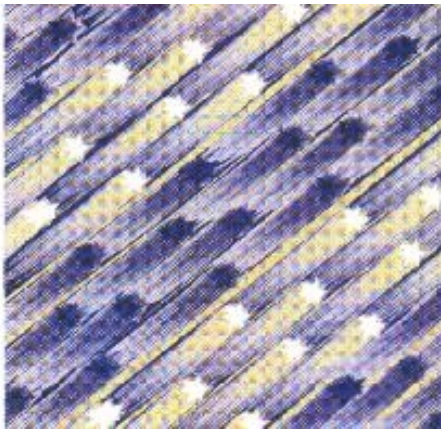


Setting up a Network with 8277s and 8265s - Part I

September 10, 1999

Joao Molina

IBM Networking Hardware Division
Product Installation Support
Cary, North Carolina

Date Last Changed: September 2, 1999

Abstract

This document, which consists of Part I and Part II, describes my experience in installing an Ethernet network with the purpose of testing various basic functions and configurations supported by the products involved.

The network consisted of an ATM backbone network with the IBM 8265 Nways ATM Switch, and Ethernet switches at the edge of the network, with the IBM 8277 Nways Ethernet RouteSwitch. Routing was performed by the IBM 8210 Nways Multiprotocol Switched Services Server (MSS), which provided LAN Emulation and Classical IP services as well.

All the steps are described in great detail, and it covers all the activities from upgrading micro code to the final operational configuration.

(number of pages 54)

Table of Contents

Preface

The objective of this document is to describe in great detail, my experience in installing a test network consisted of an ATM backbone and Ethernet switches at the edge, with the purpose of verifying how basic functions are configured, and how the network behaves when some traffic flows across it.

Traffic was generated using Chariot, a Ganymede Software Inc product.

Two different network designs are covered. A basic design with 3 VLAN Groups, 3 IP subnets, and MSS doing the routing, and a more 'advanced' design, using the concept of one arm router, with 1 VLAN Group, 3 IP subnets, and using Super ELANs in MSS.

Keywords

8210, 8265, 8277, Ethernet, ATM, Super ELAN.

Product List

IBM 8210 Nways Multiprotocol Switched Services Server (MSS)
IBM 8265 Nways ATM Switch
IBM 8277 Nways Ethernet RouteSwitch

For questions related to this document, please send an e-mail to:

Joao Molina
molina@us.ibm.com

Chapter 1: Overview

This document reports my experience in installing a test network with the purpose of verifying functions, products and concepts. It is not the intent of this document to suggest that this is the only and the best way of designing ethernet networks, but rather pass along hints and tips, and explain step-by-step, how to upgrade code and configure the network equipment used in this project.

The major network building blocks used to build the test network were the IBM 8210 Nways Multiprotocol Switched Services Server (MSS), the IBM 8265 Nways ATM Switch, and the IBM 8277 Nways Ethernet RouteSwitch.

The following is a list of the activities performed during the project, and described in great level of detail:

- upgrade 8265 CPSW boot and operational micro code
- upgrade 8265 modules FPGA pico code
- 8265 configuration
- upgrade 8210 firmware
- upgrade 8210 operational code
- 8210 configuration
- 8277 configuration 1 (configuration A - basic)
- 8277 configuration 2 (configuration B - one arm router)

Network Design

It is been a very common approach when designing networks, to have a fast and reliable backbone network built with the ATM technology, and using Fast Ethernet switches as a mean to provide network access to workstations, at a reasonable price. Switching is another technology very often used in network designs, since it improves bandwidth availability to the workstations. Those were the main technologies used in our test network. Figure 01 shows the network diagram (physical connections).

The test network simulates a campus environment, consisting of two buildings.

Three 8265 ATM switches build the backbone of the network. Two 8265s are located in Building A, *SW8265#1 and SW8265#2.* SW*8265#3* is located in building B. PNNI interconnects the switches.

Looking at the network diagram, we can clearly see that there is always an alternate path, in the case of a link failure between 8265s. For instance, in the event of a link failure between the switches SW8265#2 and SW8265#3, data will flow normally, because PNNI will reroute the traffic through the switch SW8265#1.

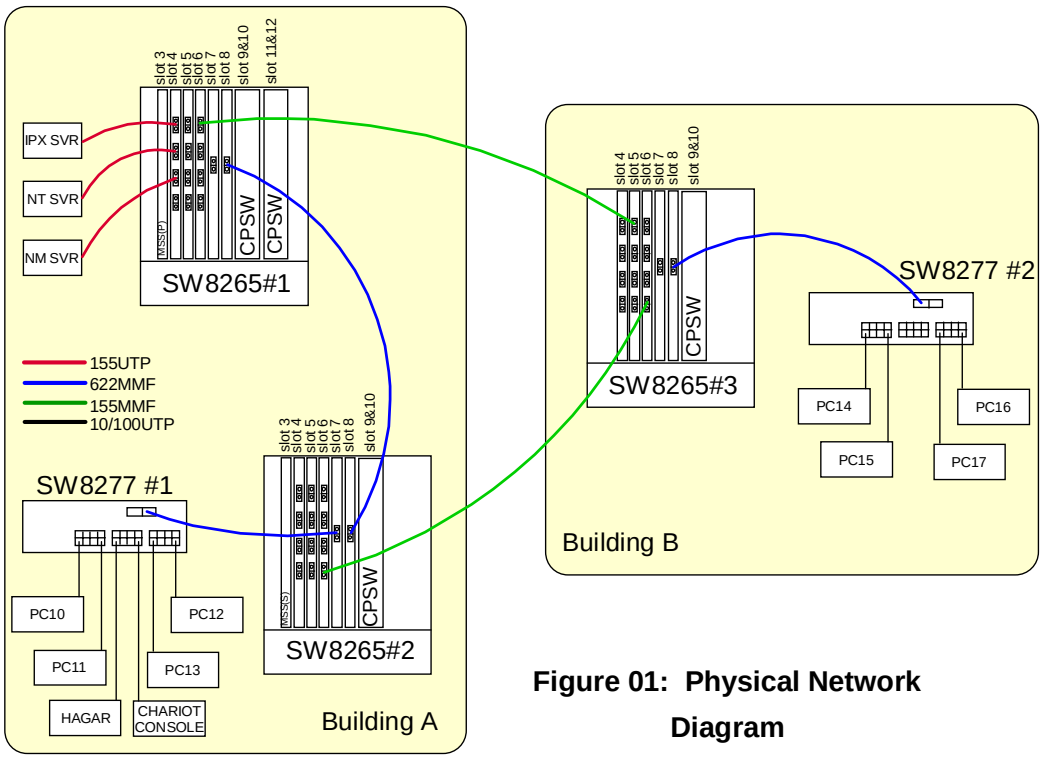


Figure 01: Physical Network Diagram

Three servers were used, named IPX_SERVER, NT_SERVER, and NM_SERVER. Each server, an IBM Netfinity 3000, has a ForeRunner ATM adapter (PCA-200EPC), manufactured by Fore Systems, running at 155 Mbps, and connected to the switch SW8265#1, using UTP category 5 cables.

The server IPX_SERVER is running Novell NetWare release 5.0, Fore device driver release 5.0.2, and Novell ATM LAN Emulation Driver ATMDRV03. The only function performed by the server is of an IPX file server.

The server NT_SERVER is running Windows NT 4.0, with service pack 3, and Fore device driver release 5.0.2. The only function performed by the server is of a NetBios file server.

The server NM_SERVER is running Windows NT 4.0, with service pack 3, Fore device driver release 5.0.2, Nways Workgroup Manager for Windows NT version 1.1.3, and Nways RouteVision Workgroup Manager release 3.2. The only function performed by the server is of a Network Management server.

The purpose of having 3 servers is to verify connectivity , running the three protocols , IPX, TCP/IP, and NetBios. The first two protocols are routeble and the last is bridgeable.

Each building contains one Ethernet switch, IBM 8277 Nways Ethernet RouteSwitch, running NRSP Release 3.2 (**N**ways 8277 **R**outeSwitch **S**oftware **P**rogram), and functioning as an edge device. SW8277#1 is located in building A, and SW8277# 2 is located in building B. Both switches have the same hardware configuration, connecting to the ATM switch through an ATM Uplink, 155Mbps, MMF connection.

There is no prevision for Ethernet or ATM switch redundancy, consequently if any of the switches fail, the end-users connected to the failing switch, will loose network connectivity.

Eight workstations running Windows/NT Workstation 4.0, with service pack 3 installed, are connected to the network, generating traffic (PC10 through PC17). One control workstation named Hagar was used as a TFTP server, in order to update code on the various switches, and as a Terminal Emulator, in order to configure the network.

The TFTP server software is part of the TCP/IP suite Chameleon, a NetManage product, that worked very well without any glitches, and the Terminal Emulator is part of the communication suite PROCOMM, which also worked very well.

The workstation running the Chariot control tool is named Chariot Console (see Figure 01), and it runs Windows NT 4.0, with service pack 3 installed.

Detailed Design

The network contains three IP subnets:

SNET1 - 10.**10**.1.xx, mask 255.255.255.0
SNET2 - 10.**20**.1.xx, mask 255.255.255.0
SNET3 - 10.**30**.1.xx, mask 255.255.255.0

Subnet SNET1 contains all the network management type of devices, such as the CPSW, the NM_SERVER server, the ethernet switches, etc. Subnet SNET2 and SNET3 contains all the end-user workstations.

By design, the routing between the subnets is performed by 8210 MSS. There is no routing enabled in the 8277s, which means that traffic destined to the same network is switched by the 8277, but traffic destined to other subnets is required to travel to 8210 MSS in order to be routed.

Using the same physical network, we designed two logical networks, called Configuration A and Configuration B. Configuration A is a basic configuration, and Configuration B implements the concept of one arm router, and uses MSS Super ELAN feature.

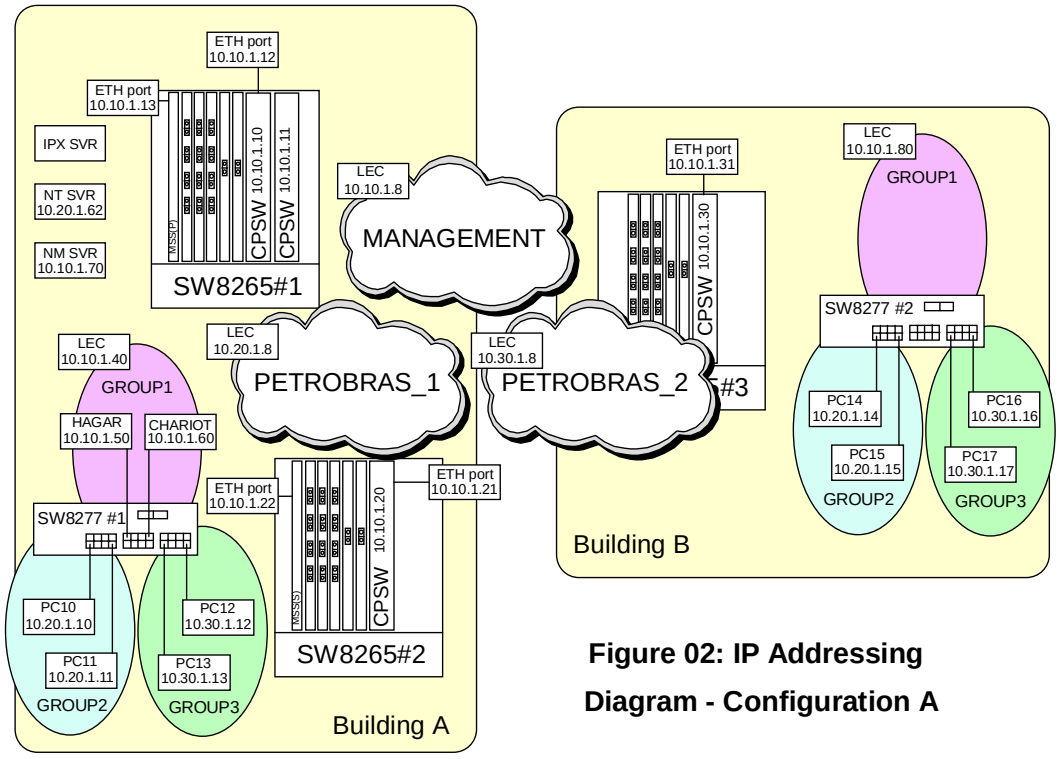


Figure 02: IP Addressing Diagram - Configuration A

Configuration A

Configuration A contains three Groups, or broadcast domains, configured in the 8277s: GROUP1, GROUP2, and GROUP3. Each Group contains one subnet, as the following: GROUP1 contains SNET1, GROUP2 contains SNET2, and GROUP3 contains SNET3. Figure 02 shows the IP addresses assigned to the various network resources.

The reason to have more than one Group in the 8277, is for dividing the broadcast domain into smaller domains. Have in mind that broadcasts are contained inside a Group.

This network design makes the workstation connections to be 8277 port dependent, which means that all the workstations that belong to SNET1, in Building A, have to be plugged to the slot 3, ports 1 through 8, in the switch SW8277#1. If flexibility or port independence is a major factor, this might not be the right network design for you.

To solve the port dependency problem, two approaches can be taken, either using the concept of Port Mobility, feature of the 8277, or using the concept of one arm router and

Super ELANs, in MSS. The latest is the approach used and explained later in this document.

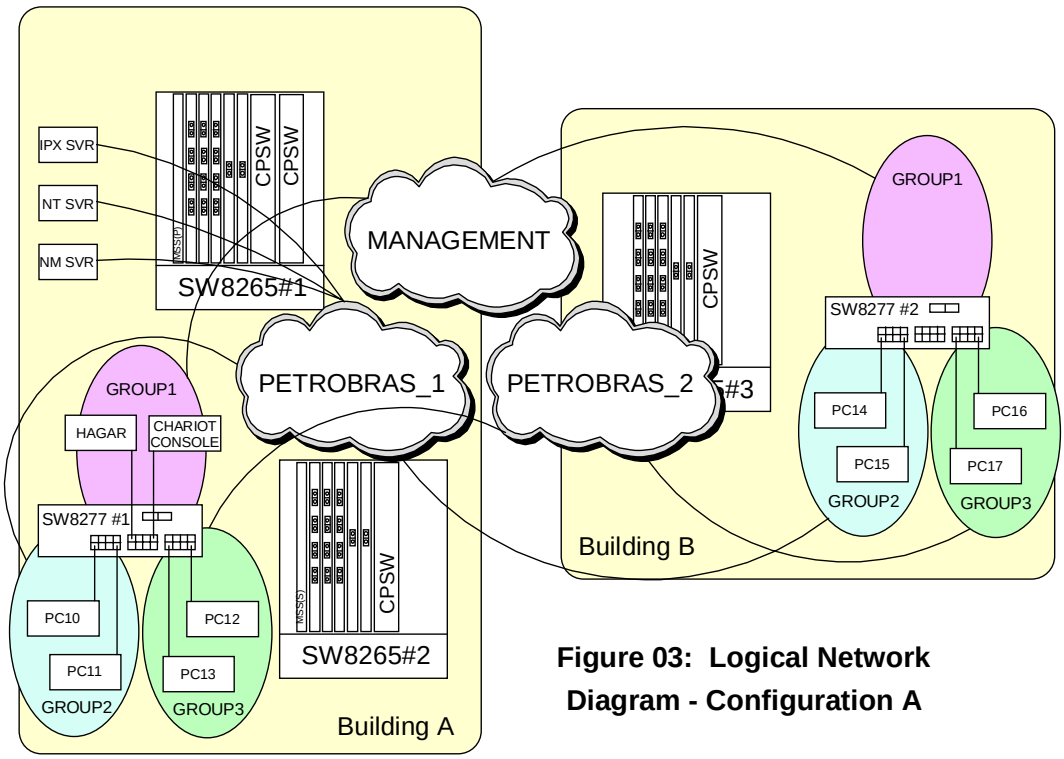


Figure 03: Logical Network Diagram - Configuration A

The backbone of the network is ATM, and the attached workstations are ethernet. The technology that allows existent ethernet or token-ring applications to use the ATM services, is called LAN emulation. Three ethernet ELANs have been configured: PETROBRAS_1, PETROBRAS_2, and MANAGEMENT.

