Update to the Cisco MGX 8850 Routing Switch Software Configuration Guide, Release 2.1

September 4, 2001

Note You can find the most current Cisco WAN Switching Command Reference documentation on Cisco Connection Online (CCO). These electronic documents may contain updates and modifications made after the hardcopy documents were printed.

This update describes the Cisco WAN switching information that has been added or modified for PNNI Software Release 2.1 to support the Migration 1B feature. Use this update with the *Cisco MGX 8850 Routing Switch Software Configuration Guide, Release 2.1*. ENNI, EUNI & XLMI connections can not be provisioned from CWM or CiscoView. You must use the procedures outlined below.

Contents

- "Enhanced NNI" on page 2
- "Enable ENNI (on the PXM)" on page 3
- "Flip an OAM Segment Endpoint for Fault Isolation (on AXSM)" on page 4
- "End-to-End tstdelay Using ENNI" on page 5
- "Fault Isolation in a PNNI Network" on page 6
- "Defect Notification" on page 7
- "Blocking of Continuity Check" on page 8
- "Extended LMI (XLMI)" on page 10
- "Two Segment XPVC connections (MGX Release 1 Feeder to MGX Release 2 Node)" on page 9
- "Add XLMI on an AXSM Port" on page 11
- "Delete XLMI on an AXSM Port" on page 12

CISCO SYSTEMS

Corporate Headquarters: Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Enhanced NNI

Enhanced NNI is useful to customers who wish to extend their Automatic Routing Management networks to include PNNI. The PXM controller uses Enhanced (or Friendly) Network-to-Network Interface (ENNI) to link individual networks of Cisco Automatic Routing Management switches to individual networks of PNNI switches. When a port is configured to be ENNI, all new SPVC/SPVPs added to that port will have the OAM segment configured as non-segment.

Note If there are already existing connections on a port, you cannot configure it from UNI or NNI to ENNI or EUNI, or vice versa. You must delete the connections.

An ENNI connection is referred to as a hybrid connection. A pure Automatic Routing Management connection may become a hybrid connection for an indefinite time during migration before it becomes a pure PNNI SPVC/SPVP connection.

In a regular connection, links joining two network domains serve as a segment endpoint for OAM (segment) loopback cell treatment. However, a hybrid connection requires all OAM cell types to traverse end-to-end without being looped back or terminated at the ingress of the intermediate links. XPVCs and XPVPs are provisioned across ENNI networks to handle segment OAM cells flowing over end-to-end OAM segment loops for the provisioned length of the XPVC or XPVP.

In an Automatic Routing Management to PNNI migration scenario, an ENNI port joins two Cisco networks (typically Automatic Routing Management and PNNI networks). The ENNI port is configured to pass OAM loopback cells to the adjacent network. All interfaces advertised by the AXSM to the PNNI controller are set to be UNI by default.

Note A UNI port converted to ENNI has a VPI range from 0 - 255. An NNI port converted to ENNI has a VPI range from 0 - 4095

Note If an ENNI port is converted to a regular NNI port, all the new connections on the port will be programmed as segment endpoints.

Enable ENNI (on the PXM)

Use the following procedure to enable ENNI on the PXM card. Remember that whether the port is UNI or NNI is defined during AXSM port configuration. Only the "Enhanced" flag is configured on the PXM card.

Step 1 Take the port out of service by entering the **dnpnport** command as follows:

`pswpop2-1.7.PXM.a >` **dnpnport** <*physical interface id*>

Example:

```
dnpnport 3:1.4:4
```

Step 2 Disable LMI on the port by entering the **cnfautocnf** command as follows:

cnfautocnf <*physical interface id*> [**yes/no**]

