpe component

Action	Command	Legend
List all the AtmMpe instances on a node	`list AtmMpe/*`	
Display the status of an AtmMpe instance	`display AtmMpe/<n>`	<n> is the number of the AtmMpe instance
Display the attributes configured under the AtmMpe component	`display -p AtmMpe/<n>`	
Display the status and attributes of a specific AtmMpe component instance using the ifTable	`display Vr/<a> Ift/<b>`	<a> is the number of the virtual router <b> is the number of the ifTable
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the AtmConnection subcomponent

The following section describes how to display configuration information and operational statistics for the *AtmMpe AtmConnection* subcomponent.

Attention: If you lock the *atmConnection* component at one end of an ATM MPE VCC, IP traffic destined for that VCC from the remote end is discarded. The remote VR does not receive notification that the VCC is out of service, and continues to transmit traffic on the VCC even though it is not operational.

Monitoring the AtmConnection subcomponent

Action	Command	Legend
List all the AtmConnection instances under the AtmMpe component	`list AtmMpe/<n> AtmConn/*`	<n> is the number of the AtmMpe instance
Display the status of an AtmConnection instance	`display AtmMpe/<n> AtmConn/<m>`	<n> is the number of the AtmMpe instance <m> is the number of the AtmConn instance
Display the attributes configured under the AtmMpe component	`display -p AtmMpe/<n> AtmConn/<m>`	
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the IlsFwdr component

The following section describes how to display configuration information and operational statistics for the *Lp IlsFwdr* component.

Monitoring the IlsFwdr component

Action	Command	Legend
Display the status of an IlsFwdr component	`display Lp/<n> IlsFwdr/<m>`	<n> is the number of the Lp <m> is the number of the IlsFwdr component
Display the attributes configured under the IlsFwdr component	`display -p Lp/<n> IlsFwdr/<m>`	
Attention: -p enables you to list provisionable components while in operational mode		

Testing ATM MPE soft PVC connectivity

The following section describes how to verify connectivity in an ATM MPE soft PVC. Check the state attribute of the *SrcPvc* and *DstPvc* components at either end of the soft PVC. If a soft PVC has been established, both endpoints are in the active state. If the soft PVC has failed, both of the endpoints are in the inactive state. In the case of failure, you can also determine the cause of the connection teardown and check the number of subsequent setup attempts.

Monitoring AtmMpe soft PVC connectivity

Action	Command	Legend
Display the status of the calling endpoint of this soft PVC.	`display AtmMpe/<n> Ac/<m> SrcPvc state`	<n> is the number of the AtmMpe instance <m> is the number of the AtmCon instance
Display the status of the called endpoint of this soft PVC.	display AtmMpe/<n> Ac/<m> DstPvc state	
Display the reason for the last teardown of the connection at the calling endpoint of this soft PVC.	display AtmMpe/<n> Ac/<m> SrcPvc lastClearCause	
Display the reason for the last teardown of the connection at the called endpoint of this soft PVC.	display AtmMpe/<n> Ac/<m> DstPvc lastClearCause	
Display the number of times the calling endpoint has attempted to reestablish the connection.	display AtmMpe/<n> Ac/<m> SrcPvc retryCount	

Testing ATM MPE soft PVC data flow

The following section describes how to test the flow of data through a soft PVC by monitoring the *AtmMpe* and *AtmIf* component statistics.

Monitoring AtmMpe soft PVC data flow

Action	Command	Legend
Display the outPackets and outOctets attributes at the calling endpoint of the soft PVC.	`display AtmMpe/<n> Ac/<m>`	<n> is the number of the AtmMpe instance <m> is the number of the AtmConn instance
Display the inPackets and inOctets attributes at the called endpoint of the soft PVC.	display AtmMpe/<n> Ac/<m>	
(1 of 2)		

Monitoring AtmMpe soft PVC data flow (continued)

Action	Command	Legend
Display the txCell count at each AtmIf component in the path of the soft PVC.	display AtmIf/<i> Vcc/<vc> Statistics	<i> is the number of the AtmIf instance <vc> is the instance value of the VCC
Display the rxCell count at each AtmIf component in the path of the soft PVC.	display AtmIf/<i> Vcc/<vc> Statistics	
(2 of 2)		

Clearing or optimizing an ATM MPE soft PVC

The following section describes how to tear down an active ATM MPE soft PVC connection using the clear command. The clear command also resets the *retryCount* attribute of the calling endpoint.

One reason for clearing an active connection is to optimize the path of the soft PVC. If a path better than the current one becomes available, the soft PVC does not take it automatically. You can clear the soft PVC to force it to take a new path. When the soft PVC attempts to reestablish the connection, the PNNI signaling network will find the better path.

Clearing the AtmMpe soft PVC

Action	Command	Legend
Clear the ATM MPE soft PVC at the calling end of the connection.	`clear AtmMpe/<n> Ac/<m> SrcPvc`	<n> is the number of the AtmMpe instance <m> is the number of the AtmConn instance
Clear the ATM MPE soft PVC at the called end of the connection.	clear AtmMpe/<n> Ac/<m> DstPvc	

Monitoring the frame relay DTE configuration

This section contains the information you need to monitor and maintain the IP over frame relay DTE configuration. Issue all commands in operational mode. See Operational mode (page 12).

For information on specific components and protocols, see the following sections:

- Frame relay DTE component states (page 252)
- Frame relay DTE remote group component states (page 253)
- Frame relay DTE data link connection identifier component states (page 253)
- Monitoring the FrDte component (page 254)
- Monitoring the StDlci and Dlci subcomponent (page 255)
- Monitoring the Rg subcomponent (page 255)

Frame relay DTE component states

The table Frame relay DTE component states (page 252) lists the operational states reported by the frame relay DTE service.

Frame relay DTE component states

Condition	States reported
The *FrDte* component is waiting for frame relay interface or the hardware interface to become available.	operational: disabled usage: idle administrative: unlocked
The *FrDte* component is ready to provide service, but currently there is no DLCI defined on the interface.	operationalState: enabled usageState: idle administrative: unlocked
A lock command is in effect. The relationship between this component and the frame relay interface or LP does not exist	operational: disabled usage: idle administrative: locked
The component is in use, i.e. there is at least one active DLCI defined on the interface.	operational: enabled usage: active administrative: unlocked
No new DLCI is available, i.e. total number of DLCIs in use >=1024.	operational: enabled usage: busy administrative: unlocked

Frame relay DTE remote group component states

The table Frame relay DTE Rg component states (page 253) lists the operational states reported by the frame relay DTE remote group (Rg) component.

Frame relay DTE Rg component states

Condition	States reported
The component is disabled because the corresponding FrDte is locked	operational: disabled usage: idle administrative: unlocked
The remote group is ready to provide service, but currently there is no DLCI linked to it.	operational: enabled usage: idle administrative: unlocked
This remote group is in use, i.e. there is at least one DLCI in use linked to this remote group.	operational: enabled usage: active administrative: unlocked
No new DLCI is available, i.e. total number of DLCIs in use >= 1024.	operational: enabled usage: busy administrative: unlocked
The lock command is in effect and the linked protocol port is disabled or the corresponding FrDte is locked.	operational: disabled usage: idle administrative: locked
The lock command is in effect but the linked protocol port is enabled or the corresponding FrDte is unlocked.	operational: enabled usage: idle administrative: locked

Frame relay DTE data link connection identifier component states

The table Frame relay DTE Dlci component states (page 254) lists the operational states reported by the frame relay DTE data link connection identifier (*Dlci*) component.

Frame relay DTE Dlci component states

Condition	States reported
The DLCI is disabled because the corresponding remote group is locked or disabled, or the link is down.	operational: disabled usage: idle administrative: unlocked
This DLCI is ready to provide service.	operationalState: enabled usageState: active administrative: unlocked
The lock command is in effect. Also the DLCI is disabled because the corresponding remote group is locked or disabled, or the link is down.	operational: disabled usage: idle administrative: locked
The lock command is in effect, but the corresponding remote group is unlocked and enabled.	operational: enabled usage: idle administrative: locked

Monitoring the FrDte component

The following section describes how to display configuration information and operational statistics for the *FrDte* component.

Monitoring the FrDte component

Action	Command	Legend
List all the FrDte instances on a node	`list FrDte/*`	
Display the status of an FrDte instance	`display FrDte/<n>`	<n> is the number of the FrDte instance
Display the attributes configured under the FrDte component	`display -p FrDte/<n>`	
Display the status and attributes of a specific FrDte component instance using the ifTable	`display Vr/<a> Ift/<b>`	<a> is the number of the virtual router <b> is the number of the ifTable
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the StDlci and Dlci subcomponent

The following section describes how to display configuration information and for the *FrDte StDlci* subcomponent and operational statistics for the *FrDte Dlci* subcomponent.

Monitoring the StDlci and Dlci subcomponents

Action	Command	Legend
List all the StDlci instances under the FrDte component	`list FrDte/<n> StDlci/*`	<n> is the number of the FrDte instance
List all the Dlci instances under the FrDte component	`list FrDte/<n> Dlci/*`	<n> is the number of the FrDte instance
Display the status of an StDlci instance	`display FrDte/<n> StDlci/<m>`	<n> is the number of the FrDte instance <m> is the number of the StDlci instance
Display the attributes configured under the StDlci subcomponent	`display -p FrDte/<n> StDlci/<m>`	
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the Rg subcomponent

The following section describes how to display configuration information and operational statistics for the *FrDte Rg* subcomponent.