Hosts within separate ELANs are unable to communicate with each other, unless an external device enables cross-ELAN communication. In this test network, the device enabling communications between emulated LANs is MSS.

Each ELAN is created by the existence of a LES/BUS server, running in MSS. When configuring the 8210 MSS, three LES/BUS servers were created, and each of the 8277 Groups were configured to join one of those three servers. GROUP1 joins the MANAGEMENT ELAN, GROUP2 joins the PETROBRAS_1 ELAN, and GROUP3 joins the PETROBRAS_2 ELAN (see Figure 03).

Each ELAN is configured with one LEC (LAN Emulation Client), and each LEC is configured with an IP address (see Figure 02). When an IP address is configured in a LEC, routing is activated for that particular interface, and that is how routing between different subnets and ELANs is accomplished in this network. Those IP addresses are considered the default gateway for the resources in those subnets. In Figure 02, for instance, the workstation PC10 can communicate with the workstation PC12, only through routing performed by the 8210 MSS.

This design approach makes all the workstations that belong to GROUP1, to automatically become a resource of the MANAGEMENT emulated LAN, and all the workstations that belong to GROUP2, to automatically become a resource of the PETROBRAS_1 ELAN, and all the workstations that belong to GROUP3, to automatically become a resource of the PETROBRAS_2 emulated LAN.

Configuration B

Only one Group (*GROUP1*), or broadcast domain, was configured in the 8277s. The Group contains resources that belong to the three different subnets (*SNET1*, *SNET2*, and *SNET3*), and consequently the workstation connections are not port dependent.

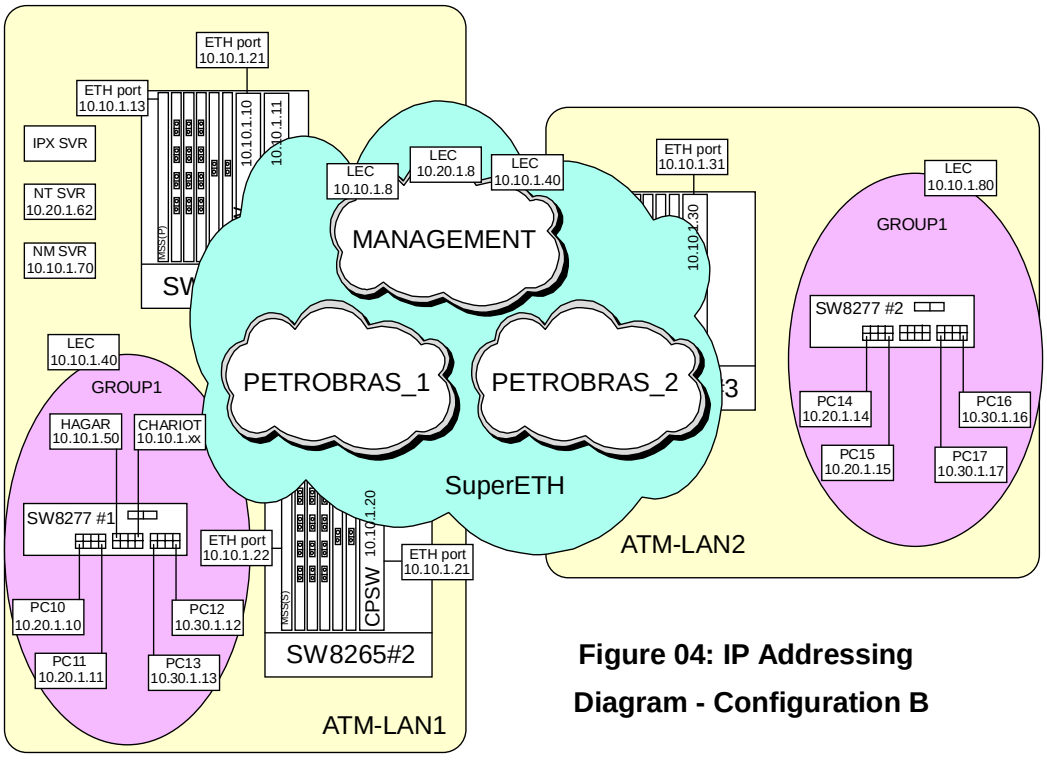


Figure 04: IP Addressing Diagram - Configuration B

The same three ELANs (PETROBRAS_1, PETROBRAS_2, and MANAGEMENT) existent in Configuration A, are configured in the MSS Server Module. GROUP1 in the Ethernet switch SW8277#1, plus the three servers, join the ELAN PETROBRAS_1, and GROUP1 in the Ethernet switch SW8277#2 joins ELAN PETROBRAS_2 (see Figure 05).

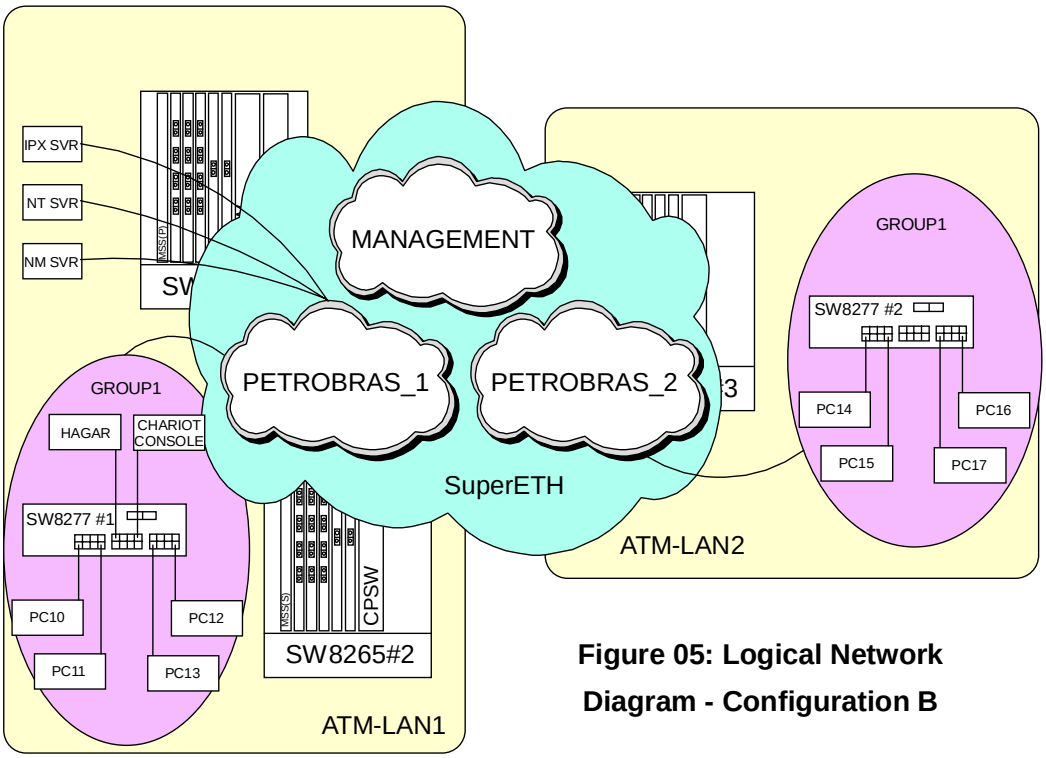


Figure 05: Logical Network Diagram - Configuration B

The MANAGEMENT ELAN is the only emulated LAN configured with three LECs (LAN Emulation Client). Each LEC has its own IP address (see Figure 04).

Using the Super ELAN concept, an MSS feature, all the emulated LANs are grouped together, forming a "super ELAN" called SuperETH. Traffic between the ELANs are bridged, and that's how the resources from different ELANs can communicate.

In order to keep the broadcast traffic down within the Super ELAN, BBCM (Broadcast Manager) can be activated in MSS.

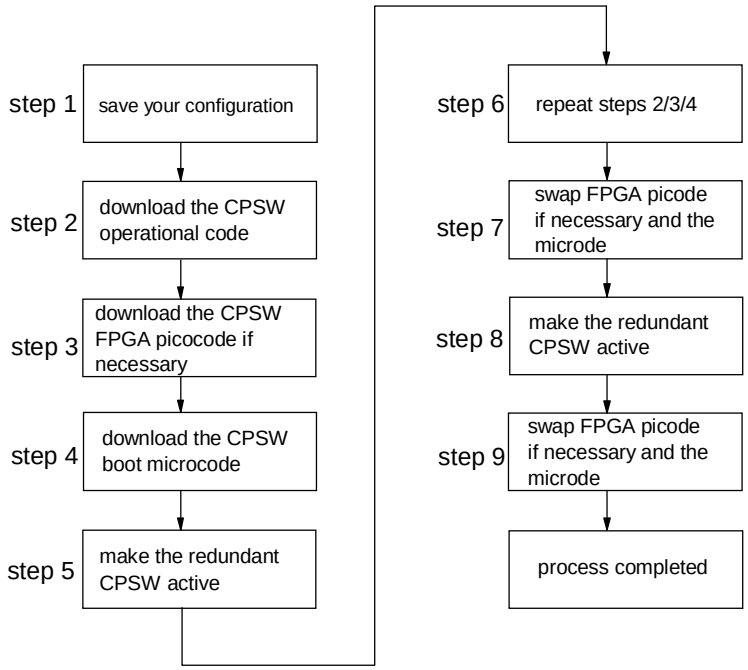
Chapter 2: Updating the 8265 CPSW Operational Code

This Chapter describes the steps performed to update the 8265 CPSW (FC6501) microcode. A number of screens have been captured during this process, and are included.

There are two different operational codes available, IISP and PNNI. As the name implies, the PNNI code allows for PNNI support, as opposed to IISP which it doesn't have. IISP can be downloaded from the IBM Web Site, but PNNI requires a USERID and PW, which is shipped with the 8265.

The CPSW came pre-loaded with operational code version 4.0, and it has been updated to 4.0.1 PNNI, which was the latest version available when the network was

Figure 06: Dual CPSW Microcode Update



assembled. As off April the 26th, the operational code available from the IBM Web Site (http://www.networking.ibm.com/support/products.nfs/support/home?opendocument) is version 4.1.2.

Figure 06 shows the steps used to update the operational code of the ATM switch SW8265#1. Notice from the physical network diagram (Figure 01) that the switch contains two

CPSWs for redundancy purposes.

Figure 07: Single CPSW Microcode Update

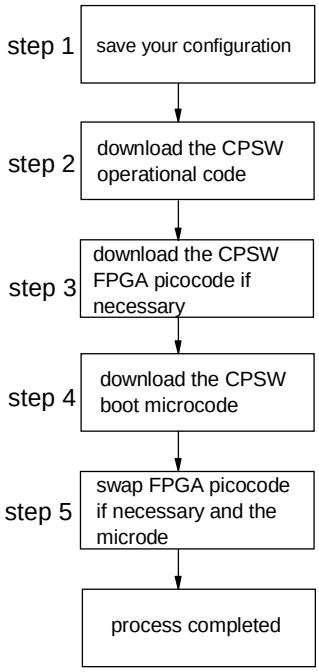
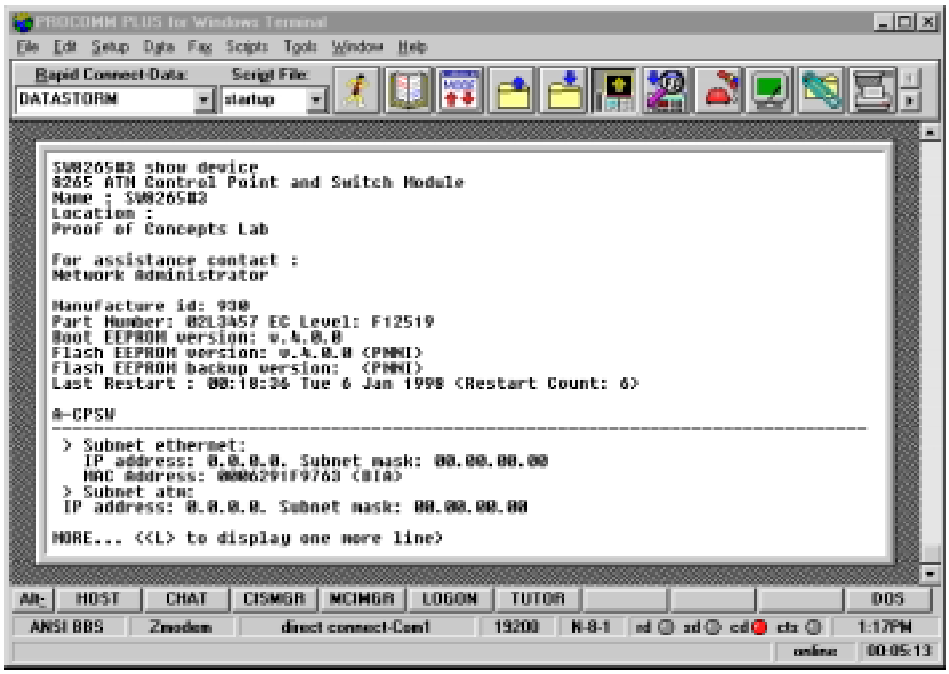


Figure 07 shows the process used to update the operational code of the other two ATM switches, SW8265#2 and SW8265#3, which contain only one CPSW.