Example:

```
pswpop2-1.7.PXM.a > cnfautocnf 3:1.4:4 no
```

Step 3 Set the protocol version on the port to ENNI by entering the **cnfpnportsig** command as follows:

cnfpnportsig <*physical interface id*> **-nniver enni**

Example:

```
pswpop2-1.7.PXM.a > cnfpnportsig 3:1.4:4 -nniver enni
```

Step 4 Bring the port back into service by entering the **uppnport** command as follows:

uppnport <*physical interface id*>

Example:

```
pswpop2-1.7.PXM.a > uppnport 3:1.4:4
```

Step 5 Display all PNNI ports by entering the **dsppnports** command as follows:

dsppnports

Example:

```
pswpop2-1.7.PXM.a > dsppnports
```

To display a specific PNNI port, enter the **dsppnport** command as follows:

dsppnport <*physical interface id*>

Example:

```
pswpop2-1.7.PXM.a > dsppnport  1:1.3:1

Port:           1:1.3:1              Logical Id:     16848897
IF status:      down                 Admin Status:     down

Auto-config:    disable              Addrs-reg:        enable
IF-side:        network              Operational IF-type:  enni
UniType:        private              version:              none
Input filter:   0                    Output filter:    0
minSvccVpi:             0            maxSvccVpi:         4095
minSvccVci:            32            maxSvccVci:        65535
minSvpcVpi:             1            maxSvpcVpi:         4095
      #Svcc:   #Svpc:  #SpvcActive:  #SpvcCfg:  #SpvpActive:  #SpvpCfg:
Total:
p2p :     0       0              0          0              0        0
p2mp:     0       0              0          0              0        0\
```

Flip an OAM Segment Endpoint for Fault Isolation (on AXSM)

Use the following procedure to flip an OAM segment endpoint for fault isolation.

Step 1 Enable the OAM segment endpoint by entering the **cnfcon** command as follows:

`pswpop2-1.7.AXSM.a >` **cnfcon** <*logical interface id*> <*vpi*> <*vci*> **-segep <enable/disable>**

Example:

```
pswpop2-1.7.AXSM.a > cnfcon 1 1 100 -segep enable
```

Step 2 Verify the segment endpoint is enabled by entering the **dspcon** command as follows:

dspcon <*logical interface*> <*vpi*> <*vci*>

Example:

```
pswpop2-1.7.AXSM.a > dspcon 2 20 1000
------------------------------------------------------------------------
Local   :                NSAP  Address                    vpi       vci
(S)       47009181000000000107B65F27C00000101180200         20      1000
Remote  :                NSAP  Address                    vpi       vci
(M)       47009181000000000107B65F30C0000010B180200         20      1000
------------------------------------------------------------------------
Conn. Type    :      VCC                          Admn Status  :  ADMN-UP
Service Type  :   abrstd                          Oper Status  :       OK
Controller    :        2                          Record #     :       18
------------------------------------------------------------------------
Local PCR     :       50                          Remote PCR   :       50
Local SCR     :      N/A                          Remote SCR   :      N/A
Local CDV     :      N/A                          Remote CDV   :      N/A
Local CTD     :      N/A                          Remote CTD   :      N/A
Local MBS     :      N/A                          Remote MBS   :      N/A
Max Cost      :       -1                          Frame discard:        N
Local CDVT    :   250000
------------------------------------------------------------------------
OAM CC Config : DISABLED                          Statistics   : DISABLED
------------------------------------------------------------------------
Loopback Type : No  Lpbk | Dir: N/A     | Status: No Lpbk | RTD:      0us
------------------------------------------------------------------------

Type <CR> to continue, Q<CR> to stop:
```

End-to-End tstdelay Using ENNI

By default, the connections originating and terminating on ENNI ports in Automatic Routing Management and PNNI domains are non-segment endpoints. This allows segment loopback cells to pass to the adjoining network. When end-to-end **tstdelay** fails, provisioned connections on ENNI ports identify the faulty network as a segment endpoint instead of the non-segment endpoint.

Use the following procedure to run **tstdelay** on ENNI ports.

