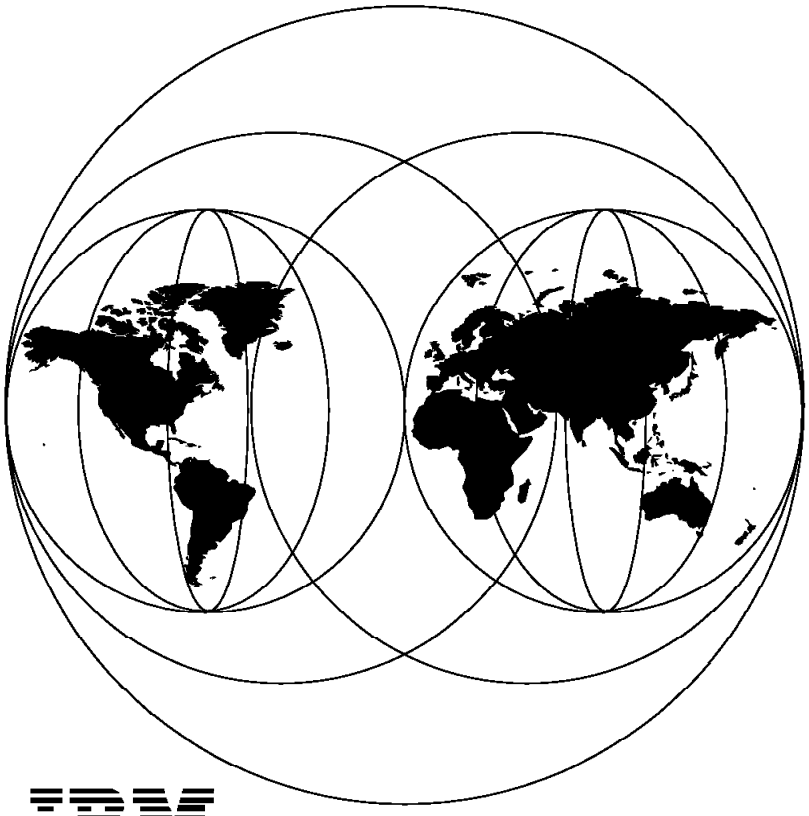
SG24-5003-00

# IBM 8260 As a Campus ATM Switch

June 1996



IBM

International Technical Support Organization
Raleigh Center

International Technical Support Organization

SG24-5003-00

# IBM 8260 As a Campus ATM Switch

June 1996

**Take Note!**

Before using this information and the product it supports, be sure to read the general information in Appendix E, “Special Notices” on page 315.

**First Edition (June 1996)**

This edition applies to V2.1.0 of the Control Point and Switch Module for use with the IBM 8260 Intelligent Switching Hub.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1996. All rights reserved.**
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

# Contents



© Copyright IBM Corp. 1996

# Preface

This redbook provides detailed coverage of the ATM switching function in the IBM 8260 Intelligent Switching Hub. It focuses on ATM switching and control point functions provided by the IBM 8260 as well as the ATM media modules that can be used to provide connectivity from workstations to the ATM network. It provides information about how to design, configure and implement ATM networks using the IBM 8260.

This book was written for customers and IBM specialists responsible for design and implementation of campus ATM networks. Some knowledge of local area networks, ATM, LAN emulation over ATM, and classical IP over ATM is assumed.

## How This Redbook Is Organized

This redbook is organized as follows:

- Chapter 1, “General Introduction to the 8260 and the ATM Implementation” This chapter provides a brief description of the IBM 8260 and the various 8260 token-ring and Ethernet media modules.
- Chapter 2, “ATM Backplane Architecture”

  This chapter provides detailed information about the ATM backplane used in the IBM 8260 Intelligent Switching hub.
- Chapter 3, “IBM 8260 ATM Modules”

  This chapter describes various 8260 ATM media modules.
- Chapter 4, “ATM Control Point and Switch (A-CPSW) Module”

  This chapter provides detailed information about the ATM switching and control point function provided by the Control Point and Switch module. It also covers the configurations steps to take advantage of these functions.
- Chapter 5, “Configuring 8260 ATM Media Modules”

  This chapter describes how to configure various 8260 ATM media modules.
- Chapter 6, “8260 Hardware Implementation”

  This chapter provides a brief description of the hardware components of the 8260 ATM switching subsystem and how they interact with each other.
- Chapter 7, “Nways Campus ATM Manager”

  This chapter describes the management functions that are available with an 8260 ATM network as well as how to use ATM Campus Manager
- Appendix A, “Introduction to ATM and Campus ATM Terminology”

  This chapter provides a brief introduction to ATM and Campus ATM terminology.
- Appendix B, “IBM LAN Emulation over ATM”

  This chapter provides a brief introduction to IBM′s proprietary LAN emulation architecture.
- Appendix C, “ATM Forum′s LAN Emulation”

© Copyright IBM Corp. 1996

This chapter provides a brief introduction to ATM Forum's LAN emulation specification.

- Appendix D, "Classical IP over ATM (RFC 1577)"

  This chapter provides a brief introduction to classical IP over ATM (RFC 1577).

## The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the Systems Management and Networking ITSO Center, Raleigh.

This project was designed and managed by:

Mohammad Shabani
International Technical Support Organization, Raleigh Center

The authors of this document are:

Mohammad Shabani
International Technical Support Organization, Raleigh Center

Zikrun H. Badri
IBM Australia

Guillermo H. Kretchmar
IBM Mexico

This publication is the result of a residency conducted at the International Technical Support Organization, Raleigh Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Volkert Kreuk
International Technical Support Organization, Raleigh Center

Chris Blenkhorn
IBM UK

Ray Collins
IBM USA, RTP

Joe Robinson
IBM USA, RTP

George Tardy
IBM France, LaGaude

## Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

**Your comments are important to us!**

# Chapter 1. General Introduction to the 8260 and the ATM Implementation

The IBM 8260 Multiprotocol Switching Hub is IBM′s flagship hub product. As its name implies, it is able to support Ethernet, token-ring and FDDI LANs, as well as Asynchronous Transfer Mode (ATM) switched campus networks, simultaneously.

The IBM 8260 is compatible with the IBM 8250 Intelligent Hub. This means that the 8260 supports all existing 8250 modules, except for the controller module. This chapter gives you a brief overview of the features of the 8260 and the non-ATM 8260 modules. For more information on the non-ATM features of the 8260, please refer to *8260 Multiprotocol Intelligent Switching Hub*, GG24-4370.

The subsequent chapters in this book discuss the details of the ATM support available in the IBM 8260.

## 1.1 8260 Models

The 8260 comes in four different models:

- 8260-010
- 8260-A10
- 8260-017
- 8260-A17

All these different models essentially have the same functionality. They only differ in their slot capacity and whether they have an ATM backplane. Models 010 and A10 have ten slots for media modules, while Models 017 and A17 have seventeen slots for media modules. All models have two additional slots reserved for controller modules. A diagram showing a Model 010 and 017 can be seen in Figure 1 on page 2.

Models A10 and A17 are similar to Models 010 and 017, respectively, except for the fact that they have an ATM backplane already installed when shipped from the factory.

You can upgrade Model 010 and 017 to Model A10 and A17, respectively, by installing the ATM backplane. The upgrade to the ATM backplane can be ordered as a field-installable feature that must be installed by an IBM service representative.

© Copyright IBM Corp. 1996

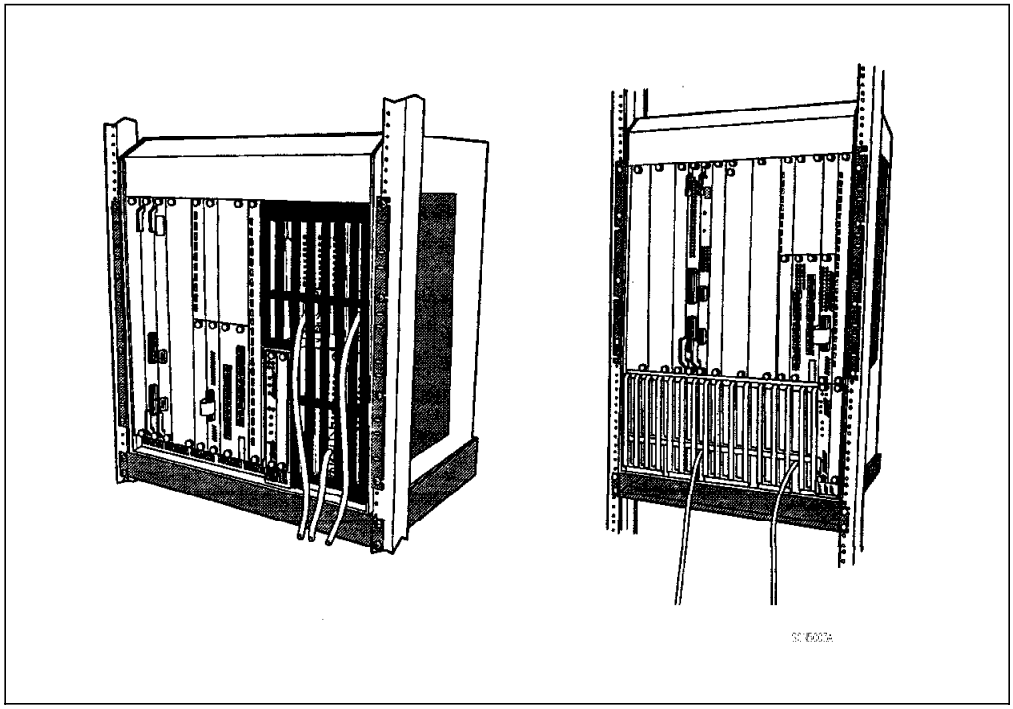


*Figure 1. 8260-010 and 8260-017*

## 1.2 8260 Backplane

In the 8260, all Ethernet, token-ring and FDDI modules have full floating capabilities. This allows the media and management modules to reside in any slot without restriction. The ATM modules also have full floating capabilities but with a few restrictions that are described in Chapter 2, “ATM Backplane Architecture” on page 21.

All modules are hot-swappable. This allows any media module to be removed without having to initially power down the hub or affect the operation of the reset of the modules installed in the hub. This allows you to replace any failed media module while the hub is operating.

When a failed module is replaced, the replacement module will automatically inherit the attributes of the original failed module if it is inserted into the same slot and is an identical module type. This allows for the replacement of the failed modules without the need to reconfigure the hub or the module.

The 8260 backplane is made up of three sections:

- Enhanced TriChannel
- ShuntBus
- ATM backplane

Figure 2 on page 3 shows how the backplane is organized within the 8260.

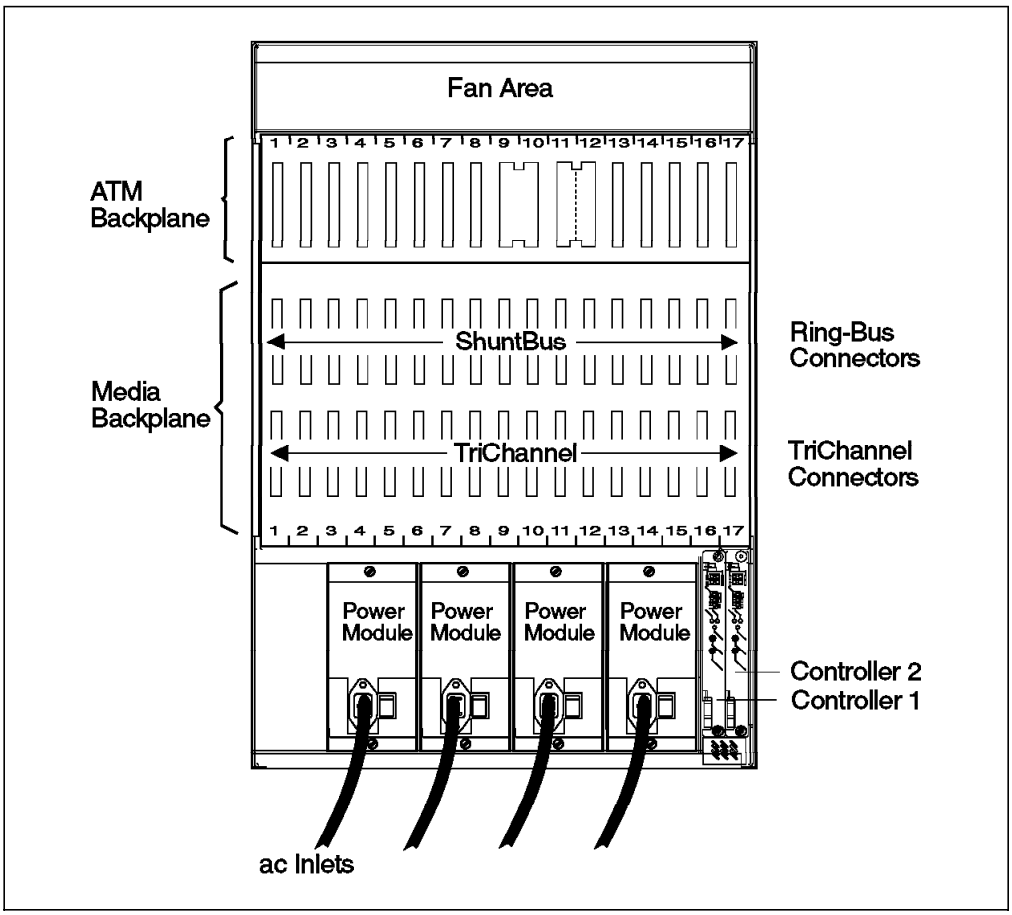



*Figure 2. 8260 Backplane*

### 1.2.1 Enhanced TriChannel Bus

The Enhanced TriChannel is physically the same as the TriChannel backplane that is found on the 8250 except that it uses some unused pins to allow the 8260 to support more Ethernet segments. The Enhanced TriChannel backplane allows you to have one of the following:

- Six Ethernet segments
- Seven token-ring segments
- Four FDDI segments

You can have a mixture of various LAN types on the Enhanced TriChannel. In this case, the number of permitted segments depends on the mixture of various LAN types installed.

### 1.2.2 ShuntBus

The ShuntBus gives the 8260 ten more token-ring segments and two more Ethernet segments. Unlike the TriChannel backplane, these token-ring and Ethernet segments are dedicated. This means there will always be ten token-ring backplane segments and two Ethernet backplane segments on the ShuntBus regardless of what other backplane segments are in use.

**Note**

The token-ring segments on the ShuntBus are in reality not totally dedicated. The ShuntBus has the capability to support FDDI segments using the same pins which are used for the token-ring segments. If these were used, the number of available token ring segments would be reduced. However, currently there are no FDDI modules that take advantage of the ShuntBus. Effectively, this means that the token-ring segments are dedicated.

### 1.2.3 ATM Bus

The ATM backplane is where all the ATM cells flow between ATM modules and the ATM switch. The ATM backplane allows the installation of one or two ATM Control Point and Switch (A-CPSW) modules, as well as up to 14 ATM media modules. The number of ATM ports supported on the 8260 depends on the type of ATM media modules that are installed. For example, by installing up to 14 100-Mbps 4-port concentrator modules, you can have up to 56 ATM ports.

### 1.2.4 Management Buses

In addition to providing Ethernet, token-ring and FDDI segments, the Enhanced TriChannel backplane provides two management buses called the management LAN (MLAN) and the serial control interface (SCI).

#### 1.2.4.1 Management LAN (MLAN)

The MLAN is a dedicated Ethernet bus that is used by the distributed management module (DMM) so that it can communicate with its media access control (MAC) daughter cards. The MLAN allows information such as network statistics and IP traffic to flow to and from the DMM and MAC daughter cards.

#### 1.2.4.2 Serial Control Interface (SCI)

The SCI is the same as that used in the 8250. All 8250, 8260 and ATM modules use the SCI to transmit module and port configuration data. The controller module uses the SCI to gather vital product data (VPD) from the media modules, and to get power and cooling status. Also, the controller module, in conjunction with the DMM, uses the SCI as the medium to change the status of the power supply to the modules, and to remove and add the power supply to the modules in the event of a change in the power or cooling subsystems. Figure 3 on page 5 illustrates the relationship between the MLAN, SCI and the 8260 modules.

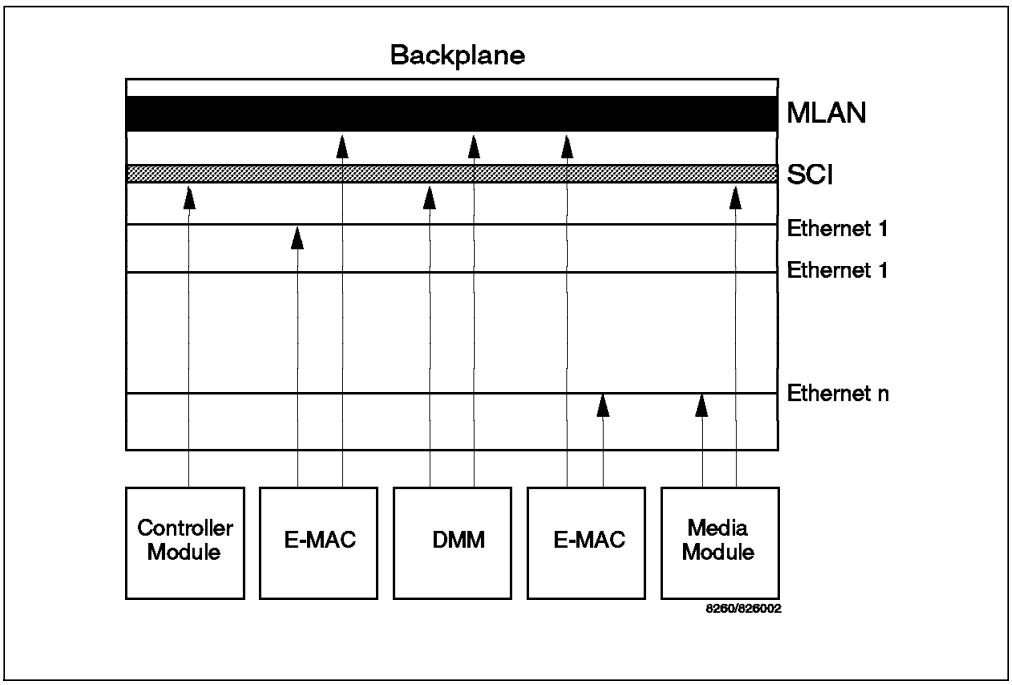



*Figure 3. 8260 Management Buses*

## 1.3 Power Subsystem

The 8260 will typically be used in backbone environments supporting hundreds of users. Recognizing that the 8260 will be used in highly critical areas, a sophisticated power management subsystem is provided by the 8260 to ensure high reliability and robustness.

The 8260 has the ability to house up to four load-sharing power supplies (or three in the Model A10 or 010). All power supplies are hot-swappable and are accessible from the front.

These power supplies can be configured in fault tolerant mode so that the 8260 is protected from a power supply failure. If used in fault tolerant mode, the power supplies keep some of their capacity in reserve so that if a power supply failure does occur there will be enough capacity left in the remaining power supplies to continue to power the entire hub. This process is nondisruptive, which means that nobody will be affected by the failure of a single power supply.

If the available power is not enough after a power supply failure (for example, if the 8260 was not in a fault tolerant power supply configuration), then the modules will automatically be powered down in a prioritized and orderly manner until enough modules are shut down to allow the remaining power supplies to cope with the power requirements of the remaining modules and chassis. This function is made possible because each 8260 module is able to have a priority assigned to it, allowing the 8260 to shut down the modules with the lowest priority before having to shut down those with the higher priority. This ensures that if a power supply failure is going to have any impact, it always causes the least possible disruption.

The 8260 has the concept of a power budget. This means that the 8260:

1. Knows how many power supplies are installed and which ones are working
2. Knows how much power is being used by the modules which are already installed
3. Interrogates newly inserted 8260 modules to determine their power requirement

The above features allow the 8260 to determine if there is enough power in the 8260 to power a newly installed module before full power is given to it. If there is, it will be powered up as usual. If not, the 8260 will not power up the module and therefore not impact any of the existing modules.

## 1.4 Cooling Subsystem

To complement the 8260's intelligent power subsystem, the 8260 also has an intelligent cooling subsystem that is operated from the controller module.

Each 8260 is shipped with three cooling fans. Each of the three fans cools an overlapped area in the hub covering eight slots as shown in Figure 4. The slots covered by each fan are:

- Fan 1 is responsible for slots 1-8.
- Fan 2 is responsible for slots 6 -13.
- Fan 3 is responsible for slots 10-17.

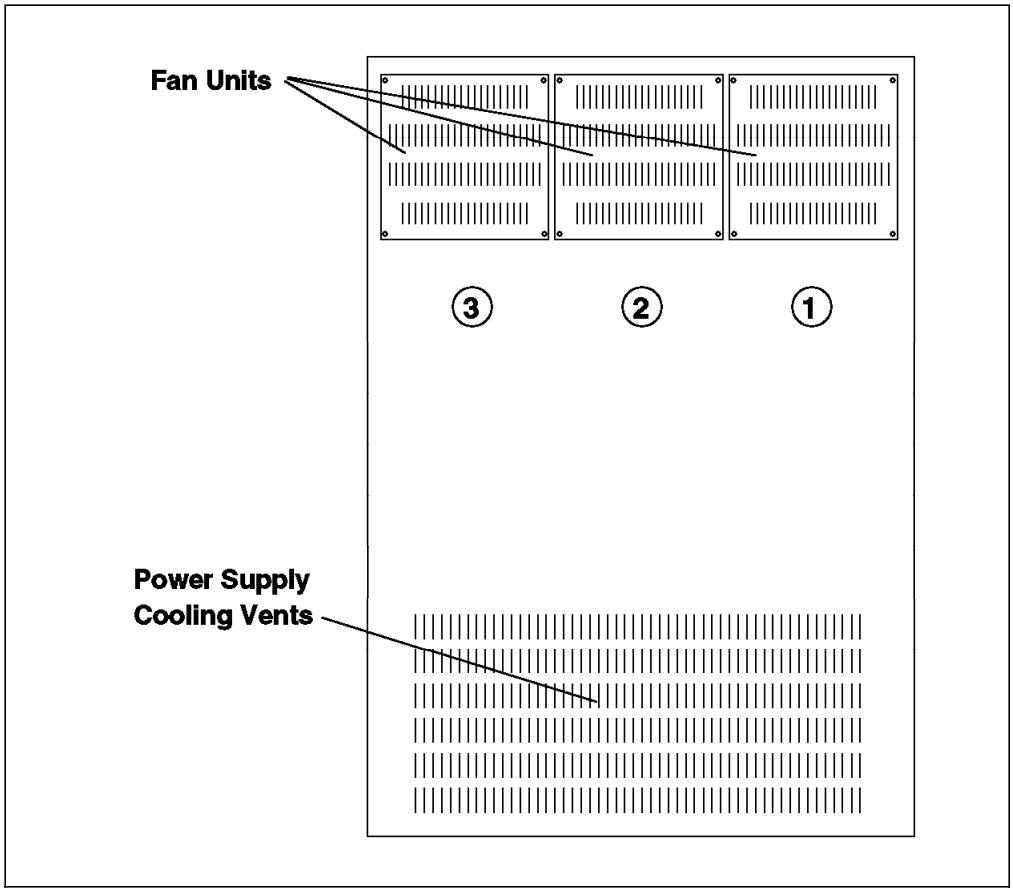



*Figure 4. 8260 Fan Units*

The controller module continually monitors all the sensors via the SCI. If a fan unit stops or the temperature in any of the cooling zones rises above 60 C, the controller module may, depending on a user configurable parameter, use the SCI to power down some of the 8260 modules in the affected cooling zone in order to bring down the temperature to an acceptable level. Note that within the affected area, the modules will be powered down in a prioritized manner as specified by the module priority.

**Note:** The module priority can be assigned by the user as part of the configuration process.

## 1.5 8260 Distributed Management Architecture

To fully manage the 8260 and the installed modules, the 8260 uses a distributed management architecture. In this architecture, the various tasks of managing the various elements of the hub are distributed across the following elements:

- Distributed management module
- MAC daughter cards
- Controller module

There are two types of distributed management modules (DMM):

- Stand-alone DMM
  - The DMM is called a stand-alone card because it does not have any mounting facility for the daughter cards.
- EC-DMM
  - This module allows you to mount up to six Ethernet Medium Access Carrier (E-MAC) daughter cards on it. At the time of writing, there was no carrier DMM available for mounting token-ring MAC (T-MAC) daughter cards.

In terms of management functions, DMM and EC-DMM are identical. The only difference between these two cards is their ability to house Ethernet MAC daughter cards. Therefore, as this section is discussing management in general, the term DMM will be used to refer to both 8260 management modules (stand-alone DMM and EC-DMM). In the next section, we look at the specific management modules and discuss their capabilities and their differences.

The DMM, along with the fault tolerant controller module, manages and controls the 8260 hub and its modules. However, to perform certain management functions such as network traffic monitoring, there is a need for a daughter card to assist the DMM. There are two types of daughter cards:

- Ethernet Medium Access Carrier (E-MAC) daughter card
- Token-ring Medium Access Carrier (T-MAC) daughter card

These daughter cards provide the following two functions:

- Interface to the backplane segments

  To be able to communicate with devices attached to any of the backplane segments, DMM requires an interface to that segment. The interface to the Ethernet segments on the backplane is provided to DMM via E-MAC, whereas T-MAC allows DMM to interface with the token-ring segments on the ShuntBus. Note that DMM requires one MAC daughter card for each

network on the backplane through which DMM is going to communicate with the other devices.

DMM will use the interface to the backplane segments to communicate with the devices attached to these segments using IP. For example, to be able to manage the 8260 via an SNMP manager, DMM must have an interface to a network through which the SNMP manager can be accessed.

- Network monitoring

  Daughter cards attach to the appropriate backplane segment (token-ring or Ethernet), listen to the traffic flow and pass all the information back to DMM.

**Note:** Ethernet MAC daughter cards can be installed on EC-DMM or Ethernet media modules, whereas token-ring MAC daughter cards must always be installed on token-ring media modules.

The combination of DMM and daughter cards provides a cost-efficient management architecture that consolidates media management into a single card, while distributing network monitoring across a series of protocol-dependent daughter cards. The DMM is a generic (protocol-independent) module that can be used for both inband and out-of-band management. As mentioned earlier, when used for inband management, DMM requires a daughter card. The flexibility and reduction in cost is achieved by distributing the network monitoring function to daughter cards that can be mounted on EC-DMM (for E-MAC only) or media modules, so they do not use any valuable payload slots. This also means you only need one DMM to manage the entire 8260. If your network grows and you need to invest in more network monitoring functions, you can install additional daughter card(s) matching the protocol of your new network(s) by simply mounting them on the existing media module or EC-DMM (for E-MAC only).

The MAC daughter cards will be assigned to the token-ring or Ethernet backplane using DMM commands. Once assigned to a backplane segment, they will be able to monitor the traffic on that segment and pass the collected information to the DMM. Note that the MAC daughter cards installed on the media modules will communicate with the DMM (or EC-DMM) using the MLAN, as shown in Figure 5 on page 9. The E-MACs installed on the EC-DMM, however, will use the onboard circuitry of the EC-DMM to communicate with DMM.

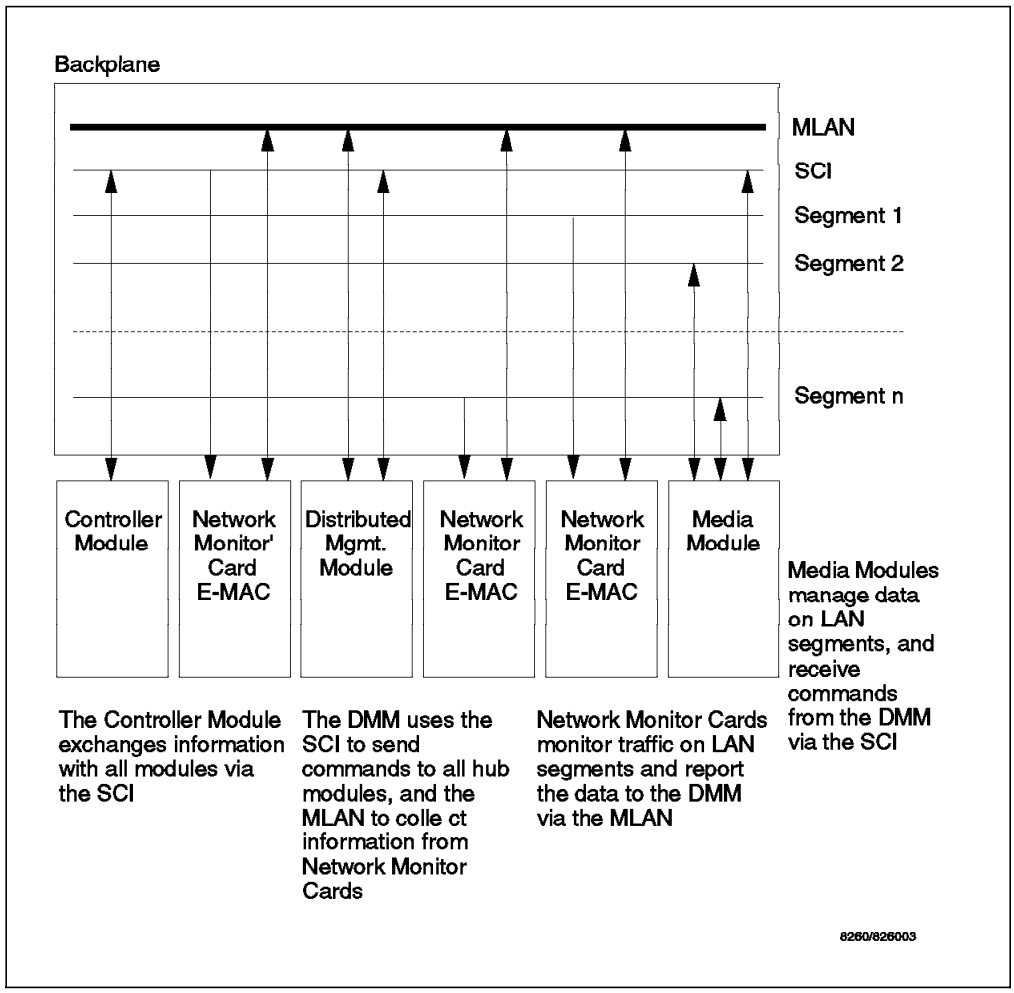



*Figure 5. Management Schematic*

The DMM (and daughter cards) provide management and control facilities in the following areas:

- **Configuration**

  The DMM, networks, modules, and port settings can be configured through the DMM using DMM commands. The DMM can be used to configure 8250 as well as 8260 modules.

- **Statistics and fault reporting**

  E-MAC and T-MAC provide support for collecting an extensive range of statistics based on RMON.

- **Out-of-band and inband downloading**

  The DMM provides both inband and out-of-band download features for downloading new software to DMM, media modules, and daughter cards. Trivial file transfer protocol (TFTP) is used for inband downloads. Out-of-band downloads allow you to download software using the Xmodem protocol from a local or modem-attached PC (with ASCII emulation software) attached to the RS-232 port on the front panel of the DMM.

- **SNMP support**

In a simple network management protocol (SNMP) managed environment, the DMM acts as the SNMP agent, responding to SNMP requests and generating SNMP traps.

- **Telnet support**

  Using Telnet you can log in remotely to any DMM on the network and manage it from the remote station. You can also use Telnet from the terminal attached to the DMM to log in to any other device that supports Telnet.

- **Inventory**

  The DMM provides a complete inventory of the hub, including power supplies, fans and modules installed in the 8260.

- **Staging**

  The media modules save their configuration information in an onboard nonvolatile RAM (NVRAM). This means flexibility for network managers as they can configure the modules at a central site and then send them out to the remote locations for installation.

- **Power management**

  The DMM, when used in conjunction with the fault tolerant controller module, can be used to manage the power subsystem. For example, it can set power classes for modules and turn power fault tolerance on and off.

- **Mapping**

  DMM allows you to display a detailed topological ring map including address-to-port mapping about the token-ring segments on the network.

### 1.5.1 MAC Daughter Cards

To be able to monitor the network traffic activity on the backplane segments, as well as to be able to communicate with other stations using IP, DMM requires the services provided by MAC daughter cards.

These daughter cards connect to the networks, listen to the traffic flow and pass traffic information back to the DMM. They also provide the DMM with the interface to the networks on the backplane so that it can communicate with the other stations on that network.

The MAC daughter cards are protocol-specific cards with the following types available:

- The E-MAC (Ethernet - Media Access Card)
- The HE-MAC (High-End Ethernet - Media Access Card)
- The T-MAC (token-ring - Media Access Card)

These daughter cards can be installed on the media modules that use the same protocol. That is, T-MACs can be installed on token-ring media modules, whereas E-MACs and HE-MACs can be installed on Ethernet media modules. Each token-ring or Ethernet media module can accommodate installation of one MAC daughter card. (Ethernet 40-port module allows the installation of two MAC daughter cards.) Additionally, the E-MACs and HE-MACs can be installed on the EC-DMM. Each EC-DMM can accommodate the installation of up to six E-MACs.

Regardless of where the MAC daughter cards are installed, they can be assigned to any of the backplane segments. However, to assign a MAC daughter card to an isolated segment on a media module, the MAC daughter card must be installed on that media module.

**Note**

E-MACs installed on EC-DMM can collect detailed statistical information about *all* of the ShuntBus and Enhanced TriChannel Ethernet segments. This statistical information includes network as well as module and port-level information. This information is collected for both 8260 and 8250 Ethernet modules. (Note that 8250 Ethernet modules may attach to Ethernet_1 through Ethernet_3 segments only.)

The E-MACs installed on the media modules can collect full statistics (network, module and port-level statistics) for Ethernet_4 through Ethernet_8 segments only. For Ethernet_1 through Ethernet_3, they can collect network, module and port-level statistics for 8260 Ethernet modules, but for the 8250 modules attached to these segments they can only collect network level statistics and cannot report module or port-level statistics. This is due to the use of parallel addressing by the 8250 modules. Therefore, if you are planning to monitor Ethernet_1 through Ethernet_3 segments which include 8250 Ethernet modules, you must ensure that the E-MACs used to monitor those segments are installed on EC-DMM.

Because of the possibility of installing MAC daughter cards on the 8260 modules, the 8260 modules are identified by *slot* and *subslot* identifiers. Note that the slot and subslot identifiers are used in DMM commands to refer to the media modules, management modules or daughter cards. The following is a summary of how to identify the slot and subslot for each media module, management module, and daughter card:

1. Each media module is always considered to be on the first subslot of the slot on which the media module is installed. For example, if you have installed a 24-port Ethernet media module in slot 2, this will be identified as module 2.1 (slot 2, subslot 1). This is regardless of the fact that the media module may or may not have a MAC daughter card installed on it.
2. If a MAC daughter card is installed on a media module, the daughter card is considered to be in subslot 2 of the slot in which the media module is installed. For example, if the previously mentioned 24-port media module had an E-MAC installed on it, the E-MAC will be considered to be module 2.2, whereas the 24-port module is 2.1.
3. The stand-alone DMM is always considered to be on the first subslot of the slot in which the stand-alone DMM is installed. Note that a stand-alone DMM does not have the housing for a MAC daughter card.
4. In the case of an EC-DMM which does have the housing for six E-MACs, the EC-DMM module is always considered to be in subslot 1 of the slot in which the EC-DMM is installed. Also, the DMM part of EC-DMM is always considered to be in subslot 8. If there are any E-MAC daughter cards installed in the EC-DMM, they will be considered to be in subslots 2 through 7 of the slot on which EC-DMM is installed.

Figure 6 on page 12 shows how the slot and subslot IDs are used on an EC-DMM.

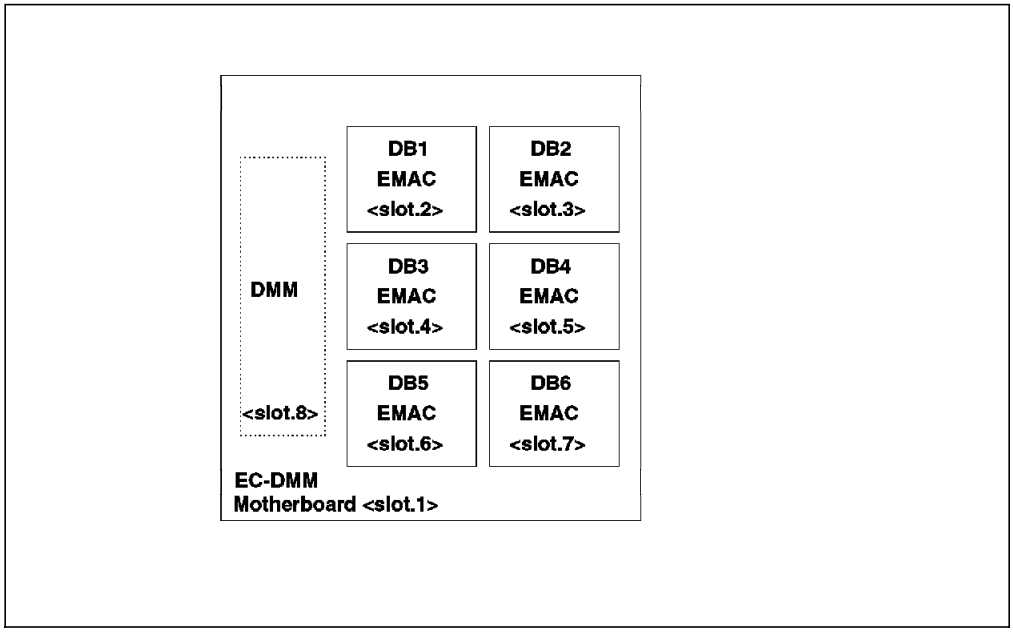



*Figure 6. EC-DMM Slots and Subslots*

## 1.6 8260 Fault Tolerant Controller Module

The controller module is an essential component of the 8260 and provides the following functions:

- Clock generation and its distribution across Enhanced TriChannel and ShuntBus
- Monitoring the hub temperature and taking appropriate action in overheated conditions
- Inventory Management by interrogating the modules' vital product data.
- Power management functions

The 8260 Fault Tolerant Controller Module does not occupy any of the payload slots in the hub. This module resides in one of the two reserved controller module slots. The second controller module slot can be used for a backup controller module.

**Note**

The 8250 controller module will not work in the 8260. This is the only 8250 module that is not supported in the 8260.

## 1.7 8250 Module Support

All 8250 modules, except for the controller module, are supported in the 8260. 8260 modules are approximately twice the height of an 8250 module, so special filler plates need to be installed when using 8250 modules in the 8260.

There are many 8250 modules that support functions from port concentration to dial-in access to LANs. Since all these modules are supported on the 8260, this

gives you a high degree of flexibility over the configuration that can be contained within a single chassis.

When using 8250 modules in the 8260, there are a few restrictions:

- 8250 modules do not support power management by the DMM.
- 8250 modules cannot use the backplane to connect to those LAN segments that reside on the ShuntBus or the additional segments on the Enhanced TriChannel. These segments can, however, be connected together via an external method.
- The 8250 modules cannot reside in slot 17. This is because in the 8250 this slot is reserved for the controller module and would never have any other type of module in that slot.

## 1.8 Ethernet Capabilities

The 8260 has the capability to support up to eight Ethernet segments on the backplane. Two of the eight segments reside on the ShuntBus and six more reside on the Enhanced TriChannel. The two Ethernet segments on the ShuntBus are only accessible via 8260 modules and will always be available regardless of what other backplane segments may be in use. There are six Ethernet segments that are available on the TriChannel backplane. Only three of these segments are accessible via 8250 modules because the original TriChannel on the 8250 supports up to three Ethernet segments.

Note that if the Enhanced TriChannel has to be shared with other token-ring or FDDI segments, then the number of available Ethernet segments on the Enhanced TriChannel will be reduced.

There are a number of 8260 Ethernet modules available. The following sections summarize these modules.

### 1.8.1 Ethernet 24-Port 10Base-T Module

This module has two telco connectors that enable the connection of up to 24 10Base-T ports. It provides eight isolated Ethernet segments on the module as well as the ability to connect the ports to the eight Ethernet segments on the backplane. The isolated segments on the module allow you to set up small workgroup LANs without using any resources on the backplane.

This module provides a per-port switching capability that allows each port on this module to be connected to any of the backplane or isolated segments. However, note that the module supports connection to a maximum of six segments (backplane or isolated) simultaneously.

The ports also support non-10Base-T compliant devices, port redundancy and auto-polarity detection.

One E-MAC daughter card and one Ethernet Security (E-SEC) daughter card can be installed on this module.

This module supports up to six segments simultaneously. This means that the ports on this module can be connected to a maximum of six segments at any one time. These segments can be a combination of backplane and isolated segments.

### 1.8.2 Ethernet 36-Port 10Base-T Module

This module is very similar to the Ethernet 24-Port 10Base-T Module except that it has three telco connectors to connect up to 36 10Base-T ports. The other difference is that it does not support per-port switching. Instead it switches ports at the connector level which means only groups of 12 ports can be assigned to an Ethernet segment. It can, however, carry two E-MAC and one E-SEC daughter cards simultaneously.

### 1.8.3 Ethernet 20-Port and 40-Port 10Base-T Module

These two modules support RJ45-based 10Base-T wiring. The 20-Port module occupies one slot while the 40-Port module occupies two slots. Each module has either 20 or 40 twisted-pair Ethernet ports to connect 20 or 40 devices respectively. These 10Base-T modules support per-port switching and are able to switch any port to any of the eight isolated segments or any of the eight backplane segments. Unlike the Ethernet 24-Port and 36-Port 10Base-T modules, these modules are able to have a maximum of eight different segment connections in use simultaneously.

### 1.8.4 Ethernet Flexible Concentration Module

This module is sometimes called the EtherFlex module. It is basically an Ethernet module shell with four bays where you can install various types of I/O cards. This allows you to have a module that can support various types of media, for example, a combination of fiber and copper ports.

The EtherFlex module can support the following I/O Cards:

- BNC 10Base2.

  This I/O card has three 10Base2 ports.

- RJ-45 10Base-T.

  This I/O card has four 10Base-T ports.

- Male AUI.

  This I/O card has three male AUI ports.

- Female AUI.

  This I/O card has three female AUI ports.

- 10Base-FB/FL.

  This I/O card has two fiber ports. Each port automatically detects whether it is 10Base-FB or 10Base-FL. The I/O card can be ordered with ST, FC or SMA connectors.

**Note**

The male and female AUI I/O cards occupy two bays of the EtherFlex module. They can be installed in Bays 1 and 2 or Bays 3 and 4. All the other I/O cards only occupy one bay.

This module is a per-port switching module so every port is able to be switched to any of the eight backplane or eight isolated segments. This is regardless of what I/O cards are installed. The maximum number of different segment connections that it can support is eight.

This module can carry up to two E-MACs and two E-SECs simultaneously.

### 1.8.5 Ethernet 10-port 10Base-FB Module

This is a single-slot module with ten 10Base-FB ports. The ports on this module can connect to any of its four isolated segments in addition to being able to connect to any of the eight backplane segments. The module supports per-port switching allowing each port to connect to any of these segments. A maximum of ten different segment connections are supported on this module.

### 1.8.6 Ethernet Security Card

The Ethernet Security Card (E-SEC) is a daughter card that can be connected on any of the Ethernet media modules or the EC-DMM. It does not occupy an additional slot in the 8260 chassis. The E-SEC secures any backplane Ethernet network to which it is connected. If more than one Ethernet segment needs to be secured, then additional E-SECs can be installed.

The E-SEC has the following security functions:

- Intrusion detection

  This prevents intruders from transmitting information to other ports.

- Eavesdropping protection

  This prevents any user from examining the contents of a packet transmitted on the backplane and is destined to another station.

- Automatic intruder disablement

  The E-SEC can automatically disable a port when it detects intruders.

### 1.8.7 Ethernet Interconnect Module

The 8260 Ethernet Interconnect module is a six- to eight-port bridge/router. The single-slot version provides six backplane ports that can be connected to six of the eight backplane Ethernet segments. The two-slot model can be configured with optional I/O cards to support two additional ports to connect to external token-ring and/or Ethernet LANs. The two-slot version supports the following optional I/O cards:

- 10Base-T I/O card
- 10Base2 I/O card
- 10Base5 I/O card
- Token-ring I/O card

The interconnect module supports many bridging and routing functions, some of which are included in the following:

- Transparent bridging between Ethernet segments
- Transparent bridging between token-ring segments
- Source route bridging between token-ring segments
- Translational bridging between token-ring and Ethernet segments when both the token-ring and Ethernet segments use translational bridging
- IEEE 802.1d Spanning Tree Protocol
- IP RIP and OSPF routing

- NetWare IPX routing
- DECnet Phase IV routing

## 1.9 Token-Ring Capabilities

The 8260 provides significant enhancements to the 8250 for the support of token-ring LANs. Some of the features are:

- Ten additional token-ring backplane segments

  These additional segments are on the ShuntBus and will always be available (as long as there are no 8260 FDDI modules installed on the ShuntBus as described in 1.2, “8260 Backplane” on page 2) irrespective of what other backplane segments may be in use.

  **Note**

  All 8260 token-ring modules use only the ten ShuntBus segments and do not connect to any of the token-ring segments on the Enhanced TriChannel backplane. The 8250 token-ring modules installed in the 8260 only connect to the token-ring segments on the Enhanced TriChannel.

- Active retiming per port

  The 8260 active token-ring modules will retime each port giving the ability to support longer lobe lengths.

- Improved beacon recovery

  The beacon recovery process has been improved to allow the 8260 media modules to detect and disable the port causing the beacon condition on the ring without the help of the DMM.

- Automatic speed detection

  When enabled, the modules will not allow any device to insert into the ring at the wrong speed.

- Address-per-port mapping

  The 8260 is able to map the MAC address to the port in the hub. This feature will greatly assist in fault diagnosis and network management. This feature also supports the use of fan-out devices, where up to eight stations can be connected to a single port using a fan-out device.

The following is a brief description of all the 8260 token-ring modules.

### 1.9.1 Token-Ring Active Port-Switching Media Module

This module is a single-slot module with 18 active retiming lobe ports. This is a per-port switching module which will allow each port to be switched onto any of the ten backplane segments available or any of the eleven isolated segments on the module. It can support a maximum connection of eleven different segments per module.

Two of the ports on this module (ports 17 and 18) can be reconfigured to be RI/RO ports to connect to other token-ring hubs.

This module supports the attachment of one T-MAC and one Jitter Attenuator Daughter Card (JADC).

Each port is actively retimed using dual-phase locked-loop circuitry. The active retiming of each port enables maximum lobe lengths, as listed in the following table.

*Table 1. Lobe Lengths Supported with Token-Ring Active Port-Switching Media Modules*

| Cable Type | 4 Mbps Ring | 16 Mbps Ring |
|---|---|---|
| UTP Category 3 | 250 m | 100 m |
| UTP Category 4 | 425 m | 210 m |
| UTP Category 5 | 425 m | 225 m |
| FTP | 425 m | 225 m |

### 1.9.2 Token-Ring Active Module-Switching Module

This module is identical to the Token-Ring Active Port-Switching module described earlier except its ability to switch individual ports to different segments. In this module, only the whole module can be switched to one of the ten backplane segments or to an isolated segment. All other features are the same.

### 1.9.3 Token-Ring Passive Media Module

This module provides 20 lobe ports in a single-slot module. It supports switching of the 20 ports on a per-module basis to any of the ten backplane segments or to the single isolated segment on the module. It is able to support the installation of one T-MAC and has a JADC built in.

### 1.9.4 Dual Fiber Repeater Module

This is a single-slot module that supports two sets of fiber RI/RO ports in addition to ten active retiming lobe ports. Each lobe port can support the same lobe lengths as those on the active token-ring modules. This is a per-port switching module which will allow each port to be connected to any of the ten backplane segments or the eleven isolated segments. It can simultaneously support the connection of up to 11 different token-ring segments. This module can accept two JADC and one T-MAC.

### 1.9.5 Token-Ring Jitter Attenuator Daughter Card (JADC)

This daughter card is supported on all 8260 token-ring modules that support RI/RO capability. One JADC card is required for every RI/RO trunk that is connected to any non-IBM 8260 equipment. This card will ensure that any excessive jitter is filtered out before being passed to the 8260.

## 1.10 FDDI Capabilities

The ShuntBus has the capability of supporting four FDDI segments. Currently, however, there are no 8260 FDDI modules available that utilize the ShuntBus. This effectively means that FDDI support is achieved only through the Enhanced TriChannel backplane with 8250 modules. The Enhanced TriChannel can support a maximum of four FDDI segments. This number will be reduced if they have to share the Enhanced TriChannel with other token-ring or Ethernet LAN segments.

## 1.11 ATM Capabilities

One of the key features of the 8260 is the ability to support ATM. It is designed to be the basic building block in a local, privately owned and administered ATM campus network. The following sections summarize the ATM components of the 8260. More details about these components are provided in the following chapters.

### 1.11.1 ATM Backplane

All the data transmitted between ATM modules in the 8260 pass through the ATM backplane. The ATM backplane coexists with the other two backplanes in the 8260 (Enhanced TriChannel and ShuntBus). It is a standard feature of the IBM 8260 Model A17 and Model A10 and can be installed in the 8260 Model 017 and 010 as an optional feature.

### 1.11.2 ATM Control Point and Switch (A-CPSW) Module

The ATM Control Point and Switch (A-CPSW) module provides the ATM cell switching function as well as the control point functions associated with the establishment and management of the ATM connections. This module consists of the following two cards, which are packaged into a two-slot module:

1. The base card (ATM switch fabric)

   The ATM switch fabric switches cells between the ATM endsystems attached to the 8260 via ATM media modules. It is physically located on the left side of the A-CPSW module when looking at the module as it would normally reside in the 8260.

   The switch fabric implemented in the A-CPSW uses the IBM switch-on-a-chip, which is described in 2.5.1, “ATM Switch” on page 30.

2. The control point card

   This card houses the processors where the Control Program resides. The Control Program performs the functions associated with the establishment and management of ATM circuits. It is physically located on the right side of the A-CPSW module when looking at the module as it would normally reside in the 8260.

   The control point card houses a RISC MC 68EC040 and anMC 68EN360 processors in companion mode for the control point functions, and an ASIC/VLSI processor in the switch and media modules for the ATM switching functions.

   An ATM campus network can comprise many 8260s, each with its own A-CPSW module. This means that the control point functions of the ATM network are distributed, which will give the ATM 8260 networks high availability and scalability. Also, using A-CPSW microcode V2.1.0, you may install a second A-CPSW module in each 8260 Model 17, to be used as a backup in case of the failure of the primary A-CPSW module.

### 1.11.3 ATM 4-Port 100 Mbps (A4-FB100) Module

This is a single-slot module that provides four 100-Mbps TAXI interfaces using 62.5 micron multimode fiber. The physical layer for this module follows the FDDI PMD specifications. The fiber connectors for this module can be either MIC duplex connectors or SC connectors.

### 1.11.4 ATM 155-Mbps Flexible Concentration (ATMflex) Module

This is a single-slot module that provides two slots for installing daughter cards. The physicals interfaces supported for these two interfaces can be fiber or copper cable, depending on the type of daughter cards used. The daughter cards that are available for this module are:

- ATM 155-Mbps 1-port Fiber Multimode I/O Card using SC connectors
- ATM 155-Mbps 1-port Fiber Monomode I/O Card using SC connectors
- ATM 155-Mbps 1-port UTP/STP I/O Card using RJ-45 connectors

### 1.11.5 8260 ATM TR/Ethernet LAN Bridge (8281) Module

The ATM LAN Bridge module is a two-slot module that provides bridging function between the ATM and LAN ports. The module mainly does the following:

- Provides four ports for interconnection to LANs. These ports can be configured for either Ethernet (IEEE 802.3 and DIX V2) or token-ring. The hardware for supporting token-ring and Ethernet is the same. It is only via the configuration of the module that you specify the ports are connected to token-ring or Ethernet. Note that all the ports should be configured for the same type of LAN. This module does not allow the mixing of various LAN types.

  The LAN ports are external ports. This means that this module does not have an interface to the ShuntBus or TriChannel of the 8260 backplane for connection to the LANs on the 8260 backplane.

- Provides an ATM connection to the A-CPSW module. This connection is via the ATM backplane and supports ATM 3.0 UNI specifications.

  The ATM port can be configured to take part in emulated LAN compliant with IBM LAN emulation. The LE client can be configured to support either an emulated token-ring or an emulated Ethernet. Note that the emulated LAN must be the same type of LAN as the one used on the four LAN ports.

- Provides bridging between the ATM port and the LAN ports, as well as bridging between the LAN ports.
- Supports source route bridging when the ports are configured to use token-ring.
- Supports transparent bridging when the ports are configured to use Ethernet.
- Supports 256 virtual circuits (VCs) over the ATM connection.
- The LE client in the ATM LAN bridge (implementing IBM′s proprietary LAN emulation) can establish connection with a single LAN emulation server. This means that the bridge can only be a member of a single emulated LAN over ATM.
- Management of the ATM LAN Bridge module is done using SNMP. The SNMP agent within the module supports various MIBs, as described in 3.5.5, “Management and Configuration Support” on page 67.

The SNMP agent in this module is accessible via IP over the ATM and LAN ports. Access over the ATM port requires IBM LAN emulation over ATM. This means that the ATM-attached stations using Classical IP over ATM (RFC 1577) cannot communicate with this module directly over ATM.

### 1.11.6 ATM 12-Port 25-Mbps Concentration Module

This is a one-slot module that provides twelve 25-Mbps interfaces using UTP/STP cabling as specified in the ATM Forum specification for 25-Mbps interface. The type of interface provided on this module is shielded RJ-45.

# Chapter 2. ATM Backplane Architecture

All data transmitted between ATM modules in the 8260 pass through the ATM backplane. This backplane can coexist with the other two standard backplanes in the 8260 (Enhanced TriChannel and ShuntBus).

The following sections provide detailed information about the ATM backplane and how it operates.

## 2.1 Introduction to the ATM Backplane

The ATM backplane is located in the upper part of the 8260 chassis, just above the ShuntBus, as shown in Figure 7. This allows all 8250 and non-ATM 8260 modules to fully coexist with the ATM modules in the same 8260.

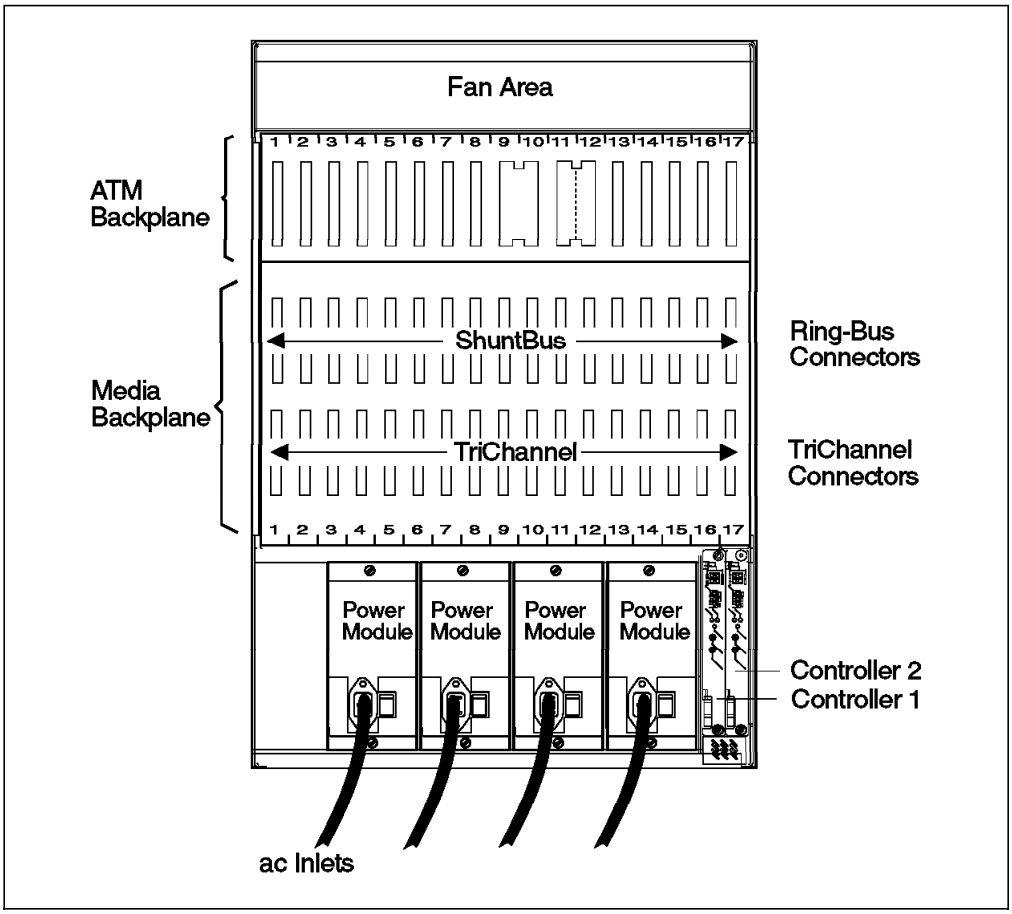



*Figure 7. 8260 Backplanes*

The ATM backplane has many similarities to the other backplanes that reside on the 8260:

- The backplane is totally passive. No active components reside on the backplane, so that extremely high reliability is achieved.
- It has female connectors. If a module is not inserted correctly, then the male pins will get bent or damaged. Since the connector on the backplane is

© Copyright IBM Corp. 1996

female, there are no pins on the backplane to get bent. It is far easier and less disruptive to replace or repair a module than a backplane.

- The ATM media modules have full-floating capabilities. These modules can be inserted into any of the slots 1 to 8 and 12 to 17.
- All ATM modules are hot-swappable. Any module can be removed or inserted without having to power down the hub.

Although similar to other backplanes on the 8260, the ATM backplane does differ in one key aspect, namely its topology. The Enhanced TriChannel′s wiring topology is a bus configuration and the ShuntBus uses a ring configuration. The ATM backplane, however, is wired in a star configuration. Each ATM media module has a dedicated set of connections to positions 9/10 and 11/12 where the ATM Control Point and Switch (A-CPSW) module may be located. This set of dedicated connections constitute a star wiring topology in which the ATM media modules are the tips of the star and the A-CPSW is in the center.

### 2.1.1 Model A10 and A17 Backplanes

As described in Chapter 1, “General Introduction to the 8260 and the ATM Implementation” on page 1, there are two different models of the 8260 that support ATM, namely the A10 and A17. The ATM backplanes on these two models are essentially the same but they do differ in some key aspects:

- Size

  The A10 has ten slots while the A17 has seventeen slots.

- A-CPSW Support

  The A17 supports the installation of the A-CPSW in slots 9 and 10 or 11 and 12. The A10 only has ten slots so it only supports the A-CPSW in slots 9 and 10. This being the case, the A10 supports the installation of only one A-CPSW in slots 9 and 10.

- Packet Switching bus

  The Packet Switching bus is not implemented on the A10 and A17 8260 Hubs.

**Note**

Since the Model A10 and the Model A17 are very similar, all further references in this book will refer to the A17 only. Unless explicitly specified, all information will be applicable to the Model A10 as well.

### 2.1.2 Upgrading an Existing 8260 Model 017

As mentioned earlier in the book, the 8260 Model 017 does not have an ATM backplane installed but can be upgraded with one. The design of the 8260 is such that the upgrade procedure is relatively straightforward, but it should only be performed by IBM service engineers. The following text describes briefly the process of upgrading the 8260. However, it is not intended to replace the installation instructions, but to give you an idea of how straightforward the process is.

The upgrade is performed from the back of the hub after all power cords are removed and the individual power supply switches are turned off. On the back of the hub there is an aluminum cover. The screws securing this cover are removed to release it, as shown in Figure 8 on page 23. The three cooling fans

are attached to this cover and have a set of wires that connect each fan to the hub. The three plugs connecting these wires are disconnected, and the entire back cover with the three fans can then be moved out of the way.

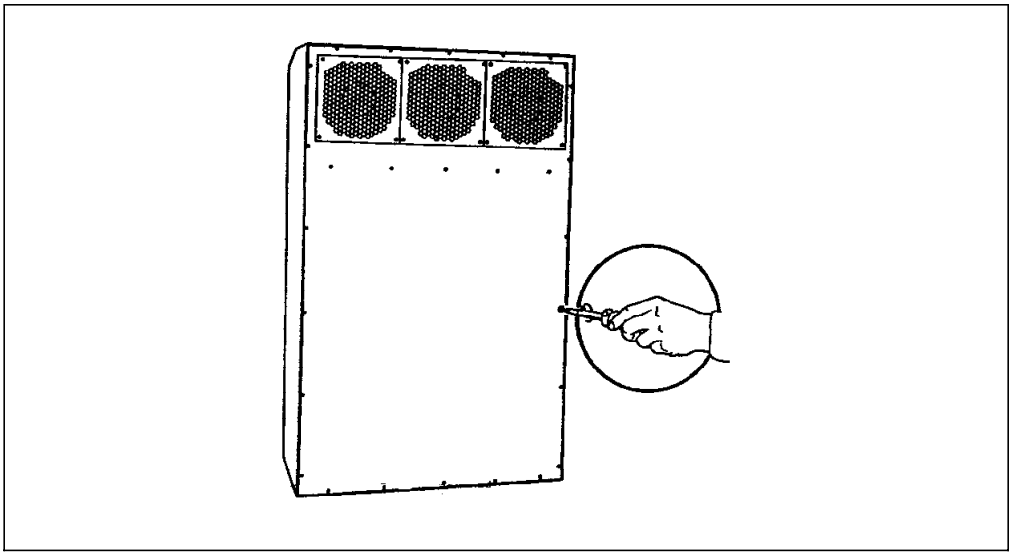


*Figure 8. Removing the Rear Aluminum Cover*

The three fan sensors can now be seen mounted together on a metal strip. The sensors are removed by removing five screws that secure the metal strip revealing an aluminum plate masking the location of the ATM backplane. This plate is removed by taking out the screws that hold it in place as seen in Figure 9.

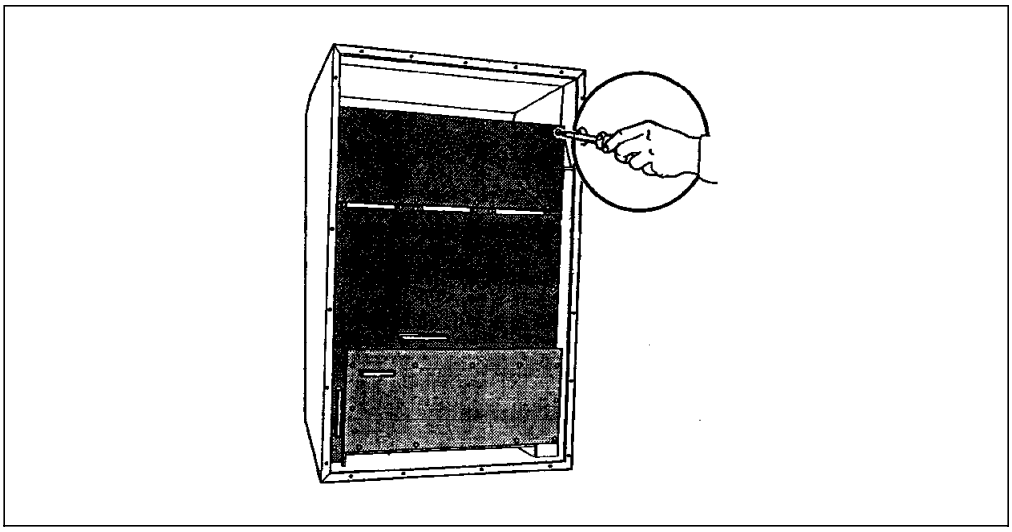


*Figure 9. Removing the Aluminum Plate (Dummy ATM Backplane)*

The new ATM backplane is then installed with the screws supplied, but the screws should not be tightened. Two ATM modules are placed in slot 1 and slot 17 to ensure proper alignment of the backplane. The screws to hold the ATM backplane are tightened after which three small ribbon cables are attached to connect the ATM backplane to the other backplanes. The upgrade is now almost complete and all that remains is to put back all the other components in reverse order: the fan sensors are screwed back in, the three fan plugs are reconnected and the rear aluminum cover is screwed onto the back again.

As you can see, the entire process is very simple. No modules need to be removed, except those that may be in slots 1 and 17, and the only tool that is needed is a screwdriver. The upgrade should be complete in 30 minutes although a 40-50 minute allowance should be planned.

**Note**

If there are 8250 modules installed in your 8260, they are normally installed in the rightmost slots (16, 15, etc.) of the 8260 along with the right boundary filler adapter, which is installed in slot 17. Removal of the right boundary adapter to allow the installation of an ATM module in slot 17 (for alignment purposes) requires the removal of all the other 8250 filter plates.

## 2.2 Backplane Connection of ATM Modules

The ATM modules have two connectors, one that inserts into the ATM backplane, and another one that inserts into the Enhanced TriChannel. On the Enhanced TriChannel, the ATM modules use the serial control interface (SCI) management bus to transmit module and port configuration data. The controller module uses the SCI to gather vital product data (VPD) from the ATM modules and to get power and cooling status. The controller module sends the VPD through SCI to a DMM (if installed) so that the information can be displayed. The ATM modules are also connected to the Common Signal bus in the Enhanced TriChannel from which every 8260 module gets power and clocking signals.

The 8260 LAN modules connect only to the ShuntBus and/or Enhanced TriChannel for passing network traffic. Like the ATM modules, all the 8250 and 8260 LAN modules connect to the SCI management bus to send VPD to the controller module. Unlike the ATM modules, the DMM and MAC daughter cards can also be connected to the MLAN management bus for passing network statistics and IP traffic to and from the 8260 LAN modules. Figure 10 on page 25 illustrates the relationship between the various buses on the 8260 backplane and the 8260/8250 modules installed in the 8260.

