CHAPTER 6

Managing PNNI Nodes and PNNI Routing

This chapter provides procedures that you can use to manage Private Network-to-Network Interface (PNNI) nodes and routes. This chapter includes the following sections:

- Managing PNNI Nodes
- Managing PNNI Route and Link Selection
- Displaying Node Configuration Information

Note The concepts behind the procedures in this chapter are introduced in the *Cisco MGX and SES PNNI Network Planning Guide*.

Managing PNNI Nodes

The following sections describe how to configure upper level peer groups and how to manage the PNNI node.

Creating Upper Level Peer Groups

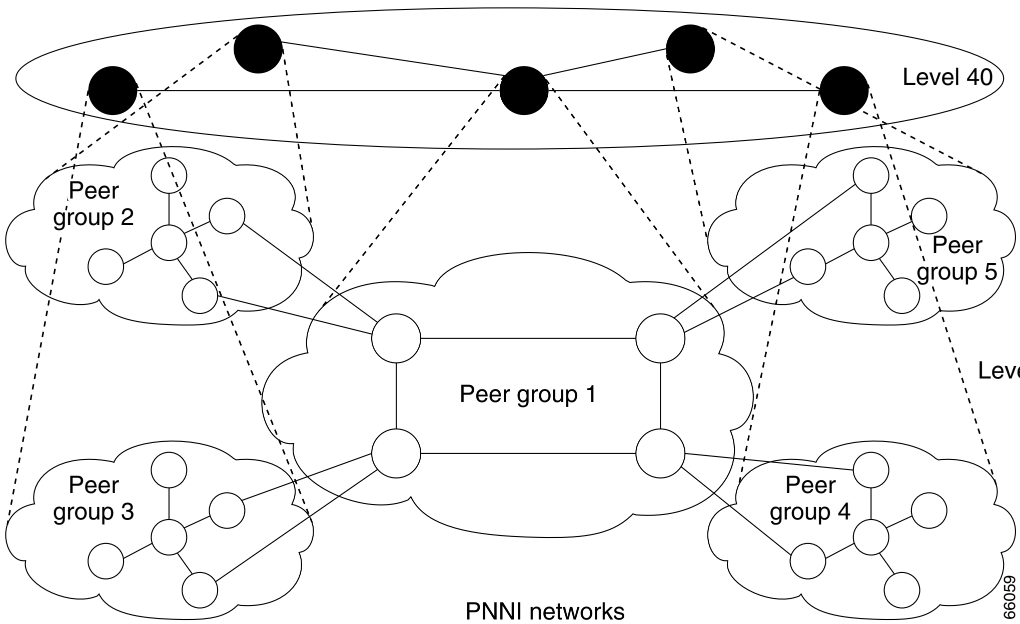
Upper level peer groups enable routing from one PNNI peer group to another. If you are managing a single peer group WAN, you do not need to create upper level peer groups.

Note "Configuring PNNI Node Parameters" in Chapter 2, "Configuring General Switch Features," describes how to configure the lowest level peer group parameters, which many upper level peer group parameters are based on. You should configure the basic PNNI node parameters before creating upper level peer groups.

After you configure the lowest level PNNI nodes, all nodes within the same peer group can communicate with each other. All you need to do to enable communications between two nodes in a peer group is to add a PNNI trunk between them as described in "MPLS and PNNI Trunk Configuration Quickstart" in Chapter 5, "Provisioning AXSM Communication Links." To enable routing between different peer groups at the same level, you must create one or more upper level peer groups.

The actual procedure for creating an upper level peer group for your WAN depends on the structure of your WAN. This section shows how to create an upper level peer group for the WAN shown in Figure 6-1.

Figure 6-1 Example Hierarchical PNNI Network Topology Showing a Two-Level Hierarchy



In Figure 6-1, the five level-56 peer groups are isolated from each other until the upper level peer group is created. The members of the upper level peer group are the peer group leaders from the lower level peer groups. To create an upper level peer group, you need to configure the peer group leaders and add the upper level PNNI process to each peer group leader (PGL) node. It is also a good practice to configure secondary peer group leaders that can take over if a PGL fails.

To configure peer group leaders, use the following procedure.

Step 1 Establish a configuration session using a user name with SUPER_GP privileges or higher.

Step 2 Add the upper level PNNI logical node that will participate in the higher level PNNI group using the following command.

```
8950_SF.7.PXM.a > addpnni-node level
```

Replace level with the PNNI level for the higher level peer group. The PNNI level value must be smaller than the level value for the lower level peer groups. The following example creates a logical PNNI node at PNNI level 48.

```
8950_SF.7.PXM.a > addpnni-node 48
```

Note You need to complete this step for all nodes that will serve as PGLs or backup PGLs.