Step 1 Test the integrity of the connection by entering the **tstdelay** command on the AXSM card as follows:

tstdelay <*portid*> <*vpi*> <*vci*>

Example:

```
pswpop2-1.7.AXSM.a > tstdelay 1.1 10 100
```

Step 2 To isolate a fault on the ENNI port, configure the connection in the PNNI Network as a segment endpoint by entering the the **cnfcon** command as follows:

cnfcon <*logical interface id*> <*vpi*> <*vci*> -**segep** <*enable*>

Example:

```
pswpop2-1.7.AXSM.a > cnfcon 1 1 100 -segep enable
```

Step 3 Execute the **tstdelay** command again as follows:

tstdelay <*portid*> <*vpi*> <*vci*>

Example:

```
pswpop2-1.7.AXSM.a > tstdelay 1.1 10 100
```

If **tstdelay** fails again, the fault is in the Automatic Routing Management network. If **tstdelay** passes, proceed to Step 4.

Step 4 To isolate the fault further, reconfigure the segment endpoint as non-segment endpoint by entering the **cnfcon** command as follows:

cnfcon <*logical interface id*> <*vpi*> <*vci*> -**segep** <*disable*>

Example:

```
pswpop2-1.7.AXSM.a > cnfcon 1 1 100 -segep disable
```

This allows the cell to pass through

Step 5 At the far end of the PNNI network, configure the connection on the ENNI port as a segment endpoint by entering the **cnfcon** command as follows:

cnfcon <*logical interface id*> <*vpi*> <*vci*> -**segep** <*enable*>

Example:

```
pswpop2-1.7.AXSM.a > cnfcon 1 1 100 -segep enable
```

Step 6 Execute the **tstdelay** command again as follows:

tstdelay <*portid*> <*vpi*> <*vci*>

Example:

```
pswpop2-1.7.AXSM.a > tstdelay 1.1 10 100
```

If **tstdelay** passes on the far end of the PNNI network, the fault lies in the Automatic Routing Management network. If **tstdelay** fails, the fault is in the PNNI network.

Note If **tstdelay** is executed at the ENNI port, it will be blocked unless it is configured as segment endpoint.

Note The connections which are flipped for fault isolation can be reversed back to their original configuration.

Fault Isolation in a PNNI Network

To do fault isolation on a PNNI segment, configure the OAM segment of the SPVC endpoints as OAM segment endpoints. The OAM segment must be configured on both the master and slave endpoints.

Run the following procedure on the AXSM card to do fault isolation on a PNNI Network:

Step 1 Enable the OAM segment endpoint by entering the **cnfcon** command as follows:

cnfcon <*logical interface id*> <*vpi*> <*vci*> -**segep** <*enable*>

Example:

```
pswpop2-1.7.AXSM.a > cnfcon 1 1 100 -segep enable
```

Step 2 Verify the continuity of the connection by entering the **tstdelay** command as follows:

tstdelay <*portid*> <*vpi*> <*vci*>

Example:

```
pswpop2-1.7.AXSM.a > tstdelay 1.1 10 100
```

Step 3 Verify the OAM segment configuration by entering the **dspcon** command as follows:

dspcon <*portid*> <*vpi*> <*vci*>

Example:

```
pswpop2-1.7.AXSM.a > dspcon 1.1 10 100
--------------------------------------------------------------------------
Local   :                NSAP  Address                     vpi       vci
(S)       47009181000000000107B65F27C00000101180200          20      1000
Remote  :                NSAP  Address                     vpi       vci
(M)       47009181000000000107B65F30C0000010B180200          20      1000
--------------------------------------------------------------------------
Conn. Type    :      VCC                          Admn Status  :  ADMN-UP
Service Type  :   abrstd                          Oper Status  :       OK
Controller    :        2                          Record #     :       18
--------------------------------------------------------------------------
Local PCR     :       50                          Remote PCR   :       50
Local SCR     :      N/A                          Remote SCR   :      N/A
Local CDV     :      N/A                          Remote CDV   :      N/A
Local CTD     :      N/A                          Remote CTD   :      N/A
Local MBS     :      N/A                          Remote MBS   :      N/A
Max Cost      :       -1                          Frame discard:        N
Local CDVT    :   250000
--------------------------------------------------------------------------
OAM CC Config : DISABLED                          Statistics   : DISABLED
--------------------------------------------------------------------------
Loopback Type : No  Lpbk | Dir: N/A     | Status: No Lpbk | RTD:      0us
--------------------------------------------------------------------------

Type <CR> to continue, Q<CR> to stop:
```

Once fault isolation is done, configure the OAM segment back to no segment endpoint.