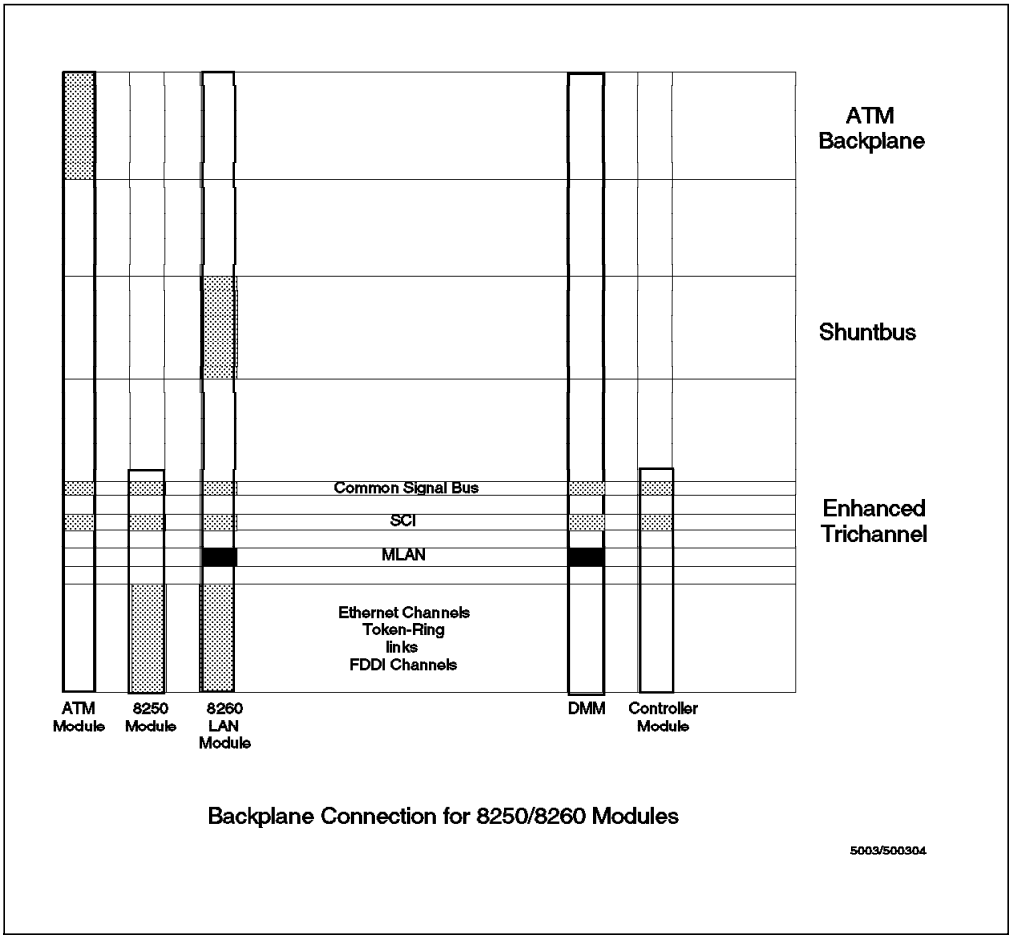



*Figure 10. Backplane Connection for 8250/8260 Modules*

Currently, all the ATM modules are connected to the ATM backplane and the SCI management bus, and cannot make use of the ShuntBus or the data section of the Enhanced TriChannel. For example, the 8260 ATM LAN Bridge module provides LAN connections through external attachment and does not support connection to the ShuntBus or Enhanced TriChannel for LAN connections.

## 2.3 Star Wiring on the 8260

The ATM backplane has a star-wired topology as opposed to a bus or ring topology found on the Enhanced TriChannel and the ShuntBus. Each of the ATM media modules has a dedicated set of connections to the A-CPSW module. This set of dedicated connections constitute a wiring star, in which the ATM media modules are at the tips and the A-CPSW is at the center.

In the 8260, the A-CPSW can reside in slots 9 and 10 or slots 11 and 12, which means two star-wired arrangements are possible. The star topology emanating from slots 9 and 10 is called Star-A and the star topology emanating from slots 10 and 11 is called Star-B. When two A-CPSW modules are used, then this is called a dual-star configuration.

**Note**

To be able to support two A-CPSWs in an 8260 (redundancy mode), you need the A-CPSW FC5100.

### 2.3.1 Switching Star-A

In a Star-A configuration slots 1 to 8 are all individually wired into slot 9, and slots 12 to 17 are all individually wired into slot 10 as shown in Figure 11. The A-CPSW resides in slots 9 and 10 which means there is a point-to-point connection from the A-CPSW module to every other slot in the 8260 except slot 11. Therefore, a maximum of 14 ATM media modules can be installed in an 8260 when the A-CPSW module is installed in slots 9 and 10.

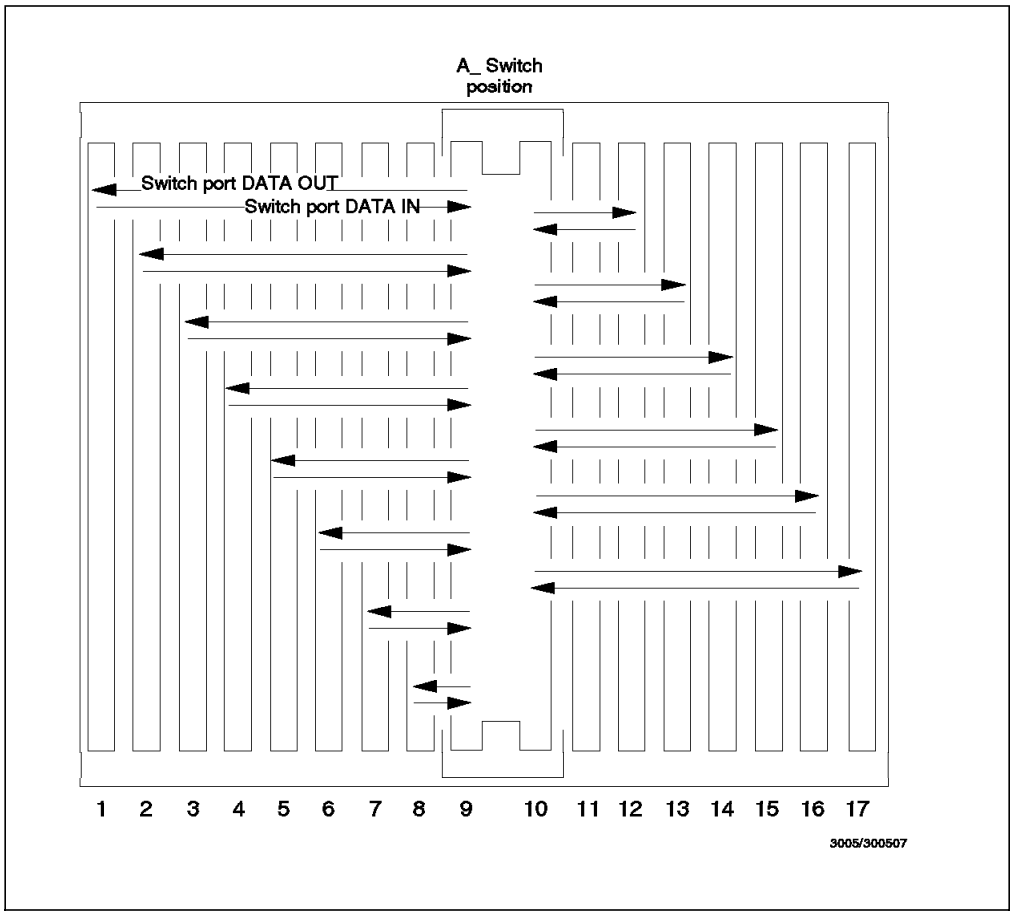



*Figure 11. 8260-A17 Switching Star-A*

You will notice that slot 11 is not wired into the star. Therefore, when using a Star-A configuration, slot 11 should be used for one of the non-ATM 8260 modules (for example, DMM) in order to maximize the ATM module capacity of the hub.

### 2.3.2 Switching Star-B

In a Star-B configuration slots 1 to 8 are all individually wired into slot 11, and slots 13 to 17 are all individually wired into slot 12 as shown in Figure 12. The A-CPSW resides in slots 11 and 12 which means there is a point-to-point connection from the A-CPSW module to every other slot in the 8260 except slots 9 and 10. Therefore, a maximum of 13 ATM media modules can be installed in an 8260 when the A-CPSW module is installed in slots 11 and 12.

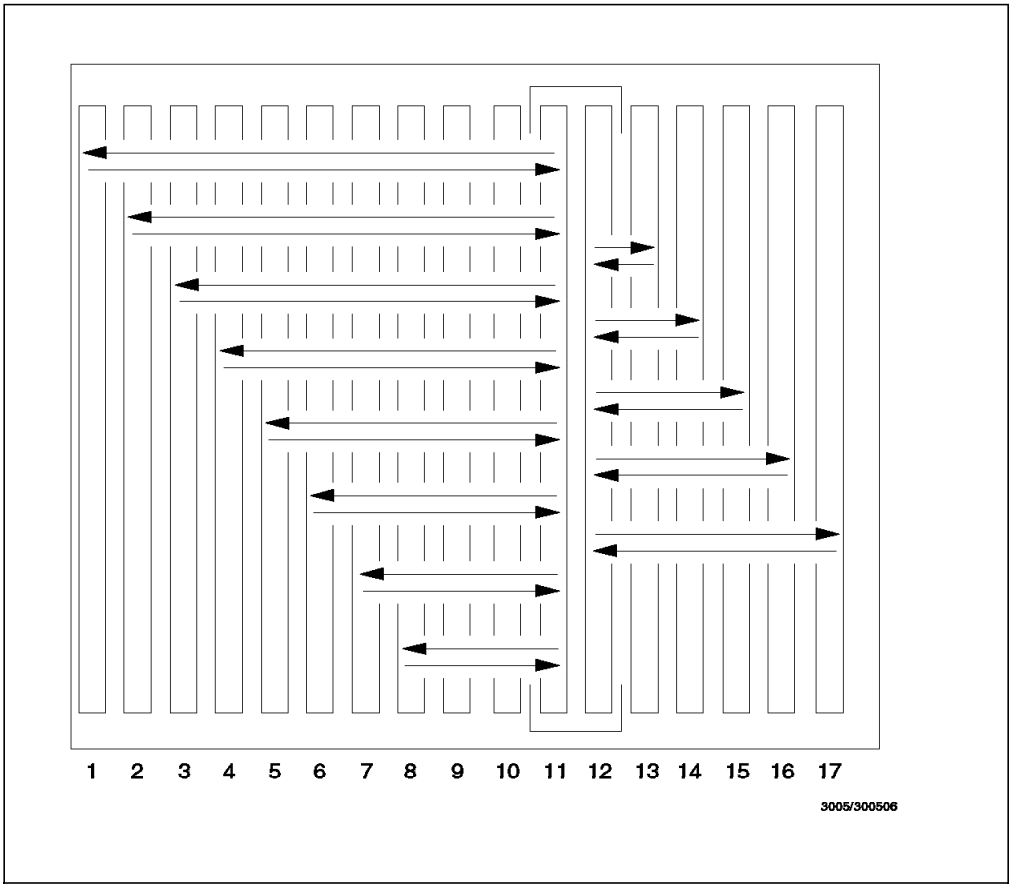



*Figure 12. 8260-A17 Switching Star-B*

You will notice that slots 9 and 10 are not wired into Star-B. Therefore, when using a Star-B configuration in a hub containing non-ATM 8260 modules (for example, DMM, slots 9 and 10 should be occupied with these modules in order to maximize the ATM module capacity in the hub.

### 2.3.3 Dual-Star Configuration

In this configuration Star-A and Star-B are used. Two A-CPSW modules are installed in slots 9 and 10 and slots 11 and 12. In this configuration slots 1 to 8 and slots 13 to 17 are all individually wired to both A-CPSW modules as per the Star-A and Star-B arrangement. Also, there is a *serial interface* on the ATM backplane that connects the two A-CPSW modules together. This interface is used for information exchange between the two A-CPSW modules as described in the following text.

In a dual-star configuration, one of the A-CPSW modules will be the *active* and the other the *backup*.

The following is the procedure used to determine which A-CPSW becomes the active and which one becomes the backup:

- If you install a second A-CPSW module in an 8260 that already has an A-CPSW installed and operating, the new A-CPSW will become the backup A-CPSW.
- If there are two A-CPSW modules installed in an 8260 when the hub is powered on (or reset), the active and the backup will be determined based on a user defined priority. If both A-CPSW modules have the same priority, the A-CPSW module in slots 9 and 10 has priority over the A-CPSW module in slots 11 and 12.

When two A-CPSW modules are installed in an 8260, the active A-CPSW module is performing the control point and switching function while the backup A-CPSW is a passive standby. The method used to support a backup A-CPSW is via a *mirroring* technique initiated from the active A-CPSW to the backup A-CPSW. The main way to initiate the mirroring is to issue a save command at the active A-CPSW. When this occurs, a table is sent from the active to the backup via the serial interface that connects the two A-CPSW modules.

The table sent from the active to the backup contains the following information:

- Terminal Community names
- Device Inventory
- Logical links/static routes
- Module port clock
- PVCs
- End System Identifiers (ESI)
- LECS configurations

The method used by the active A-CPSW to determine if the backup A-CPSW is ready for takeover is to send a poll every second to the backup A-CPSW module. When the backup A-CPSW receives a poll from the active A-CPSW, it returns a *table correlator* to the active A-CPSW. This will be used by the active A-CPSW to determine if the backup A-CPSW has the current level of the backup table. If the correlator indicates that the information in the backup is not current, the active A-CPSW will send a copy of the current table to the backup A-CPSW.

The active A-CPSW, also, will periodically ask the backup A-CPSW to perform diagnostics, thus insuring that the 8260 has a working backup A-CPSW.

**Note**

Although the serial interface between the active and backup A-CPSW is over the ATM backplane, it is not an ATM interface, and the design of the 8260 ensures that the mirroring information exchanged over this interface has a lower priority than the ATM cells sent over the ATM interface. In this fashion, the ATM call setup is not impacted by the mirroring traffic.

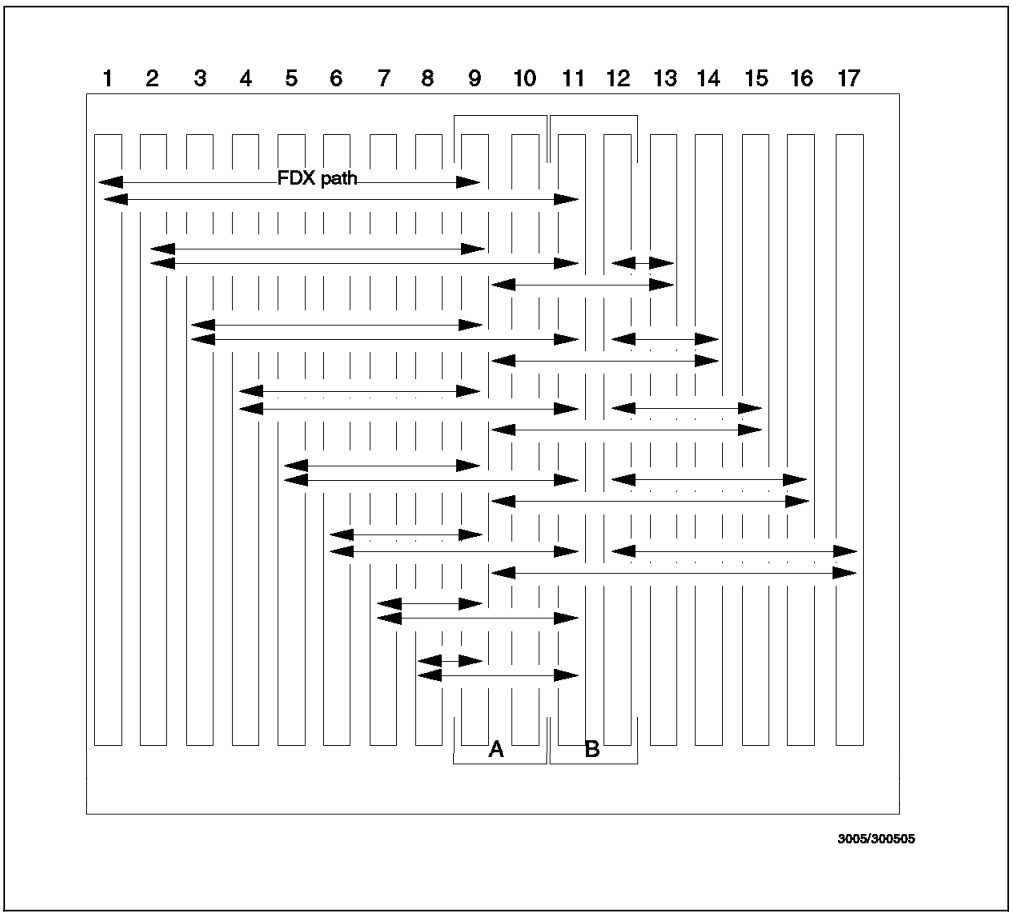



*Figure 13. 8260-A17 Dual-Star Configuration*

## 2.4 Pin Description

The ATM backplane is composed of 300 female pins, arranged in a matrix form, with 60 rows and 5 columns. The ATM modules only use the top 24 rows (120 pins) for passing network traffic and for clocking and control signals between the ATM media modules and A-CPSW. The other 36 rows (180 pins) are reserved for a high-speed packet bus.

**Note**

At the time of this writing, IBM is not offering modules that can take advantage of the packet bus.

In the ATM backplane, from each of the slots 1 to 8 and 13 to 17, 32 pins are available for passing network traffic between the media modules and the switch inside the A-CPSW. 16 of those pins are connected to slots 9 and 10 (Switching Star-A), and the other 16 pins to slots 11 and 12 (Switching Star-B).

## 2.5 ATM Subsystem Bandwidth

The backplane bandwidth calculation involves a number of variables and components. One of the critical components of the calculation is the switch on the A-CPSW module. This section briefly describes the switch and explains how the bandwidth calculation is derived.

### 2.5.1 ATM Switch

All the switching functions for the 8260 are performed on a switch in the A-CPSW module. This is an IBM-developed switch called the IBM switch-on-a chip, since it is an ATM switch contained within a single integrated circuit. The characteristics of IBM switch-on-a-chip include:

- 16 input ports and 16 output ports.
- 8-bit parallel connections on each input and output port data path. This results in a total of 472 pins on the IBM switch-on-a-chip.
- Each port can operate at speeds up to 256 Mbps.

  **Note:** The 256-Mbps data rate assumes a clock rate of 32 MHz, which is implemented in the current releases of the 8260 A-CPSW. However, the architecture of the IBM switch-on-a-chip allows the products to use clock rates of up to 50 MHz (20ns-cycle) resulting in a per-port throughput of 256 Mbps.
- Each port can run at full 256 Mbps simultaneously with the other ports resulting in a 16-way nonblocking switch. This will result in a total throughput of 8.2 Gbps. However, because of the provision to have a Redundant A-CPSW module, the number of ports on the switch have been limited to 14, providing a capacity of (14x2x256) = 7.2 Gbps.
- The switch-on-a-chip is a self routing cell switch *not* an ATM switch per se. That is, the switch-on-a-chip does not understand the ATM header structure or meaning; it treats the ATM header as just plain data. It can switch data blocks of any length from 14 to 64 bytes. Implications of this are discussed in 2.5.2, “Bandwidth Calculation” on page 31.
- Latency of around one microsecond (assuming no queueing).
- Built-in multicast and broadcast.
- 2.4 million transistors built on a 15 mm square chip using 0.7-µ CMOS technology.
- The design of the IBM switch-on-a-chip allows for the interconnection of multiple chips to increase the number of ports or support higher port speeds than is possible with a single chip. This would allow IBM to be able to support higher port speeds such as 622-Mbps ATM connections within the 8260 should there be a need for such connections.
- Transit delay (latency per port) = 33 microsecond
- Transit delay under heavy load = 40 microsecond
- Variation in cell interarrival time = 1 microsecond

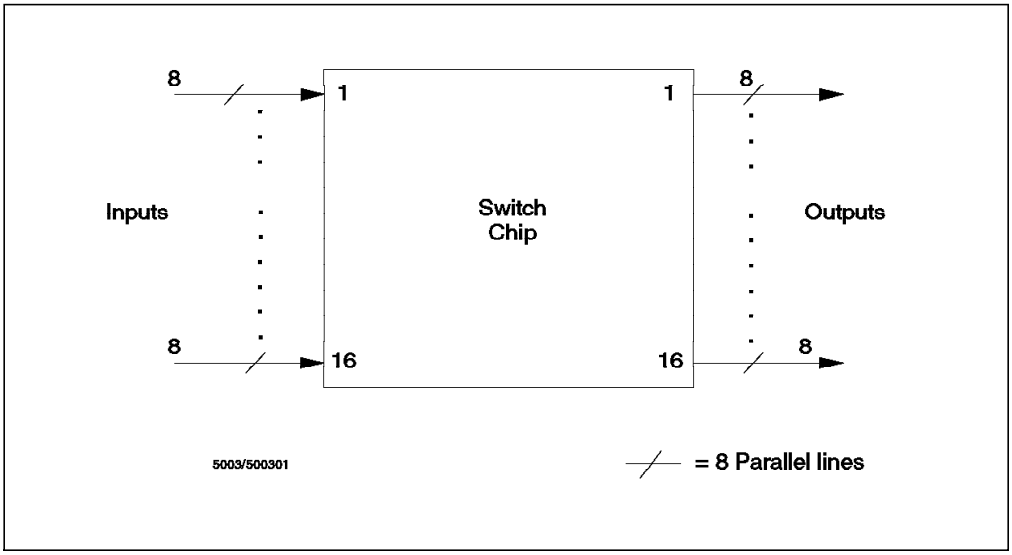



*Figure 14. IBM Switch-on-a-Chip*

### 2.5.2 Bandwidth Calculation

The A-CPSW module utilizes an IBM switch-on-a-chip operating at a clock rate of 32 MHz. This means the speed at which each input and output port operates is 256 Mbps (32000000 x 8bits). This figure for the speed of each port does not correlate to the actual throughput of ATM cells, because the IBM switch-on-a-chip switches data blocks of 64 bytes using an internal cell format. Before any ATM media module sends the ATM cells into the switch, each one of the 53-byte ATM cells goes through a process in which the last byte of the 5-byte ATM header is stripped, and an 8-byte internal routing header and a 4-byte trailer is added. This results is converting a 53-byte ATM cell into a 64-byte internal cell, as shown in Figure 15.

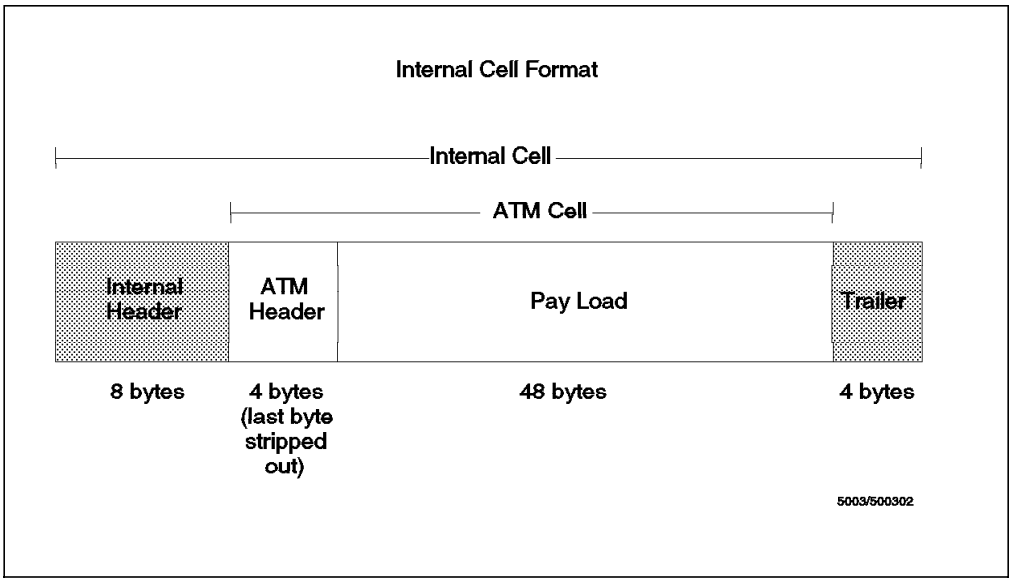



*Figure 15. Internal Cell Format*

As previously discussed, each of the ports in the IBM switch-on-a-chip operates at 256 Mbps. Each ATM media module is connected to an input port and output port of the switch, however, the actual ATM throughput of each of these ports is

212 Mbps (53/64 x 256 Mbps). This is because of the internal cell format, which carries a 53-byte ATM cell within a 64-byte internal cell. Since each ATM media module is connected to an input and output port of the switch (that is, it operates full-duplex), the throughput of each module is 424 Mbps (212 Mbps x 2). Since you can have up to 14 ATM media modules inside an 8260, the aggregate throughput of the ATM backplane is 5.936 Gbps (424 Mbps x 14).

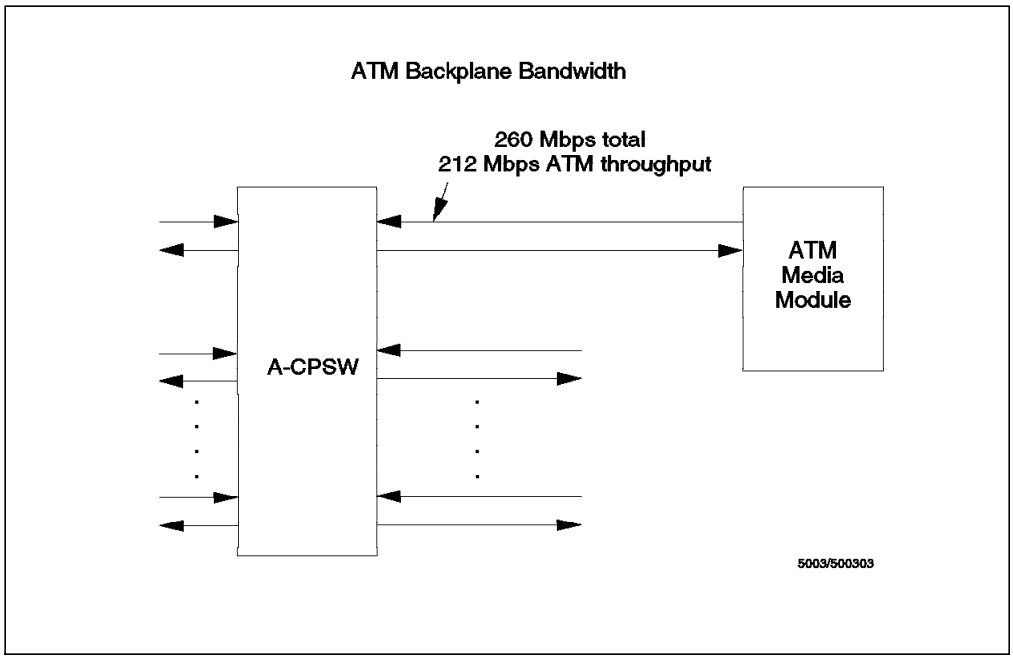



*Figure 16. ATM Backplane Bandwidth*

You may have noticed that the throughputs between the media modules and the switch (212 Mbps) is smaller than when you sum up the capacity of all the ports on a single module. For example, the aggregate port speed on a 4-port 100-Mbps ATM concentrator module is 400 Mbps. The actual maximum speed into the backplane is still only going to be 212 Mbps even though the ports have a potential of delivering 400 Mbps. A further explanation of this can be found in Appendix A, “Introduction to ATM and Campus ATM Terminology” on page 253.

You must remember that the backplane bandwidth is dependent on the speed of the switch in the current implementation. However, the backplane itself can support higher speeds than the current implementation. This means that if a new version of the A-CPSW was released that implemented a faster switch, the 8260 ATM backplane will be able to support an even higher aggregate throughput than 5.936 Gbps.

# Chapter 3. IBM 8260 ATM Modules

This chapter describes the various functions and features that are available on 8260 ATM modules.

## 3.1 Architecture of the 8260 ATM Subsystem

The current ATM modules for the IBM 8260 are:

- ATM Control Point and Switch (A-CPSW) module
- ATM 4-Port 100-Mbps (A4-FB100) module
- ATM 155-Mbps Flexible Concentration (ATMflex) module
- 8260 ATM TR/Ethernet LAN Bridge (8281) module
- ATM 12-Port 25-Mbps Concentration module

These modules are designed to implement ATM switching to the latest standards available from the ATM Forum and ITU. In a number of cases these standards have been developed and published. In other cases there is only a draft standard or indeed one may be some way off in the future. In the latter case, the 8260 uses the draft standards, or in some cases, proprietary techniques.

This changing environment, where standards are being updated or just taking effect, must be taken into account in the design of any ATM switch if it is to have an acceptable lifetime. This is certainly the case in the 8260 implementation where care has been taken to ensure that all modules are easily upgradable to allow the future standards to be implemented with minimal impact and to protect the customer investment.

This upgradeability has been achieved in two main ways:

1. Functions carried out by hardware for maximum performance are implemented using field programmable gate arrays (FPGAs). These programmable devices can be upgraded in the field if there is a need to modify the hardware driven functions.
2. All key functions are controlled by software; the code that can easily be upgraded.

Whether it is an FPGA function or operational code that has to be upgraded, the adding/updating function is very straightforward and can be carried out in just a few minutes with limited network disruption. This process is described in detail in 4.15, “Upgrading Microcode” on page 179.

## 3.2 ATM Control Point and Switch Module

The ATM Control Point and Switch (A-CPSW) module consists of the following two cards, which are packaged into a two-slot module:

1. The base card (ATM switch fabric)

   The ATM switch fabric switches cells between the ATM endsystems attached to the 8260 via ATM media modules. It is physically located on the left side of the A-CPSW module when looking at the module as it would normally reside in the 8260.

© Copyright IBM Corp. 1996

The switch fabric implemented in the A-CPSW uses the IBM switch-on-a-chip which is described in 2.5.1, “ATM Switch” on page 30.

**Note:** The switch-on-a-chip provides 16 ports, but because of the provision to have a redundant A-CPSW module in the 8260, the number of ports on the switch has been limited to 14, thus providing you with the ability to install up to a maximum of 14 ATM media modules in a single 8260 Model A17 (8 ATM modules in 8260 Model A10).

2. The control point card

   This card houses the processors where the Control Program resides. The Control Program performs the functions associated with the establishment and management of ATM circuits. It is physically located on the right side of the A-CPSW module when looking at the module as it would normally reside in the 8260.

   The control point card houses a RISC MC 68EC040 and an MC 68EN360 processor in companion mode for the control point functions, and ASIC/VLSI processors in the switch and media modules for the ATM switching functions.

   An ATM campus network can comprise many 8260s, each with its own A-CPSW module. This means that the control point functions of the ATM network are distributed which will give the ATM 8260 networks high availability and scalability. Also, you may install a second A-CPSW module in each 8260 Model A17 to be used as a backup in case of the failure of the primary A-CPSW module.

The following memories are installed in the A-CPSW module:

- 32 KB of NVRAM to store configuration data
- 4 MB of flash EEPROM for 2 x control point (CP) microcode load modules
- 8 MB of dynamic RAM for buffers and control blocks

**Note**

The previous A-CPSW modules with 8 MB of memory are withdrawn and replaced by a new A-CPSW module. The memory capacity on the new A-CPSW module has been extended from 8 to 16 MB of DRAM compared to the previous module. Customers who already have an A-CPSW module and wish to benefit from the new functionality offered by the future microcodes must increase the DRAM capacity of their module to 16 MB via an MES Upgrade. This MES is customer installable.

The switch redundancy and the chassis monitoring functions supported by microcode level V2.1.0 run only on the new A-CPSW module (feature number 5100), which, in addition to 16 MB of memory, includes other hardware enhancements that are not available in the earlier A-CPSW modules. Therefore, customers who upgrade their existing A-CPSW module to 16 MB of memory will not be able to take advantage of the switch redundancy and the chassis monitoring. To do so, they must replace their A-CPSW modules with the new A-CPSW modules.

### 3.2.1 A-CPSW Front Panel

There are a number of ports and LEDs on the front panel of the A-CPSW:

- Seven LEDs. These LEDs give the operational status of the A-CPSW.
- ATM Reset Button. This is a recessed button that, when depressed, will reset the ATM subsystem. This includes all ATM media modules that may be connected to the hub at the time.
- 9-pin RS-232 Console Port. This port is used to attach an ASCII terminal to the A-CPSW to access the management functions.
- 9-pin RS-232 Auxiliary Port. This port is reserved for IBM service engineers.

Figure 17 shows the front view of the A-CPSW module.

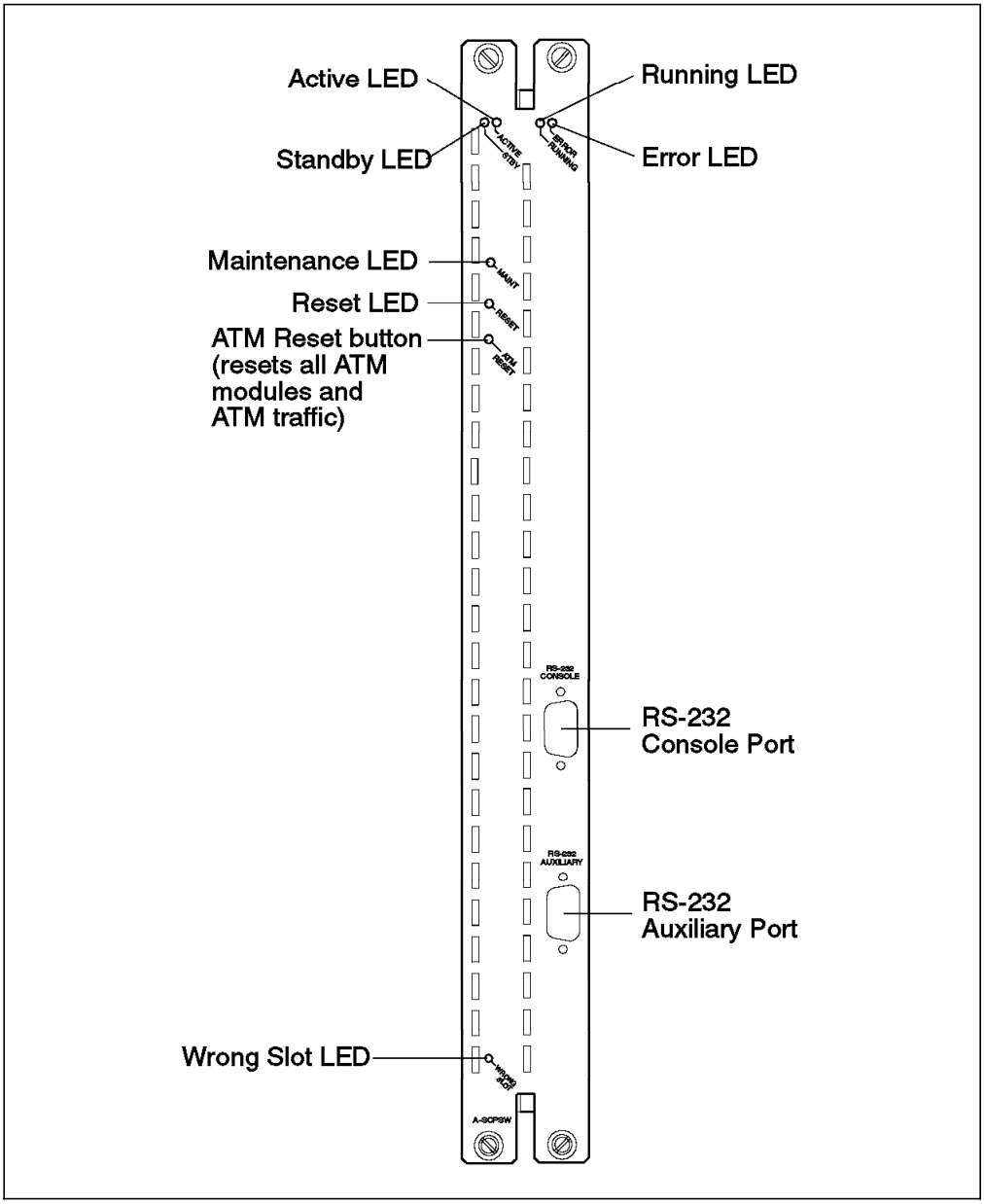



*Figure 17. Front View of the A-CPSW Module*

### 3.2.1.1 A-CPSW LEDs

Table 2 describes the meaning of each LED on the front panel of the A-CPSW.

*Table 2. Meaning of the A-CPSW LEDs*

| LED Name | Color | State | Meaning |
|---|---|---|---|
| Active | Yellow | OFF | A-CPSW module is not able to control ATM traffic and ATM media modules. |
| | | ON | A-CPSW module is able to control ATM traffic and media modules. |
| Standby (STDBY) | Yellow | OFF | Either a second A-CPSW module is not installed or, if a second A-CPSW is installed, it is not active. |
| | | ON | The second A-CPSW module is installed and active. |
| Running | Yellow | OFF | A-CPSW software is not running. The Error LED lights up. |
| | | ON | The A-CPSW software is started and running properly. |
| Error | Red | OFF | A-CPSW is functioning properly. |
| | | ON | A-CPSW module is not operational due to an error. |
| Maintenance (MAINT) | Yellow | OFF | A-CPSW module is functioning properly. |
| | | ON | Maintenance mode is active. |
| Reset | Yellow | OFF | A-CPSW module is functioning properly. |
| | | ON | A-CPSW and ATM media modules are being reset. |
| Wrong Slot | Yellow | OFF | Normal operation. A-CPSW module is installed in slots 9 and 10 (or in future releases in slots 11 and 12). |
| | | ON | A-CPSW module is not installed in the correct slots. |

### 3.2.1.2 Reset Button

This button is recessed to prevent accidental resets. When pressed, the A-CPSW and all ATM media modules will be reset. This means all traffic will be stopped, all connections will be terminated and any unsaved ATM configuration settings are lost. Pressing this button is like issuing the RESET ATM_SUBSYSTEM FORCE command. To ensure that no configuration changes are lost, a SAVE command should be issued before pressing the Reset button.

Non-ATM modules are not affected by this Reset button.

#### 3.2.1.3 Console Port

The console port is a DTE male DB9 connector that allows you to connect an ASCII terminal (local or modem-attached) to the A-CPSW. From here you can view and modify the configuration of the A-CPSW and ATM media modules.

Non-ATM ports cannot be configured from the console port. This is done via the DMM. These include:

- Power management subsystem
- Hub inventory information
- Ethernet, token-ring or FDDI modules

The factory default settings for the console port are as follows:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit

These settings can be changed once a terminal with the above configuration is connected by using the SET TERMINAL command.

The ASCII terminal is also used to download new versions of software to the A-CPSW or the media modules using out-of-band download.

The following table describes the pin assignments for the RS-232 Console port.

*Table 3. Console Port Pin Assignments*

| Pin Number | Signal Name |
|---|---|
| 1 | Carrier Detect (CD) |
| 2 | Receive Data (RX) |
| 3 | Transmit Data (TX) |
| 4 | Data Terminal Ready (DTR) |
| 5 | Signal Ground (GND) |
| 6 | Data Set Ready (DSR) |
| 7 | Request To Send (RTS) |
| 8 | Clear To Send (CTS) |
| 9 | Not Used |

For information about the functions supported by the A-CPSW and also how to configure the A-CPSW to take advantage of these functions, please refer to Chapter 4, “ATM Control Point and Switch (A-CPSW) Module” on page 73.

## 3.3 ATM 4-Port 100-Mbps Concentration Module

The 8260 ATM 4-Port 100-Mbps (A4-FB100) module is a single-slot module that functions as part of the ATM subsystem in the 8260 Multiprotocol Intelligent Hub.

The main features of the module are:

- Four 100-Mbps ports using multimode fiber, and support of the transparent asynchronous transmitter-receiver (TAXI) interface to attach ATM devices.
- The physical interface is of optical fiber cable with a MIC duplex (as specified in ISO DIS 9314-3) or an SC connector. The type of connector provided by the module is determined at the time of ordering the module.
- Hot-pluggable in any slot in the 8260 hub, except for slots 9, 10 and 11, which are reserved for A-CPSW modules. Note that slot 12, which can be used by the A-CPSW, is also available for the A4-FB100 module if no A-CPSW module is installed in slots 11 and 12.
- This module provides four ports, each capable of supporting 100 Mbps (full-duplex). The traffic received from these ports will be all multiplexed into the backplane connection between the A4-FB100 module and the A-CPSW. Since the aggregate throughput of the backplane connection between the A4-FB100 module and the A-CPSW is 212 Mbps (full-duplex), as derived in 2.5.2, “Bandwidth Calculation” on page 31, the total traffic from all the ports cannot exceed 212 Mbps. For example, if one port is using the full bandwidth (100 Mbps), the other three ports use the remaining bandwidth (112 Mbps). A-CPSW will employ a flow control mechanism (and/or cell discards) to ensure that the total traffic from all four ports does not exceed 212 Mbps.
- Support for workstation connections up to 2 km from the 8260.
- Support for aconnection between two A4-FB100 ports up to 3 km.
- Up to 14 A4-FB100 modules can be used in the 8260 hub at the same time.
- Support of UNI (Version 3.0 and 3.1), SSI (switch-to-switch interface), and NNI (IISP) on each of the ports.
- Support of up to 992 SVCs (switched virtual circuits) per module. 32 of these SVCs are reserved for signalling purposes.

The A4-FB100 module has an interface to the ATM backplane which allows it to communicate with the A-CPSW module in the network and use its services to access the ATM network.

The A4-FB100 module processes the ATM cells by:

- Checking their validity
- Accessing the switching tables to locate the destination module
- Preparing the internal ATM cell format required by A-CPSW
- Sending cells to the A-CPSW module

These processes are explained in detail in 6.1.2, “CAP/CAD” on page 205.

### 3.3.1 A4-FB100 Front Panel

Figure 18 on page 39 shows the front view of the A4-FB100 module. As can be seen, this module provides LED indicators on the front panel that allow you to monitor the status of the module and the individual ports.

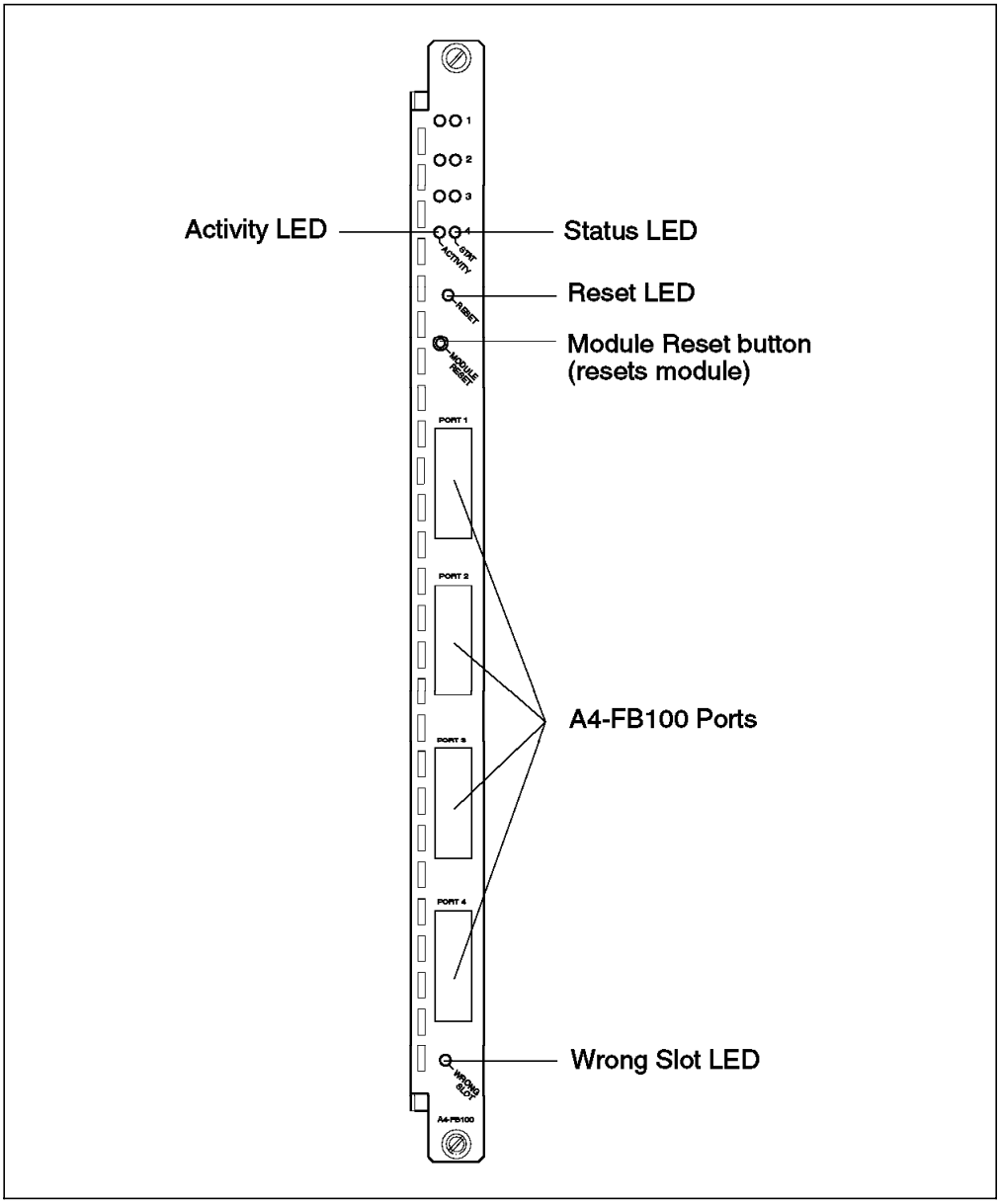



*Figure 18. Front View of A4-FB100 Module*

Table 4 describes the meaning of these LEDs.

*Table 4 (Page 1 of 2). A4-FB100 LED Descriptions*

| LED Name | Color | State | Description |
|---|---|---|---|
| Status | Green | OFF | Port is disabled. |
| | | ON | Port is enabled. |
| | | Blinking | No signal is detected on the port. |
| Activity | Yellow | OFF | No ATM data traffic is being transmitted. |
| | | ON | ATM cells are being transmitted. |

| Table 4 (Page 2 of 2). A4-FB100 LED Descriptions | | | |
|---|---|---|---|
| **LED Name** | **Color** | **State** | **Description** |
| Reset | Yellow | OFF | Normal operation. Module is not being reset. |
| | | ON | Module is being reset. Traffic is interrupted. |
| | | Blinking | A4-FB100 module is not receiving its clock because no A-CPSW module has been installed. |
| Wrong Slot | Yellow | OFF | Normal operation. A4-FB100 module is installed in the correct slot (1 to 8 or 12 to 17). |
| | | ON | A4-FB100 module is installed in an incorrect slot. |

The module reset button interrupts and resets the operation of the A4-FB100. All ATM data traffic and connections are stopped. The change in the status of the A4-FB100 module (from normal operation to Reset Status) is reported to the A-CPSW module.

### 3.3.2 Planning for Fiber Connections with the A4-FB100 Module

Some general guidelines you should follow when planning and setting an ATM network using fiber and A4-FB100 modules are as follows:

- IBM recommends using 62.5/125 micron multimode fiber cable.
- Calculate the power loss in the ATM fiber link and verify that it does not exceed the optical power budget. When calculating the power loss, you must take into account all sources of optical power loss such as connectors, splices, patch panels and type of cable used.

Table 5 shows the optical power budget for a UNI link between an A4-FB100 port and an ATM endsystem.

| Table 5. Optical Power Budget for A4-FB100 MIC Port-to-SC Device Connections | | | | |
|---|---|---|---|---|
| **Fiber Cable Size (microns)** | **Minimum Transmitted Power** | **Maximum Received Power** | **Optical Power Budget** | **Maximum Link Distance** |
| 62.5/125<br>NA 0.275 | -20 dB | -29 dB | 9 dB | 2 km<br>(1.24 mi) |

Table 6 shows the optical power budget of the A4-FB100 ports with MIC connectors when used to connect two 8260 ATM subsystems using SSI or NNI links.

| Table 6 (Page 1 of 2). Optical Power Budget for A4-FB100 MIC Port-to-Port Connections | | | | |
|---|---|---|---|---|
| **Fiber Cable Size (microns)** | **Minimum Transmitted Power** | **Maximum Received Power** | **Optical Power Budget** | **Maximum Link Distance** |
| 50/125<br>NA 0.20 | -20.5 dB | -33 dB | 12.5 dB | 2.5 km<br>(1.55 mi) |

| Table 6 (Page 2 of 2). Optical Power Budget for A4-FB100 MIC Port-to-Port Connections | | | | |
|---|---|---|---|---|
| **Fiber Cable Size (microns)** | **Minimum Transmitted Power** | **Maximum Received Power** | **Optical Power Budget** | **Maximum Link Distance** |
| 62.5/125<br>NA 0.275 | -18.5 dB | -33 dB | 14.5 dB | 3 km<br>(1.86 mi) |

Table 7 shows the optical power budget of the A4-FB100 ports with SC connectors when used to connect two 8260 ATM subsystems using SSI or NNI links.

| Table 7. Optical Power Budget for A4-FB100 SC Port-to-Port Connections | | | | |
|---|---|---|---|---|
| **Fiber Cable Size (microns)** | **Minimum Transmitted Power** | **Maximum Received Power** | **Optical Power Budget** | **Maximum Link Distance** |
| 50/125<br>NA 0.20 | -21 dB | -30 dB | 9 dB | 2 km<br>(1.24 mi) |
| 62.5/125<br>NA 0.275 | -19 dB | -30 dB | 11 dB | 2.2 km<br>(1.36 mi) |

## 3.4 ATM 155-Mbps Flexible Concentration Module

The ATM 155-Mbps Flexible Concentration (ATMflex) module is a single-slot concentrator module, which has the following characteristics:

- Support of two daughter boards, which will allow you to mix and match different media types. By taking advantage of this flexibility, you can create customized mixed-media solutions for your individual networking needs.

  The following daughter cards are available for the ATMflex module:

  – ATM 155-Mbps 1-port Fiber Multimode I/O Card

  – ATM 155-Mbps 1-port Fiber Monomode I/O Card

  – ATM 155-Mbps 1-port UTP/STP I/O Card

- The 155-Mbps multimode and monomode fiber daughter cards use an SC connector.
- The 155-Mbps UTP/STP daughter cards use an RJ-45 connector.
- The ATMflex module supports the following transmission procedures on a per-port basis:

  – SONET STS-3C

    Also known as *SONET-LITE*, this transmission protocol is compliant with ANSI synchronous transport optical networks (SONET), synchronous transport signal 3 (STS-3C) at a 155.520-Mbps line rate.

  – SDH-STM-1

    This transmission protocol is compliant with ITU synchronous digital hierarchy (SDH), synchronous transport module 1 (STM-1) at a 155.520-Mbps line rate. This interface is being tested around the world (except in the Northern Hemisphere) as a WAN interface. Please refer to

"SONET and SDH" on page 275 for more information on the SONET standard.

- Hot-pluggable in any slot in the 8260 hub, except for slots 9, 10 and 11, which are reserved for A-CPSW modules. Slot 12 is also reserved, but you can insert an ATMflex module in slot 12 if no A-CPSW module is installed in slots 11 and 12.
- This module provides two ports, each capable of supporting 155 Mbps (full-duplex). The traffic received from these ports will be all multiplexed into the backplane connection between the ATMflex module and the A-CPSW. Since the aggregate throughput of the backplane connection between the ATMflex module and the A-CPSW is 212 Mbps (full-duplex), as derived in 2.5.2, "Bandwidth Calculation" on page 31, the total traffic from the ports cannot exceed 212 Mbps. For example, if one port is using the full bandwidth (155 Mbps), the other port uses the remaining bandwidth (57 Mbps). A-CPSW will employ a flow control mechanism (and/or cell discards) to ensure that the total traffic from the four ports does not exceed 212 Mbps.
- Up to 14 ATMflex modules can be used in the 8260 hub at the same time.
- Supports UNI, SSI, NNI on each of the two ports.
- Supports up to 979 SVCs (switched virtual circuits) per module. 32 of these SVCs are reserved for signalling purposes.

### 3.4.1 ATMflex Traffic Management

The ATMflex module provides reserved bandwidth (RB) and available bit rate (ABR), the same as the A4-FB100 module, with the following difference:

- For RB traffic, the maximum bandwidth that can be reserved is eighty-five percent of the total throughput capacity, which is 155 Mbps at the ATMflex port interface and 212 Mbps at the A-CPSW interface.

Based on these total aggregate throughput capacities, bandwidth for ATMflex ports set with SSI interfaces (on the same module) is reserved as follows:

- If you are setting a single SSI link between two 8260s, the maximum bandwidth that can be reserved is 131 Mbps (85% of 155 Mbps).
- If a second SSI port is configured, the amount of reserved bandwidth for the first port will be recalculated and each SSI link will be allocated 90 Mbps. This is because the maximum bandwidth you can reserve for the module is 180 Mbps (85% of 212 Mbps). If the peak cell rate (PCR) in the first SSI port exceeds 90 Mbps, the command setting for the second SSI port will be rejected.

### 3.4.2 ATMflex Front Panel

The ATMflex front panel is shown in Figure 19 on page 43. As can be seen, this module provides LED indicators on the front panel that allow you to monitor the status of the module and the individual ports.

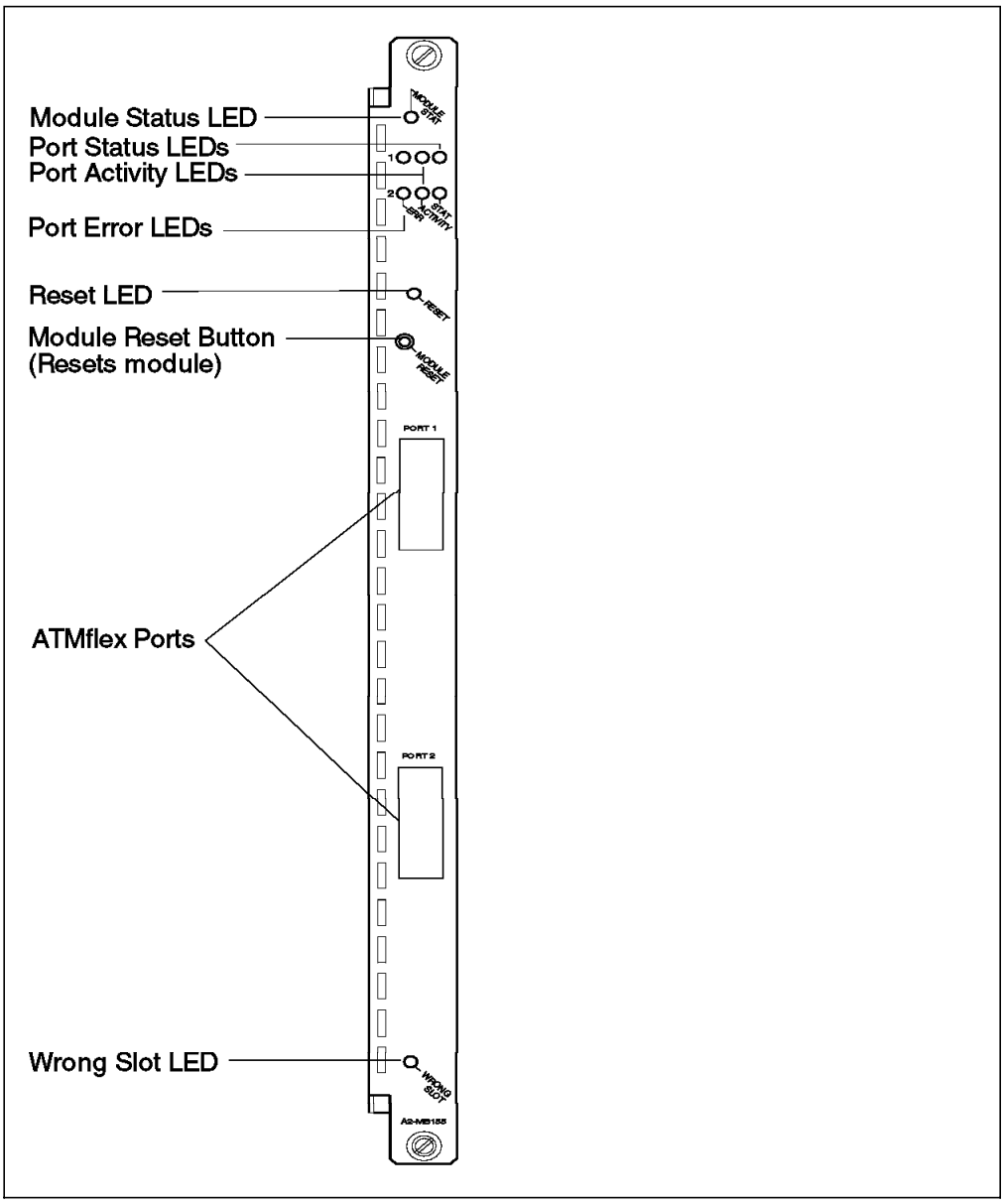



*Figure 19. Front View of ATMflex Module*

Table 8 describes the meaning of each ATMflex LED.

*Table 8 (Page 1 of 2). ATMflex LED Descriptions*

| LED Name | Color | State | Description |
|---|---|---|---|
| Module Status | Green | OFF | No power or isolated. |
| | | ON | Normal operation. |
| Port Error | Yellow | OFF | Normal operation. |
| | | ON | No signal or error condition detected. |
| Port Activity | Yellow | OFF | No ATM data is being transmitted. |
| | | ON | ATM cells are being transmitted. |

| Table 8 (Page 2 of 2). ATMflex LED Descriptions | | | |
|---|---|---|---|
| **LED Name** | **Color** | **State** | **Description** |
| Port Status | Green | OFF | Port is disabled. |
| | | ON | Port is enabled. |
| | | Blinking | Port is enabled, but either no cable is connected or the cable is damaged. |
| Reset | Yellow | OFF | Normal operation. Module is not being reset. |
| | | ON | Module is being reset; data traffic is interrupted. |
| | | Blinking | ATMflex module is not receiving its clock because no A-CPSW module has been installed. |
| Wrong Slot | Yellow | OFF | Normal operation. |
| | | ON | Module installed in wrong slot. |

The module reset button interrupts and resets the operation of the ATMflex module. All ATM data traffic and connections are stopped. The change in the status of the ATMflex module (from normal operation to Reset Status) is reported to the A-CPSW module.

### 3.4.3 Planning for Fiber Connections with the ATMflex Module

Some general guidelines you should follow when planning and setting an ATM network using fiber as the backbone medium are as follows:

- When using multimode fiber, IBM recommends using 62.5/125-micron fiber that conforms with the 10Base-F standard.
- When using monomode fiber, IBM recommends using 9-micron fiber that conforms with the 10Base-F standard.
- Calculate the power loss in the ATM fiber link and verify that it does not exceed the optical power budget. When calculating the power loss, you must take into account all sources of optical power loss such as connectors, splices, patch panels and type of cable used.

Table 9 shows the optical power budget for a UNI link between an ATMflex port and an endsystem.

| Table 9. Optical Power Budget for ATMflex Port-to-Device Connections | | | | |
|---|---|---|---|---|
| **Fiber Cable Size (microns)** | **Minimum Transmitted Power** | **Maximum Received Power** | **Optical Power Budget** | **Maximum Link Distance** |
| Multimode 50/125 micron NA 0.20 | -21 dB | -30 dB | 9 dB | 2 km (1.24 mi) |
| Multimode 62.5/125 micron NA 0.275 | -20 dB | -29 dB | 9 dB | 2 km (1.24 mi) |
| Monomode 9/125 micron | - | - | - | 20 km (12.4 mi) |

Table 10 on page 45 shows the optical power budget of the ATMflex ports when used to connect two 8260 ATM subsystems using SSI or NNI links.

*Table 10. Optical Power Budget for ATMflex Port-to-Port Connections*

| Fiber Cable Size (microns) | Minimum Transmitted Power | Maximum Received Power | Optical Power Budget | Maximum Link Distance |
|---|---|---|---|---|
| Multimode 50/125 micron NA 0.20 | -22.5 dB | -30 dB | 7.5 dB | 2 km (1.24 mi) |
| Multimode 62.5/125 NA 0.275 | -19 dB | -30 dB | 11 dB | 2.2 km (1.36 mi) |
| Monomode 9/125 | -15 dB | -32.5 dB | 17.5 dB | 20 km (12.4 mi) |

### 3.4.4 Assembling the Motherboard and Daughter Cards

Before installing the ATMflex module, you must first assemble it by installing the daughter cards in the motherboard. To assemble the ATMflex module, follow these steps:

1. Remove the two screws and faceplate that cover the ATMflex port on the front panel of the motherboard.
2. Align the screw holes of the daughter card with the small metal posts in the motherboard, as shown in Figure 20 on page 46.
3. Place two of the screws that come with the daughter card into the metal posts and tighten them with a screwdriver.
4. Place the remaining two screws into their holes on the front panel and tighten them with a screwdriver. This secures the daughter card on the motherboard.

**Note**

If you replace a daughter card after configuring the ATMflex module (see 5.2, “Configuring ATMflex Module” on page 196) and reinsert the module in the hub, the new card is automatically configured with the settings of the previous card.

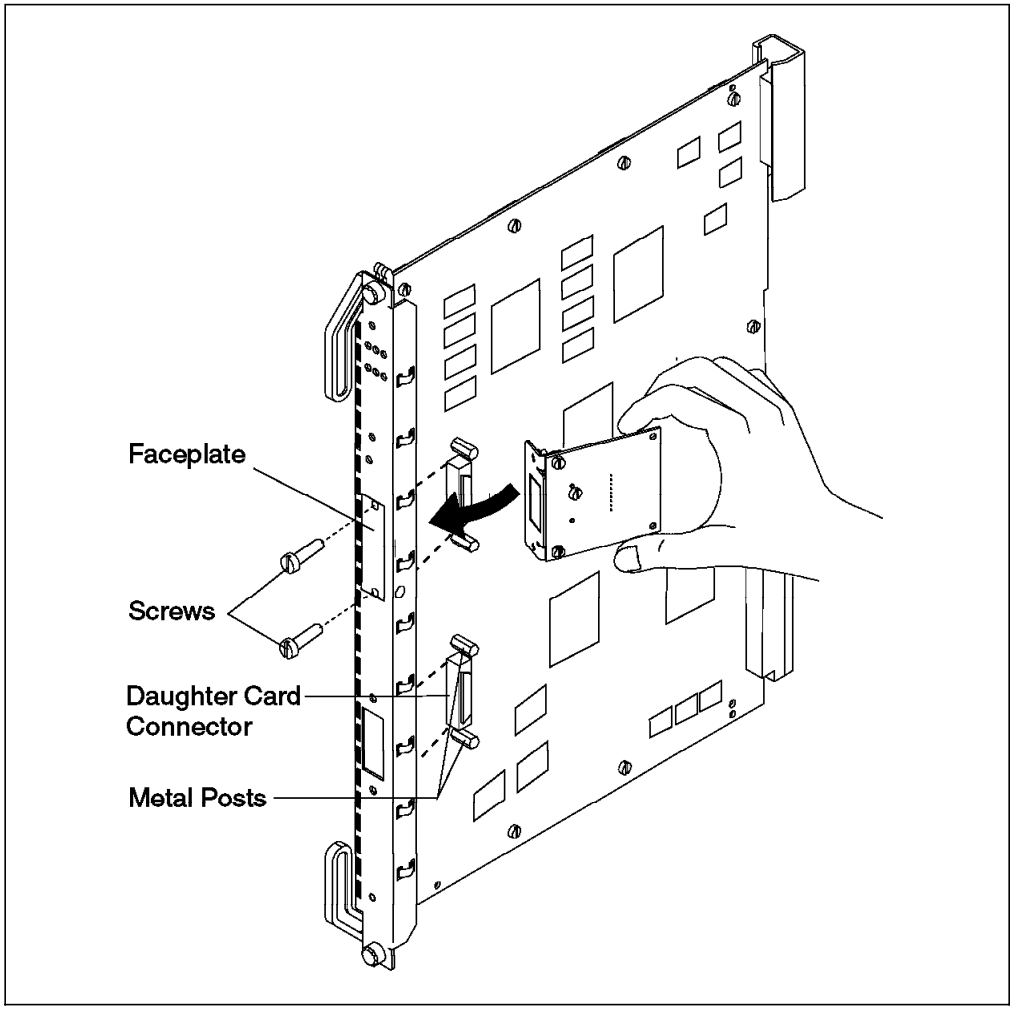



*Figure 20. Installing a Daughter Card on the ATMflex Motherboard*

## 3.5 ATM TR/Ethernet LAN Bridge Module

The ATM LAN Bridge module is a two-slot module that provides bridging functions between the ATM and LAN ports. The main features of this module are:

- Provides four ports for interconnection to LANs. These ports can be configured for either Ethernet (IEEE 802.3 and DIX V2) or token-ring. The hardware for supporting token-ring and Ethernet is the same; it is only via the configuration of the module that you specify the ports are connected to token-ring or Ethernet. Note that all the ports should be configured for the same type of LAN. This module does not allow the mixing of various LAN types.

  The LAN ports are external ports. This means that this module does not have an interface to the ShuntBus or TriChannel of the 8260 backplane for connection to the LANs on the 8260 backplane.

- Provides an ATM connection to the A-CPSW module. This connection is via the ATM backplane and supports ATM 3.0 UNI specifications.

  The ATM port can be configured to take part in emulated LAN compliant with IBM LAN emulation. The LE client can be configured to support either an

emulated token-ring or an emulated Ethernet. Note that the emulated LAN must be the same type of LAN as the one used on the four LAN ports.

- Provides bridging between the ATM port and the LAN ports, as well as bridging between the LAN ports.
- Supports source route bridging when the ports are configured to use token-ring.
- Supports transparent bridging when the ports are configured to use Ethernet.
- Supports 256 virtual circuits (VCs) over the ATM connection.
- The LE client in the ATM LAN bridge can establish a connection with a single LAN emulation server. This means that the bridge can only be a member of a single emulated LAN over ATM.
- Management of the ATM LAN Bridge module is done using SNMP. The SNMP agent within the module supports various MIBs, as described in 3.5.5, “Management and Configuration Support” on page 67.

  The SNMP agent in this module is accessible via IP over the ATM and LAN ports. Access over the ATM port requires IBM LAN emulation over ATM. This means that the ATM-attached stations using Classical IP over ATM (RFC 1577) cannot communicate with this module directly over ATM.

### 3.5.1 Front Panel of the ATM-LAN Bridge Module

As shown in Figure 21 on page 48, there are a number of LEDs and ports on the front panel of the ATM-LAN Bridge module. The following subsections describe the ports and LEDs.