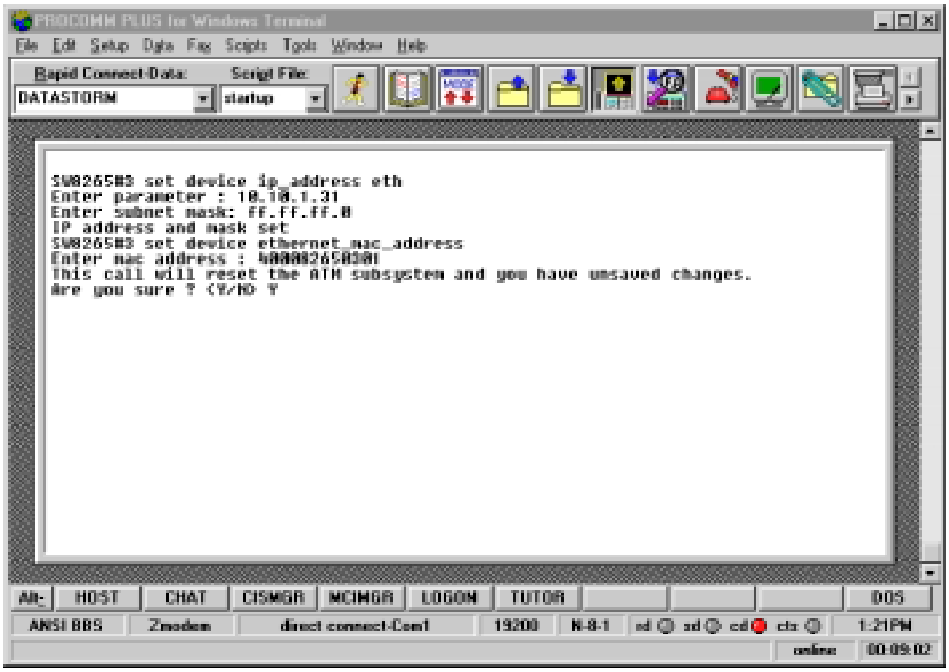
There are four different methods of performing in-band download to a 8265, Classical IP mode, Ethernet or Token-Ring LAN emulation mode, SLIP mode, and CPSW Ethernet Port mode, which is the preferred and used method, requiring very little setup.

Unfortunately, the old CPSWs don't have the Ethernet port, and in this case, any other method will work just fine. Have in mind that in-band downloads are always much faster than the out-band downloads.

In order to use the 8265 Ethernet interface, it is required to configure it. The interface is not usually configured on a new 8265. The following figure shows the use of the command *SHOW DEVICE* to verify whether the Ethernet subnet is configured and active. Notice that the interface doesn't have an IP address configured.



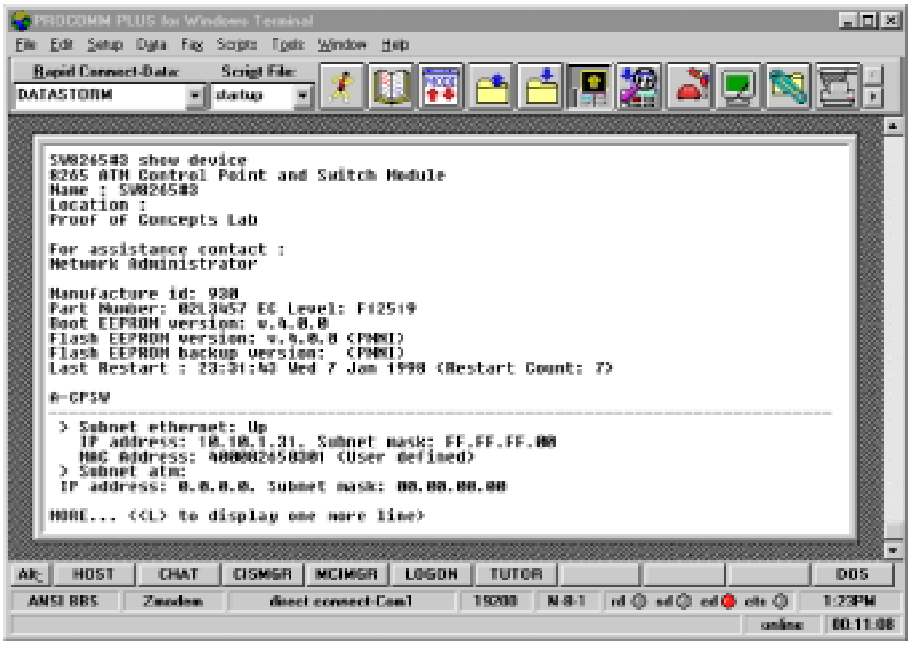
The following figure shows the commands used to configure an IP and a MAC address on the Ethernet interface. After the first command has been entered, *SET DEVICE IP_ADDRESS ETH xx.xx.xx.xx*, the 8265 prompts for the subnet mask, which must be entered in hexadecimal form.



An LAA (**L**ocal **A**dministered **A**ddress) MAC address can also be configured by using the command *SET DEVICE ETHERNET_MAC_ADDRESS xxxxxxxxxxxx* A good reason to use the LAA is to make easier to identify frames on a trace, and as my own rule, everywhere where I can, I use it.

Notice that the Ethernet interface has been given the IP address of 10.**10**.1.31 and mask 255.255.255.0. It means that the TFTP server has to be in the same subnet 10, otherwise the two resources can't communicate.

The command *SHOW DEVICE* now shows that the Ethernet interface does have an IP and a MAC address, and is active.



Now that the Ethernet interface is active, we can connect the TFTP server to the switch, and start to download code.

Figure 08 shows the connections used to update the 8265 microcode. The TFTP server connects to the RJ-45 Ethernet port on the 8265, and the workstation running the Terminal Emulator program connects to the RS232 Console port.

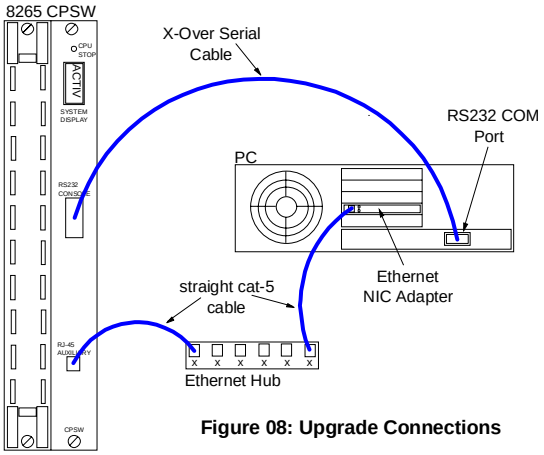
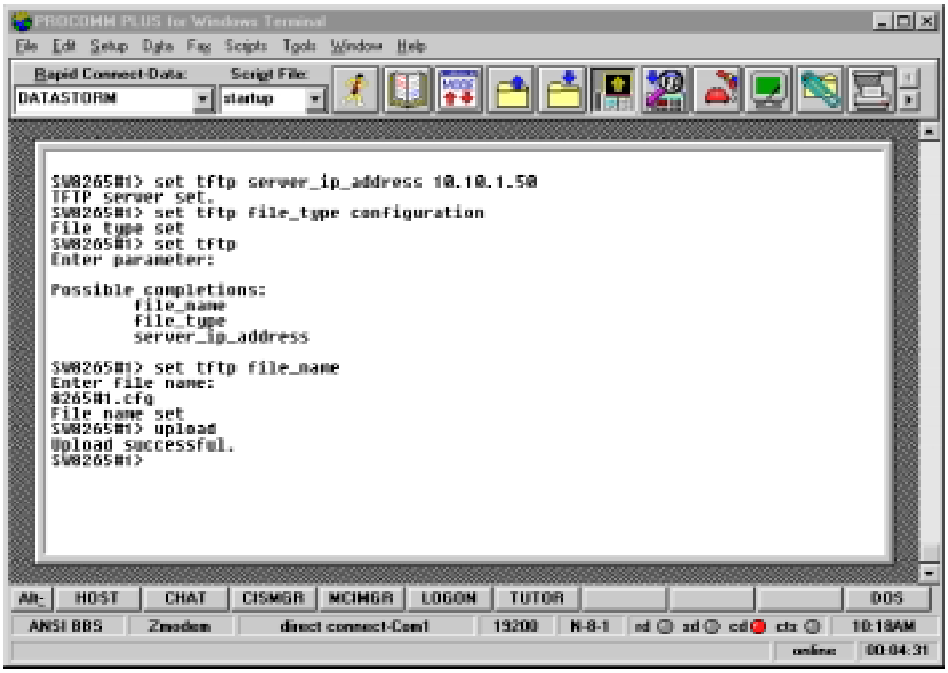


Figure 08: Upgrade Connections

Saving the Current Switch Configuration Before the Update

It is always a good practice to save the current configuration before doing any update changes. Three variables must be set on the switch, using the *SET TFTP* command: the TFTP server IP address (*10.10.1.50*), the file type (*configuration*), and the file name (*8265#1.cfg*), which is the name that the configuration file will have on the TFTP server.

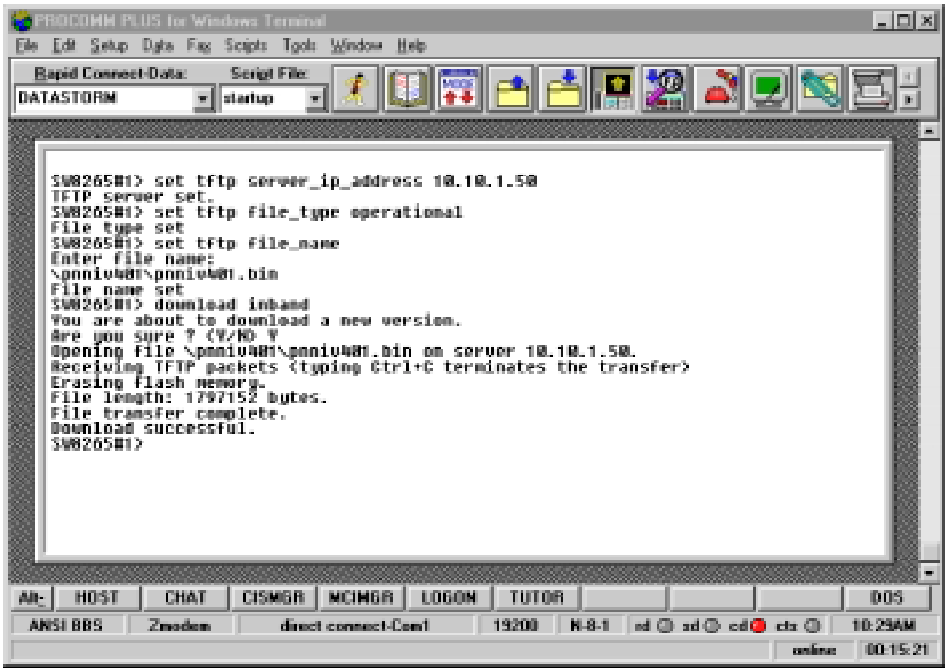
After setting up the parameters, the command *UPLOAD* transfers the current 8265 configuration from the CPSW to the TFTP server.



Download In-band the CPSW Operational Code

Three variables must be set on the switch, using the *SET TFTP* command: the TFTP server IP address (*10.10.1.50*), the file type (*operational*), and the file name (*\pnniv401\pnniv401.bin*), which is the full name of the operational microcode file, on the TFTP server.

After setting up the parameters, start the download in-band process, using the command *DOWNLOAD INBAND,* which will transfer the 8265 operational microcode file from the TFTP server to the CPSW.

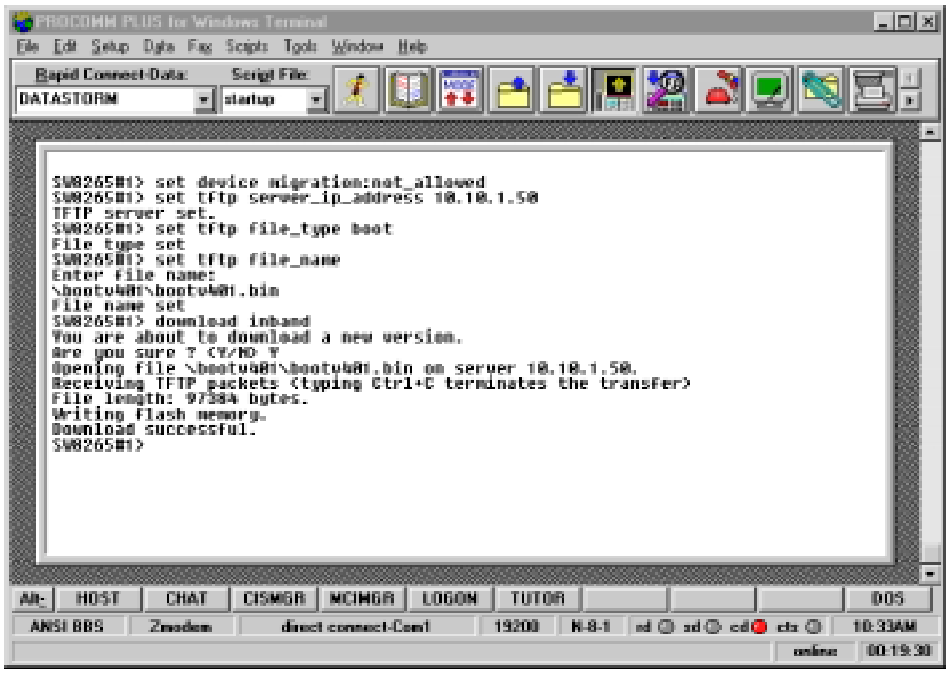


Download In-band the CPSW Boot Microcode

Three variables must be set on the switch, using the *SET TFTP* command: the TFTP server IP address (*10.10.1.50*), the file type (*boot*), and the file name (*\bootv401\bootv401.bin*), which is the full name of the boot microcode file, on the TFTP server.

After setting up the parameters, start the download in-band process, using the command *DOWNLOAD INBAND,* which will transfer the 8265 boot microcode file from the TFTP server to the CPSW.

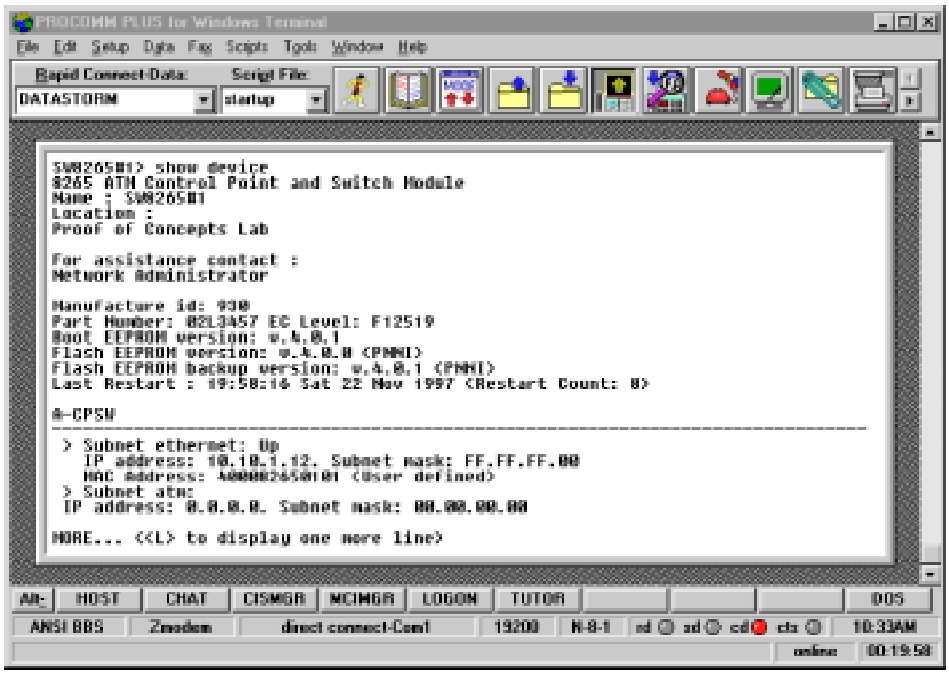
Notice that the first command, *SET DEVICE MIGRATION NOT_ALLOWED* is required.



