CHAPTER 5

Configuring the MGX 8850 (PXM1-based) Switch

This chapter describes switch-level tasks for bringing up and configuring an MGX 8850 (PXM1-based) switch. The tasks begin at the point where all hardware is in place and the power is on and alarm-free. These initial tasks require you to use the command line interface (CLI) on an ASCII terminal. You can use either the CiscoView application or the CLI to continue with switch configuration.

- "Bringing Up an MGX 8850 (PXM1-based) Switch" introduces initial bring-up of the MGX 8850 (PXM1-based) switch.
- "Switch Access Ports" describes the role of each switch control port.
- "Initial Switch Bring-Up" is a list that identifies the tasks for configuring the switch.
- "Bringing Up a PXM with No Runtime Firmware" describes how to load runtime firmware.
- "Configuring Node-Level Parameters" describes how to configure node-level parameters.
- "Downloading Firmware to a Service Module" describes service module firmware loading.

Bringing Up an MGX 8850 (PXM1-based) Switch

This section describes the initial configuration steps for a new switch from Cisco. The tasks in this section apply to the switch whether the use is a feeder or stand-alone node. The topics are:

- How the access ports for controlling the switch relate to configuration.
- Initial bring-up of a Processor Switching Module (PXM) that has no runtime (or "on-line") firmware image.
- Configuration tasks for the PXM with runtime firmware.

Note The words "switch," "node," and "shelf" are synonymous for the MGX 8850 (PXM1-based) product. The word "bay" is the upper or lower half of the enclosure.

Note To add an MGX 8850 (PXM1-based) feeder to the network, you execute **addshelf** at the BPX 8600-series switch. See Appendix C, "Switch Commands That Apply to the MGX 8850 Node," for a description of this and related commands.

Switch Access Ports

Three external ports exist for controlling the switch through the Processor Switching Module User Interface card (PXM-UI):

1. Through the *control port* (sometimes called the console port), you can use the command line interface (CLI) on an ASCII terminal. The purpose of this port is:
   - Initial assignment of IP addresses to the Ethernet port, maintenance port, the inband ATM IP address, and the IP address of the statistics manager. The ATM IP address belongs to the link between the PXM and the BPX 8600-series switch and applies to only the feeder application of the MGX 8850 (PXM1-based) node. Note also that, for a stand-alone application, only the workstation connected to the switch can detect these IP addresses.

     Before you use the CiscoView or the Cisco WAN Manager (formerly StrataView Plus) network management applications, the IP addresses you intend for the switch must reside on the workstation in the *etc/hosts* file. Also, the text file *config.sv* on the workstation must contain the name of the switch you intend to be the gateway node, the network ID, the network name, and so on. See the Cisco WAN Manager documentation for the file system requirements on the workstation.
   - Low-level control or troubleshooting. (You can also use the CLI through a window in the Cisco WAN Manager application.)

Note When you use the CLI, you must *type* all required parameters and any optional parameters before you press *Return* or *Enter*.

2. Through the Ethernet port, you can use a workstation running a Cisco network management application such as the Cisco WAN Manager or CiscoView application. Typically, the workstation on a LAN is co-located with the MGX 8850 (PXM1-based) switch.
3. Through the maintenance port (sometimes called the modem port), you can connect either a single workstation running an IP-based application or a terminal server that supports multiple workstations. The workstation must support SLIP. Typically, use of this port includes a modem because the switch resides at a remote location. The typical applications are software and firmware download or tasks that require low-level access.

Other ports exist on the PXM-UI. These ports support external clock sources and external, third-party audio or visual alarm systems. For information on the function of other ports on the PXM-UI, see Chapter 6, “Card and Service Configuration.”

The maintenance port and Ethernet port support IP-based applications. Through these ports, the following applications run:

- Telnet supports CLI command execution from any IP-based application window as well as a window in the Cisco WAN Manager application.
- TFTP lets you download firmware and upload and download configuration information.
- SNMP supports equipment management through the CiscoView application and connection management through the Cisco WAN Manager application.

Note Although the Cisco WAN Manager itself does not communicate with the maintenance port, this port supports IP-based communication.