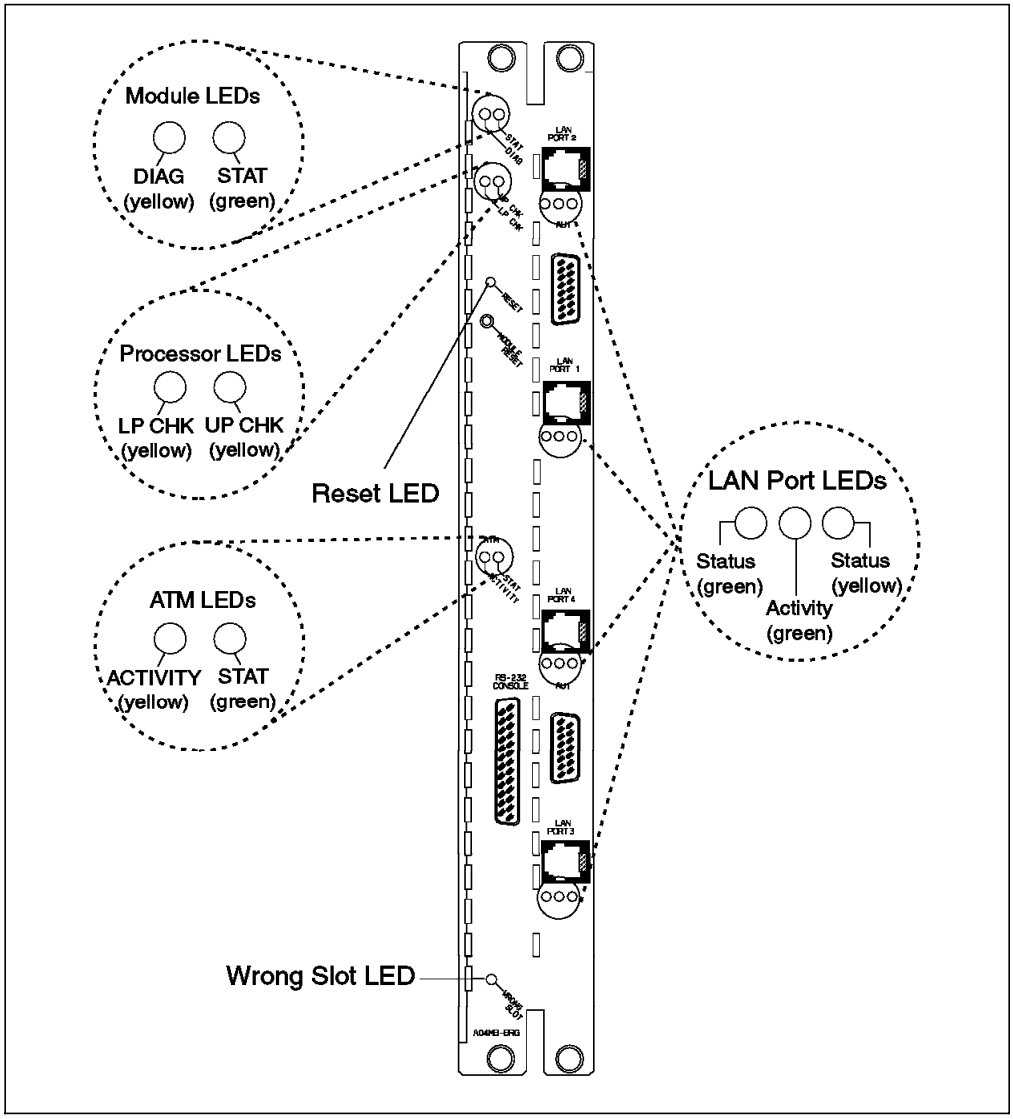



*Figure 21. Front View of the ATM-LAN Bridge Module*

### 3.5.1.1 8281 Module LEDS

The combination of these two LEDs indicates the operational and diagnostics status of the ATM-LAN bridge module, as shown in Table 11.

*Table 11 (Page 1 of 2). ATM-LAN Bridge Module - Module LEDs*

| **STAT LED** | **DIAG LED** | | |
|---|---|---|---|
| | **OFF** | **ON** | **Blinking** |
| **OFF** | No power to the module. | Diagnostics failed. | Not used. |
| **ON** | Module operational. | Initial reset state.<br>Processor failure.<br>Minimal Mode 1 | Not used. |

| Table 11 (Page 2 of 2). ATM-LAN Bridge Module - Module LEDs | | | |
|---|---|---|---|
| **STAT LED** | **DIAG LED** | | |
| **Blinking** | Diagnostics passed but configuration data still required to enable the module's operation. | Not used. | Diagnostics in process. |

### 3.5.1.2 LAN Port LEDs

Each LAN port has the following three LEDs:

- Left Status LED (green)
- Middle Activity LED (yellow)
- Right Status LED (green)

For Ethernet ports, the *activity* LED shows the rate of activity on the Ethernet LAN. The rate of blinking of this LED is proportional to the amount of traffic on the Ethernet segment.

For token-ring ports, the *activity* LED, combined with *status* LEDs, show the operation status of the port. Table 12 shows the possible combinations and their meaning.

**Note:** The two *status* LEDs display the same information.

| Table 12. ATM-LAN Bridge Module - LAN Port LEDs | | | |
|---|---|---|---|
| **Status LED** | **Activity LED** | | |
| | **OFF** | **ON** | **Blinking** |
| **OFF** | Port initialization in progress. | Port closed due to hardware error. Possible causes:<br>- Diagnostics failed<br>- Adapter check | Port closed due to LAN error. Possible causes:<br>- Wire fault<br>- Open failed |
| **ON** | Port open. | Not used. | Port open but LAN in error. Possible causes:<br>- Beaconing<br>- Hard error |
| **Blinking** | Port diagnostics passed but port is closed. Possible causes:<br>- Port disabled by the operator<br>- Auto removal | Not used. | Diagnostics has not begun. |

### 3.5.1.3 ATM Port LEDs

The ATM port has the following two LEDs:

- Status LED (green)
- Activity LED (yellow)

Table 13 on page 50 shows the meaning of these LEDs.

| Table 13. ATM-LAN Bridge Module - LAN Port LEDs | | | |
|---|---|---|---|
| **Status LED** | **Activity LED** | | |
| | **OFF** | ON | Blinking |
| **OFF** | Port disabled. | Fault on the ATM backplane. | Not used. |
| **ON** | Port enabled - no traffic. | Port enabled - heavy traffic. | Port enabled - normal traffic. |

### 3.5.1.4 Reset LED

Table 14 shows the meaning of this LED.

| Table 14. ATM-LAN Bridge Module - Reset LED | |
|---|---|
| **Reset LED** | **Meaning** |
| **ON** | Module is being reset. |
| **OFF** | Normal operation. |

### 3.5.1.5 Wrong Slot LED

Table 15 shows the meaning of this LED.

| Table 15. ATM-LAN Bridge Module - Wrong Slot LED | |
|---|---|
| **Wrong Slot LED** | **Meaning** |
| **ON** | Module is installed in an incorrect slot. |
| **OFF** | Module is installed in a correct slot. |

### 3.5.1.6 LAN Ports

The ATM-LAN Bridge module consists of the chassis equipped with four LAN ports. The LAN ports can be configured (via the configuration program) to be used either as token-ring or Ethernet ports. This means that whether you need a token-ring or Ethernet bridge, you receive the same hardware, but as part of the configuration, you will specify if the ports are to be used as Ethernet or token-ring.

Ports 1 and 3 on the module are always accessed via an RJ-45 connector for both token-ring and Ethernet.

Ports 2 and 4 can be accessed via the following connectors:

- An RJ-45 connector that can be used by token-ring or Ethernet
- An AUI connector that can be used by Ethernet only

### 3.5.1.7 ATM Port

The ATM port on the ATM-LAN bridge does not have an external connector and communicates with the A-CPSW via the ATM backplane on the 8260. The ATM interface complies with UNI 3.0 specifications.

#### 3.5.1.8 RS-232 Console Port

In addition to the LAN ports, the ATM-LAN Bridge module has an RS-232 console port (also known as service port). The service port enables you to connect a workstation to the ATM-LAN Bridge module to load new configuration, microcode, etc.

**Note:** After the initial configuration, you can also access the IBM 8281 inband through LAN or ATM ports to load new configuration, microcode, etc.

#### 3.5.1.9 Wrong Slot LED

The ATM-LAN Bridge module can be installed on any pair of slots on the 8260, except the ones that include slots 9, 10, and 11.

### 3.5.2 Sample Configurations Using ATM-LAN Bridge Module

The following sections show you some examples of using the ATM-LAN Bridge module.

#### 3.5.2.1 Local Token-Ring to Token-Ring Bridging

Figure 22 shows an example of using the ATM-LAN bridge as a means of bridging token-ring networks on two different 8260s. Note that in this example, the module is acting as a pure LAN bridge and does not involve the ATM connection.

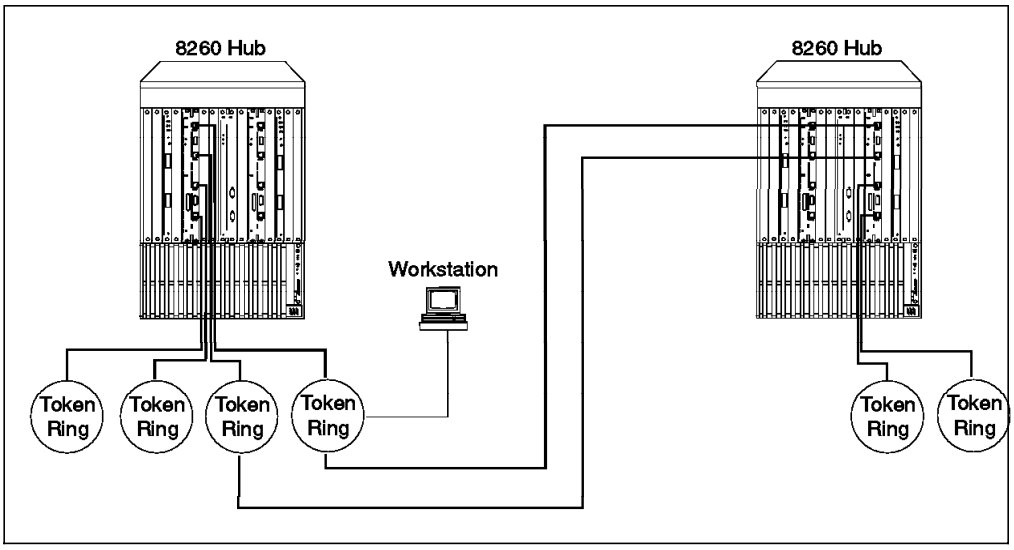



*Figure 22. Token-Ring Bridging Using Two Bridge Modules*

#### 3.5.2.2 Token-Ring to Token-Ring Bridging over ATM

Figure 23 on page 52 shows an example of using the ATM-LAN Bridge as a means of bridging two token-ring networks via an emulated token-ring over ATM. Note that in this example, the ATM ports on both bridges are configured as an LE client of the emulated token-ring over ATM.

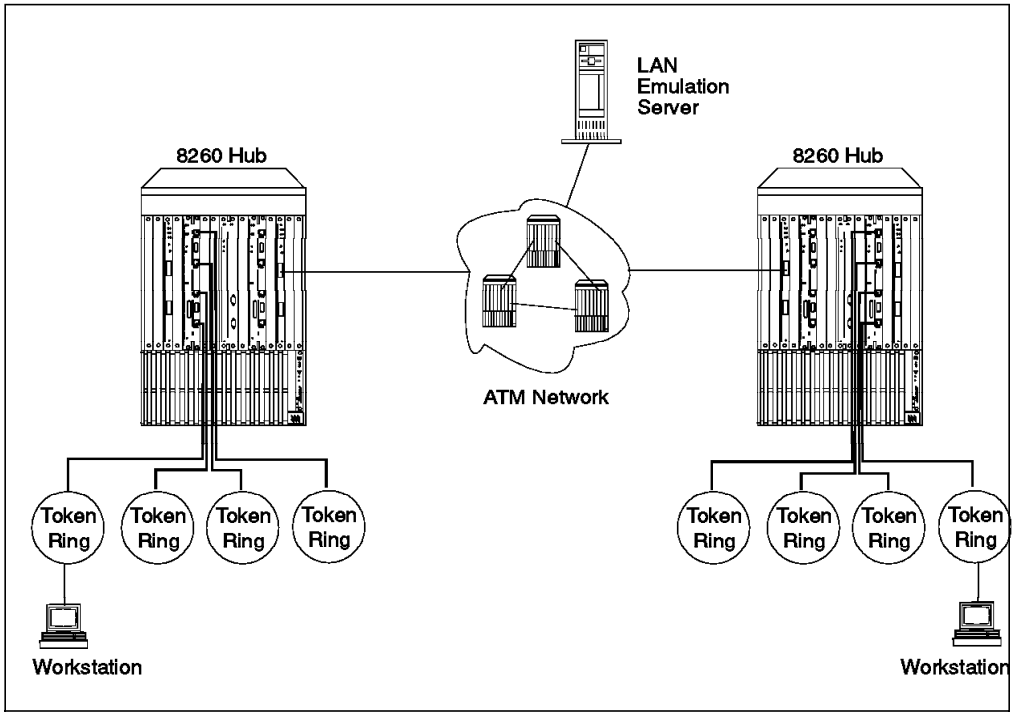



*Figure 23. Token-Ring to Token-Ring Bridging over ATM*

**Note:** In the previous example, the ATM-LAN bridge also supports the configuration where the token-ring LAN and the emulated token-ring over ATM are replaced with Ethernet LAN and an emulated Ethernet over ATM.

### 3.5.2.3 Token-Ring to ATM Bridging

Figure 24 on page 53 shows an example of using the ATM-LAN bridge as a means of bridging between token-ring and ATM-attached workstations using IBM LAN emulation.

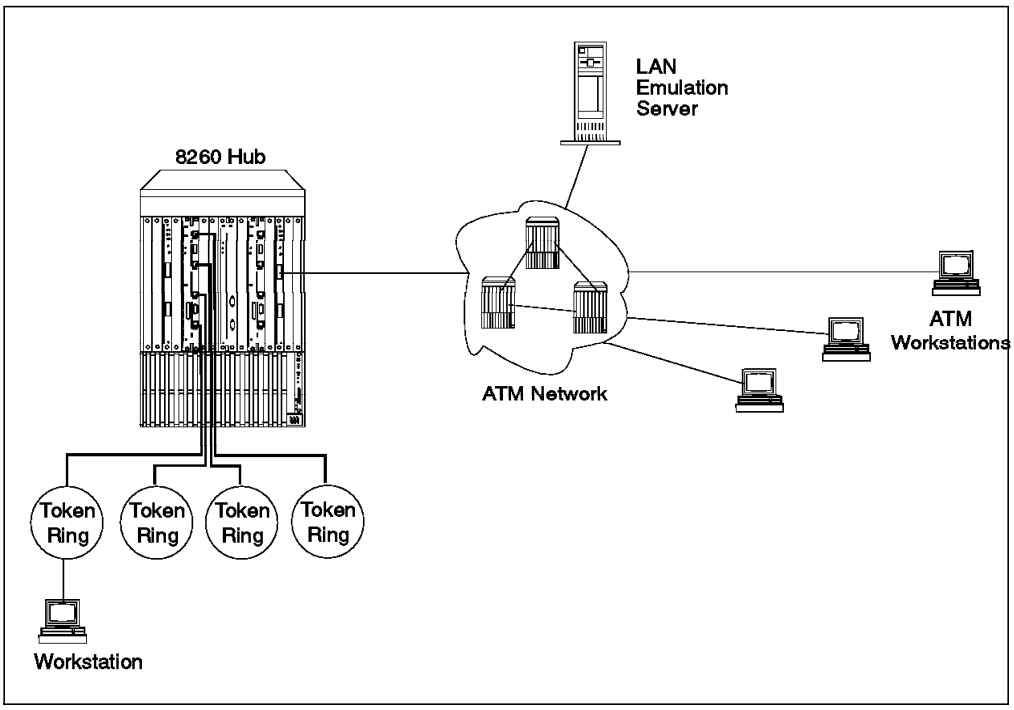



*Figure 24. Token-Ring to Token-Ring Bridging over ATM*

**Note:** In the previous example, the ATM-LAN bridge also supports the configuration where the token-ring LAN and the emulated token-ring over ATM are replaced with Ethernet LAN and an emulated Ethernet over ATM.

**Important**

In a token-ring environment, the bridge number is used to identify the bridge in the routing information field (RIF). The requirement for the bridge number is that it has to be unique between any two segments that are connected to each other (this means that the parallel bridges connecting two segments must have different numbers from each other). You may have more than one bridge with the same bridge number in your network as long as there is no other bridge parallel to it with the same bridge number.

However, using the current level of the code available for the ATM-LAN Bridge module (Rel 1.0), each ATM bridge connected to the same emulated token-ring LAN over ATM must have a different bridge number. This is regardless of the fact that the bridges may or may not be parallel to each other. This limitation will reduce the number of bridges connected to an emulated token-ring LAN to 15.

This is a limitation of the current release of the microcode and may be removed in the future releases.

### 3.5.3 ATM-LAN Bridge Module and LAN Emulation

The ATM-LAN Bridge module provides connectivity between traditional LANs (token-ring or Ethernet) and ATM networks by sending LAN frames transparently over the ATM network using LAN emulation to resolve MAC-to-ATM addresses.

**Important**

Currently, the ATM-LAN Bridge module only supports the IBM proprietary LAN emulation architecture and does not support the ATM Forum′s LAN emulation specification.

The IBM LAN emulation service allows the ATM network to emulate the services of either a token-ring or an Ethernet LAN. The LAN emulation service is provided jointly by LAN emulation server (LE server) and the LAN emulation client (LE client) software running in the workstation attached to the ATM network.

As an LE client, the ATM-LAN Bridge module is able to find the correct ATM destination, to set up the connection, and to switch LAN traffic to that destination on behalf of a LAN endstation. It is self-learning, meaning that it is able to discover its ATM partners and establish connections on an as-needed basis. More details about LAN emulation is provided in Appendix B, “IBM LAN Emulation over ATM” on page 279.

The following sections provide details about the data flow when using IBM LAN emulation in the ATM-LAN Bridge module.

#### 3.5.3.1 802.3 ATM Station-to-802.3 LAN Station

In this section we examine the message flow between an ATM-attached station (ATMES1) emulating 802.3 and an Ethernet/802.3 attached station (LANES1) that is connected to the ATM network via an ATM-LAN transparent bridge (ATMBR1). The message flow between these two stations is shown in Figure 25 on page 55.

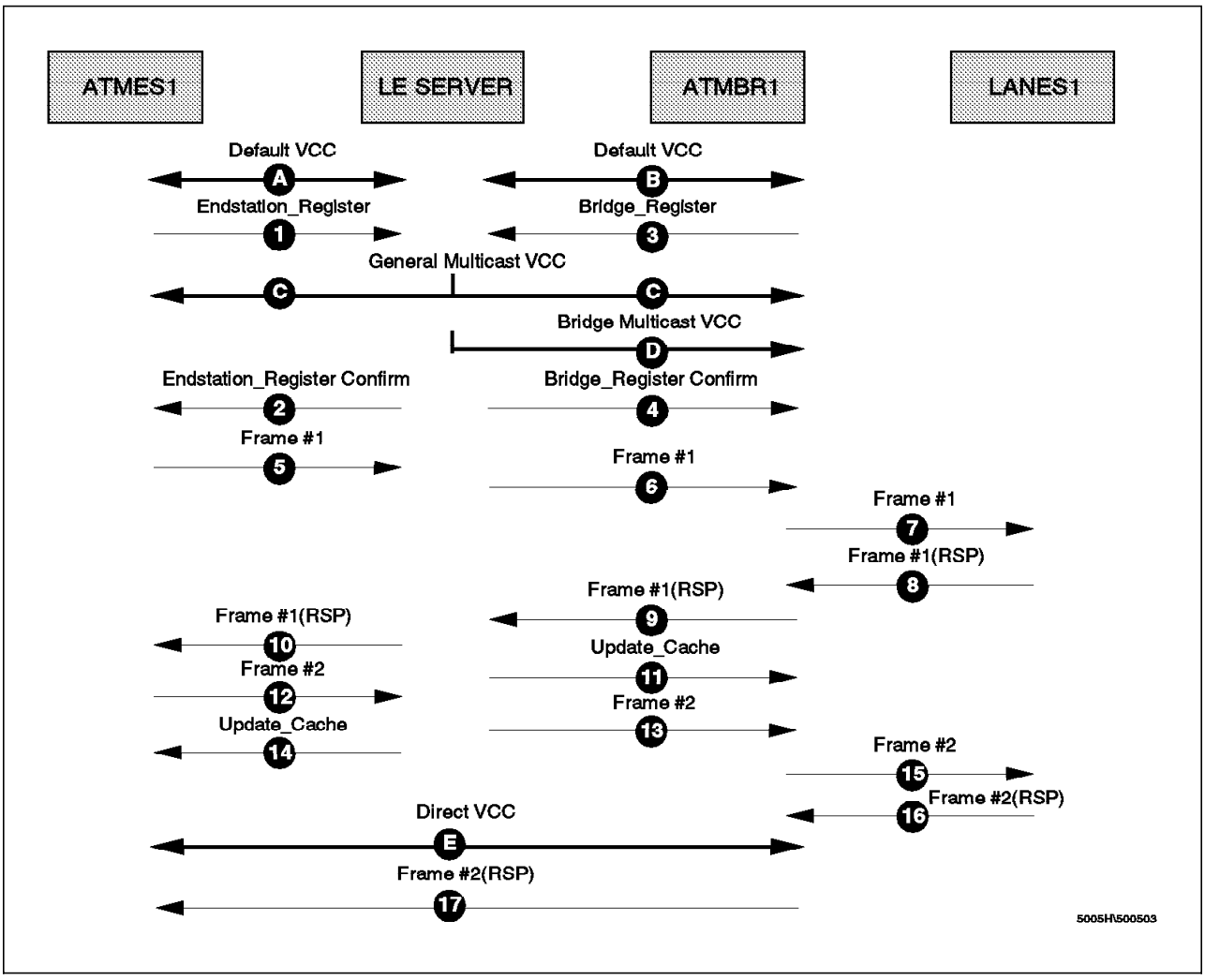



*Figure 25. 802.3 ATM Station-to-802.3 LAN Station Message Flow*

In this scenario, it is assumed that the ATM endsystem (ATMES1), the ATM-LAN bridge (ATMBR1), and the LAN emulation server have all completed their ILMI process and have initialized at both physical and ATM layers. They already know their ATM address. Also, ATMES1 and ATMBR1 know the ATM address of the LE server, as this is a configuration parameter specified by the administrator as part of the configuration of ATMES1 and ATMBR1.

1. ATMES1 establishes the Default VCC **A** with the LE server.
2. ATMES1 sends a REGISTER_ENDSTATION **1** to the LE server using the Default VCC **A**.

   This frame contains information including the type of frame, registration or deregistration, type of multicast service required, and whether ATMES1 will establish Direct VCCs. It may optionally include the ATM address of ATMES1.
3. The LE server updates its SAAT forwarding/filtering table.
4. The LE server adds ATMES1 as a leaf to the General Multicast VCC **C**.
5. The LE server issues a REGISTER_ENDSTATION_CONFIRM **2** to ATMES1 using the Default VCC **A**.

This frame contains information including a list of all successfully registered addresses, the emulated LAN identifier (ELID) and the originator identifier (OID).

6. The ATM-LAN bridge (ATMBR1) establishes the Default VCC **B** with the LE server.
7. ATMBR1 issues a REGISTER_BRIDGE **3** to the LE server using the Default VCC **B**.
8. The LE server updates its SAAT forwarding/filtering table.
9. The LE server adds ATMBR1 to the General Multicast VCC **C**.
10. The LE server adds ATMBR1 to the Bridge VCC **D**.
11. The LE server issues a REGISTER_BRIDGE_CONFIRM **4** to ATMBR1 using the Default VCC **B**.

    This frame contains information including a list of all successfully registered addresses, the emulated LAN identifier (ELID) and the originator identifier (OID).
12. When ATMES1 wishes to send a unicast data frame to the LAN-attached device LANES1, it examines its Destination Address Association Table (DAAT) to determine whether it knows the ATM address of the target. In this scenario it is assumed that the address of LANES1 is not in the ATMES1′s DAAT and is not known by the LE server either.

    **Note:** In our discussion we have assumed that the upper layer protocol already knows the MAC address of the destination. Normally, a broadcast frame is used by the upper layer protocol to obtain this MAC address. For the sake of simplicity, we have not shown this step in out flows.
13. The LE server examines its SAAT to ATMBR1.
14. ATMES1 sends the data frames **5** for LANES1 over its Default VCC **A** to the LE server.
15. Since the LANES1′s address is not in the LE server′s SAAT, the LE server forwards this frame **6** to all the bridges over the Bridge Multicast VCC **D**.

    **Note:** Since we only have a single ATM-LAN bridge in our configuration example, **6** is sent to ATMBR1 only.
16. The LE layer in the ATMBR1 receives this message and passes it to the Bridge Relay function.
17. The Bridge Relay function either does not have an entry in its filtering database for the MAC address of the LANES1, or it has an entry that shows that LANES1 is on the LAN port. In either case, the Bridge Relay function forwards this message to LANES1 over the LAN port **7**. The Bridge Relay function will also update its transparent bridging database to reflect the fact that ATMES1 is on the port attached to the ATM side.
18. LANES1 issues a response **8** over the Ethernet LAN.
19. The Bridge Relay function receives this message and passes it to the ATM port. (In reality the Bridge Relay function does not know that this is an ATM port, it thinks that it is an Ethernet LAN and will pass the message to the LE layer of the ATM port.)
20. Since the LE layer in ATMBR1 does not have ATMES1 in its DAAT, it forwards the response **9** to the LE server over the Default VCC **B**.

21. The LE server learns that the LANES1 is reachable via the ATM address of ATMBR1. Since the LE server already has an entry for the MAC and ATM address of ATMBR1 in its SAAT, it will append the MAC address of the LANES1 to this table entry.
22. The LE server forwards the response **10** to ATMES1 over the Default VCC **A**.
23. The LE server issues an Update_Cache **11** to ATMBR1 over the Default VCC **B**, with the ATMES1′s MAC-to-ATM address mapping so that ATMBR1 can update its DAAT.
24. ATMBR1 updates its DAAT.
25. When ATMES1 receives another frame destined to LANES1, it again sends it to the LE server **12** over its Default VCC **A**.
26. Since the LE server now knows that LANES1 is reachable via the ATM address of the ATMBR1, it forwards the message **13** directly to ATMBR1 over the Default VCC **B**.
27. The LE server issues an Update_Cache to ATMES1 **14** over the Default VCC **A** with the LANES1′s MAC to ATMBR1′s ATM address mapping so that ATMES1 can update its DAAT.
28. ATMES1 updates its DAAT.
29. The message received by ATMBR1 will be passed to the Bridge Relay function which, in turn, will send it over the LAN port to LANES1 **15**.
30. The response sent by LANES1 **16** will be passed to the LE layer in the bridge.
31. Since the LE layer in ATMBR1 already knows the MAC and ATM address of ATMES1, it will establish a direct connection **E** with ATMES1 and send the response from LANES1 to ATMES1 over this connection **17**.
32. Any subsequent unicast frames between ATMES1 and LANES1 will be exchanged over this direct connection.

**Note**

In this scenario, if ATMES1 requires communication with a station other than LANES1 on the Ethernet/802.3 LAN, it has to use the LAN server′s directory service to discover that this station is reachable via the ATM address of ATMBR1. Once it has made such a determination, it will use its existing Direct VCC to ATMBR1 to exchange messages with LANES2 (there is no need for a new connection to be established). Also, the LE server will use the General Multicast VCC to discover that the destination station on Ethernet/802.3 LAN is behind ATMBR2. Once it has done so, it will update its SAAT and will use its existing Default VCC to ATMBR1 to forward messages destined to that station.

#### 3.5.3.2 802.5 ATM Station-to-802.5 LAN Station

In this section, we examine the message flow between an ATM-attached station (ATMES1) emulating 802.5 and a token-ring-attached station (LANES1) that is connected to the ATM network via an ATM-LAN source route bridge (ATMBR1). The message flow between these two stations is shown in Figure 26 on page 58.

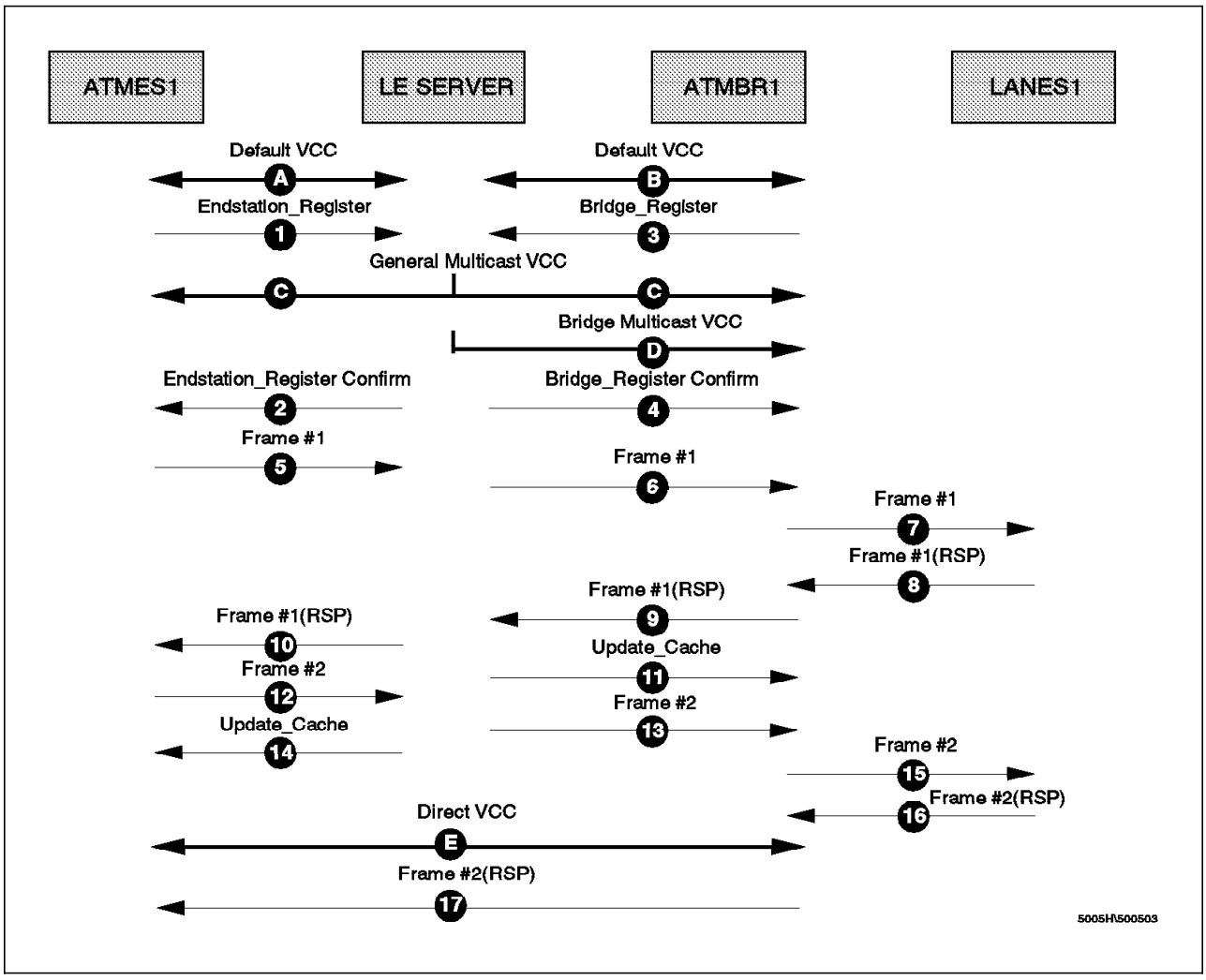



*Figure 26. 802.5 ATM Station-to-802.5 LAN Station Message Flow*

In this scenario, it is assumed that the ATM endsystem (ATMES1), the ATM-LAN bridge (ATMBR1), and the LAN emulation server have all completed their ILMI process and have initialized at both physical and ATM layers. They already know their ATM address. Also, ATMES1 and ATMBR1 know the ATM address of the LE server, as this is a configuration parameter specified by the administrator as part of the configuration of ATMES1 and ATMBR1.

1. ATMES1 establishes the Default VCC **A** with the LE server.
2. ATMES1 sends a REGISTER_ENDSTATION **1** to the LE server using the Default VCC **A**.

   This frame contains information including the type of frame, registration or deregistration, type of multicast service required, and whether ATMES1 will establish Direct VCCs. It may optionally include the ATM address of ATMES1.
3. The LE server updates its SAAT forwarding/filtering table.
4. The LE server adds ATMES1 as a leaf to the General Multicast VCC **C**.
5. The LE server issues a REGISTER_ENDSTATION_CONFIRM **2** to ATMES1 using the Default VCC **A**.

This frame contains information including a list of all successfully registered addresses, the emulated LAN identifier (ELID) and the originator identifier (OID).

6. ATM-LAN bridge (ATMBR1) establishes the Default VCC **B** with the LE server.
7. ATMBR1 issues a REGISTER_BRIDGE **3** to the LE server using the default VCC **B**. Among others, the REGISTER_BRIDGE frame will be used to register the route descriptor (bridge ID for ATMBR1 and segment ID for the token-ring LAN) with the LE server.
8. The LE server updates its SAAT forwarding/filtering table.
9. The LE server adds ATMBR1 to the General Multicast VCC **C**.
10. The LE server adds ATMBR1 to the Bridge VCC **D**.
11. The LE server issues a REGISTER_BRIDGE_CONFIRM **4** to ATMBR1 using the Default VCC **B**.

    This frame contains information including a list of all successfully registered addresses, route descriptor fields, the emulated LAN identifier (ELID) and the originator identifier (OID).
12. When ATMES1 wishes to send a unicast data frame to a LAN-attached device (LANES1), it examines its Destination Address Association Table (DAAT) to determine whether it knows the ATM address of the target. In this scenario, it is assumed that the address of LANES1 is not in the ATMES1′s DAAT.

    **Note:** In our discussion we have assumed that the upper layer protocol already knows the MAC address of the destination. Normally, a broadcast frame is used by the upper layer protocol to obtain this MAC address. For the sake of simplicity, we have not shown this step in out flows.
13. ATMES1 sends the data frames as an all route explorer (ARE) frame **5** destined for LANES1 over its Default VCC **A** to the LE server.

    **Note:** Some stations use spanning tree explorer (SRE) frames instead of ARE frames. However, this would make no difference as far as the LAN emulation flows are concerned.
14. Since the LE server does not have a VCC associated with the destination address of LANES1, it broadcasts the frame **6** over the Bridge Multicast VCC **D**.

    **Note:** Since we have only a single ATM-LAN bridge in our configuration example, **6** is sent to ATMBR1 only.
15. The LE layer in the ATMBR1 receives this message and passes it to the Bridge Relay function.
16. The Bridge Relay function in ATMBR1 forwards the ARE frame **7** into the token-ring network. In doing so, it also updates the routing information field in the frame.
17. LANES1 issues a response **8** over the token-ring LAN. Note that this response is a specifically routed frame (SRF) containing the necessary routing information (that is, token-ring segment number, ATM-LAN bridge number, and ATM-LAN segment number).
18. The Bridge Relay function receives this message and passes it to the ATM port. (In reality, the Bridge Relay function does not know that this is an ATM

port. It thinks that it is a token-ring LAN, so it will pass the message to the LE layer of the ATM port.

19. Since the LE layer in the ATMBR1 bridge does not have an entry for ATMES1, it forwards the response **9** to the LE server over the Default VCC **B**.
20. The LE server forwards the response to ATMES1 **10**.
21. The LE server issues an Update_Cache **11** to ATMBR1 over the Default VCC **B**, with the ATMES1′s MAC-to-ATM address mapping so that ATMBR1 can update its DAAT.
22. ATMBR1 updates its DAAT.
23. When ATMES1 receives another frame destined to LANES1, it again sends it to the LE server (but this time as a specifically routed frame) **12** over its Default VCC **A**.
24. The LE server examines the frame to determine if it has a Default VCC associated with the next hop. Since the next hop points to the ATMBR1 and token-ring LAN segment, and this bridge and segment are already registered with the LE server, the LE server forwards this frame **13** directly to ATMBR1 over the Default VCC **B**.
25. The LE server issues an Update_Cache to ATMES1 **14** over the Default VCC **A** with the route descriptor (ATMBR1 bridge number and token-ring LAN segment number) to ATMBR1′s ATM address mapping so that ATMES1 can update its DAAT.
26. ATMES1 updates its DAAT.
27. The message received by ATMBR1 will be passed to the Bridge Relay function which, in turn, will send it over the LAN port to LANES1 **15**.
28. The response sent by LANES1 **16** will be passed to the LE layer in the bridge.
29. Since the LE layer in ATMBR1 already knows the MAC and ATM addresses of ATMES1, it will establish a direct connection **E** with ATMES1 and send the response from LANES1 to ATMES1 over this connection **17**.
30. Any subsequent unicast frames between ATMES1 and LANES1 will be exchanged over this direct connection.

**Note**

In this scenario, if ATMES1 requires the sending of a specifically routed frame to a station attached to the token-ring LAN, it will send it directly to ATMBR1, because ATMES1 has in its DAAT address mapping between the route descriptor (ATMBR1 bridge number and token-ring LAN segment number) and the ATM address of the ATMBR1.

However, if ATMES1 needs to send an explorer frame to a station attached to ATMBR1, it will use the LE server to forward the explorer frame, as described in the previous steps.

### 3.5.3.3 802.3 LAN Station-to-802.3 LAN Station over ATM

In this section we examine the message flow between two Ethernet/802.3-attached stations (LANES1 and LANES2) communicating with each other over an ATM network emulating an 802.3 LAN. The Ethernet/802.3 LANs are bridged to the emulated Ethernet/802.3 LAN (over ATM) using the ATMBR1 and ATMBR2 ATM-LAN transparent bridges. The configuration of this scenario is shown in Figure 27.

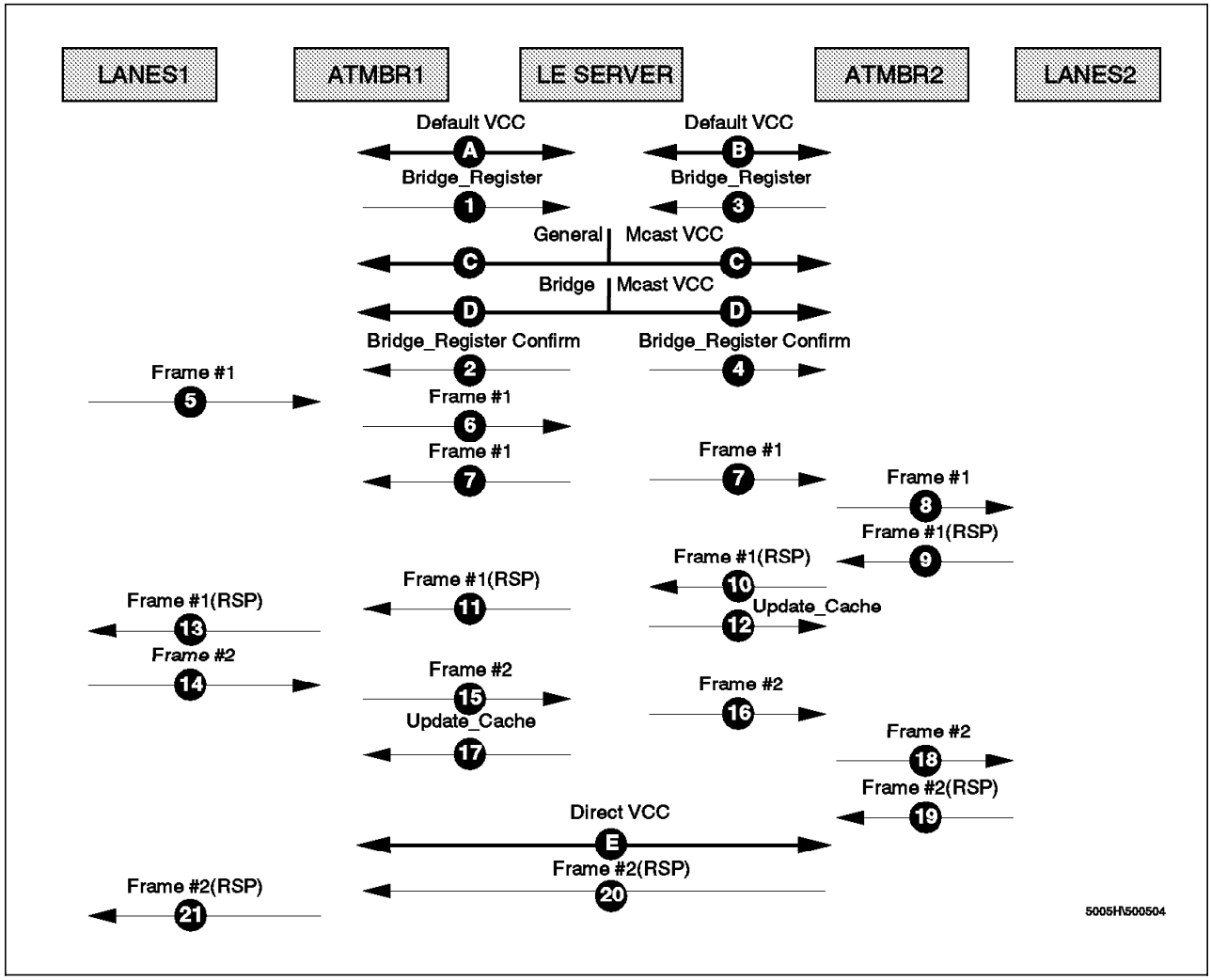



*Figure 27. 802.3 LAN Station-to-802.3 LAN Station over ATM*

In this scenario, it is assumed that the ATM-LAN bridges (ATMBR1 and ATMBR2) and the LAN emulation server have all completed their ILMI process and have initialized at both physical and ATM layers, and they all know their ATM address. Also, ATMBR1 and ATMBR2 know the ATM address of the LE server, as this is a configuration parameter specified by the administrator as part of the configuration of ATMBR1 and ATMBR2.

1. ATMBR1 establishes the Default VCC **A** with the LE server.
2. ATMBR1 sends a REGISTER_BRIDGE **1** to the LE server using the Default VCC **A**.
3. The LE server updates its SAAT forwarding/filtering table.
4. The LE server adds ATMBR1 to the General Multicast VCC **C**.

5. The LE server adds ATMBR1 to the Bridge VCC **D**.
6. The LE server issues a REGISTER_BRIDGE_CONFIRM **2** to ATMBR1 using the Default VCC **A**.

   This frame contains information including a list of all successfully registered addresses, the emulated LAN identifier (ELID) and the originator identifier (OID).
7. ATM-LAN bridge (ATMBR2) establishes the Default VCC **B** with the LE server.
8. ATMBR2 issues a REGISTER_BRIDGE **3** to the LE server using the Default VCC **B**.
9. The LE server updates its SAAT forwarding/filtering table.
10. The LE server adds ATMBR1 to the General Multicast VCC **C**.
11. The LE server adds ATMBR1 to the Bridge VCC **D**.
12. The LE server issues a REGISTER_BRIDGE_CONFIRM **4** to ATMBR2 using the Default VCC **B**.

    This frame contains information including a list of all successfully registered addresses, the emulated LAN identifier (ELID) and the originator identifier (OID).
13. When LANES1 wishes to send a unicast data frame to LANES2, it will send it over the Ethernet/802.3 network **5**.
14. It is assumed that at this stage, the Bridge Relay function in ATMBR1 does not have an entry in its filtering database for the MAC address of the LANES2. In this case, the Bridge Relay function passes this message to the LE layer on the ATM port.
15. The LE layer in the ATMBR1 does not have LANES2 in its DAAT, therefore, it will forward this message **6** to the LE server over its Default VCC **A**.
16. It is assumed that at this stage the LE server does not have LANES2 in its SAAT. Therefore, the LE server multicasts this frame to all of the bridges **7** over the Bridge Multicast VCC **D**.
17. The LE server, also, learns that LANES1 is reachable via ATMBR1. Since the LE server already has an entry for the MAC and ATM addresses of ATMBR1 in its SAAT, it will append the MAC address of the LANES1 to this table entry.
18. ATMBR1 discards the frame received from the LE server over the Bridge Multicast VCC, because it can identify the frame that it originated itself by investigating the OID field.
19. The LE layer in the ATMBR2 passes this message to the Bridge Relay function.
20. The Bridge Relay function in ATMBR2 either does not have an entry in its filtering database for the MAC address of the LANES2, or it has an entry that shows that LANES2 is on the LAN port. In either case, the Bridge Relay function forwards this message **8** to LANES2 over the LAN port. The Bridge Relay function will also update its transparent bridging database to reflect the fact that LANES1 is on the port attached to the ATM side.
21. LANES2 issues a response **9** to this message.

22. The Bridge Relay function in ATMBR2 will forward this response over the ATM port. In reality, the Bridge Relay function does not know that this is an ATM port; it thinks that it is an Ethernet/802.3 LAN, and it will pass the message to the LE layer in the bridge.
23. Since the LE layer in ATMBR2 does not have an entry in its SAAT for LANES1, it forwards this frame **10** to the LE server over the Default VCC **B**.
24. The LE server learns that the LANES2 is reachable via the ATM address of ATMBR2. Since the LE server already has an entry for the MAC and ATM addresses of ATMBR2 in its SAAT, it will append the MAC address of the LANES2 to this table entry.
25. The LE server forwards the response to ATMBR1 **11** over the Default VCC **A**.
26. The LE server issues an Update_Cache **12** to ATMBR2 over the Default VCC, informing it of the fact that LANES1 can be reached via the ATM address of ATMBR1.
27. ATMBR2 updates its DAAT.
28. The Bridge Relay function in ATMBR1 will send the frame received from the LE server over the Ethernet/802.3 LAN **13**.
29. When LANES1 receives a second message destined to LANES2, it will send it over the LAN port **14**.
30. The Bridge Relay function in ATMBR1 will pass this message to the LE layer. The LE layer does not have an entry for LANES2 in its DAAT, therefore, it will send this message **15** over the Default VCC **A** to the LE server.
31. Since the LE server now knows that LANES2 is reachable via the ATM address of the ATMBR2, it forwards the message directly to ATMBR2 over the Default VCC **16**. (It does not broadcast the message as it did with the first message.)
32. The LE server issues an Update_Cache to ATMES1 **17** over the Default VCC **A** with the LANES2′s MAC to ATMBR2′s ATM address mapping, so that ATMBR1 can update its DAAT.
33. ATMBR1 updates its DAAT.
34. The message received by ATMBR2 will be passed to the Bridge Relay function which, in turn, will send it over the LAN port to LANES2 **18**.
35. The response sent by LANES2 **19** will be passed to the LE layer in the bridge.
36. Since the LE layer in ATMBR2 now knows the MAC address of LANES1 is reachable via the ATM address of ATMBR1, it will establish a direct connection **E** with ATMBR1 and send the response from LANES2 to LANES1 over this connection **20** to ATMBR1.
37. ATMBR1 will, in turn, send this message over the Ethernet/802.3 port to LANES1 **21**.
38. Any subsequent messages between LANES1 and LANES2 will be exchanged over this direct connection.

### 3.5.3.4 802.5 LAN Station-to-802.5 LAN Station over ATM

In this section we examine the message flow between two token-ring-attached stations (LANES1 and LANES2) communicating with each other over an ATM network emulating a token-ring LAN. The token-ring LANs are bridged to the emulated token-ring LAN (over ATM) using the ATMBR1 and ATMBR2 ATM-LAN source route bridges. The message flow between these stations is shown in Figure 28.

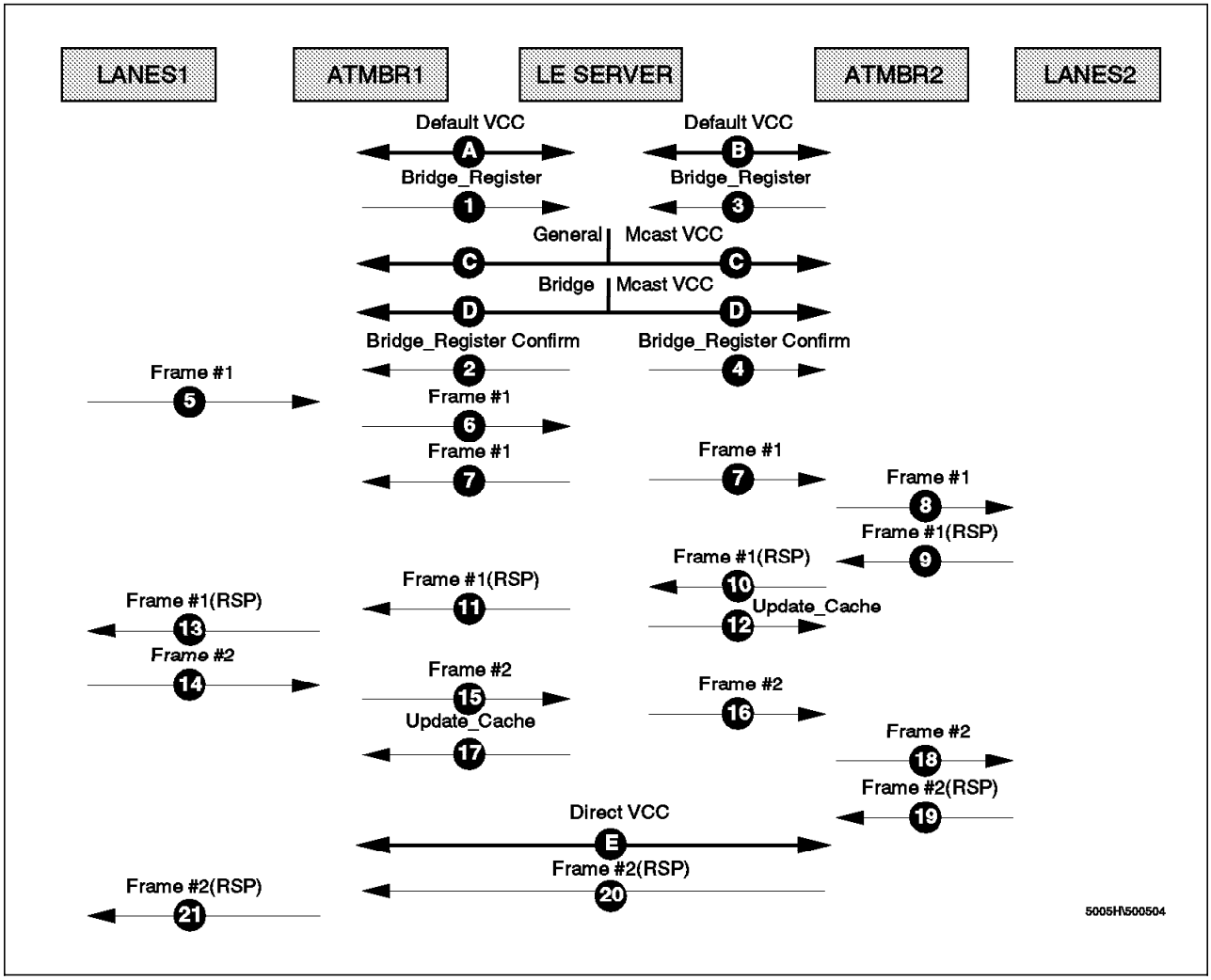



*Figure 28. 802.5 LAN Station-to-802.5 LAN Station over ATM*

In this scenario, it is assumed that the ATM-LAN bridges (ATMBR1 and ATMBR2) and the LAN emulation server have all completed their ILMI process, have initialized at both physical and ATM layers, and they all know their ATM address. Also, ATMBR1 and ATMBR2 know the ATM address of the LE server, as this is a configuration parameter specified by the administrator as part of the configuration of ATMBR1 and ATMBR2.

Figure 27 on page 61 shows the message flows used in IBM's implementation of LAN emulation:

1. ATMBR1 establishes the Default VCC **A** with the LE server.

2. ATMBR1 sends a REGISTER_BRIDGE 1 to the LE server using the Default VCC A. As part of this message, the ATMBR1 registers its bridge ID and the LAN segment identifier of its token-ring LAN.
3. The LE server updates its SAAT forwarding/filtering table.
4. The LE server adds ATMBR1 to the General Multicast VCC C.
5. The LE server adds ATMBR1 to the Bridge VCC D.
6. The LE server issues a REGISTER_BRIDGE_CONFIRM 2 to ATMBR1 using the Default VCC A.

   This frame contains information including a list of all successfully registered addresses, route descriptor fields, the emulated LAN identifier (ELID) and the originator identifier (OID).
7. ATM-LAN bridge (ATMBR2) establishes the Default VCC B with the LE server.
8. ATMBR2 issues a REGISTER_BRIDGE 3 to the LE server using the Default VCC B. As part of this message, the ATMBR2 registers its bridge ID and the LAN segment identifier of its token-ring LAN.
9. The LE server updates its SAAT forwarding/filtering table.
10. The LE server adds ATMBR1 to the General Multicast VCC C.
11. The LE server adds ATMBR1 to the Bridge VCC D.
12. The LE server issues a REGISTER_BRIDGE_CONFIRM 4 to ATMBR2 using the Default VCC B.

    This frame contains information including a list of all successfully registered addresses, route descriptor fields, the emulated LAN identifier (ELID) and the originator identifier (OID).
13. When LANES1 wishes to send a unicast data frame to LANES2, it will send it as an all route explorer (ARE) frame destined to LANES1 over the token-ring network 5.
14. The Bridge Relay function in ATMBR1 passes this message to the LE layer on the ATM port.
15. The LE layer in the ATMBR1 does not have LANES2 in its DAAT, therefore, it will forward this message 6 to the LE server over its Default VCC A.
16. Since the LE server does not have an entry for LANES2 in its SAAT, it multicasts this frame to all the bridges 7 over the Bridge Multicast VCC D.
17. ATMBR1 discards the frame received from the LE server over the Bridge Multicast VCC, because it can identify the frame that it originated itself by investigating the OID field.
18. The LE layer in the ATMBR2 passes this message to the Bridge Relay function.
19. The Bridge Relay function in ATMBR2 forwards this message 8 to LANES2 over the LAN port after updating the routing information field in the frame.
20. LANES2 issues a response 9 to this message. This response is a specifically routed frame.
21. The Bridge Relay function in ATMBR2 will forward this response over the ATM port. In reality, the Bridge Relay function does not know that this is an

ATM port; it thinks that it is a token-ring LAN, and it will pass the message to the LE layer in the bridge.

22. Since the LE layer in ATMBR2 does not have an entry in its DAAT for LANES1 or the route descriptor field (for ATMBR1 and the token-ring segment attached to it), it forwards this frame **10** to the LE server over the default VCC **B**.
23. Since the LE server already has an entry for the route descriptor (for ATMBR1 and its token-ring LAN segment) in its SAAT pointing to ATMBR1, it forwards the response to ATMBR1 **11** over the Default VCC **A**.
24. The LE server issues an Update_Cache **12** to ATMBR2 over the Default VCC, informing it of the fact that token-ring_1 reached via the ATM address of ATMBR1.
25. ATMBR2 updates its DAAT.
26. The Bridge Relay function in ATMBR1 will send this frame **13** over the token-ring LAN.
27. When LANES1 receives a second message destined to LANES2, it will send it as a specifically routed frame **14** over the LAN port.
28. The Bridge Relay function in ATMBR1 will pass this message to the LE layer which will, in turn, send it **15** over the Default VCC **A** to the LE server. This is because ATMBR1 does not have an entry in its SAAT for the route descriptor (for ATMBR2 and its token-ring LAN segment).
29. Since the LE server now knows that token-ring_2 is reachable via the ATM address of the ATMBR2, it forwards the message directly to ATMBR2 **16** over the Default VCC **B**.
30. The LE server issues an Update_Cache to ATMBR1 **17** over the Default VCC **A** informing it that token-ring_2/ATMBR2 can be reached via the ATM address of ATMBR2.
31. ATMBR1 will update its DAAT.
32. The message received by ATMBR2 will be passed to the Bridge Relay function, which, in turn, will send it over the LAN port to LANES2 **18**.
33. The response sent by LANES2 **19** will be passed to the LE layer in the bridge.
34. Since the LE layer in ATMBR2 now knows that token-ring_1 is reachable via the ATM address of ATMBR1, it will establish a direct connection **E** with ATMBR1 and send the response from LANES2 to LANES1 over this connection to ATMBR1 **20**.
35. ATMBR1 will, in turn, send this message over the token-ring port to LANES1 **21**.
36. Any subsequent messages between LANES1 and LANES2 will be exchanged over this direct connection.

### 3.5.4 Filtering Facilities in ATM-LAN Bridge Module

The ATM-LAN Bridge module has filtering capabilities that can be used between local LANs and between a local LAN and the ATM network.

In the token-ring environment the ATM bridge offers comprehensive filters based on the following:

- Hop count
- MAC address
- Ring number
- Source service access point (SAP)
- Sub-network access protocol (SNAP)

In an Ethernet environment, filters can be set up based on the following:

- MAC address
- Source service access point (SAP)
- Ethertype

Note that the filters only apply to the inbound traffic received on the LAN ports. There are not filters for the ATM ports or the outbound traffic on the LAN ports.

### 3.5.5 Management and Configuration Support

The IBM ATM-LAN Bridge module is primarily managed using the Simple Network Management Protocol (SNMP). The following is a list of the MIBs supported by the ATM-LAN Bridge module:

- MIB-II (RFC 1213)
- Bridge MIB (RFC 1493)
- Source Route Bridging MIB (RFC 1525)
- ATM (AToM) MIB (RFC 1695)
- ILMI MIB (ATM Forum UNI_3.0)
- IEEE 802.5 Token-Ring MIB (RFC 1231)
- IBM Extensions to RFC 1231
- Ethernet Interfaces MIB (RFC 1643)
- ATM-LAN Bridge module MIB

A Configuration Utility Program, running in a DOS/Windows environment, provides configuration, change, and fault management capabilities for the ATM-LAN Bridge module. The station running the Configuration Utility Program can communicate with the ATM-LAN Bridge module using the following connections:

1. Direct attachment to the serial port on the ATM-LAN Bridge module

   In this case the workstation must be configured to use TCP/IP over the SLIP interface.

2. LAN attachment

   In this case the workstation must be configured to use TCP/IP over a LAN (token-ring or Ethernet) interface to communicate with the ATM-LAN Bridge module.

3. ATM attachment

   In this case the workstation must be configured to use TCP/IP over an emulated LAN to communicate with the ATM-LAN Bridge module via the ATM interface.

Detailed information concerning parameters used to configure the ATM-LAN bridge is provided in *Nways 8260 ATM TR/Ethernet LAN Bridge Module, Installation and User's Guide,* GA27-4070.

### 3.5.6 Association between IP and MAC Address

The ATM-LAN Bridge module can be reached via four LAN ports and/or one ATM port, but it has only a single IP address that is assigned to it at the time of configuration. This IP address will be associated with the first port on the ATM-LAN Bridge module that connects to a network successfully. If more than one port is configured for the ATM-LAN Bridge module, there will be a race condition to determine which port is associated with the IP address.

The MAC address of the port associated with the IP address will be used in the response to the ARP requests sent to the IP address of the ATM-LAN Bridge module, regardless of the port on which the ARP request is received. If the port which is associated with the IP address of the ATM-LAN Bridge module becomes disabled (say the cable is disconnected), the IP-to-MAC address association will remain unchanged. This means that the ATM-LAN Bridge module will still respond to ARP requests with the MAC address of the port that was initially associated with the IP address of the ATM-LAN Bridge module. This ensures that the ARP table entry in the stations that communicate with the ATM-LAN Bridge module via the IP will still be valid regardless of the fact that the port with that MAC address may be down.

If the ATM-LAN Bridge module is reset and the MAC address of another port is associated with the IP address of the ATM-LAN Bridge module, the ARP table entry in the stations that were communicating with the ATM-LAN Bridge module will become invalid. Those stations will not be able to communicate with the ATM-LAN Bridge module via IP until either their ARP table entry is aged-out or deleted by the user to allow the IP station to discover the new MAC address associated with the ATM-LAN Bridge module. Therefore, if you had problems communicating with the ATM-LAN Bridge module via the IP, one of the first things that you can do is to delete the ARP entry in your IP workstation to enable it to rediscover the ATM-LAN Bridge module via the ARP.

### 3.5.7 ATM-LAN Bridge Module ATM Bridge Software Modes

To understand how to use the ATM-LAN Bridge module, you must know how its software functions when in different operational modes. These modes define the operating state of the ATM-LAN bridge.

When the ATM-LAN bridge has been powered on and has completed its hardware diagnostics, it can be in one of the following three modes.

#### 3.5.7.1 Unconfigured Mode

In this mode, the ATM-LAN Bridge module is capable of running but requires initial configuration. This is the state of the ATM-LAN bridge when it is shipped.

After you have received the ATM-LAN Bridge module, the first time that you start the ATM-LAN Bridge module, it will start in unconfigured mode. At this point it cannot operate in a network because it has not been configured yet. You must use the Configurator Utility Program to configure it.

#### 3.5.7.2 Operational Mode

After successful configuration, the ATM-LAN Bridge module runs in operational mode. In this mode, the ATM-LAN Bridge module can be managed using SNMP. This means that you can use an SNMP management application to monitor and control the module.

The ATM-LAN Bridge module Configuration Utility Program, although not a management application, can be used to change and view the configuration, to change the operational software, and to help diagnose faults.

#### 3.5.7.3 Minimal Mode

This is the fallback mode. Minimal mode is used when the ATM-LAN bridge must be taken out of operational mode to download new operational software, to erase the current configuration, or to perform a memory dump. This mode enables you to get the ATM-LAN Bridge module back into the operational mode when the Configuration Utility Program cannot update the ATM-LAN Bridge module parameters, or when the ATM-LAN Bridge module's operation has to be interrupted to perform one of the utility functions.

Minimal mode provides the following three separate utility functions:

1. The ability to download new operational code
2. The ability to erase the bridge configuration
3. The ability to perform a memory dump

To enter minimal mode, set up the Configuration Utility Program in a workstation connected to the ATM-LAN Bridge module service port, select **Utilities**, and choose one of the available selections.

### 3.5.8 ATM-LAN Bridge Module Configuration Utility Program

The Configuration Utility Program is a DOS/Windows-based application that enables a user to modify the ATM-LAN Bridge module's configuration parameters, to change the operating code, and to use minimal mode. The following is the list of the functions that can be performed using the Configuration Utility Program:

- Create a bridge profile
- Edit a bridge profile
- View a bridge profile
- Save a bridge profile to the hard disk
- Delete a bridge profile from the hard disk
- Send a bridge profile to the ATM-LAN Bridge module
- Retreive current configuration parameters from an ATM-LAN Bridge module
- Load new operational parameters from an ATM-LAN bridge module
- View vital product data (VPD) for an ATM-LAN Bridge module
- Erase the configuration for an ATM-LAN Bridge module
- Perform a memory dump of the ATM-LAN Bridge module

To install the Configuration Utility Program, insert the diskette that contains the program in the diskette drive in your workstation, start Windows, and select **Run**

from the Program Manager File menu. This procedure starts the execution of the install.exe file from the diskette, which installs the Configuration Utility Program. When the installation is complete, the ATM-LAN Bridge module configuration group will appear as an icon on the Program Manager window (see Figure 29).

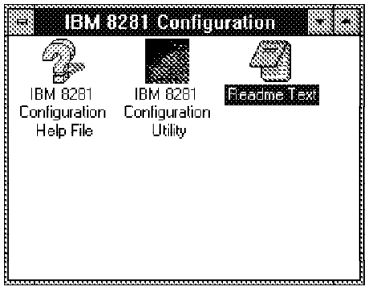



*Figure 29. ATM-LAN Bridge Module Configuration Window*

To use the Configuration Utility Program to manage the ATM-LAN Bridge module, the workstation running this program must be able to access the ATM-LAN Bridge module either via the service port or through a LAN or ATM port.

To use the service port, the workstation must be directly attached to the serial EIA 232 port (labeled service on the ATM-LAN Bridge module, see Figure 30) and must use the Serial Line Internet Protocol (SLIP) to communicate with the ATM-LAN Bridge module. Therefore, make sure that the TCP/IP protocol is running in the workstation, and that SLIP is correctly set up in the TCP/IP configuration.

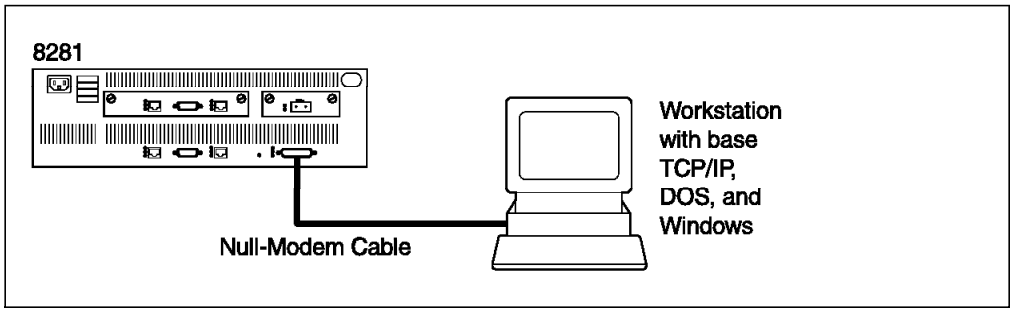



*Figure 30. The ATM-LAN Bridge Module Service Port Connection*

To access the ATM-LAN Bridge module via a LAN or ATM port, the workstation running the Configuration Utility Program must have IP connectivity through the network (either directly or through bridges, routers, etc.) to be able to reach the ATM-LAN Bridge module's LAN or ATM port. In this case, the TCP/IP stack in the workstation must be configured to provide such a connectivity.

**Note:** After the initial startup of the ATM-LAN Bridge module, you must use the direct connection to access the ATM-LAN Bridge module to load a valid configuration for the ATM-LAN Bridge module. After that, you may use either direct or LAN/ATM connections to access the ATM-LAN Bridge module for subsequent configurations.