Verifying the Updates

The command *SHOW DEVICE* is used to display a number of parameters set in the 8265 CPSW, including the microcode level. Notice that the Flash EEPROM version has two instances, one is the current version V.4.0.0 (PNNI), and the other is the Flash EEPROM backup version V.4.0.1 (PNNI), which is the updated version, just installed.

In order to have the update code taking effect, a microcode swap and a switch reset must be performed.



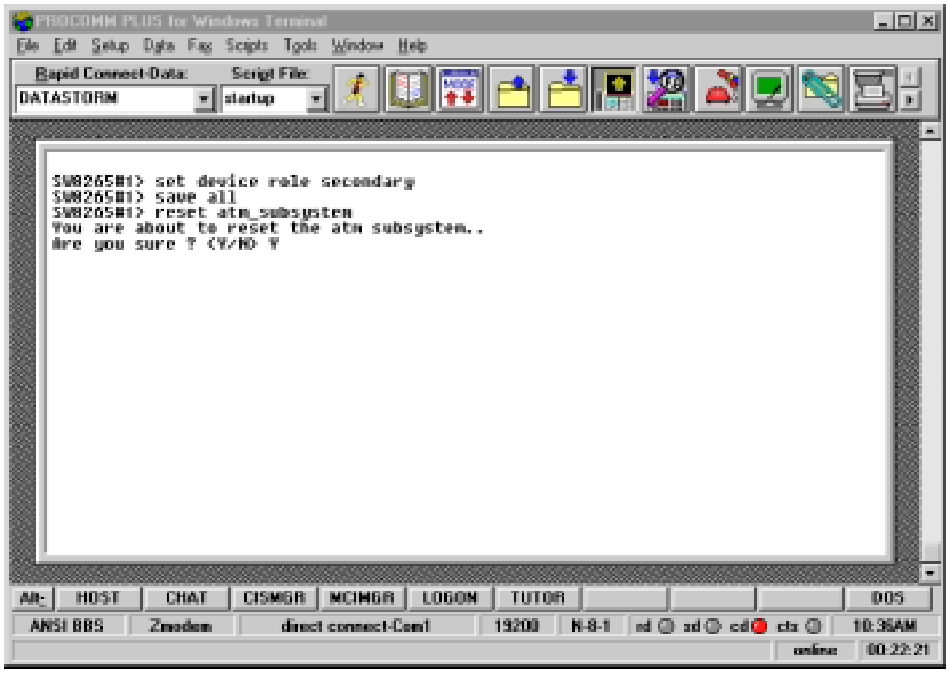
Make the Backup CPSW Active

Before a microcode swap can be performed, the microcode on the redundant CPSW has to be updated. That is the reason we are making the secondary CPSW the primary, so the microcode can be updated.

These are the required commands to make the backup CPSW active:

- *SET DEVICE ROLE SECONDARY*
- *SAVE ALL*
- *RESET ATM_SUBSYSTEM*
- move the Terminal Emulator cable to the new active CPSW
- log back in as administrator

At this point, repeat the steps 2, 3, and 4, shown in Figure 04, updating the code on the 2nd CPSW module.



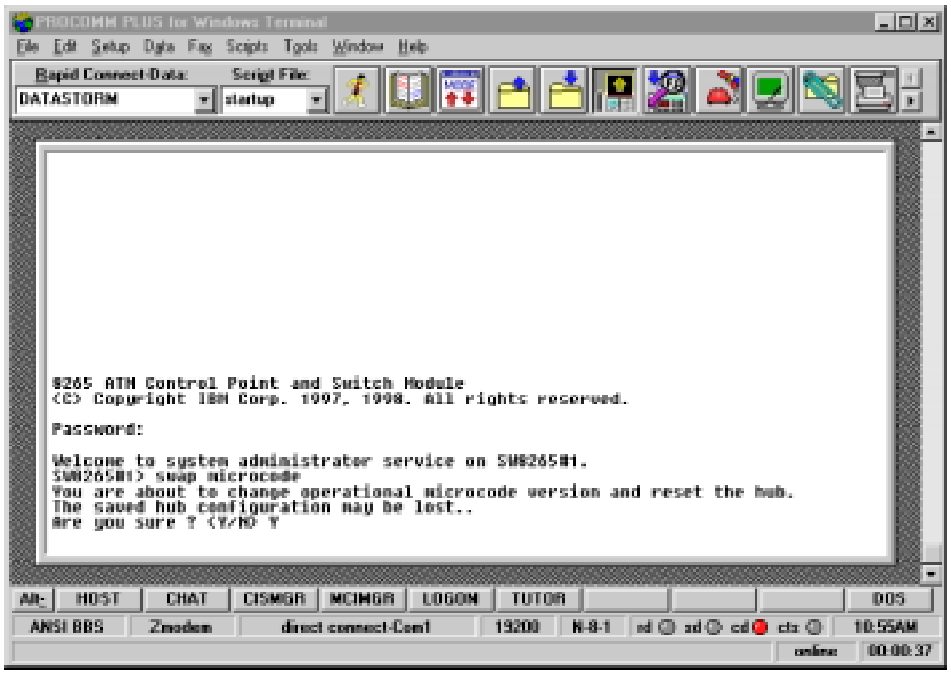
Activate the New CPSW Microcode and FPGA Pico code

To activate the microcode and FPGA pico code, it is necessary to swap the current pre-installed code with the new updated code. First, the FPGA pico code is swapped, and then the operational microcode.

The CPSW FPGA pico code version required by the CPSW microcode V.4.0.1 is 1D13, which is the level that came installed on the CPSWs, and consequently, we don't have to download the CPSW FPGA pico code or to swap it.

The CPSW microcode is swapped performing the following command sequence:

-SWAP MICROCODE
- log back in as administrator

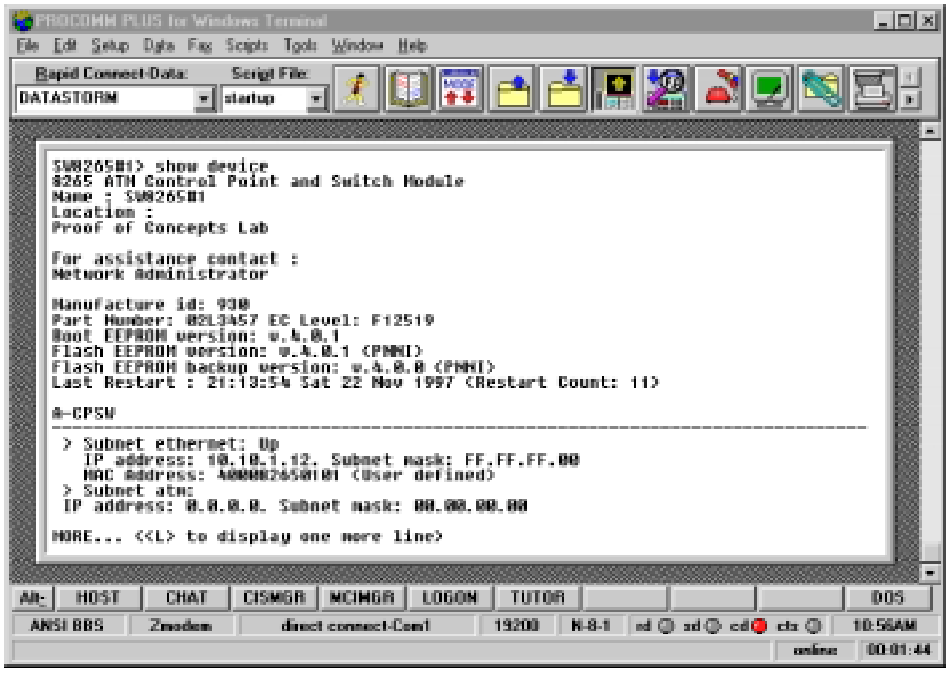


In the event of having to perform a CPSW FPGA pico code swap, these would be the commands to be executed:

- *SAVE ALL*
- *SWAP FPGA_PICOCODE 9* (assuming CPSW in slot 9)
- log back in as administrator

Verifying the Updates

The following figure now shows that the Flash EEPROM version is V.4.0.1, and the Flash EEPROM backup version is V.4.0.0.



Chapter 3: Updating the 8265 Modules' FPGA Pico code

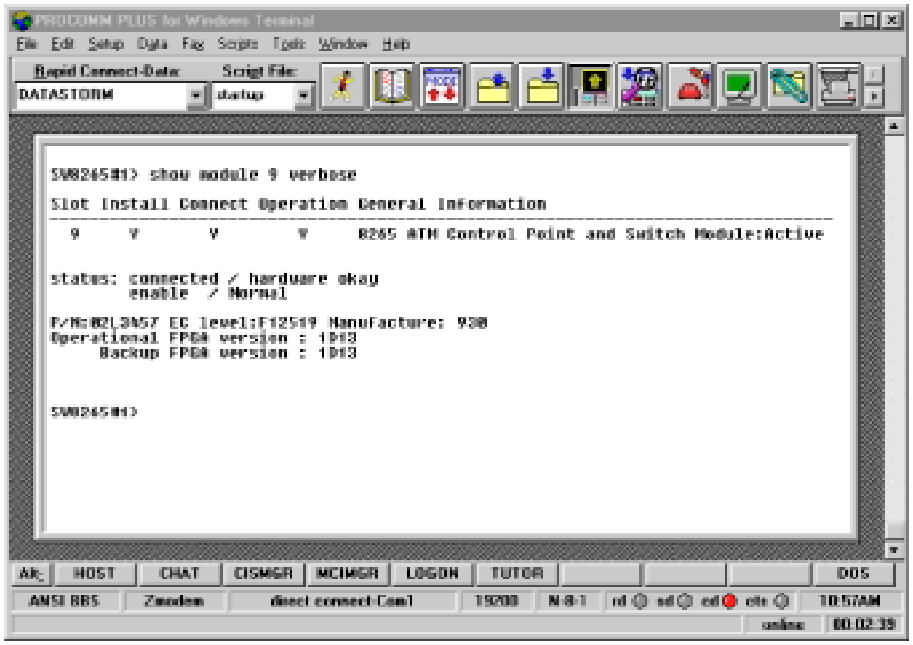
Once the boot and the operational microcode have been updated, it is time to update the modules' FPGA pico code. Each 8265 module has a different requirement, as far as version is concerned.

The following table shows the required FPGA pico code levels for the 8265 modules, when the CPSW is running microcode V 4.0.1.

Module	Feature Code	Faceplate	FPGA code
CPSW	6501	CPSW	1D13
CPSW2	6502	CPSW2	2D13
155 Mbps 4P Flex module	6543	A4-MB155	1D23 2D04 3D04
155 Mbps 4P Integrated module	6540	A4-MF155	1D23 2D04 3D04
622 Mbps 1P MMF module	6511	A1-MF622	2D04 3D04
622 Mbps 1P SMF module	6512	A1-SF622	2D04 3D04
Carrier 2.0 module	6558	A-CMU2	2D04 3D04
Carrier 2.5 module	6559 6560 6561	A-CMU2.5S A-CMU2.5A A-WAN2.5	2D14 2D14 2D14

Verifying the CPSW FPGA Pico code Version Requirements

The command *SHOW MODULE x VERBOSE* allows to verify what is the FPGA pico code level of a specified module. From the following figure, we can see that both, the operational CPSW FPGA and the backup FPGA are version 1D13.



After verifying the current running FPGA levels, we found out that it was necessary to upgrade the FPGA pico code of the 155Mbps 4P Flex module, which was running code 2D03, and the 622 Mbps 1P MMF module, which was also running code 2D03.

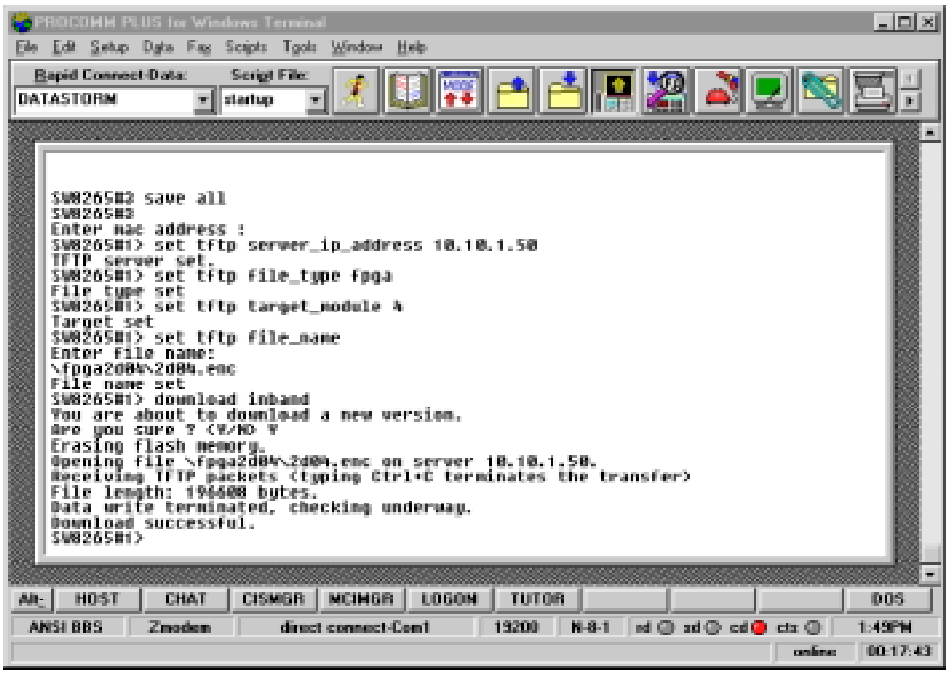
Performing the FPGA Pico code Update

All the ATM switches in this network have the same hardware configuration, with the exception of switch SW8265#1, which contains 2 CPSWs, and switch SW8265#3, which doesn't have an 8210 MSS module.

Each ATM switch contains 3 modules 155Mbps 4P, and 2 modules 622Mbps 1P that requires FPGA updates, and consequently, the process here described must be executed 15 times.