Monitoring the Rg subcomponent

Action	Command	Legend
List all the Rg instances under the FrDte component	`list FrDte/<n> Rg/*`	<n> is the number of the FrDte instance
Display the status of an Rg instance	`display FrDte/<n> Rg/<m>`	<n> is the number of the FrDte instance <m> is the number of the Rg instance
Display the attributes configured under the Rg subcomponent	`display -p FrDte/<n> Rg/<m>`	
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the Ethernet access media configuration

This section contains the information you need to monitor and maintain the Ethernet access media configuration. Issue all commands in operational mode. See Operational mode (page 12).

Monitoring the LanApplication component

The following section describes how to display configuration information and operational statistics for the *LanApplication* (La) component.

Action	Command	Legend
Display the La component	`display La/<v>`	<v> is the number of the *LanApplication* instance
Display the physical and protocol ports that are linked to the La component	`display -p La/<v>`	<v> is the number of the *LanApplication* instance
Display the status of the physical port	`display -p La/<v> Framer`	<v> is the number of the *LanApplication* instance
Display the Ethernet statistics for the La component	`display La/<v> ethernetStatistics`	<v> is the number of the *LanApplication* instance
Display the VLAN component	`display La/<v> vlan/<w>`	<v> is the number of the *LanApplication* instance <w> is the number of the vlan instance
Display the physical port that is linked to the VLAN component	`display -p La/<v> vlan/<w>`	<v> is the number of the *LanApplication* instance <w> is the number of the vlan instance
Display the Ethernet statistics for the VLAN component	`display La/<v> vlan/<w> ethernetStatistics`	<v> is the number of the *LanApplication* instance <w> is the number of the vlan instance
Display the card and all its physical ports	`display lp/<x> ethernet/*`	<x> is the number of the logical processor instance
Display all the protocol ports for a specific Vr	`display Vr/<y> pp/*`	<y> is the name of the Vr
(1 of 2)		

Action	Command	Legend
Display the statistics for a particular protocol port	`display Vr/<y> IfTableEntry/<z>`	<y> is the name of the Vr <z> is the ifindex related to a particular protocol port
Display the statistics for all protocol ports under a VR	`display Vr/<y> IfTableEntry/*`	<y> is the name of the Vr
(2 of 2)		

Monitoring the PPP configuration

This section contains the information you need to monitor and maintain the IP over PPP configuration. Issue all commands in operational mode. See Operational mode (page 12).

For information on specific components and protocols, see the following sections:

- PPP component states (page 258)
- Monitoring the Ppp component (page 258)
- Monitoring the Link subcomponent (page 259)
- Monitoring the Lqm subcomponent (page 259)
- Monitoring the Leq subcomponent (page 260)

PPP component states

The table PPP component states (page 258) lists the operational states reported by the PPP service.

PPP component states

Condition	States reported
A lock -force operator command is in effect.	operational: disabled usage: idle administrative: locked
The component is unlocked. PPP is not in the Enabled operational state.	operationalState: disabled usageState: idle administrative: unlocked
The component is unlocked and in service. PPP is in the Enabled operational state.	operational: enabled usage: busy administrative: unlocked
PPP is going from the Unlocked state to the Locked state. It is in the process of an orderly shutdown.	operational: enabled usage: busy administrative: shutting down

Monitoring the Ppp component

The following section describes how to display configuration information and operational statistics for the *Ppp* component.

Monitoring the Ppp component

Action	Command	Legend
List all the ppp instances on a node	`list Ppp/*`	
Display the status of a ppp instance	`display Ppp/<n>`	<n> is the number of the Ppp instance
Display the attributes configured under the ppp component	`display -p Ppp/<n>`	
Display the status and attributes of a specific ppp component instance using the ifTable	`display Vr/<a> Ift/<b>`	<a> is the number of the virtual router <b> is the number of the ifTable
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the Link subcomponent

The following section describes how to display configuration information and operational statistics for the *Ppp Link* subcomponent.

Monitoring the Link subcomponent

Action	Command	Legend
Display the status of a link instance	`display Ppp/<n> Link`	<n> is the number of the Ppp instance
Display the attributes configured under the link subcomponent	`display -p Ppp/<n> Link`	
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the Lqm subcomponent

The following section describes how to display configuration information and operational statistics for the *Ppp Lqm* subcomponent.

Monitoring the Lqm subcomponent

Action	Command	Legend
Display the status of an Lqm instance	`display Ppp/<n> Lqm`	<n> is the number of the Ppp instance
Display the attributes configured under the Lqm subcomponent	`display -p Ppp/<n> Lqm`	
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the Leq subcomponent

The following section describes how to display configuration information and operational statistics for the *Ppp Leq* subcomponent.

The *Ppp Leq* subcomponent is only available on a Nortel Networks Multiservice Switch 7400 nodes with SBIC-based FPs.

Monitoring the Leq subcomponent

Action	Command	Legend
Display the status of an Leq instance	`display Ppp/<n> Leq`	<n> is the number of the Ppp instance
Display the attributes configured under the Leq subcomponent	`display -p Ppp/<n> Leq`	
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the IP and virtual router configuration

You can perform the tests described in this section to determine if you have configured IP and virtual routers properly. Issue all commands in operational mode. See Operational mode (page 12). For more information, see the following sections:

- Monitoring the IP component (page 261)
- Monitoring the IpInterfaceEntry subcomponent (page 262)
- Monitoring the IP cache subcomponent (page 263)
- Monitoring the ICMP subcomponent (page 264)
- Testing connectivity using the ICMP subcomponent (page 264)
- Monitoring the TCP subcomponent (page 266)
- Monitoring the UDP subcomponent (page 266)
- Monitoring the IpPort component (page 267)
- Monitoring the Arp subcomponent (page 268)
- Monitoring the BootpPort component (page 269)
- Monitoring the RelayBroadCast subcomponent (page 269)

Monitoring the IP component

The following section describes how to lock and unlock components, display configuration information and operational statistics, and test connectivity for the *Ip* component and its subcomponents.

For more information, see the following tables:

- Locking and unlocking the IP component (page 262)
- Monitoring the IP component (page 262)

CAUTION
Provisioning session termination
The Ip component can be locked. However, this disables access to IP and all of its subcomponents. It also makes it impossible to further provision using IP applications such as telnet, SNMP, or FTP. IP datagrams are not forwarded while the Ip component is locked. Locking of IP and its related protocols is immediate when you issue the lock command.

Locking and unlocking the IP component

Action	Command	Legend
Lock the Ip component. Locking takes the IP component and all of its configured or dynamic subcomponents out of service.	`lock Vr/<vr_name> Ip`	<vr_name> is the name of the virtual router
Unlock the Ip component. Unlocking returns the Ip component and all of its configured or dynamic subcomponents to service.	`unlock Vr/<vr_name> Ip`	

Monitoring the IP component

Action	Command	Legend
List the configured components of the IP service.	`list -p Vr/<vr_name> Ip`	<vr_name> is the name of the virtual router
Display the configured attributes of the IP component.	`display -p Vr/<vr_name> Ip`	
Display operational attributes and statistics for an IP component	`display Vr/<vr_name> Ip`	
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the IpInterfaceEntry subcomponent

The following section describes how to display operational statistics for the *IpInterfaceEntry* (*If*) subcomponent. For more information, see the table Monitoring the IpInterfaceEntry subcomponent (page 263).

Monitoring the IpInterfaceEntry subcomponent

Action	Command	Legend
Display operational statistics of the IpInterfaceEntry (if) subcomponent	`display Vr/<vr_name> Ip IpInterfaceEntry/ <Ipaddr>`	<vr_name> is the name of the virtual router <Ipaddr> is the 32-bit IP address assigned to the IP interface
Attention: The *hardwareAddress* attribute displays the MAC address in canonical format. The *ncHardwareAddress* attribute displays the MAC address in non-canonical format.		

Monitoring the IP cache subcomponent

The *Cache* component is a dynamic subcomponent of the *Ip* component. It represents the IP cache table on an LP. You can list *Cache* component instances and display operational statistics.

For more information, see the following tables:

- Monitoring the IP cache subcomponent (page 263)
- Clearing IP cache table entries (page 263)

Monitoring the IP cache subcomponent

Action	Command	Legend
List all *Cache* component instances	`list Vr/<vr_name> Ip Cache/*`	<vr_name> is the name of the virtual router
Display operational statistics for a particular *Cache* component instance	`display Vr/<vr_name> Ip Cache/<lp_id>`	

Clearing IP cache table entries

Action	Command	Legend
Clear all entries from the IP cache table	`clear vr/<vr_name> Ip Cache/<lp_id>`	<lp_id> is the instance value assigned in the IP subcomponent to a particular LP

Monitoring the ICMP subcomponent

Icmp is a subcomponent of the *Ip* component responsible for processing the internet control message protocol (ICMP). ICMP sends control messages to source hosts to indicate special conditions. For more information, see the table Monitoring the ICMP subcomponent (page 264).

You can use the ping (packet internet groper) command to test connections to other IP nodes. The ping command issues an echo request to the specified address and then compares the response to the request. For more information see Testing connectivity using the ICMP subcomponent (page 264).

Monitoring the ICMP subcomponent

Action	Command	Legend
Display the ICMP provisionable attributes	`display -p Vr/<vr_name> Ip Icmp`	<vr_name> is the name of the virtual router
Display the ICMP subcomponent statistics	`display Vr/<vr_name> Ip Icmp`	

Testing connectivity using the ICMP subcomponent

The *Icmp* component uses the *ping* verb to test connectivity. *Ping* sends an echo request to the specified IP node, compares the reply with the request, and sends a message to the console indicating the result. ICMP packets can range from 64 to 5000 bytes in length.