The Configuration Utility Program provides a set of windows that allow you to configure and manage the ATM-LAN Bridge module. Figure 31 on page 71

shows how you can navigate between various windows provided by the Configuration Utility Program.

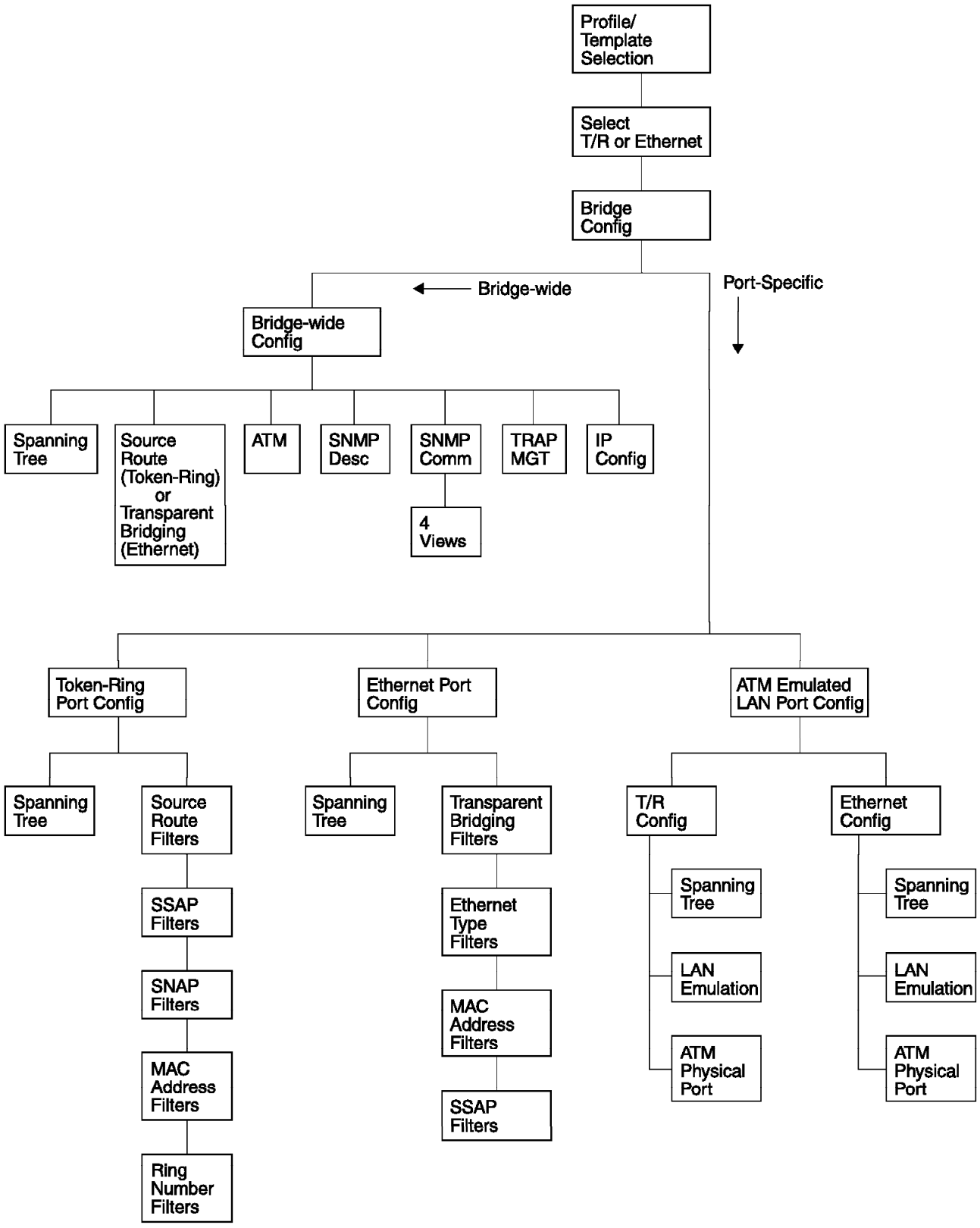



*Figure 31. Windows Displayed by the ATM-LAN Bridge Module Configurator*

For more information about how to install and use the Configuration Utility Program, please refer to *Nways 8260 ATM TR/Ethernet LAN Bridge Module Installation and User's Guide*, SA33-0361.

### 3.5.9 Running and Stored Configuration Parameters

When the ATM-LAN Bridge module is running, there are always the following two sets of parameters stored:

- Running parameters

  These are the parameter values that are in use by the operational code.

- Stored parameters

  These are the parameter values that exist in FLASH memory and are used only during the startup.

The stored parameters in the FLASH memory can be changed using the Configuration Utility Program by downloading new values for the configuration parameters. This can be done by creating a file of new parameter values (called Profile) and sending this file to the ATM-LAN Bridge module. To use new parameters as the running parameters, you need to restart the ATM-LAN Bridge module.

Once the ATM-LAN Bridge module is in operational mode, you can only view and change the stored parameters using the Configuration Utility Program. To view and change the running parameters, you must use an SNMP management station.

# Chapter 4. ATM Control Point and Switch (A-CPSW) Module

This chapter describes the various functions and features that are supported by the A-CPSW module. It also describes how to configure the A-CPSW module and the other components of the 8260 ATM switching system to take advantage of these functions.

## 4.1 Command Line Interface

As described in 3.2, “ATM Control Point and Switch Module” on page 33, the A-CPSW module is a two-slot module that houses the ATM switching fabric and the hardware and software components necessary to run the Control Point function.

To be able to configure and manage the A-CPSW module and the ATM media modules installed in the IBM 8260, the A-CPSW provides a command line interface that can be accessed via an ASCII terminal connected locally (or via a modem) to the RS-232 port on the front panel of the module.

The command line interface allows you to configure and display the status of the various components of the 8260 ATM switch system. Additionally it allows you to maintain the various software components of the A-CPSW by downloading new levels of microcode for these components. Finally, the command line interface provides you with the ability to collect traces and dumps of the various components in the event of problems that may occur in the ATM switching subsystem.

In the remainder of this chapter we discuss the various functions that are provided by the A-CPSW and how you can use the command line interface (accessible via the RS-232 interface) to select the configuration options that are most suitable for your environment.

### 4.1.1 How to Access the Command Line Interface

Before describing the functions of the A-CPSW, let’s first discuss how you can access the command line interface on the A-CPSW.

To be able to access the command line interface, you need to connect an ASCII terminal (VT100 or compatible) to the RS-232 port on the A-CPSW module. This connection can be either a local connection or through a telecommunication line using a pair of modems. The pinout of the cable used for connecting the ASCII terminal to the RS-232 port is provided in 3.2.1.3, “Console Port” on page 37. You must ensure that the ASCII terminal is configured according to the following factory default settings for the RS-232:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit

These settings can be changed once a terminal with the previous configuration is connected. The process of how to change these settings is discussed in 4.1.5, “How to Change Terminal Settings” on page 76.

© Copyright IBM Corp. 1996

Once your ASCII terminal with the correct settings is attached to the RS-232, press the Enter key. The following message is displayed:

```
ATM Control Point and Switch Module
(c) Copyright IBM Corp. 1994, 1995. All rights reserved.

Password:
```

*Figure 32. A-CPSW Login Message*

Enter 8260 as the password and press the Enter key. The module is shipped from the factory with this password. At this point you are logged into the A-CPSW with full access to all commands.

The command line interface has the following characteristics:

- The commands are not case-sensitive. The system interprets ABC (uppercase) the same as abc (lowercase).
- The A-CPSW command line accepts abbreviated command input, which enables you to enter a command by typing the minimum number of characters (that uniquely identifies the command) followed by the space bar. Pressing the space bar automatically fills the rest of the command.
- The system prompts you if you forget to enter a mandatory command.
- To get help, simply type ? on the command line. Also, when entering a command you get the system to display the next available options by entering ?.
- All the commands can be abnormally terminated by pressing the Ctrl and C keys simultaneously.

The following table is a quick reference to the procedures required to configure the ATM subsystem of the 8260. Some of them are mandatory; others are recommended.

*Table 16 (Page 1 of 2). A-CPSW Interface Configuration Quick Reference*

| **Procedure** | **A-CPSW Command** |
|---|---|
| Configure user and administrator passwords | SET DEVICE PASSWORD |
| Configure 8260 model | SET HUB_NUMBER_OF_SLOTS |
| Configure A-CPSW terminal settings | SET TERMINAL BAUD<br>SET TERMINAL DATA_BITS<br>SET TERMINAL PARITY<br>SET TERMINAL STOP_BITS<br>SET TERMINAL PROMPT<br>SET TERMINAL TIME_OUT<br>SET TERMINAL HANGUP<br>SET TERMINAL CONSOLE_PORT_PROTOCOL |
| Configure A-CPSW device configuration | SET CLOCK<br>SET DEVICE NAME<br>SET DEVICE LOCATION<br>SET DEVICE CONTACT<br>SET DEVICE DIAGNOSTICS |
| Configure A-CPSW ATM address | SET DEVICE ATM_ADDRESS |

| *Table 16 (Page 2 of 2). A-CPSW Interface Configuration Quick Reference* | |
|---|---|
| **Procedure** | **A-CPSW Command** |
| Configure Classical IP parameters for A-CPSW | SET DEVICE IP_ADDRESS<br>SET DEVICE DEFAULT_GATEWAY<br>SET DEVICE ARP_SERVER |
| Configure LE IP parameters for A-CPSW | SET DEVICE LAN_EMULATION_CLIENT |
| Configure SLIP parameters for A-CPSW | SET TERMINAL CONSOLE_PORT_PROTOCOL<br>SET TERMINAL SLIP_ADDRESS<br>SET TERMINAL BAUD |
| Configure LAN Emulation Configuration Server | SET LAN_EMUL |
| Configure SNMP Parameters for A-CPSW | SET COMMUNITY<br>SET ALERT |
| Configure ATM media modules | SET MODULE |
| Configure ATM ports | SET PORT |
| Configure NNI connection between A-CPSWs | SET LOGICAL_LINK |
| Configure ESI for ATM stations | SET ATM_ESI |
| Configure PVCs and PVPs | SET PVC |
| Configure NNI connections ATM-subnetworks | SET STATIC_ROUTE |
| Configure TFTP parameters for download/upload | SET TFTP |
| Configure trace/dump facilities | SET TRACE |

### 4.1.2 Administrator and User Access

There are two levels of access to A-CPSW via the console port:

- **User Level**

  When you are logged on as a user, you have access to some A-CPSW commands with read-only access. This allows you only to display the configuration of the 8260 ATM switching subsystem.

- **Administrator Level**

  When you are logged on as administrator, you have access to all the A-CPSW commands with read-write access, which allows you to display and modify the ATM switching subsystem in the 8260.

There are no user IDs associated with the user and administrator level. Once you connect to the A-CPSW, you will be prompted to enter a password. The password that you enter determines if you get administrator or user access. The factory defaults password for the user level is the Enter key, and for the administrator level is 8260 followed by the Ente r key.

### 4.1.3 How to Change Administrator and User Password

After logging in to A-CPSW for the first time, you are strongly advised to change the default password for the administrator, so you can prevent unauthorized users from being able to log on to the A-CPSW module to view or modify your ATM network configuration.

Figure 33 on page 76 shows how you can change the administrator password. For security purposes, the values you enter are not displayed on the screen.

```
8260A> set device password administrator
Enter current administrator password:
New password:
Re-enter password:
Password changed.
8260A>
```

*Figure 33. Changing Administrator Password*

**Note**

A-CPSW passwords are case-sensitive.

You can also change the user password. Figure 34 shows an example of how it is changed.

```
8260A> set device password user
Enter current administrator password:
New password:
Re-enter password:
Password changed.
8260A>
```

*Figure 34. Changing User Password*

### 4.1.4 Resetting the Password to Factory Default

If you forget the administrator password for the A-CPSW, you can use the following procedure to reset the password to the factory default:

1. Enter FORCE at the password prompt.
2. Press the ATM Reset button on the A-CPSW front panel.

### 4.1.5 How to Change Terminal Settings

You can customize the Terminal settings of the A-CPSW module if you are logged on as administrator. The following commands are provided:

- **Set Terminal Baud**

  This command allows you to set the baud rate at which A-CPSW will communicate with the attached console or modem. The following example shows you how to change the baud rate to 2400 bps:

```
8260A> set terminal baud 2400
```

*Figure 35. Changing the Terminal Baud Rate*

- **Set Terminal Data_Bits**

  This command lets you configure the A-CPSW module to the number of data bits used by the attached console. For example, the following command allows you to change the number of data bits to 7:

```
8260A> set terminal data_bits 7
```

*Figure 36. Changing the Terminal Data Bits*

- **Set Terminal Parity**

  This command lets you configure the A-CPSW module to the same parity used by the attached console. The following example shows you how to change the parity bit to even:

```
8260A> set terminal parity even
```

*Figure 37. Changing the Terminal Parity*

> **Note**
>
> When you change the settings for the baud, data_bits, or parity parameters, a mismatch will result between your ASCII terminal and the A-CPSW module. This mismatch will result in you not being able to access the A-CPSW module until you change the configuration of your terminal.

- **Set Terminal Stop_Bits**

  This command allows you to configure the A-CPSW module to the number of stop bits used by the attached console or modem. The following example shows you how to change the number of stop bits to 2:

```
8260A> set terminal stop_bits 2
```

*Figure 38. Changing the Terminal Stop Bits*

- **Set Terminal Prompt**

  This command allows you to customize the prompt that is displayed when you are connected to the A-CPSW. An example of this this command and the result is shown in Figure 39:

```
8260A> set terminal prompt 8260ATM1>
8260ATM1>
```

*Figure 39. Changing the Terminal Prompt*

  This command is useful for identifying the A-CPSW to which you are connected. The factory default prompt is 8260ATM>. It is recommended that you set the prompt to the device name you specify for the A-CPSW. See ″Set Device Name″ on page 83 for how to configure the A-CPSW device name. This will help you to identify the A-CPSW to which you are connected when using Telnet. For more information about Telnet access, please refer to 4.4, “Accessing the A-CPSW Using Telnet” on page 88.

- **Set Terminal Hangup**

  This command allows you to configure A-CPSW to automatically disconnect the modem (drop DTR) when you log off. To do so, you must issue the following command:

```
8260A> set terminal hangup disable
```

*Figure 40. Disabling the Terminal Auto Hangup*

The default is *disable*, which means that the modem will not hangup, and an unauthorized user may pick up your A-CPSW modem session.

- **Set Terminal Time_Out**

  This command allows you to set the number of minutes that you can remain logged on to an A-CPSW session without keyboard activity. This is a security measure that prevents unauthorized users from accessing and working in an open A-CPSW session when the A-CPSW console is left unattended. The default value is 0, which means that the terminal will never time out. An example of this command is as follows:

```
8260A> set terminal time_out 10
```

*Figure 41. Changing the Terminal Timeout*

The value previously specified is in minutes and can go up to 30.

After setting all the parameters for the terminal, you must ensure that you save them using the following command:

```
8260A> save terminal
```

*Figure 42. Saving the Terminal Settings*

You can display the current settings for the terminal using the following command:

```
8260A> show terminal
```

*Figure 43. Showing the Terminal Settings*

An example of the output you could get is shown in Figure 44.

```
Terminal Parameters:
Baud         9600
Data bits    8
Hangup       DISABLE
Parity       NONE
Stop bits    1
Timeout time 0
8260A>
```

*Figure 44. Output from Show Terminal Command*

## 4.2 ATM Address for A-CPSW

As part of the configuration of the A-CPSW module, you must use the Set Device ATM_Address command to define a unique ATM address for each 8260 ATM subsystem in the ATM network. This ATM address serves the following purposes:

1. It provides the NSAP prefix (the first 13 bytes of the ATM address) for all ATM devices attached to this 8260. When an ATM station attaches to the 8260, it uses the ILMI process to obtain the NSAP from the A-CPSW module. Then, after appending its ESI (endsystem identifier) to the SNAP, it will register the resulting address with the 8260.
2. It provides the endsystem identifier for the A-CPSW module. This ESI, combined with the NSAP prefix, will be the ATM address of the A-CPSW module. This address will be used by the IP component in the A-CPSW module to communicate with the other ATM-attached stations (using Classical IP over ATM or LAN emulation).

Each A-CPSW module has a default ATM address. You must change the default value to a unique ATM address in the network. The ATM address you assign is automatically saved, and the A-CPSW module is reset. As shown in A.1.7, “ATM Address Format” on page 268, the ATM address consists of 20 bytes. An 8260-based ATM system uses the Area field, the two lower bytes of the network prefix in the ATM address (bytes 12 and 13), to perform call routing, as described in 4.9, “Topology and Route Selection (TRS) Services” on page 108.

In the 8260 implementation, the *area* field consists of two parameters:

**ATM Cluster Number (ACN)** Valid values: 1-255

**Hub Number (HN)** Valid values: 1-255

An ATM cluster is a set of 8260 hubs interconnected by switch-to-switch (SSI) trunks. Each hub in an ATM cluster uses the same leftmost 12 bytes in the network prefix, including the ATM cluster number (ACN), and is assigned a unique hub number (HN), the rightmost byte in the network prefix.

An ATM subnetwork is a collection of clusters. Each ATM cluster in an ATM subnetwork uses the same 11 high-order bytes in the network prefix and is identified by a unique ACN number.

An ATM campus network is a collection of ATM subnetworks. Within an ATM campus network, each ATM subnetwork has a different NSAP prefix. For detailed information please refer to 4.9, “Topology and Route Selection (TRS) Services” on page 108.

### 4.2.1 How to Configure A-CPSW ATM Address

Figure 45 on page 80 shows how you set an A-CPSW ATM address with the following characteristics:

- Network Prefix = 39.09.85.11.11.11.11.11.11.11.11.03.04
  - Subnetwork address = 39.09.85.11.11.11.11.11.11.11.11
  - Cluster Number = 03
  - Hub Number = 04
- 8260 ESI = 40.00.00.82.60.A1.01

```
8260A> set device atm_address
39.09.85.11.11.11.11.11.11.11.11.03.04.40.00.00.82.60.A1.01
This call will reset the ATM subsystem.
Are you sure? (Y/N) Y
```

*Figure 45. Set Device ATM_Address for A-CPSW*

**Important**

Entering SET DEVICE ATM_ADDRESS performs a reset of the A-CPSW module. If there are any unsaved settings, you will be prompted if you want to proceed with reset. In this case:

- If you choose *yes*, the unsaved settings will be lost.
- If you choose *no*, the ATM address will not be set.

Therefore, before entering the ATM address, save your current settings with the SAVE DEVICE or SAVE ALL commands.

## 4.3 SNMP Agent

In addition to the command line interface, the A-CPSW provides an SNMP agent that is accessible using IP. The SNMP agent allows the 8260 ATM switching system to be managed via an SNMP manager. The IP component of the A-CPSW is accessible via:

- IP using the SLIP interface provided via the RS-232 interface on the A-CPSW module
- Classical IP over ATM as described in RFC 1577

  The IP over ATM client of the A-CPSW supports up to 64 concurrent IP over ATM connections. If the A-CPSW has 64 connections, and you try to establish a new connection, the oldest connection will be terminated and the new connection will be established.
- LAN emulation over ATM as described in ATM Forum's LAN emulation specification

  **Note:** IP using LAN emulation is provided with A-CPSW microcode V2.1.0.

The supported functions by the SNMP agent and how the A-CPSW can be managed using the Nways Campus Manager program is described in Chapter 7, "Nways Campus ATM Manager" on page 211. However, the following sections describe how you can configure the A-CPSW module to use IP and SNMP.

### 4.3.1 How to Configure IP over SLIP

The SLIP support on the A-CPSW module allows you to connect a workstation to the console port of the A-CPSW (directly or over a modem connection) and use IP over SLIP for communication between your workstation and the A-CPSW module. This will allow the workstation to use Telnet (to access command line interface) and TFTP (to perform UPLOAD and DOWNLOAD) to communicate with the A-CPSW.

To set up the A-CPSW to use the SLIP interface, you must perform the following steps:

1. Connect an ASCII terminal (or a workstation emulation of an ASCII terminal) to the console port on the A-CPSW module.
2. Log in to the A-CPSW module (as an administrator) from the ASCII terminal.
3. Use the SET TERMINAL BAUD command to set the transmission rate for the communication line to be used for the SLIP interface. The following example sets the transmission rate to 19200 bps:

```
8260ATM> set terminal baud 19200
```

4. Use the SET TERMINAL SLIP_ADDRESS command to set the IP address to be used by the A-CPSW over the SLIP connection. The following example sets 192.168.2.1 as the IP address to be used by the A-CPSW over the SLIP connection:

```
8260ATM> set terminal slip_address 192.168.2.1
```

5. Use the following SET TERMINAL CONSOLE_PORT_PROTOCOL command to switch the console port's operating mode to SLIP:

```
8260ATM> set terminal console_port_protocol slip
```

6. Configure the IP stack within the workstation with the IP address of the A-CPSW specified as the destination address.
7. Unplug the console cable from the ASCII terminal to the console port on the A-CPSW module.
8. Plug the cable from the IP workstation (or modem connecting the IP station to the A-CPSW) into the console port of the A-CPSW module.
9. Start the IP stack in the workstation and verify the connectivity between the workstation and the A-CPSW using PING.
10. Once the A-CPSW and workstation can communicate, you may use Telnet to log in as an administrator. From now on, your workstation can be used to manage the A-CPSW via the command line interface. You can also use TFTP to perform inband DOWNLOAD and UPLOAD between the workstation and the A-CPSW module.
11. When the SLIP connection is not required anymore, you can switch the operating mode to normal using the following command:

```
8260ATM> set terminal console_port_protocol normal
```

The normal mode allows the connection of ASCII terminals to the A-CPSW console port.

**Note:** An A-CPSW module reset will always restore the console port to normal. Also, in case of workstation inactivity for 20 minutes, the A-CPSW module's console port will return to normal. The workstation inactivity can be caused by the following:

- Unattended workstation
- Logout from Telnet
- Unplugging the workstation cable from the A-CPSW console port

The 20-minute timeout is built-in to A-CPSW and cannot be modified by the user.

### 4.3.2 How to Configure Classical IP over ATM

If you want to enable the 8260 ATM subsystem to use IP (over ATM), you must use the following commands to set the parameters for the A-CPSW module:

- **Set Device IP_Address**

  This command lets you assign an Internet Protocol (IP) address to the A-CPSW module and define the subnetwork mask used. A unique IP address must be assigned to each A-CPSW. An example is as follows:

```
8260A> set device ip_address atm 9.67.46.201 FF.FF.FF.F0
IP address and mask set
8260A>
```

*Figure 46. Set Device IP_Address for A-CPSW*

**Important**

The IP addresses in the format 10.*n.n.n* are reserved and cannot be used. Please refer to 4.9.1.4, “Internal Addressing” on page 113 for the discussion on the purpose of these reserved addresses for the A-CPSW module.

- **Set Device ARP_Server**

  This command lets you define the ATM address of an ARP (address resolution protocol) server. This server is used in a Classical IP over ATM network to map IP addresses to ATM addresses. An example is as follows:

```
8260A> set device arp_server
39.09.85.11.11.11.11.11.11.11.11.03.01.00.80.05.09.93.7C.00
ATM Address set
8260A>
```

*Figure 47. Set Device ARP_Server for A-CPSW*

- **Set Device Default_Gateway**

  This command allows you to set the IP address of the router that will be used to send and receive IP packets to stations that are not in the same network as the A-CPSW. The following example shows you how to set an IP address for the default gateway:

```
8260A> set device default_gateway 9.67.46.203
Default gateway set
8260A>
```

*Figure 48. Set Device Default_Gateway for A-CPSW*

### 4.3.3 How to Configure the 8260 Model Type

In addition to the IP parameters, the SET DEVICE commands allow you to specify the number of slots (10 or 17) for your 8260 ATM hub. An example for setting a 10-slot 8260 ATM hub is as follows:

```
8260A> set hub_number_of_slots 10
```

*Figure 49. Configuring Number of Slots for 8260*

**Note**

If you enter the SET HUB_NUMBER_OF_SLOTS 10 command for a 17-slot 8260, slots 11 to 17 will no longer be seen by the A-CPSW.

### 4.3.4 How to Configure the 8260 Clock, Name, Contact and Location

The SET DEVICE commands allow you to configure the following parameters for the A-CPSW module:

- **Set Clock**

  This command allows you to set the time, day and date for the internal clock of the A-CPSW. You need to set it only once, because the clock has its own battery and will continue to operate even in the case of a power failure in the hub. An example of using this command is as follows:

```
8260A> set clock 15:20 1995/09/26
clock set
```

  This command sets the clock to 3:20 p.m., September 26th, 1995.

- **Set Device Name**

  This command allows you to assign a name to the A-CPSW module. It is recommended that you assign a unique name to each A-CPSW in the network. It is also recommended that the A-CPSW device name be used as the console prompt. This name can be up to 31 characters long.

  The following command assigns the device name of 8260A to the A-CPSW:

```
8260A> set device name 8260A
Device name set
```

- **Set Device Contact**

  This command lets you enter information on the name of the person or qualified personnel responsible for maintaining the 8260 into which the A-CPSW is installed. The name can be up to 78 alphanumeric characters long and is used as the SNMP MIB 2 variable Syscontact. An example of this command is as follows:

```
8260A> set device contact
Enter text:
Mohammad Shabani
8260A>
```

- **Set Device Location**

  This command allows you to describe the location of the 8260 into which the A-CPSW is installed. The location can be up to 78 alphanumeric characters long and is used as the SNMP variable Syslocation. An example of this command is as follows:

```
8260A> set device location
Enter text:
ITSO LAB, Raleigh
8260A>
```

- **Set Device Diagnostics**

  This command lets you enable and disable diagnostics each time the A-CPSW module starts up or is reset. ATM diagnostics are enabled by default and by issuing the following command you can make the A-CPSW bypass diagnostics and boot faster:

```
8260A> set device diagnostics disable
```

To permanently save your changes, enter the following command:

```
8260A> save device
```

### 4.3.5 How to Display the A-CPSW Device Settings

To display all the configuration information of the A-CPSW, including the IP, ATM and device parameters, you must use the SHOW DEVICE command. An example is as follows:

```
8260A> show device
8260 ATM Control Point and Switch Module
Name : 8260A
Location :
ITSO LAB, Raleigh

For assistance contact :
Mohammad Shabani

Manufacture id: VIME
Part Number: 58G9470 EC Level: C38846
Boot EEPROM version: x.1.0.2
Flash EEPROM version: x.1.0.4
Restart count: 118

A-CPSW
--------------------------------------------------------------------------------
 ATM address: 39.09.85.11.11.11.11.11.11.11.11.03.04.40.00.00.82.60.A1.01

 > Subnet atm:
 IP address: 9.67.46.201
 Subnet mask: FF.FF.FF.F0

Default Gateway :
MORE...
--------------------------------------------------------------------------------
 IP address: 9.67.46.203

ARP Server:
--------------------------------------------------------------------------------
 ATM address: 39.09.85.11.11.11.11.11.11.11.11.03.01.00.80.05.A9.93.7C.00

Diagnostics: ENABLED
```

*Figure 50. Output from Show Device Command*

### 4.3.6 How to Configure SNMP Parameters

If you want to manage the ATM subsystem in an 8260 hub from an SNMP workstation, you must set the following parameters for the A-CPSW module:

- **Set Community**

  This command allows you to create a community table of SNMP management stations that can access information in the A-CPSW module. The management stations communicate with the A-CPSW using the Classical IP over ATM protocol.

  This community table can have up to ten entries and identifies each SNMP station by its IP address and community name. Each of these stations can have one of the following access rights:

  **read_only** — Allows the specified SNMP manager to display A-CPSW configuration parameters.

  **read_write** — Allows the specified SNMP manager to display and modify A-CPSW configuration parameters.

  **trap** — A-CPSW alerts will be sent to the station whose IP address you specify.

**read_trap** Allows the specified SNMP manager to display A-CPSW configuration parameters, and also to receive traps.

**all** Allows the specified SNMP manager to display and modify A-CPSW configuration parameters, and also to receive traps.

The following example defines an SNMP manager with the IP address of 9.67.46.203 and the community name of public and assigns read_write access:

```
8260A> set community public 9.67.46.203
Entry set.
8260A>
```

*Figure 51. Set Community for A-CPSW*

To permanently save your settings, enter the following command:

```
8260A> save community
```

- **Show Community**

  This command allows you to display the contents of the community table you created with the set community command. Figure 52 shows an output of this command:

```
8260A> show community
Index Community_Name  IP_Address            Accesses
------------------------------------------------------------
   1  public          9.67.46.205       Read - Write - Trap
   2  public          9.67.46.45        Read - Write - Trap
   3  public          9.67.46.203       Read - Write - Trap
7 entries empty.
8260A>
```

*Figure 52. Output from Show Community Command*

- **Clear Community**

  This command deletes an entry in the community table. You must verify the index number, as shown in Figure 52 before deleting it. You can also delete all entries using *all* instead of the index number. An example is as follows:

```
8260A> clear community 3
Entry cleared.
8260A>
```

*Figure 53. Clear Community Command*

- **Set Alert**

  This command allows you to enable or disable the function for sending alerts via SNMP traps to the A-CPSW local console and network management stations. These alert features are the following:

**Authentication** The A-CPSW sends an alert when an unauthorized SNMP access is attempted to A-CPSW.

**Change** The A-CPSW sends an alert when any configuration change is made in the ATM subsystem of the 8260.

**Hello** When the A-CPSW is activated, it sends one trap every minute, up to 255 times until a valid SNMP message is received.

The following example shows how you enable the sending of *authentication* traps to the SNMP manager:

```
8260A> set alert authentication trap
```

The following example shows you how to disable the sending of *change* traps:

```
8260A> set alert change notrap
```

You can also send the alerts to the console terminal of the A-CPSW. An example is as follows:

```
8260A> set alert hello trap display
```

The default setting for all three types of alerts is NOTRAP. The default setting for the terminal is NODISPLAY, which means that no traps will be sent to the console of the A-CPSW.

To permanently save your settings, enter the following command:

```
8260A> save alert
```

- **Show Alert**

  This command allows you to display the current alert settings. An example is as follows:

```
8260A> show alert
Alert AUTHENTICATION set to NOTRAP NODISPLAY
Alert     CHANGE     set to NOTRAP NODISPLAY
Alert     HELLO      set to NOTRAP NODISPLAY
8260A>
```

*Figure 54. Show Alert Command*

## 4.4 Accessing the A-CPSW Using Telnet

After you have configured the IP parameters of the A-CPSW using the console port, it is possible to have inband access through Telnet to the A-CPSW module.

The A-CPSW module's remote login feature allows you to log on to an A-CPSW module from a remote A-CPSW console or network workstation that supports Telnet.

To log on to a remote A-CPSW module using the local A-CPSW console, you must enter its IP address (as configured with the SET DEVICE IP_ADDRESS command). An example is as follows:

```
8260A> telnet 9.67.46.202
```

Once you log on to the remote A-CPSW, you are prompted to enter the correct password. If you enter as the administrator, all the commands that you enter locally affect the remote module.

You can remotely log on to only one A-CPSW at a time. If you have already started a remote A-CPSW session and want to connect to another remote A-CPSW module, you must first log off the active remote session.

To log off from the remote A-CPSW session, you must log off using the LOGOUT command, which disconnects you from the remote connection and reconnects you to the local A-CPSW module. An example is as follows:

```
8260A> logout
Bye
Remote session completed
8260A>
```

*Figure 55. Logging Off from a Remote A-CPSW Session*

You will not be allowed to log off if there are any unsaved changes to the configuration of the A-CPSW. However, you may log off from the A-CPSW without saving the changes by, for example, terminating the Telnet session abnormally. In this case any unsaved changes are in effect, but will be lost if the remote A-CPSW is reset. To save these changes, you must reestablish the remote session and save them using the SAVE command.

## 4.5 ATM Physical Layer Support

The ATM physical layer interfaces supported in the 8260 ATM switching subsystem are dependent on the type of media modules that are installed in your 8260. The following interfaces are currently supported by the 8260 ATM media modules.

### 4.5.1 ATM 4-Port 100-Mbps (A4-FB100) Module

This module provides four 100-Mbps TAXI interfaces using 62.5 micron multimode fiber. The physical layer for this module follows the FDDI PMD specifications. The fiber connectors for this module can be either MIC duplex connectors or SC connectors. The type of the connector is specified at the time of ordering the module using the following part numbers and feature codes:

*Table 17. ATM 4-Port 100-Mbps Module*

| Module Name | Part Number | Feature Code |
|---|---|---|
| ATM 4-Port 100-Mbps module with MIC connectors | 58G5845 | 5004 |
| ATM 4-Port 100-Mbps module with SC connectors | 58G5845 | 5104 |

### 4.5.2 ATM 155-Mbps Flexible Concentration (ATMflex) Module

This module provides two slots for installing daughter cards. The physical interfaces supported for these two interfaces can be fiber or copper cable, depending on the type of daughter cards used. The following table shows part number and the feature codes for the daughter cards that are available for this module and its I/O daughter cards.

*Table 18. ATM 155-Mbps Flexible Concentration Module and I/O Cards*

| I/O Daughter Card | Part Number | Feature Code |
|---|---|---|
| ATM 155-Mbps Flexible Concentration Module | 58G9863 | 5002 |
| ATM 155-Mbps 1-port Fiber Multimode I/O Card | 58G9667 | 8800 |
| ATM 155-Mbps 1-port Fiber Monomode I/O Card | 58G9855 | 8801 |
| ATM 155-Mbps 1-port UTP/STP I/O Card | 58G9856 | 8801 |
| ATM 4-Port 100-Mbps module with SC connectors | 58G5845 | 5104 |

The following table shows the type of connectors and cables supported by the various I/O daughter cards on the ATM 155 Flexible Concentration module.

*Table 19. Connectors and Cable Types for ATMflex Module*

| I/O Daughter Card | Connector Type | Cable Type |
|---|---|---|
| ATM 155-Mbps 1-port Fiber Multimode I/O Card | SC | Multimode Fiber |
| ATM 155-Mbps 1-port Fiber Monomode I/O Card | SC | Singlemode Fiber |
| ATM 155-Mbps 1-port UTP/STP I/O Card | Shielded RJ45 | 100 ohm category 5 UTP |

### 4.5.3 ATM 12-Port 25 Mbps Concentration Module

This module provides twelve 25-Mbps interfaces using UTP/STP cabling as specified in the ATM Forum specifications for a 25-Mbps interface. The type of interface provided on this module is shielded RJ-45. The following table shows the part number and the feature code that can be used to order this module.

*Table 20. ATM 25 Mbps 12-Port RJ45 Concentration Module*

| Module Name | Part Number | Feature Code |
|---|---|---|
| ATM 25 Mbps 12-Port RJ45 Concentration Module | 51H3828 | 5012 |

**Note:** The ATM TR/Ethernet LAN Bridge module does not provide an external ATM connection. The ATM connection on this module is via the backplane connection.

## 4.6 ATM Connections

The IBM 8260 A-CPSW module provides support for the following types of connections:

- Permanent Virtual Connection (PVC)

  Note that PVCs are not supported when they span over NNI links using A-CPSW V2.0.4. However, they are supported over NNI links using A-CPSW V2.1.0. Also, PVCs are supported over SSI links using both A-CPSW V2.0.4 and A-CPSW V2.1.0. For information about NNI links, please refer to 4.9, “Topology and Route Selection (TRS) Services” on page 108.

- Permanent Virtual Path (PVP)

  Note that PVPs are not supported when they span over NNI links using A-CPSW V2.0.4. However, they are supported over NNI links using A-CPSW V2.1.0. Also, PVPs are supported over SSI links using both A-CPSW V2.0.4 and A-CPSW V2.1.0. For information about NNI links, please refer to 4.9, “Topology and Route Selection (TRS) Services” on page 108.

- Switched Virtual Connection (SVC)

**Note:** Switched virtual path (SVP) is not supported by the IBM 8260.

### 4.6.1 Supported VPI and VCI Range

The virtual path identifiers (VPIs) and virtual channel identifiers (VCIs) supported by the A-CPSW module are:

- VPIs are in the range 0-15 for 100-Mbps and 155-Mbps Media Modules.
- VPIs are in the range 0-3 for 25 Mbps Media Module 12 ports.
- VCIs are in the range 32-1023.

Note that certain workstation adapters have limited addressing capability as far as the supported VPIs and VCIs are concerned. These limitations are based on the number of bits in the ATM header that are recognizable by the workstation adapter and are defined in the ILMI packets exchanged by the adapter. The A-CPSW module dynamically adjusts the supported VPI and VCI range on a port to the capability of the attached workstation at the ILMI exchange.

### 4.6.2 Supported Virtual Connection Types

Table 21 shows the type of virtual connections supported by the IBM 8260.

*Table 21. Supported Connection Type by the A-CPSW Module*

| Connection Type | Supported? |
|---|---|
| Unidirectional point-to-point | No |
| Bidirectional point-to-point with symmetric bandwidth | Yes |
| Bidirectional point-to-point with asymmetric bandwidth | No 1 |
| Unidirectional point-to-multipoint | No |
| Bidirectional point-to-multipoint | Yes |
| multipoint-to-multipoint | No |

1 If a call setup request with asymmetrical bandwidth requirement is received, the A-CPSW module will establish the call with the higher peak rate used for both directions.

### 4.6.3 Maximum Number of Connections Supported

The maximum number of supported connections depends on their type (point-to-point or point-to-multipoint) and for point-to-multipoint connections on the number of parties per connection. The following are the rules you can use to determine the number of connections supported in your environment:

- The IBM 8260 has 8,192 connection control blocks.

  **Note:** The A-CPSW module with 16 MB of memory in conjunction with microcode Version 2.1 supports 12,288 control blocks. For more details about the maximum number of connections supported by the new modules and the microcode V2.1.0, please refer to 4.18.3, “Increased Number of Supported ATM Connections” on page 187.

- Each point-to-point connection requires two control blocks.
- Each party on a point-to-multipoint connection requires one connection control block.
- The maximum number of point-to-point connections supported by an A-CPSW is 4,096.

  **Note:** The A-CPSW module with 16 MB of memory in conjunction with microcode Version 2.1 supports 6,144 point-to-point connections.

- The maximum number of point-to-multipoint trees supported by an A-CPSW is 127.
- The maximum number of parties supported by an A-CPSW for all the point-to-multipoint trees is 2048.
- The maximum number of PVCs supported by an A-CPSW is 100.
- The maximum number of point-to-point connections per 8260 media modules is 992.
- The maximum number of parties for point-to-multipoint connections per 8260 media modules is 992.
- The maximum number of point-to-point connections per 8260 port is 992.

- The maximum number of virtual path (VP) connections supported per port is 16. This means that the A4-FB100 modules support 64 VP connections and the ATMflex modules support 32 VP connections.

### 4.6.4 How PVCs Are Supported

To support PVCs, the A-CPSW module maps them internally onto SVCs. This allows the PVC to be automatically reestablished using an alternate path in case of a link or node failure on the original path supporting the PVC. In addition, the parameters specified for the setting of the PVCs are saved in the NVRAM of the original 8260 to provide automatic reestablishment of the PVC after the A-CPSW power off or reset condition.

Note that the information about PVCs is only stored in NVRAM after the connection is activated. This is to ensure that only the current and valid PVCs are restarted.

When an 8260 is restarted and an SVC is to be established before all the PVCs have been reestablished, a problem could occur if that SVC is allocated a label that is owned by one of the PVCs. To overcome this problem, the A-CPSW module always checks to see if a label is not reserved by a PVC before allocating it to an SVC.

**Note:** The PVC management between A-CPSW V2.1.0 and previous releases of the A-CPSW are not compatible. Therefore, you will not be able to activate a PVCs defined between A-CPSW V2.1.0 and previous releases. Therefore, to be able to define PVC between your A-CPSW modules in a network that includes A-CPSW module V2.1.0, you must upgrade all the A-CPSW modules to V2.1.0. During the upgrade of the A-CPSW module to V2.1.0, a conversion function is provided that automatically converts the previous PVC definitions to V2.1.0.

### 4.6.5 How to Configure PVCs

PVCs can be set up using the command line interface or the Nways Campus manager program. The following example shows how to configure a PVC for the configuration shown in Figure 56 on page 93.

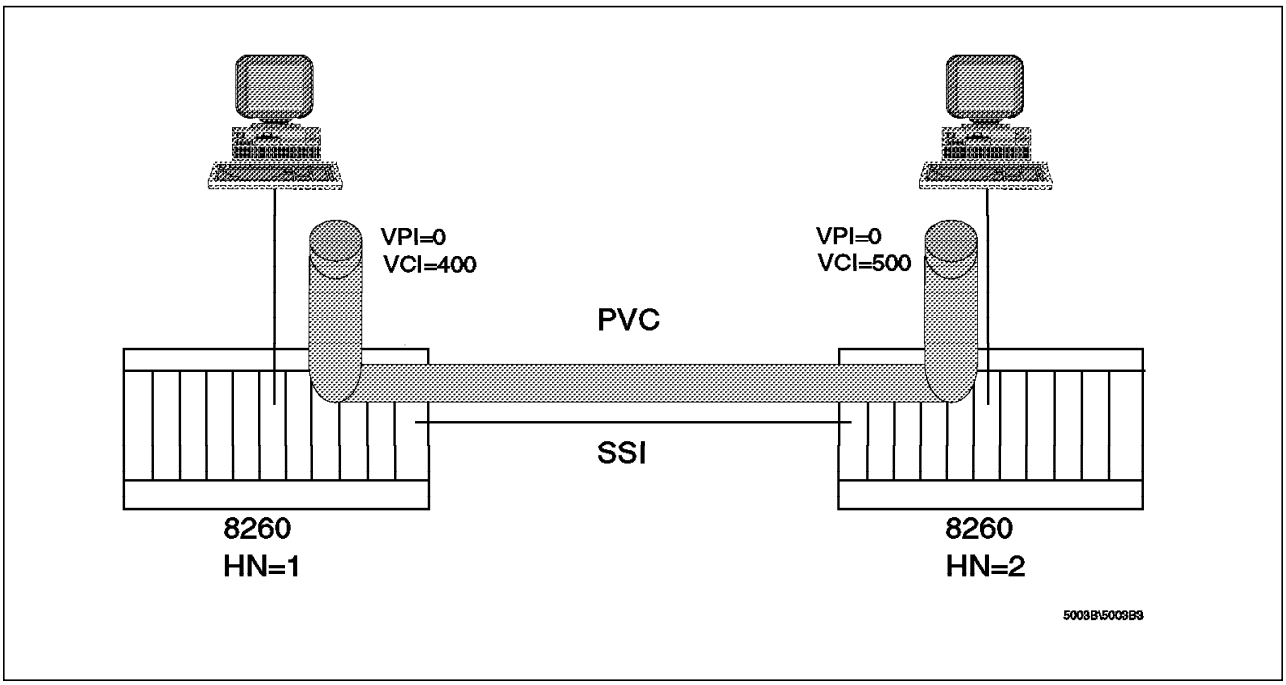



*Figure 56. Example PVC Configuration*

```
8260A>set pvc 5.1 100 6.1 2 channel 0.400 0500 best_effort
```

Note that in this example, we have chosen the following attributes for the PVC:

- Slot.port on the local 8260 = 5.1
- Slot.port on the remote 8260 = 6.1
- PVC_id = 100

  This is an arbitrary number that you can use to identify the PVC on various displays.
- Remote hub identifier = 2

  This identifies the hub number (HN) of the remote hub (the hub on which the PVC terminates) within the cluster.
- VPI/VCI on the local hub:
  - VPI = 0
  - VCI = 400
- VPI/VCI on the remote hub:
  - VPI = 1
  - VCI = 500
- PVC type = best_effort

The VPI/VCI values chosen for each port must be free at the time of defining the PVC. You can find out the VPI/VCI values that are currently allocated to the other connections on the port by using the ATM Campus manager for AIX. If you are not sure which VPI/VCI is available for allocation, you may use the following command, which will allow the A-CPSW to select an available VPI/VCI pair that is assigned for the PVC on each port:

```
8260A>set pvc 5.1 100 6.1 2 channel * * best_effort
```

You can display the configuration information about a specific PVC or all the PVCs using the SHOW PVC command. The following example shows the output that will be displayed as a result of this command:

```
8260A> set pvc 5.1 100 6.1 2 channel 0.400 0.500 best_effort
PVC set and started.
8260A> show pvc all

       Local end point        | Remote end point  |
------------------------------|-------------------|
 Port   id   type   Vpi/Vci   | Port Vpi/Vci   HNb|   role   |QOS| Status
------------------------------|-------------------|----------|---|--------
5.01    100 PTP-PVC   0/400   |6.01    0/500     2| Primary  | BE|Active
6.01   1001 PTP-PVC   0/500   |5.01    0/400     1|Secondary | BE|Active
8260A>
```

You may display additional information about the configuration of the PVC by using the verbose parameter in the SHOW PVC command as shown in the following example:

```
8260A> show pvc 5.1 100 verbose

       Local end point        | Remote end point  |
------------------------------|-------------------|
 Port   id   type   Vpi/Vci   | Port Vpi/Vci   HNb|   role   |QOS| Status
------------------------------|-------------------|----------|---|--------
5.01    100 PTP-PVC   0/400   |6.01    0/500     1| Primary  | BE|Active
Remote address : 39.09.85.11.11.11.11.11.11.11.11.01.02
Quality of Service : Best Effort.
Last Active Date : 16:38:55 2 Apr 96 (0 failures)
8260A>
```

### 4.6.6 How PVPs Are Supported

PVPs are supported through the PVCs.

### 4.6.7 How to Define PVPs

PVPs can be set up using the command line interface or the Nways Campus manager program. The following example shows how to configure a PVC for the configuration shown in Figure 57 on page 95.

```
8260A>set pvc 6.1 100 7.1 2 path 14 15 best_effort
```

Note that in this example, the following attributes are defined for PVP:

- Slot.port on the local 8260 = 6.1
- Slot.port on the remote 8260 = 7.1
- PVP_id = 100

  This is an arbitrary number that you can use to identify the PVP on various displays.

- Remote hub identifier = 2

  This identifies the hub number (HN) of the remote hub (the hub on which the PVP terminates) within the cluster.
- VPI on the local hub = 14
- VPI on the remote hub = 15
- PVP type = best_effort

The VPI values chosen for each port must be free at the time of defining the PVP. You can find out the VPI values that are currently allocated to the other connection on the ports by using the ATM Campus manager for AIX. If you are not sure which VPI is available for allocation, you may use the following command which will allow the A-CPSW to select the VPI value that is assigned for the PVP on each port:

```
8260A>set pvc 6.1 100 7.1 2 path * * best_effort
```

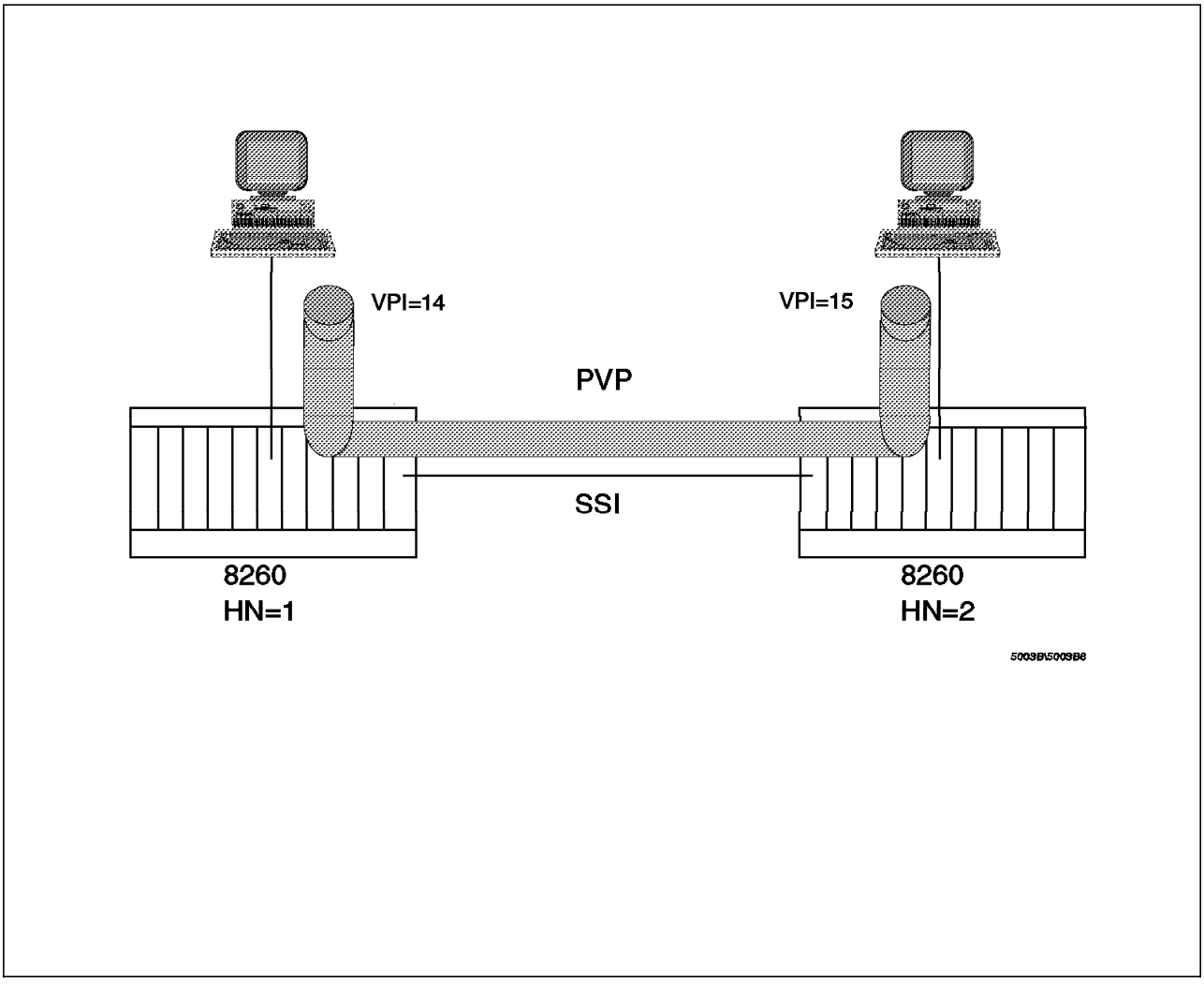



*Figure 57. Example PVP Configuration*

You display configuration information about PVPs using the SHOW PVC command, as described in 4.6.5, “How to Configure PVCs” on page 92.

The following is an example of the output display for a PVP.

```
8260A> set pvc 6.1 100 7.1 2 path 14 15 best_effort
PVC set and started.
8260A> show pvc all

       Local end point        | Remote end point  |
------------------------------|-------------------|
 Port   id   type   Vpi/Vci   | Port Vpi/Vci   HNb|   role   |QOS| Status
------------------------------|-------------------|----------|---|---------
6.01    100 PTP-PVP  14/*     | 7.01  15/*       2| Primary  | BE|Active
8260A>
```

## 4.6.8 How a VPI/VCI Is Allocated to SVCs

For virtual connections (both SVCs and PVCs), the VPI/VCI allocation is performed on a per-port basis.

For an SVC, it′s always the A-CPSW that will allocate any VPI/VCI used by the SVC at each segment of the connection. The procedure for allocating a VPI/VCI for the SVCs is based on the following considerations:

- The VPI value is always 0 on the UNI and SSI links. On the NNI links, the VPI value is as defined in the SET LOGICAL_LINK command for that NNI link. For information on NNI and SSI links, please refer to 4.9, “Topology and Route Selection (TRS) Services” on page 108.
- The VCI values 0 through 31 are always reserved for ITU and ATM Forum.
- The 127 odd-numbered VCIs between 32 and 286 (that is VCI 33, 35, 37, etc., through 285) are reserved for point-to-multipoint connections.
- The even-numbered VCIs between 32 and 286 (32, 34, 36, etc., through 286) and all the VCIs from 287 up to and including 1023 can be used by the SVCs.
- The VPI/VCI allocation algorithm is to increment the highest previously allocated VCI value and verify that this value is not in use by a VC or a VP connection. When incrementing the VCI value, the following considerations apply:
  - If the connection is a point-to-point connection and the currently highest allocated VCI is less than 286, the VCI value is incremented by two.
  - If the connection is a point-to-point connection and the currently highest allocated VCI is 286 or higher, the VCI value is incremented by one.
  - If the connection is a point-to-multipoint connection, the VCI value is incremented by one.
  - If the currently allocated VCI value for the point-to-point connections is 1023, the next VCI allocated will be the first free VCI value starting from 32.
  - If the currently allocated VCI value for the point-to-multipoint connections is 285, the next VCI allocated will be the first free VCI value starting from 33.

For PVCs, as described in 4.6.5, “How to Configure PVCs” on page 92, you can either specify the VPI/VCI values allocated to the PVC at the two ports, which are the endpoints of the PVC, or you may leave it to the A-CPSW to select the VPI/VCIs which are allocated. If you choose the latter, the A-CPSW will use the algorithm described for SVCs to allocate the VPI/VCI values. For the intermediate links on a PVC, it is always up to the A-CPSW to allocate the VPI/VCI values using the previous algorithm.

For PVPs, as described in 4.6.7, “How to Define PVPs” on page 94, you can either specify the VPI value allocated to the PVP at the two ports, which are the endpoints of the PVP or you may leave it to the A-CPSW to select the VPI/VCIs which are allocated. If you choose the latter, the A-CPSW will increment the highest previously allocated VPI and check to see if this value is not already in use. When the VPI value reaches the upper bound of VPI (15), the next VPI value wraps to 0. For intermediate links on a PVP, it is always up to the A-CPSW to allocate the VPI values using the previous algorithm.

### 4.6.9 How Point-to-Multipoint Connections Are Supported

To support point-to-multipoint connections, one cell destined for multiple ports occupies only one cell location in the shared switch memory. However, multiple output queues (one per media module) point to that one cell location. When the multicast cell arrives at the top of the output queue, it is sent to the output module. Within the output module, if the point-to-multipoint connection spans over multiple ports, then the multicast cell is replicated as required. The switch keeps track of when the last output port has transmitted the cell, thereby, allowing its memory locations to be freed. This technique minimizes the amount of memory space required for multicast messages.

Please refer to 4.9.1.3, “Support for Point-to-Multipoint Connections” on page 113 for a description of how the point-to-multipoint tree is set up.

## 4.7 ATM Signalling

To establish switched connections, the A-CPSW module supports the following signalling protocols:

- ATM Forum 3.0 (ITU Q.93B-based) for UNI 3.0 stations
- ATM Forum 3.1 (ITU Q.2931-based) for UNI 3.1 stations
- ATM Forum P-NNI phase 0 (Interim Inter-Switch Protocol) for NNI connections

The A-CPSW supports registration of endsystems and identification of the UNI version supported by the endsystem using the following procedures:

1. If the endsystem attached to the 8260 ATM media module supports ILMI, the registration will be done using the SNMP GET and SET commands over the SAAL, as specified in the ATM Forum′s ILMI process defined in UNI specifications. Using ILMI, the endsystem attached to the 8260 will first query the A-CPSW to obtain the ATM network prefix (13 octets). The endsystem will append its own 7-octet endsystem identifier (ESI) to the ATM network prefix and will register the resulting ATM address (20 bytes) with the A-CPSW.

   The endsystems connected to an 8260 port can be either UNI 3.0 or UNI 3.1. For endsystems that support ILMI, they may or may not support automatic detection of the UNI version via the ILMI MIB.

If the endsystem connected to the 8260 port supports UNI version identification via ILMI, you must configure the ATM port on the 8260 using the following command:

```
8260A> SET PORT slot.port ENABLE UNI NORMAL_ILMI
```

However, if the endsystem does not support UNI version identification via ILMI (although it supports ILMI), then you must configure the ATM port on the 8260 using the following command:

```
8260A> SET PORT slot.port ENABLE UNI ILMI_FORCED_SIG_3_0 or
8260A> SET PORT slot.port enable UNI ILMI_FORCED_SIG_3_1
```

2. In case the endsystem attached to the 8260 port does not support ILMI, you can provide the station address to the A-CPSW using the A-CPSW console. To do so, you must do the following:

   a. Define the port as not supporting IMLI using the following command:

   ```
   8260A> SET PORT slot.port ENABLE UNI NO_ILMI_SIG_3_0 or
   8260A> SET PORT slot.port ENABLE UNI NO_ILMI_SIG_3_1
   ```

   **Note:** The previous command is also used to specify if the port is using UNI 3.0 or UNI 3.1 signalling.

   b. Define the endsystem identifier (ESI) of the station to the A-CPSW using the following command:

   ```
   8260A> SET ATM_ESI slot.port xx.xx.xx.xx.xx.xx.
   ```

   Where xx.xx.xx.xx.xx.xx. specifies the ESI field (bytes 13 through 19) of the ATM workstation attached to the port.

**Notes**

- SSI links accommodate UNI 3.0 and UNI 3.1 calls.
- For NNI links, the UNI version must be specified at *logical link* definition for the NNI link.
- Mixing UNI 3.0 logical links and UNI 3.1 logical links over the same physical port is permitted.

For information about SSI, NNI and logical links, please refer to 4.9, “Topology and Route Selection (TRS) Services” on page 108.

### 4.7.1 UNI 3.0 and UNI 3.1 Translation

Since the UNI 3.0 and UNI 3.1 are incompatible (that is, a UNI Version 3.0 endsystem will not be able to understand the information element sent by UNI 3.1 endsystem and vice versa), the IBM 8260 will *attempt* to make it possible for two endsystems using these incompatible UNI versions to communicate with each other. Note that this is done on a *best-effort* basis and there is *no guarantee* that it will succeed, since the success of this process is based on how much checking is done in the endsystem.

When a connection request is received between two stations using UNI 3.0 and UNI 3.1, the 8260 will perform a translation at the last possible point on the connection. The following is a description of if and when the translation is done by the 8260:

- If an 8260 receives a call setup from an SSI link, and that call setup is destined to leave that 8260 via another SSI link, no translation is done on the information element of the call setup.
- If an 8260 receives a call setup from a UNI link, and that call setup is destined to leave that 8260 via an SSI link, no translation is done on the information element of the call setup.
- If an 8260 receives a call setup from a UNI link and that call setup is destined to another station attached to this 8260, and the UNI version support by both stations is the same, no translation is done on the information element of the call setup.
- If an 8260 receives a call setup from a UNI link, that call setup is destined to another station attached to this 8260, and the UNI version supported by these stations is different, the information element of the call setup will be translated.
- If an 8260 receives a call setup from an SSI link, and that call setup is destined to a station attached to this 8260 via a UNI link, the information element of the call setup will be translated whether it is necessary or not.
- If an 8260 receives a call setup from an NNI link, and that call setup is destined to a station attached to this 8260 via a UNI link, and the UNI version of the logical link defined for the NNI link is different from the UNI version of the attached station, the information element of the call setup will be translated.
- If an 8260 receives a call setup from an NNI link and that call setup is destined to leave that 8260 via another NNI link, and the logical links defined for these NNI links have the same UNI version specified, no translation is done on the information element of the call setup.

Table 22 illustrates a summary of the previous rules.

*Table 22. 8260's UNI 3.0 to UNI 3.1 Translation Process*

| Receive From | Send To | Translation Done |
|---|---|---|
| SSI | SSI | None |
| UNI | SSI | None |
| UNI x | UNI x | None |
| SSI | UNI | Yes |
| UNI x | UNI y | Yes |
| SSI | NNI | Yes |
| UNI | NNI | Maybe |
| NNI | UNI | Maybe |
| NNI | NNI | Maybe |

### 4.7.2 Maximum Number of Registered Stations

The number of stations that can register with an A-CPSW is 1,024.

### 4.7.3 Displaying Registered ATM Addresses

You may display the endsystem identifier of the ATM addresses that are registered with an 8260. You can do this using the SHOW ATM_ESI command. The format of this command is as follows:

```
8260A> SHOW ATM_ESI slot.port|all [DYNAMIC|STATIC]
```

The meaning of the parameters are as follows:

**slot.port|all** Specifies whether you wish to display the registered addresses for a specified port or all of the ports on the 8260.

**DYNAMIC** This parameter allows you to display the registered addresses learned dynamically using the ILMI procedure.

**STATIC** This parameter allows you to display the addresses configured statically using the SET ATM_ESI command.

The following is an example of displaying all the dynamic addresses registered with an 8260:

```
8260A> show atm_esi all dynamic
Port   ATM_ESI           Type
---------------------------------------
 5.01 40.00.00.60.00.01 dynamic
 6.03 40.00.00.82.81.C1 dynamic
13.02 70.00.80.00.90.01 dynamic
13.03 40.00.00.82.81.B1 dynamic
13.04 08.00.5A.99.0A.B3 dynamic
15.03 40.00.00.82.82.A1 dynamic
15.03 40.00.30.01.A1.2C dynamic
15.03 40.00.5A.95.F0.57 dynamic
15.04 70.00.80.00.90.02 dynamic
8260A>
```

## 4.8 Traffic Management

The following sections provide background information about traffic management in ATM networks and how it is implemented in the 8260.

### 4.8.1 Traffic Service Classes

The following traffic service classes are identified:

- Constant bit rate (CBR)

  CBR is designed for real-time applications that require a consistent, fixed quantity of bandwidth (leased-line emulation). CBR applications must be predictable, have short transit delay and be jitter-tolerant, and must send traffic into the network at a constant and continuous bit rate. Examples of CBR traffics are telephone and video.

- Variable bit rate (VBR)

VBR traffic can be either real-time traffic (rt-VBR) or non-real-time traffic (nrt-VBR).

In rt-VBR traffic the source input may vary in its data rate, but the network must provide short transit delay and guaranteed delivery service. Examples of this type of traffic are compressed video, compressed voice with silence suppression and high priority data applications.

nrt-VBR traffic is less dependent on traffic delay and jitter than rt-VBR (a network transit delay of a few seconds is not a problem here), but it too requires a guaranteed delivery service. An example of nrt-VBR traffic is MPEG-2 (standard for video compression and coding) video distribution.

- Unspecified bit rate (UBR)

  UBR is designed for non-real-time applications that do not require a particular quality of service. The applications that will use this service must be delay and loss tolerant. Delivery of data is done on a best-effort basis. Data can be sent into the network up to link speed, and if there is any congestion in any resource, then the network will throw the data away. Examples of UBR applications are Telnet, E-mail and other traditional data applications.

- Available bit rate (ABR)

  ABR is designed for non-real-time applications that will vary their transmission rates to make fair use of the available network resources. The idea is to use whatever bandwidth is available in the production network after other traffic using guaranteed bandwidth services have been serviced.

  Although it is an NRB service, it needs a minimum quality of service guarantee to service applications (for example, SNA) that require the interchange of acknowledgments to maintain sessions.

Table 23 summarizes the characteristics of various traffic types.

*Table 23. Types of Traffic*

| | **CBR** | **VBR** | **UBR** | **ABR** |
|---|---|---|---|---|
| **Connection Mode** | Connection-oriented | Connection-oriented | Connection-oriented | Connection-oriented |
| **Timing Sensitive** | Yes | Yes | No | No |
| **ATM Adaptation Layer** | AAL 1 | AAL 2 | AAL 3/4, 5 | AAL 5 |
| **Quality of Service** | Yes | Yes | No | Yes |
| **Reserved Bandwidth** | Yes | Yes | No | No |
| **Flow Control** | No | No | No | Yes |
| **Traffic Types** | Voice, Video | Compressed Voice, Video | Data | Data |

### 4.8.2 ATM Traffic Management

The function of traffic management is to control the use of network resources, optimizing network utilization and avoiding bottlenecks in the network. For reserved bandwidth, connection requests are only accepted if there are enough network resources. This control mechanism is called admission control. For nonreserved bandwidth, the ATM switch will accept cells when it has enough available bandwidth.

The following sections describe the different service classes that an ATM network offers for the transmission of different classes of traffic. It illustrates for each class the different methods that can be used for traffic control.

#### 4.8.2.1 Constant Bit Rate Service

This is a reserved bandwidth service, where a contract is established with the network. The user provides to the network a set of parameters describing the traffic to be sent over the connection. These are as follows:

- Cell Peak Rate (CPR)

  CPR indicates the aggregated traffic the user is going to transmit, irrespective of the value of the cell loss priority.

- Cell Loss Ratio (CLR)

  CLR indicates the admissible number of lost cells for a connection.

- Cell Delay Variation Tolerance (CDVT)

  CDVT indicates the maximum variation in the delay that the network must support.

This information is used for call admission control (CAC) and usage parameter control (UPC). If the network has a path that has adequate bandwidth available for the cell peak rate requested by the user and can support all the other traffic descriptors, it will accept the call. Once the call is accepted, it is the users responsibility to comply with its contracted service levels. However, to safeguard against the stations that may not conform to their traffic contract, the ATM switch needs to monitor the traffic before entering the network at the UNI interface, and should either tag the nonconforming traffic (using the *cell loss priority* both in the ATM cell header) before allowing it into the network or discard it. To do this, the UPC in the ATM switch will use a leaky-bucket.

#### 4.8.2.2 Variable Bit Rate Service

This is also a reserved bandwidth service. The operational principle is similar to constant bit rate, however, for VBR the user has to provide a sustained cell rate (SCR) and maximum burst size (MBS) in addition to the cell peak rate (CPR). The sustained cell rate is the transmit rate, while the MBS indicates the maximum number of bytes that will be sent during a single burst.

These parameters allow the network to do statistical multiplexing, allocating fewer resources than with CBR and thereby saving bandwidth.

#### 4.8.2.3 Unspecified Bit Rate Service

This is a nonreserved bandwidth (NRB) service, based on best-effort network delivery. When the network has enough available resources, it will share these between the NRB users. This traffic will use the currently available bandwidth for the moment. This available bandwidth includes any reserved bandwidth which is not currently used. To ensure that the bandwidth used by the UBR traffic does not exceed the available bandwidth, a congestion control mechanism is required for this service to ensure that the network can stop or slow down those nodes sending more data than can be managed.

The ATM Forum has not yet defined a flow-control mechanism for the unspecified bit rate (UBR) service.

#### 4.8.2.4 Available Bit Rate Service

ABR is a mixed reserved and nonreserved bandwidth service, similar to UBR, while ensuring a minimum guaranteed bandwidth. This minimum guaranteed bandwidth is called the minimum cell rate (MCR).

ABR congestion is avoided by implementing a flow control feedback loop which notifies sending stations when the network is congested.