Step 3 Display the current PGL priority of the node that will become PGL or a back up PGL by entering the **dsppnni-election** command as shown in the following example:

```
8950_SF.7.PXM.a > dsppnni-election

node index: 1
   PGL state......     OperNotPgl     Init time(sec).......        15
   Priority.......              0     Override delay(sec)..        30
                                      Re-election time(sec)        15
   Pref PGL..............0:0:00.000000000000000000000000.000000000000.00
   PGL...................0:0:00.000000000000000000000000.000000000000.00
   Active parent node id..0:0:00.000000000000000000000000.000000000000.00



node index: 2
   PGL state......       Starting     Init time(sec).......        15
   Priority.......              0     Override delay(sec)..        30
                                      Re-election time(sec)        15
   Pref PGL..............0:0:00.000000000000000000000000.000000000000.00
   PGL...................0:0:00.000000000000000000000000.000000000000.00
   Active parent node id..0:0:00.000000000000000000000000.000000000000.00
```

In the example above, the PGL state indicates the PGL status of each of two logical nodes, and the priority value is what is used to determine if the node will become PGL. Since a PGL represents the peer group at a higher level, logical node 1 (node index 1) is the only node that can become a PGL. In this example, both logical nodes are set to the default value 0, and this value prevents a node from becoming a peer group leader.

Step 4 Set the PNNI priority for the node with the cnfpnni-election command as follows:

```
8950_SF.7.PXM.a > cnfpnni-election node-index -priority value
```

Replace *node-index* with the index that identifies the logical node you are modifying, and replace *value* with the new priority value. A zero value prevents the node from becoming a PGL. If only one node in a peer group has a non-zero priority, that node will become PGL. If multiple nodes have non-zero priority values, the node with the highest priority value becomes PGL. The following example shows what happens after you set the priority level and view the PGL status.

```
8950_SF.7.PXM.a > cnfpnni-election 1 -priority 200

8950_SF.7.PXM.a > dsppnni-election

node index: 1
   PGL state...... AwaitUnanimity     Init time(sec).......        15
   Priority.......            200     Override delay(sec)..        30
                                      Re-election time(sec)        15
   Pref PGL..............56:160:47.00918100000100036b5e31b3.00036b5e31b3.01
   PGL...................0:0:00.000000000000000000000000.000000000000.00
   Active parent node id..0:0:00.000000000000000000000000.000000000000.00



node index: 2
   PGL state......       Starting     Init time(sec).......        15
   Priority.......              0     Override delay(sec)..        30
                                      Re-election time(sec)        15
   Pref PGL..............0:0:00.000000000000000000000000.000000000000.00
   PGL...................0:0:00.000000000000000000000000.000000000000.00
   Active parent node id..0:0:00.000000000000000000000000.000000000000.00
```

The first time the dsppnni-election command was entered, the PGL state was OperNotPgl, which means that the node is operating, but is not operating as a PGL. After the priority is changed, the PGL state changes to AwaitUnanimity, which means the node is communicating with the other nodes in its peer group to see if it has the highest priority and should be PGL. If you enter the dsppnni-election command again after about 15 seconds, the PGL state changes as shown in the following example:

```
8950_SF.7.PXM.a > dsppnni-election

node index: 1
   PGL state......        OperPgl     Init time(sec).......        15
   Priority.......            250     Override delay(sec)..        30
                                      Re-election time(sec)        15
   Pref PGL..............56:160:47.00918100000100036b5e31b3.00036b5e31b3.01
   PGL...................56:160:47.00918100000100036b5e31b3.00036b5e31b3.01
   Active parent node id..48:56:47.009181000001000000000000.00036b5e31b3.00



node index: 2
   PGL state......     OperNotPgl     Init time(sec).......        15
   Priority.......              0     Override delay(sec)..        30
                                      Re-election time(sec)        15
   Pref PGL..............0:0:00.000000000000000000000000.000000000000.00
   PGL...................0:0:00.000000000000000000000000.000000000000.00
   Active parent node id..0:0:00.000000000000000000000000.000000000000.00
```

In the example above, the PGL state changes to show that logical node 1 is now the PGL. Notice that the priority value is 250. An earlier example in this procedure set the priority to 200. When a node is elected PGL, the node adds 50 to its priority value to prevent instability that might be caused by other peer group nodes with a marginally higher priority value.

Step 5 Repeat this procedure for backup peer group leaders and be sure to set their priority value to a lower value so that they operate as backup PGLs.

Enabling and Disabling Routes Through a Node

The restricted transit option allows you to allow or block call routes that pass through the node and terminate on other nodes. The default setting for this option enables calls to pass through.