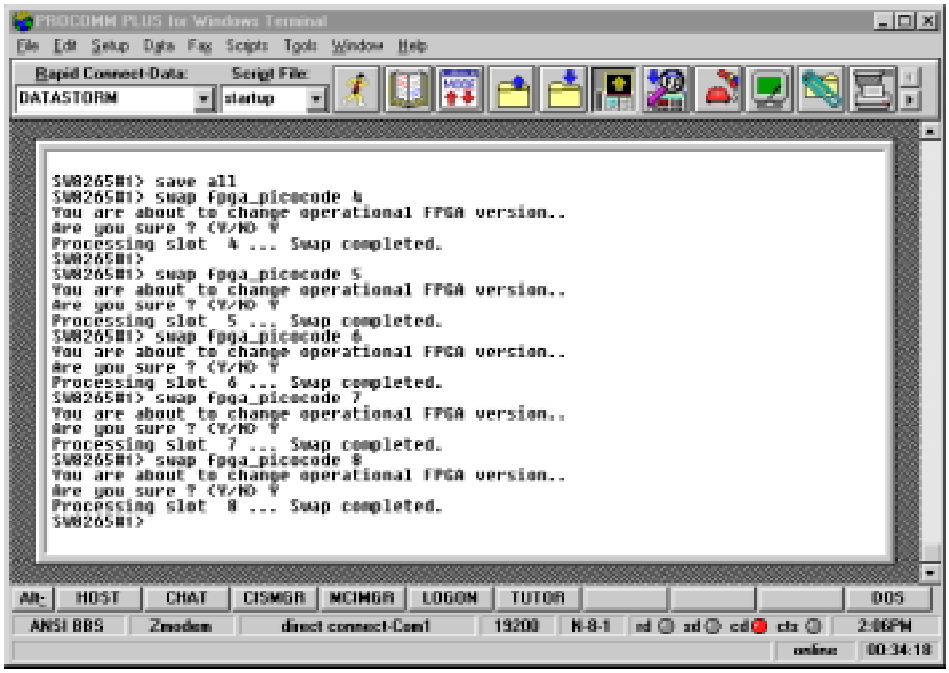
Four variables must be set on the switch, using the *SET TFTP* command: the TFTP server IP address (*10.10.1.50*), the file type (*fpga*), the file name (*\fpga2d04\2d04.enc*), which is the full name of the FPGA pico code file, on the TFTP server, and the target module, which is the slot number where the module is plugged in.

After setting up the parameters, start the download in-band process, using the command *DOWNLOAD INBAND,* which will transfer the FPGA pico code file from the TFTP server to the specified module.



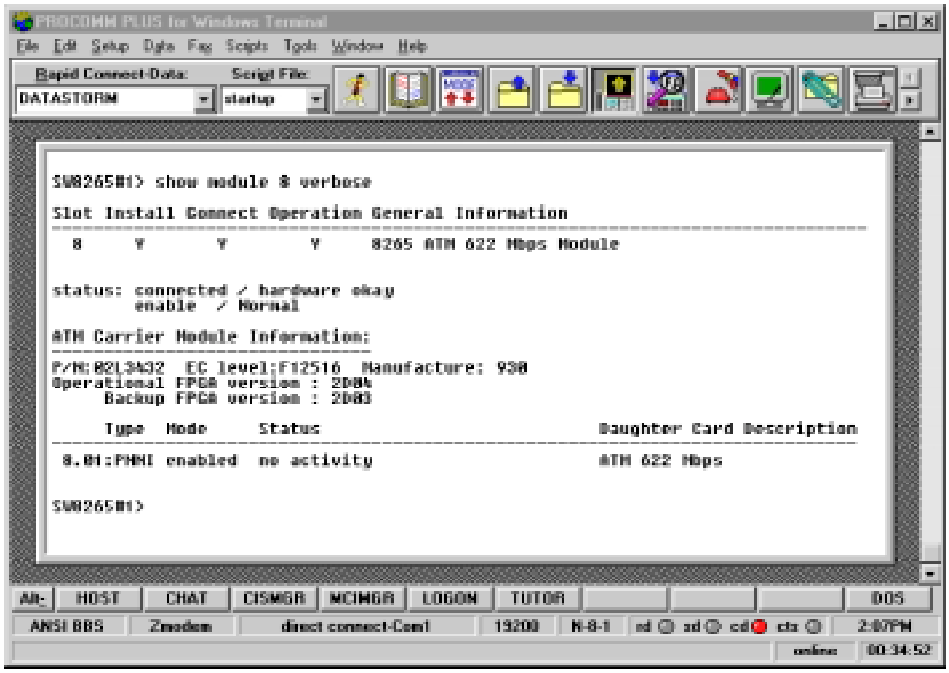
After the FGPA pico code has been updated in all the 8265 modules that require the update, the operational and the backup code now need to be swapped, using the command SWAP *FPGA-PICOCODE x*, where x is the slot number of the module which the code is to be swapped.

The SWAP command can specify one or more modules. The following figure shows the command specifying only one module at a time.



Show the Current FPGA Level for Module 8

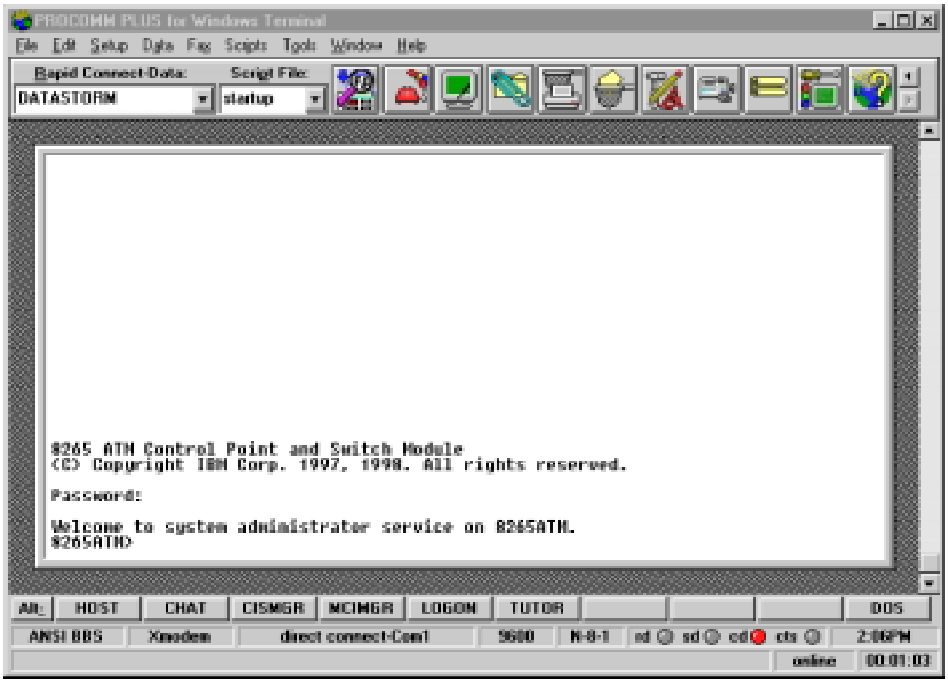
The command *SHOW MODULE x VERBOSE* is used to verify the FPGA level of a specified module. Notice that after the code has been swapped, the operational FPGA code version is now 2D04, and the backup FPGA code version is 2D03. The 8265 keeps always both versions, and at any time for any reason, the previous code can be swapped back.



Chapter 4: Configuring the 8265

Configuring an 8265 is a very simple task to perform, particularly in our network. The simulated network doesn't have any WAN connections, all the PNNI paths have the same cost and priority, and no policing or traffic shaping is configured.

The default administrator password is 8265.



Set the ATM Switch Address

When a PNNI switch is powered on for the first time, it loads a default conf iguration, which also includes a default ATM address. This ATM address has to be reconfigured, so that each switch in the network has a unique address. The default ATM address is

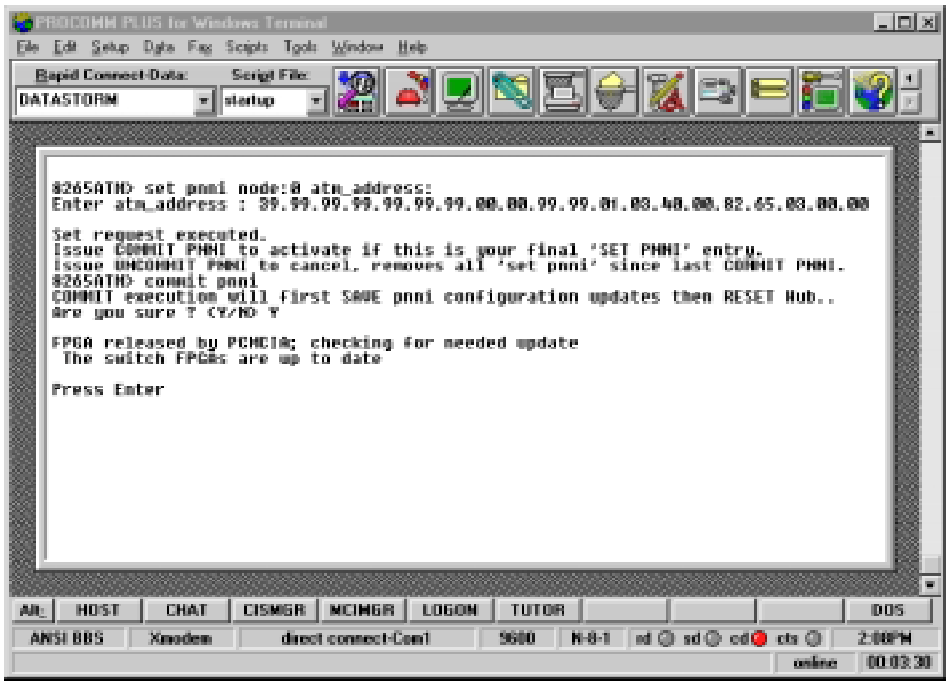
39.99.99.99.99.99.99.00.00.99.99.01.01.99.99.99.99.99.99.00

All the three ATM switches are in the same peer group, and have a level identifier of 96 bits, which means that all the three switches have the same ATM address prefix, and that its length is 96 bits, or 12 bytes.

To be able to identify eas ily the ATM switches based in their ATM address, we defined the following addresses:

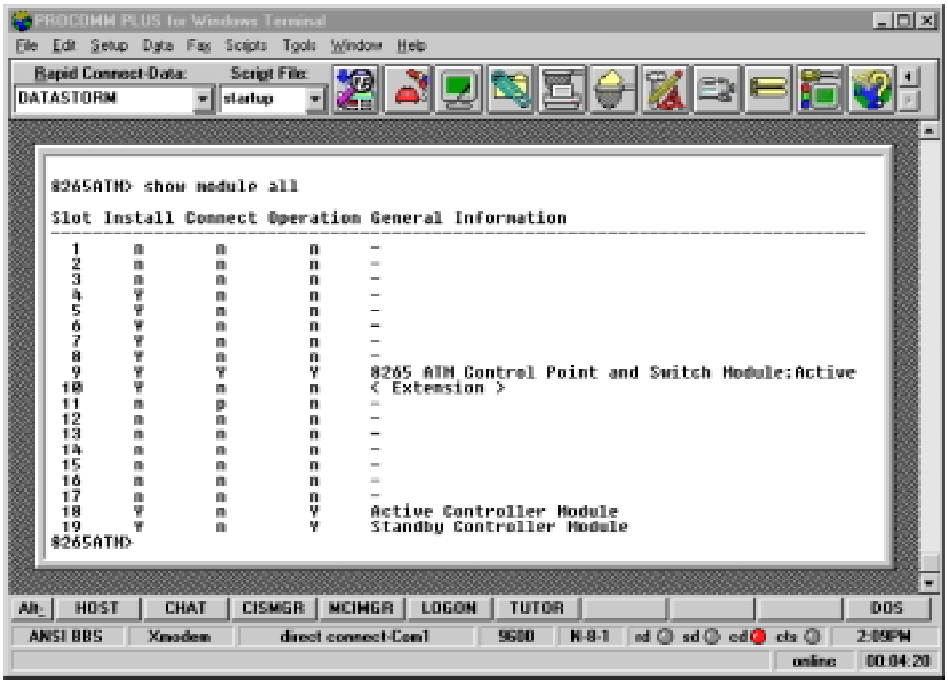
SW8265#1 > **39.99.99.99.99.99.99.00.00.99.99.01**.01.40.00.82.65.01.99.00
SW8265#2 > **39.99.99.99.99.99.99.00.00.99.99.01**.02.40.00.82.65.02.99.00
SW8265#3 > **39.99.99.99.99.99.99.00.00.99.99.01**.03.40.00.82.65.03.99.00

The following figure shows switch SW8265#3 being configured. The command SET PNNI NODE:0 ATM_ADDRESS set up the address, and to activate it, the command COMMIT PNNI was used.



Verifying Installed Modules

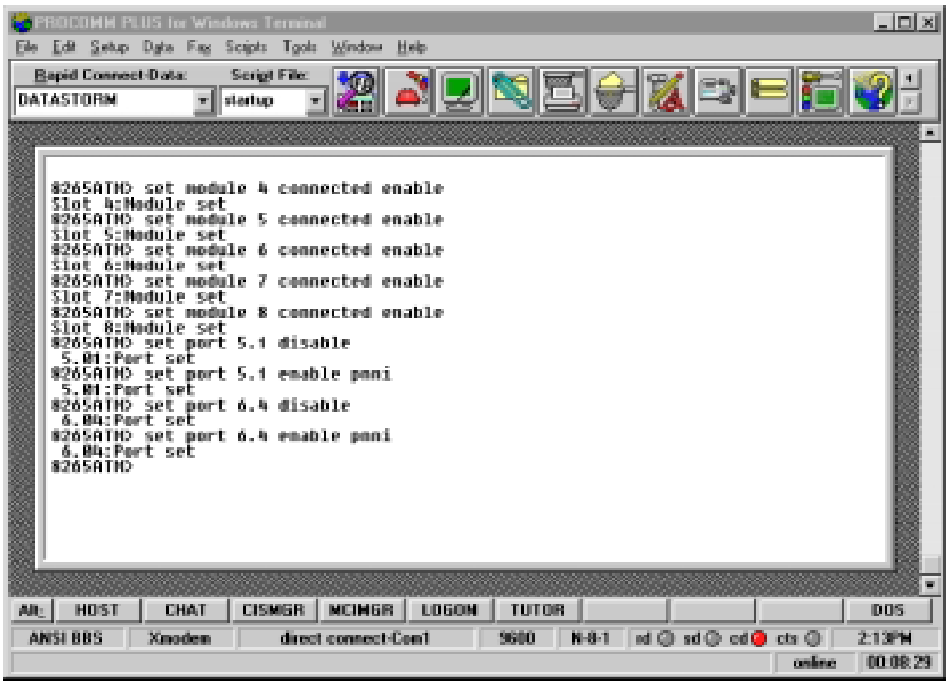
The only modules that don't require any command to be "connected" to the switch's back plane are the CPSW and the Controller modules. As we can see from the following figure, only those two modules are recognized by the switch, even though there are 5 more modules plugged in, plus MSS.