Operation is credit or rate-based. For credit-based connections, stations receive a credit to transmit an agreed number of cells. This credit is renewed when the receiving node has enough buffers available. For rate-based connections, the network informs the user about the rate at which the user must transmit. If the network is beginning to be congested, the user is told to stop or slow down his or her transmission rate. The ATM Forum is still developing a standard scheme for ABR flow using a rate-based approach.

The most important parameters signalled for ABR connections are CPR and MCR, indicating the maximum transmission rate allowed and the minimum guaranteed transmission rate required.

ABR service is used mainly for data applications that require a minimum quality of service, such as SNA, to avoid session timeouts. Whereas, UBR can be used for data applications that do not require any quality of service, for example UDP/IP.

### 4.8.3 Service Classes Supported by 8260

The following quality of service (QoS) classes are supported by the A-CPSW module:

- Constant bit rate (CBR)
- Variable bit rate - real time (VBR-rt)

  This class is supported as CBR.
- Variable bit rate - non-real time (VBR-nrt)

  This class is supported as CBR.
- Unspecified bit rate (UBR)

**Note:** Currently, the 8260 does not support available bit rate (ABR).

#### 4.8.3.1 Constant Bit Rate Traffic Support by 8260

For constant bit rate (CBR) connections, the call admission control (CAC) is done by the control point (CP) function running in the A-CPSW module. Based on the peak rate requested for the call, the A-CPSW module looks at its precomputed routes for a path available to add the new connection. If no route is found or 85% of the link capacity would be exceeded by adding the new connection, the call is rejected.

To protect the overall network service, usage parameter control (UPC) polices constant bit rate traffic at the UNI by means of a leaky bucket set to the cell peak rate (CPR) per connection. The nonconforming traffic will be discarded. No usage parameter control is implemented at the NNI or SSI.

Cells from the CBR service are queued in a single CBR queue. The IBM 8260 has a priority mechanism that allows CBR traffic to be processed first, so that if there is any UBR traffic, it may be stopped to enable the processing of CBR cells. For the 8260, the cell loss ratio (CLR) is $10^{-27}$ if there is no UBR traffic, and $4 \times 10^{-25}$ if there is heavy concurrent UBR traffic. The cell transfer delay (CTD) through each 8260 is 35 microseconds.

On the IBM 8260, the constant bit rate traffic bandwidth reservation is symmetrical, that is, the same bandwidth is used both for transmitting and receiving information.

#### 4.8.3.2 Variable Bit Rate Traffic Support by 8260

On the IBM 8260, variable bit rate (VBR) is processed by constant bit rate services and so only the cell peak rate (CPR) is acted upon. The sustained cell rate (SCR) and maximum burst size (MBS) are ignored. Therefore, more resources than necessary for the connection might be allocated.

The over-allocation of resources is not a problem if the available bandwidth (allocated but not used or not allocated) is fully used by UBR traffic. However, a situation may arise where, due to over-allocation, new constant bit rate or variable bit rate calls are rejected despite unused (available) bandwidth. Currently this is a minor issue due to the high bandwidth available, but over time this may become more problematic.

On the IBM 8260, the variable bit rate traffic bandwidth reservation is symmetrical, that is, the same bandwidth is used both for transmitting and receiving information.

The admission control for variable rate traffic is equivalent to constant bit rate traffic. Traffic is policed at the UNI by means of a leaky bucket set to the cell peak rate (CPR) per connection, and the nonconforming cells will be discarded. No usage parameter control is implemented at the NNI or SSI.

#### 4.8.3.3 Unspecified Bit Rate Traffic Support by 8260

For the UBR service, the IBM 8260 uses a cell peak rate (CPR) at call setup that is equal to the link access rate. As the cell loss ratio (CLR) is not specified for the UBR service, the network does not have to reserve resources. However, IBM 8260 will reserve 51 Kbps for each UBR connection. The 8260 has some features to reduce cell loss and so avoid indiscriminate discarding of cells when congestion occurs.

Queuing and scheduling mechanisms have been implemented for the UBR service combined flow-control mechanism to ensure that this type of traffic will be able to access the network only if enough bandwidth is available to accommodate the reserved and nonreserved bandwidth traffic. Queuing is done per port and scheduling is done in a round-robin scheme. Hop-by-hop flow control is implemented by a port at SSI, UNI and switch internal interface with a global and selective backpressure mechanism.

The UBR service is not loss-free, therefore, a mechanism must be implemented to discard the cells in a controlled way. If the UBR service is used for LAN traffic, it can be understood that a single cell discarded in a packet of 192 cells forces the retransmission of the whole packet, resulting in unnecessary retransmissions.

The most efficient discarding process would be to discard packets rather than cells when congestion occurs. The IBM 8260 has implemented smart discard on congestion control with early packet discard (EPD). This is done by the IBM 8260 monitoring the utilization of its output buffers in each of the ATM media modules. If the output buffers reach the early packet discard (EPD) threshold, then the ATM ports on the congested media module will start to discard all the cells arriving for the new data packets to be sent over these congested ports. However, despite the early packet discard, the utilization of output buffers may reach a more critical threshold called a *firewall* threshold. This could be due to the sending of the residual cells belonging to the packets whose transmission had already started before the EPD threshold was reached. In this case, the ATM ports on the congested media module will start discarding all the arriving cells (including residual cells) to be sent over these congested ports. This process is used to prevent the UBR traffic from getting all the buffers available on the module, and thus adversely affecting the ability of the IBM 8260 to provide the appropriate service to the CBR/VBR traffic.

When the EPD threshold is reached, the hop-by-hop global backpressure will be started on the input ports feeding the congested module. Global backpressure means that when a concentration module gets congested (EPD threshold is reached), backpressure signals are sent on all the input ports to request the stations to temporarily suspend their UBR transmission. Transmission will be resumed after the module leaves the EPD threshold.

To perform global backpressure on the 100-Mbps ports, the IBM 8260 uses the physical symbols SR/RS (that is, XON and XOFF). On the 155 Mbps ports, the GFC protocol, as defined in ITU I361-I150, is used.

### 4.8.3.4 Available Bit Rate Traffic Support by 8260

What can we say here apart from saying that 8260 does not support it.

Table 24 provides you with the information about the traffic management functions supported by the IBM 8260.

*Table 24 (Page 1 of 2). Traffic Management Functions Support*

| Traffic Management Function | Supported |
|---|---|
| Connection admission control (CAC) | Yes |
| Usage parameter control (UPC) | No |
| Generic cell rate algorithm (GRCA) | No |

| Table 24 (Page 2 of 2). Traffic Management Functions Support | |
|---|---|
| **Traffic Management Function** | **Supported** |
| Network parameter control (NPC) | No |
| Cell loss priority (CLP) bit = CLP 0 \| 1 | No |
| Dual leaky bucket | No |
| Selective cell discarding | No |
| Traffic shaping mechanism | No |
| Network resource management | No |
| Cell tagging | No |
| Fast resource management | No |
| Feedback control mechanism | No |
| Explicit forward congestion indication (EFCI) | No |
| Per VC queuing | No |
| Full packet level discard | Yes |
| Dynamic discard threshold | No |
| Cell gapping | No |
| Random early detection (RED) | No |

### 4.8.4 Flow-Control Support in 8260

From the previous discussion we can conclude that in a network of 8260 ATM switches, the ATM connections are categorized into reserved bandwidth (RB) and nonreserved bandwidth (NRB) connections.

For RB connections, adequate bandwidth and switching capacity will be reserved to minimize cell loss. To safeguard against the stations that may not conform to their traffic contract, the 8260 will use a leaky bucket mechanism to monitor the traffic before entering the network at the UNI interface and will discard the nonconforming cells.

Traffic for the NRB connections will be transmitted across the network on a best-effort basis. To avoid cell loss for NRB traffic, IBM has implemented flow-control primitives in the ATM endstations and the 8260 ATM switch.

This flow-control procedure operates between adjacent nodes and allows each node to control the transmit rate of the adjacent node by periodically granting a credit (window) of cells that can be sent. The flow-control procedures are used on UNI and SSI interfaces, however, on a UNI, only the transmit rate of endstations is controlled as it is assumed that endstations have adequate resources to receive data from the ATM network, up to the line speed. On the SSI, the flow-control procedure is used in both directions.

**Note:** On the SSI, the term back pressure is used rather than flow control. As these functions are equivalent, we use the term flow control only.

The method used to implement the flow-control mechanism depends on the type of physical interface:

- 25-Mbps Links

  No flow control is implemented on 25-Mbps ATM interfaces.

- 100-Mbps Links

On the 100-Mbps ATM interfaces, line codes known as XON and XOFF (that are currently reserved in the ATM Forum specification) are used to notify the adjacent node when it can or cannot transmit data. During periods of congestion an XOFF code is sent between each cell. When there is no congestion the XON code is sent in the same manner. The continuous sending of XON is done to avoid deadlock situations in which a link (following the loss of an XON line code) remains permanently blocked.

- IBM 155-Mbps ATM Links

  On the 155-Mbps links, the control of NRB traffic is achieved by setting the second bit of the general flow control (GFC) field in ATM cells. The protocol is credit-based. If the 8260 ATM switch has available bandwidth for the NRB service, it sends a cell with the GFC credit bit set to one (x1xx). The station may send up to the number of credited cells and then wait for the next credit. Sending cells with the GFC credit bit set to zero (x0xx) has no effect on the counter, so when the 8260 ATM switch is congested for NRB service, it sends GFC=x0xx within all the cells.

In addition to using flow control on the UNI ports, the IBM 8260 is able to use the flow control primitives on the 100-Mbps and 155-Mbps ports configured as SSI ports.

The IBM 8260 grants the adjacent node permission to send when the number of remaining free buffers drops below a fixed threshold. When the adjacent station has no further credit to send, it must stop transmitting NRB cells without affecting the transmission of RB data cells. It may continue to transmit NRB traffic as soon as a new send-credit is received.

The 8260 ATM switch will request adjacent nodes to reduce their transmit rate when a receive buffer threshold is exceeded. It should be realized that all ports on the same blade share the same receive buffers. Therefore, care must be taken when mixing ports that are enabled for flow-control with ports that are disabled for flow-control on the same blade. When experiencing congestion, the 8260 ATM switch will only request some of the stations to slow down, while others continue sending.

For its 100-Mbps ports, the IBM 8260 uses the XON/XOFF codes in the same way on both UNI and SSI interfaces. The use of the GFC bit is slightly different on UNI and SSI when using a 155-Mbps link; the GFC (generic flow control) field is not present in the cell at SSI, and, therefore, the first 2 bits of the VPI field (in the same position of the SSI cell as the first 2 bits of GFC in the UNI cell) are used for flow-control. The second bit in the cell is used to implement the GFC protocol. The difference with the flow-control procedure used on the UNI is that the receipt of a cell with the GFC credit bit gives a credit to send 50 cells while the credit on the UNI is substantially lower (typically between 5 and 8).

### 4.8.5 Flow-Control Support in IBM ATM Adapters

- IBM 25-Mbps Adapter

  No flow-control is used on the IBM 25-Mbps ATM adapters.

- IBM 100-Mbps Adapter

  For traffic from workstations equipped with its 100-Mbps ATM adapters, IBM uses the XON/XOFF codes for flow-control.

- IBM 155-Mbps Adapter

The IBM 155-Mbps ATM adapters currently available do not use the GFC flow-control mechanism designed for them. It is IBM′s intention to make the GFC flow-control available in a later stage. Check the 155-Mbps ATM adapter technical publication for details.

### 4.8.6 Flow-Control Support in IBM 8281

During configuration of the IBM 8281, the user can specify whether generic flow-control should be used or not. The default setting is NO.

**Note:** The stand-alone version of the IBM 8281 connects to the ATM campus network using a 100-Mbps link, which uses XON/XOFF rather than using the GFC bits. The IBM 8260 ATM LAN bridge connects using a 155-Mbps link for which the GFC fields in the UNI cells are used for flow-control.

### 4.8.7 Flow-Control Support in IBM 8282

The IBM 8282 has not implemented the flow-control mechanism discussed. The IBM 8282 ignores XON/XOFF codes received from the adjacent ATM switch on its 100-Mbps uplink. Also, no flow-control is available on the 25-Mbps links. When the IBM 8282 receives more data than can be forwarded, cells will be buffered. If the buffering capacity is exceeded, data will be discarded.

## 4.9 Topology and Route Selection (TRS) Services

In order to be able to interconnect the IBM 8260 ATM subsystems together and also connect the IBM 8260 ATM subsystem to the ATM switches provided by other vendors, the IBM 8260 provides the topology and route selection (TRS) services. The TRS function enables you to route ATM signalling requests between ATM switches by using the following types of interfaces:

1. Switch-to-Switch Interface (SSI)

   SSI is based on IBM′s proprietary protocols and can be used to interconnect IBM 8260 ATM subsystem in a fully automatic way with minimal user definitions required.

2. Network Node Interface (NNI)

   NNI is based on the ATM Forum′s Interim Inter-Switch Signalling Protocol (IISP), also known as the Private Network-to-Network (PNNI) phase 0. This interface can be used for the following purposes:

   a. Interconnect IBM 8260 ATM subsystems together.

   b. Connect the IBM 8260 subsystem to ATM switches provided by the vendors who have implemented IISP.

An ATM campus network may consist of one or both of the above interfaces. To use these interfaces, a network of the IBM 8260 can be designed to consist of the following components:

- ATM cluster
- ATM subnetwork
- ATM campus network

The following sections describe these components and the architecture behind them. It also describes how to configure the IBM 8260 to take advantage of the different interfaces for interconnecting ATM switching subsystems.

### 4.9.1 ATM Cluster

All the IBM 8260 ATM switches within the same cluster share the same 12 bytes for their ATM address (byte 12 specifies the cluster number within the ATM subnetwork). Byte 13 of the ATM address is used to identify the hub number (HN) within a cluster. The hubs within a cluster are connected to each other via a switch-to-switch interface (SSI). There can be more than one SSI. There is no maximum number of links between any two hubs within a cluster. Parallel SSI links can be used to provide additional capacity, load-balancing, and backup links when interconnecting two 8260s. Figure 58 shows an example of a cluster of three 8260 ATM switches.

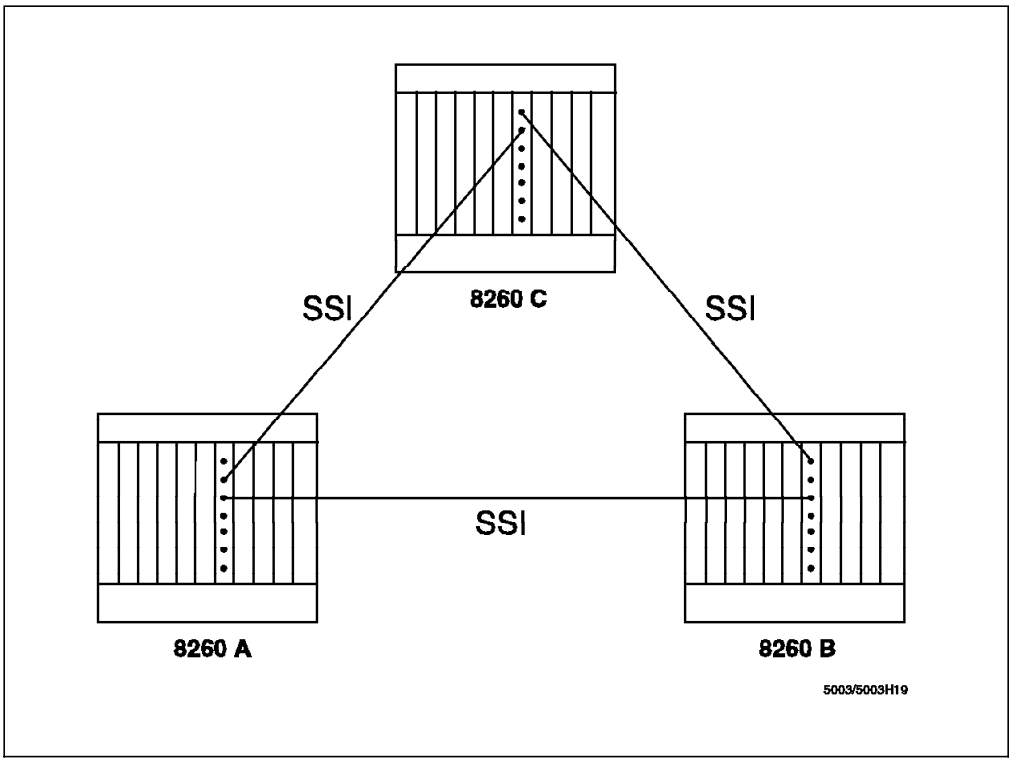



*Figure 58. Single Cluster of 8260s*

Each SSI link is assigned a unique six-bit *trunk identifier*. (This assignment is performed automatically by the IBM 8260 ATM subsystem and is transparent to the operator.) The SSI includes a protocol called *node identification eXchange* (NIX) that is used when an SSI link is enabled. Adjacent 8260 ATM subsystems within a cluster use this protocol to exchange hub numbers (HNs) and trunk identifiers. The 8260 ATM subsystem with the lower hub number (HN) becomes the *primary* and performs the role of the network-side with respect to the ATM signalling. The higher numbered 8260 ATM subsystem becomes *secondary* and performs the role of the user-side. The primary hub is responsible for allocating link resources (VPI/VCI and bandwidth) for the connection requests received over the SSI link. In fact, the primary 8260 ATM subsystem always allocates the label during the call setup process and updates the VPI/VCI table during the call connect process. The secondary never allocates the label and always updates the VPI/VCI table during the call connect process. This ensures that there will never be a label collision on an SSI segment when two connections are established simultaneously and in the opposite direction.

The two 8260 ATM subsystems agree that the SSI link is known to both of them by the hub number (HN) and the trunk identifier of the primary. The SSI protocol includes an exchange of link-state information among the 8260 ATM subsystems within a cluster to determine the best path through the ATM network. The protocol used for this process is described in the following section.

#### 4.9.1.1 Route Selection within a Cluster

The topology function of the A-CPSW lets every 8260 node within a cluster acquire a complete view of the topology of the cluster. This view is stored in a topology database within each 8260 node. A route computation algorithm called widest path OSPF (a variation of OSPF) is then used to compute the routes from the local node to every other node within the cluster. This algorithm will use the widest path (as opposed to the shortest path) available between the source and the destination (widest path algorithm is described in 4.9.1.2, “Widest Path OSPF” on page 111). The result of the route computation is stored in a widest path tree. Within each 8260, this tree describes the routes to all the other 8260s inside the cluster.

When a request is received for a connection between two UNI stations:

- If the destination is attached to the same 8260 node as the origin, the A-CPSW module will determine the port to which the call setup request is routed.
- If the destination is attached to another 8260 node within the same cluster, the TRS function within the hub attached to the station originating the connection will use its stored widest path tree to determine the route through the network between the origin and destination. This route, which is encoded in *route vector*, specifies the set of links and intermediate 8260 nodes that have to be used to set up the connection. The *route vector* is carried in the ATM SETUP message as an information element. (This is an extension to the UNI signalling protocol.)

  The use of precomputed widest path tree instead of computing the route when each connection request is received will result in a much faster connection setup. This is specially important when there are periods of high rates of connection requests.

  Each 8260 ATM subsystem will pass the SETUP message to the next 8260 ATM subsystem node along the route using the information contained in the route vector. This means that the intermediate nodes do not need to recompute the route to the destination, resulting in a simpler and faster processing of the SETUP message in these nodes.

  At the destination 8260 ATM subsystem, the A-CPSW will translate the destination address into the local port identifier and will then pass the SETUP message to the destination UNI station.
- If the destination lies outside the cluster, the path is computed to the NNI link by which the connection exits, and the call is forwarded to that connection. An independent path computation is performed in each subsequent cluster for forwarding the call setup message to the next cluster along the route to the destination, until the call setup reaches the final cluster. Within the final cluster, route computation will be used to determine the route and forward the setup message to the destination workstation. For detailed information about NNI links, please refer to 4.10, “Private Network Node Interface (P-NNI)” on page 128.

The A-CPSW modules within a cluster maintain the topology view of the cluster via exchanging *link state advertisement (LSA)* messages with each other. The LSA messages carry information about the state of the links interconnecting the hubs within the cluster, as well as the bandwidth currently available on those links. The LSA messages are generated when the state of the link changes or the available bandwidth on the link is changed by 5% (equal to 4 Mbps on a 100-Mbps link). The arrival of LSA messages will trigger the recomputation of the widest path tree.

### 4.9.1.2 Widest Path OSPF

Widest path OSPF in a network of 8260 ATM subsystems serves the same function as OSPF in a traditional IP router network. This is, it tries to find the best route between the nodes. However, unlike OSPF, widest path OSPF defines the cost of a link as the available bandwidth on a link, and the cost of a path is defined as the minimum of the cost of its component links. With widest path OSPF the route that will be chosen is the path with the highest cost or greatest width. This algorithm is used to determine the route for all the point-to-point connections. Figure 59 shows an example of this. In this example, the width of route A-1-C-2-B is 25, since 25 Mbps is the minimum bandwidth available on the component links of this path at the time of route computation. The width of route A-3-D-4-E-5-B is 45, since 45 Mbps is the minimum bandwidth available on the component links of this path at the time of the route computation. Therefore, in this scenario, the widest path OSPF will choose the route A-3-D-4-E-5-B to get from node A to node B because it is the widest path.

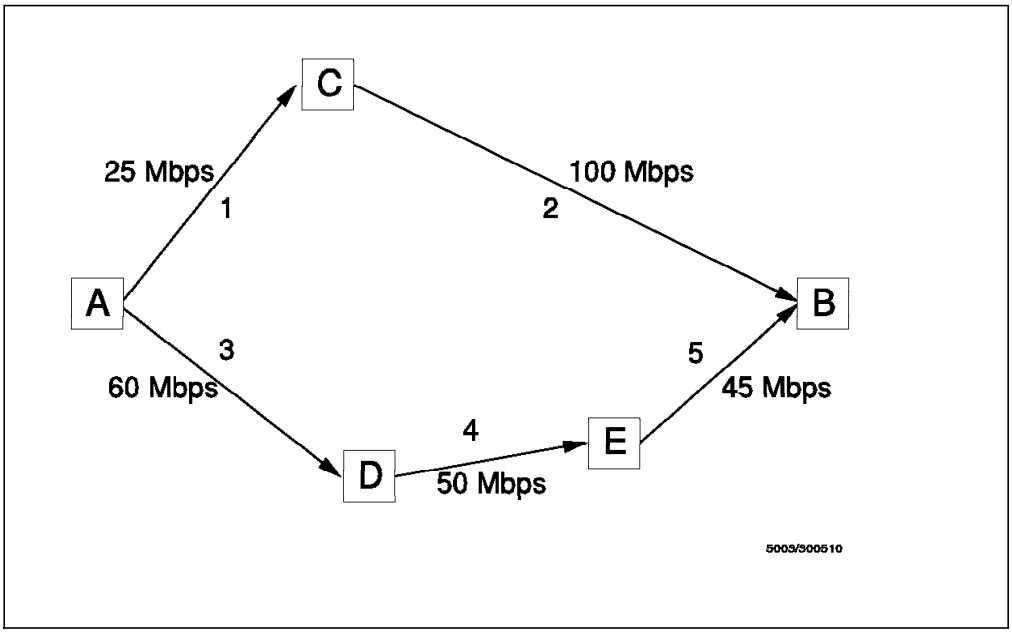



*Figure 59. Widest-Path Routing*

A disadvantage of using this algorithm is that it does not minimize the number of hops. The implementation does, however, put a limit on the number of hops by limiting the number of links that can be stored in the route vector, currently set at 15.

By using a widest path method for route computation, RB connections can easily be supported. In this case, a particular bandwidth requirement is asked for during connection setup, and the network has to guarantee that it can support that connection, otherwise the network must reject the connection request. If

any route exists to support the connection with the required bandwidth, then widest path OSPF will find it.

Widest path OSPF is also very suitable for NRB connections. In this type of connection, the caller does not specify how much bandwidth is required, it just expects the network to do the best it can. Typically the caller does not really know how much bandwidth is needed. Therefore, in order to find a route, widest path OSPF allocates an arbitrary amount of bandwidth for this type of connection. The current implementation of the IBM 8260 uses 50 Kbps.

In NRB on nonaggregate links, 80 calls are done on the widest path link then the other 80 are done on the other link. aso. (50Kbps X 80)= 4Mbps. Refer to 4.9.1.1, “Route Selection within a Cluster” on page 110.

In NRB on aggregate links, the calls are done alternatively on each link.

Widest path OSPF has very few changes from the original OSPF code. However, since the links in an 8260 ATM environment are all point-to-point, and there are no shared links, such as token-ring or Ethernet LANs, a number of the components of the OSPF that allow OSPF to be used over multiaccess broadcast LANs, such as token-ring and Ethernet, will not be needed. Furthermore, OSPF has the ability to exchange topology information outside an OSPF area (which is similar to the concept of cluster in the 8260 ATM subsystem) unlike the current 8260 ATM implementation, which does not exchange topology information outside a cluster (for information on routing between clusters please refer to 4.10, “Private Network Node Interface (P-NNI)” on page 128). The impact of these differences is that when using widest path OSPF:

- All neighbor nodes become adjacent as links are point-to-point.
- Designated nodes are required for multiaccess broadcast networks as all links are point-to-point.
- There are no area borders or AS boundary nodes as these are only required for inter-area and external topology exchange.

Because of these differences, the TRS code can now be significantly simplified and, therefore, able to operate more quickly and efficiently when compared with OSPF code.

There are a number of processes that an OSPF router network goes through. In an 8260 ATM network, these processes are identical except for the differences listed above. The following describes the OSPF processes that the TRS also implements:

1. Discovery of neighbors using the Hello protocol

   When a node and its TRS component initializes, it first tries to discover neighbors using the OSPF Hello protocol.

2. Establishing Adjacencies - Database Exchange

   The next step is that nodes establish adjacencies with their neighbor nodes and go through the database exchange process. During this process the topology databases are synchronized so that all nodes have the same map of the cluster.

3. Precomputed source routes

   When a node gets the complete network topology, it computes the route from itself to all other nodes. The widest path OSPF algorithm is used to

determine the path. The node then stores the routes to all the other nodes in a widest path tree. By doing this, the time consuming process of route computation at connection setup is avoided. This is particularly desirable in a campus environment where there is a high connection setup rate.

4. Complete route to the destination

   When a route to a destination is requested, TRS provides the node with the complete route to the destination rather than just the next hop as with OSPF. This avoids the problem of determining the route within each intermediate node.

5. Dampening function

   In OSPF, when a link changes its status, link state information is flooded throughout the network. This is done with link state updates (LSUs). This is normally done when the link goes up or down.

   In an ATM network, it would be undesirable if every time a new connection was established, LSUs were generated to recompute new routes. Therefore, a dampening threshold is set before new link state information is advertised into the cluster. In the current implementation of the 8260, if the available bandwidth on a link changes by 5% of the link capacity (4 Mbps on a 100-Mbps link) or the status of the link changes (the link goes up or down), then a new LSU will be sent to advertise link state changes.

#### 4.9.1.3 Support for Point-to-Multipoint Connections

Support for route computation in multicast connections is done a little differently. First, multicast connections are not precomputed. This is not such a problem with regard to connection setup time because it is assumed that point-to-multipoint connections will be very few compared to point-to-point connections.

The second difference is that a shortest path rather than a widest path algorithm is used. The process of building a multicast tree is as follows:

- If the node that needs to be added is the first leaf in the tree, then the route is computed as if the connection is a point-to-point connection, that is, widest path OSPF is used.
- Additional leaves are added to the tree by using a shortest path algorithm. Unlike the widest path OSPF, the cost of the route is defined as follows:

  1. All links that have been established in the multicast tree have a cost of 1.
  2. All links that do not have enough bandwidth to support the connection have an infinite cost.
  3. All other links have a cost of 1000.

  This method will ensure that the leaf is added to the nearest part of the tree, that is, the leaf will be added to the tree where there are the least number of hops. The costs that were given to these links are purely arbitrary.

#### 4.9.1.4 Internal Addressing

OSPF requires that a node and the links that attach to that node be identified with IP addresses. Widest path OSPF also needs these components to be identified, so an internal IP addressing scheme was developed to enable this.

Three different internal IP address formats are used to describe links, nodes and external NNI links to other clusters. These are shown in Figure 60 on page 114.

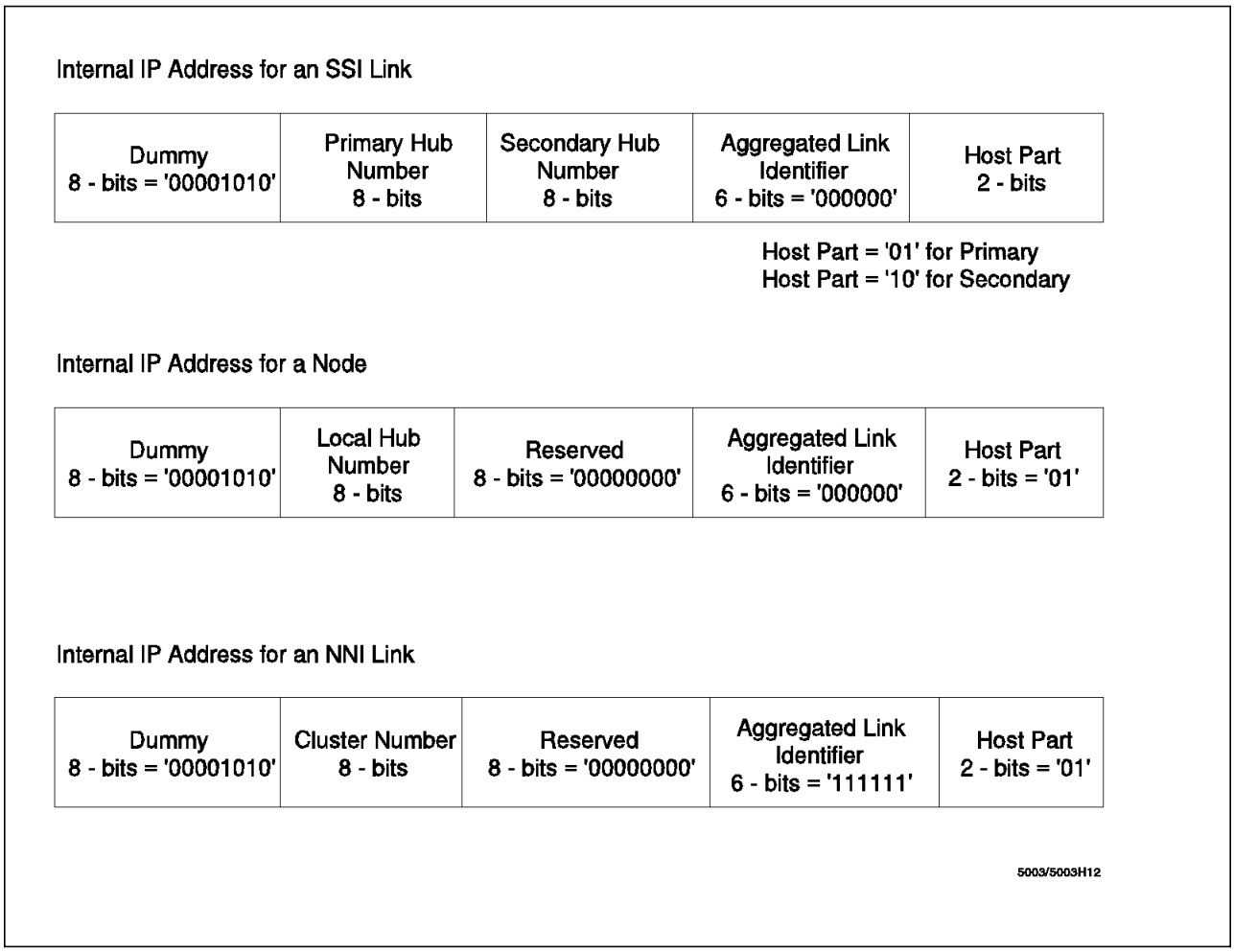



*Figure 60. Internal IP Addresses*

Note that each address can be easily identified:

- An SSI IP address format is identified when the aggregated link identifier (AgID) = ′000000′.
- An NNI IP address format is identified when AgID = ′111111′.
- A Node IP address format is identified when AgID = ′000000′ and secondary hub number = ′00000000′.

During link initialization, a Neighbor Information eXchange (NIX) process takes place at the SSI link. During this process an IP address is assigned to each end of an SSI link. NIX determines which is the primary and which is the secondary end of the link and from there the subnetwork part of the IP address can be determined. The subnetwork part of the IP address on both ends of the same link will be identical. The 2-bit field, called the host part, that indicates which end of the link is primary and which end is secondary is the only difference between the two IP addresses that identify each end of the same SSI link.

You may have noticed that the Dummy field of the internal IP address is always ″00001010″ (10 in decimal). This is because the same component of the A-CPSW code that provides IP functions used by the SNMP agent is also used by the TRS function. To allow the control point to distinguish IP packets that are for TRS from those that are for the Classical IP function, the internal IP address will

always have the first byte set to 10. This is why you cannot set the IP address of the switch to 10.xx.xx.xx.

Another consequence of using a common IP component is that TRS flows over AAL5. These are distinguished from Classical IP packets that also use AAL5 in that TRS uses VPI=0/VCI=8.

#### 4.9.1.5 Link Aggregation

One of the features of an 8260 ATM network is the ability to support link aggregation. Link aggregation is where you have two or more parallel links between two nodes but they are treated as one. The aggregated link is able to perform load balancing between the two nodes and will be resilient to link failure. Link aggregation is supported on both NNI and SSI links.

Link aggregation cannot occur between more than two nodes. If the parallel links all originate from one node and terminate at one other node, then the links can be aggregated.

The aggregated link is treated as if it is a single link. This means that route computation by TRS need not be changed. An aggregated link will still have only one IP address, but the cost of that link will be that of the link with the greatest amount of available bandwidth. When a call setup reaches the node with the aggregated link, it is only then that the decision as to which physical link the call will actually go across is made. The decision is made in the following way:

- If the call is an NRB connection, the link with the least number of NRB connections is selected.
- If the call is an RB connection, the link with the maximum amount of bandwidth available is selected.

If a failure occurs on one of the component links, then only a bandwidth update needs to be advertised. If the failing link was the least loaded, then no new topology updates need to be advertised. The benefit of this is that any new connections will be handled without any further delay because new route computations do not need to be performed.

Detection of an aggregated link is done automatically and requires no additional operator configuration:

- If the link is an SSI, link aggregation is done during the NIX process. During this process the ATM cluster number (ACN) and hub number (HN) are exchanged to determine which end of the link is primary and which is secondary. By looking at the ATM cluster number (ACN) and hub number (HN) for the remote end of a link, the 8260 node can determine if the links are parallel. If they are, they will automatically be aggregated.
- If the link is an NNI link the process is even simpler. When defining an NNI link, the operator must define a logical link which includes the ATM cluster number (ACN) that resides on the other end of a link. If a number of links are defined with the same ATM cluster number (ACN) for the remote end, then they are parallel and are automatically aggregated. For information about NNI links, please refer to 4.9.2, “ATM Subnetwork” on page 127.

**Note:** Load balancing, performed by the link aggregation function, is based on connection not packets. This means that once a connection has been established using a link, all the packets (that is, their corresponding cells) will be

sent on that link. This also means that if one of the parallel links fails, all the connections using that single link will fail and the users have to issue the call setup again to enable the A-CPSW to reestablish these connections on one of the operational members of the parallel links.

#### 4.9.1.6 Bandwidth Available on SSI Links

The amount of bandwidth available on the SSI links is dependent on the level of the A-CPSW microcode.

***Microcode Level V1.2.9:*** One each ATM link (UNI, SSI, or NNI) connected to an 8260 ATM subsystem, 85% of the bandwidth on the link is available for RB data transfer. The remaining 15% is reserved for NRB traffic.

When using SSI links in RB mode, the amount of bandwidth available per SSI link will be determined by the number of other SSI links enabled on that module.

Four SSI links can be configured on an A4-FB100 module and two SSI links on the ATMflex module. The A-CPSW will use the following procedure to determine the bandwidth reserved for the SSI links:

- A4-FB100 module
  - With one SSI link enabled on the module, 85 Mbps (85% of 100 Mbps) will be reserved for the RB SSI link on that port.
  - With two SSI links enabled on the module, 85 Mbps will be reserved for each SSI link.
  - With three SSI links enabled on the module, 60 Mbps will be reserved for each SSI link.

    **Note:** When you configure the third SSI port, the first and second SSI ports will automatically be scaled down to 60 Mbps. However, if the first and/or second SSI ports cannot be scaled down (because more than 60 Mbps on at least one of these SSI links is already allocated for the connections over these links), the enabling of the third SSI port is rejected and the console operator is notified.
  - With four SSI links enabled on the module, 45 Mbps will be reserved for each SSI link.

    **Note:** When you configure the fourth SSI port, the first, second and third SSI ports will automatically be scaled down to 45 Mbps. However, if the first, second, and/or third SSI ports cannot be scaled down (because more then 45 Mbps on at least one of these SSI links is already allocated for the connections over these SSI links), the enabling of the fourth SSI port is rejected and the console operator is notified.
- ATMflex module
  - With one SSI link enabled on the module, 131 Mbps (85% of 155 Mbps) will be reserved for the SSI link on that port.
  - With two SSI links enabled on the module, 90 Mbps will be reserved for the SSI link on that port.

    **Note:** When you configure the second SSI port, the first SSI port will automatically be scaled down to 90 Mbps. However, if the first SSI port cannot be scaled down (because more then 90 Mbps on the first SSI link

is already allocated to the connections using that SSI link), the enabling of the second SSI port is rejected and the console operator is notified.

Disabling of SSI ports will change the reserved bandwidth in the reverse order of the mechanism described earlier.

***Microcode Level V2.0.4:*** This release of A-CPSW microcode allows you to specify the bandwidth to be allocated for the SSI port. The format of the command is as follows:

```
8260A> SET PORT ENABLE SSI bandwidth
```

With the bandwidth for SSI ports, the following consideration must be taken into account:

- The total amount of bandwidth for SSI ports must not exceed 212 Mbps per ATM media module (A4-FB100 or ATMflex modules).
- The amount of bandwidth allocated to an SSI port must be greater than or equal to 51 Kbps, otherwise your command will be rejected. This is due to the fact that the 8260 will treat nonreserved bandwidth (NRB) connections as reserved bandwidth connections with a rate of 51 Kbps.
- The amount of bandwidth allocated to an SSI port must not be greater than the total bandwidth available on that port, otherwise your command will be rejected.
- If you omit specifying the bandwidth for the SSI port, the whole bandwidth of the port will be allocated by default if the port has not been configured before. However, if the port has been configured before, the previously specified bandwidth will be allocated to the SSI port. For example, as can be seen, the bandwidth allocated to the SSI port after the following sequence of commands will be 50 Mbps.

```
8260A> set port 6.1 disable
Port set
8260A> set port 6.1 enable ssi 50000
Port set
8260A> set port 6.1 disable
Port set
8260A> set port 6.1 enable ssi
Port set
8260A> show port 6.1 verbose


     Type  Mode      Status
-------------------------------------------------------------------------
 6.01:SSI enabled  UP-OKAY

SSI Bandwidth        : 50000 kbps
Connector            : MIC
Media                : fiber
Port speed           : 100000 kbps
Remote device is active
IX status            : IX KO

8260A>
```

**Note:** The bandwidth for the SSI link must always be specified in units of Kbps. The value that you specify will be rounded up by A-CPSW to the next integral number of 10 Kbps.

- To ensure that the amount of bandwidth allocated to an SSI port per an ATM module does not exceed 212 Mbps, it is required to specify bandwidth for the SSI port in the following cases:
  - More than two SSI ports are configured on an A4-FB100 module.
  - Two SSI ports are configured on an ATMflex module.

**Important**

You must ensure that you allocate an identical amount of bandwidth on each side of the SSI link to avoid route computation problems by the TRS function using the widest path algorithm.

### 4.9.1.7 Configuration of a Single ATM Cluster

If you interconnect your 8260s to work in a single cluster, use the following procedure for each 8260:

1. **Set Device ATM Address**

   In an ATM cluster, each hub uses the same 12 high-order bytes in the network prefix, including the ATM cluster number (ACN), and is assigned a unique hub number (HN), the rightmost byte in the network prefix.

2. **Connect ATM Media Modules to the Backplane**

   After setting the ATM address you must connect the ATM media modules to the network. This is necessary because the factory default setting isolates them from receiving network traffic.

3. **Enable the Ports and Interfaces (UNI, SSI)**

   After connecting an ATM media module to the network, you must enable its ports and configure the type of interface used by each port to transmit and receive ATM data.

   For connecting 8260 hubs in the same cluster, you must configure the ports that connect them as SSI interfaces. For connecting an endsystem to the network, you must use a UNI interface.

Using the steps described previously, the following example shows you how to configure two 8260s in the same cluster, as in Figure 61.

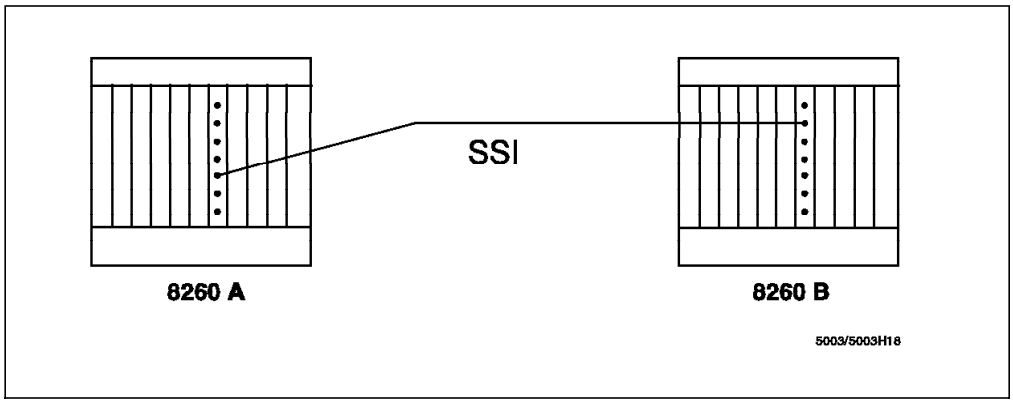



*Figure 61. Setting an SSI Link between Two 8260s*

In this example, the first 11 bytes of the ATM address are 3909851111111111111111, the cluster number (ACN) is 1, and the hub number (HN) for 8260A is 1, and for 8260B is 2.

1. **8260A:**

   a. Set device ATM address:

```
8260A> set device atm_address
39.09.85.11.11.11.11.11.11.11.11.01.01.40.00.00.82.60.A1.00
This call will reset the ATM subsystem.
Are you sure? (Y/N) Y
```

   b. Connect ATM media modules to the backplane:

```
8260A> set module 14 connected enable
```

   c. Enable the ports and interfaces:

```
8260A> set port 14.3 enable ssi
Port set
8260A>
```

2. **8260B:**

   a. Set device ATM address:

```
8260B> set device atm_address
39.09.85.11.11.11.11.11.11.11.11.01.02.40.00.00.82.60.B1.00
This call will reset the ATM subsystem.
Are you sure? (Y/N) Y
```

   b. Connect ATM media modules to the backplane:

```
8260B> set module 3 connected enable
```

   c. Enable the ports and interfaces:

```
8260B> set port 3.2 enable ssi
Port set
8260B>
```

To save your settings issue the following command in each hub:

```
8260> save module_port
```

Using the method described previously, you can configure several 8260s to work in the same cluster.

**Note**

You can have more than one SSI link between two 8260s within a cluster. This allows you to increase the bandwidth and provide redundancy between two 8260s within the same cluster.

Multiple SSI links between two 8260s will be aggregated as described in 4.9.1.5, “Link Aggregation” on page 115.

### 4.9.1.8 Design Consideration for a Cluster

In an ATM campus network design, we must consider how many switches are needed and how they must be interconnected to give the best performance, reliability, and availability. To design an ATM campus network, the following issues need to be considered:

- Number of ATM devices that will be connected to the network

  The number of ATM devices that attach to your network decide the number of ATM ports and the number of switches required. Make sure during the network design that you provide sufficient ports to accommodate all your ATM devices. You must also realize that when the number of switches within your network grows, the number of inter-switch links (and ports used for these) grows as well. In addition, more bandwidth is required to interconnect your switches. You must reserve capacity for future growth.

- Bandwidth and virtual circuit requirements of ATM-attached devices (for example, servers and bridges)

  ATM switches and ports have finite capacity in terms of bandwidth and VCCs. Make sure during your network design that the network has sufficient capacity to handle all your connections and is capable of providing the bandwidth required for them. Reserve capacity for future growth.

- Requirements to balance or alleviate heavy traffic in hub-to-hub connections and hub-to-server connections

  The bandwidth available on the hub-to-hub and the hub-to-server links must be sufficient to satisfy the bandwidth requirements for all connections established over these links, especially for nonreserved bandwidth connections, as it is unclear what the actual bandwidth demand will be. It is advised to calculate your bandwidth requirements conservatively.

  **Note:** Every network requires proper management; skilled personnel and adequate tools are required to design and maintain the network properly, solve problems quickly, and, if possible, prevent problems from impacting the networking service. In networks with inadequate resources, however, the need for problem determination, network tuning, and capacity planning increases rapidly. In a campus network, where the cost of bandwidth is much lower than in a WAN environment, the investment for extra switching capacity easily offsets the extra costs introduced by additional network management.

- Requirements for performance, reliability, and availability of the ATM campus network

  Connectivity to critical devices (for example, servers, routers, and bridges) requires extra care. You must focus, not only on the network attachments, but you must also consider all networking components (links and switches)

that provide access to these devices. You must ensure that single points of failure within your network are avoided.

Adequate (optical or shielded copper) cabling, with proper grounding, increases the reliability of your links. Make sure that backup is available for the networking components. For maximum availability of your hubs, make sure that they are equipped with multiple power-supplies, powered from different sources and protected by uninterruptible power supplies (UPS).

Some of these requirements can be addressed by using hub clustering and link aggregation.

In an 8260 ATM campus network, the ATM hubs are identified by the hub number (HN). The 8260 ATM hubs can be grouped together in clusters (groups of hubs linked by SSI interfaces). Clusters can be linked by NNI interfaces, creating an ATM subnetwork. SSI supports topology and routing services, allowing optimal route selection and minimal configuration requirement.

All the ATM stations attached to the same ATM hub have the same first 13 bytes in their ATM address. In the same way, all the devices connected to the same cluster have the same first 12 bytes in their addresses, and those connected to the same ATM subnetwork have the same first 11 bytes.

ATM switches within a cluster may be configured using the following topologies:

- Fully meshed
- Star configuration
- Ring configuration

A fully meshed topology, as shown in Figure 62 on page 122, optimizes performance and availability as the number of intermediate hops between two communicating endstations is minimized. An important drawback of this topology is the high number of SSI links (and ports supporting these links) required for the inter-hub connections. Another drawback is that the large number of SSI links increases the size of the network topology database, which will result in longer restart times due to a higher number of LSUs advertised and routes precomputed. During steady-state, however, no increased number of LSUs or route computations are required.

Adding an ATM switching node to a fully meshed cluster with *n* ATM switches requires 2x*n* extra ports and *n* additional links. For larger clusters (cluster with $\geq$ 4 ATM switches), a fully meshed topology will, in most cases, not be feasible.

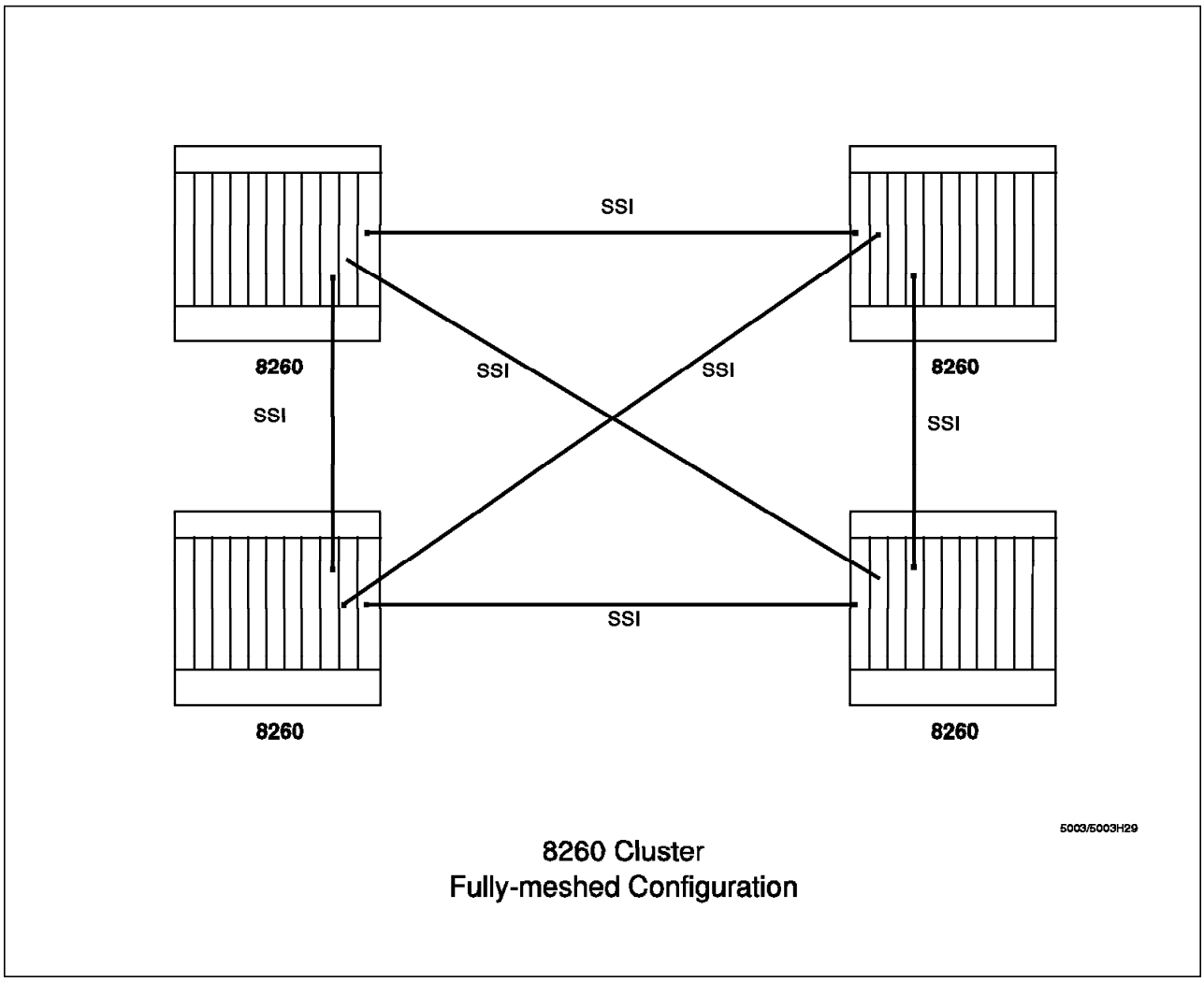



*Figure 62. Fully Meshed Topology within an ATM Cluster*

An alternative to the fully meshed network is the ring topology, as shown in Figure 63 on page 123. It requires a lower number of SSI links and ports for inter-hub connectivity when compared to fully meshed topology. Adding a node to a cluster with ring topology consisting of *n* ATM switches, requires two extra ports and two additional links. Also, the addition of the new node could be disruptive to some of the existing connections as the ring must be broken temporarily to allow for the addition of the new node. The ring topology provides an alternate route, enabling service continuation during a link or a node failure.

The ring topology is less suited for large networks due to the transit delays caused by hub latency if a connection between two ATM stations must traverse several ATM switches. It is advisable to make sure that your VCCs, in general, do not traverse more than 5 hubs.

**Note:** The 8260 latency is around 60 microseconds per hub, while the transmission time of a 1000-byte frame on a 100-Mbps link is around 80 microseconds (due to the small ATM cell size).

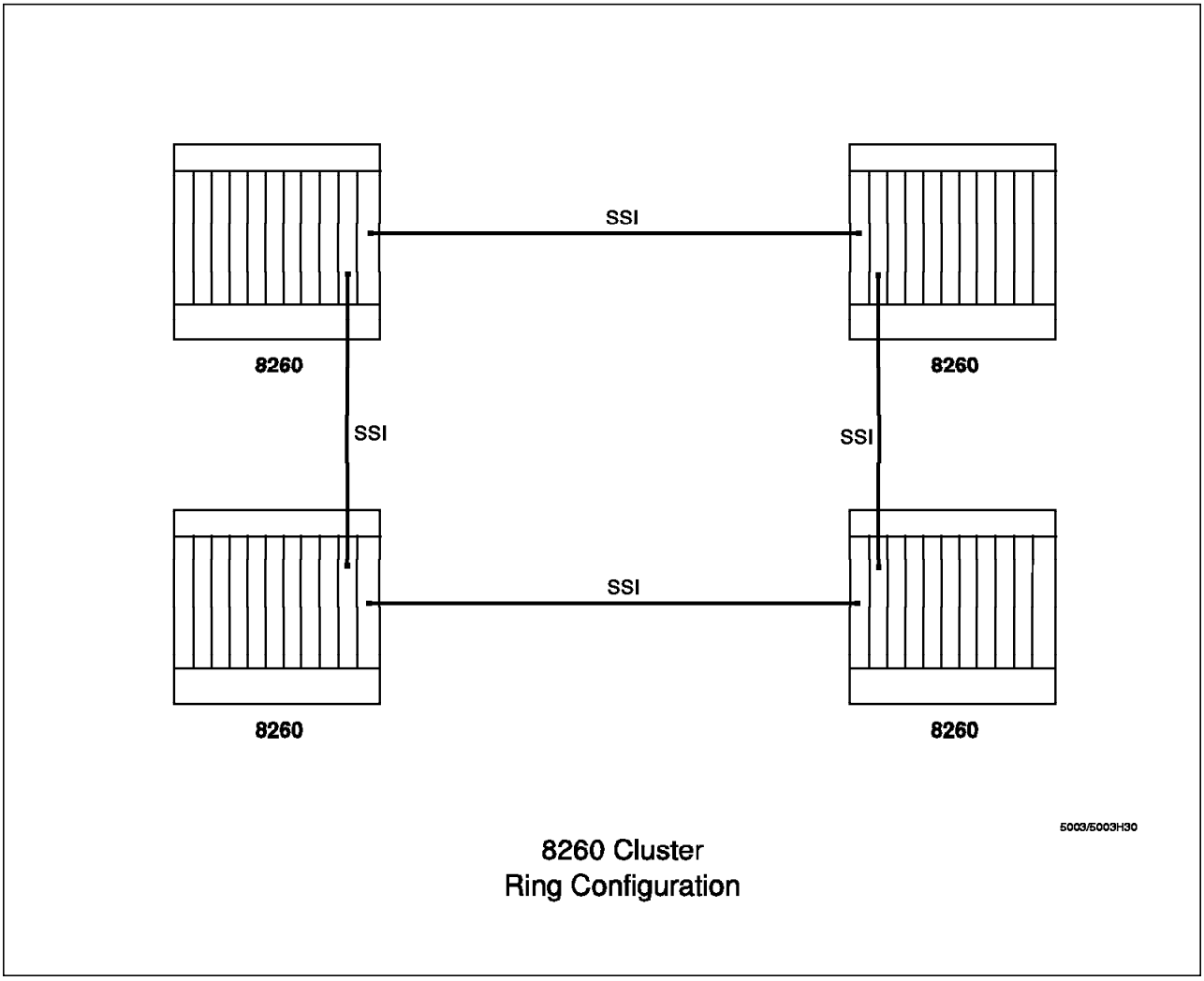



*Figure 63. Ring Topology within an ATM Cluster*

A star topology, as shown in Figure 64 on page 124, is most suited for networks in which all of the servers, routers, bridges, etc., are attached to a single, or a limited number of ATM switch(es), and traffic is predominately to/from the servers. The main advantage of a star topology is that it is less expensive than the fully meshed or ring topology as it requires the lowest number of SSI links (and ports connecting them) and allows you to use smaller switches (for example, the IBM 8285) with a limited number of high-speed ports for the peripheral ATM switches. Furthermore, a star topology is easily expandable. Adding a switch to a star topology with *n* ATM switches requires only 2 extra ports and 1 additional link. Due to its simplicity, this topology is easy to define and maintain. Drawbacks are that all communication between endstations attached to two different *peripheral* ATM switches will flow through the *central* ATM switch. This may lead to an excessive load on the central ATM switch. Also, the status of the central ATM switch is crucial to the operation of the network as a whole.

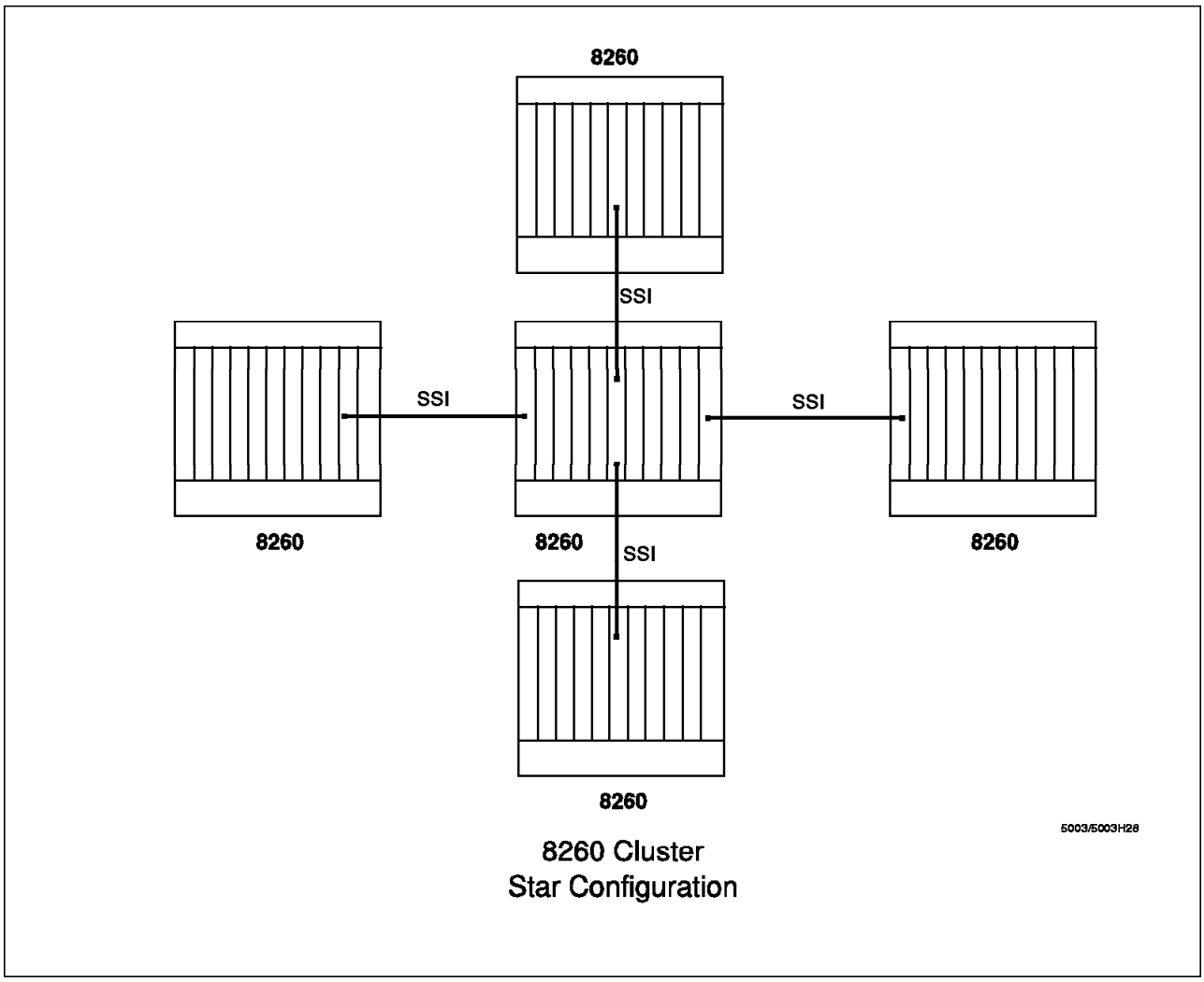



*Figure 64. Star Topology within an ATM Cluster*

The choice of topology for your network depends on your traffic patterns, availability and performance requirements, and your networking budget. You may design a network which may be a combination of all the topologies discussed. An example of such a topology is provided in Figure 65 on page 125.

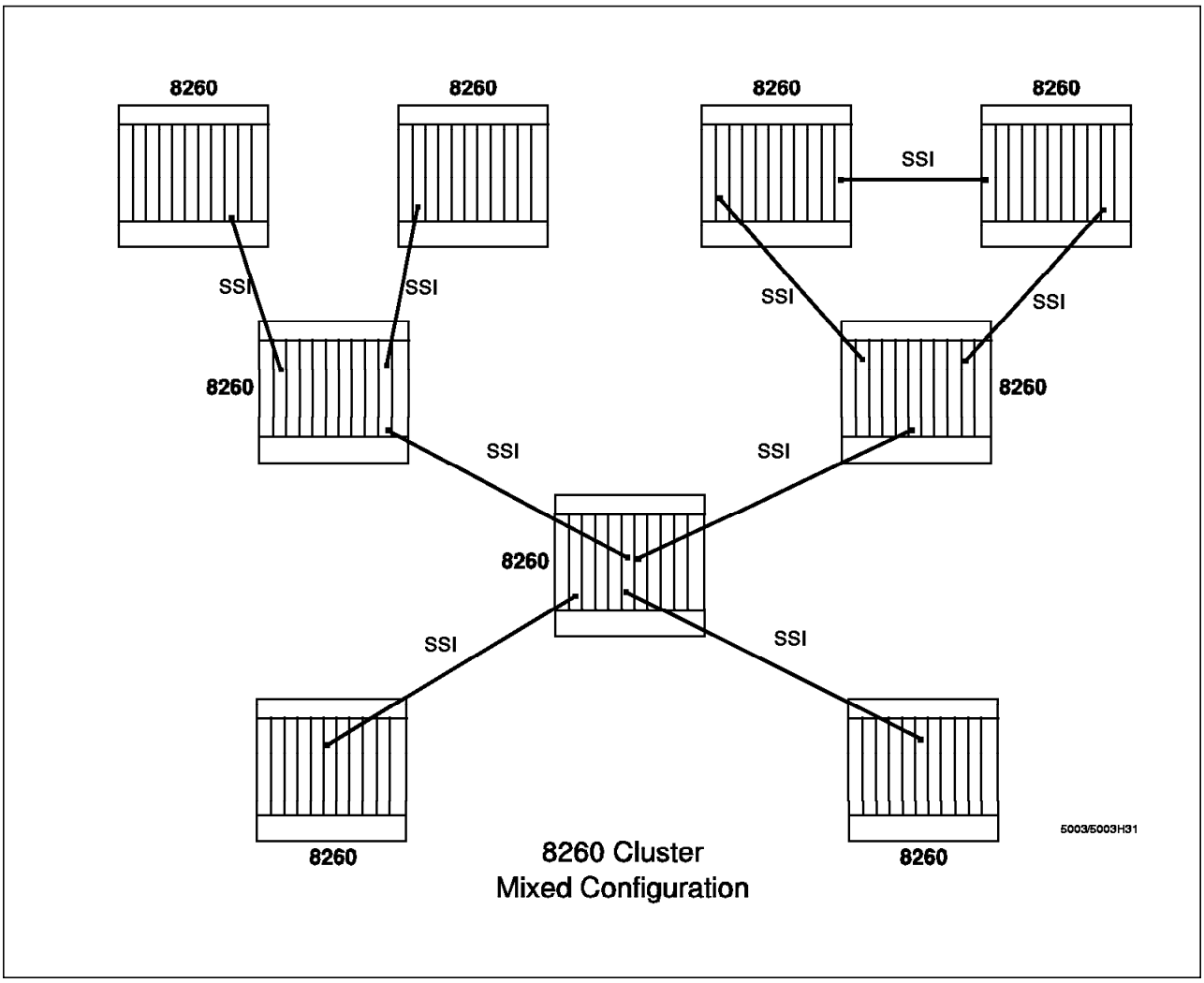



*Figure 65. Mixed Topology within an ATM Cluster*

Before you design your ATM campus network, make sure you have a proper understanding of the topology and routing services used on the ATM switches. IBM ATM campus switches make routing decisions on the basis of the widest path available as described in 4.9, “Topology and Route Selection (TRS) Services” on page 108. Therefore, you must ensure that the design of your network takes into account the effects of the widest path algorithm.

Figure 66 on page 126 shows an example of the effects of the widest path algorithm. Despite a direct link between A and D, the widest path from A to D is via A-B-C-D. The smallest, or bottleneck, link in this path is B-C. This link has more bandwidth than the alternate path from A to D. The alternate path will be used when the bandwidth available on any of the intermediate links drops below 25 Mbps.

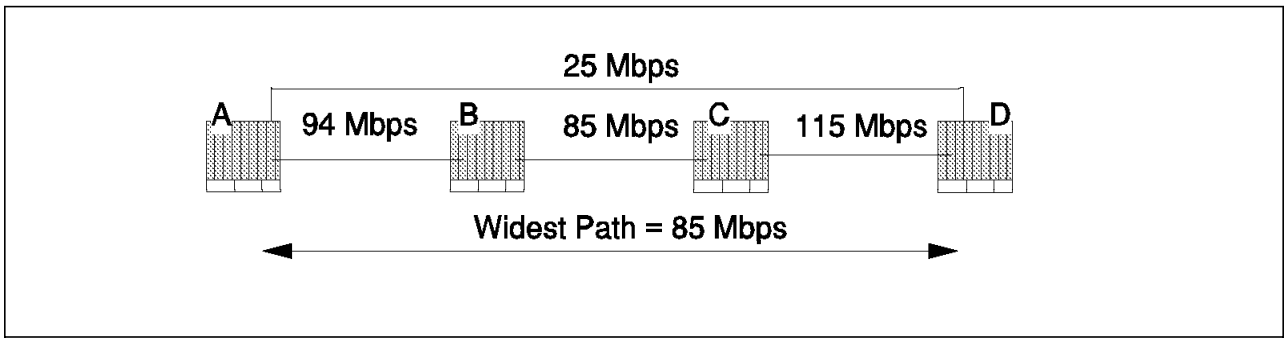



*Figure 66. Widest Path Route Computation*

The widest path algorithm has an important implication; when the reason for multiple paths in your network is load balancing, make sure the paths have an equivalent width. For example, the direct link between A and D in Figure 66 is more likely to be used when upgraded to 100 Mbps.