To enable or disable PNNI routing through a node, use the **cnfpnni-node** command as follows:

```
8850_LA.7.PXM.a > cnfpnni-node <node-index> -transitRestricted on|off
```

Replace *node-index* with the index that identifies the logical node you are modifying, and enter either *on* or *off* for the -transitRestricted parameter. When this parameter is set to on, the node only accepts calls that terminate on this node. When the -transitRestricted parameter is set to off, the node accepts calls that pass through the node and terminate on other nodes.

To view the status of the -transitRestricted option, enter the **dsppnni-node** command as shown in the following example:

```
8850_LA.7.PXM.a > dsppnni-node

node index: 1                      node name: 8850_LA
   Level...............        56     Lowest..............      true
   Restricted transit..        on     Complex node........       off
   Branching restricted        on
   Admin status........        up     Operational status..        up
   Non-transit for PGL election..       off
   Node id...............56:160:47.00918100000100001a531c2a.00001a531c2a.01
   ATM address...........47.00918100000100001a531c2a.00001a531c2a.01
   Peer group id.........56:47.00.9181.0000.0100.0000.0000.00
```

Enabling and Disabling Point-to-Multipoint Routes

The branching restricted option allows you to allow or block point-to-multipoint calls. The default setting for this option enables point-to-multipoint calls.

To enable or disable point-to-multipoint routes through a node, use the **cnfpnni-node** command as follows:

```
8850_LA.7.PXM.a > cnfpnni-node <node-index > -branchingRestricted on|off
```

Replace *node-index* with the index that identifies the logical node you are modifying, and enter either *on* or *off* for the -branchingRestricted parameter. When this parameter is set to on, the node does not accept point-to-multipoint calls. When the -branchingRestricted parameter is set to off, the node accepts point-to-multipoint calls.

To view the status of the --branchingRestricted option, enter the **dsppnni-node** command as shown in the following example:

```
8850_LA.7.PXM.a > dsppnni-node

node index: 1                      node name: 8850_LA
   Level...............        56     Lowest..............      true
   Restricted transit..       off     Complex node........       off
   Branching restricted        on
   Admin status........        up     Operational status..        up
   Non-transit for PGL election..       off
   Node id...............56:160:47.00918100000100001a531c2a.00001a531c2a.01
   ATM address...........47.00918100000100001a531c2a.00001a531c2a.01
   Peer group id.........56:47.00.9181.0000.0100.0000.0000.00
```

Adding an ATM Summary Address Prefix

Use the **addpnni-summary-addr** command to add an ATM summary address prefix for a PNNI logical node on the switch:

```
Geneva.7.PXM.a > addpnni-summary-addr <node-index> <address-prefix> <prefix-length> [-type]
[-suppress] [-state]
```

Table 6-1 lists the parameter descriptions for the **addpnni-summary-addr** command.

Table 6-1 Parameters for addpnni-summary-addr Command

Parameter	Description
node-index	The node index assigned to a PNNI logical node on a network. Range = 1 - 65535
addressprefix	The ATM address prefix assigned to the network.
prefixlength	The length of the summary address-prefix in number of bits, equal or less than 152 bits. Currently, the zero-length summary address is not supported.
-type	The type of the summary address.
-suppress	true = summary address is not advertised.
-state	The summary address is advertised | notadvertised | inactive.

Configuring SVCC RCC Variables

Configure SVCC-based RCC variables with the **cnfpnni-svcc-rcc-timer** command:

```
Geneva.7.PXM.a > cnfpnni-svcc-rcc-timer <node-index> [-initTime] [-retryTime]
[-callingIntegrityTime] [-calledIntegrityTime]
```

This defines a node's initial PNNI SVCC-based variables, as shown in Table 6-2.

Table 6-2 Parameters for cnfpnni-svcc-rcc-timer Command

Parameter	Description
node-index	Node index.
-inittime	Time (in seconds) that the node delays establishment of an SVCC to a neighbor with a numerically lower ATM address, after determining that such an SVCC should be established.
-retrytime	Time (in seconds) that the node delays before attempting to re-establish an SVCC-based RCC after the RCC is unexpectedly torn down.
-callingintegritytime	Time (in seconds) that the node waits for a sent SVCC to become fully established before giving up and tearing it down.
-calledintegritytime	Time (in seconds) that the node waits for a received SVCC to become fully established before giving up and tearing it down.

Configuring Routing Policies for Background Routing Tables

Configure the routing policies used for background routing tables generation with the command **cnfpnni-routing-policy**:

```
Geneva.7.PXM.a > cnfpnni-routing-policy [-sptEpsilon] [-sptHolddown] [-bnPathHolddown]
[-loadBalance] [-onDemand] [-awBgTable] [-ctdBgTable] [-cdvBgTable]
```

Table 6-3 lists the parameter descriptions for the **cnfpnni-routing-policy** command.