Attention: If you are transmitting ICMP packets over an ATM VCC with a CBR service category, the default transmission queue size for the VCC limits the packet size to 3680 bytes. To ensure that ICMP packets are not discarded, reduce the maximum IP packet size to 3600 bytes on ATM VCCs with a service category of CBR. ICMP packets are not discarded if the VCC has a service category of UBR.

The retry option specifies the number of times to ping a specific IP address until an echo packet is returned. If after x pings no echo packet is returned, ping reports that the node is not responding.

The continuous option allows the operator to continuously ping an IP address or a range of IP addresses.

The netmask option, also called performing a ping sweep, allows the operator to ping a range of addresses specified by the -Ipaddr and -netmask options.

The table Testing connectivity using the ICMP subcomponent (page 265) specifies the commands you can use to test connectivity.

Testing connectivity using the ICMP subcomponent

Action	Command	Legend
View commands related to the ping verb for the Icmp component	`help -v(ping) vr ip icmp`	
Ping an IP interface	`ping -Ipaddr(<IpAddress>) -size(<pingSize>) Vr/<vr_name> Ip Icmp`	<IpAddress> is the 32-bit address of the interface to be pinged <pingSize> is the *Icmp* packet payload (30 to 20000 bytes) carried in the echo request. If you do not choose this option, the default ping size is 64 bytes. <vr_name> is the name of the virtual router
Execute IP traceRoute on a Multiservice Switch node	`ping -Ipaddr(<IpAddress>) -size(<pingSize>) -traceRoute Vr/<vr_name> Ip Icmp`	-traceRoute is the option to execute IP trace route
Set the retry option	`ping -Ipaddr(<IpAddress>) -size(<pingSize>) -retry(<retry_number>) Vr/<vr_name> Ip Icmp`	<retry_number> is the number of times to ping the node until it returns an echo packet. The range is 0-9.
Ping a specific address	`ping -Ipaddr(<IpAddress>) -continuous Vr/<vr_name> Ip Icmp`	-netmask is the network mask used with the IP address
Ping a range of addresses	`ping -Ipaddr(<IpAddress>) -continuous -netmask(<IpAddress>) Vr/<vr_name> Ip Icmp`	
(1 of 2)		

Testing connectivity using the ICMP subcomponent (continued)

Action	Command	Legend
Set the netmask option	`ping -Ipaddr(<IpAddress>) -netmask(<IpAddress>) Vr/<vr_name> Ip Icmp`	
Stop a continuous ping, or a ping sweep (the netmask option)	`ping -stop Vr/<vr_name> Ip Icmp`	
Attention: You can also stop a continuous ping or a ping sweep by typing Ctrl-C.		
Support ping through certain local IP interfaces	`ping -src(<IpAddress>) Vr<vr_name> Ip Icmp`	
(2 of 2)		

Monitoring the TCP subcomponent

The following tables describe how to display TCP provisioning, statistics, and connections:

- Monitoring the TCP subcomponent (page 266)
- Displaying the TCP connection table (page 266)

Monitoring the TCP subcomponent

Action	Command	Legend
Display *Tcp* subcomponent statistics	`display Vr/<vr_name> Ip Tcp`	<vr_name> is the name of the virtual router

Displaying the TCP connection table

Action	Command	Legend
Display the TCP connection table	`display Vr/<vr_name> Ip Tcp Tcpentry/*`	<vr_name> is the name of the virtual router

Monitoring the UDP subcomponent

The following tables describe how to display user datagram protocol (UDP) statistics and listen tables:

- Monitoring the UDP subcomponent (page 267)
- Displaying the UDP listen table (page 267)

Monitoring the UDP subcomponent

Action	Command	Legend
Display *Udp* subcomponent statistics	`display Vr/<vr_name> Ip Udp`	<vr_name> is the name of the virtual router

Displaying the UDP listen table

Action	Command	Legend
Display the UDP listen table	`list Vr/<vr_name> Ip Udp listen/*`	<vr_name> is the name of the virtual router

Monitoring the IpPort component

The following tables describe how to lock, unlock, and display provisioning and operational statistics for the *IpPort* component:

- Locking and unlocking the IpPort component (page 267)
- Monitoring the IpPort component (page 268)

CAUTION

Locking IP port stops IP routing

The IpPort component can be locked. However, this stops IP routing on the locked port. IP datagrams are not forwarded through that port while the IpPort component is locked. Locking of the IpPort component is immediate when you issue the lock command.

Locking and unlocking the IpPort component

Action	Command	Legend
Lock the IP port	`lock Vr/<vr_name> ProtocolPort/ <pp_name> IpPort`	<vr_name> is the name of the virtual router
Unlock the IP port	`unlock Vr/<vr_name> ProtocolPort/ <pp_name> IpPort`	

Monitoring the IpPort component

Action	Command	Legend
Display IpPort LogicalIf addresses	`display -p Vr/<vr_name> ProtocolPort/<pp_name> IpPort LogicalIf/*`	<vr_name> is the name of the virtual router <pp_name> is the name of the protocol port
Display the *IpPort* component information	`display Vr/<vr_name> Pp/<pp_name> IpPort`	

Monitoring the Arp subcomponent

The following tables describe how to monitor and clear ARP entries:

- Monitoring the Arp subcomponent (page 268)
- Clearing ARP table dynamic host entries (page 269)

Monitoring the Arp subcomponent

Action	Command	Legend
Display the ARP static host entries in canonical form.	`display Vr/<vr_name> Ip Arp HostEntry/*`	<vr_name> is the name of the virtual router
Display the ARP static host entries in non-canonical form.	`display -noTabular Vr/<vr_name> Ip Arp HostEntry/*`	
Display the ARP dynamic host entries in canonical form.	`display Vr/<vr_name> Ip Arp DynHostEntry/*`	
Display the ARP dynamic host entries in non-canonical form.	`display -noTabular Vr/<vr_name> Ip Arp DynHostEntry/*`	
Attention: The *physAddress* attribute displays the MAC address in canonical form. To view the MAC address in non-canonical form, specify the *ncPhysAddress* attribute in the display command.		

Clearing ARP table dynamic host entries

Action	Command	Legend
Clear all *DynamicHostEntries* in the ARP table.	`clear vr/<vr_name> ip arp`	<vr_name> is the name of the virtual router
Clear the *DynamicHostEntries* learned from a specific Ip port that match the specified IP address.	`clear vr/<vr_name> ip arp dyn/<Ip_address>`	
Clear all *DynamicHostEntries* learned from a specific Ip port.	`clear -log(<Ip_address>) vr/<vr_name> ip arp`	

Monitoring the BootpPort component

The *BootpPort* component contains information about the provisioned BOOTP ports under the *IpPort* component. The table Monitoring the BootpPort component (page 269) describes how to display BOOTP port statistics.

Monitoring the BootpPort component

Action	Command	Legend
Display BOOTP port statistics using the *BootpPort* component	`display Vr/<vr_name> pp/ <protocolPort_name> IpPort BootpP`	<vr_name> is the name of the virtual router

Monitoring the RelayBroadCast subcomponent

The table Monitoring the RelayBroadCast subcomponent (page 270) describes how to display RelayBroadCast statistics.

Monitoring the RelayBroadCast subcomponent

Action	Command	Legend
Display the *RelayBroadCast* subcomponent global broadcast counter	`display Vr/<vr_name> Ip RelayBC`	<vr_name> is the name of the virtual router
Display the *RelayBroadCast* subcomponent counters on individual interfaces	`display Vr/<vr_name> ProtocolPort/ <pp_name> IpPort`	<pp_name> is the name of the protocol port

Monitoring the IP routing management configuration

Nortel Networks Multiservice Switch nodes store routing information in the IP forwarding table and the routing database. The forwarding table provides information on the routes with the best metric. The routing database provides all sources of routing information. This section describes how to monitor the IP forwarding table and the routing database.

For more information, see the following tables:

- Monitoring the Ip ForwardTable component (page 271)
- Monitoring the Ip RouteDataBaseEntry component (page 273)

Monitoring the Ip ForwardTable component

Action	Command	Legend
Display the forwarding table	`display Vr/<vr_name> Ip Fwd/*,*,*,*`	<vr_name> is the wildcard (*), to display the routes for all virtual routers or the name of a virtual router, to display the routes for only that virtual router
(1 of 2)		

Monitoring the Ip ForwardTable component (continued)

Action	Command	Legend
Display the routes in the forwarding table covered by a particular IP address or subnet	`display Vr/<vr_name> Ip Fwd/ <a>.<b>.<c>.<d>,*,*,*`	<vr_name> is the wildcard (*), to display the routes for all virtual routers or the name of a virtual router, to display the routes for only that virtual router <a>.<b>.<c>.<d> are the octets in an IP address. They can be numerals or, in the case of a subnet, the wildcard (*). For example, 47.138.*.*
Display the routes in the forwarding table covered by a particular IP address or subnet	`display Vr/<vr_name> Ip Fwd/ <a>[.<b>.<c>.<d>],*,*,*`	<vr_name> is the wildcard (*), to display the routes for all virtual routers or the name of a virtual router, to display the routes for only that virtual router <a> is the first octet in an IP address. It can be numerals or, for all routes, the wildcard (*). [.<b>.<c>.<d>] are the remaining octets in an IP address. They can be numerals or wildcards (*). Once you enter a wildcard, you do not need to complete the address. For example, * instead of *.*.*.*; 47.138.* instead of 47.138.*.*
(2 of 2)		

Monitoring the Ip RouteDataBaseEntry component

Action	Command	Legend
Display the *Ip RouteDataBaseEntry* component	`display Vr/<vr_name> Ip Rdb/*`	<vr_name> is the name of the virtual router
Display the routes in the routing table covered by a particular IP address or subnet for a particular routing protocol	`display Vr/<vr_name> Ip Rdb/ <a>[.<b>.<c>.<d>],*,*,*(protocol=<pro_name>)`	<vr_name> is the wildcard (*), to display the routes for all virtual routers or the name of a virtual router, to display the routes for only that virtual router <a> is the first octet in an IP address. It can be numerals or, for all routes, the wildcard (*). [.<b>.<c>.<d>] are the remaining octets in an IP address. They can be numerals or wildcards (*). Once you enter a wildcard, you do not need to complete the address. For example, * instead of *.*.*.*; 47.138.* instead of 47.138.*.* <pro_name> is the name of the routing protocol. For example, *ospf*.