The easiest way to realize load balancing within your network is to make sure that all inter-hub connections have the same available bandwidth. By default, in RB 85% of the link capacity is reserved, however, an operator may assign less bandwidth to the SSI connections when required (for example, when introducing 155-Mbps links in a network that already has multiple 100-Mbps trunks). Remember that the available bandwidth is used for route calculation, and nonreserved bandwidth connections may use bandwidth above that value.

The following needs to be considered when connecting IBM ATM campus hubs in a cluster structure:

- Hub numbers within a cluster must be unique, while hub numbers in different clusters may be the same.
- Each 8260 takes 11 milliseconds to handle a call SET UP and its associated CONNECT (average time measured for 100 SET UP/CONNECT).

  **Recommendation**

  A star hub topology configuration is recommended, with the LAN emulation servers, ARP servers, and file servers at the center of the network to achieve shorter connection times (with only two hubs involved, connection establishment is in about 22 milliseconds).

- The 8260 transit delay per port is 33 microseconds (40 under heavy load)

  **Recommendation**

  To avoid excessive connection and transfer delays, limit the number of intermediate hubs to five (four SSI interfaces). Using five hubs the network-induced connection time is 5x11=55 milliseconds, and the maximum network transit delay around (40 microseconds x 5)= 200 microseconds.

  **Note:** Due to the small ATM cell size, the network transmit time is decided by the hub latency only, and virtually independent of the transmission delays on intermediate links (assuming the access links are not faster than your network trunks).

- The maximum network size is 255 clusters with 255 hubs each.

**Recommendation**

The maximum number of hubs recommended in a cluster is 25, because with more, the WET_OSPF update cycling will be too long.

- You can have multiple parallel SSI links between two hubs using link aggregation. See Figure 65 on page 125.

  **Recommendation**

  Connect parallel SSI links to different modules on your switch to achieve redundancy for link failure as well as for module failure.

- You can define multiple SSI links on each high-speed blade, but in any case the maximum bandwidth you can reserve cannot exceed 212 Mbps. For more detailed information on the SSI bandwidth, please refer to 4.9.1.6, “Bandwidth Available on SSI Links” on page 116.

  **Recommendation**

  If SSI link aggregation is being used with more than two links, distribute the links between two or more modules.

- A single 8260 can handle approximately 100 calls per second.

In Figure 62 on page 122 you can see a fully meshed four-hub cluster. Full backup capabilities are provided for link and module failures. If any of the links between two hubs fails, an alternate path is always available through a third hub.

If a high-speed module fails, we have another module with an available path to the destination hub. For example, if we have communication between a user in hub A and a server in hub B and the primary connection fails, an alternate route exists.

### 4.9.2 ATM Subnetwork

All the ATM switching subsystems within the network that may be the same 11 bytes for their ATM address are said to be in the same ATM subnetwork. An ATM subnetwork consists of one or more clusters. Byte 12 of the ATM address is used to identify the cluster number (CN) within an ATM subnetwork. Clusters are connected to each other using NNI links. The NNI links use ATM Forum′s Interim Inter-Switch Protocol (IISP).

Figure 67 on page 128 shows an example of an ATM network consisting of three clusters of the 8260 ATM switching subsystems.

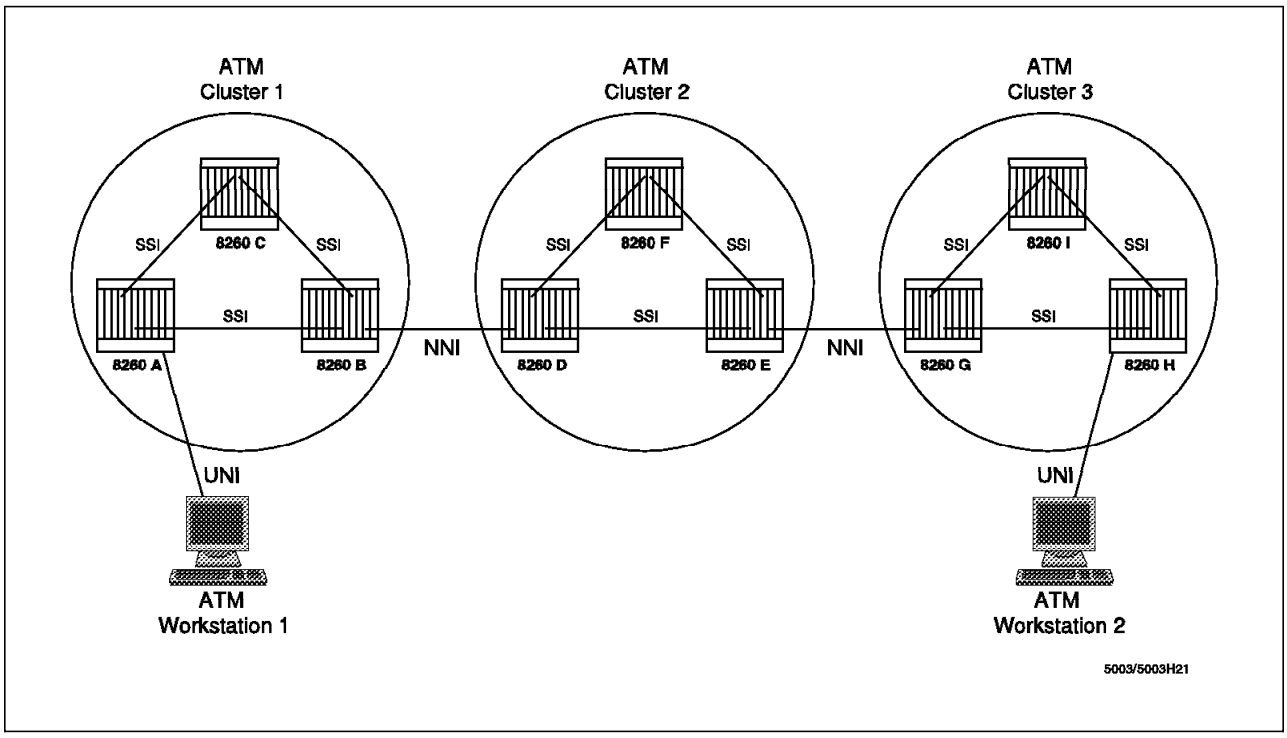



*Figure 67. 8260 ATM Subnetwork*

The following section provides you with a description of IISP and its implementation within IBM 8260 ATM subsystem.

## 4.10 Private Network Node Interface (P-NNI)

ATM is a connection-oriented network that consists of a set of ATM switches interconnected by point-to-point ATM links. The interface between ATM switches is referred to as network node interface (NNI). The NNI protocols are used within ATM networks to route ATM signalling requests between ATM switches. The ATM Forum is in the process of defining the Private NNI (P-NNI) protocol for use within the private ATM networks. The P-NNI protocol consists of two components:

1. P-NNI signalling

   This is used to relay ATM connection requests within the networks between the source and destination. To do so, the UNI signalling request is mapped to the NNI signalling at the source switch. The NNI signalling is then remapped back into UNI signalling at the destination switch.

   The P-NNI signalling is an extension of the UNI signalling and incorporates additional information elements for NNI-related parameters. P-NNI signalling is carried across NNI links on virtual channel 5. The VPI value depends on whether the NNI link is over a physical or logical connection. Figure 68 on page 129 shows a configuration where two ATM switching systems (8260C and 8260D) are connected to each other via a virtual path (VP) over a public ATM network. This connection is an example of the logical NNI connection. Whereas, the connection between 8260C and 8260G is is an example of a physical NNI connection.

When the NNI connection is physical, the VP that is used to carry the NNI signalling is dependent on the VPI range supported by the ATM switching system. For example, this range for 8260 is 0-15.

When the NNI connection is over a logical path, the VP that is used to carry the NNI signalling is the same as the VPI of the VP that provides the connection. In this case, you need to contact the public ATM switching supplier to find out the VPI of the VP set up for you. However, note that this VPI number cannot be outside the range supported by your ATM switching system. In the case of the 8260, this range is 0-15.

**Note:** In Figure 68, the links connecting the 8260C and 8260D to the public ATM switching system are configured as UNI interfaces on the public ATM switches but are configured as NNI links on the 8260C and 8260D.

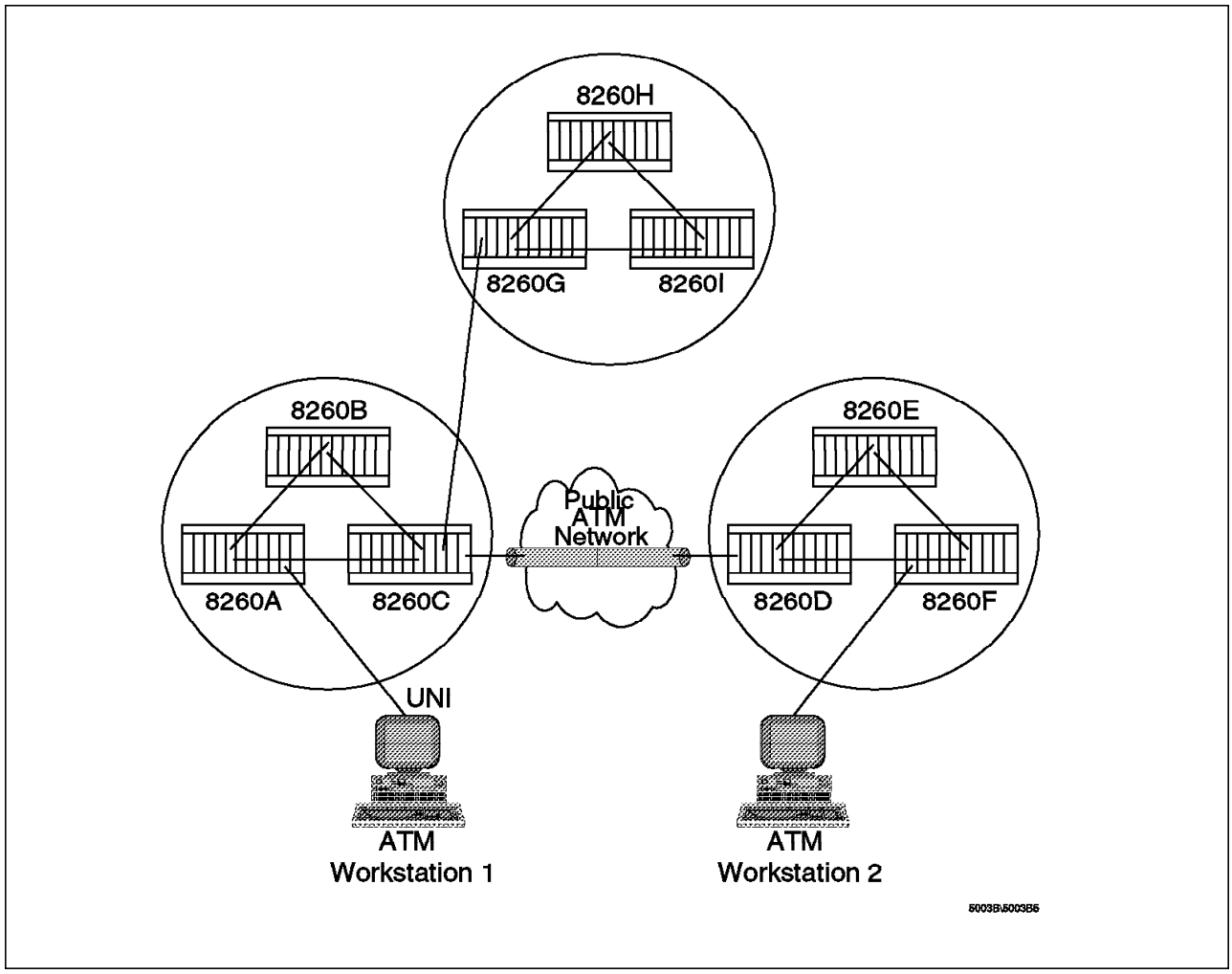



*Figure 68. NNI Connections over Physical and Logical Links*

2. Virtual circuit routing protocol

   This component of P-NNI is used to route the signalling request through the ATM network. The route along which the signalling request is transmitted is also used to set up the ATM connection, which, in turn, is used to carry the user traffic.

Currently, the ATM Forum is working on the specification of the P-NNI (also referred to as P-NNI phase 1). In the interim period, to facilitate the construction

of multivendor ATM networks, ATM Forum has developed a very simple UNI-based signalling protocol to be used between ATM switching subsystems. This protocol, originally designated as P-NNI phase 0, was later renamed the Interim Inter-Switch Signalling Protocol (IISP) to avoid the confusion with P-NNI phase 1.

The following section provides a description of the IISP specifications.

### 4.10.1 Interim Inter-Switch Signalling Protocol (IISP)

IISP is a UNI 3.1 (and optionally UNI 3.0) based signalling protocol used between ATM switching subsystems to provide some level of multivendor ATM switching subsystem interoperability as shown in Figure 69.

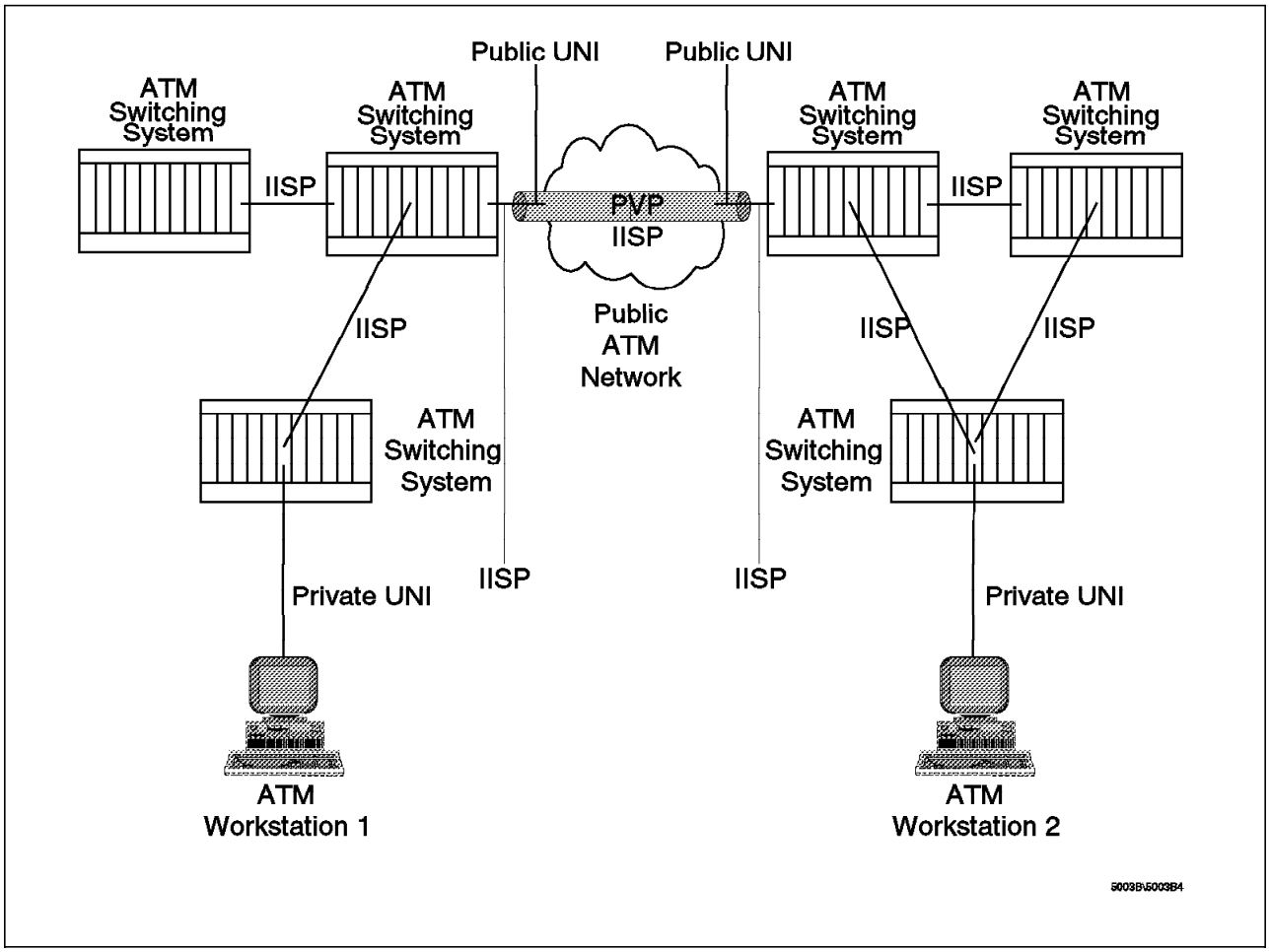



*Figure 69. Multivendor Connectivity Using IISP*

The characteristics of the IISP links are described in the following sections.

#### 4.10.1.1 ATM Physical Layer

All the physical interfaces specified in the ATM Forum UNI 3.1 may be used at the IISP interface.

#### 4.10.1.2 Cell Format

The cell format across IISP link is the same as UNI cell format.

#### 4.10.1.3 ILMI Support

The ILMI procedures shall not be used across the IISP links. In particular, ILMI address registration procedures will not be employed across IISP links.

#### 4.10.1.4 Traffic Management

Usage of the policing functions, such as the usage parameter control (UPC) as specified in UNI 3.1 specifications, is optional.

#### 4.10.1.5 VPI/VCI Range

ATM switching subsystems using IISP must be able to accept and establish calls within the following VPI/VCI ranges:

- VPI = 0
- VCI = 32-255

Optionally, other VPI/VCI ranges may be generated and accepted across IISP links.

**Note:** The implementation of IISP in the IBM 8260 accepts and establishes calls within the following VPI/VCI ranges:

- VPI = 0 - 15
- VCI = 32-1023

#### 4.10.1.6 Signalling

Since IISP signalling is based on the UNI 3.1 signalling (with optional support for UNI 3.0 signalling), the ATM switching subsystems assume the role of the network or user side across a particular IISP link. The roles are assigned manually as part of the configuration of the ATM switching subsystem.

There are no restrictions regarding the role that an ATM switching subsystem can play at different links. This means that an ATM switching subsystem can play one role at one link and another role at another link. Moreover, an intermediate ATM switching subsystem can play one role at the incoming side of a call, and the same or the other role at the outgoing side of the same call. Figure 71 on page 133 shows an example of a configuration where ATM Switching Subsystem B assumes different roles on different links.

Note that the network-side is responsible for the assignment of the VPI and VCI values. This will, therefore, eliminate the possibility of call collisions on an IISP link due to the assignment of identical VPI/VCI by the two ATM switching subsystems for the calls occurring simultaneously.

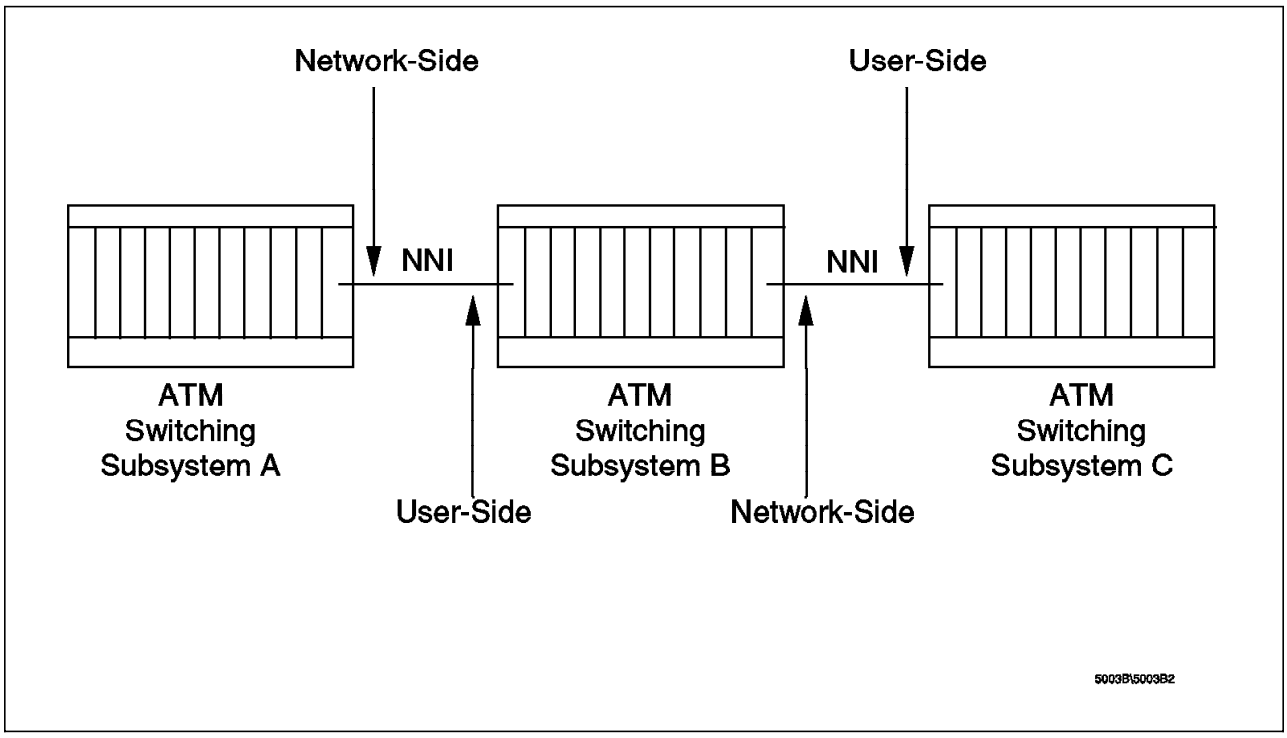



*Figure 70. Network and User-Side Roles Using IISP*

#### 4.10.1.7 Support for PVC

The use of PVCs across the IISP links are supported. However, the current implementation of the 8260 does not support PVC connections over NNI (IISP) links.

#### 4.10.1.8 Support for Point-to-Multipoint Connections

Point-to-multipoint connections across IISP links are supported.

#### 4.10.1.9 Routing

When two ATM switching subsystems are connected to each other over an IISP link, there is no routing information exchanged between them. Therefore, the signalling requests are routed between ATM switching subsystems using preconfigured static routing tables. These static routing tables, which are configured by the user manually for each ATM switching subsystem, specify the ATM address prefixes that are reachable through each port of the ATM switching subsystem.

When a signalling request is received by the ATM switching subsystem, it will check the destination address against the preconfigured static routing table and will forward the signalling request across the port with the longest prefix match. This routing is done on a hop-by-hop basis in each ATM switching subsystem along the route, each using their own static routing table.

If there is no match found in the static routing table for the destination address specified in the call setup, the IISP routing function will indicate this fact to the signalling function in the ATM switching subsystem, which, in turn, should generate a RELEASE COMPLETE message with the cause code indicating no route to destination.

**Note:** In case the call admission control is implemented in the ATM switching subsystem, a call can still be rejected even if a match is found in the static routing table.

The static routing table entries consist of the following three fields:

- ATM address

  This is the 20-octet string defined by the UNI specification.

- Address length

  This field specifies the number of significant bits that need to be taken into account while comparing the destination ATM address in the call setup with the static routing table entry.

- Interface index

  This field identifies the port on the ATM switching subsystem that should be used for routing the call setup request.

To understand the concept of static routing tables used in IISP, we will use the configuration shown in Figure 71 as an example. Note that the following information is intended to provide the concept of IISP. For details about how to configure the 8260 ATM subsystem to use IISP, please refer to 4.10.2, “IISP Implementation in the 8260” on page 138.

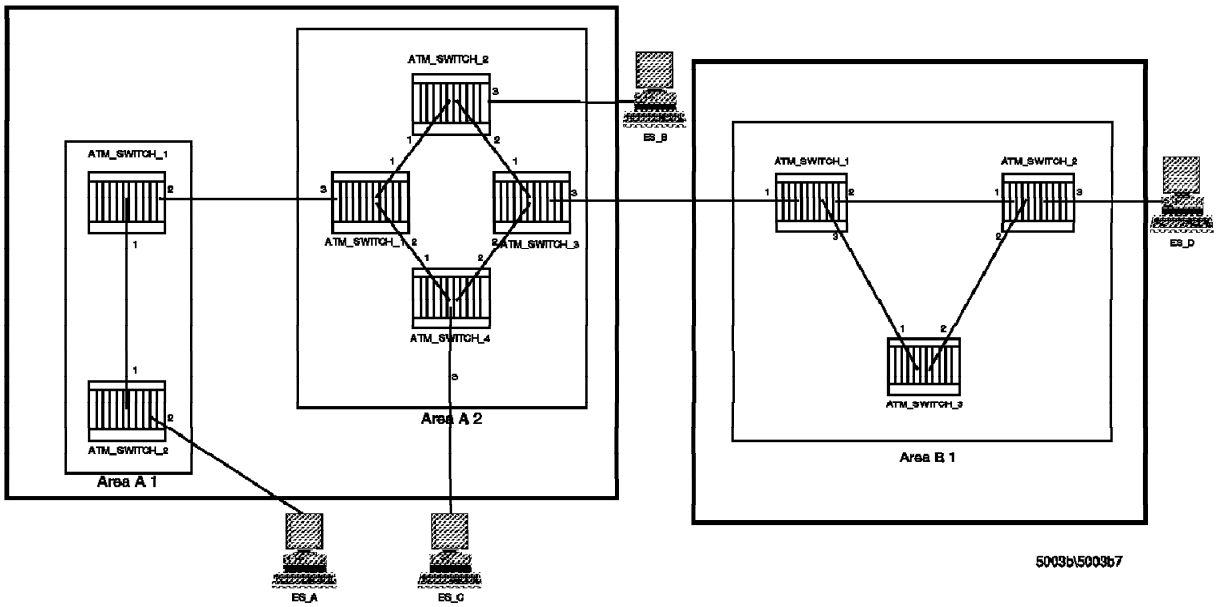



*Figure 71. Static Routes Using IISP*

The configuration used in Figure 71 consists of two areas:

- Area A - This area is further divided into:
  - Area A.1 which consists of two ATM switching subsystems with the following ATM address prefixes assigned to them:
    - ATM_SWITCH_1: 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.1.1
    - ATM_SWITCH_2: 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.1.2

**Note:** An endsystem (ES_A) is attached to ATM_SWITCH_2 and has the following ATM address:

- 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.01.02.40.00.00.00.00.0A.00

– Area A.2 which consists of four ATM switching subsystems with the following ATM address prefixes assigned to them:

- ATM_SWITCH_1: 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.2.1
- ATM_SWITCH_2: 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.2.2

  **Note:** An endsystem (ES_B) is attached ATM_SWITCH_3 and has the following ATM address:

  - 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.02.40.00.00.00.00.0B.00

- ATM_SWITCH_3: 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.03.40.00.00.00.00.0B.00

  **Note:** An endsystem (ES_C) is attached to ATM_SWITCH_3 and has the following ATM address:

  - 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.03.40.00.00.00.00.0C.00

- ATM_SWITCH_4: 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.2.4

• Area B - This area consists of:

– Area B.1 which consists of two ATM switching subsystems with the following ATM address prefixes assigned to them:

- ATM_SWITCH_1: 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.1.1
- ATM_SWITCH_2: 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.1.2

  **Note:** An endsystem (ES_D) is attached to ATM_SWITCH_2 and has the following ATM address:

  - 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.01.02.40.00.00.00.00.0D.00

- ATM_SWITCH_3: 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.1.3

ATM_SWITCH_1 in Area A.1 and ATM_SWITCH_1 in Area A.2 are used to connect these two areas together. Also, ATM_SWITCH_3 in Area A.2 and ATM_SWITCH_1 in Area B.1 are used to connect these two areas together. Figure 71 on page 133 shows the interface numbers used to connect the various ATM switching subsystems together. The static route tables that are configured in the ATM switching subsystems are shown in Table 25 through Table 33 on page 137.

*Table 25. ATM_Switch_1 in Area A.1*

| Destination ATM Address | Length | Interface |
|---|---|---|
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.01.02 **1** | 104 | 1 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02 **2** | 96 | 2 |
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB **3** | 88 | 2 |

**1** Static route to ATM_Switch_2 in Area A.1

**2** Static route to Area A.2

**3** Static route to Area B.1

*Table 26. ATM_Switch_2 in Area A.1*

| Destination ATM Address | Length | Interface |
|---|---|---|
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.01.02.40.00.00.00.00.0A.00 1 | 152 | 2 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.01.01 2 | 104 | 1 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02 3 | 96 | 2 |
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB 4 | 88 | 2 |

1 Static route to ES_A

2 Static route to ATM_Switch_1 in Area A.1

3 Static route to Area A.2

4 Static route to Area B.1

*Table 27. ATM_Switch_1 in Area A.2*

| Destination ATM Address | Length | Interface |
|---|---|---|
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.02 1 | 104 | 1 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.03 2 | 104 | 1 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.04 3 | 104 | 2 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.01 4 | 96 | 3 |
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB 5 | 88 | 1 |

1 Static route to ATM_Switch_2 in Area A.2

2 Static route to ATM_Switch_3 in Area A.2

3 Static route to ATM_Switch_4 in Area A.2

4 Static route to Area A.1

5 Static route to Area B.1

*Table 28. ATM_Switch_2 in Area A.2*

| Destination ATM Address | Length | Interface |
|---|---|---|
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.02.40.00.00.00.00.0B.00 1 | 152 | 3 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.01 2 | 104 | 1 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.03 3 | 104 | 2 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.04 4 | 104 | 2 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.01 5 | 96 | 1 |
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB 6 | 88 | 2 |

1 Static route to Area A.1

2 Static route to ATM_Switch_1 in Area A.2

3 Static route to ATM_Switch_3 in Area A.2

4 Static route to ATM_Switch_4 in Area A.2

5 Static route to ES_B

6 Static route to Area B.1

*Table 29. ATM_Switch_3 in Area A.2*

| Destination ATM Address | Length | Interface |
|---|---|---|
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.01 1 | 104 | 1 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.02 2 | 104 | 1 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.04 3 | 104 | 2 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.01 4 | 96 | 1 |
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB 5 | 88 | 3 |

1 Static route to Area B.1

2 Static route to ATM_Switch_2 in Area A.2

3 Static route to ATM_Switch_4 in Area A.2

4 Static route to Area A.1

5 Static route to ATM_Switch_1 in Area A.2

*Table 30. ATM_Switch_4 in Area A.2*

| Destination ATM Address | Length | Interface |
|---|---|---|
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.02.40.00.00.00.00.0C.00 1 | 152 | 3 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.01 2 | 104 | 1 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.02 3 | 104 | 4 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.03 4 | 104 | 4 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.01 5 | 96 | 1 |
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB 6 | 88 | 4 |

1 Static route to ES_C

2 Static route to ATM_Switch_1 in Area A.2

3 Static route to ATM_Switch_2 in Area A.2

4 Static route to ATM_Switch_3 in Area A.2

5 Static route to Area A.1

6 Static route to Area B.1

*Table 31. ATM_Switch_1 in Area B.1*

| Destination ATM Address | Length | Interface |
|---|---|---|
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.01.02 1 | 104 | 2 |
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.01.03 2 | 104 | 3 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA 3 | 88 | 1 |

1 Static route to ATM_Switch_2 in Area B.1

**2** Static route to ATM_Switch_3 in Area B.1

**3** Static route to Area A

*Table 32. ATM_Switch_2 in Area B.2*

| Destination ATM Address | Length | Interface |
|---|---|---|
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.02.02.40.00.00.00.00.0D.00 **1** | 152 | 3 |
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.01.01 **2** | 104 | 1 |
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.01.03 **3** | 104 | 2 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA **4** | 88 | 1 |

**1** Static route to ES_D

**2** Static route to ATM_Switch_1 in Area B.1

**3** Static route to ATM_Switch_3 in Area B.1

**4** Static route to Area A

*Table 33. ATM_Switch_2 in Area B.2*

| Destination ATM Address | Length | Interface |
|---|---|---|
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.01.01 **1** | 104 | 1 |
| 39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.01.03 **2** | 104 | 2 |
| 39.AA.AA.AA.AA.AA.AA.00.00.AA.AA **3** | 88 | 1 |

**1** Static route to ATM_Switch_1 in Area B.1

**2** Static route to ATM_Switch_3 in Area B.1

**3** Static route to Area A

Using these tables, the call initiated from ES_A to ES_B will be routed on the following path:

```
39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.01.02.40.00.00.00.00.0A.00 To:
39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.01.02 To:
39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.01.01 To:
39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.01 To:
39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.02 To:
39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.02.40.00.00.00.00.0B.00
```

whereas a call initiated from ES_C to ES_D will be routed on the following path:

```
39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.04.40.00.00.00.00.0C.00 To:
39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.04 To:
39.AA.AA.AA.AA.AA.AA.00.00.AA.AA.02.03 To:
39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.01.01 To:
39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.01.02 To:
39.BB.BB.BB.BB.BB.BB.00.00.BB.BB.02.02.40.00.00.00.00.0D.00
```

#### 4.10.1.10 Support for Parallel Links

If the destination address is reachable through parallel links, the ATM switching subsystem must make a decision to determine which link should be used. The algorithm for making such a distinction is not specified in the IISP specification and is left to the ATM switching subsystem manufacturer to employ a scheme that takes into account variables that are pertinent to the functions employed in the ATM switch.

If there are parallel NNI links defined between two 8260s, the IBM 8260 will choose the first link that is fully defined and operational. If that link fails after a period of time, the IBM 8260 will switch to the alternate link. In our testing within the ITSO lab, when two A-CPSW modules were communicating with each other over parallel NNI links, when the primary NNI link failed, we noticed that the switch over to backup NNI link and the establishment of a new SVC between the two A-CPSW modules took about 20 seconds.

#### 4.10.1.11 Support for Alternate Routes

IISP specification allows for the configuration of alternate routes in the static routing table. The alternate routes will be selected when the interface selected in the first match is in inactive state. The management interface used in the ATM switching subsystem must support the configuration of alternate routes such that the routes leading to loops are avoided.

#### 4.10.1.12 Support for IISP Link Failure

In the event of the failure of the IISP link, all the SVCs using that link will be cleared.

#### 4.10.1.13 Compatibility between IISP and P-NNI Phase 1

Because of the fact that IISP uses UNI signalling whereas P-NNI phase 1 uses NNI signalling, IISP is not interoperable with P-NNI phase 1 implementation.

The following areas describe the implementation of the IISP in the 8260 and how the 8260 must be configured to take advantage of this function.

### 4.10.2 IISP Implementation in the 8260

8260 implementation allows you to use IISP links for the following purposes:

- Interconnect clusters within the same ATM subnetwork
- Interconnect ATM subnetworks within an ATM campus network

The following sections describe how to configure IISP links for various types of configuration.

**Note:** All the ATM switching subsystems within an ATM subnetwork share the same 11 high order bytes in the network prefix of their ATM address.

### 4.10.3 Configuring NNI Connection between Adjacent Clusters within an ATM Subnetwork

Two clusters are defined in this book to be adjacent when there is a physical connection between an ATM switching subsystem within one of those clusters to an ATM switching subsystem within the other cluster, as shown in Figure 72 on page 139. In this configuration, we refer to the ATM switching subsystems that are used to connect the two clusters as the boundary ATM switching subsystems. Figure 72 on page 139 shows an example of two adjacent clusters

within the same ATM subnetwork. In this example, 8260B and 8260D are the boundary ATM switching subsystems.

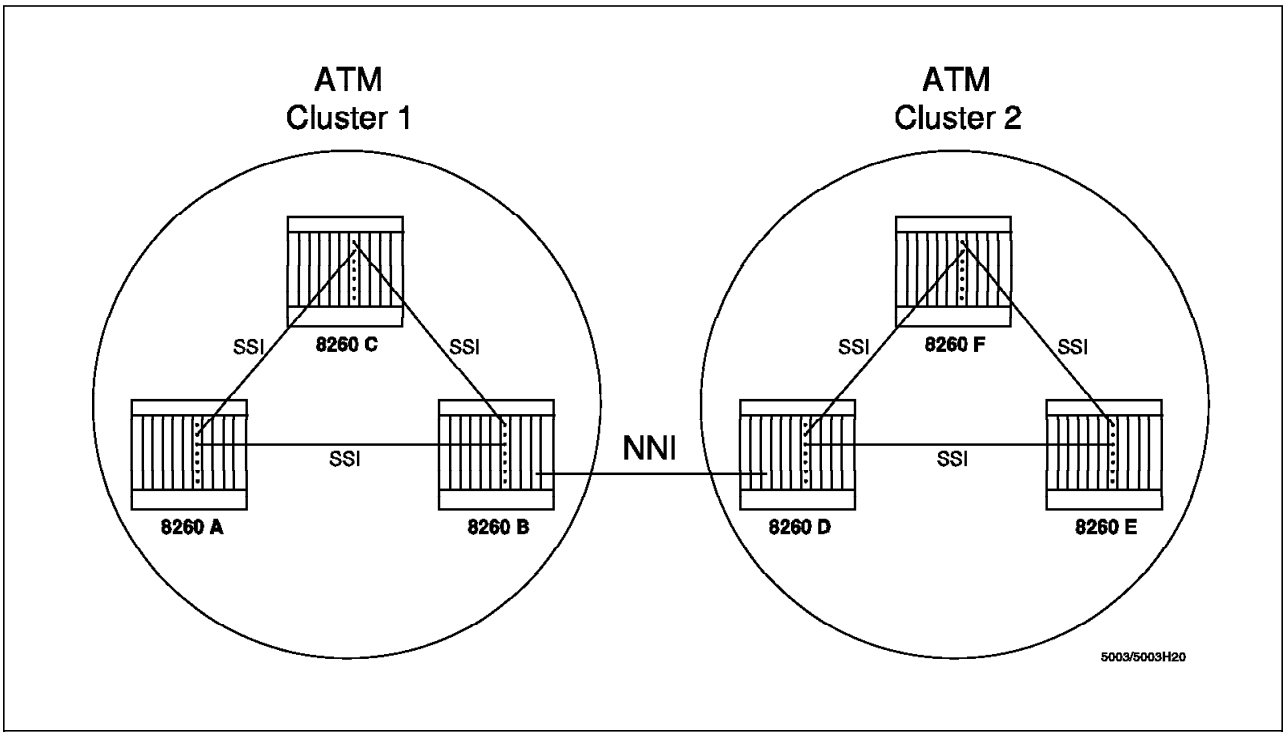



*Figure 72. Setting an NNI Link between Two Clusters*

To configure a connection between two adjacent clusters in the same subnetwork you must:

1. Configure the ports that are providing the connections between the two clusters as NNI ports. To do this, you must use the SET PORT command as described in "SET PORT Command."
2. Configure a *logical link* over the NNI connection within each boundary 8260 ATM switching subsystem. To do this, you must use the SET LOGICAL_LINK command as described in "SET LOGICAL_LINK Command" on page 140.

***SET PORT Command:*** To configure a port as an NNI port, you must use the SET PORT command. The format of this command is as follows:

```
SET PORT slot.port enable NNI
```

| | |
|---|---|
| **slot** | Slot number of the ATM media module in the ATM switching subsystem (1 to 17, except 9 to 11). |
| **port** | Port number of the ATM port. |
| **enable** | Enables the port. |
| **NNI** | Defines the port as an NNI port. |

**Note:** If the port that is to be configured as an NNI port is already enabled and configured as either UNI or SSI, you must disable the port before using the previous command, otherwise your command will be rejected. You can disable a port, using the following command:

```
SET PORT slot.port disable
```

***SET LOGICAL_LINK Command:*** To configure a logical link you must use the SET LOGICAL_LINK command on each of the boundary ATM switching subsystem. The format of this command is as follows:

```
SET LOGICAL LINK slot.port vpi acn role uni_version traffic_type
```

**slot** Slot number of ATM media module (1 to 17, except 9 to 11).

**port** Port number of the ATM port.

**vpi** Virtual path identifier used to identify the logical link (0-15). You must assign the same VPI to the ports at each side of the logical link. If you configure more than one logical link for a port, you must assign a different VPI for each logical link.

**acn** When connecting two ATM clusters in the same subnetwork, this is the cluster number of the remote boundary ATM switching subsystem.

**role** This parameter defines the Q.2931 role that is assumed by the boundary ATM switching subsystems. *Network_side* means that the 8260 assigns ATM labels for this logical link. *User_side* means that the 8260 does not assign labels. Note that one side must always be configured as the *network_side* and the other side as the *user_side*.

**uni_version** This parameter defines the version of UNI signalling protocol (3.0 or 3.1) used on this logical link.

**traffic_type** This parameter allows you to define the type of connections supported by the NNI link using this logical link.

When you specify reserved_bandwidth for a logical link, the NNI connection using this logical link will only be used for reserved_bandwidth calls. In this case you must specify the amount bandwidth available for the logical link. The reserved_bandwidth call will only be established if the requested bandwidth of the call can be satisfied by this NNI link.

When you specify non_reserved_bandwidth for a logical link, the NNI connection using that logical link will only be used for non_reserved_bandwidth calls. You cannot specify a bandwidth value for a non_reserved_bandwidth logical links, therefore, the amount of bandwidth available will depend upon how much bandwidth is available for the module and how many *reserved_bandwidth* calls have been established already.

When you specify any for a logical link, the NNI connection using that logical link will be used for both reserved_bandwidth and non_reserved_bandwidth calls.

With the logical links defined using any, you also specify a value for the bandwidth which determines the amount of bandwidth available for the reserved_bandwidth calls. In this case, the amount of bandwidth available for non_reserved_bandwidth calls will be a function of how much bandwidth is left over from the 212 Mbps available for that module used by the logical link.

If you have several NNI links going to the same remote cluster, that is if you are using link aggregate function, you will really have two aggregated links. One will be for all the NNI links that you have specified as reserved_bandwidth and any, the other will be all the NNI links that you have specified as non_reserved_bandwidth and what is left from the any bandwidth specification. For more information about link aggregation, please refer to 4.9.1.5, “Link Aggregation” on page 115.

### 4.10.3.1 NNI Connection Between Two Adjacent Clusters - Example

The following example shows the configuration of an NNI connection between the boundary 8260 ATM switching subsystems for the subnetwork shown in Figure 72 on page 139.

In this example, the port 4.2 on 8260B is connected to port 6.1 on 8260D. For this connection, we have decided that the logical link will use VPI 5 and will have a reserved bandwidth of 500 Kbps. Also, 8260B will have a network-side role and the UNI 3.1 signalling will be used on this logical link.

When choosing the VPI value for the logical link, you must take the following considerations into account:

1. The VPI used for the logical link can be any free VPI in the range of 0-15.
2. You must assign the same VPI to the ports at each side of the logical link.
3. If you configure more than one logical link for a port, you must assign a different VPI for each logical link.

- **8260B:**

```
8260B> set port 4.2 enable nni 1
8260B> set logical_link 4.2 5 2 network_side 3.1 reserved_bandwidth 500 2
Logical link set
8260B>
```

**1** This command configures port 4.2 as an NNI port.

**2** This command defined a logical link from 8260B (the boundary 8260 ATM switching subsystem within cluster 1) to 8260D (the boundary 8260 ATM switching subsystem within cluster 2).

- **8260D:**

```
8260D> set port 6.1 enable nni 1
8260D> set logical_link 6.1 5 1 user_side 3.1 reserved_bandwidth 500 2
Logical link set
8260D>
```

**1** This command configures port 6.1 as an NNI port.

**2** This command defined a logical link from 8260D (the boundary 8260 ATM switching subsystem within cluster 2) to 8260B (the boundary 8260 ATM switching subsystem within cluster 1).

To save your logical link settings, issue the following command in each 8260 ATM switching subsystem:

```
8260> save module_port
```

The following shows how to display the settings for the logical link defined for the boundary hub in cluster 1:

```
8260B> show logical_link
 Port Vpi Acn Side Mode Sig Traf  Bwidth Status              Index
------------------------------------------------------------------------
 4.02   1  02 netw enab 3.1  RB      500         UP             1

49 entries empty
8260B>
```

**Important**

All the 8260 ATM switching subsystems within a cluster use SSI connections to communicate with each other.

In the example given above, it is assumed that all the 8260s have already been configured to use SSI links to communicate with the other 8260s within their cluster. Therefore, after configuring the NNI link between the 8260B and 8260D, the stations attached to any of the ATM switching subsystems within these two clusters can communicate with each other without any requirement for additional definitions within the nonboundary ATM switching subsystems 8260A, 8260C, 8260E and 8260F. This is due to the fact that the boundary nodes (8260B and 8260D) will advertise themselves within their corresponding clusters (cluster 1 and cluster 2, respectively) as the gateway to the other cluster. This advertisement is part of the TRS functions, which is explained in 4.9, “Topology and Route Selection (TRS) Services” on page 108.

This allows you to connect two adjacent clusters together with only simple configuration required in the boundary ATM switching subsystems connecting the two clusters together.

### 4.10.4 Configuring Parallel NNI Connections between Adjacent Clusters within an ATM Subnetwork

You may have more than one NNI link between two clusters. In this case the multiple NNI links will be aggregated, as described in 4.9.1.5, “Link Aggregation” on page 115, to provide you with increased bandwidth and redundancy.

To configure parallel NNI links between two adjacent clusters, you must do the following:

- Configure each of the ports that are providing the connections between the two boundary ATM switching subsystems as NNI ports.

- Configure a *logical link* for each NNI connection between the two boundary ATM switching subsystems.

### 4.10.4.1 Parallel NNI Connections between Two Adjacent Clusters - Example

The following example shows the configuration of two NNI links between the two adjacent clusters shown in Figure 73.

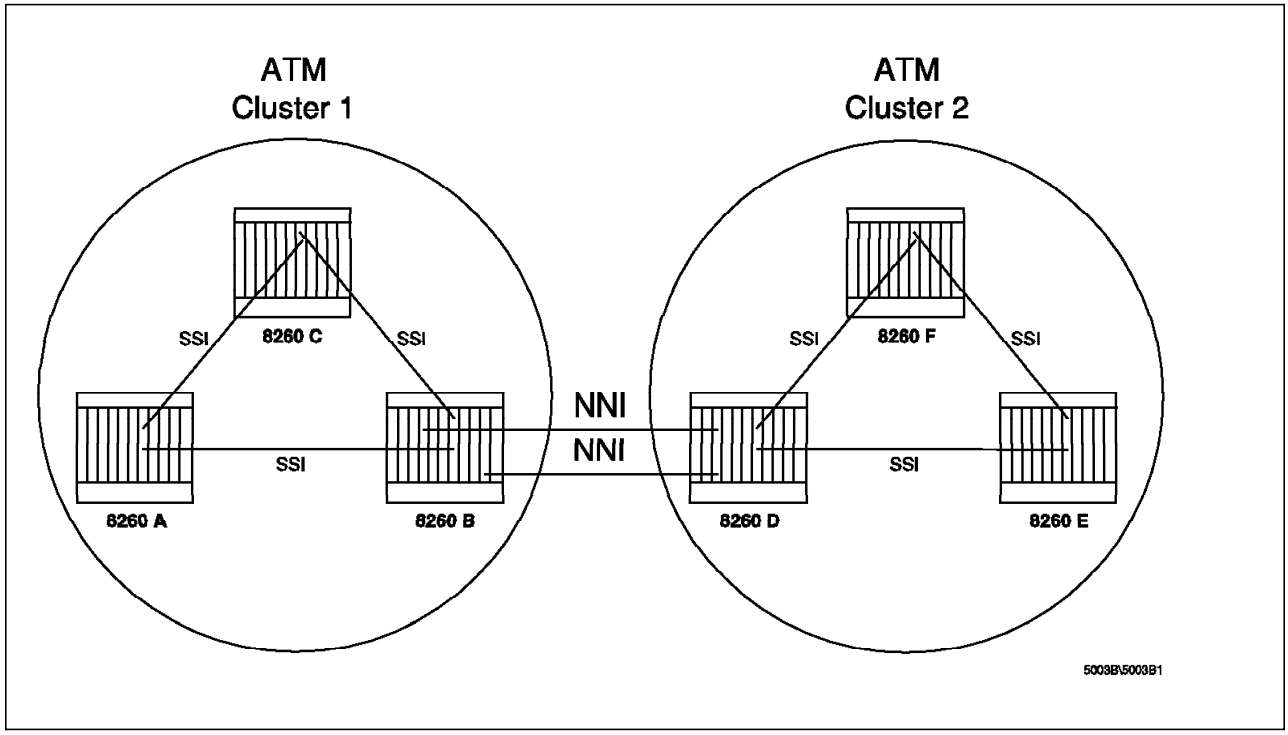



*Figure 73. Parallel NNI Connections between Two Adjacent Clusters - Example*

In this example, the ports 4.2 and 13.1 on 8260B are connected to ports 6.1 and 6.2 on 8260D, respectively. For this connection, we have decided that the logical link between ports 4.2 and 6.1 will use VPI 5 and the logical link between ports 13.1 and 6.2 will use VPI 6. Both logical links will have a reserved bandwidth of 500 Kbps. Also, 8260B will have a network-side role and the UNI 3.1 signalling will be used on the NNI link.

**Note**

When choosing the VPI value for the logical link, you must take the following considerations into account:

1. The VPI used for the logical link can be any free VPI in the range of 0-15 for all ATM media modules except for the 12-port 25-Mbps media module for which the VPI range is (0-3).
2. You must assign the same VPI to the ports at each side of the logical link.
3. If you configure more than one logical link for a port, you must assign a different VPI for each logical link.

1. **8260B:**

```
8260B> set port 4.2 enable nni  1
8260B> set logical_link 4.2 5 2 network_side 3.1 reserved_bandwidth 500  2
Logical link set
8260B> set port 13.1 enable nni  3
8260B> set logical_link 13.1 6 2 network_side 3.1 reserved_bandwidth500  4
Logical link set
8260B>
```

**1** This command configures port 4.2 as an NNI port.

**2** This command defined a logical link from 8260B (the boundary 8260 ATM switching subsystem within cluster 1) to 8260D (the boundary 8260 ATM switching subsystem within cluster 2).

**3** This command configures port 13.1 as an NNI port.

**4** This command defined a logical link from 8260B (the boundary 8260 ATM switching subsystem within cluster 1) to 8260D (the boundary 8260 ATM switching subsystem within cluster 2).

2. **8260D**:

```
8260D> set port 6.1 enable nni  1
8260D> set logical_link 6.1 5 1 user_side 3.1 reserved_bandwidth 500  2
Logical link set
8260D> set port 6.2 enable nni  3
8260D> set logical_link 6.2 6 1 user_side 3.1 reserved_bandwidth 500  4
Logical link set
8260D>
```

**1** This command configures port 6.1 as an NNI port.

**2** This command defines a logical link from 8260B (the boundary 8260 ATM switching subsystem within cluster 2) to 8260D (the boundary 8260 ATM switching subsystem within cluster 1).

**3** This command configures port 6.2 as an NNI port.

**4** This command defined a logical link from 8260D (the boundary 8260 ATM switching subsystem within cluster 2) to 8260B (the boundary 8260 ATM switching subsystem within cluster 1).

To save your logical link settings, issue the following command in each 8260 ATM switching subsystem:

```
8260B> save module_port
```

The following shows how to display the settings for the logical link defined for the boundary hub in cluster 1.

```
8260B> show logical_link
 Port Vpi Acn Side Mode Sig Traf  Bwidth Status              Index
-------------------------------------------------------------------------
 4.02   1  02 netw enab 3.1  RB      500         UP             1
13.01   1  02 netw enab 3.1  RB      500         UP             2

48 entries empty
8260B>
```

### 4.10.5 Configuring NNI Connections between Nonadjacent Clusters within an ATM Subnetwork

Two clusters are defined in this book to be nonadjacent when there is no direct physical connection between any of the ATM switching subsystems within these clusters, but the two clusters can reach each via an intermediate ATM network. The nonadjacent clusters can be interconnected using:

1. NNI connection using a logical link through the intermediate ATM network

   In this case the intermediate ATM network must be a private ATM network that supports IISP. An example of such a network is a cluster of the IBM 8260 ATM switching subsystems as shown in Figure 74 on page 146. In this example, ATM cluster 2 is used as an intermediate cluster to interconnect ATM cluster 1 and ATM cluster 3.

   In such a configuration, you can provide the following:

   - Communication between cluster 1 and cluster 2
   - Communication between cluster 3 and cluster 2
   - Communication between cluster 1 and cluster 3 using cluster 2 as an intermediate cluster

   The above communications are independent from each other and each should be configured separately. Also, these communications can share physical links or have their own dedicated physical links. In Figure 74 on page 146, a single physical link between each nonadjacent cluster and the intermediate cluster (links connecting 8260B to 8260D and 8260E to 8260G) is used to provide any-to-any connectivity between clusters 1, 2 and 3.

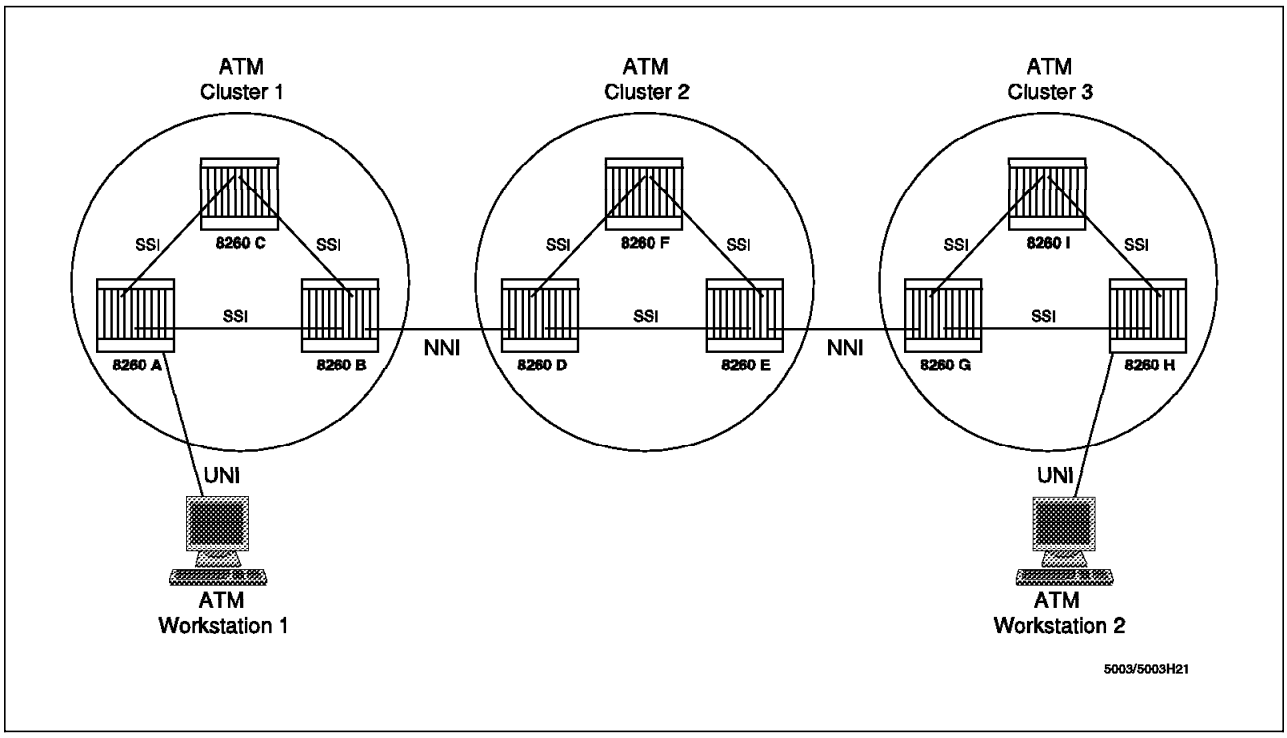



*Figure 74. Sharing Links for NNI Connections over an Intermediate Cluster*

Because of bandwidth and availability requirements, you may want to use separate links to carry the traffic between each pair of clusters. For example, in Figure 75 on page 147 the traffic is carried as follows:

- Link between 8260B and 8260D is used to carry the traffic between cluster 1 and cluster 2.
- Link between 8260E and 8260G is used to carry the traffic between cluster 2 and cluster 3.
- Links from 8260C to 8260F and 8260F to 8260I are used to carry the traffic between cluster 1 and cluster 3.

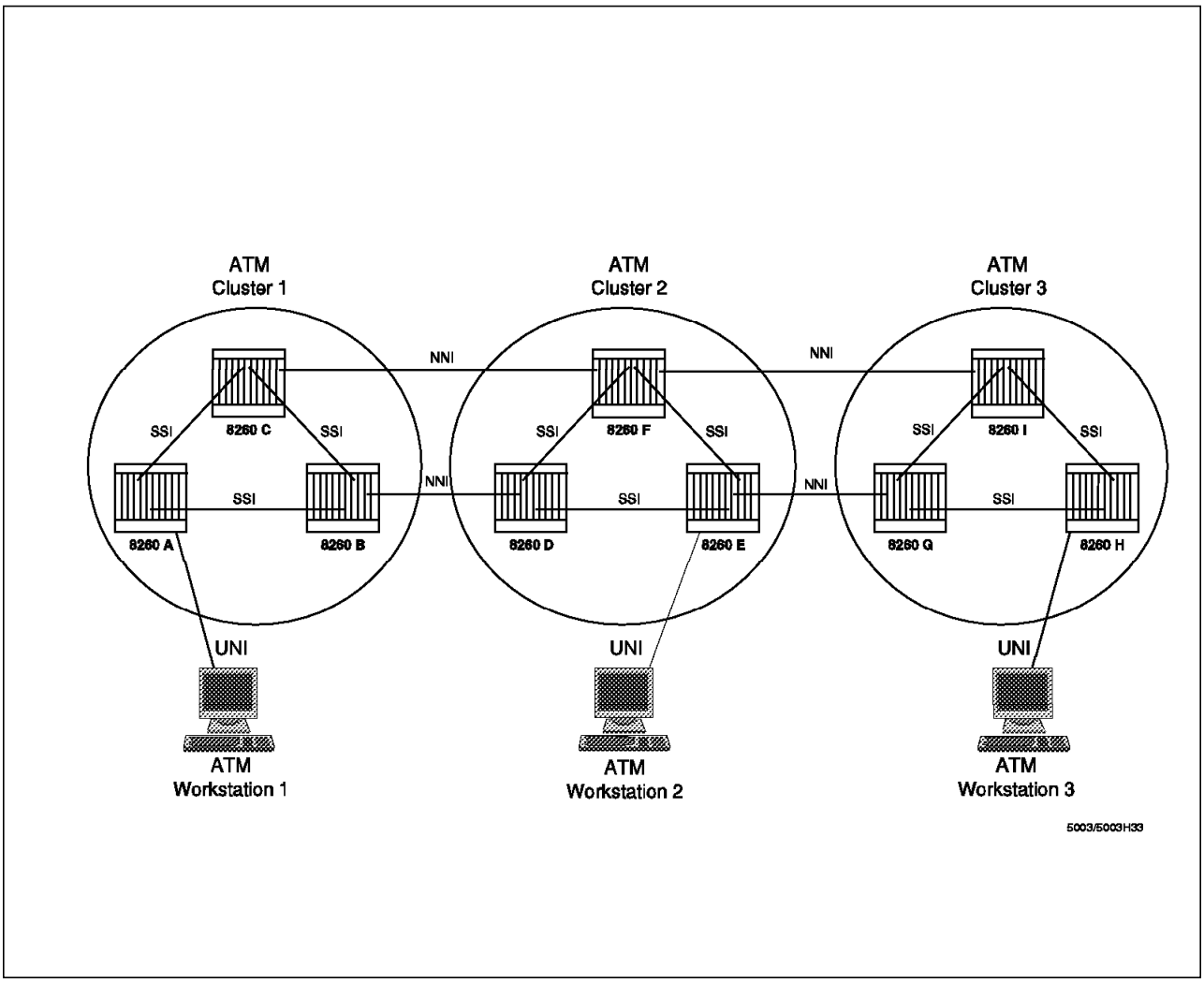



*Figure 75. Dedicated Links for NNI Connections over an Intermediate Cluster*

4.10.6, “NNI Connection between Nonadjacent Clusters Using Logical Links” on page 151 describes the configuration steps that should be taken to provide connectivity options, which were described earlier.

**Note**

The NNI links between each pair of clusters must originate and terminate in the same boundary ATM switching subsystems.

2. NNI connection using a permanent virtual path (PV) through the intermediate ATM network

   In this case, the intermediate ATM network can be either a private ATM network or a public ATM network.

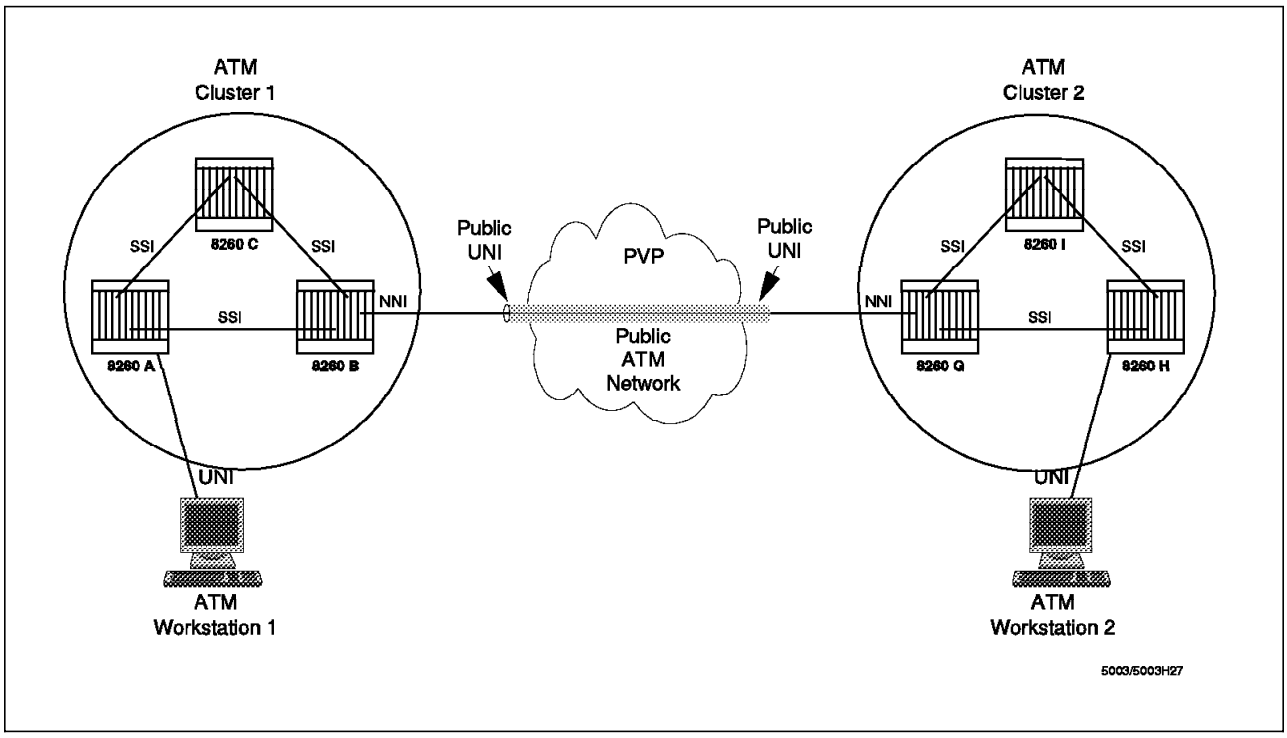



*Figure 76. NNI Connection Using PVP over a Public ATM Network*

If the intermediate network is a public ATM network, as shown in Figure 76, the public ATM network provides the path for establishing connections and exchanging data between various stations attached to the nonadjacent clusters , but the public ATM network is not part of your ATM network. This means that the stations attached to the public ATM network cannot communicate with the stations attached to these clusters. 4.10.7, "Non-Adjacent Clusters Connected by Permanent Virtual Path (PVP)" on page 165 describes the configuration steps that should be taken to provide this type of connection.

If the intermediate network is an ATM cluster of 8260s, it can provide you with the following:

a. It can be used to provide the path for establishing connections and exchanging data between various stations attached to the nonadjacent clusters, without the intermediate cluster being part of your ATM network. This is shown in Figure 77 on page 149.

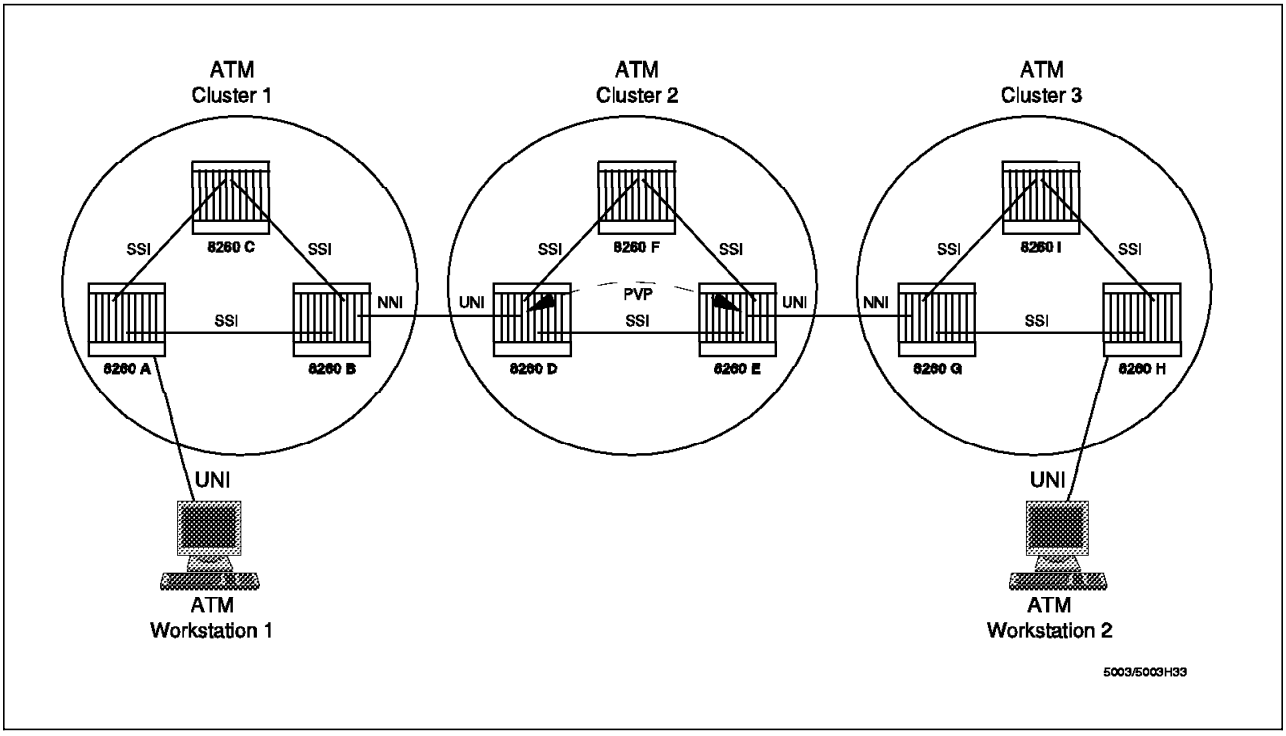



*Figure 77. NNI Connection Using PVP over an Intermediate Cluster. Only the nonadjacent clusters communicate with each other. The intermediate cluster cannot communicate with the other clusters.*

b. In addition to providing the path for establishing connections and exchanging data between various stations attached to the nonadjacent clusters, the intermediate cluster can be configured to be part of your ATM network so that you will have any-to-any connectivity between the stations attached to the nonadjacent clusters and the intermediate cluster. This is shown in Figure 78 on page 150. As can be seen, the same 8260s are used as the boundary ATM switching subsystems between all the clusters. For example, 8260B provides the connection from cluster 1 to both cluster 2 and cluster 3.