Step 4 Generate segment loopback cells to the CPE by entering the **tstconseg** command as follows:

tstconseg <*ifNum*> <*vpi*> <*vci*>

Example:

```
pswpop2-1.7.AXSM.a > tstconseg 1 10 100

Test started; Use dspcon/dspcontests to see test results
```

Note **tstconseg** will be blocked at an ENNI port if the connection was configured as non-segment.

Defect Notification

Hybrid networks use Alarm Indication Signal (AIS) cells and Remote Defect Indication (RDI) cells for defect notification in PNNI.

The following sections describe the results of different defect cases in hybrid networks:

- End-to-End (E2E) AIS cells coming in from a CPE or Automatic Routing Management network.
- Connection derouted due to trunk failure.
- ENNI Port down.

End-to-End (E2E) AIS Cells Coming in from CPE or Automatic Routing Management Network

When the AXSM card detects E2E AIS cells coming into the PNNI network from a CPE or Automatic Routing Management network, it passes AIS cells to the next network and reports channel traps to CWM. The PNNI Network forwards RDI cells to the CPE.

Connection Derouted Due to Trunk Failure

When there is trunk failure in a PNNI Network, the AXSM card generates End to End AIS cells in both the directions. The ENNI side of the connection sends individual channels to CWM.

ENNI Port Down

Table 1 describes four scenarios that can cause an ENNI port to be in a downed state.

Table 1 ENNI Port Down Scenarios

Scenario	Result
dnpnport command is run on PXM	The AXSM generates AIS cells in both directions for all the connections on the port. AIS cells are generated from ENNI port and trunk side of the connection.
ENNI port cable pulled out	The AXSM generates AIS cells where the cable is pulled out. The AIS cells travel towards the remote end of the connection. A port trap is sent for the ENNI Port on which **dnpnport** command is issued. This trap triggers the service module to generates AIS cells toward the remote end of the connection. The remote end of the connection sends individual connection traps to CWM.
AXSM card pulled out	The AXSM generates AIS cells towards the remote end of the connection. The AIS cells are generated from the ENNI port. A card pullout trap is sent to CWM if the card is pulled out. The remote end of the connection sends individual connection traps to CWM.
dnpnport command on AXSM	The AXSM generates AIS cells towards the remote end of the connection. The AIS cells are generated from the ENNI port. A port trap is sent to the ENNI Port on which **dnpnport** command is issued. The remote end of the connection sends individual connection traps to CWM.

Blocking of Continuity Check

The AXSM card supports a Continuity Check (CC) function to detect whether the far end of the connection is still alive. A CC failure generates AIS cells.

Use the **cnfcon** command to enabled/disable the CC function at the time you add the Master or Slave end of a connection. The CC function is added on a per connection basis, and is set to **disable** by default.

Note The Continuity Check (CC) function can not be enabled at an ENNI termination point. The Continuity Check (CC) function on ENNI endpoint is blocked, if the connection is non-segment during XPVC provisioning. The Continuity Check (CC) function on ENNI endpoint is blocked, if the connection is non-segment, during cnfcon of SPVC."

Two Segment XPVC connections (MGX Release 1 Feeder to MGX Release 2 Node)

Table 2 describes the results of different failure cases for two segment XPVC connections (Automatic Routing Management segment and PNNI segment).

Table 2 Two Segment XPVC Failure Cases

Scenario	Result
E2E AIS cells coming in from a CPE or Automatic Routing Management network	An A-bit LMI message and E2E AIS cells are sent to the MGX feeder. The MGX feeder sends a channel fail trap to CWM.
Connection derouted in a PNNI Network due to trunk failure.	AXSM card generates AIS cells in both directions. An A-bit LMI message and AIS cells are sent to the MGX feeder. A channel fail trap is sent from the MGX feeder to CWM.
Feeder trunk failure between the MGX feeder and another MGX.	AXSM card detects loss of signal. A port trap is sent to CWM for the failed port and the AXSM card generates AIS cells on the MGX. On the MGX feeder, AIS cells are generated towards the CPE.
UNI port failure on MGX feeder due to line card pulled out on the UNI side.	MGX feeder generates ian A-bit messages to the AXSM card. The AXSM card marks the connections in A-bit alarm, but does not generate AIS cells. Therefor, the far end of the connection does not know about the failed connection.
Connection Failure on MGX feeder side	MGX feeder generates A-bit and AIS cells towards the MGX. The MGX feeder sends a connection fail trap to CWM.