Table 6-3 Parameters for cnfpnni-routing-policy Command

Parameter	Description
-epsilon	Indicates the node's policy in determining equal-cost path during routes calculation.
-holddown	Defines the node's minimum time interval between two consecutive calculations for generating routing tables.
-bnpathholddown	Defines the minimum time interval between two consecutive calculations for generating border node path in a peer group for a complex node representation at the next higher level. (Complex nodes are not supported by MGX 8850, Release 2.1 software image)
-loadBalance	Defines the node's load balancing rule if alternative equal-lose routes exist for the call request.
-onDemand	Defines the node's on-demand routing rule, either to: firstfit = select a route that is the first it can find bestfit = select the best route Default = **firstfit**
-awBgTable	Enable or disable administrative weight for the background routing table. Default = **off**
-ctdBgTable	Enable or disable CTD for the background routing table. Default = **off**
-cdvBgTable	Enable or disable CDV for the background routing table. Default = **off**

Configuring PNNI Timers

Configure the PNNI timers with the **cnfpnni-timer** command:

`Geneva.7.PXM.a >` **`cnfpnni-timer`** <*node-index*>

You can define the initial PNNI timer values and significant change thresholds of a PNNI logical node. Table 6-4 lists the parameter descriptions for the **cnfpnni-timer** command.

Table 6-4 Parameters for cnfpnni-timer Command

Parameter	Description
nodeindex	Logical node's node index.
-ptseholddown	The number is used as a multiplier of the Hello interval of the peer neighbor: the product is the maximum time that the neighbor is considered to be alive without the reception of its Hello packets. Range: (0.1 through 10) second Default = **1**
-helloholddown	Value for the Hello hold down timer that limits the rate at which it sends Hellos.
-hellointerval	Initial value for the Hello timer.
-helloinactivityfactor	Inactivity time factor on a horizontal link between two logical nodes.
-ptserefreshinterval	Time allowed for the PTSE to re-originate.
-ptselifetimefactor	Value for the lifetime multiplier, expressed as a percentage. The product of it and the **ptserefreshinterval** is sets the remaining lifetime of a self-originated PTSE.
-retransmitinterval	Period between retransmissions of unacknowledged DS, PTSE request, and PTSP.
-ptsedelayedackinterval	Minimum time allowed between transmissions of delayed PTSE acknowledgment packets.
-avcrpm	Proportional multiplier used in the algorithms that determines significant change for AvCR parameters.
-avcrmt	Minimum threshold used in the algorithms that determine significant change for AvCR parameters.
-cdvpm	Proportional multiplier used in the algorithms that determine significant change for CDV parameters.
-ctdpm	Proportional multiplier used in the algorithms that determine significant change for CTD parameters.

Managing PNNI Route and Link Selection

The following sections describe how to control route and link selection for the links on each PNNI node.

Configuring the Route Selection Method (First Fit or Best Fit)

When the PNNI controller searches for routes, it can choose the first route that meets the call requirements, or it can choose the route that provides the best performance. The first fit method chooses the first available route and reduces call processing time. The best fit method chooses the optimum route, but it takes longer to select the route. The default setting is first fit.

Note The route selection process is described in the *Cisco MGX and SES PNNI Network Planning Guide*.

To configure the route selection method, use the **cnfpnni-routing-policy** command as follows:

```
8850_LA.7.PXM.a > cnfpnni-routing-policy -onDemand firstfit|bestfit
```

Enter *firstfit* to select the first route discovered, or enter *bestfit* to select the optimum route.

To display the route selection method, enter the **dsppnni-routing-policy** command as follows:

```
8850_LA.7.PXM.a > dsppnni-routing-policy

   SPT epsilon.........          0     Load balance........    random
   SPT holddown time...          1     On demand routing... first fit
   SPT path holddown time        2     AW Background Table         on
   CTD Background Table         on     CDV Background Table        on
```

The parameter labeled *On demand routing* shows which route selection method is configured.

Configuring the Best-Fit Route Selection Method

When the PNNI controller is configured to choose the best route and it discovers multiple eligible routes, the load balancing option determines which route to select. The option settings are random and maxbw, which selects the route with the greatest available bandwidth. Random selection is used to balance the load.

Note The route selection process is described in the *Cisco MGX and SES PNNI Network Planning Guide*.

To configure the best-fit route selection method, use the **cnfpnni-routing-policy** command as follows:

```
8850_LA.7.PXM.a > cnfpnni-routing-policy -loadBalance random|maxbw
```

Enter *random* to balance route selection, or enter *maxbw* to select the route with the greatest available bandwidth.