Initial Switch Bring-Up

This section describes how to start up the switch for the first time. It begins with a PXM that has only boot-mode firmware. The descriptions tell you how to:

1. Establish communication with the switch.
2. Configure one or more boot-level IP addresses to make the switch available to the network.
3. Download PXM firmware.
4. Configure a new, switch-level Ethernet IP address for the PXM as needed or other SLIP or IP addresses.
5. Specify a name for the switch.
6. Specify the time on the switch.
7. Optionally configure a time zone for the Western Hemisphere, or configure a time zone relative to Greenwich Mean Time if the switch resides outside the Western Hemisphere.
8. Download firmware to the service modules.

If the PXM has no runtime (or "on-line") firmware image, begin with the boot-mode description in the section titled "Bringing Up a PXM with No Runtime Firmware." If the PXM has a runtime firmware image, go to the section titled "Configuring Node-Level Parameters."

Bringing Up a PXM with No Runtime Firmware

The section describes the tasks for loading runtime firmware onto a PXM that has only a boot loader.

Step 1 Establish communication with the switch by doing one of the following:

- If you are using an ASCII terminal connected to the control port, the prompt for the next command is already present upon power-up. (If the display is skewed, make sure the terminal speed and PXM-UI port speeds are the same.)
- If you are using a utility such as Hyper Terminal on a PC, the firmware may reside on either a floppy or the hard drive.

Step 2 Execute the command **bootChange** to configure boot-level IP parameters.

If the switch has a redundant PXM, execute **bootChange** on each PXM to configure unique, boot-level IP addresses. (During the subsequent switch-level configuration, you must configure another Ethernet IP address that applies to *both* PXMs.) The following are the only parameters that are meaningful at this point, so press *Return* for other parameters:

- Mandatory "host name" is a name for the workstation. For the MGX 8850 (PXM1-based) node, enter the letter "c."
- Ethernet IP address and subnet mask for the PXM LAN port are mandatory (see "inet on Ethernet" in the following example). Follow the IP address with a colon and a net mask. The netmask is eight hexadecimal numbers with no embedded periods. Do not type spaces on either side of the colon.
- If the workstation from which you download firmware is on a subnet other than the subnet of the PXM, enter a gateway IP address ("gateway inet").

Note the three editing functions near the top of the following example. Of these, typing a period to clear the current field is the most commonly used.

```
>bootChange
'.' = clear field;  '-' = go to previous field;  ^D = quit
boot device          : lnPci
processor number     : 0
host name            :c
file name            :
inet on ethernet (e) : 188.29.37.14:ffffff00
inet on backplane (b):
host inet (h)        :
gateway inet (g)     : 188.29.37.1
user (u)             :
ftp password (pw) (blank = use rsh):
flags (f)            : 0x0
target name (tn)     :
startup script (s)   :
other (o)            :
```

The PXM now has a boot-level IP address. Remember to repeat the **bootChange** command on the redundant PXM if the system has one.

Step 3 Enter **reboot** to reset the PXM.

The PXM is ready to receive a firmware image through the Ethernet port. Use the workstation for the next steps.

Step 4 At the workstation, you can optionally ping the PXM using the IP address to confirm that the node is reachable.

Step 5 Establish communication with the PXM according to the user-communication device type. For example, at the prompt on a UNIX workstation, you could enter:

>tip -9600 /dev/ttya

The device specification could also be ttyb.

Step 6 Enter the **tftp** command with the IP address set at the ASCII terminal. For example, if the console port is connected to the serial port of the workstation:

$**tftp** 162.29.38.101

Step 7 At the tftp prompt, enter binary mode:

>bin

Step 8 From the directory where the firmware resides, enter the **put** command and include the arguments that specify the firmware release number, the statement that this firmware applies to the active PXM, and the release directory.

If necessary, refer to the release notes for new firmware release numbers. The entries are case-sensitive. For example:

>put pxm_*release_number*.fw POPEYE@PXM_ACTIVE.FW

where *release_number* is a decimal number in the form *n.n.nn.* Currently, the initial *n* typically is a "1." An example filename for PXM firmware is "pxm_1.0.03." Note that the download automatically includes the firmware for the standby PXM (if present). You can subsequently see POPEYE@PXM_STANDBY.FW in c:/FW.

Check the console to verify that the transfer completed and the checksum passed.

Step 9 Quit the tftp application, then go to the ASCII terminal connected to the control port:

>quit

Step 10 At the ASCII terminal, **cd** to FW directory on the hard drive.