Extended LMI (XLMI)

On the MGX 8850 switch, the AXSM card uses the XLMI feature for Automatic Routing Management coexistence with PNNI. XLMI supports the following functions in the network:

- network topology discovery for CWM.
- connection addition/deletion status reports sent to the adjoining network.

XLMI on the MGX 8850 has the following limitations:

- The XLMI feature is supported only on AXSM cards.
- A maximum of 16 feeder nodes or Automatic Routing Management networks combined can be configured with XLMI. The maximum number of feeders per node is 16. The maximum number of ports that can support Automatic Routing Management is limited to the maximum number of physical ports configured on the AXSM card.
- When the port is configured for XMLI in an Automatic Routing Managementnetwork, IP connectivity must be disabled.
- XLMI cannot be provisioned on a port which already has connections provisioned.
- The various XLMI timers are not configurable on the AXSM. Timer configuration is done on the BPX. The values for the LMI timers on AXSM are:

  LMI Timers :

  - SPVC Status Enquiry Timer :10sec
  - SPVC Update Status Timer :10sec
  - Retry Timers :5sec

Add XLMI on an AXSM Port

Before adding XLMI on an AXSM port, consider the following limitations:

- XLMI can be added only to a port which exists for the interface number.
- XLMI can not be added to a port which already has ILMI enabled.
- XLMI can not be enabled on a port which has LMI enabled already.

Use the following procedure to add XLMI on a port.

Step 1 At the AXSM card, verify whether the port already has LMI enabled by entering the **dsplmi** command as follows:

dsplmi <*ifNum*>

Example:

```
MGX2.2.1.7.PXM.a > dsplmi 1
LMI Interface Number:1
Remote Network IP:172.29.25.203
Remote End Shelf:1
Remote End Slot:3
Remote End Port:1
Remote Model Number:8860
LMI Configuration:Up
LMI Link Status:Up
```

Step 2 If LMI is not enabled on the port, proceed to Step 3.

If LMI is enabled on the port, remove all connections that exist on the port, then disable LMI by entering the **dellmi** command as follows:

dellmi <*ifNum*>

Example:

```
MGX2.2.1.7.PXM.a > dellmi 1
```

Note The dellmi command will succeed only if no connections exist on the specified port.

Step 3 Enable XLMI on the port by entering the **addlmi** command as follows:

addlmi <*ifNum*> **-type** <*fdr*| *xlmi*>

Example:

```
MGX2.2.1.7.PXM.a > addlmi 1 2 xlmi
```

Step 4 Verify your XLMI configuration on the AXSM port by entering the **dspilmi** command as follows:

dsplmi <*ifNum*>

Example:

```
MGX2.2.1.7.PXM.a > dsplmi 1
LMI Interface Number:1
Remote Network IP:172.29.25.203
Remote End Shelf:1
Remote End Slot:3
Remote End Port:1
Remote Model Number:8860
LMI Configuration:Up
LMI Link Status:Up
```

Delete XLMI on an AXSM Port

Use the following procedure to delete XLMI on an AXSM port.

Step 1 Disable XLMI on the port by entering the **dellmi** command as follows:

dellmi <*ifNum*>

Example:

```
MGX2.2.1.7.PXM.a > dellmi 1
```

Step 2 Verify that XLMI was deleted from the AXSM port by entering the **dsplmi** command as follows:

dsplmi <*ifNum*>

Example:

```
MGX2.2.1.7.PXM.a > dsplmi 1
LMI Interface Number:1
Remote Network IP:172.29.25.203
Remote End Shelf:1
Remote End Slot:3
Remote End Port:1
Remote Model Number:8860
LMI Configuration:Up
LMI Link Status:Up
```