To display the route selection method, enter the **dsppnni-routing-policy** command as follows:

```
8850_LA.7.PXM.a > dsppnni-routing-policy

   SPT epsilon.........          0     Load balance........    random
   SPT holddown time...          1     On demand routing... first fit
   SPT path holddown time        2     AW Background Table         on
   CTD Background Table         on     CDV Background Table        on
```

The parameter labeled *Load balance* shows which best-fit route selection method is configured.

Configuring Link Selection for Parallel Links

When parallel links exist between two nodes on a route, the node closest to the originating node selects a link based on one of the following:

- The lowest Administrative Weight (minaw)
- The maximum available cell rate (maxavcr)
- The maximum cell rate configured for the link (maxcr)
- Random link selection (loadbalance)

Note The route selection process is described in the *Cisco MGX and SES PNNI Network Planning Guide*.

To configure the link selection method, use the **cnfpnni-link-selection** command as follows:

```
8850_LA.7.PXM.a > cnfpnni-link-selection pnportid minaw|maxavcr|maxcr|loadbalance
```

Replace *pnportid* with the port ID in the format slot[:subslot].port[:subport]. (This is the same format that appears when you display ports with the *dsppnport* command.) Enter one link selection method after the port ID.

To display the link selection method, enter the **dsppnni-link-selection** command as follows:

```
8850_LA.7.PXM.a > dsppnni-link-selection 1:2.1:1

physical port id:          1:2.1:1     link selection: minaw
 logical port id:         16848897
```

Configuring the Maximum Bandwidth for a Link

The maximum bandwidth for a link is defined when a PNNI partition is configured for a port. For more information, see Chapter 5, Provisioning.

Configuring the Administrative Weight

The link Administrative Weight (AW) is used to calculate the total cost of a route and can be used by the PNNI controller when it has to choose between multiple parallel links. You can assign different AW values for each ATM class of service.

Note The role of AW in route and link selection is described in more detail in the *Cisco MGX and SES PNNI Network Planning Guide*.

To configure the AW for a link, enter the **cnfpnni-intf** command as follows:

```
8850_LA.7.PXM.a > cnfpnni-intf <pnportid> [-awcbr] [-awrtvbr] [-awnrtvbr] [-awabr]
[-awubr] [-awal]
```

Replace *pnportid* with the port ID in the format slot[:subslot].port[:subport]. (This is the same format that appears when you display ports with the *dsppnport* command.) For each class of service for which you want to change the AW value, enter the appropriate option followed by the new value. For example, the following command sets the AW for CBR calls over the link:

```
8850_LA.7.PXM.a > cnfpnni-intf 1:2.1:1 -awcbr 2000
```

To display the AWs assigned to a PNNI port, enter the **dsppnni-intf** command as follows:

```
8850_LA.7.PXM.a > dsppnni-intf 1:2.1:1

Physical port id: 1:2.1:1          Logical port id:   16848897
   Aggr token..........         0     AW-NRTVBR...........      5040
   AW-CBR..............      2000     AW-ABR..............      5040
   AW-RTVBR............      5040     AW-UBR..............      5040
```

Configuring the Bandwidth Overbooking Factor

The bandwidth overbooking factor represents the percentage of the actual available bandwidth that is advertised for links as the Available Cell Rate (AvCR). The default overbooking factor is 100, and this specifies that 100% of the actual available bandwidth should be advertised as the AvCR. When the overbooking factor is set below 100, a link is underbooked because only a portion of the available bandwidth is used for routing. When the overbooking factor is set above 100, the link can be overbooked.

Note For more information on the bandwidth overbooking factor, refer to the *Cisco MGX and SES PNNI Network Planning Guide*.

To configure the bandwidth overbooking factor for a PNNI port, enter the **cnfpnportcac** as follows:

```
8850_LA.7.PXM.a > cnfpnportcac <pnportid> <service_catogory>
[-bookfactor <utilization-factor>]
```

Replace *pnportid* with the port ID in the format slot[:subslot].port[:subport]. (This is the same format that appears when you display ports with the *dsppnport* command.) Replace *service_catogory* with the ATM class of service for which you are defining the overbooking factor, and replace *utilization-factor* with the new overbooking factor. For example:

```
8850_LA.7.PXM.a > cnfpnportcac 1:2.1:1 cbr -bookfactor 120
WARNING: New CAC parameters apply to existing connections also
```

To display the bandwidth overbooking factor for all classes of service, enter the **dsppnportcac** command as shown in the following example:

```
8850_LA.7.PXM.a > dsppnportcac 1:2.1:1

                  cbr:       rt-vbr:       nrt-vbr:          ubr:          abr:
sig:
bookFactor:       120%          100%          100%          100%          100%
100%
maxBw:       100.0000%     100.0000%     100.0000%     100.0000%     100.0000%
100.0000%
minBw:         0.0000%       0.0000%       0.0000%       0.0000%       0.0000%
0.3473%
maxVc:            100%          100%          100%          100%          100%
100%
minVc:              0%            0%            0%            0%            0%
1%
maxVcBw:            0             0             0             0             0
0
```

Displaying Node Configuration Information

The following sections describe commands that display PNNI configuration information.