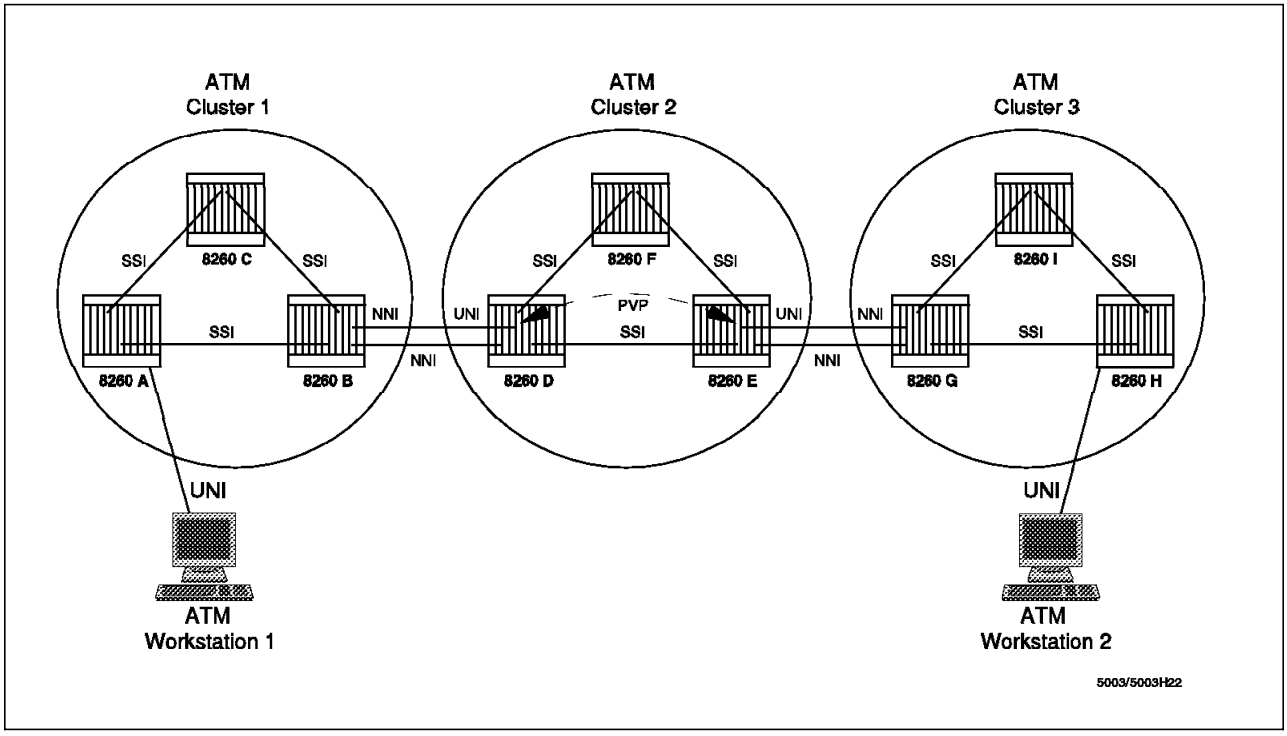



*Figure 78. NNI Connection Using PVP over an Intermediate Cluster. This configuration provides any-to-any communication between all the clusters. The nonadjacent clusters can communicate with each other as well as with the intermediate cluster.*

Although the NNI links connecting two clusters must originate in the same boundary ATM switching subsystems, different boundary ATM switching subsystems can be used to provide the NNI links to different clusters. For example, in Figure 79 on page 151, 8260B is the boundary node providing NNI connections from cluster 1 to cluster 2, whereas 8260C is the boundary node providing NNI connection from cluster 1 to cluster 3.

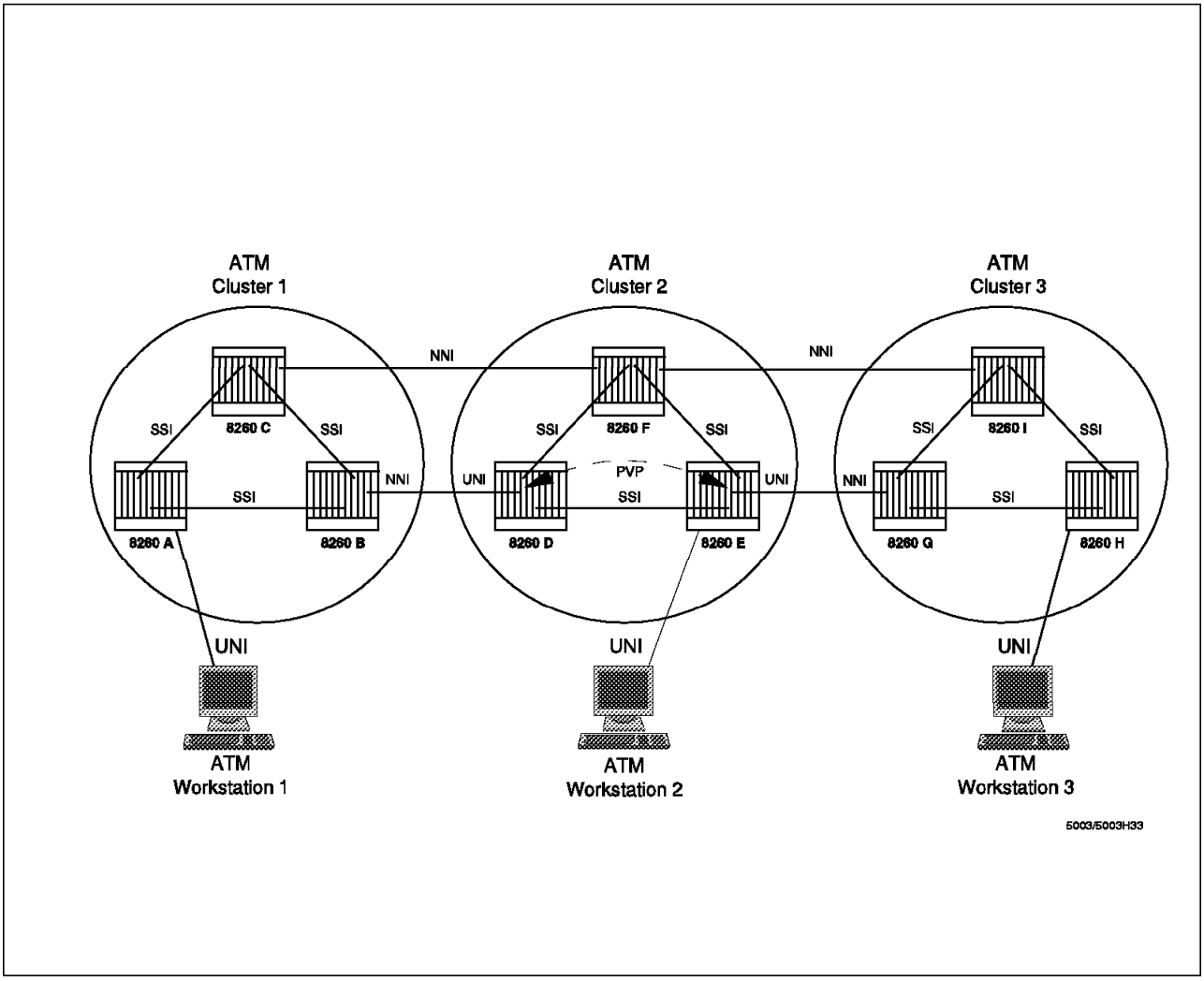



*Figure 79. NNI Connections Using Different Boundary ATM Switching Subsystems*

The following sections provide you with the details of the configurations steps described earlier.

### 4.10.6 NNI Connection between Nonadjacent Clusters Using Logical Links

To configure an NNI connection using a logical link between two nonadjacent clusters within the same subnetwork (as shown in Figure 80 on page 153) you must:

1. Configure the connecting nonadjacent cluster and the intermediate clusters as NNI ports. Note that this should be done at both ends of the link. In our example, the ports on 8260B, 8260D, and 8260G provide the physical connections between the clusters 1, 2 and 3. These clusters must be defined as NNI links.

2. Configure logical links between the nonadjacent clusters (cluster 1 and 3 in our example) and the intermediate cluster (cluster 2 in our example). This step uses the SET LOGICAL_LINK command as discussed in "SET LOGICAL_LINK Command" on page 140 and is required to provide communication between the nonadjacent clusters and the intermediate cluster. If you do not intend to allow the stations attached to the nonadjacent

clusters and the intermediate cluster to communicate with each other, you must skip this step.

3. Configure a set of logical links between the nonadjacent clusters through the intermediate cluster. The purpose of this set of logical links is to provide a pipe between the two nonadjacent clusters through the intermediate cluster. This set of logical links will consist of the following logical links:

   a. A logical link from the boundary ATM switching subsystems in the first nonadjacent cluster (8260B in our example) to the boundary ATM switching subsystems within the intermediate cluster (8260D in our example). The *acn* parameter in this logical link should specify the second nonadjacent cluster (cluster 3 in our example) as the destination cluster.

   b. A logical link from the boundary ATM switching subsystems in the intermediate cluster (8260D in our example) to the boundary ATM switching subsystems in the first nonadjacent cluster (8260B in our example) providing the logical link in the opposite direction to the logical link specified in step 3a. However, the *acn* parameter for this logical link should specify the first cluster (cluster 1 in our example) as the destination.

   c. A logical link from the boundary ATM switching subsystems in the second nonadjacent cluster (8260G in our example) to the boundary ATM switching subsystems within the intermediate cluster (8260E in our example). The *acn* parameter in this logical link should specify the first nonadjacent cluster (cluster 1 in our example) as the destination cluster.

   d. A logical link from the boundary ATM switching subsystems in the intermediate cluster (8260E in our example) to the boundary ATM switching subsystems in the second nonadjacent cluster (8260G in our example) providing the logical link in the opposite direction to the logical link specified in step 3c. However, the *acn* parameter for this logical link should specify the second cluster (cluster 3 in our example) as the destination cluster.

      **Note:** All the previous logical links must specify the same VPI number when possible. A 12-port 25-Mbps Module has a VPI range of 0 to 3, and all other ATM media modules range from 0 to 15. Keeping the same VPI number is useful but not mandatory. The following examples propose solutions using the same VPI numbers.

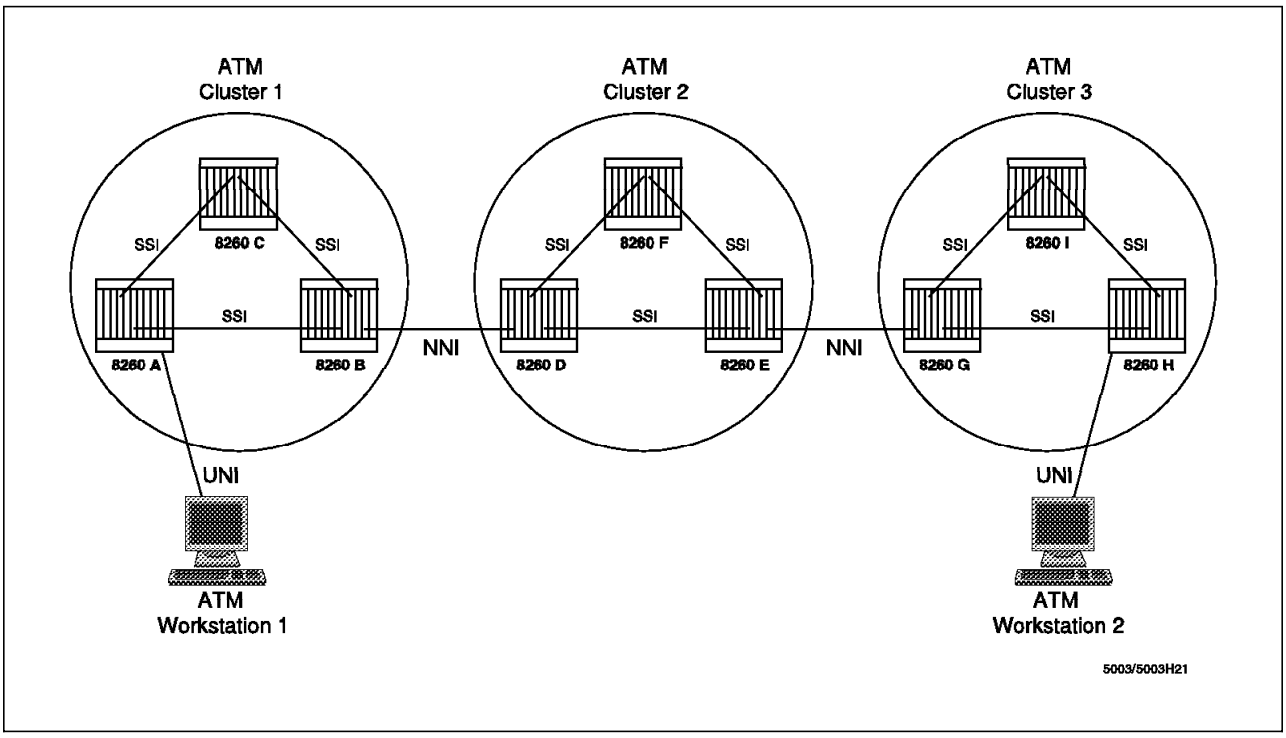



*Figure 80. Sharing Links for NNI Connections over an Intermediate Cluster*

The following examples demonstrate how to perform the previous steps to configure NNI connections between nonadjacent clusters.

### 4.10.6.1 NNI Connection Between Nonadjacent Clusters Using Logical Links - Example 1

In this scenario, we assume that there are three 8260s, each configured as a separate cluster as shown in Figure 81 on page 154.

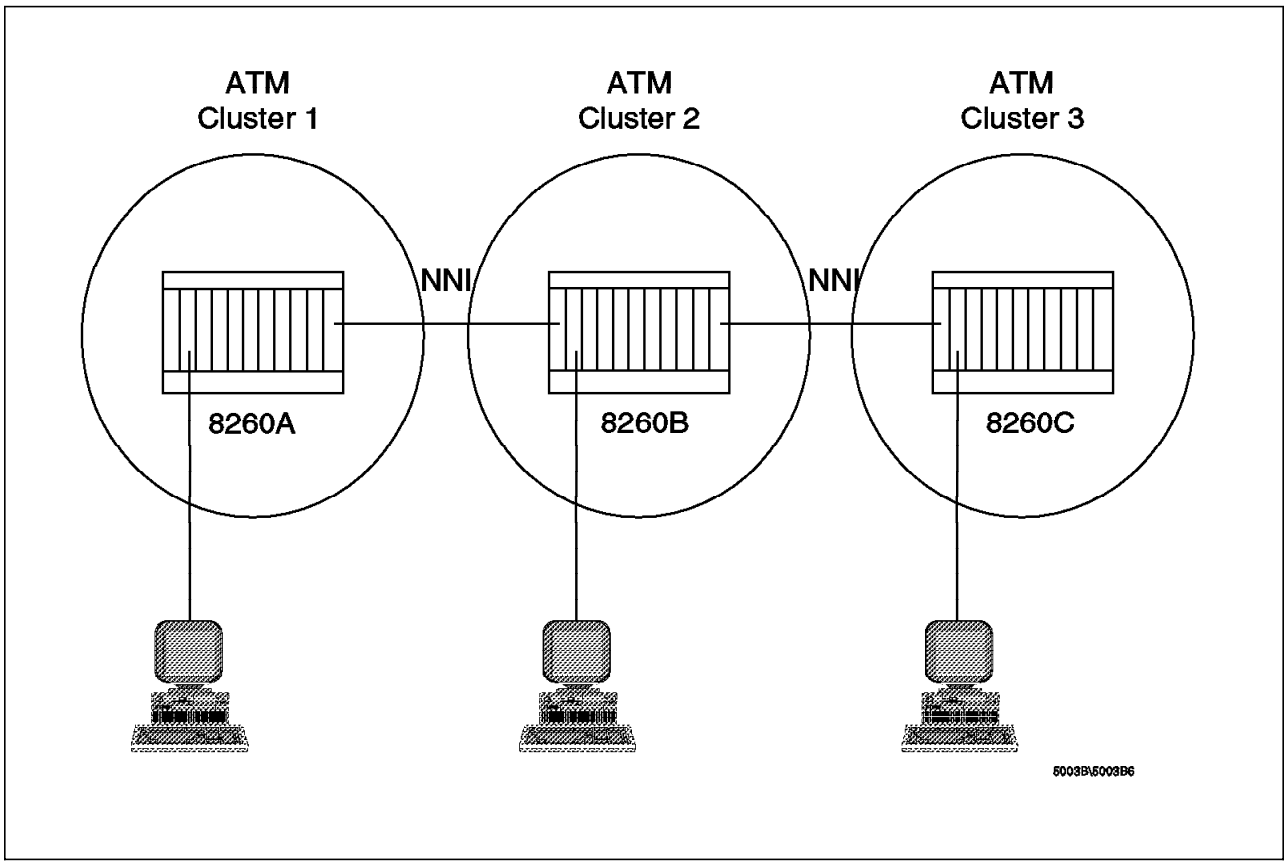



*Figure 81. Nonadjacent Clusters within an ATM Subnetwork - Example 1*

Table 34 shows the ATM addresses configured for the 8260s used in this example.

*Table 34. ATM Addresses for Scenario 1*

| Cluster | Hub | ATM Address |
|---|---|---|
| Cluster 1 | 8260A | 39.9.85.11.11.11.11.11.11.11.11.1.1.40.00.00.00.00.01.00 |
| Cluster 2 | 8260B | 39.9.85.11.11.11.11.11.11.11.11.2.2.40.00.00.00.00.02.00 |
| Cluster 3 | 8260C | 39.9.85.11.11.11.11.11.11.11.11.3.3.40.00.00.00.00.03.00 |

The following are the definitions that are required for this example:

1. **8260A:**

```
8260A> set port 3.1 enable nni  1
8260A> set logical_link 3.1 1 2 user_side 3.1 non_reserved_bandwidth  2
8260A> set logical_link 3.1 2 3 user_side 3.1 non_reserved_bandwidth  3
```

**1** This command defines the NNI link between 8260A in cluster 1 and 8260B in cluster 2.

**2** This command defines the logical link used to provide connectivity from cluster 1 to cluster 2. If you do not intend to allow ATM stations in cluster 1 and cluster 2 to communicate with each other, you may skip this step as it has no bearing in providing connectivity between cluster 1 and cluster 3.

Note that the VPI used for this logical link is 1, and 8260A assumes the role of user-side on this NNI connection.

**3** This command defines the logical link used to provide connectivity from cluster 1 to cluster 3 using cluster 2 as the intermediate cluster. The VPI used for this logical link is 2. This VPI must must be different from the VPI chosen in **2** as both of these VPIs are using the same physical link. On this logical link, the 8260A is configured to assume the role of network-side. Note that the role of 8260A on this logical link is independent of the role of 8260A for the logical link defined in **2**, although they both are using the same physical link. Also, this VPI must be the same as the VPI used in steps **5** and **6** of 8260B and step **3** of 8260C definitions.

2. **8260B:**

```
8260B> set port 3.1 enable nni  1
8260B> set port 3.2 enable nni  2
8260B> set logical_link 3.1 1 1 network_side 3.1 non_reserved_bandwidth  3
8260B> set logical_link 3.2 1 3 network_side 3.1 non_reserved_bandwidth  4
8260B> set logical_link 3.1 2 1 netwok_side 3.1 non_reserved_bandwidth  5
8260B> set logical_link 3.2 2 3 netwok_side 3.1 non_reserved_bandwidth  6
```

**1** This command defines the NNI link between 8260B in cluster 2 and 8260A in cluster 1.

**2** This command defines the NNI link between 8260B in cluster 2 and 8260C in cluster 3.

**3** This command defines the logical link used to provide connectivity from cluster 2 to cluster 1. The VPI used for this logical link is 1, which must be the same as the VPI chosen in step **2** of the 8260A configuration. Also, 8260B assumes the role of network-side on this NNI connection, because the 8260A is configured to assume the role of user-side. If you do not intend to allow ATM stations in cluster 1 and cluster 2 to communicate with each other, you may skip this step as it has no bearing in providing connectivity between cluster 1 and cluster 3.

**4** This command defines the logical link used to provide connectivity from cluster 2 to cluster 3. The VPI used for this logical link is 1, which must be the same as the VPI chosen in step **2** of the 8260C configuration. Also, 8260B assumes the role of network-side on this NNI connection, because the 8260C is configured to assume the role of user-side. If you do not intend to allow ATM stations in cluster 2 and cluster 3 to communicate with each other, you may skip this step as it has no bearing in providing connectivity between cluster 1 and cluster 3.

**5** This command defines segments of the logical link used to provide connectivity from cluster 1 to cluster 3. The VPI used for this logical link is 2, which must be the same as the VPI chosen in step **3** of the 8260A, step **3** of 8260C and step **6** of 8260B configuration. Also, 8260B assumes the role of network-side on this NNI connection, because the 8260A is configured to assume the role of user-side.

**6** This command defines a segment of the logical link, which is used to provide connectivity from cluster 1 to cluster 3. The VPI used for this logical link is 2, which must be the same as the VPI chosen in step **3** of the 8260A, step **3** of 8260C and step **5** of 8260B configuration. Also, 8260B assumes

the role of network-side on this NNI connection, because the 8260C is configured to assume the role of user-side.

3. **8260C:**

```
8260C> set port 3.1 enable nni  1
8260C> set logical_link 3.1 1 2 user_side 3.1 non_reserved_bandwidth  2
8260C> set logical_link 3.1 2 1 user_side 3.1 non_reserved_bandwidth  3
```

**1** This command defines the NNI link between 8260C in cluster 3 and 8260B in cluster 2.

**2** This command defines the logical link used to provide connectivity from cluster 3 to cluster 2. Note that the VPI used for this logical link is 1, which is the same as the VPI specified in step **4** of 8260B configuration. Also, 8260C assumes the role of user-side on this NNI connection as the 8260B is configured as the network-side on this logical link. If you do not intend to allow ATM stations in cluster 2 and cluster 3 to communicate with each other, you may skip this step as it has no bearing in providing connectivity between cluster 1 and cluster 3.

**3** This command defines the logical link used to provide connectivity from cluster 3 to cluster 1 using cluster 2 as the intermediate cluster. The VPI used for this logical link is 2. This VPI must must be the same as the VPI chosen in step **3** of 8260A and steps **5** and **6** of 8260B configuration. On this logical link, the 8260C is configured to assume the role of network-side. Note that the role of 8260C on this logical link is independent of the role of 8260C for the logical link defined in step **2**, although they both are using the same physical link.

### 4.10.6.2 NNI Connection Between Nonadjacent Clusters Using Logical Links - Example 2

In this example, we assume that there are three clusters connected to each other as shown in Figure 82 on page 157.

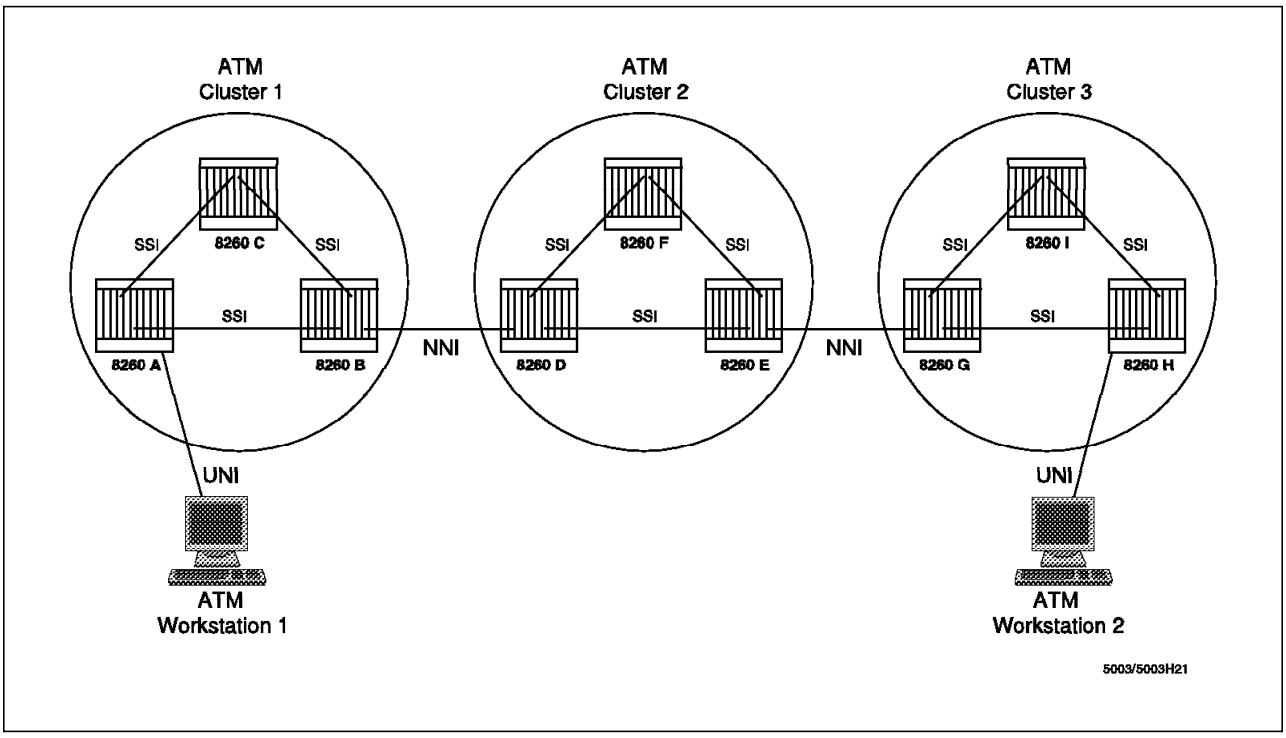



*Figure 82. Nonadjacent Clusters within an ATM Subnetwork - Example 2*

Table 34 on page 154 shows the ATM addresses configured for the 8260s used in this example.

*Table 35. ATM Addresses for Scenario 2*

| Cluster | Hub | ATM Address |
|---|---|---|
| Cluster 1 | 8260A | 39.9.85.11.11.11.11.11.11.11.11.1.1.40.00.00.00.00.01.00 |
| | 8260B | 39.9.85.11.11.11.11.11.11.11.11.1.2.40.00.00.00.00.02.00 |
| | 8260C | 39.9.85.11.11.11.11.11.11.11.11.1.3.40.00.00.00.00.03.00 |
| Cluster 2 | 8260D | 39.9.85.11.11.11.11.11.11.11.11.2.1.40.00.00.00.00.04.00 |
| | 8260E | 39.9.85.11.11.11.11.11.11.11.11.2.2.40.00.00.00.00.05.00 |
| | 8260F | 39.9.85.11.11.11.11.11.11.11.11.2.3.40.00.00.00.00.06.00 |
| Cluster 3 | 8260G | 39.9.85.11.11.11.11.11.11.11.11.3.1.40.00.00.00.00.07.00 |
| | 8260H | 39.9.85.11.11.11.11.11.11.11.11.3.2.40.00.00.00.00.08.00 |
| | 8260I | 39.9.85.11.11.11.11.11.11.11.11.3.3.40.00.00.00.00.09.00 |

The following are the definitions required for this example:

- **8260A:**

```
8260A> set port 4.1 enable ssi  1
8260A> set port 6.1 enable ssi  2
```

**1** This command defines the SSI link between 8260A and 8260B within cluster 1.

**2** This command defines the SSI link between 8260A and 8260C within cluster 1.

- **8260B:**

```
8260B> set port 5.2 enable ssi  1
8260B> set port 6.3 enable ssi  2
8260B> set port 7.1 enable nni  3
8260B> set logical_link 7.1 1 2 user_side 3.1 non_reserved_bandwidth  4
8260B> set logical_link 7.1 2 3 user_side 3.1 non_reserved_bandwidth  5
```

**1** This command defines the SSI link between 8260B and 8260A within cluster 1.

**2** This command defines the SSI link between 8260B and 8260C within cluster 1.

**3** This command defines the NNI link between 8260B within cluster 1 and 8260D within cluster 2.

**4** This command defines the logical link used to provide connectivity from cluster 1 to cluster 2. Note that the VPI used for this logical link is 1, and 8260B assumes the role of user-side on this NNI connection. You may skip this step if you do not intend to allow stations connected to cluster 1 to communicate with stations connected to cluster 2.

**5** This command defines a logical link (one in a series of logical links) which are used to provide a pipe between cluster 1 and cluster 3 using the cluster 2 as the intermediate cluster. The VPI used for this logical link is 2. This VPI must be different from the VPI chosen in **3** as both of these VPIs are using the same physical link. On this logical link, the 8260B is configured to assume the role of user-side. Note that the role of 8260B on this logical link is independent of the role of 8260B for the logical link defined in step **3**, although they both are using the same physical link.

- **8260C:**

```
8260C> set port 4.1 enable ssi  1
8260C> set port 6.1 enable ssi  2
```

**1** This command defines the SSI link between 8260C and 8260A within cluster 1.

**2** This command defines the SSI link between 8260C and 8260B within cluster 1.

- **8260D:**

```
8260D> set port 5.2 enable ssi  1
8260D> set port 6.3 enable ssi  2
8260D> set port 7.1 enable nni  3
8260D> set logical_link 7.1 1 1 network.side 3.1 non_reserved_bandwidth  4
8260D> set logical_link 7.1 2 1 network.side 3.1 non_reserved_bandwidth  5
```

**1** This command defines the SSI link between 8260D and 8260E within cluster 2.

**2** This command defines the SSI link between 8260D and 8260F within cluster 2.

**3** This command defines the NNI link between 8260D within cluster 2 and 8260B within cluster 1.

**4** This command defines the logical link used to provide connectivity from cluster 2 to cluster 1. Note that the VPI used for this logical link is 1, and 8260D assumes the role of network-side on this NNI connection. The VPI must be the same as the one specified in step **4** of 8260B configuration. You may skip this step if you do not plan to allow stations connected to cluster 2 to communicate with stations connected to cluster 1.

**5** This command defines a logical link (one in a series of logical links), which are used to provide a pipe between cluster 1 and cluster 3 using cluster 2 as the intermediate cluster. The VPI used for this logical link is 2. This VPI must be the same as the VPI specified in **5** of the 8260B definitions. 8260D assumes the role of the network-side on this link. Note that the role of 8260D on this logical link is independent of the role of 8260D for the logical link defined in step **4**, however, it must be specified to complement the role of 8260B as defined in **5** of the 8260B definitions.

- **8260E:**

```
8260E> set port 5.2 enable ssi 1
8260E> set port 6.3 enable ssi 2
8260E> set port 7.1 enable nni 3
8260E> set logical_link 7.1 1 3 network_side 3.1 non_reserved_bandwidth 4
8260E> set logical_link 7.1 2 3 network_side 3.1 non_reserved_bandwidth 5
```

**1** This command defines the SSI link between 8260E and 8260D within cluster 2.

**2** This command defines the SSI link between 8260E and 8260F within cluster 2.

**3** This command defines the NNI link between 8260E within cluster 2 and 8260G within cluster 3.

**4** This command defines the logical link used to provide connectivity from cluster 2 to cluster 3. Note that the VPI used for this logical link is 1, and 8260E assumes the role of network-side on this NNI connection. You may skip this step if you do not plan to allow stations connected to cluster 2 to communicate with stations connected to cluster 3.

**5** This command defines a logical link (in a series of logical links), which is used to provide a pipe from cluster 2 to cluster 3 using cluster 2 as the intermediate cluster. The VPI used for this logical link is 2, which must be the same as the VPI used in step **5** of 8260B, 8260D, and 8260G definitions. On this logical link, the 8260E is configured to assume the role of network-side, which must complement the role specified for the 8260G as specified in step **5** of the 8260G definitions. Note that the role of 8260E on this logical link is independent of the role of 8260E for the logical link defined in **4**, although they both are using the same physical link.

- **8260F:**

```
8260F> set port 4.1 enable ssi 1
8260F> set port 6.1 enable ssi 2
```

**1** This command defines the SSI link between 8260F and 8260D within cluster 1.

**2** This command defines the SSI link between 8260F and 8260E within cluster 1.

- **8260G:**

```
8260G> set port 5.2 enable ssi  1
8260G> set port 6.3 enable ssi  2
8260G> set port 7.1 enable nni  3
8260G> set logical_link 7.1 1 2 user.side 3.1 non_reserved_bandwidth  4
8260G> set logical_link 7.1 2 1 user.side 3.1 non_reserved_bandwidth  5
```

**1** This command defines the SSI link between 8260G and 8260H within cluster 3.

**2** This command defines the SSI link between 8260G and 8260I within cluster 3.

**3** This command defines the NNI link between 8260G within cluster 3 and 8260E within cluster 2.

**4** This command defines the logical link used to provide connectivity from cluster 2 to cluster 3. Note that the VPI used for this logical link is 1, and 8260G assumes the role of user-side on this NNI connection. This VPI must be the same as the VPI used in step **4** of the 8260E definition. You may skip this step if you do not plan to allow stations connected to cluster 3 to communicate with stations connected to cluster 2.

**5** This command defines a logical link (in a series of logical links), which are used to provide a pipe from cluster 1 to cluster 3 using cluster 2 as the intermediate cluster. The VPI used for this logical link is 2, which must be the same as the VPI specified in step **5** of 8260B, 8260D, and 8260E definitions. On this logical link, the 8260G is configured to assume the role of user side which complements the role of 8260E as defined in step **5** of the 8260 E definitions. Note that the role of 8260G on this logical link is independent of the role of 8260G for the logical link defined in **2**, although they both are using the same physical link.

- **8260H:**

```
8260H> set port 4.1 enable ssi  1
8260H> set port 6.1 enable ssi  2
```

**1** This command defines the SSI link between 8260H and 8260G within cluster 3.

**2** This command defines the SSI link between 8260H and 8260I within cluster 3.

- **8260I:**

```
8260I> set port 4.1 enable ssi  1
8260I> set port 6.1 enable ssi  2
```

**1** This command defines the SSI link between 8260I and 8260G within cluster 3.

**2** This command defines the SSI link between 8260I and 8260H within cluster 3.

### 4.10.6.3 NNI Connection Between Nonadjacent Clusters Using Logical Links - Example 3

In this example, we assume that there are three clusters connected to each other as shown in Figure 83.

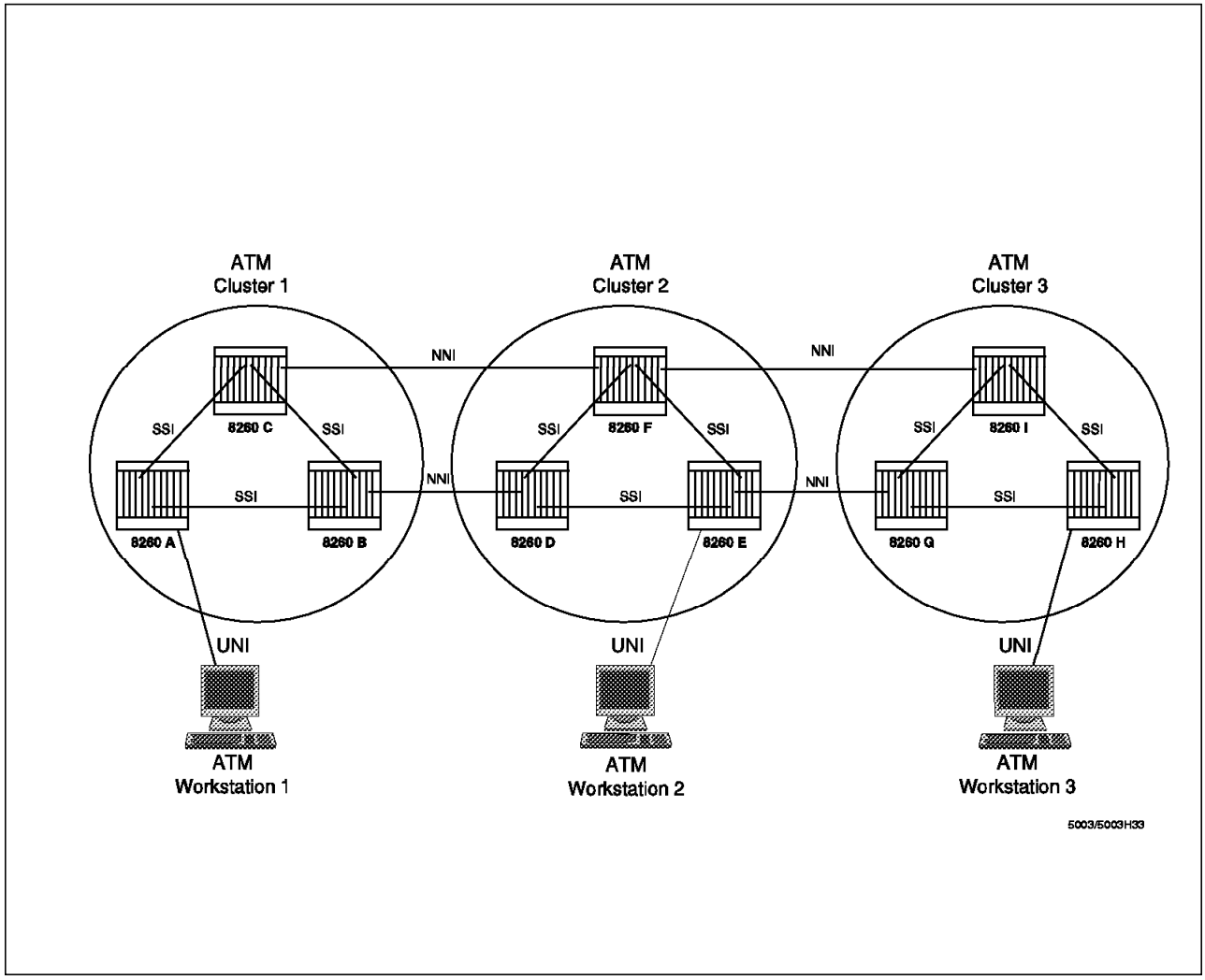



*Figure 83. Nonadjacent Clusters within an ATM Subnetwork - Example 3*

The difference between this example and example 2 is that in example 2 we used a single link from each nonadjacent cluster to the intermediate cluster to provide all the NNI connections. In this example, we are using a dedicated link to carry each NNI connection as follows:

- Link between 8260B and 8260D is used to carry the NNI traffic between cluster 1 and cluster 2.
- Link between 8260E and 8260G is used to carry the NNI traffic between cluster 2 and cluster 3.

- Links from 8260C to 8260F and from 8260I to 8260F are used to carry the NNI traffic between cluster 1 and cluster 3.

Table 36 shows the ATM addresses configured for the 8260s used in this example.

*Table 36. ATM Addresses for Example 3*

| Cluster | Hub | ATM Address |
|---|---|---|
| Cluster 1 | 8260A | 39.9.85.11.11.11.11.11.11.11.11.1.1.40.00.00.00.00.01.00 |
| | 8260B | 39.9.85.11.11.11.11.11.11.11.11.1.2.40.00.00.00.00.02.00 |
| | 8260C | 39.9.85.11.11.11.11.11.11.11.11.1.3.40.00.00.00.00.03.00 |
| Cluster 2 | 8260D | 39.9.85.11.11.11.11.11.11.11.11.2.1.40.00.00.00.00.04.00 |
| | 8260E | 39.9.85.11.11.11.11.11.11.11.11.2.2.40.00.00.00.00.05.00 |
| | 8260F | 39.9.85.11.11.11.11.11.11.11.11.2.3.40.00.00.00.00.06.00 |
| Cluster 3 | 8260G | 39.9.85.11.11.11.11.11.11.11.11.3.1.40.00.00.00.00.07.00 |
| | 8260H | 39.9.85.11.11.11.11.11.11.11.11.3.2.40.00.00.00.00.08.00 |
| | 8260I | 39.9.85.11.11.11.11.11.11.11.11.3.3.40.00.00.00.00.09.00 |

The following definitions are required for this example:

- **8260A:**

```
8260A> set port 4.1 enable ssi  1
8260A> set port 6.1 enable ssi  2
```

**1** This command defines the SSI link between 8260A and 8260B within cluster 1.

**2** This command defines the SSI link between 8260A and 8260C within cluster 1.

- **8260B:**

```
8260B> set port 5.2 enable ssi  1
8260B> set port 6.3 enable ssi  2
8260B> set port 7.1 enable nni  3
8260B> set logical_link 7.1 1 2 user_side 3.1 non_reserved_bandwidth  4
```

**1** This command defines the SSI link between 8260B and 8260A within cluster 1.

**2** This command defines the SSI link between 8260B and 8260C within cluster 1.

**3** This command defines the NNI link between 8260B within cluster 1 and 8260D within cluster 2.

**4** This command defines the logical link used to provide connectivity from cluster 1 to cluster 2. Note that the VPI used for this logical link is 1, and 8260B assumes the role of user-side on this NNI connection.

- **8260C:**

```
8260C> set port 4.1 enable ssi  1
8260C> set port 6.1 enable ssi  2
8260C> set port 7.1 enable nni  3
8260C> set logical_link 7.1 2 3 user_side 3.1 non_reserved_bandwidth  4
```

**1** This command defines the SSI link between 8260C and 8260A within cluster 1.

**2** This command defines the SSI link between 8260C and 8260B within cluster 1.

**3** This command defines the NNI link between 8260C and 8260F. This NNI link will be used to carry traffic between cluster 1 and cluster 3.

**4** This command defines a logical link in a series of logical links which are used to provide a pipe between cluster 1 and cluster 3 using the cluster 2 as the intermediate cluster. The VPI used for this logical link is 2. On this logical link, the 8260C is configured to assume the role of user side.

- **8260D:**

```
8260D> set port 5.2 enable ssi  1
8260D> set port 6.3 enable ssi  2
8260D> set port 7.1 enable nni  3
8260D> set logical_link 7.1 1 1 user_side 3.1 non_reserved_bandwidth  4
```

**1** This command defines the SSI link between 8260D and 8260E within cluster 2.

**2** This command defines the SSI link between 8260D and 8260F within cluster 2.

**3** This command defines the NNI link between 8260D within cluster 2 and 8260B within cluster 1.

**4** This command defines the logical link used to provide connectivity from cluster 2 to cluster 1. Note that the VPI used for this logical link is 1, and 8260D assumes the role of network-side on this NNI connection. This VPI must be the same as the VPI specified in step **4** of 8260B configuration.

- **8260E:**

```
8260E> set port 5.2 enable ssi  1
8260E> set port 6.3 enable ssi  2
8260E> set port 7.1 enable nni  3
8260E> set logical_link 7.1 1 3 network_side 3.1 non_reserved_bandwidth  4
```

**1** This command defines the SSI link between 8260E and 8260D within cluster 2.

**2** This command defines the SSI link between 8260E and 8260F within cluster 2.

**3** This command defines the NNI link between 8260E within cluster 2 and 8260G within cluster 3.

**4** This command defines the logical link used to provide connectivity from cluster 2 to cluster 3. Note that the VPI used for this logical link is 1, and 8260E assumes the role of network-side on this NNI connection.

- **8260F:**

```
8260F> set port 4.1 enable ssi  1
8260F> set port 6.1 enable ssi  2
8260F> set port 7.1 enable nni  3
8260F> set port 7.2 enable nni  4
8260F> set logical_link 7.1 2 1 user_side 3.1 non_reserved_bandwidth  5
8260F> set logical_link 7.2 2 3 user_side 3.1 non_reserved_bandwidth  6
```

**1** This command defines the SSI link between 8260F and 8260D within cluster 1.

**2** This command defines the SSI link between 8260F and 8260E within cluster 1.

**3** This command defines the NNI link between 8260F and 8260C within cluster 1.

**4** This command defines the NNI link between 8260F and 8260I within cluster 3.

**5** This command defines a logical link (in a series of logical links), which are used to provide a pipe between cluster 1 and cluster 3 using cluster 2 as the intermediate cluster. The VPI used for this logical link is 2. This VPI must be the same as the VPI specified in **4** of the 8260C definitions. 8260F assumes the role of the network-side on this link.

**6** This command defines a logical link (in a series of logical links), which are used to provide a pipe from cluster 1 to cluster 3 using cluster 2 as the intermediate cluster. The VPI used for this logical link is 2, which must be the same as the VPI used in **4** of 8260C, 8260F, and 8260I definitions. On this logical link, the 8260F is configured to assume the role of network-side, which must complement the role specified for the 8260I in step **4** of the 8260I configuration.

- **8260G:**

```
8260G> set port 5.2 enable ssi  1
8260G> set port 6.3 enable ssi  2
8260G> set port 7.1 enable nni  3
8260G> set logical_link 7.1 1 2 user.side 3.1 non_reserved_bandwidth  4
```

**1** This command defines the SSI link between 8260G and 8260H within cluster 3.

**2** This command defines the SSI link between 8260G and 8260I within cluster 3.

**3** This command defines the NNI link between 8260G within cluster 3 and 8260E within cluster 2.

**4** This command defines the logical link used to provide connectivity from cluster 2 to cluster 3. Note that the VPI used for this logical link is 1, and 8260G assumes the role of user-side on this NNI connection. This VPI must be the same as the VPI used in step **4** of the 8260E configuration.

- **8260H:**

```
8260H> set port 4.1 enable ssi  1
8260H> set port 6.1 enable ssi  2
```

**1** This command defines the SSI link between 8260H and 8260G within cluster 3.

**2** This command defines the SSI link between 8260H and 8260I within cluster 3.

- **8260I:**

```
8260I> set port 4.1 enable ssi  1
8260I> set port 6.1 enable ssi  2
8260I> set port 7.1 enable nni  3
8260I> set logical_link 7.1 2 1 network_side 3.1 non_reserved_bandwidth  4
```

**1** This command defines the SSI link between 8260I and 8260G within cluster 3.

**2** This command defines the SSI link between 8260I and 8260H within cluster 3.

**3** This command defines the NNI link between 8260I and 8260F within cluster 2.

**4** This command defines a logical link in a series of logical links, which are used to provide a pipe from cluster 1 to cluster 3 using cluster 2 as the intermediate cluster. The VPI used for this logical link is 2, which must be the same as the VPI specified in steps **4** of 8260C and 8260F, as well as step **5** and **6** of 8260F configuration. On this logical link, the 8260I is configured to assume the role of network side, which complements the role of 8260F as defined in step **6** of the 8260F definitions.

## 4.10.7 Non-Adjacent Clusters Connected by Permanent Virtual Path (PVP)

Two nonadjacent clusters may be connected to each other via an NNI connection over permanent virtual path (PVP). This PVP may be over a public network, as shown in Figure 84 on page 166, or over an intermediate cluster as shown in Figure 85 on page 168.

### 4.10.7.1 NNI Connection Using PVP over a Public ATM Network

When using the PVP over a public network to provide connectivity between two nonadjacent clusters, the following configuration steps must be performed:

1. Connect each boundary ATM switching subsystem in the nonadjacent clusters (8260C and 8260D in Figure 84 on page 166) to the public ATM network.
2. These physical connections must be defined in the public ATM network as UNI interfaces.
3. The public network provider must define a permanent virtual path (PVP) between these ports and provide you with the virtual path identifier (VPI) used by this PVP.

4. You must configure the ports that are providing the connections between the ATM boundary switching nodes and the public ATM network as NNI connections.

   **Note:** This means that the link between the boundary ATM switching subsystem and the public ATM network is seen as an NNI connection by the boundary ATM switching subsystem and as a UNI connection by the public ATM network.

5. You must configure a *logical link* in each boundary ATM switching subsystem using NNI port as described in “SET LOGICAL_LINK Command” on page 140. The VPI used on these logical links must be the same as the VPI provided to you by the public ATM network provider.

The above procedure means that once the boundary ATM switching subsystems are connected to the public ATM network, and the public ATM network provider has defined the PVP, the boundary ATM switching subsystems will behave as if they were two adjacent clusters connected to each other via a physical connection. The configuration steps in the 8260 clusters are identical to the procedure described in 4.10.3.1, “NNI Connection Between Two Adjacent Clusters - Example” on page 141, except that in the SET LOGICAL_LINK command, you must use the PVP identifier given to you by the public ATM provider.

**Note:** In this configuration, the public ATM network provides the path for establishing connections and exchanging data between various stations attached to these clusters, but it will not be part of your ATM network. This means that the stations attached to the public ATM network cannot communicate with the stations attached to these clusters.

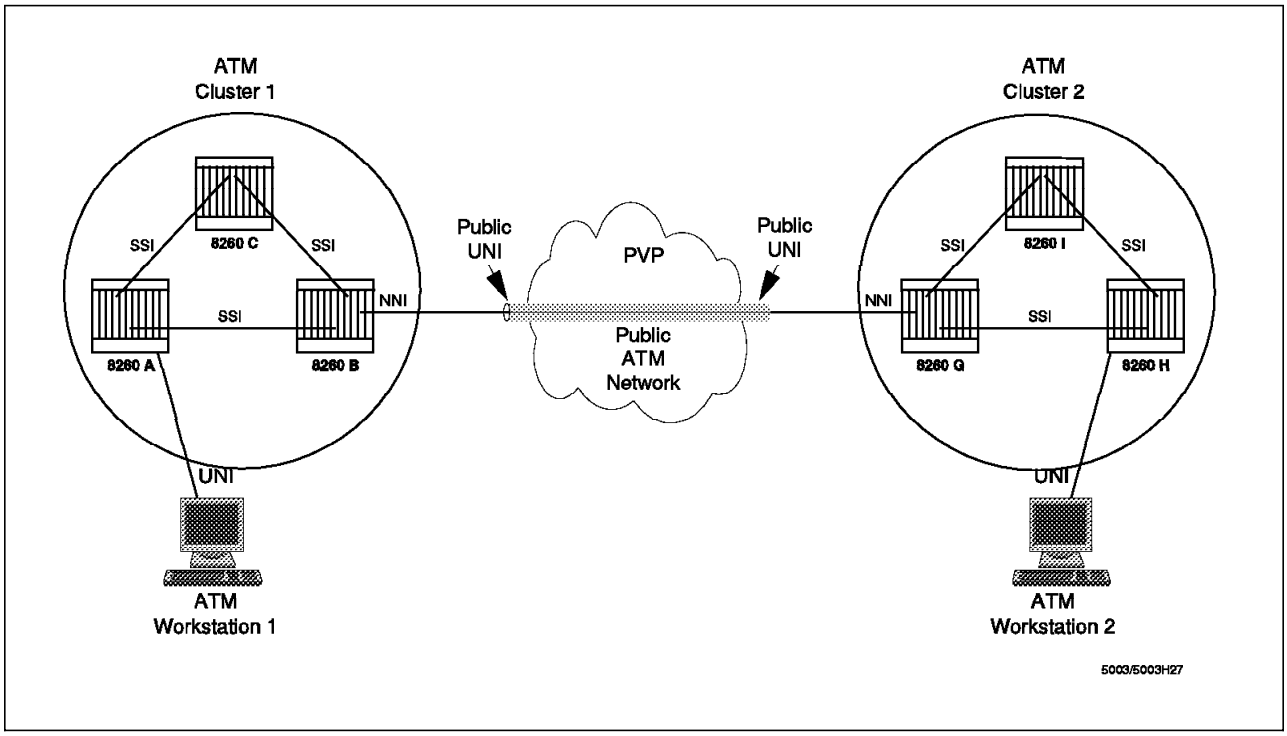



*Figure 84. NNI Connections Using PVP over a Public ATM Network*

### 4.10.7.2 NNI Connection Using PVP over an Intermediate Cluster

When using the PVP over an intermediate cluster, as shown in Figure 85 on page 168, to provide connectivity between two nonadjacent clusters, you must perform the following configuration steps:

1. Connect each boundary ATM switching subsystem in the nonadjacent clusters (8260C and 8260D in Figure 85 on page 168) to the intermediate cluster.
2. These physical connections must be defined in the intermediate cluster as UNI interfaces. This means that the ports connecting 8260D to 8260B and 8260E to 8260G must be defined as UNI ports in the 8260D and 8260E, respectively.
3. In the intermediate cluster, you must define a permanent virtual path (PVP) between the ports connecting the boundary ATM switching subsystems within the intermediate cluster and the boundary ATM switching subsystems within the nonadjacent clusters. In the example shown in Figure 85 on page 168, you must define a PVP between the port connecting 8260D to 8260B and the port connecting 8260E to 8260G.

   To define a PVP between two ports, they must first be configured as UNI ports and then the PVP must be defined. The procedure on how to define a PVP is described in 4.6.7, “How to Define PVPs” on page 94.
4. In the nonadjacent clusters, you must configure the ports that are providing the connections between these clusters and the intermediate cluster as NNI ports.

   **Note:** This means that the link between the boundary ATM switching subsystem and the intermediate cluster is seen as an NNI connection by the boundary ATM switching subsystem and as a UNI connection by the ATM switching subsystem in the intermediate cluster. This type of link is referred to as a *nonsymmetric* link in this book.
5. You must configure a *logical link* in each boundary ATM switching subsystem using the NNI port (8260B and 8260G in Figure 85 on page 168). Note that the VPI used on these logical links must be the same as the VPI used to define the PVP in the intermediate cluster.

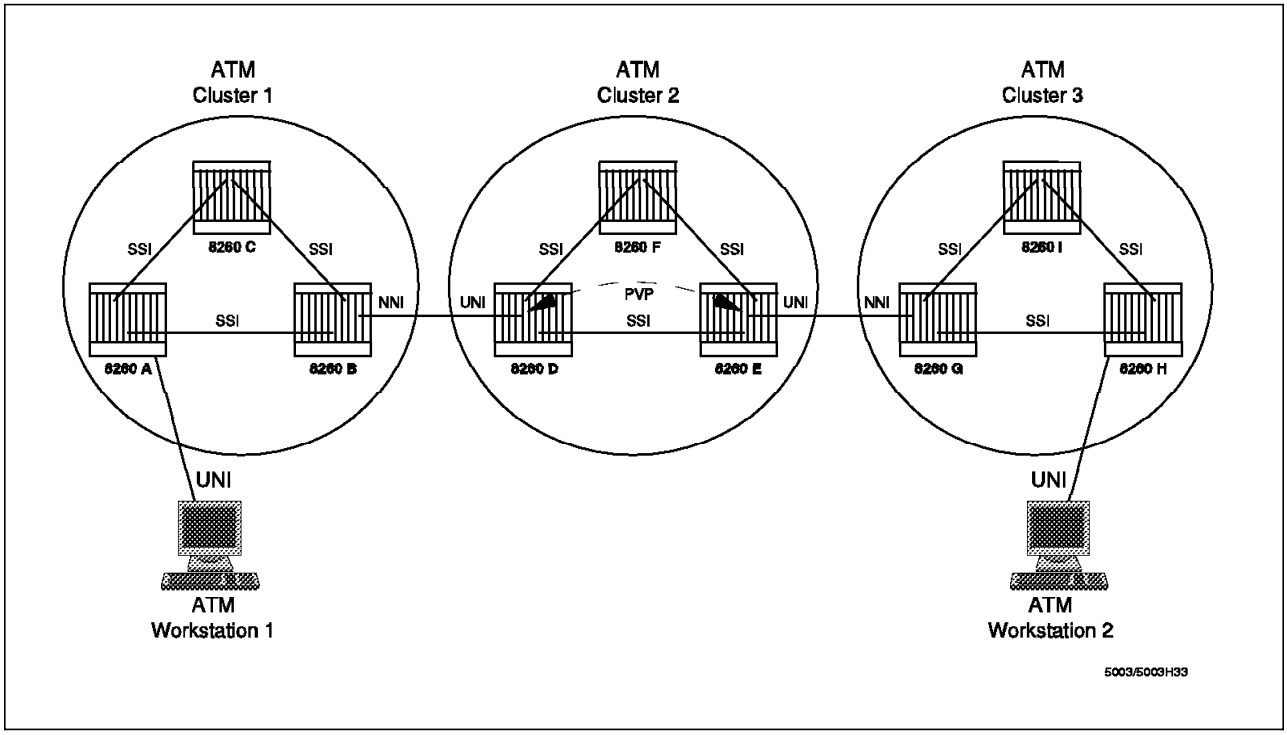



*Figure 85. NNI Connections Using PVP over an Intermediate Cluster*

The procedure above means that once the boundary ATM switching subsystems in the nonadjacent clusters are connected to the intermediate cluster and you have defined a PVP in the intermediate cluster, the boundary ATM switching subsystems in the nonadjacent clusters will behave as if they are two adjacent clusters connected to each other via a physical connection. Therefore, the configuration steps in the nonadjacent clusters are identical to the procedure described in 4.10.3.1, "NNI Connection Between Two Adjacent Clusters - Example" on page 141.

The configuration steps above will provide the path for establishing connections and exchanging data between various stations attached to the nonadjacent clusters, but the stations attached to the intermediate cluster cannot communicate with the stations attached to these clusters. However, you can provide connectivity between the stations attached to the intermediate cluster and the stations attached to the nonadjacent clusters by creating an NNI connection between the nonadjacent clusters and the intermediate cluster. To do so, you must do the following:

1. Provide a physical connection between each nonadjacent cluster and the intermediate cluster.

   **Note:** This connection must be a separate connection from the one used to provide the PVP connection between the nonadjacent clusters. This means that you will have two physical connections between each boundary ATM switching subsystem in the nonadjacent clusters and the intermediate cluster.

2. Use the SET PORT command (explained in "SET PORT Command" on page 139) to define the connection above as NNI ports.

3. Use the SET LOGICAL_LINK command (explained in "SET LOGICAL_LINK Command" on page 140) to define logical links between the nonadjacent clusters and the intermediate cluster.

Figure 86 on page 169 shows an example configuration where the intermediate cluster has an NNI connection to the nonadjacent clusters, as well as providing a PVP between the two nonadjacent clusters.

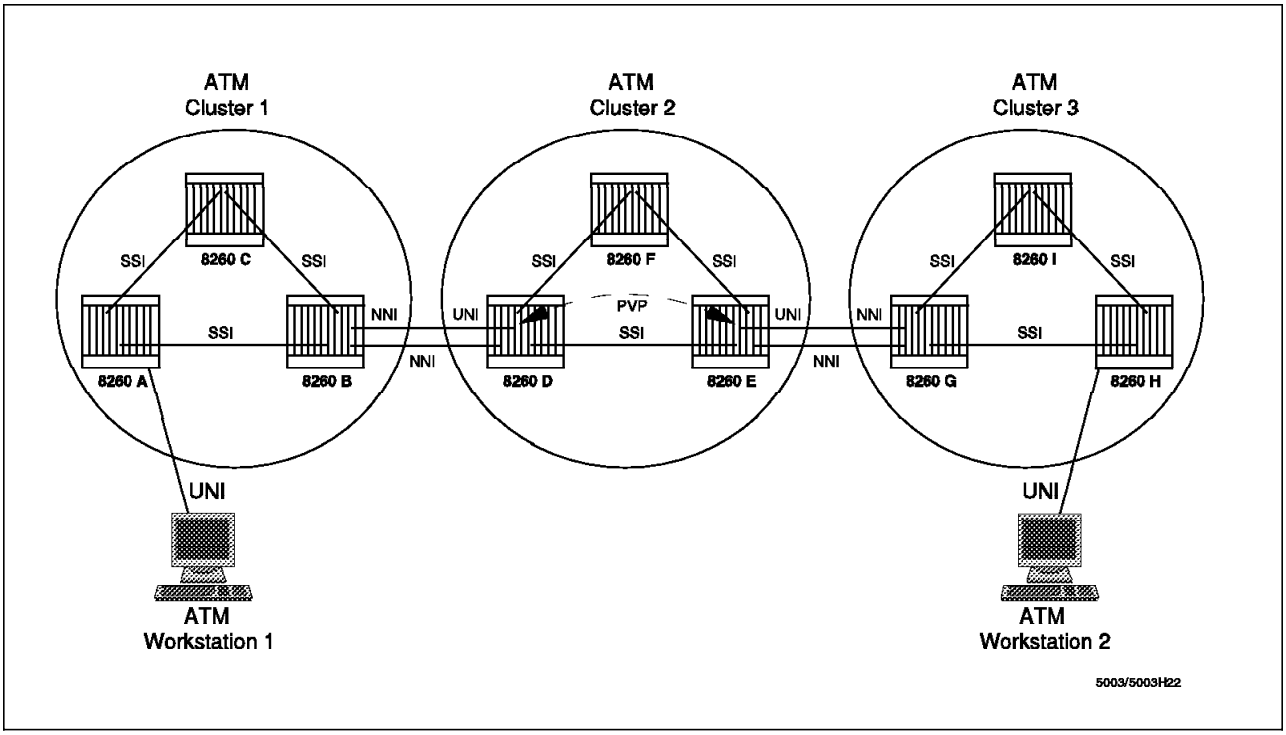



*Figure 86. NNI Connections Using PVP over an Intermediate Cluster*

## 4.11 Design Consideration for Clustering

Grouping ATM switches into separate clusters limits your network restart time, limits the number of topology updates sent (as topology information remains within a cluster), and reduces the resources required on each node to maintain the network topology. ATM clusters can be interconnected in a bigger structure called a subnetwork. Subnetworks can be interconnected to form Campus ATM networks.

An NNI interface is required between any pair of ATM clusters. Clusters can be immediately adjacent, connected via an intermediate cluster, or even connected over a WAN. Within an ATM subnetwork, you may employ a combination of intercluster connections; for example, one pair of clusters may be connected over a WAN, while another pair are connected via direct links. Figure 87 on page 170 shows three interconnected ATM clusters. These are:

- A network-to-network interface between clusters 1 and 2

  For this purpose a link between nodes 8260C and 8260F is used. In both nodes this link is defined as an NNI link.

- A network-to-network interface between clusters 2 and 3

  For this purpose two (parallel) links between nodes 8260F and 8260I are used. On both sides the links are defined as NNI links.

- A network-to-network interface between clusters 1 and 3

Clusters 1 and 3 are nonadjacent. Physical connectivity is obtained via cluster 2, using (nonsymmetrical[1]) links between node 8260B and 8260D and nodes 8260E and 8260G. Within nodes 8260B and 8260G, the link is defined as an NNI link, while the links ending in 8260D and 8260E are defined as a UNI link, with a permanent virtual path (PVP) connecting these links.

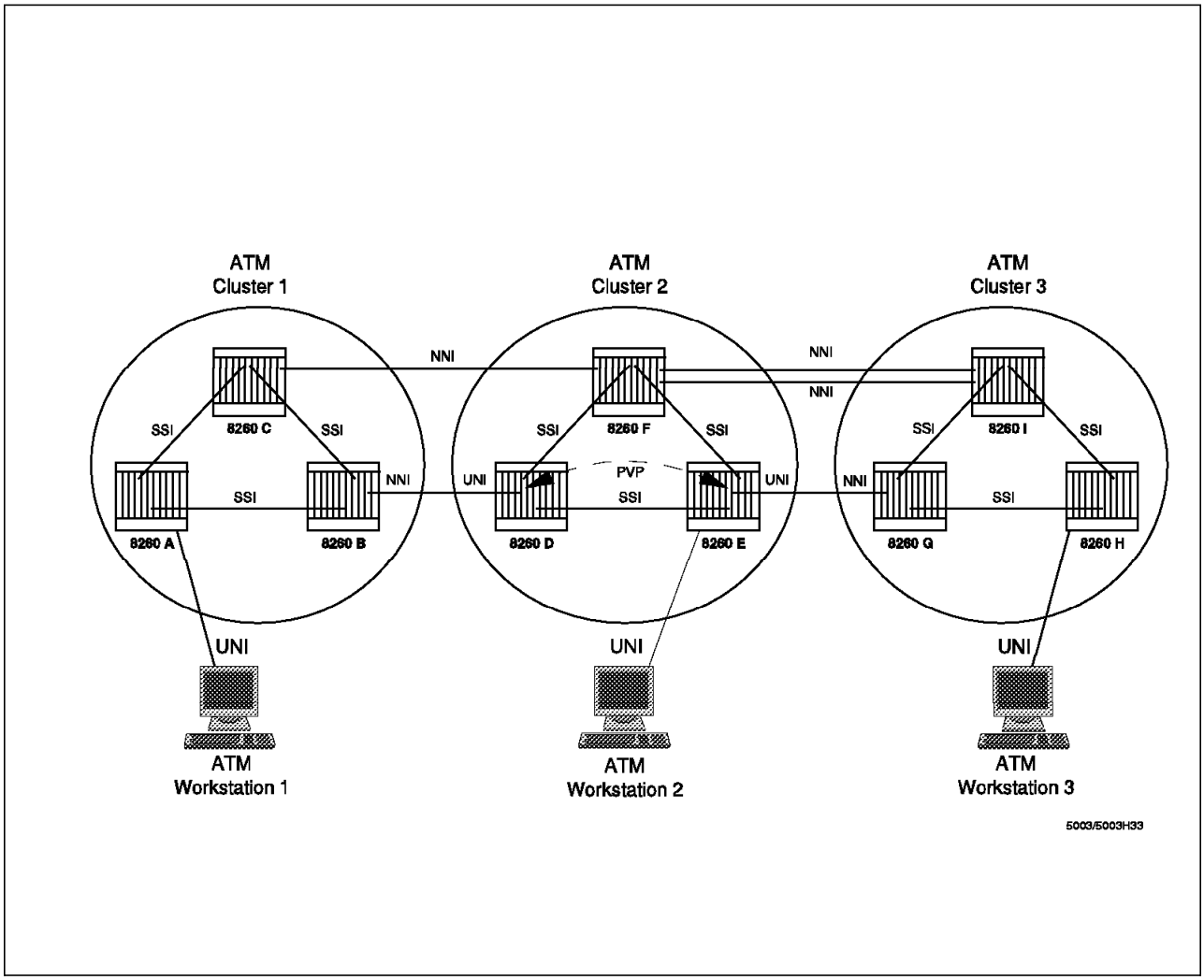



*Figure 87. Cluster Interconnection*

Two clusters can be interconnected via multiple links (see, for example, the parallel links between clusters 2 and 3 within Figure 87). NNI link aggregation (multiple parallel links appearing as a single link for topology and routing) enhances performance and availability and allows for traffic balancing and link redundancy. Physical links should be connected on different modules, providing backup capabilities for links and modules.