Monitoring the virtual media configuration

The *VirtualMedia* (*Vm*) component is an optional root component that you can provision to provide connectivity between VRs or to create an always-up IP interface for RIP, OSPF, or BGP-4. The *Vm* component has no provisionable or operational attributes. The *Vm* component has one subcomponent, the *Interface* (*If*) component.

For more information, see the following tables:

- Locking and unlocking the Vm If component (page 274)
- Monitoring the Vm component (page 274)
- Monitoring the Vm If component (page 275)

Locking and unlocking the Vm If component

Action	Command	Legend
Lock an instance of the *Vm If* component.	`lock Vm/<m> If/<n>`	<m> is the instance value of the *Vm* component
Unlock an instance of the *Vm If* component.	`unlock Vm/<m> If/<n>`	
Attention: Only the *Vm If* component can be locked. The *Vm* component cannot be locked.		

Monitoring the Vm component

Action	Command	Legend
Display all *Vm* component instances	`display Vm/*`	
Display specific *Vm* component instances	`display Vm/<m>`	<m> is the instance value of the *Vm* component

Monitoring the Vm If component

Action	Command	Legend
Display all *If* component instances for a *Vm* component in tabular format.	`display Vm/<m> If/*`	
Display all *If* component instances for a *Vm* component in non-tabular format.	`display -noTabular Vm/<m> If/*`	
Display all the operational attributes for all instances of the *Vm* and *If* components	`display -o Vm/* If/*`	
Display all the operational attributes for a specific instance of the *Vm* component and all instances of its *If* component	`display -o Vm/<m> If/*`	

Monitoring the RIP configuration

The following tables describe how to lock and unlock the *Ip Rip* component, monitor the *Ip Rip* and *Ip Rip If* components, and monitor RIP import and export policy. Issue all commands in operational mode. See Operational mode (page 12).

- Locking and unlocking the Rip component (page 276)
- Monitoring the Rip component (page 276)
- Monitoring the Rip If component (page 277)
- Monitoring RIP import and export policy (page 277)

Locking and unlocking the Rip component

Action	Command	Legend
Lock the *Ip Rip* component to take it and all related subcomponents out of service.	`lock Vr/<vr_name> Ip Rip`	<vr_name> is the name of the virtual router
Unlock the *Ip Rip* component to return it all related subcomponents to service.	`unlock Vr/<vr_name> Ip Rip`	<vr_name> is the name of the virtual router

Monitoring the Rip component

Action	Command	Legend
Display configurable attributes for the *Rip* component	`display -p Vr/<vr_name> Ip Rip`	<vr_name> is the name of the virtual router
Display operational statistics for the *Rip* component	`display Vr/<vr_name> Ip Rip`	

Monitoring the Rip If component

Action	Command	Legend
Display configurable attributes for the *RipIf* component	`display -p Vr/<vr_name> Pp/<pp_name> IpPort LogicalIf/<ipAddress> RipIf`	<vr_name> is the name of the virtual router
Display operational attributes for the *Rip If* component	`display -p Vr/<vr_name> Pp/<pp_name> IpPort LogicalIf/<ipAddress> RipIf`	<vr_name> is the name of the virtual router

Monitoring RIP import and export policy

Action	Command	Legend
Display a RIP import policy	`display Vr/<vr_name> Ip Rip Import/<import_id>`	<vr_name> is the name of the virtual router <import_id> is the instance/identifier of the particular import policy
Display a RIP export policy	`display Vr/<vr_name> Ip Rip Export/<export_id>`	<vr_name> is the name of the virtual router <export_id> is the instance/identifier of the particular export policy
Attention: To display all import or export polices, use an asterisk (*) as the instance value.		

Monitoring the OSPF configuration

The following tables describe how to lock and unlock the Ip Ospf component, monitor the Ospf and OspfIf components and subcomponents, and monitor the OSPF export policy. Issue all commands in operational mode. See Operational mode (page 12).

- Locking and unlocking the Ospf component (page 278)
- Monitoring the Ospf component (page 278)
- Monitoring the OspfIf component (page 279)
- Monitoring OSPF export policy (page 279)
- Monitoring OSPF areas (page 279)
- Monitoring OSPF hosts (page 279)
- Monitoring OSPF virtual links (page 280)
- Monitoring OSPF stub areas (page 280)
- Monitoring OSPF neighbors (page 280)
- Monitoring the OSPF link state database (page 280)

Locking and unlocking the Ospf component

Action	Command	Legend
Lock the Ospf component to take it and all related subcomponents out of service.	`lock Vr/<vr_name> Ip Ospf`	<vr_name> is the name of the virtual router
Unlock the Ospf component to return it and all related subcomponents to service.	`unlock Vr/<vr_name> Ip Ospf`	<vr_name> is the name of the virtual router

Monitoring the Ospf component

Action	Command	Legend
Display the Ip Ospf subcomponent operational attributes.	`display Vr/<vr_name> Pp/<pp_name> IpPort logicalIf/<IpAddress> OspfIf`	<vr_name> is the name of the virtual router <pp_name> is the name of the protocol port <IpAddress> is the 32-bit Ip address of the interface

Monitoring the OspfIf component

Action	Command	Legend
Display a particular OspfIf component (under the IpPort LogicalIf component).	display -p Vr/ <vr_name> Ip Ospf ProtocolPort/ <pp_name> IpPort LogicalIf/ <IpAddress> OspfIf	<vr_name> is the name of the virtual router <pp_name> is the name of the protocol port <IpAddress> is the 32-bit Ip address of the interface

Monitoring OSPF export policy

Action	Command	Legend
Display all OSPF export policies.	display -p Vr/ <vr_name> Ip Ospf export/*	<vr_name> is the name of the virtual router

Monitoring OSPF areas

Action	Command	Legend
Display all provisioned OSPF areas attached to this Vr component instance.	display -d Vr/ <vr_name> Ip Ospf AreaEntry/*	<vr_name> is the name of the virtual router
Display all OSPF aggregate areas.	display -d Vr/ <vr_name> Ip Ospf aggregate/*	

Monitoring OSPF hosts

Action	Command	Legend
display Ip OSPF provisioned host entries	display -d Vr/ <vr_name> Ip Ospf host/*	<vr_name> is the name of the virtual router

Monitoring OSPF virtual links

Action	Command	Legend
Display all provisioned OSPF virtual links.	`display Vr/<vr_name> Ip Ospf virtif/*`	<vr_name> is the name of the virtual router
Display OSPF virtual link operational data.	`display Vr/<vr_name> Ip Ospf virtif/*`	

Monitoring OSPF stub areas

Action	Command	Legend
Display OSPF stub areas.	`display -d Vr/<vr_name> Ip Ospf stub/*`	<vr_name> is the name of the virtual router

Monitoring OSPF neighbors

Action	Command	Legend
Display all OSPF neighbors that have been provisioned or dynamically discovered using the OSPF hello protocol.	`display Vr/<vr_name> Ip Ospf neighbor/*`	<vr_name> is the name of the virtual router

Monitoring the OSPF link state database

Action	Command	Legend
Display the OSPF link state database.	`display Vr/<vr_name> Ip Ospf Lsdb/*`	<vr_name> is the name of the virtual router
Display the OSPF external link state database.	`display Vr/<vr_name> Ip Ospf extLsdb/*`	

Monitoring the BGP-4 configuration

The following tables describe how to lock and unlock the *bgp* component, display BGP operational statistics, and display BGP routes in the databases. Issue all commands in operational mode. See Operational mode (page 12).

- Locking and unlocking the Bgp and Bgp Peer components (page 281)
- Monitoring the Bgp and Bgp Peer components (page 281)
- Monitoring routes in the BGP routing information base (RIB) (page 282)

Locking and unlocking the Bgp and Bgp Peer components

Action	Command	Legend
Disable a BGP-4 instance for a particular virtual router (VR).	`lock Vr/<vr_name> Ip Bgp`	<vr_name> is the name of the virtual router
Put the BGP-4 instance back in service.	`unlock Vr/<vr_name> Ip Bgp`	
Shut down a BGP-4 peer connection.	`lock Vr/<vr_name> Ip Bgp Peer/<peer>`	
Re-establish the BGP-4 peer connection.	`unlock Vr/<vr_name> Ip Bgp Peer/<IP_address>`	

Monitoring the Bgp and Bgp Peer components

Action	Command	Legend
Display operational statistics for the BGP-4 instance.	`display Vr/<vr_name> Ip Bgp`	<vr_name> is the instance identifier of the parent VR.
List all of the subcomponents for the BGP-4 instance.	`list Vr/<vr_name> Ip Bgp`	
Display statistics for a BGP-4 peer under a BGP-4 instance.	`display Vr/<vr_name> Ip Bgp Peer/<peer>`	<peer> is the IP address of the BGP-4 peer. To specify all BGP-4 peers under the BGP-4 instance, enter *

Monitoring routes in the BGP routing information base (RIB)

Action	Command	Legend
Display information for one or more routes received from a BGP-4 peer.	`display Vr/<vr_name> Ip Bgp Indb/ <in_entry>`	<vr_name> is the instance identifier of the parent VR. <in_entry> specifies the input database entry. To specify all entries in the Indb, enter *.
Display information for one or more entries in the BGP-4 local database.	`display Vr/<vr_name> Ip Bgp Localdb/ <loc_entry>`	<loc_entry> specifies the local database entry. To specify all entries in the Localdb, enter *.
Display information for one or more entries in the BGP-4 output database.	`display Vr/<vr_name> Ip Bgp Outdb/ <out_entry>`	<out_entry> specifies the output database entry. To specify all entries in the Outdb, enter *.
Attention: The *inLocal* and *calcLocalPref* attributes apply to routes in the Indb only.		