Displaying the PNNI Node Table

Once a PNNI node is configured, enter the **dsppnni-node** command to show the WAN nodal table. The node list is displayed in ascending order of each node index, all with one setting the node to the lowest PNNI hierarchy.

The significant information that will display is as follows:

- Node index
- Node name
- Node level (56 for all nodes until multiple peer groups are supported)
- Restricted transit—a flag that can prevent PNNI routing from transmitting this node
- Branching restricted—a flag that can prevent cpu-intensive branching at this node
- Admin status—up/down
- Operational status—up/down
- Nontransit for PGL election—a flag that indicates that node's level of eligibility as a PGL
- Node id—The 22-byte PNNI logical identification
- ATM address
- pg id—Peer group ID

The following example shows the report for this command:

```
Geneva.7.PXM.a > dsppnni-node
node index: 1                          node name: Geneva
   Level...............         56     Lowest..............       true
   Restricted transit..        off     Complex node........        off
   Branching restricted         on
   Admin status........         up     Operational status..         up
   Non-transit for PGL election..        off
   Node id...............56:160:47.0091810000000030ff0fef38.0030ff0fef38.01
   ATM address...........47.0091810000000030ff0fef38.0030ff0fef38.01
   Peer group id.........56:47.00.9181.0000.0000.0000.0000.00
Geneva.7.PXM.a >
```

Displaying the PNNI Summary Address

Use the **dsppnni-summary-addr** command to display PNNI summary addresses as follows:

```
Geneva.7.PXM.a > dsppnni-summary-addr [node-index]
```

If you specify the node-index, this command displays the summary address prefixes of the node-index PNNI node.

If you do not specify the node-index, this command displays summary address prefixes for all local nodes on network.

Table 6-5 shows the objects displayed for the **dsppnni-summary-addr** command.

Table 6-5 Objects Displayed for dsppnni-summary-addr Command

Parameter	Description
node-index	The node index number assigned to a PNNI logical node on a network. Replace [*node-index*] with a number in the range from 1 to 65535.
addressprefix	The ATM address prefix assigned to the network.
prefixlength	The length of the summary address-prefix in number of bits, equal or less than 152 bits. Currently, the zero-length summary address is not supported.
-type	The type of the summary address.

Table 6-5 Objects Displayed for dsppnni-summary-addr Command (continued)

Parameter	Description
-suppress	true = summary address is not advertised.
-state	The summary address state can be advertising, notadvertised, or inactive.

This example shows the **dsppnni-summary-addr** command line that displays the PNNI address prefixes.

```
8850_LA.7.PXM.a > dsppnni-summary-addr

node index: 1
   Type..............    internal     Suppress..............   false
   State............. advertising
   Summary address........47.0091.8100.0000.0000.1a53.1c2a/104
```

Displaying System Addresses

The **dsppnsysaddr** command is more specific; it displays the following list of addresses from the System Address Table:

- ilmi
- uni
- static
- host

The following example shows the report for this command:

```
Geneva.7.PXM.a > dsppnsysaddr
47.0091.8100.0000.0030.ff0f.ef38.0000.010b.180b.00/160
Type:     host     Port id:   17251106

47.0091.8100.0000.0030.ff0f.ef38.0000.010b.1816.00/160
Type:     host     Port id:   17251106

47.0091.8100.0000.0030.ff0f.ef38.0000.010b.1820.00/160
Type:     host     Port id:   17251106

47.0091.8100.0000.0030.ff0f.ef38.0000.010b.1821.00/160
Type:     host     Port id:   17251106

47.0091.8100.0000.0030.ff0f.ef38.0000.010d.1820.00/160
Type:     host     Port id:   17251106

47.0091.8100.0000.0030.ff0f.ef38.0000.010d.1821.00/160
Type:     host     Port id:   17251106

47.0091.8100.0000.0030.ff0f.ef38.0000.010d.1822.00/160
Type:     host     Port id:   17251106

47.0091.8100.0000.0030.ff0f.ef38.0000.010d.180b.00/160
Type:     host     Port id:   17251106

47.0091.8100.0000.0030.ff0f.ef38.0030.ff0f.ef38.01/160
Type:     host     Port id:   17251106
```

```
47.0091.8100.0000.0030.ff0f.ef38.0030.ff0f.ef38.99/160
Type:     host     Port id:   17251106

47.0091.8100.0000.0030.ff0f.ef38.1111.1101.0001.01/160
Type:     host     Port id:   17251106

47.0091.8100.0000.0050.0fff.e0b8/104
Type:   static     Port id:   17635339

39.6666.6666.6666.6666.6666.6666.6666.6666.6666/152
Type:      uni     Port id:   17504267

Geneva.7.PXM.a >
```