Step 11 List the contents to confirm that the firmware resides in the FW directory:

>cd "c:/FW"

>ll

Note these required quote marks are absent when you use the CLI after you reboot the PXM with its runtime image (see "Configuring Node-Level Parameters").

Step 12 Enter the following:

>setPXMPrimary "*version*"

where *version* is the version number of the firmware. The name of a PXM firmware file has the format pxm_*version*.fw. For example: in PXM_1.0.03.fw, *version* is 1.0.03.

Step 13 Reboot the system again:

>reboot

A login prompt appears on the ASCII terminal. The PXM is now the same as a PXM that Cisco ships with a runtime firmware image.

Configuring Node-Level Parameters

Except for adding a user and creating a password, all the tasks described in this section can be performed through the CiscoView application. For descriptions of the commands you enter at the CLI, see the *Cisco MGX 8850 (PXM1-based) Wide Area Edge Switch Command Reference*. A representation of the feeder application of the MGX 8850 (PXM1-based) switch appears in Figure 5-1. A representation of the stand-alone application of the switch appears in Figure 5-2.

Figure 5-1 Feeder Application

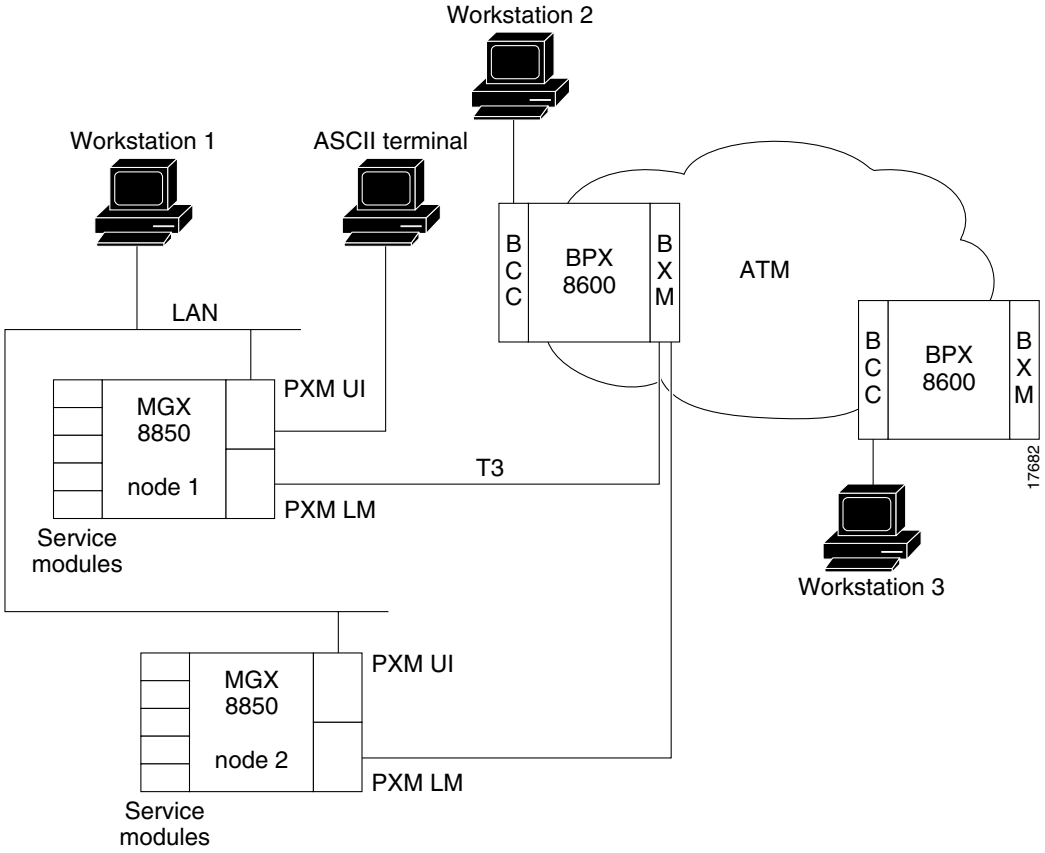
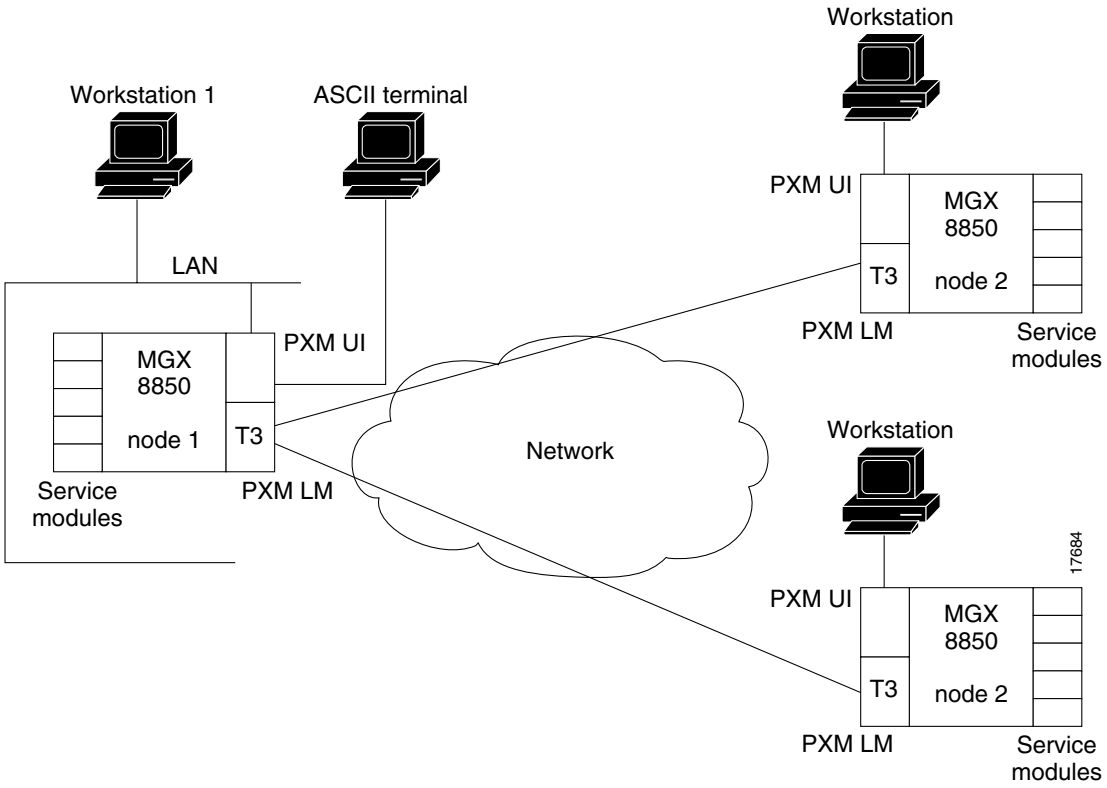