Connecting the Modules to the Backplane

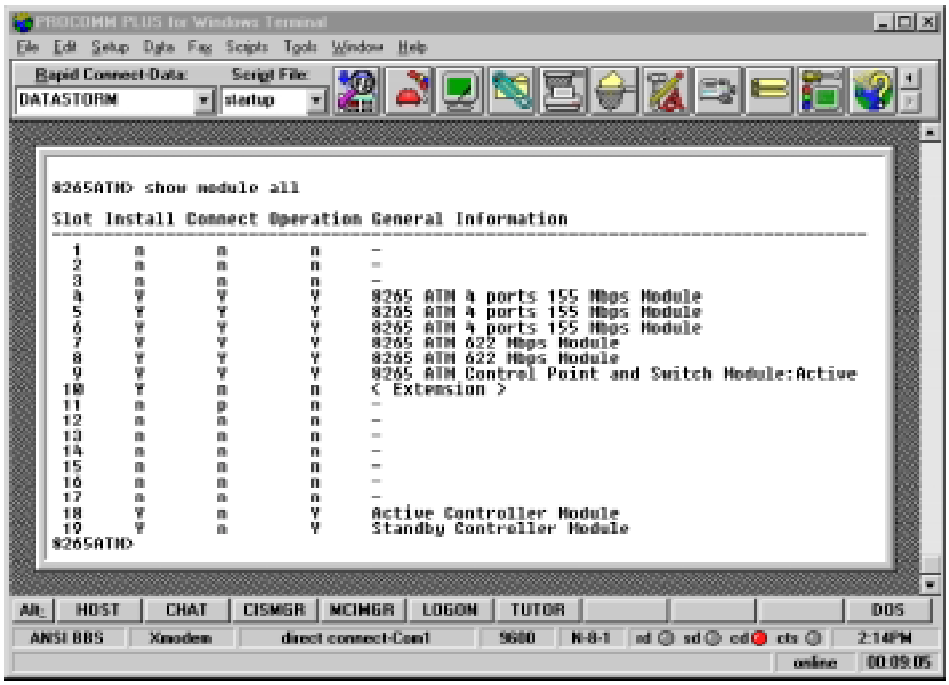
The command *SET MODULE x CONNECT ENABLE* is used to connect modules to the back plane. If the option ENABLE is entered, then all the existent ports on the module will be enabled, and the user interface will be UNI , which is the default value.

Switch SW8265#3 has only two PNNI ports, slot 5 port 1, and slot 6 port 4. Because all the ports of the switch have been enabled as UNI interface , those two ports need to be disabled, and enabled again as PNNI ports.



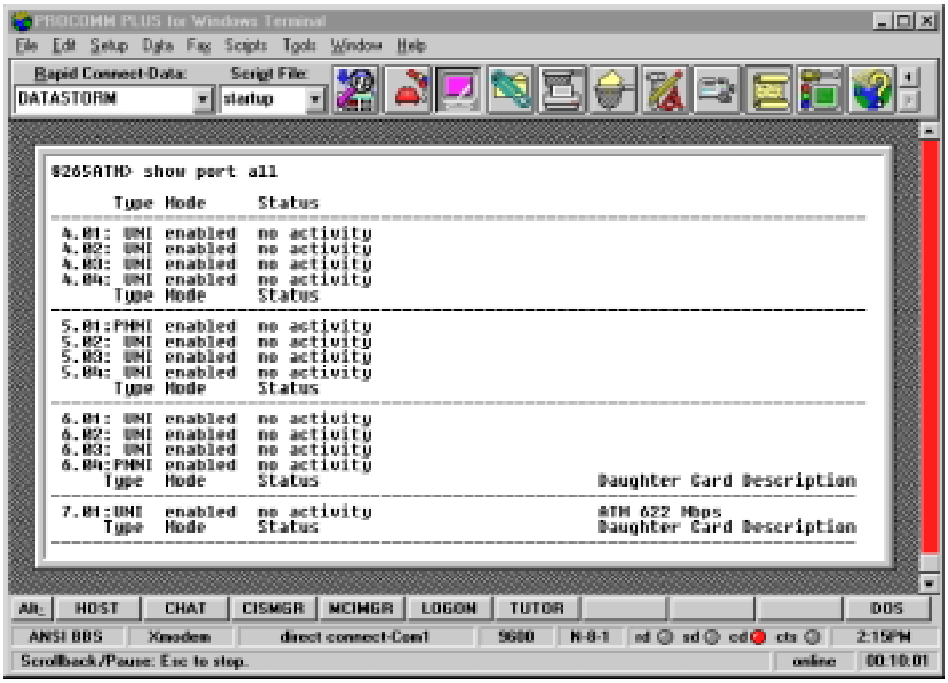
Verifying the Connected Modules

After connecting “logically” all the modules to the switch’s back plane, let’s verify whether the switch now recognizes the plugged modules, using the *SHOW MODULE ALL* command.



Verifying the Status of the Ports

The command *SHOW PORT ALL* is used to verify the status of the ports. The command displays the port number (slot.port), what type of interface the port has, whether the port is enabled, and its status.

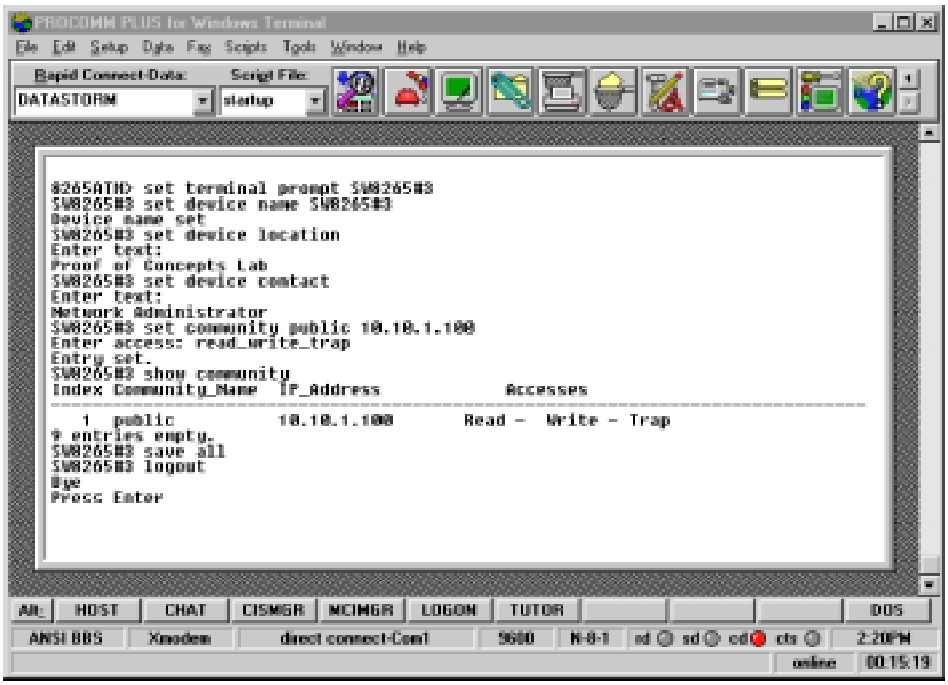


General and SNMP Parameters

There is a set of general parameters, such as device name, location, etc ., that is recommended to be configured, because they identify the device , and make the identification process easier, in the eventuality of a problem.

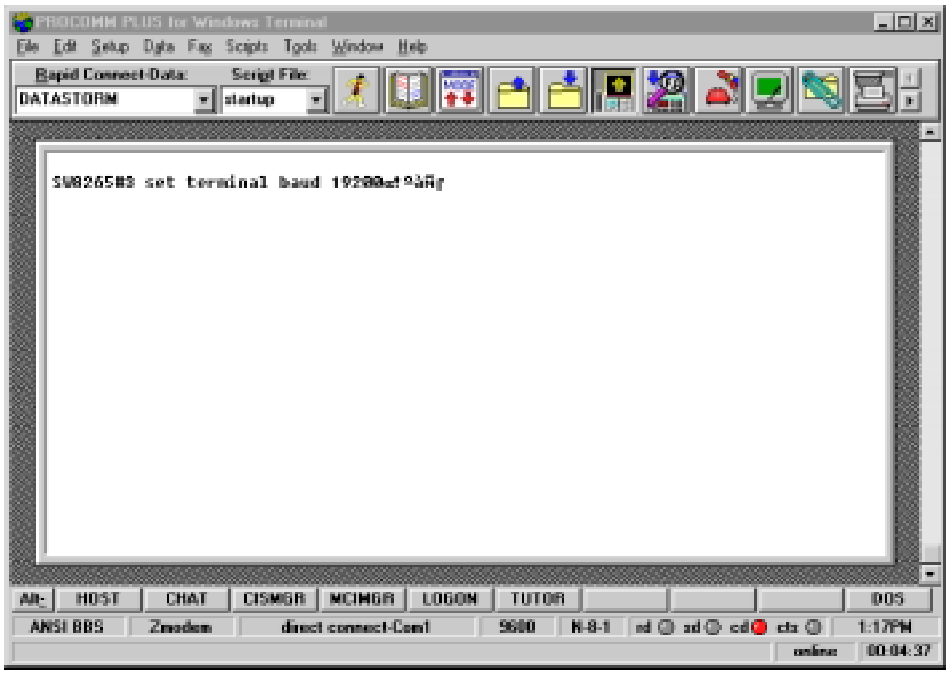
The network is to be managed by the Nways Workgroup Manager for Windows NT version 1.1.3, and it requires to defin e the SNMP parameters, such as community names, traps, etc.

The following figure shows the general and SNMP parameters been setup.



Changing the Terminal Speed to 19,200 bps

The default terminal speed on the 8265 RS232 port is 9,600 bps . Changing the speed to 19,200 bps will give a better response time when long replies are sent to the terminal emulator program.



After changing the 8265 RS232 port speed, don't forget to change also the speed on the Terminal Emulator program.

Chapter 5: 8210 MSS Server Firmware Update

There are two types of MSS Servers, the IBM 8210 Nways Multiprotocol Switched Services Server, which is the standalone product, and the IBM Multiprotocol Switched Services Server Module, which is a module on the 826x, and used in this project.

MSS came pre-loaded with firmware version 3.2.1, and it has been updated to firmware version 3.2.8, which was the latest version available when the network was assembled. As off March the 23th, the firmware available from the IBM Web Site is version 4.

The latest MSS Server information, operational code, firmware, fixes, etc., can be downloaded from the IBM WWW site (http://www.networking.ibm.com/support/products.nfs/support/home?opendocument). An USERID and PW is required, and it is normally shipped with the MSS Server CD-ROM.

The following table shows compatibility between the various versions of software and hardware:

	MSS Operational Code				
MSS Hardware	1.0	1.1	2.0	2.0.1	2.1
8210-001					
- 32 MB	FW 1.0+ CP 1.0	FW 2.0+ CP 1.1	X X	X X	X X
- 64 MB	X X	FW 3.0+ CP 1.1	FW 3.0+ CP 2.0	FW 3.0+ CP 2.0.1	FW 3.0+ CP 2.1
2-Slot Blade+					
- 32 MB	FW 1.0+ CP 1.0	FW 2.0+ CP 1.1	X X	X X	X X
- 64 MB	X X	FW 3.0+ CP 1.1	FW 3.0+ CP 2.0	FW 3.0+ CP 2.0.1	FW 3.1+ CP 2.1
1-Slot Blade					
- 64 MB	X X	X X	X X	FW 3.1+ CP 2.0.1	FW 3.1+ CP 2.1
FW = Firmware Level CP = Configuration Program X = Not Supported					

The new firmware can be loaded using TFTP server, XMODEM mode, or a Local File Copy, from the current firmware console.

Figure 09 shows the connections used to update the 8210 firmware. The TFTP server connects to the RJ-45 Ethernet port on the 8210, and the workstation running the Terminal Emulator program, connects to the RS-232 Console port.

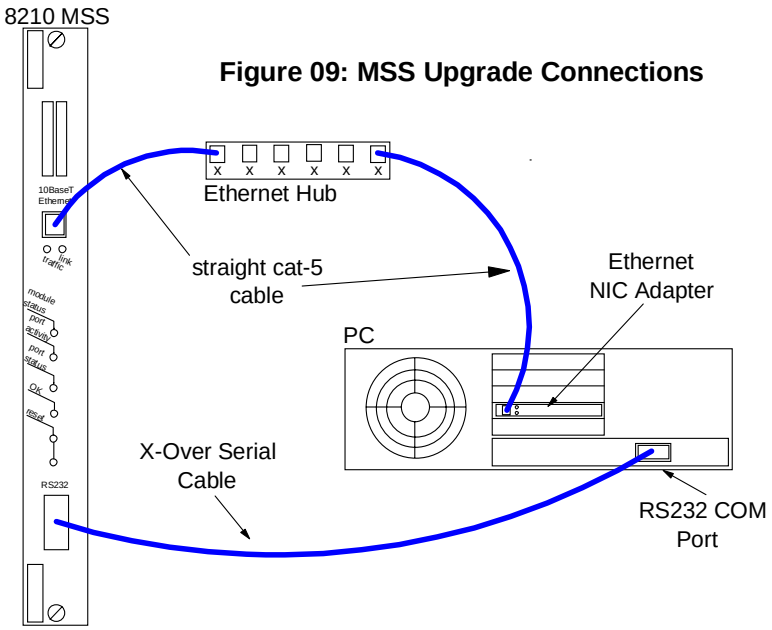
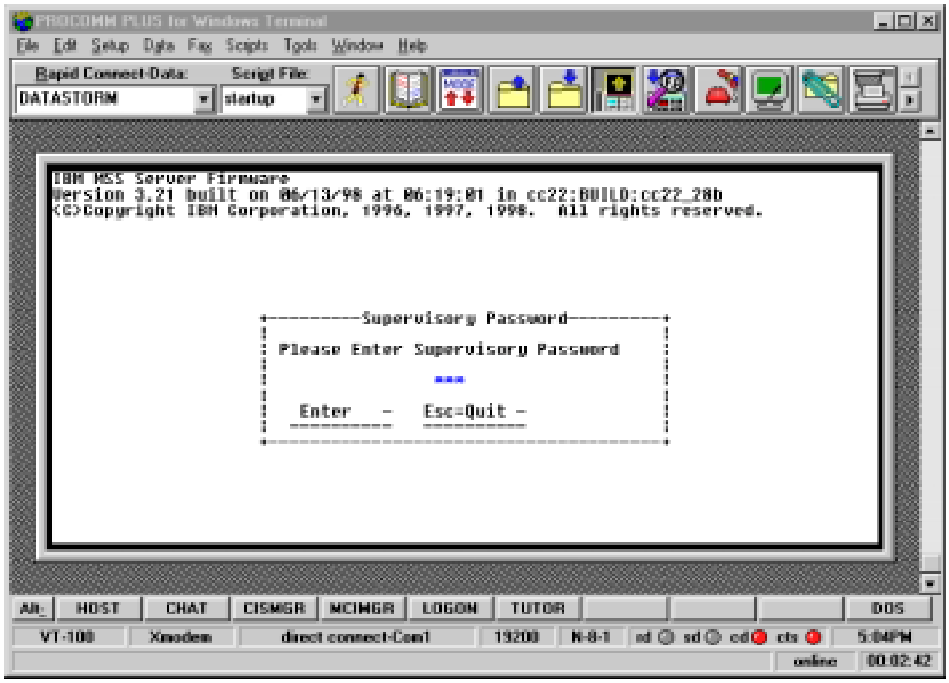


Figure 09: MSS Upgrade Connections

When using the Ethernet interface, the default IP address of the MSS module is 10.1.2.2, mask 255.255.255.0, and the default IP address of the TFTP server workstation is 10.1.2.3, mask 255.255.255.0.