Monitoring the BGP-4 MD5 Authentication

If a BGP Peer MD5 Authentication alarm is raised, the operator can isolate the source of the issue using the Peer operational attributes described in the following table:

- BGP Peer operational attributes (page 283)

You can display the attributes using the following command:

```
d -o Rtr/<rtr_id> Bgp Peer/<IP addr> Operational
```

During the key change transition, the counters will not increment so long as the packets received are authenticated with either the old key or the new key (including the cases where either the old or new key is the null key, implying a non-authenticated connection). Once both ends have exchanged packets authenticated with the new key, the counters will resume their normal behavior using only the currently provisioned key. Legitimate errors encountered during the key change transition will result in the counters incrementing as expected and alarms being generated as appropriate.

BGP Peer operational attributes

Attribute	Description
authenticationType	This attribute indicates which type of authentication is used on the TCP connection to the specified Peer. Values are none and md5.
noAuthenticationPkts	This attribute indicates the number of TCP packets have been received from the specified Peer which were unauthenticated when authentication was expected. When this counter is increasing it could mean that the Peer is not configured to use authentication or a hacker is attempting to disrupt the BGP connection and is injecting unauthenticated packets.
badAuthenticationPkts	This attribute indicates the number of TCP packets have been received from the specified Peer which were authenticated incorrectly. When this counter is increasing could mean that the Peer is configured with a different key or a hacker is attempting disrupt the BGP connection and is injecting packets authenticated using a different key.
unexpectedAuthenticationPkts	This attribute indicates the number of TCP packets have been received from the specified Peer which were authenticated when authentication was not expected. When this counter is increasing it could mean that the Peer is configured to use authentication or a hacker is attempting to disrupt the BGP connection and is injecting authenticated packets.

Monitoring the static route configuration

The following tables describe how to lock and unlock, and monitor the *Ip Static* component. Issue all commands in operational mode. See Operational mode (page 12).

- Locking and unlocking the Ip Static component (page 284)
- Monitoring the Ip Static component (page 284)

Locking and unlocking the Ip Static component

Action	Command	Legend
Lock the *Ip Static* component to prevent routing over static routes	`lock Vr/<vr_name> Ip Static`	<vr_name> is the name of the virtual router
Unlock the *Ip Static* component to put it back into a service-providing role	`unlock Vr/<vr_name> Ip Static`	

Monitoring the Ip Static component

Action	Command	Legend
Display the *Ip Static* component route entries	`display -p Vr/<vr_name> Ip Static Route/*`	<vr_name> is the name of the virtual router
Display *Ip Static* component discard route entries	`display -p Vr/<vr_name> Ip Static DiscardRouteEntry/*`	

Monitoring the IP multicast configuration

The following tables describe how to lock and unlock, and monitor the *Ip Mcast* component. Issue all commands in operational mode. See Operational mode (page 12).

- Monitoring IP multicast and IGMP (page 285)
- Monitoring PIM-SM (page 285)

Monitoring IP multicast and IGMP

Action	Command
Display the multicast cache.	`Display Vr/<vr_name> Ip Mcast Cache/<n>`
Display the *Mcast* component operational statistics.	`Display Vr/<vr_name> Ip Mcast`
Display all configured static routes	`Display Vr/<vr_name> Ip Mcast Static RouteEntry/*`
Lock the *Mcast* component	`Lock Vr/<vr_name> Ip Mcast`
Unlock the *Mcast* component	`Unlock Vr/<vr_name> Ip Mcast`
Display the *Igmp* component operational statistics	`Display Vr/<vr_name> Ip Mcast Igmp`
Display the IGMP group cache	`Display Vr/<vr_name> Ip Mcast Igmp Cache/*`
Lock the *Igmp* component	`Lock Vr/<vr_name> Ip Mcast Igmp`
Unlock the *Igmp* component	`Unlock Vr/<vr_name> Ip Mcast Igmp`

Monitoring PIM-SM

Action	Command
Display multicast forwarding table(s)	`Display Vr/<vr_name> Ip Mcast Fwd/*`
Display the outbound interface table(s) for the forwarding table	Display Vr/<vr_name> Ip Mcast Fwd/* Oif/*
Display a multicast PIM neighbor	`Display Vr/<vr_name> Ip Mcast PimNbr/*`
Display the PIM-SM Bootstrap router (BSR) and other operational information in a domain	Display Vr/<vr_name> Ip Mcast PimSm Domain/<n>

(1 of 2)

Monitoring PIM-SM (continued)

Action	Command
Display a multicast PIM-SM domain and RP-Set	`Display Vr/<vr_name> Ip Mcast PimSm Domain/<n> RpSet/*`
Lock the *PimSm* component	`Lock Vr/<vr_name> Ip Mcast PimSm`
Unlock the *PimSm* component	`Unlock Vr/<vr_name> Ip Mcast PimSm`
(2 of 2)	

Monitoring the virtual router redundancy protocol configuration

This section contains the information you need to monitor and maintain the virtual router redundancy protocol (VRRP) configuration. Issue all commands in operational mode. See Operational mode (page 12).

For information on specific components and protocols, see the following sections:

- Displaying VRRP operational information (page 287)
- Locking and unlocking the VRRP component (page 287)

Displaying VRRP operational information

You can display operational information about VRRP using the table Displaying the VRRP operational information (page 287).

Displaying the VRRP operational information

Action	Command	Legend
Display the operational attribute defined by the *VRRP* component:	display Vr/<vrrp_router> Protocolport/<enet> IPport Vrrp/<VRid> virtualRouterState	<vrrp_router> is the name of the Multiservice Switch node VR running VRRP. <VRid> is the instance of VRRP.

Locking and unlocking the VRRP component

The following section describes how to lock and unlock the VRRP component.

Locking and unlocking the VRRP component

Action	Command	Legend
Lock a *Vrrp* component. Locking an active VRRP master router results in the VRRP backup router assuming the master router role.	lock vr/<vrrp_router> Protocolport/<enet> IPport Vrrp/<VRid>	<vrrp_router> is the name of the Multiservice Switch node VR running VRRP. <VRid> is the instance of VRRP.
Unlock a *Vrrp* component. Unlocking a designated master router results in return of the acting master to a backup state.	unlock vr/<vrrp_router> Protocolport/<enet> IPport Vrrp/<VRid>	

Monitoring the IP CoS configuration

This section contains the information you need to monitor and maintain the IP CoS configuration. Issue all commands in operational mode. See Operational mode (page 12).

For information on specific components and protocols, see the following sections:

- Using the ping command with IP CoS (page 288)
- Monitoring the IP CoS configuration (page 288)

Using the ping command with IP CoS

You can send an ICMP packet to a remote IP address with a specific CoS and ToS value. The table Testing connectivity using the ICMP subcomponent (page 264) specifies the commands you can use to test connectivity.

Using the ping command with IP CoS

Action	Command	Legend
Send a ping packet with a Cos value to a specific IP address:	ping -i(<ipaddr>) -cos(<cos>) Vr/<vr> Ip Icmp	<ipaddr> is the remote IP address. <cos> is the CoS value for the packet, and a value of 0, 1, 2, or 3. <vr> is the instance identifier of the remote VR.
Send a ping packet with a TOS byte value to a specific IP address:	ping -i(<ipaddr>) -tos(<tos>) Vr/<vr> Ip Icmp	<tos> is the hexadecimal ToS byte value for the packet.
Send a ping packet with a DSCP value to a specific IP address:	ping -i(<ipaddr>) -dscp(<dscp_value>) Vr/<vr> Ip Icmp	<dscp_value> is the decimal DSCP value for the packet.

Monitoring the IP CoS configuration

The following section describes how to display configuration information and operational statistics for IP CoS.

Monitoring IP CoS

Action	Command	Legend
Display the configurable attributes for specific classification policies on ingress:	display -p Vr/<vr> Ip Pg/<grp> Policy/<plcy> TosMap	<plcy> is the instance identifier of the classification policy.
	display -p Vr/<vr> Ip Pg/<grp> Policy/<plcy> IpAddrLayer4Flow/<flw>	<flw> is the instance identifier of the flow identification policy
	display -p Vr/<vr> Ip Pg/<grp> ingressCosTreatment/*	<grp> is the instance identifier of the CoS policy group
Display the configurable attributes for packet treatment profiles under a specific policy group applied on egress:	display -p Vr/<vr> Ip Pg/<grp> egressCosTreatment/*	<grp> is the instance identifier of the CoS policy group
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring IP DiffServ configuration

Monitor IP DiffServ configuration to help you maintain and monitor your differentiated services configuration.