Figure 5-2 Stand-Alone Application



At the CLI prompt on the ASCII terminal:

Step 1 Enter the default login and password provided in the release notes.

The terminal displays the slot number of the PXM you have logged into by default:

card number [7]:

Step 2 Press *Return* to enter the CLI of this PXM.

At runtime, you could also enter the slot number of a service module or a standby PXM. In this case, the CLI prompt shows:

NODENAME.1.7.PXM.a>

where NODENAME shows that the node has no name; the slot number of the PXM is 7; and this PXM is active. The general format of the CLI prompt is:

nodename.1.*slot*.*cardtype*.a>

where *nodename* is the name of the node; the shelf (node) number is always 1; *slot* is the card location; *cardtype* identifies the card; and the card state is active (a) or standby (s).

Step 3 Display the cards in the system:

NODENAME.1.7.PXM.a> **dspcds**

Step 4 Display any IP addresses in the system:

NODENAME.1.7.PXM.a> **dspifip**

Step 5 Change any IP addresses as needed:

NODENAME.1.7.PXM.a> **cnfifip** <*interface*> <*IP_Addr*> <*Net_Mask*> [*BrocastAddr*]

where *interface* is a number: 26 is the Ethernet (LAN AUI) port, 28 is the maintenance port (SLIP), or 37 for the ATM IP address (feeder application only). Note that *BrocastAddr* applies to only the Ethernet interface (number 26).