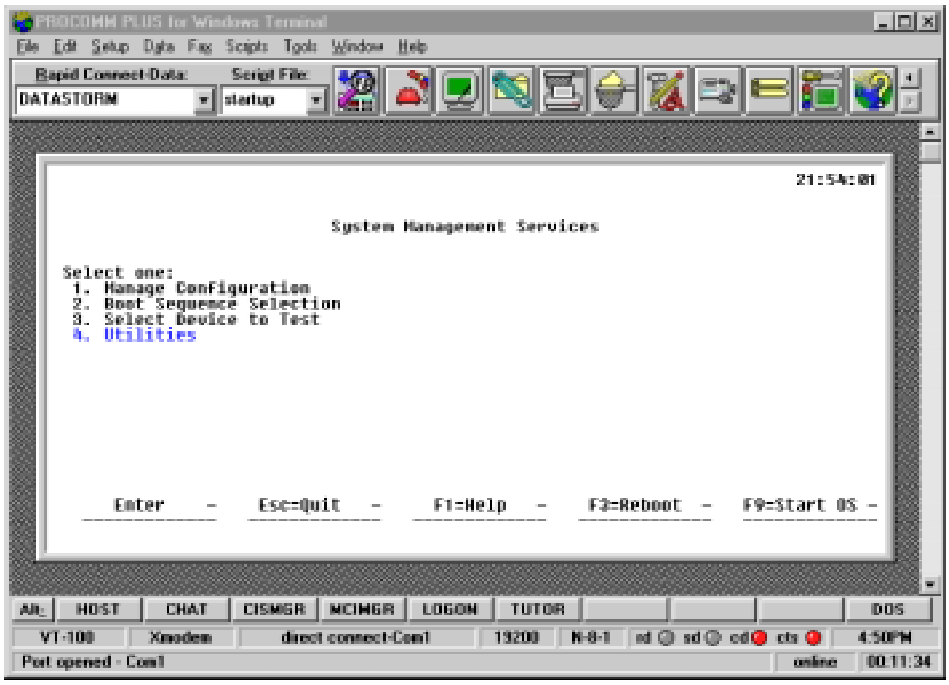
Accessing the System Management Services Menu

To access the System Management Services menu, the boot process has to be interrupted. To perform this, press and hold Ctrl-C or press F1 when prompted. The default supervisory password is MSS. Notice at the top of the window that the current level of firmware is (version 3.21).



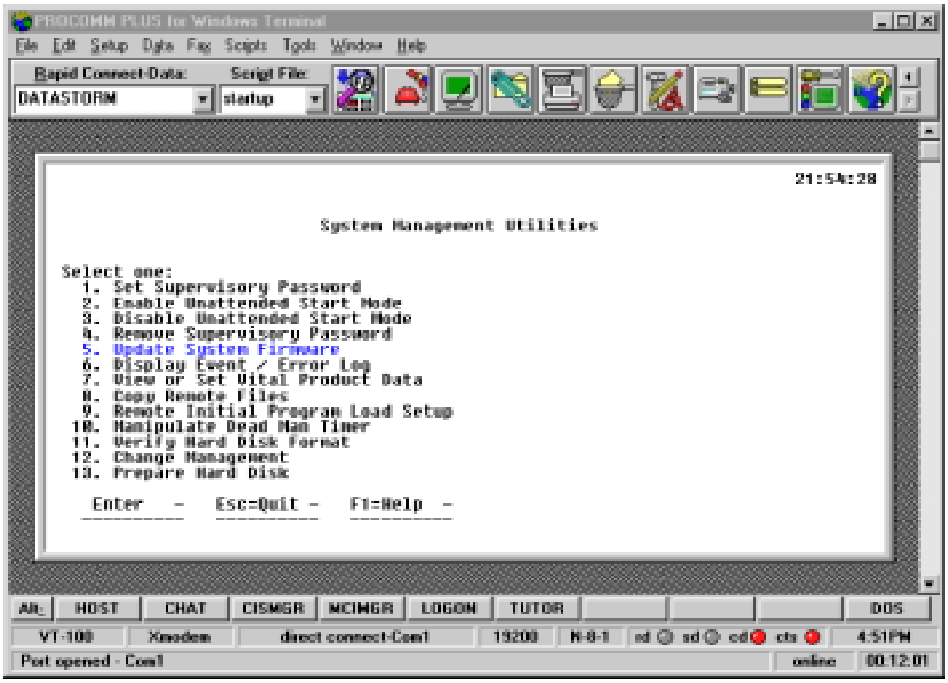
Accessing the System Management Utilities Menu

The following figure shows the options available. Select option *4. Utilities* and press the enter key.



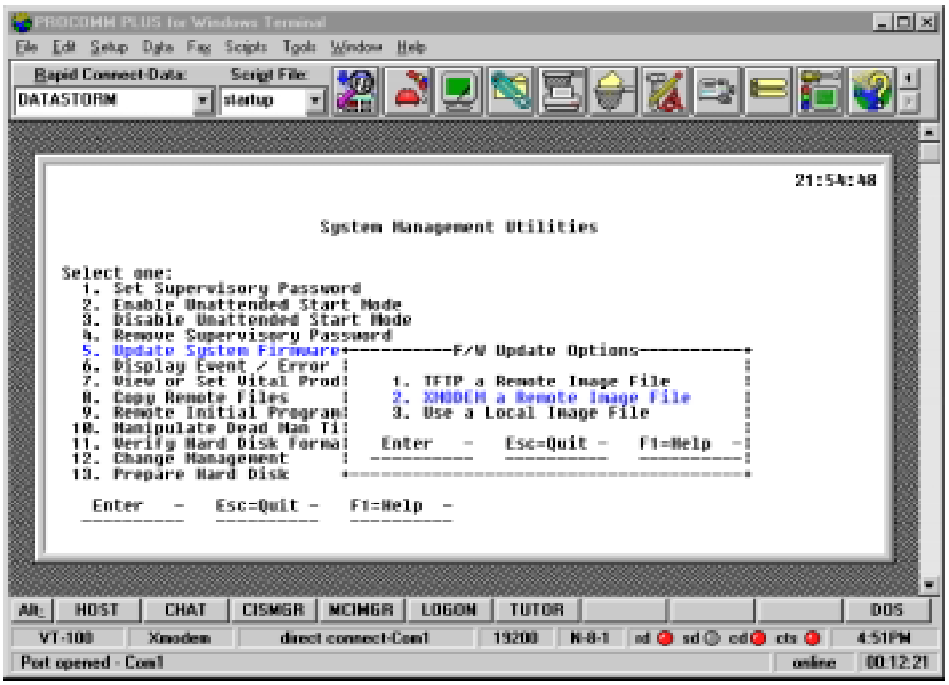
Accessing the Firmware Update Options Menu

Select option *5. Update System Firmware*, and press the enter key.



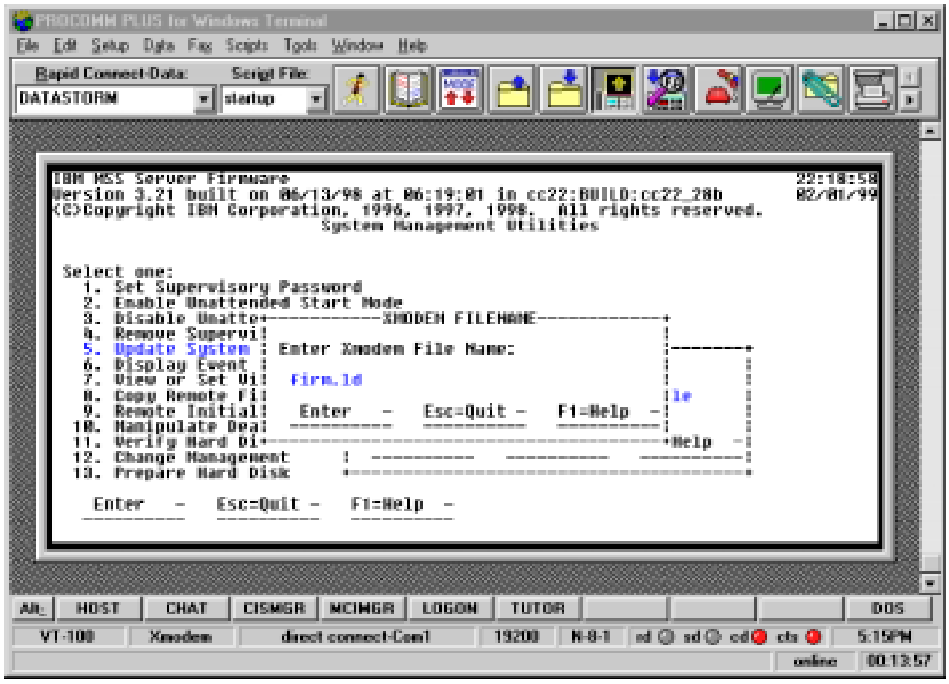
Selecting a File Transfer Method (XMODEM)

Select option *2. XMODEM a Remote Image File* and press the enter key. If a TFTP server is available, option 1 can be selected. Notice at bottom of the PROCOMM Terminal Emulator window, that the XMODEM file transfer protocol has been selected, so both ends, MSS and VT-100 Emulator, are talking the same protocol.



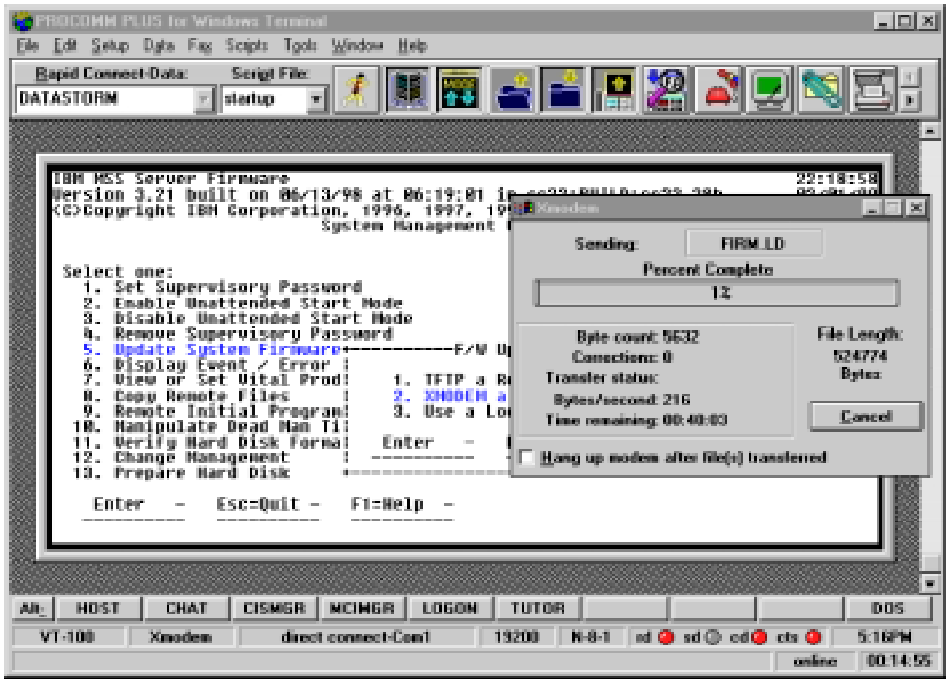
Selecting the File to be Transferred

Type the name of the file *FIRM.LD*, and press the enter key.



Start the XMODEM File Transfer Process

When prompted to start sending the file, go to the PROCOMM Terminal Emulator menu bar, click on the *Data* option, and then select *Send Data...* An window opens, where you select the directory and the file to be transferred to MSS, in this case *FIRM.LD*. After successfully transferring the file, reboot MSS.



Chapter 6: 8210 MSS Server Operational Code Update

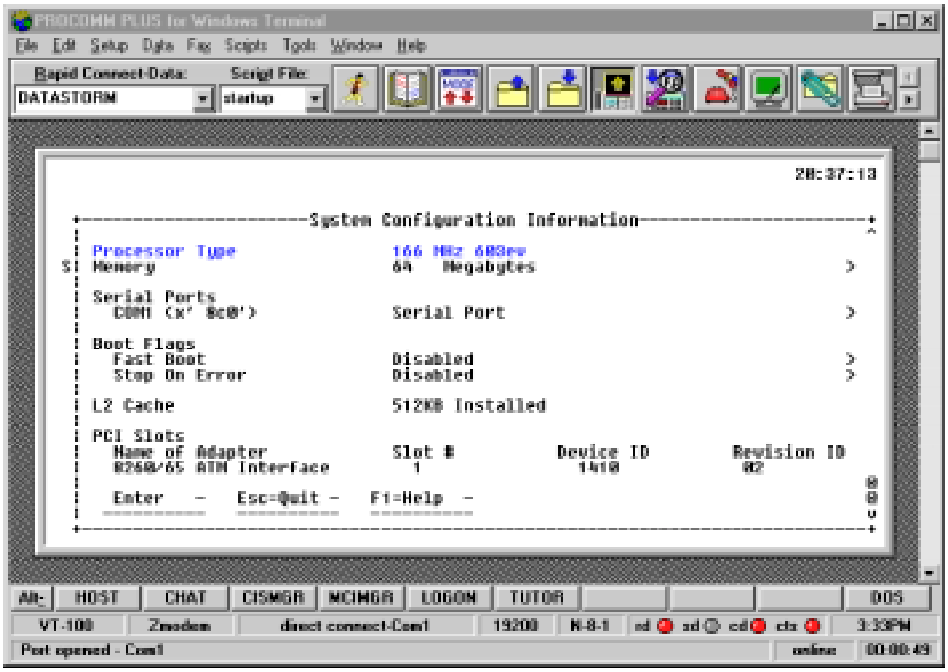
The 8210 MSS Server came pre-loaded with operational code version 2.1 PTF 2, and it has been updated to version 2.1 PTF 4, which was the latest version available when the network has been assembled. As off April the 6th, the available operational code version at the IBM Web Site is 2.2 PTF 2.

MSS operational code version 2.1 requires 64 Mb of memory, and firmware version 3.21 or later. And since we are using the MSS Server Module, it is also required that the 8265 CPSW is running operational code version 3.3.4 or later. All the above requirements are met, and the operational code can then be updated.