- Confirming IP DiffServ interface profile usage (page 291)
- Confirming IP DiffServ interface profile configuration (page 293)
- Confirming connection class of connected media (page 295)
- Displaying per hop behaviors (page 296)
- Pinging ICMP with IP DiffServ (page 298)

For more information on monitoring connected media see Monitoring the ATM MPE configuration (page 246) or Monitoring the Arp subcomponent (page 268).

Confirming IP DiffServ interface profile usage

Confirm IP DiffServ interface profile usage to verify how differentiated services have been deployed.

Prerequisites

- You must have completed the procedure IP DiffServ configuration in NN10600-591 *Nortel Networks Multiservice Switch 7400/15000/20000 Layer 3 Traffic Management Configuration*.
- See NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/ 20000 IP Technology Fundamentals* for more information on differentiated services.
- See NN10600-060 *Nortel Networks Multiservice Switch 7400/15000/ 20000 Component Reference* for more information on the components and attributes used in this procedure.

Procedure steps

1 Display the subcomponents that are linked to differentiated services.

```
d -p Vr/<vr_name> Ip Ds <ds_inst> linkToDiffServUsers
```

2 Display the differentiated services the Ipport is using.

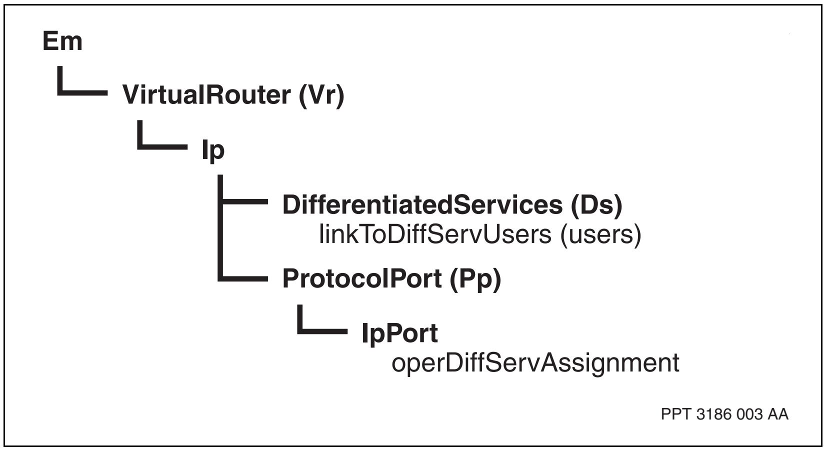
```
d -o Vr/<vr_name> Pp/<pp_name> ipPort
operDiffServAssignment
```

Variable values

Variable	Value
<vr_name>	The name of the virtual router.
<ds_inst>	The instance of the DiffServ component.
<pp_name>	The name of the protocol port.

Procedure job aid

Confirming IP DiffServ interface profile usage component hierarchy



Confirming IP DiffServ interface profile configuration

Confirm IP DiffServ interface profile configuration to verify that the DSCP treatments and discard priority have been configured correctly.

Prerequisites

- You must have completed the procedure IP DiffServ configuration in NN10600-591 *Nortel Networks Multiservice Switch 7400/15000/20000 Layer 3 Traffic Management Configuration*.
- You must have completed the procedure Adding an IP DiffServ interface profile to a virtual router in NN10600-591 *Nortel Networks Multiservice Switch 7400/15000/20000 Layer 3 Traffic Management Configuration*
- See NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Technology Fundamentals* for more information on differentiated services.
- See NN10600-060 *Nortel Networks Multiservice Switch 7400/15000/20000 Component Reference* for more information on the components and attributes used in this procedure.

Procedure steps

1 Display the ingress services.

```
d vr/<vr_name> Ip Ds/<ds_inst> Is
```

2 Display the behavior aggregate components of the ingress services.

```
d vr/<vr_name> Ip Ds/<ds_inst> Is Ba/*
```

3 Display the egress services.

```
d Vr/<vr_name> Ip Ds/<ds_inst> Es
```

4 Display the behavior aggregate components of the egress services.

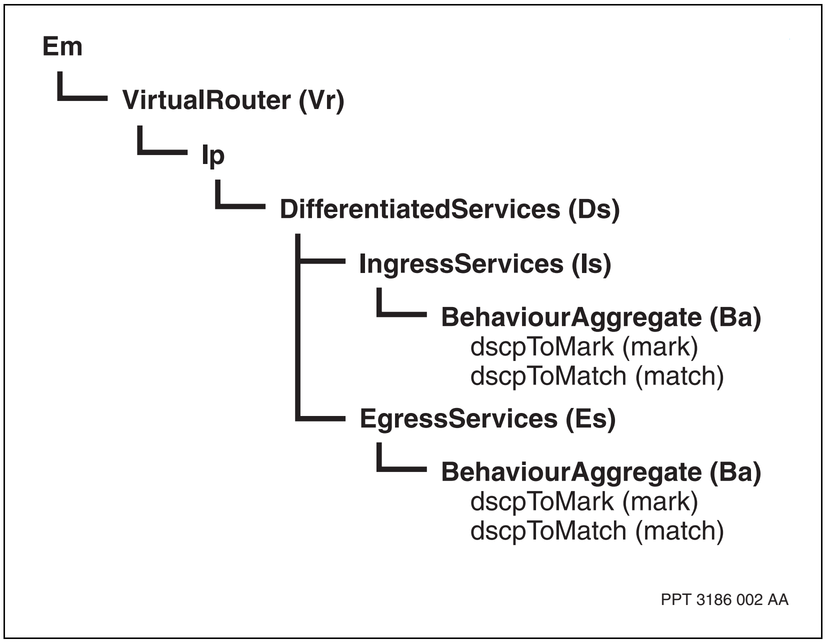
```
d vr/<vr_name> Ip Ds/<ds_inst> Es Ba/*
```

Variable values

Variable	Value
<vr_name>	The name of the virtual router.
<ds_inst>	The instance of the DiffServ component.

Procedure job aid

Confirming DiffServ configuration component hierarchy



Confirming connection class of connected media

Confirm the connection class of connected media to verify that the connection class values specified by *AtmMpe Ac ipCos* matches the connection class value of the connected media.

Prerequisites

- You must have completed the procedure IP DiffServ configuration in NN10600-591 *Nortel Networks Multiservice Switch 7400/15000/20000 Layer 3 Traffic Management Configuration*.
- See NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Technology Fundamentals* for more information on differentiated services.
- See NN10600-060 *Nortel Networks Multiservice Switch 7400/15000/20000 Component Reference* for more information on the components and attributes used in this procedure.

Procedure steps

1 Display information for all the dynamic host entries.

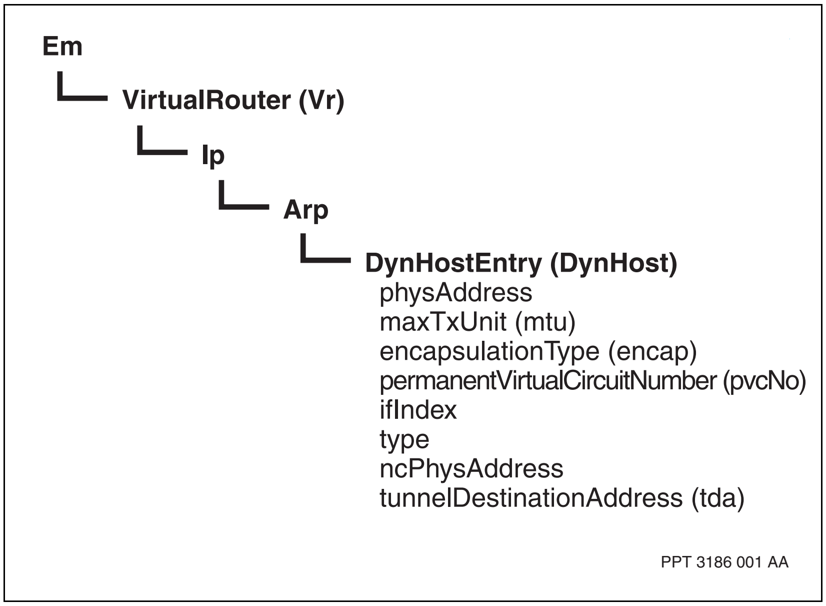
```
d Vr/<vr_name> Ip Arp DynHost/*
```

Variable values

Variable	Value
<vr_name>	The name of the virtual router.

Procedure job aid

Confirming connection class of connected media component hierarchy



Displaying per hop behaviors

Display per hop behaviors (PHB) to verify that each PHB is delivering the proper treatment to the IP packets.

Prerequisites

- You must have completed the procedure IP DiffServ configuration in NN10600-591 *Nortel Networks Multiservice Switch 7400/15000/20000 Layer 3 Traffic Management Configuration*.
- See NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/20000 IP Technology Fundamentals* for more information on differentiated services.
- See NN10600-060 *Nortel Networks Multiservice Switch 7400/15000/20000 Component Reference* for more information on the components and attributes used in this procedure.