Displaying PNNI Interface Parameters

Enter the **dsppnni-intf** command to display the service category-based administrative weight and aggregation token parameters:

```
Geneva.7.PXM.a > dsppnni-intf [node-index] [port-id]
```

The following example shows the report for this command:

```
Geneva.7.PXM.a > dsppnni-intf 11:2.2:22
Physical port id: 11: 2.2:22       Logical port id:   17504278
   Aggr token.........         0     AW-NRTVBR..........      5040
   AW-CBR.............      5040     AW-ABR.............      5040
   AW-RTVBR...........      5040     AW-UBR.............      5040
Geneva.7.PXM.a >
```

Table 6-6 describes the objects displayed for the **dsppnni-intf** command.

Table 6-6 Objects Displayed for the dsppnni-intf Command

Parameter	Description
portid	The Port Identifier.
token	The 32-bit number used for link aggregation purpose.
aw	The 24-bit number used as administrative weight on this interface. The maximum possible value is a 24-bit unsigned integer.

Displaying the PNNI Link Table

Enter the **dsppnni-link** command to show the PNNI link table:

```
Geneva.7.PXM.a > dsppnni-link [node-index] [port-id]
```

If you specify:

- Both <*node-index*> and <*port-id*>, the command displays information about that specific <*port-id*> port.
- Only <*node-index*>, the command displays information about all PNNI link attached to the <*node-index*> node.
- Nothing, command displays all links attached to all PNNI nodes on this switching system.

The final option allows you to see all communication lines in the PNNI network.

The following example shows the report for this command:

```
Geneva.7.PXM.a > dsppnni-link

node index   : 1
Local port id:   17504278          Remote port id:   17176597
Local Phy Port Id: 11:2.2:22
   Type. lowestLevelHorizontalLink     Hello state....... twoWayInside
   Derive agg...........         0     Intf index...........  17504278
   SVC RCC index........         0     Hello pkt RX.........     17937
                                       Hello pkt TX.........     16284
   Remote node name.......Paris
   Remote node id.........56:160:47.00918100000000107b65f27c.00107b65f27c.01
   Upnode id..............0:0:00.000000000000000000000000.000000000000.00
   Upnode ATM addr........00.000000000000000000000000.000000000000.00
   Common peer group id...00:00.00.0000.0000.0000.0000.0000.00

node index   : 1
Local port id:   17504288          Remote port id:   17045536
Local Phy Port Id: 11:2.1:32
   Type. lowestLevelHorizontalLink     Hello state....... twoWayInside
   Derive agg...........         0     Intf index...........  17504288
   SVC RCC index........         0     Hello pkt RX.........     18145

Type <CR> to continue, Q<CR> to stop:
                                       Hello pkt TX.........     19582
   Remote node name.......SanJose
   Remote node id.........56:160:47.00918100000000309409f1f1.00309409f1f1.01
   Upnode id..............0:0:00.000000000000000000000000.000000000000.00
   Upnode ATM addr........00.000000000000000000000000.000000000000.00
   Common peer group id...00:00.00.0000.0000.0000.0000.0000.00

node index   : 1
Local port id:   17504289          Remote port id:   17045537
Local Phy Port Id: 11:2.1:33
   Type. lowestLevelHorizontalLink     Hello state....... twoWayInside
   Derive agg...........         0     Intf index...........  17504289
   SVC RCC index........         0     Hello pkt RX.........     17501
                                       Hello pkt TX.........     18877
   Remote node name.......SanJose
   Remote node id.........56:160:47.00918100000000309409f1f1.00309409f1f1.01
   Upnode id..............0:0:00.000000000000000000000000.000000000000.00
   Upnode ATM addr........00.000000000000000000000000.000000000000.00
   Common peer group id...00:00.00.0000.0000.0000.0000.0000.00
```

Displaying the PNNI Routing Policy

Enter the **dsppnni-routing-policy** command to display the routing policies used for background routing tables generation:

```
Geneva.7.PXM.a > dsppnni-routing-policy
```

The following example shows the report for this command:

```
Geneva.7.PXM.a > dsppnni-routing-policy
   SPT epsilon.........         0     Load balance........    random
   SPT holddown time...         1     On demand routing...  best fit
   SPT path holddown time       2     AW Background Table         on
   CTD Background Table        on     CDV Background Table        on
Geneva.7.PXM.a >
```

Table 6-7 describes the objects displayed for the **dsppnni-routing-policy** command.