Note Check the Release Notes for any variations in how to configure IP addresses.

Step 6 Execute the **cnfname** command to assign a name to the switch:

UNKNOWN.1.7.PXM.a> **cnfname** <*node name*>

where *node name* is a case-sensitive name up to eight characters. For example:

UNKNOWN.1.7.PXM.a> **cnfname** cisco22

Step 7 Execute the **cnftime** command to specify the time on the switch:

cisco22.1.7.PXM.a> **cnftime** <*hh:mm:ss*>

where *hh* is the hour of the day in the range 1–24; *mm* is the minute of the hour in the range 1–60; and *ss* is the number of seconds in the minute and has a range of 1–60.

Step 8 Optionally configure a time zone for the node. Use **cnftmzn** to specify a time zone in the Western Hemisphere. To configure a time zone outside the Western Hemisphere, first specify Greenwich Mean Time (GMT) with **cnftmzn** then specify the offset from GMT by using **cnftmzngmt**:

- cisco22.1.7.PXM.a> **cnftmzn** <*timezone*>

where *timezone* is 1 for GMT, 2 for EST, 3 for CST, 4 for MST, 5 for PST.

- cisco22.1.7.PXM.a> **cnftmzngmt** <*timeoffsetGMT*>

where *timeoffsetGMT* is the offset in hours from GMT. The range of possible values for *timeoffsetGMT* is -12 through +12.

Step 9 Execute the **cnfstatsmgr** command to specify the IP address of the workstation that runs the Cisco WAN Manager application.

Before it sends statistics, the MGX 8850 (PXM1-based) node must have the IP address of the workstation with this application. The syntax is:

>**cnfstatsmgr** <*IP_Addr*> where *IP_Addr* is the IP address of the workstation.

If the node has a redundant PXM, it automatically receives the same IP addresses and configuration as the primary PXM. With the IP addresses in place, you can configure the logical ports for the broadband interface through the CiscoView application or the CLI.

Step 10 Add one or more users by executing **adduser** once for each new user.

Note that the access privilege level is case-sensitive as the syntax description indicates. After you enter the privilege level, the system prompts for a new password for the user. (This password parameter does not appear in the help information for **adduser**.)

adduser <*user_Id*> <*accessLevel*>

user_Id is 1–12 alphanumeric characters.

accessLevel is the case-sensitive privilege level. It can be ANYUSER or within the range GROUP1–GROUP5. For example, to specify a privilege level 2, type GROUP2.

After you enter a user-name and privilege level, the system prompts for a password. The password is a string of 5–15 characters. If you press Return without entering a password, the system assigns the default password “newuser.”

Step 11 Optionally change your password or another user's password by executing:

cnfpasswd [*username*]

username is the name of another user whose password you are changing. That user must have a privilege level that is lower than your privilege. To change your own password, enter **cnfpasswd** with no *username*.

Step 12 To specify the switch as a feeder, execute the **cnfswfunc** command:

cnfswfunc <*-ndtype*>

and follow -ndtype with "fdr."

Step 13 Configure as needed an external clock by executing **cnfextclk**.

Step 14 Configure as needed double-speed clocks for individual Cellbuses by using **cnfcbclk**:

cnfcbclk <*cellBus*> <*clockRate*>

cellBus is a string in the range CB1–CB8 that identifies the Cellbus.

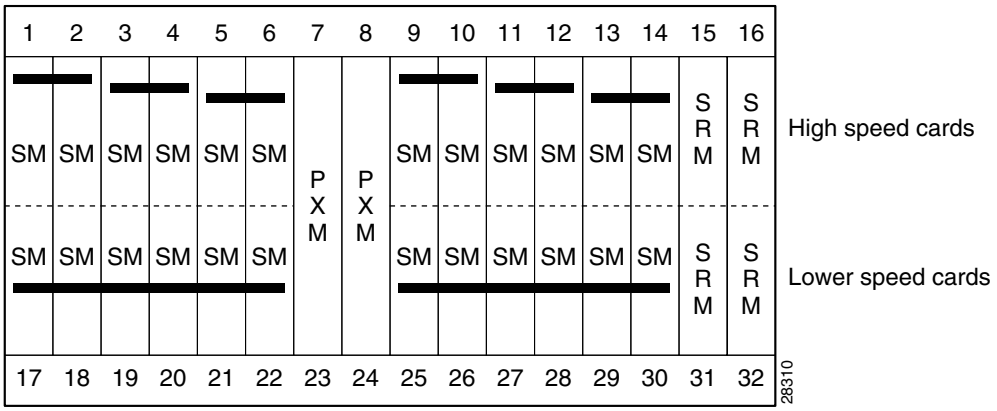
clockRate is a number that identifies the rate in megahertz. Choices are 21 or 42.