Procedure steps

1 Display the per hop behaviors for the differentiated services domain.

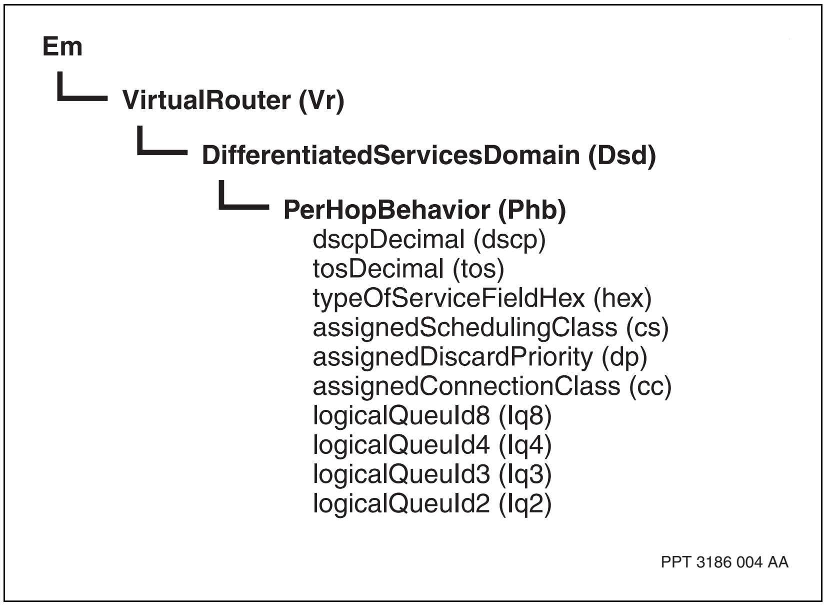
```
d -o Vr/<vr_name> Dsd/<domain_type> Phb/*
```

Variable values

Variable	Value
<vr_name>	The name of the virtual router
<domain_type>	The type of IP DiffServ domain.

Procedure job aid

Displaying per hop behaviors component hierarchy



Em
- VirtualRouter (Vr)
  - DifferentiatedServicesDomain (Dsd)
    - PerHopBehavior (Phb)
      - dscpDecimal (dscp)
      - tosDecimal (tos)
      - typeOfServiceFieldHex (hex)
      - assignedSchedulingClass (cs)
      - assignedDiscardPriority (dp)
      - assignedConnectionClass (cc)
      - logicalQueuId8 (lq8)
      - logicalQueuId4 (lq4)
      - logicalQueuId3 (lq3)
      - logicalQueuId2 (lq2)

PPT 3186 004 AA

Pinging ICMP with IP DiffServ

Ping ICMP with IP DiffServ to determine the time it takes for a packet to travel to an address in the network.

Prerequisites

- You must complete the procedure Adding a DiffServ domain to the virtual router in NN10600-591 *Nortel Networks Multiservice Switch 7400/15000/ 20000 Layer 3 Traffic Management Configuration*
- See NN10600-800 *Nortel Networks Multiservice Switch 7400/15000/ 20000 IP Technology Fundamentals* for more information on differentiated services.
- See NN10600-060 *Nortel Networks Multiservice Switch 7400/15000/ 20000 Component Reference* for more information on the components and attributes used in this procedure.

Procedure steps

1 Ping an IP address with a specified DSCP value.

```
ping -i(<ip_addr>) -dscp (<dscp_value>) Vr/<vr_name> Ip
Icmp
```

2 Ping an IP address with a specified TOS value.

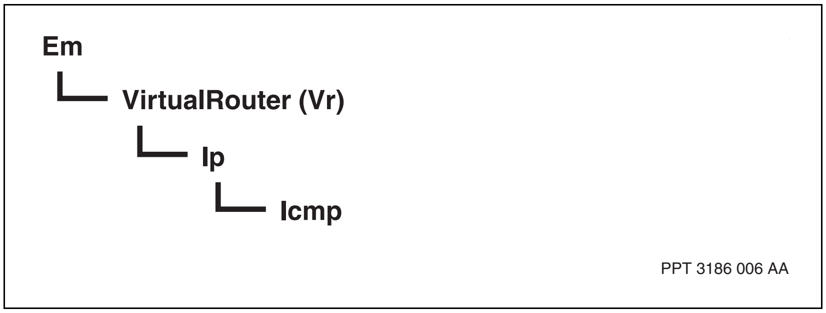
```
ping -i(<ip_addr>) -tos (<tos_value>) Vr/<vr_name> Ip
Icmp
```

Variable values

Variable	Value
<ip_addr>	The IP address you want to ping.
<dscp_value>	The decimal DSCP value you want to mark in the IP packet header.
<tos_value>	The hexadecimal ToS byte value you want to mark in the IP packet header.
<vr_name>	The name of the virtual router

Procedure job aid

Pinging ICMP with IP DiffServ component hierarchy



Monitoring the IP flow filters configuration

You can perform the tests described in this section to determine if you have configured IP flow filters properly. Issue all commands in operational mode. See Operational mode (page 12). For more information, see the following sections:

- Monitoring the filter component (page 300)
- Monitoring the filterFlow subcomponent (page 300)

Monitoring the filter component

The following section describes how to display operational statistics for the *filter* component. For more information, see the table Monitoring the IP cache subcomponent (page 263).

Monitoring the filter component

Action	Command	Legend
List all *filter* component instances	`list Vr/<vr_name> Ip filter/*`	<vr_name> is the name of the virtual router
List all subcomponents of particular *filter* component instance	`list Vr/<vr_name> Ip filter/<filter_name>`	<filter_name> is the name of the filter
Display operational statistics for a particular *filter* component instance	`display Vr/<vr_name> Ip filter/<filter_name>`	

Monitoring the filterFlow subcomponent

The following section describes how to display operational statistics for the *filterFlow* subcomponent. For more information, see the table Monitoring the filterFlow subcomponent (page 300).

Monitoring the filterFlow subcomponent

Action	Command	Legend
List all *filter* component instances	`list Vr/<vr_name> Ip filter/* Ip FilterFlow/*`	<vr_name> is the name of the virtual router

(1 of 2)

Monitoring the filterFlow subcomponent (continued)

Action	Command	Legend
List all subcomponents of particular *filterFlow* subcomponent instance	`list Vr/<vr_name> Ip filter/<filter_name> Ip FilterFlow/<filterflow_number>`	<filter_name> is the name of the filter <filterflow_number> is the number of the flow filter
Display operational statistics for a particular *filterFlow* subcomponent instance	`display Vr/<vr_name> Ip filter/<filter_name> Ip FilterFlow/<filterflow_number>`	
(2 of 2)		

Monitoring the IP tunnel configuration

This section contains the information you need to monitor and maintain the IP tunnel configuration. Issue all commands in operational mode. See Operational mode (page 12).

- Locking and unlocking the Tunnel component (page 302)
- Monitoring the Tunnel component (page 303)
- Monitoring the ProtocolPort component for an IP tunnel (page 303)
- Monitoring the IpPort component for an IP tunnel (page 303)
- Monitoring the LogicalIf component for an IP tunnel (page 304)

Locking and unlocking the Tunnel component

Action	Command	Legend
Lock the *Ip Tunnel* component to take it and all related subcomponents out of service	`lock Vr/<vr_name> Pp/<pp_name>`	<vr_name> is the name of the virtual router <pp_name> is the name of the protocol port
Unlock the *Ip Tunnel* component to return it all related subcomponents to service	`unlock Vr/<vr_name> Pp/<pp_name>`	
Lock all instances of IP tunnel end points for this instance of the *Vr* component	`lock Vr/<vr_name> Ip Tunnel`	
Unlock all instances of IP tunnel end points for this instance of the *Vr* component	`unlock Vr/<vr_name> Ip Tunnel`	

Monitoring the Tunnel component

Action	Command	Legend
Display configurable attributes for the *Tunnel* component	`display -p Vr/<vr_name> Tunnel StaticEndPoint/<endpoint_id>`	
Display operational statistics for the *Tunnel* component	`display Vr/<vr_name> Tunnel StaticEndPoint/<endpoint_id>`	
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the ProtocolPort component for an IP tunnel

Action	Command	Legend
Display configurable attributes for the tunnel *ProtocolPort* component	`display -p Vr/<vr_name> Pp/<pp_name>`	<vr_name> is the name of the virtual router <pp_name> is the name of the protocol port
Display operational attributes for the tunnel *ProtocolPort* component	`display Vr/<vr_name> Pp/<pp_name>`	
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the IpPort component for an IP tunnel

Action	Command	Legend
Display configurable attributes for the tunnel *IpPort* component	`display -p Vr/<vr_name> Pp/<pp_name> IpPort`	<vr_name> is the name of the virtual router <pp_name> is the name of the protocol port
Display operational attributes for the tunnel *IpPort* component	display -p Vr/<vr_name> Pp/<pp_name> IpPort	
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring the LogicalIf component for an IP tunnel

Action	Command	Legend
Display configurable attributes for the *LogicalIf* component	`display -p Vr/<vr_name> Pp/<pp_name> IpPort LogicalIf/<ip_addr>`	<pp_name> is the name of the tunnel protocol port. <ip_addr> is the IP address of the IP logical interface.
Display operational attributes for the *LogicalIf* component	display Vr/<vr_name> Pp/<pp_name> IpPort LogicalIf/<ip_address>	
Attention: -p enables you to list provisionable components while in operational mode		

Monitoring IP CPP on the VR

Perform the tests described in this section to determine if you have configured control plane protection (CPP) properly. Issue all commands in operational mode. See Operational mode (page 12). For more information, see the following sections:

- Determining if traffic is isolated (page 305)
- Restoring traffic for isolated DAs (page 305)

Determining if traffic is isolated

Use the following table to determine if any destination address (DA) on the VRs is currently being isolated.

Determining if traffic is isolated

Action	Command	Legend
Display all *Cpp* component instances	`display Vr/<vr_name> Ip cpp`	<vr_name> is the name of the virtual router
List the DAs that are being isolated	`list Vr/<vr_name> Ip cpp isolatedDa/*`	

If the *monitoringState* is *isolating* then CPP protection is in effect. The count of how many DAs on the VR are currently isolated is contained in *isolatedDACount*. If a DA is isolated on more than one FP, all instances are counted.