There are two ways to update the operational code: in-band, using a TFTP server, or out-band, using XMODEM protocol. It is very, very advisable to use the first method, being the second method, very, very slow, and impractical.

Verifying the Memory Size

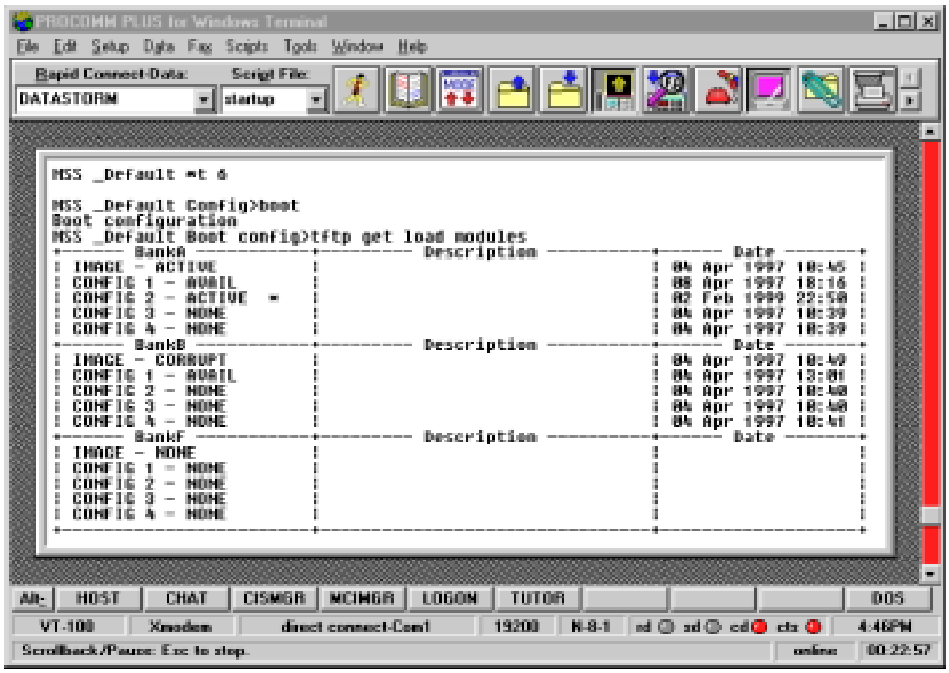
To determine the amount of memory installed in the MSS Server Module, it is necessary to access the firmware console (see section Accessing the System Management Services Menu, in Chapter 5). Select option *1. Manage Configuration* from the System Management Services menu, to display the System Configuration Information screen.



Setup the TFTP Parameters (1 of 2)

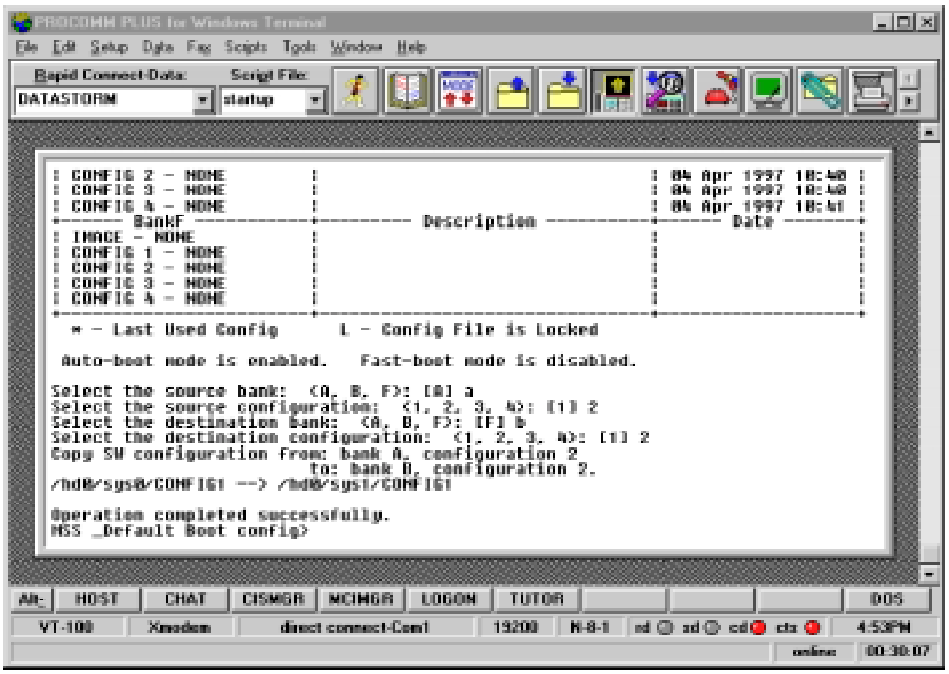
The TFTP parameters are setup using the Terminal Emulator program, and typing *T 6* from the MOS Operator Control console. Enter *BOOT* to go to the Boot Configuration screen. Enter the command *TFTP GET LOAD MODULES*.

Notice from the following window, that *bank A* contains the active operational code and *config 2* of bank A is the active configuration. Active operational code is the code that is actually running on the MSS Server Module, and active configuration is the configuration currently being used .



Setup the TFTP Parameters (2 of 2)

The MSS Server Module prompts for the TFTP server IP address (*10.1.2.3*), the directory where the operational code set of files is stored (\ *microcode21ptf4*), and the destination bank (*bank B).* At the end of the file transfer operation, MSS informs that the operation was completed successfully.



Copy MSS Server Module Initial Configuration (1 of 2)

There are two ways to initially configure MSS. One way is using the shipped default initial configuration, stored in bank A as configuration 1, which allows to configure it remotely, using the Configuration Program, or the command line interface, or the Web browser interface. The other way is through the QUICK configuration process, using the *QCONFIG* command.

MSS#1 was initially configured using the QCONFIG command, and the initial configuration was stored in bank A, configuration 2. Later, MSS#1 was reconfigured using the Configuration Program.

The following lists the default configuration parameters :

LECS - General Parameters

parameter	**value**
ATM device	0
ESI	400082100001
Selector	00

LECS - Assignment Policies

parameter	**value**
Priority	10, 20
Policy (priority 10)	by ELAN name
Policy (priority 20)	by LAN type

Signaling Protocol

parameter	**value**
ATM	auto detect

IP Configuration - Classical IP

parameter	**value**
Interface	0
Type	ATM
Slot	1
Port	1
IP Address	10.1.0.1
IP Mask	255.255.255.0
ARP Server	enabled
Refresh	5 minutes
Auto-refresh	enabled
ESI	400082100001
Selector	00
Maximum SDU size	9188

Emulated LAN Configuration - Token Ring

parameter	value
Name	TRelan1
Device	0
LES/BUS ESI	400082100001
LES/BUS selector	04
Policies (name)	TRelan1
Policies (type)	Token Ring

LEC Interfaces - Token Ring

parameters	value
Interface	1
MAC address	400082100001
Name	TRelan1
Type	Token Ring
Device	0
ESI	400082100001
Selector	02

IP Configuration - Token Ring LAN Emulation

parameter	value
Interface	1
LEC	TRelan1
IP address	10.1.1.1
IP mask	255.255.255.0

Emulated LAN Configuration - Ethernet

parameter	value
Name	ETHelan1
Device	0
LES/BUS ESI	400082100002
LES/BUS Selector	04
Policies (name)	ETHelan1
Policies (type)	Ethernet

LEC Interfaces - Ethernet

parameter	value
Interface	2
MAC address	400082100002
Name	ETHelan1
Type	Ethernet
Device	0
ESI	400082100002
Selector	02

IP Configuration - Ethernet LAN Emulation

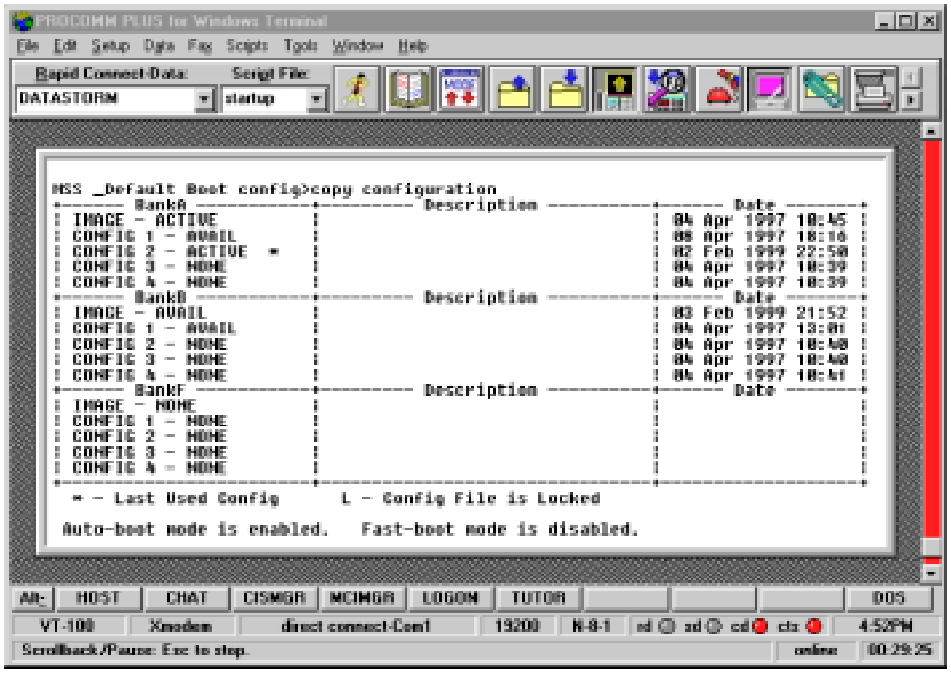
parameter	value
Interface	2
LEC	ETHelan1
IP address	10.1.2.1
IP mask	255.255.255.0

SNMP Communities

parameter	value
name	public
access type	read/write

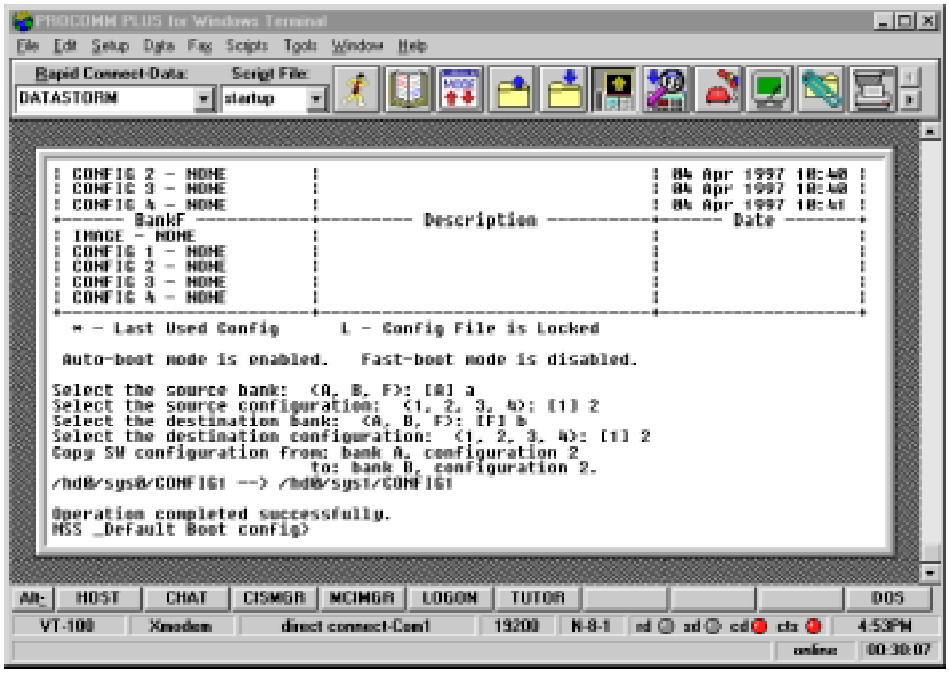
Because the MSS Server Module operational code update was installed in bank B, it is necessary to copy the initial configuration 2 from bank A to bank B. It is not possible to boot MSS from one bank, and use the configuration stored in another bank.

The *COPY CONFIGURATION* command has been used to copy the configuration.



Copy MSS Server Module Initial Configuration (2 of 2)

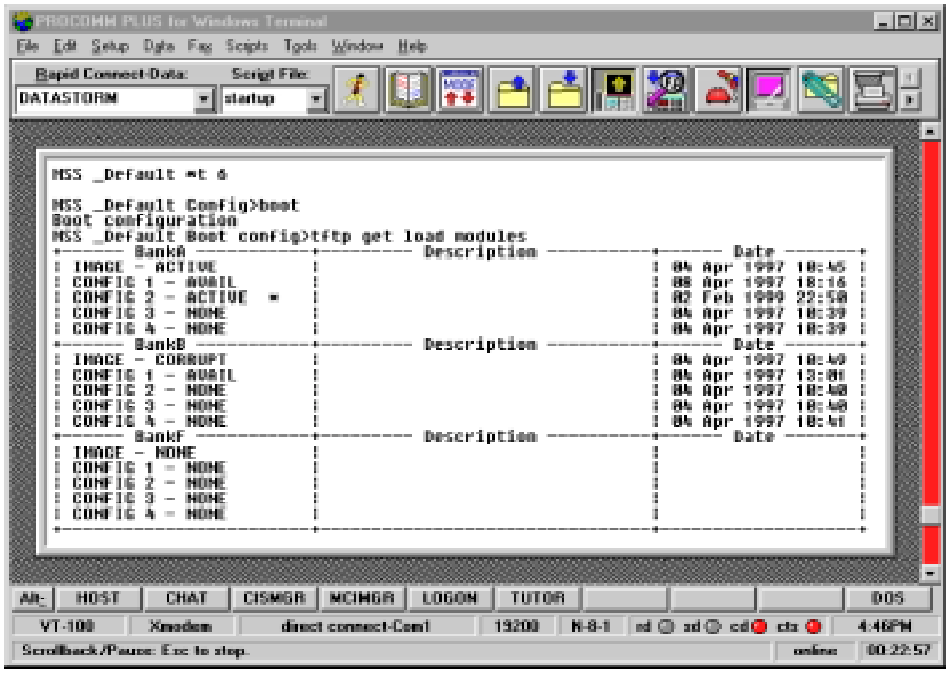
MSS Server Module prompts for the source bank (*bank A*), the source configuration (*config 2*), the destination bank (*bank B*), and the destination configuration (*config 2*).



Setup MSS Server Module Boot Information (1 of 2)

The new operational code has been installed in bank B. Just by rebooting MSS will not make it to load and run that microcode. It is required to instruct MSS to load the microcode from bank B, as opposed to bank A.

The *SET* command is used to setup the boot information, defining what bank to boot from, and what configuration to use.



Setup MSS Boot Information (2 of 2)

The MSS Server Module prompts for the source bank (*bank B*), the source configuration (*config 2*), and whether MSS is to boot only once from bank B, at the next reboot, or always.