Table 6-7 Objects Displayed for the dsppnni-routing-policy Command

Parameter	Description
sptEpsilon	The tolerance used during route calculation to determine which paths qualify as equal-cost. The range is from 0 - 20.
sptHolddown	The interval between two consecutive calculations for generating routing tables. The range is from 1 (0.1 sec) to 600 (60 sec).
bnPathHolddown	The minimum time that can elapse between consecutive calculations that generate routing tables for border nodes. The range is from 2 (0.2 sec) to 600 (60 sec).
-loadBalance	Defines the load balancing rule if alternative equal-cost routes exist for a given call request.
onDemand	The on-demand routing rule. On-demand routing is used. *Firstfit* routing selects the first route found that goes to the selected destination. Firstfit route search time is minimized, but the selected route is not optimum. *Bestfit* routing selects a route based on the least-cost. The average route-search-time is greater, and more CPU-intensive, but the optimum route is selected.
awBgTable	Displays whether the administrative weight for the background routing table is enabled or disabled.
ctdBgTable	Displays whether cell transfer delay (CTD) for the background routing table is enabled or disabled. CTD is the time interval between a cell exiting source node and entering the destination node.
cdvBgTable	Displays whether cell delay variation (CDV) for the background routing table is enabled or disabled. CDV is a component of cell transfer delay, and is a quality of service (QoS) delay parameter associated with CBR and VBR service.

Displaying the SVCC RCC Timer

Enter the **dsppnni-svcc-rcc-timer** command to display SVCC-based RCC variables:

```
Geneva.7.PXM.a > dsppnni-svcc-rcc-timer
```

The following example shows the report for this command:

```
Geneva.7.PXM.a > dsppnni-svcc-rcc-timer
node index: 1
   Init time...........          4     Retry time..........        30
   Calling party integrity time...        35
   Called party integrity time....        50
```

Table 6-8 shows the objects displayed for the **dsppnni-svcc-rcc-timer** command.

Table 6-8 Objects Displayed for the dsppnni-svcc-rcc-timer Command

Parameter	Description
node-index	The node index assigned to a PNNI logical node on a network. The range is from 1 to 65535.
initTime	The amount of time (in seconds) this node will delay advertising its choice of preferred an SVCC to a neighbor with a numerically lower ATM address, after determining that such an SVCC should be established. The range is from 1 to 10
retryTime	The amount of time (in seconds) this node will delay after an apparently still necessary and viable SVCC-based RCC is unexpectedly torn down, before attempting to re-establish it. The range is from 10 to 60
callingIntegrityTime	The amount of time (in seconds) this node will wait for an SVCC, which it has initiated establishment of as the calling party, to become fully established before giving up and tearing it down. The range is from 5 to 300
calledIntegrityTime	The amount of time (in seconds) this node will wait for an SVCC, which it has decided to accept as the called party, to become fully established before giving up and tearing it down. The range is from 10 to 300.

Displaying Routing Policy Parameters

Enter the **dsppnni-timer** command to display the routing policy parameters:

```
Geneva.7.PXM.a > dsppnni-timer
```

The following example shows the report for this command:

```
Geneva.7.PXM.a > dsppnni-timer
node index: 1
   Hello holddown(100ms)...        10     PTSE holddown(100ms)...        10
   Hello int(sec)..........        15     PTSE refresh int(sec)..      1800
   Hello inactivity factor.         5     PTSE lifetime factor...       200
   Retransmit int(sec).....         5
   AvCR proportional PM....        50     CDV PM multiplier......        25
   AvCR minimum threshold..         3     CTD PM multiplier......        50
   Peer delayed ack int(100ms)...................        10
   Logical horizontal link inactivity time(sec)..       120
```

Displaying the SVCC RCC Table

Enter the **dsppnni-svcc-rcc** command to display the PNNI SVCC RCC Table:

```
Geneva.7.PXM.a > dsppnni-svcc-rcc [node-index] [svc-index]
```

If you specify:

- Both node-index and svc-index, command displays information about an SVCC-based RCC.
- Only node-index, command displays all SVC-based RCCs attached to the svc-index node.
- Nothing, command displays all SVC-based RCCs attached to all PNNI nodes on this WAN.

```
Geneva.7.PXM.a > dsppnni-svcc-rcc

Objects Displayed (for each RCC):
``
node index - 32-bit number.
svc index - 32-bit number.
hello state - ascii string.
Down
Attempt
1wayInside
2wayInside
1wayOutside
2wayOutside
Common.
remote node id - 22-byte hex string.
remote node ATM address - 20 byte hex string.
interface index - 32-bit number.
Hello packets received - 32-bit number.
Hello packets transmitted - 32-bit number.
SVCC VPI - 32-bit number.
SVCC VCI - 32-bit number.
```