If *isolationTime* is zero, control plane traffic will only be restored on the VR through operator intervention.

Restoring traffic for isolated DAs

Use the following table to determine how to manually stop and restore VR control.

Restoring traffic for isolated DAs

Action	Command	Legend
If CPP is currently isolating DAs, restore control traffic to the VR	`clear Vr/<vr_name> Ip cpp`	<vr_name> is the name of the virtual router
Restore traffic to single DA	`clear Vr/<vr_name> Ip cpp isolatedDa/<da>,<lpnum>`	<da> is the destination address <lpnum> is the number of the LP

Troubleshooting

This section contains information about the following:

Troubleshooting ATM MPE

The table Handling problems with ATM MPE (page 308) provides guidelines on how to respond to problems that might occur when using the ATM MPE service.

Handling problems with ATM MPE

Problem	Possible cause	Solution
ATM MPE does not provide service.	The ATM MPE interface is locked.	Unlock the *AtmMpe* component.
	The ATM MPE AtmConnection is locked.	Unlock the *AtmMpe AtmConnection* component.
	The *ifAdminStatus* of the ATM MPE VCC is down.	Configure the *ifAdminStatus* as up.
	The ILS Forwarder interface is locked	Unlock the *ilsFwdr* component.
	The *ifAdminStatus* of the ILS Forwarder interface is down	Configure the *ifAdminStatus* as *up*.
	The ATM FP or ATM port is locked.	Unlock the ATM FP or ATM port.
IP over ATM MPE is not functioning.	The encapsulation type for the *AtmMpe* component is *ipVcEncap*, and there are no static ARP entries provisioned.	Change the encapsulation type to *llcEncap* or add static ARP entries.
	The ARP table has been cleared. This flushes all inverse ARP entries (these are not supplied dynamically).	Lock and unlock the *AtmMpe* component.
	The encapsulation type used by the ATM MPE interfaces at each end of a VCC is not consistent. Both *AtmMpe* instances that terminate a VCC must use the same encapsulation type.	Configure both ends of the VCC to use the same encapsulation type.
	IP is not on the feature list of the ILS Forwarder FP.	Add IP to the ILS Forwarder FP feature list.
(1 of 2)		

Handling problems with ATM MPE (continued)

Problem	Possible cause	Solution
An instance of the *AtmMpe* component is disabled, but not locked, and the *ilsFwdr* component is enabled.	The ATM MPE feature is not in the feature list for the ATM FP.	Add the ATM MPE feature to the ATM software feature list.
The ATM MPE soft PVC fails to establish.	The called endpoint is either busy or does not exist.	Correct the *remoteAddress* and *remoteCi* attributes for the *SrcPvc* component. Make sure that only one calling endpoint in the ATM network is trying to connect to the called endpoint. For more information on the *lastClearCause* attribute, see NN10600-715 *Nortel Networks Multiservice Switch 7400/15000/20000 ATM Fault and Performance Management*.
	There is a link, node, or routing failure somewhere in the PNNI network. You can tell that this is the case because the *lastClearCause* will not be 34.	Use the *lastClearCause* to diagnose the problem.
(2 of 2)		

Troubleshooting frame relay DTE

The table Handling problems with frame relay DTE (page 310) provides guidelines on how to respond to problems that might occur when using the frame relay DTE service.

Handling problems with frame relay DTE

Problem	Possible cause	Solution
Data transfer over all connections is suspended.	The number of local management interface (LMI) procedure errors within the last monitoredEvents window has exceeded the threshold errorThreshold.	Verify that the network equipment has the LMI protocol enabled.
		Verify that the LMI parameters set on the network equipment are compatible with those on the router.
		Turn off the LMI protocol for the frame relay DTE if the network equipment does not support any of the available LMI protocols. Do this by setting the attribute of the LMI component to none.

Troubleshooting PPP

The table Handling problems with PPP (page 311) provides guidelines on how to respond to problems that might occur when using the PPP service.

Handling problems with PPP

Problem	Possible cause	Solution
The link does not come up. Link Control Protocol (LCP) cannot enter the “open” state.	There is incorrect provisioning data.	Make sure that both ends of the link have compatible provisioning data.
	There is a bad cable or cables.	Test cables using port and line tests and remove bad cables. Look at the *lineCondition* attribute in the *Ppp/n Link* component and make sure the line is correct.
	There has been an operator error.	Check the physical layer attributes to ensure that essential components have not been locked.
	There is no clock from the modem or peer DCE connection.	Check the physical layer statistics to see if the physical layer is operational and ready to provide link service to the PPP application.
	Peer link is not initiating the LCP configure request transmission or not responding with an LCP configure acknowledge.	Check *Ppp/n Link operState* attribute to determine the state of the LCP connection. If it is in *reqsent* then the PPP is trying to connect and there must be something wrong with the physical link connection, or the peer PPP is not talking to us.

(1 of 4)

Handling problems with PPP (continued)

Problem	Possible cause	Solution
The link does not come up. Link Control Protocol (LCP) cannot enter the open state.	The quality of the connection may be poor.	If LQM is turned on, then a Bad Line Quality alarm may be set. This can happen only after the *Ppp/n Link operState* attribute is open.
	LCP packets are not getting through to or from peer PPP connection.	Look at the *Ppp/n Framer* statistics of both sides of the link and confirm that there are no CRC errors etc. Look at the ifTable for the physical layer component such as V.35 or DS1 etc. Also look at *Ppp/n Link* attributes to see if PPP is receiving frames that are either too long, too short, of badly formed frames.
A Network Control Protocol (NCP) does not enter the open state.	The network layer protocol is not provisioned on either the local or peer shelf.	Check the provisioning of the network layer protocol on each side of the connection. Look at the *ppp/n ncp state* attributes for the protocol to help determine which side does not want to connect.
	The peer PPP application does not want to open the NCP in question.	Check peer connection and confirm that the network layer protocol does indeed want to connect to the PPP.
(2 of 4)		

Handling problems with PPP (continued)

Problem	Possible cause	Solution
The link connection keeps going down due to bad line quality. (Note this can only occur when Link Quality Monitoring (LQM) is enabled.)	The link quality could be bad.	Either take measures to improve link quality, or set the quality threshold down in the *ppp/n link qualityThreshold* attribute. Check for CRC errors on both sides of the connection.
	The PPP connection may be over-driving the peer's link, or the peer may be over-driving the side of the link resulting in dropped data or LQM reporting packets. This typically happens only at DS3/E3 link speeds.	Disable LQM or throttle the network layer applications data rate.
	The reporting period for link quality reports is set for too short a period for the data rate being used.	Set the *ppp/n lqm configPeriod* attribute to 100 centiseconds or set the period to zero (default) and allow the period to be determined by the peer or set the value to something over 180,000 centiseconds which allows the reporting period to be calculated based on the data rate.
The link connection keeps going down due to bad line quality. (Note that this can only occur when Link Quality Monitoring (LQM) is enabled.)	The window size may be too small. This is the interval of time during which samples may be accumulated. The window may be provisioned such that very bursty traffic results in an erroneous quality determination.	Set the ppp/n link qualityWindow to a larger value. Typically the 30 second default should be fine for all applications.
	The peer PPP connection might not be sending Link Quality Reports (LQRs).	Check the ppp/n inLqrs counter. If it is not incrementing, the peer PPP application is not responding correctly to LQM.
(3 of 4)		

Handling problems with PPP (continued)

Problem	Possible cause	Solution
The link connection keeps going down due to loss of link continuity. (Note that this can only happen when Link Continuity Monitoring (LCM) is enabled.)	The link quality could be bad.	Improve link quality. Check for CRC or any other errors in the physical layer statistics on both sides of the link.
	The PPP connection may be over driving the peer's link, or the peer may be over driving the side of the link resulting in dropped LCM packets. This typically only happens at DS3/E3 link speeds.	Disable LCM or throttle the network layer applications data rate.
Network layer traffic gets dropped when bursts of data occur.	The congestion threshold of the link interface is being reached forcing packets to be discarded. Check PPPs ifTable to determine if the ifOutDiscards counter is incrementing. DS3/E3 link speeds may use the direct hardware transmit method. As a result, ifOutDiscards are not incremented when congestion thresholds are reached because congestion management is done in hardware.	On a Nortel Networks Multiservice Switch 7400 node with SBIC-based FPs, add an LEQ to the PPP application. This allows more elasticity in the data path for bursty traffic conditions. Adding an LEQ turns off direct hardware transmit because all packets are processed by the outbound FP running the LEQ code.
(4 of 4)		

Troubleshooting PPP/ATM interworking

The following list contains information on troubleshooting PPP/ATM interworking:

- If an SPVC does not establish, you need to check the *PppIwf* status and the *AtmIf* port status.
- If the port state is enabled and busy, you need to verify the *lastStepFailureCause*. For example, if this attribute is set to 3 and has no route to destination, PNNI cannot find the route to the destination node. Another possible cause is that the PppIwf feature is not loaded on the ATM FP configured with PNNI.
- If the OSPF is in exchange mode on PPP and at the *AtmMpe* end, it is possible that the static ARP entry is not added on the *AtmMpe* end.

Nortel Networks Multiservice Switch 7400/15000/20000

IP Configuration Management

Publication: NN10600-801
Document status: Standard
Document issue: 6.1S2
Document date: November 2004
Product release: Release 6.1
Job function: Operations
Type: NTP
Language type: U.S. English