The distribution of the eight Cellbuses follows.

- Cellbus 1 (CB1) connects slots 1 and 2 to the PXM1
- Cellbus 2 (CB2) connects slots 3 and 4 to the PXM1
- Cellbus 3 (CB3) connects slots 5 and 6 to the PXM1
- Cellbus 4 (CB4) connects slots 9 and 10 to the PXM1
- Cellbus 5 (CB5) connects slots 11 and 12 to the PXM1
- Cellbus 6 (CB6) connects slots 13 and 14 to the PXM1
- Cellbus 7 (CB7) connects slots 17, 18, 19, 20, 21 and 22 to the PXM1
- Cellbus 8 (CB8) connects slots 25, 26, 27, 28, 29 and 30 to the PXM1

In the top bay, each of the six Cellbuses serves two card slots. In the bottom bay, each of the two Cellbuses serves six card slots. Therefore, each top slot has three times the available bandwidth of the lower slots and is therefore better suited to the higher speed cards. The bandwidth concentration for Cellbuses in the upper and lower bays is illustrated in Figure 5-3.

Figure 5-3 Bandwidth Concentration in Upper and Lower Bays



Downloading Firmware to a Service Module

This section describes how to download firmware for a service module from a workstation. The descriptions apply whether you are upgrading the existing firmware or downloading because no runtime firmware resides on the hard drive.

Service modules do not retain runtime firmware. The hard drive on the PXM may come with default firmware for the service modules, but the details of the customer order actually determine whether firmware is on the disk. If default firmware exists on the hard drive, the PXM downloads it upon power-up or when you reset the card, otherwise you can download firmware from the workstation according to the instructions that follow.

Note that if you download firmware from a workstation to the hard drive, the PXM does not automatically load the firmware to the card. You must reset the card (**resetcd** on the CLI) to download firmware from disk to the card. With the single execution of a command, you can load either generic firmware for all cards of a certain type or firmware destined to a specific slot.

To load service module firmware from a workstation to the hard drive on the PXM:

Step 1 Start the tftp application:

$**tftp** <*IP address*>

then

>**bin**

Step 2 To download generic firmware for a type of service module to the PXM hard drive:

>**put** *cardtype*.fw POPEYE@SM_1_0.FW

where *cardtype* is the firmware for a type of card; the shelf number always is 1; and the 0 represents the slot number for the purpose of generic download. An example of *cardtype*.fw is "frsm8t1e1_10.0.11.fw." Note the space between ".fw" and "POPEYE."

Step 3 To load slot-specific firmware at a particular card:

>**put** *cardtype*.fw POPEYE@SM_1_*slot*.FW

where *cardtype* is the firmware, and *slot* is the number of the card slot. Note the space between ".fw" and "POPEYE." Repeat this step for each slot as needed.

Note Slot-specific firmware overwrites the current firmware at a slot.

With slot-specific firmware, the card does not come up if you do either of the following:

- Specify the wrong firmware, where the firmware specified by *cardtype* does not match the targeted card at *slot.*
- Insert a different card (which does not use the firmware specified for the slot).

An example command for downloading specific firmware for an FRSM-2CT3 in slot 3 is:

>**put** frsm2ct3_10.0.01.fw POPEYE@SM_1_3.FW

where "frsm2ct3_10.0.0" refers to the firmware for the FRSM-2CT3, and "3" is the slot.

Note See the Release Notes for current names of firmware files and release directories.

Step 4 When you have finished downloading firmware, enter **quit** to quit the tftp application.

Step 5 At the CLI on either the workstation or the ACSII terminal, display the firmware files. Note that the directory specification **ll** c:/FW has no quote marks.

cisco22.1.7.PXM.a> **ll** c:/FW

Step 6 If you want to download the firmware from the disk to a card, execute **resetcd**.