The IBM 8260 ATM Switches require that all NNI links connecting two clusters must originate from the same nodes (one within each cluster); these nodes are referred to as *boundary ATM switching subsystems*. To connect to multiple clusters, the boundary ATM switching subsystem function can be distributed on

[1] A *non-symmetrical* link in this context means a link that has been defined as an NNI on one end, and as a UNI on the other end of the link.

multiple nodes, or be concentrated in a single node. In Figure 87, the boundary ATM switching subsystems are distributed between various nodes, so that each boundary node provides the connectivity with one external cluster. For example, in cluster 3, the boundary function to cluster 2 is provided by 8260I, whereas the boundary function to cluster 1 is provided by 8260G. This type of configuration allows you to distribute traffic across various nodes so that if a node or link fails, only part of the traffic will be disrupted. In this example, if 8260I, 8260F or all the links between the 8260I and 8260F fail, only the traffic between cluster 3 and cluster 2 will be disrupted, while the traffic between cluster 3 and cluster 1 will continue as they use a different set of links and nodes to communicate with each other.

However, you may design your network as shown in Figure 88, in which the 8260G provides the boundary function for communication between cluster 3 and both cluster 1 and cluster 2. In this case, if 8260G fails, cluster 3 is isolated from the rest of the network. In this configuration, the failure of the link between 8260G and 8260E only affects the traffic between cluster 3 and cluster 1, whereas the failure of the links between 8260G and 8260F affect the traffic between cluster 3 and cluster 2.

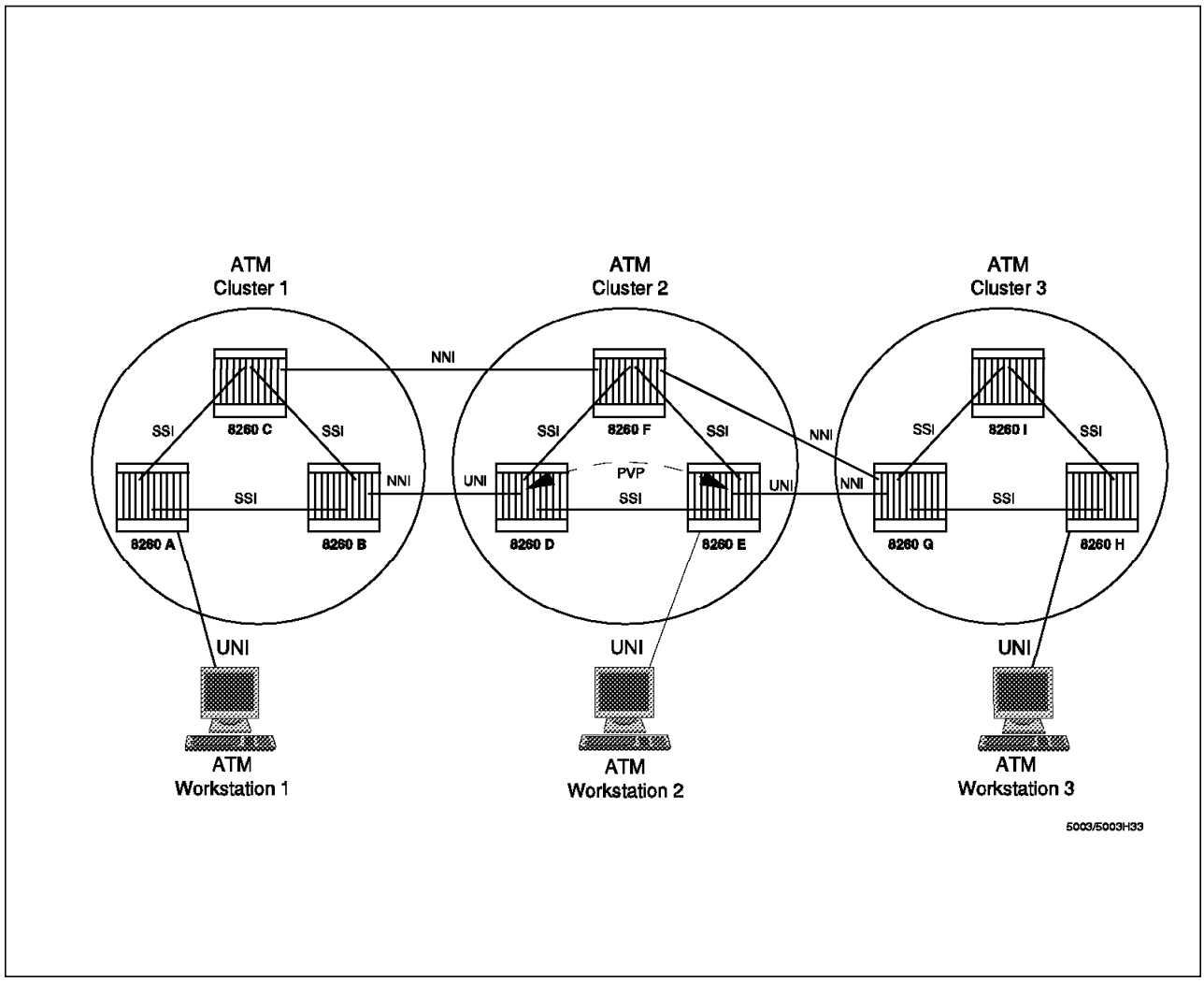



*Figure 88. Cluster Interconnection*

A more restrictive design would be as shown in Figure 89 on page 172, where the same link and boundary is used to provide the boundary function between all the clusters. In this example, the failure of either 8260G or 8260E will result in cluster 3 to loose connectivity with both cluster 1 and cluster 2.

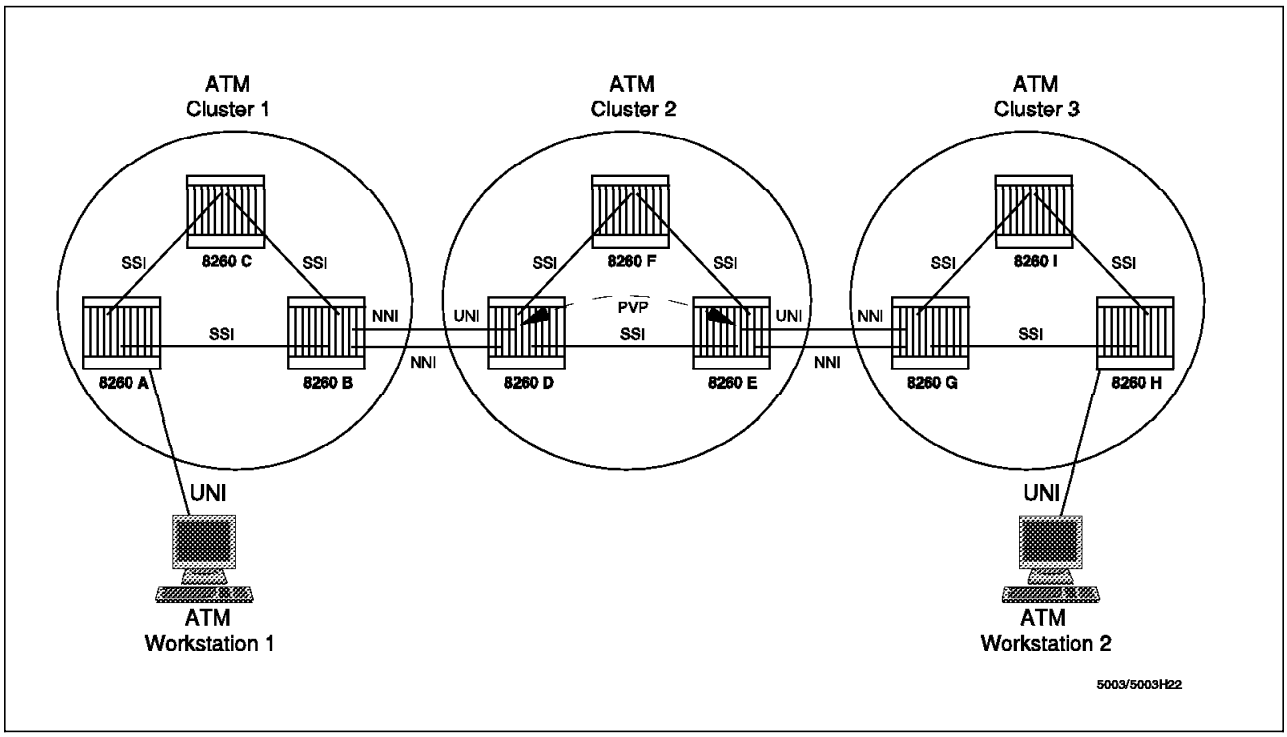



*Figure 89. Cluster Interconnection*

When using an intermediate cluster to connect two nonadjacent clusters, you have the option of using either a logical link or a PVP through the intermediate cluster. When using PVP, the physical link used to provide the connectivity between the nonadjacent clusters cannot be used to provide the connectivity with the intermediate cluster. Therefore, you would require another link for this purpose. For example, in Figure 89, there are two links between 8260G and 8260E. One of those links is used to provide connectivity between cluster 3 and cluster 2, the other link is used to provide the connectivity between cluster 3 and cluster 1 via a PVP. Note that the first link is defined as an NNI link in both 8260G and 8260E, whereas the second link is defined as NNI in 8260G but UNI in 8260E.

When using a logical link to provide NNI connections between nonadjacent clusters, the same link can be used to provide the connectivity between the nonadjacent clusters as well as the connectivity between the nonadjacent clusters and the intermediate cluster. This is shown in Figure 90 on page 173, where a single link between 8260G and 8260E enables cluster 3 to communicate with both cluster 1 and cluster 2.

**Note:** It is not possible to define an NNI interface between two nonadjacent clusters, when the number of intermediate clusters is larger than 1. For complex cases like these, static routes are required.

Using NNI connections (that is, dividing your network into two or more clusters and connecting them via NNI links) when compared to using a single cluster (that is, connecting all the 8260s using SSI links) is more cumbersome and

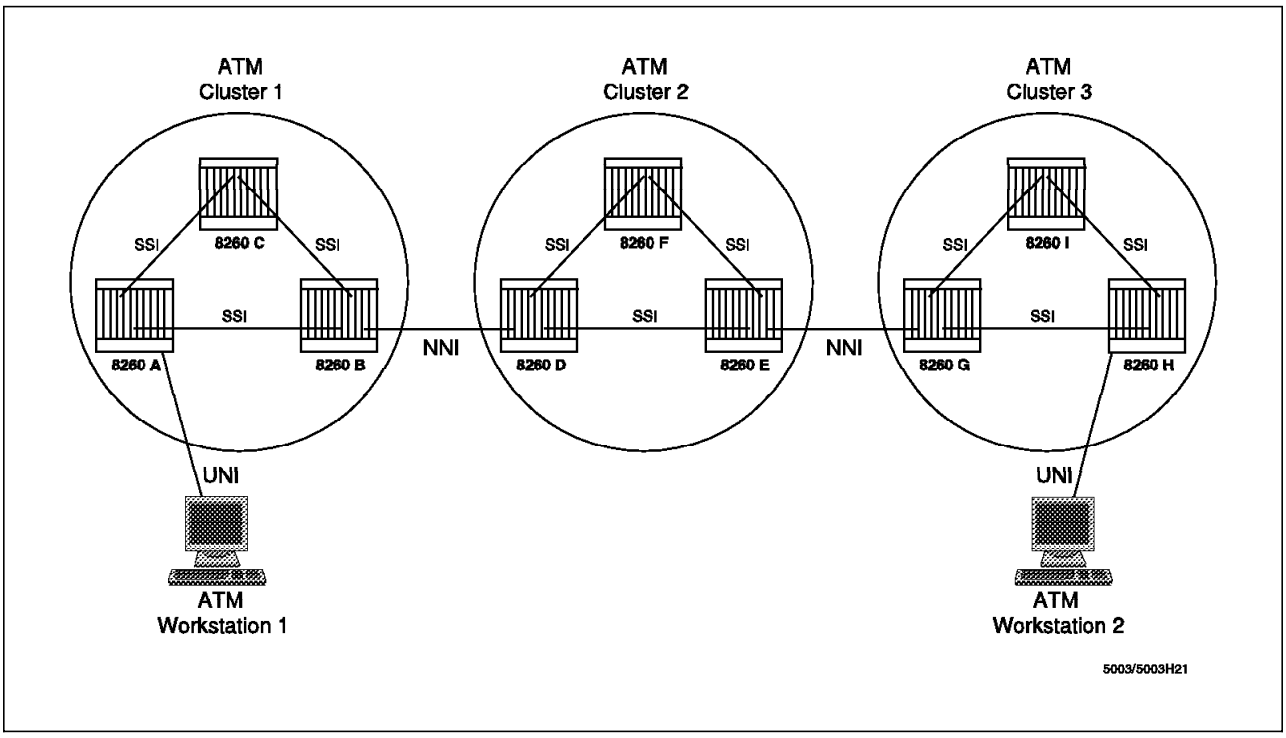



*Figure 90. Cluster Interconnection*

requires additional configuration steps. However, you may have to use NNI connections when:

1. The number of 8260s within your network is more than the recommended 25 8260s within a cluster.
2. Your network consists of 8260 and OEM switches (SSI is an IBM proprietary protocol).
3. The ATM network consists of ATM switches located in different locations that are connected to each other via public ATM networks.

### 4.11.1 ATM Campus Network

An ATM campus network is a group of interconnected IBM 8260s that all share the same 9 bytes for their ATM address. An ATM campus network consists of one or more ATM subnetworks that are connected to each other using NNI (IISP) interfaces. Figure 91 on page 174 shows an example of an ATM campus network consisting of two ATM subnetworks.

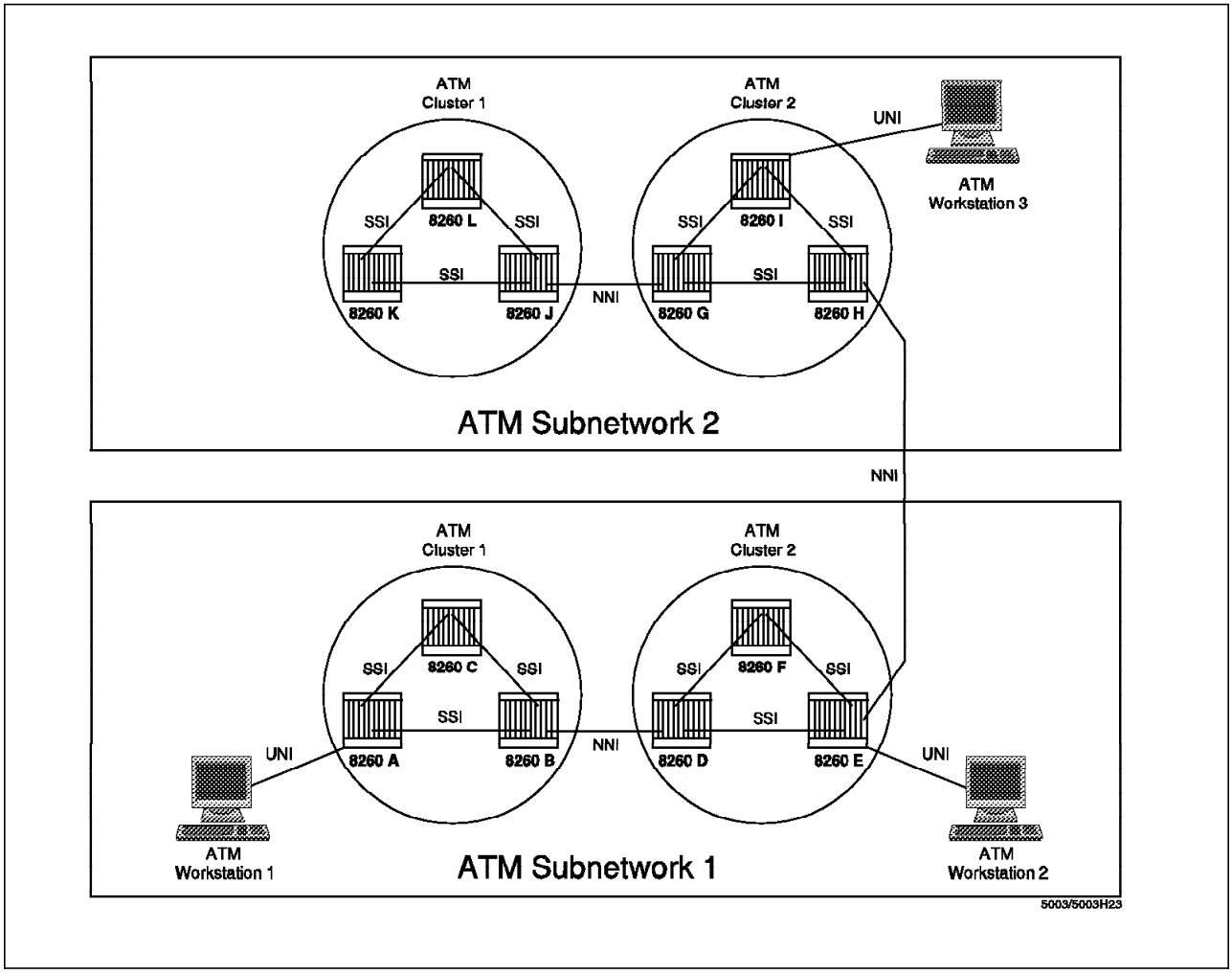



*Figure 91. ATM Campus Network*

When connecting two ATM subnetworks, besides defining a logical link between the boundary hubs within each subnetwork, you must perform the following configuration steps:

1. Configure each ATM subnetwork consisting of SSI and NNI links separately as described in the previous sections.
2. Define an NNI interface for the link between the boundary ATM switching subsystems connecting the two ATM subnetworks together. The command that lets you define the NNI interface is SET PORT and is described in "SET PORT Command" on page 139.
3. Define the logical links for the NNI interface between the boundary ATM switching subsystems connecting the two ATM subnetworks together. The command that lets you define the logical link is SET LOGICAL_LINK, which is described in "SET LOGICAL_LINK Command" on page 140.

   **Note:** The acn parameters specified in the SET LOGICAL_LINK for the NNI link between two ATM subnetworks is a *logical* cluster number (ACN) that must not be used by any of the clusters within that subnetwork. For example, in Figure 91, the logical links defined in 8260E and 8260H to interconnect the two ATM subnetworks may, for example, specify *99* for their acn parameter. Also, note that the logical cluster number in one ATM subnetwork is completely independent from the logical cluster number in the other ATM

subnetwork. For example, in Figure 91, the logical link defined in 8260E may specify *10* as the logical cluster number (as this cluster number is free within ATM subnetwork 1), whereas the logical link defined in 8260H may specify *20* as the logical cluster number (as this cluster number is free within ATM subnetwork 2).

**Note:** The logical cluster number is specified using hexadecimal value.

4. Define one or more static routes to map the ATM network addresses (maximum length of 19 bytes) of the external subnetwork (or the components of the external subnetwork such as clusters, hubs or workstations) to the ATM cluster number (ACN) that can be used to reach that subnetwork (or its components such as clusters, hub or ATM workstations). This gateway cluster is the *logical* cluster specified in the previous step.

   The command that lets you configure these parameters is SET STATIC_ROUTE, which is described in the following section.

***SET STATIC_ROUTE Command:*** This command has the following syntax:

```
8260A> SET STATIC_ROUTE static_route acn
```

**static_route** Varying length (19 bytes) ATM network address that identifies a set of ATM addresses in the destination subnetwork.

**acn** ATM cluster number of a logical hub used in the SET LOGICAL_LINK command to define the NNI interface between the boundary ATM switching subsystems connecting the two ATM subnetworks together. The logical ACN must be unique within its corresponding ATM subnetwork.

**Note:** The ACN value must be specified in hexadecimal.

To provide an example, we assume that in Figure 91 on page 174, the network prefix of the ATM subnetwork 1 is 3909851111111111111111 and that of ATM subnetwork 2 is 3909851111111111119999. We have also assumed that each ATM subnetwork is already configured so that the intrasubnetwork communication is possible. We now want to connect the two ATM networks together so that intersubnetwork communication is possible (that is, any stations attached to these two ATM subnetworks can communicate with each other). To do so, the following configuration steps are required:

1. **8260A:**

```
8260A> set static_route 3909851111111111119999 10
```

2. **8260B:**

```
8260B> set static_route 3909851111111111119999 10
```

3. **8260C:**

```
8260C> set static_route 3909851111111111119999 10
```

4. **8260D:**

```
8260D> set static_route 3909851111111111119999 10
```

5. **8260E:**

```
8260E> set static_route 3909851111111111119999 10
```

6. **8260F:**

```
8260F> set logical link 4.2 5 10 500 network_side
8260F> set static_route 3909851111111111119999 10
```

7. **8260G:**

```
8260G> set logical_link 6.2 5 11 500 network_side
8260G> set static_route 3909851111111111111111 10
```

8. **8260H:**

```
8260H> set static_route 3909851111111111111111 11
```

9. **8260I:**

```
8260I> set static_route 3909851111111111111111 11
```

10. **8260J:**

```
8260J> set static_route 3909851111111111111111 11
```

11. **8260K:**

```
8260E> set static_route 3909851111111111111111 11
```

12. **8260L:**

```
8260F> set logical link 4.2 5 10 500 network_side
8260F> set static_route 3909851111111111119999 11
```

**Note:** In the above example, we have used logical cluster number 10 in subnetwork 1 and logical cluster number 11 in subnetwork 2. However, you must remember that there is no relationship between these two logical cluster numbers, that is, they can be the same or different as long as each one is unique within its own ATM subnetwork.

The above example provides you with information on how to configure static routes to provide any-to-any connectivity between two ATM subnetworks. However, depending of your requirements, you may want to use more restrictive static routes to limit the connectivity between various parts of the network. For example, in the above configuration, if you use the following definition in the 8260A, then the ATM stations attached to the 8260A will only be able to talk to the ATM stations attached to cluster 2 in ATM subnetwork 2.

```
8260A> set static_route 3909851111111111199992 11
```

In general, the longer the ATM address prefix in the SET STATIC_ROUTE, the more restrictive it becomes.

**Note:** In the above example, the static route command defined for the 8260A does not affect the connectivity between 8260A and the other 8260s within ATM subnetwork 1.

## 4.12 Network Management

The A-CPSW supports inband ATM network management using the SNMP agent accessible via the ATM network using Classical IP over ATM (RFC 1577), IP over ATM Forum-compliant LAN emulation and IP over SLIP interface. The SNMP agent of the A-CPSW is a function of the Control Program and implements the following MIBs:

- IETF AToMIB (RFC 1695)

  The AToMIB covers the management of ATM PVC-based interfaces, devices, and services.

- IMLI MIB

  ILMI is defined by the ATM Forum and is based on SNMP. It permits endsystems to register their local address with the 8260 and receive notification of their network address. ILMI is an interim standard that is intended to be used until a standardized ATM layer management interface is defined.

- MIB-II Versions 1.1 and 1.2
- OSPF MIB

  The OSPF MIB is used to manage topology and route computation between a cluster of switches.

- 8260 Private MIB extensions for topology

The IBM MIB extensions cover:

- Hardware specific components, including the switch, modules and ports
- Enhanced PVC management, such as automatic route computation and recovery
- Signalling (Q.2391 and SAAL) configurations and statistics
- ATM statistics

In addition to the inband management, there is an out-of-band service using a command interface, which is available via an ASCII terminal attached either locally or remotely to the RS-232 connector of the A-CPSW module.

## 4.13 Resetting A-CPSW and ATM Media Modules

To reset the ATM subsystem or an ATM media module, use the following commands:

- **Reset ATM_Subsystem**

  This command lets you reset the A-CPSW module and all ATM media modules in the hub. The contents of the main TRS trace file and dumps are not erased. The error log is not erased. An example is as follows:

```
8260A> reset atm_subsystem
You are about to reset the atm subsystem..
Are you sure ? (Y/N)
```

  If you run this command with the FORCE parameter, any unsaved configuration changes made in your current session will be lost. This has the same result as pressing the ATM reset button on the A-CPSW module.

- **Reset Module**

  This command lets you perform a hardware reset of an ATM media module. Entering this command has the same effect as pressing the ATM reset button on the ATM media module.

  You can specify any slot in which a media module is installed (1 to 17, except 9, 10 and 11). To reset the A-CPSW you must use the RESET ATM_SUBSYSTEM command. An example is as follows:

```
8260A> reset module 4
Reset started
```

## 4.14 Reverting Configuration Changes

To restore the configuration parameters that were active at the time of the last SAVE, you must use the REVERT command. Any changes made using the SET command since the last save are lost. The REVERT command has the following options (the same as the SAVE command):

- Alert

- All
- Community
- Device
- Module_port
- Static_route
- Terminal
- TFTP

**Note**

A reset is performed when you revert the configuration settings for the device, module_port and all options.

For example, you may want to change to revert all the changes made during your current session, as follows:

```
8260A> revert all
Reverting device or module_port will reset the ATM subsystem.
Are you sure ? (Y/N)
```

## 4.15 Upgrading Microcode

There are three code components in the 8260 ATM subsystem that can be upgraded:

- **Boot Code**

  This resides in flash memory on the A-CPSW and is the first thing that executes after a power-on or reset. It contains initialization, diagnostics, download out-of-band, and auxiliary commands. This code executes straight from flash memory and is normally used to load the operational code.

- **Operational Code**

  This code is also on the A-CPSW and is executed once the boot code has finished. There are two copies of the code stored in flash memory. One of these copies is identified as current and is loaded into RAM during the initialization process. This code is executed from RAM. The second copy of the operational code allows the loading of the new operational into the A-CPSW module code while the module is operational.

  The second copy of the operational code allows you to load a new operational code while the A-CPSW is running.

- **FPGA Code**

  This code configures the various Xylinx chips on the A-CPSW and ATM media modules so that they perform their desired ATM functions. There are two copies of the code stored in flash memory. One of these copies is identified as current code. The current code is loaded into the Xylinx chips of the module during initialization. The second copy of the FPGA code

allows the loading of the new FPGA code while the module is operational. This is described below.

Note that the A-CPSW also has Xylinx chips and may also require an FPGA code upgrade. The FPGA code on the A-CPSW is not the same code as the boot or operational code that also resides on the A-CPSW.

The reason that the FPGA and the A-CPSW module's operational codes are structured to have a second copy is to be able to update these codes with as little disruption to the network as possible and to have minimal impact should the upgrade process not complete properly.

When the FPGA code for the modules needs to be updated, the new code actually gets stored in the noncurrent area of flash memory. After the new code is stored successfully, the user will be able to make the new code the operational one via a command from the command line interface.

**Note:** This is also the case for the operational code on the A-CPSW, but the boot code only has one area of flash memory where the code can reside, so the code is downloaded to RAM.

The result of this implementation is that the time consuming process of downloading new code can be carried out while the network is still operational. The time when the network will be disrupted is when the pointers have to be changed to point to the recently downloaded code. This also means that disruption due to a failed download will be minimal because you will still be running from the code until the pointers get changed.

The only time there might be an exposure due to a failed download of code is when the boot code gets updated. With this portion of code there is only one area in which it is stored where there are no pointers associated with the boot code. When an administrator initiates the process to use the new boot code, the code that has recently been downloaded into RAM gets copied into flash memory. Since the code is already in RAM, the copy process is very quick, and therefore, any risk of a failed download of the boot code is greatly minimized.

Although the 8260 can support out-of-band download operation, it is inherently unreliable; hence, it's recommended that inband operation should only be used in code upgrades. This means that before any upgrade occurs you should insure that your ATM network is configured for Classical IP (RFC 1577) or LAN emulation.

When performing an upgrade, explicit instructions will be provided with the upgrade kit. These instructions should be followed exactly to ensure a successful upgrade. This section describes the general process involved in code upgrade so that you may understand why particular processes take place. However, it is not intended to replace the installation instructions.

Since upgrade by the inband method is recommended, you must ensure that the Classical IP or LAN emulation environment is configured properly and operational. Please refer to 4.3.2, "How to Configure Classical IP over ATM" on page 82 and 4.18.1, "IP Support for the A-CPSW Agent Using LAN Emulation" on page 187 for details.

The next step is to prepare the TFTP server with the code. There will be diskettes provided in the upgrade kit that contain the code, along with

instructions on how to extract the files. Please note that there are different files for each of the different components that need to be upgraded.

The instructions will normally mention that the download must be performed from an AIX TFTP server. Strictly speaking this not true. The diskettes run an AIX script file that copies the files to the appropriate directory, but these files could just as easily be put on a non-AIX machine manually. As long as the TFTP server can be reached by the A-CPSW (perhaps via a router or bridge if the TFTP server is not directly attached to ATM) there should be no problems in using a non-AIX machine.

The actual upgrade process is described in the following sections.

### 4.15.1 A-CPSW Operational and Boot Code Upgrade

To upgrade the operational and boot code for the A-CPSW, you must perform the following steps:

1. Load the boot code.

   Although there is a single area for the A-CPSW′s boot code, you can download a new copy of the boot code while A-CPSW is operational. This is due to the fact that the A-CPSW module′s boot code will only be used during the initialization of the A-CPSW (after power-on or reset) and is not used during the normal operation.

2. Load the operational code.

   When you download the new operation code for the A-CPSW, it will be loaded into the noncurrent areas of memory in the A-CPSW module. This means that this stage can be performed while the ATM network is still operational; it also means that after the load operation is performed, you are still using the older version of the operational code.

3. Swap the operational code so that the new code becomes the active code.

   When you do this step, the noncurrent copy of the operational code becomes the active copy. This step will result in the hub to be reset, which means that all the connections using the 8260 ATM subsystems that are being updated will be disrupted. However, you will be provided with a warning and will be asked to verify that you want to go ahead with the swap operation.

The mechanics of the download process are quite simple. The first stage is to set the TFTP server IP address, then set the file type (operational or boot), then the full path to the code file, then perform download and swap operations. In all cases, the SWAP command must be the last command to be performed after you have loaded all upgrades.

The SWAP command must first address the media modules.

The following is an example of the scenario to perform the above steps for an A-CPSW module:

```
8260A> set tftp server_ip_address 9.24.104.204
TFTP server set.
8260A> set tftp file_type boot
File type set
8260A> set tftp file_name
Enter file name:
d:\temp\A_CPSW.bt
File name set
8260A> download inband
8260A> set tftp server_ip_address 9.24.104.204
TFTP server set.
8260A> set tftp file_type operational
File type set
8260A> set tftp file_name
Enter file name:
d:\temp\A_CPSW.op
File name set
8260A> download inband
8260A> swap microcode
You are about to change operational microcode version and reset the hub.
The saved hub configuration may be lost..
Are you sure ? (Y/N) Y
```

### 4.15.2 FPGA Code Upgrade

To upgrade the FPGA code for the A-CPSW or ATM media modules, you must perform the following:

1. Load the FPGA code.

   When you download the new FPGA code for a module, it will be loaded into the noncurrent areas of memory in that module which is reserved for FPGA. This means that this stage can be performed while the ATM network is still operational and also means that after the load operation is performed, you are still using the older version of the FPGA.

2. Swap the FPGA code so that the new code becomes the active code.

   When you do this step, the noncurrent copy of the FPGA code becomes the active copy. In the case of A-CPSW, this step will result in the hub to be reset, which means that all the connections using the 8260 ATM subsystems that are being updated will be disrupted. In the case of ATM media modules, only the connections using that module will be disrupted. However, for both types of modules, you will be provided with a warning and you will be asked to verify that you want to go ahead with the swap operation.

The mechanics of the download process are quite simple. The first stage is to set the TFTP server IP address, then set the file type to FPGA, then the full path to the code file. Set the TFTP target module and then perform the download followed by the swap operation.

Below is an example of downloading the FPGA code for an ATMflex module.

```
8260A> set tftp server_ip_address 9.24.104.204
TFTP server set.
8260A> set tftp file_type fpga
File type set
8260A> set tftp file_name
Enter file name:
d:\temp\afmflex.fp
File name set
8260A> set tftp target_module 15
Target module set
8260A> download inband
You are about to download a new version..
Are you sure ? (Y/N) Y
8260A> swap fpga_picocode 15
You are about to change operational FPGA version..
Are you sure ? (Y/N) Y
Processing slot 15 ...
```

**Important**

Remember the code for each different type of module is different; that is, the code for the FPGA on a 100-Mbps MIC module is different from the FPGA for a 100-Mbps SC module, which is different than the FPGA for the A-CPSW and so on.

After each download you should get a confirmation message saying that the download was successful. If the download failed, try again; otherwise, go through the troubleshooting guide in the A-CPSW installation manual.

## 4.16 Trace and Dump Facility

The A-CPSW has a number of facilities to assist in problem determination:

- **Main Trace**

  Records a general trace and signalling

- **TRS Trace**

  Records the topology updates involved with the TRS subsystem

- **TRS Dump**

  Records the contents of the TRS subsystem at a particular point in time

- **Error Log**

  Records any errors that the A-CPSW encounters

To activate one of the tracing mechanisms, you have to turn it on using the set trace command. An example of turning on the main trace is as follows:

```
8260A> set trace main_trace on
```

Once you have finished tracing, you must then turn off the tracing function using the set trace command again:

```
8260A> set trace main_trace off
```

The A-CPSW has now stored the trace information, which must be uploaded to a TFTP server for viewing. To do this, you must define the TFTP server, indicate that you want to upload a trace file, give the full file name of the file you wish the trace to be stored as and initiate the upload.

The following screen shows an example of the above process for uploading a main trace file:

```
8260A> set tftp server_ip_address 9.24.104.252
TFTP server set.
8260A> set tftp file_type main_trace
File type set
8260A> set tftp file_name
Enter file name:
d:\temp\trace.mai
File name set
8260A> upload inband
Upload successful.
8260A>
```

To get a TRS dump, you must first make the A-CPSW dump the TRS contents by issuing the command:

```
8260A> dump trs
```

You then upload the dump file by using the same process as uploading a trace file but you must specify a TFTP file type of trs_dump:

```
8260A> set tftp file_type trs_dump
```

To upload the error log, you use the same process described above except you set the TFTP file type to error_log with the following command:

```
8260A> set tftp file_type error_log
```

## 4.17 Upload/Download of A-CPSW Configuration

After changing the configuration of the ATM components of the 8260, you may save the configuration on the NVRAM of the A-CPSW module by using the SAVE command. This configuration information saved on the NVRAM will be used to configure the ATM components of the 8260 during a power-on or reset of the A-CPSW module.

You may want to store the configuration of the ATM components of the 8260 in a file on a TFTP server using the UPLOAD command. This file, known as the *configuration file*, contains the following information:

- Alert settings
- Community name tables
- Device configuration
- Module and port configuration
- Logical link tables
- Static route tables
- Terminal configuration
  - CONSOLE_PORT_PROTOCOL will be set to NORMAL.
- TFTP settings
- ESI settings
- PVC definitions
- LAN emulation configuration server address table

The configuration file cannot be interpreted and must not be modified. However, you may use this file to restore the configuration of the A-CPSW. To do so, you must use the DOWNLOAD command. However, note that after a successful download of the configuration file, the A-CPSW will be reset.

### 4.17.1 LAN Emulation Configuration Server Address Advertisement

This function of the A-CPSW module provides the LAN emulation clients the means to obtain the ATM address of the LAN emulation configuration server (LECS). Once they know the address of the LECS, the LAN emulation clients will communicate with the LECS to obtain the address of the LAN emulation server (LES).

LAN emulation clients may obtain the address of LECS using one of the following procedures:

- During ILMI the LAN emulation client gets the unicast ATM address of LECS that is available in the ILMI MIB (atmSrvcRegATMAdress). The A-CPSW module supports the ILMI MIB. Therefore, you can define the LECS ATM address that is to be returned by the A-CPSW to the LE client during the ILMI exchange. To do so, you must use the following command:

```
8260A> set lan_emul configuration_server inactive_wka
Enter ATM address : 39.09.85.aa.aa.aa.aa.aa.aa.aa.aa.01.01.40.00.00.00.00.aa.00
Entry set.
8260A>
```

  **Note:** The LECS ATM address must be defined in all the A-CPSW modules that are going to support LAN emulation clients requesting the LECS ATM address during the ILMI process.

- The LAN emulation clients can connect to the LECS, using the LAN Emulation Forum′s well-known address for the LECS. This well-known address is:

```
47.00.79.00.00.00.00.00.00.00.00.00.00.A0.3E.00.00.01.00
```

You can define an ATM address to be substituted for this well-known address. You can do this using the following command:

```
8260A> set lan_emul configuration_server active_wka
Enter ATM address : 39.09.85.aa.aa.aa.aa.aa.aa.aa.aa.01.01.40.00.00.00.00.bb.00
Entry set.
8260A>
```

**Note:** The LECS ATM address to be used instead of the well-known address must be defined in all the A-CPSW modules that are going to support LAN emulation clients requesting the LECS ATM address using the well-known address.

- The LAN emulation clients connect to the LECS using VPI=0 and VCI=17.

You can display the address of the LECS specified in the A-CPSW module, using the following example:

```
8260A> show lan_emul configuration_server
Index            ATM address
--------------------------------------------------------------------------------
  1            39.09.85.AA.AA.AA.AA.AA.AA.AA.AA.01.01.40.00.00.00.00.AA.00
  2 WKA active 39.09.85.AA.AA.AA.AA.AA.AA.AA.AA.01.01.40.00.00.00.00.BB.00
8260A>
```

## 4.18 A-CPSW Microcode V2.1.0

A-CPSW microcode V2.1.0 provides you with the following enhancement:

- IP support for the A-CPSW agent using ATM Forum-compliant LAN emulation client
- Increased number of supported ATM connections
- Redundant A-CPSW module support
- 8260 chassis monitoring using a subset of DMM command

The amount of memory required to run A-CPSW microcode V2.0.4 (and earlier versions) require 8 MB of memory. However, the amount of memory required to run A-CPSW microcode V2.1.0 has been increased to 16 MB. Therefore, the previous A-CPSW modules with 8 MB of memory are withdrawn and replaced by a new A-CPSW module that can be ordered using FC 5100. The memory capacity on the new A-CPSW module has been extended from 8 to 16 MB of DRAM compared to the previous module. Customers who already have an A-CPSW module and wish to benefit from the new functionality offered by V2.1.0 and the future microcodes must increase the DRAM capacity of their module to 16 MB via an MES upgrade. This MES is customer-installable.

However, the switch redundancy and the chassis monitoring functions supported by microcode level V2.1.0 run only on the new A-CPSW module (feature number 5100), which, in addition to 16 MB of memory, includes other hardware enhancements that are not available in the earlier A-CPSW modules. Therefore, customers who upgrade their existing A-CPSW module to 16 MB of memory will not be able to take advantage of the switch redundancy and the chassis monitoring. To do so, they must replace their A-CPSW modules with the new A-CPSW modules.

### 4.18.1 IP Support for the A-CPSW Agent Using LAN Emulation

Previous releases of the A-CPSW microcode allowed the inband management of the 8260 ATM switching subsystem using Classical IP over ATM. This support has been extended into the LAN emulation environment. The LAN emulation support provided by the A-CPSW V2.1.0 is ATM Forum-compliant and allows you to manage the 8260 from either an ATM station with LAN emulation or a LAN station through an ATM Forum-compliant ATM-LAN bridge.

**Note:** This release of the A-CPSW supports Ethernet 802.3 or DIX V2 LAN emulation only. There is not support for token-ring LAN emulation. Also, note that the 802.3 or DIX support are mutually exclusive.

### 4.18.2 How to Configure LAN Emulation Client

To configure the LAN emulation client for the A-CPSW module, you must use the SET DEVICE LAN_EMULATION_CLIENT command. An example of this command is as follows:

```
8260A> set device lan_emulation_client eth eth_type dIX ip_address 192.168.20.7
les_atm_address 39.09.85.11.11.11.11.11.11.11.11.01.01.70.00.80.00.90.00.00
mac_address 4000008260A1 subnet_mask ff.ff.ff.00
```

Note that this example configures a LAN emulation client, emulating an Ethernet DIX V2 device with the endsystem identifier (ESI) of 4000008260A1 and using the services of the LAN emulation server with the ATM address of 39.09.85.11.11.11.11.11.11.11.11.01.01.70.00.80.00.90.00.00. This LAN emulation client will use the IP address of 192.168.20.7 and subnetwork mask of 255.255.255.0.

### 4.18.3 Increased Number of Supported ATM Connections

The maximum number of ATM connections that are supported by the A-CPSW module are based on the connection control blocks that the module can manage, and the type of connections (point-to-point or point-to-multipoint). For point-to-multipoint connections, the number of parties per connection play an important role on the number of supported connections.

The new A-CPSW module (FC5100) and the older A-CPSW modules (FC5000) upgraded to 16 MB of memory, combined with the microcode V2.1.0, provide you with a higher number of supported ATM connections as follows:

- The switch supports 12,128 connection control blocks.
- Two control blocks are required per point-to-point connection.
- The maximum number of supported point-to-point connections is 6000.
- The switch has 6000 control blocks for point-to-multipoint connections.
- Two party control blocks are required per party.
- The maximum number of parties over a point-to-multipoint connection is 3000.
- The maximum number of PVCs supported per switch is 100.
- The maximum number of connections supported per port is 992.
- The maximum number of connections supported per module is 992.

### 4.18.4 Redundant A-CPSW Module Support

A-CPSW module FC 5100 offers redundant mode capability. With two A-CPSWs installed in the 8260 (positions 9/10 and 11/12), the control point and switch function is duplicated. It will protect the network against ATM switch or control point failures.

#### 4.18.4.1 Election

At power-on time, a redundant ATM subsystem elects its active A-CPSW by priority on the module installed in slots 9/10. When one A-CPSW is elected active, the other one operates in standby mode.

When the subsystem is operational, the A-CPSW election can be forced using the following command:

```
SET DEVICE ROLE primary/secondary
```

***Switchover Functionality:*** If one A-CPSW is forced active, the other is set automatically to backup mode.

In case of failure detected by the active A-CPSW, the standby A-CPSW is automatically elected as the new active module.

Every hour, the active A-CPSW runs diagnostics on the standby unless it is in maintenance (to allow out-of-band download). If diagnostics fail after two retries, the error is reported by the active A-CPSW and can generate an SNMP trap.

Every second, the active A-CPSW polls the backup A-CPSW. If there is a serial link failure between the A-CPSWs, a lack of polling will be detected by the Standby A-CPSW. A switchover loop request will be initiated by the standby A-CPSW if 20 consecutive pollings are not received. The process will stop after three loops and the active CPSW will be stopped and set in error-maintenance mode with prompt>>0040>>. In the error log, an error 540 will say:

```
Short reset loop detected between active and standby CPSW,
check SPI link and ATM backplane.
```

Depending on the activity done by the active module, the active A-CPSW will decide according to the events if the switchover must occur earlier, either after one backup request or two backup requests. After three A-CPSW backup requests, the active A-CPSW is forced in standby mode and the backup A-CPSW becomes active. All communications going through the switch are lost during the switching period.

The last configuration saved on the failing A-CPSW using the command:

```
SAVE CONFIGURATION
```

is transferred at polling time when a configuration change has been detected.

The network saved configuration mirrored in the backup A-CPSW during the polling will be used to reconfigure the hub network TAT switching time frame.

Switching request cases detected by the active module include:

- Control point failure detected by the boot program and by memory diagnostics
- FPGA failure detected by the boot program

- Prizma switch failure detected by the operational code
- SWITCH CAP CAD failure detected by the operational code
- Backplane failure detected by the operational code or wrap test (no failure can be detected on the backplane backup lines that are not activated)

#### 4.18.4.2 FC5000 Limitation

The A-CPSW feature 5000 cannot be granted with the redundancy code that is only dedicated to feature 5100.

If the FC5000 with 16 Mbps is upgraded by error, you will get very limited redundancy support, as in the following:

- An active FC5000 works but does not start mirroring, which will lead to potential problems with the network configuration in case of redundancy (mirroring option not available on FC5000).
- A standby FC5000 will be properly seen and diagnosed by the active FC5000, but could not be mirrored.
- A standby FC5000 just returns to maintenance-standby. It is not seen by the active and will take over only on clear-cut failures of the active.

# Chapter 5. Configuring 8260 ATM Media Modules

The following ATM modules are provided for the 8260:

- ATM 4-Port 100-Mbps (A4-FB100) Module
- ATM 155-Mbps Flexible Concentration (ATMflex) Module
- 8260 ATM TR/Ethernet LAN Bridge (8281) Module
- ATM 12-Port 25-Mbps Concentration Module

This chapter provides you with the information on how to configure these modules.

## 5.1 Configuring A4-FB100 Module

To configure the A4-FB100 module, you must do the following:

1. Install the module in the 8260.

   This module may be installed in slots 1-8 and 11-17 of the 8260 Model A17 and slot 1-8 of the 8260 Model A10.

   When you install an A4-FB100 module in a slot that was not being used previously by another A4-FB100 module (that is, there is no saved configuration for an A4-FB100 module in the A-CPSW module for that slot), the module is, by default, set to isolated mode.

   If you display the status of such a module (using the SHOW MODULE command), you will see information similar to the following:

```
8260A> show module 13 verbose

Slot Install Connect Operation General Information
-------------------------------------------------------------------------
 13     Y       n         n     8260 ATM 100-Mbps Module


status: not connected / hardware okay
        enable  / Normal

P/N:58G9611 EC level:C38844 Manufacture:VIME
8260A>
```

   If the module is installed in a slot previously used by another A4-FB100 module, the new module is automatically configured with the parameters that were last saved in the A-CPSW module for that slot.

   When an A4-FB100 module is isolated, no network activity takes place on it and it cannot be used to access the ATM network.

2. Connect the module to the ATM backplane.

   To be able to use the A4-FB100 module, you must first connect the module to the backplane using the SET MODULE command. An example for an A4-FB100 module in slot 7 is as follows:

```
8260> set module 7 connected
```

© Copyright IBM Corp. 1996

When you use the above command, the module will be connected to the backplane and all the ports will be configured as UNI ports and will be *disabled* by default.

You can establish the A4-FB100 module's connection to the backplane and *enable* all of its ports using the following example:

```
8260A> set module 7 connected enable
```

The above command will result in all the ports to be configured as UNI ports, with the following settings:

- Flow_control = disabled
- ILMI = forced_sig_3_0

3. Enable the A4-FB100 ports and configure its interfaces.

   After you connect an A4-FB100 module to the ATM backplane, you must enable the ports you want to use and set the type of ATM interface used on each port.

   You can configure the following interfaces:

   **User-to-Network (UNI)** — Defines the interface between an ATM user device (such as a workstation, server, ATM-to-LAN bridge, etc.) and the ATM network.

   **Switch-to-Switch (SSI)** — Defines the interface between a pair of ATM hubs in the same ATM cluster.

   **Network-to-Network (NNI)** — Defines the interface between a pair of ATM clusters.

   The default type of interface in the module is UNI. To enable individual A4-FB100 ports and set the type of ATM interface to be used (UNI, SSI or NNI), you must use the SET PORT command. The following example set port 3 of the A4-FB100 in slot 4 with an NNI interface:

   ```
   8260A> set port 4.3 enable nni
   ```

   On a UNI port, the endsystem attached to the 8260 ATM media module may or may not support ILMI. Also, it supports UNI 3.0, 3.1 or UNI version identification via ILMI. Based on the capability of the endsystem, you may configure the UNI to use the following parameters:

   **flow_control_enabled** — Activates flow control (XON/XOFF) on this port.

   **flow_control_disabled** — Deactivates flow control (XON/XOFF) on this port.

   **ilmi_forced_sig_3_0** — Address registration using ILMI is activated and UNI 3.0 signalling is forced. This setting should be used when the attached station supports ILMI and uses UNI 3.0 signalling.

   **ilmi_forced_sig_3_1** — Address registration using ILMI is activated and UNI 3.1 signalling is forced. This setting should be used when the attached

station supports ILMI and uses UNI 3.1 signalling.

**no_ilmi_sig_3_0** — Address registration using ILMI is deactivated and UNI 3.0 signalling is forced. This setting should be used when the attached station does not support ILMI and uses UNI 3.0 signalling.

**no_ilmi_sig_3_1** — Address registration using ILMI is deactivated and UNI 3.1 signalling is forced. This setting should be used when the attached station does not support ILMI and uses UNI 3.1 signalling.

**normal_ilmi** — Address registration using ILMI is activated and the signalling protocol version (UNI 3.0 or 3.1) will be detected automatically using the ILMI procedure. This setting must be used when the endsystem connected to the 8260 port supports UNI version identification via ILMI.

When the attached station does not support ILMI, you must define the endsystem identifier (ESI) of the station to the 8260 using the SET ATM_ESI command. In the following example, 40.00.00.00.00.01 specifies the ESI field (bytes 13 through 19) of the ATM workstation attached to port 7.1.

```
8260A> SET ATM_ESI 7.1 40.00.00.00.00.01
```

For more information about the above parameters, please refer to 4.7, “ATM Signalling” on page 97. Also, you may display the registered addresses (dynamic or static) in your 8260 using the SHOW ATM_ESI command as described in 4.7.3, “Displaying Registered ATM Addresses” on page 100.

4. Configure the logical links (NNI connections only).

   If you are configuring an A4-FB100 port for an NNI interface to connect two clusters or subnetworks, you must also configure a logical link between the two boundary hubs of each side of the connection.

   The command that defines the logical link is:

```
8260A> SET LOGICAL_LINK slot.port vpi acn role uni_version traffic_type bandwidth
```

   **slot** — Slot number of ATM media module (1 to 17, except 9 to 11).

   **port** — Port number of the ATM port.

   **vpi** — Virtual path identifier used to identify the logical link (0-15). You must assign the same VPI to the ports at each side of the logical link. If you configure more than one logical link for a port, you must assign a different VPI for each link.

   **acn** — When connecting two ATM clusters in the same subnetwork, this is the ACN of the remote boundary hub. When connecting two ATM subnetworks, this is a

logical cluster number which must be unique within the ATM subnetwork in which the logical link is defined.

**role** This parameter defines the Q.2931 role. Network_side means that the 8260 hub assigns ATM labels for this logical link. User_side means that the hub does not assign labels. You can assign network_side to only one hub, the other must be configured as user_side.

**uni_version** This parameter defines the version of UNI signalling protocol (3.0 or 3.1) used on this logical link.

**traffic_type** This parameter allows you to define the type of connections supported by the NNI link using this logical link.

When you specify reserved_bandwidth for a logical link, the NNI connection using this logical link will only be used for reserved_bandwidth calls. In this case you must specify the amount of bandwidth available for the logical link. The reserved_bandwidth call will only be established if the requested bandwidth of the call can be satisfied by this NNI link.

When you specify non_reserved_bandwidth for a logical link, the NNI connection using that logical link will only be used for non_reserved_bandwidth calls. You cannot specify a bandwidth value for a non_reserved_bandwidth logical links. Therefore, the amount of bandwidth available will depend upon how much bandwidth is available for the module and how many reserved_bandwidth calls have been established already.

When you specify any for a logical link, the NNI connection using that logical link will be used for both reserved_bandwidth and non_reserved_bandwidth calls. With the logical links defined using any, you also specify a value for the bandwidth which determines the amount of bandwidth available for the reserved_bandwidth calls. In this case, the amount of bandwidth available for non_reserved_bandwidth calls will be a function of how much bandwidth is left over from the 212 Mbps available for that module used by the logical link.

If you have several NNI links going to the same remote cluster, that is if you are using link aggregate function, you will really have two aggregated links. One will be for all the NNI links that you have specified as reserved_bandwidth and any, the other will be all the NNI links that you have specified as non_reserved_bandwidth and what is left from the any bandwidth specification.

An example of using this command is explained in 4.10.2, “IISP Implementation in the 8260” on page 138.

5. Save the configuration changes.

After configuring A4-FB100 module and port settings, save your configuration with the following command:

```
8260A> SAVE module_port
```

6. Review your configuration changes.

   To display status about the configuration of the A4-FB100 module, use the SHOW MODULE command. An example of an A4-FB100 module in slot 4 is as follows:

```
8260A> show module 4 verbose

Slot Install Connect Operation General Information
--------------------------------------
4       Y       Y        Y     8260 ATM High Speed Module

8260A>
```

   To display more detailed information, use the SHOW MODULE VERBOSE command. An example is as follows:

```
8260A> show module 4 verbose

Slot Install Connect Operation General Information
--------------------------------------
4       Y       Y        Y     8260 ATM High Speed Module

P/N: 58G9471  EC level: C38844  Manufacture:VIME
Operational FPGA  version : 4
     Backup FPGA  version : 4

status: connected / hardware okay
        enable / Normal

Port type/ mode     / status
 1 : UNI / disabled / NO ACTIVITY
 2 : UNI / disabled / NO ACTIVITY
 3 : NNI / enabled  / OKAY
 4 : UNI / disabled / NO ACTIVITY

8260A>
```

   To display information about only one port, use the SHOW PORT command. An example follows for port 3 in slot 4.

```
8260A> show port 4.3 verbose
Port display for module 8260 ATM 100-Mbps Module

Port   Type   Mode      Status
-------------------------------------
4.03   NNI    enabled   OKAY

8260A>
```

   This command also accepts VERBOSE to display more detailed information of the port. An example is as follows:

```
8260A> show port 4.3 verbose
Port display for module 8260 ATM 100-Mbps Module

Port   Type   Mode      Status
-------------------------------------
4.03   NNI    enabled   OKAY

Connector            :  MIC
Media                :  fiber
Port speed           :  100000 Kbps
Remote device is active
IX status            :  IX OK

8260A>
```

## 5.2 Configuring ATMflex Module

The settings for the ATMflex module are the same as those for the A4-FB100, as explained in 5.1, “Configuring A4-FB100 Module” on page 191, except when you configure the ports, using SET PORT command, you can specify the following additional parameters:

- **Clocking**

  **internal_clock** This parameter specifies that the transmit clock is provided by the ATMflex module. This parameter is applicable when the port is configured to use sonet_sts-3c.

  **external_clock** This parameter specifies that the transmit clock is provided by the network or the station connected to this port. This parameter is applicable when the port is configured to use sonet_sts-3c.

- **Network**

  This parameter specifies the type of network to which this UNI or NNI port is connected.

## 5.3 Configuring ATM-LAN Bridge Module

To configure the ATM-LAN Bridge module, you must do the following:

1. Install the module in the 8260.

   This module may be installed in slots 1-8 and 12-17 of the 8260 Model A17 and slot 1-8 of the 8260 Model A10. Note that the ATM-LAN Bridge module is a two-slot module with the ATM backplane connection on the left side of the module when looking at the module as it would normally reside in the 8260. Therefore, this module cannot be the installed in slots 11-12 as slot 11 does not have an ATM connection.

   When you install an ATM-LAN Bridge module in slots that were not being used previously by another ATM-LAN Bridge module (that is, there is no saved configuration for an ATM-LAN Bridge module in the A-CPSW module for those slot), the module is, by default, set to isolated mode. In this case, if you display the status of such a module (using the SHOW MODULE command), you will see information similar to the following:

```
8260A> show module 7 verbose

Slot Install Connect Operation General Information
------------------------------------------------------------------------
  7     Y       n        n     8260 ATM LAN Bridge Module


status: not connected / hardware okay
        enable  / Normal

ATM Carrier Module Information:
-------------------------------
P/N:51H3862  EC level:E28091  Manufacture:VIME
8260A>
```

**Note:** All the commands entered for this module must refer to the left slot position occupied by the module. In the previous example, the ATM-LAN Bridge module is installed in slots 7 and 8.

2. Connect the module to the ATM backplane,

   To be able to use the ATM-LAN Bridge module, you must first connect the module to the backplane using the SET MODULE command. An example for an ATM-LAN Bridge module in slot 7 is as follows:

```
8260> set module 7 connected
```

   When you use the previous command, the module will be connected to the backplane and it will be *disabled* by default.

   **Note:** Since the ATM-LAN Bridge module uses the Utopia ATM carrier module, it has two backplane interfaces. However, only the first interface is used by the ATM-LAN module.

   You can establish the ATM-LAN Bridge module's connection to the backplane and *enable* its ports using the following example:

```
8260A> set module 7 connected enable
```

   The above command will result in the ATM port being configured as a UNI port, with the following settings:

   - Flow_control = disabled
   - ILMI = forced_sig_3_0

3. Enable the ATM-LAN Bridge ATM port.

   After you connect an ATM-LAN Bridge module to the ATM backplane, you must enable the ATM port using the following command:

```
8260A> set port 7.3 enable nni
```

   **Note:** The only parameters that you may specify for this port are:

   - Flow_control = disabled
   - ILMI = forced_sig_3_0

   With flow control disabled, the ILMI flow control is deactivated.

4. Configure a bridge profile using the Configuration Utility Program.

   The Configuration Utility Program allows you to configure a *bridge profile*, which specifies the parameters to be used in the ATM-LAN Bridge module.

This profile, which is created offline, can be saved on a disk file. The parameters that must be used to configure the bridge profile are fully described in *Nways 8260 ATM TR/Ethernet LAN Bridge Module Installation and User's Guide*, SA33-361. The following is an example of such a profile, which is used to configure one token-ring port and the ATM port, using no filtering.

```
Bridge Type
    TOKEN RING
        BRIDGE-WIDE CONFIGURATION
            BRIDGE NAME = 8281MOD
            MAXIMUM FRAME SIZE = 4399
            BRIDGE CONFIGURATION
                SPANNING TREE
                    BRIDGE PRIORITY = 32768
                    HELLO TIME = 2
                    FORWARD DELAY = 15
                    MAXIMUM AGE = 20
                SOURCE ROUTING
                    BRIDGE NUMBER = F
                    DEFAULT INBPUND FILTER ORDER
                        HOPE COUNT = 1
                        MAC ADDRESS = 2
                        RING NUMBER = 3
                        SNAP FILTER = 4
                        SSAP FILTER = 5
                ATM PHYSICAL PORT
                     ILMI VIRTUAL CONNECTION
                        VPI = 0
                        VCI = 16
                     ATM ADDRESS TERMINAL PORTION
                        LOCAL = 40-0-0-82-81-A1
                     GENERIC FLOW CONTROL = OFF
                     OPERATION AND MAINTENANCE/FLOW F5 = OFF
            SNMP CONFIGURATION
                DESCRIPTION
                    BRIDGE CONTACT = MOHAMMAD SHABANI
                    BRIDGE LOCATION = ITSO, CARY
                COMMUNITIES
                     MONITOR VIEW
                         COMMUNITY NAME = PUBLIC
                         HOST IP ADDRESS = 9.24.104.12
                         ADDRESS MASK = 255.255.255.0
                         ACTION = ADD
                     RUN-TIME CONTROL VIEW
                         COMMUNITY NAME = PUBLIC
                         HOST IP ADDRESS = 9.24.104.12
                         ADDRESS MASK = 255.255.255.0
                         ACTION = ADD
                     CONFIGURATION VIEW
                         COMMUNITY NAME = PUBLIC
                         HOST IP ADDRESS = 9.24.104.12
                         ADDRESS MASK = 255.255.255.0
                         ACTION = ADD
                     SECURITY VIEW
                         COMMUNITY NAME = PUBLIC
                         HOST IP ADDRESS = 9.24.104.12
                         ADDRESS MASK = 255.255.255.0
                         ACTION = ADD
```

```
        TRAP MANAGEMENT
                COMMUNITY NAME = PUBLIC
                HOST IP ADDRESS = 9.24.104.12

        INTERNET PROTOCOL (IP)
                BRIDGE IP ADDRESS = 9.24.104.202
                BRIDGE NETWORK MASK = 255.255.255.0
                BRIDGE DEFAULT GATEWAY = 9.24.104.1
PORT 1
    PORT NAME = TR1
    MAC ADDRESS FORMAT = NON-CANONICAL
    MAC ADDRESS TO USE = UNIVERSAL
    MEDIA SPEED = 4 MBPS
    SPANNING TREE INITIAL PORT STATE = ENABLED
    TOKEN RING SETTINGS
    RING NUMBER = 58E
    MAXIMUM FRAME SIZE = 4399
        EARLY TOKEN RELEASE = OFF
        ACTIVE MONITOR PARTICIPATE = OFF
    SPANNING TREE
        SPANNING TREE MODE = AUTOMATIC
        PORT PATH COST = 63
    SOURCE ROUTE FILTERS
        HOPE COUNT FILTER
            MODE = OFF
        MAC ADDRESS FILTER
            MODE = OFF
        RING NUMBER FILTER
            MODE = OFF
        SNAP FILTER
            MODE = OFF
        SOURCE SAP FILTER
 ATM PORT
     PORT NAME = ATM_PORT
     MAC ADDRESS FORMAT = NON-CANONICAL
     MAC ADDRESS TO USE = LOCAL
         MAC ADDRESS = 40:00:00:82:81:A1
     SPANNING TREE INITIAL PORT STATE = ENABLED
     RING NUMBER = 58E
     MAXIMUM FRAME SIZE = 4399
     HOP COUNT LIMIT = 7
         ACTIVE MONITOR PARTICIPATE = OFF
     SPANNING TREE
         SPANNING TREE MODE = AUTOMATIC
         PORT PATH COST = 10
     LAN EMULATION
         LAN EMULATION SERVER ADDRESS
          USE NETWORK PREFIX SAM AS BRIDGE = SPECIFY
          SPECIFY NETWORK PREFIX =39:09:85:11:11:11:11:11:11:11:11:01:01
              TERMINAL PORTION =70:00:80:00:90:01:00
         LAN EMULATION SERVER SETTINGS
             REGISTRATION RETRY MAXIMUM = 3
             KEEP ALIVE COUNT MAXIMUM = 10
      ATM PHYSICAL PORT
          ILMI VIRTUAL CONNECTION
              VPI = 0
              VCI = 16
          ATM ADDRESS TERMINAL PORTION
              LOCAL = 40-0-0-82-81-A1
```

```
GENERIC FLOW CONTROL = OFF
OPERATION AND MAINTENANCE/FLOW F5 = OFF
```

5. Download the bridge profile into the ATM-LAN Bridge module.

   To download the bridge profile and activate the new bridge parameters, you must do the following:

   a. Connect the program running the Configuration Utility Program to the service port on the ATM-LAN Bridge module.

   b. Send the bridge profile to the ATM-LAN Bridge module and activate the new bridge parameters. Note that when sending the profile, you have the option of specifying **send** or **send and reset**. If you specify **send**, the profile is sent to the EPROM of the module overwriting the *stored* parameters. This means that the new parameters will become effective and overwrite the operational parameters the next time the module is reset or powered on.

      If you specify **send and reset**, the profile is sent to the EPROM of the module overwriting the *stored* parameters, and then the module is reset to make the new parameters the operational parameters. Note that the *reset* will interrupt all the traffic on the ATM-LAN Bridge module.

You may display information about the ATM-LAN Bridge module using the following example:

```
8260A> show module 7 verbose

Slot Install Connect Operation General Information
------------------------------------------------------------------------
  7     Y       Y        Y     8260 ATM LAN Bridge Module


status: connected / hardware okay
        enable  / Normal

ATM Carrier Module Information:
-------------------------------
P/N:51H3862  EC level:E28091  Manufacture:VIME
Operational FPGA version : B2E4
     Backup FPGA version : B2E4

     Type  Mode     Status                          Daughter Card Descr
------------------------------------------------------------------------
 7.01:UNI enabled  UP-OKAY                         ATM LAN Bridge
 7.02:UNI disabled NOT IN SERVICE                  --- NOT USED ---

8260A>
```

If you display the ATM connection on the ATM-LAN Bridge, the following information will be shown:

```
8260A>show port 7.1 verbose

     Type  Mode     Status                              Daughter Card Description
-------------------------------------------------------------------------------
 7.01:UNI enabled  UP-OKAY                             ATM LAN Bridge

Signalling Version   : with ILMI, forced 3.0
Flow Control         : Off

ATM LAN Bridge Card Information:
--------------------------------
 P/N:58G9720  EC level: 55953  Manufacture:VIME
 Model Number               : 8281
 Operational Status         : OKAY
 Boot Code Version          : v.1.1.0
 Operational Code Version   : v.1.14.0
 Number of Front Panel Ports: 4
 Number of Utopia Interfaces: 1
 Default Gateway IP Address : 9.24.104.1
 IP address                 : 9.24.104.202
 IP Subnet Mask             : FF.FF.FF.0

Physical Port:   Status   Network   Speed   Connector   MAC Address
-------------------------------------------------------------------
 01              BLOCKED  TR        4Mbps   RJ-45       0004AC87A081
 02              BLOCKED  TR        4Mbps   RJ-45       0004AC87A041
 03              FORWARD  TR        4Mbps   RJ-45       0004AC87A0C1
 04              BLOCKED  TR        4Mbps   RJ-45       0004AC87A021
 05              FORWARD  ATM       N/A     Backplane   4000008281A1

8260A>
```

# Chapter 6. 8260 Hardware Implementation

This chapter provides a brief description of the hardware components of the 8260 ATM switching subsystem and how they interact with each other.

## 6.1 Introduction

The 8260 hardware that is used to implement the ATM functions described earlier can be grouped into three categories:

- CAP/CAD
- Control Point
- Switch

Figure 92 shows how these three sections interrelate:

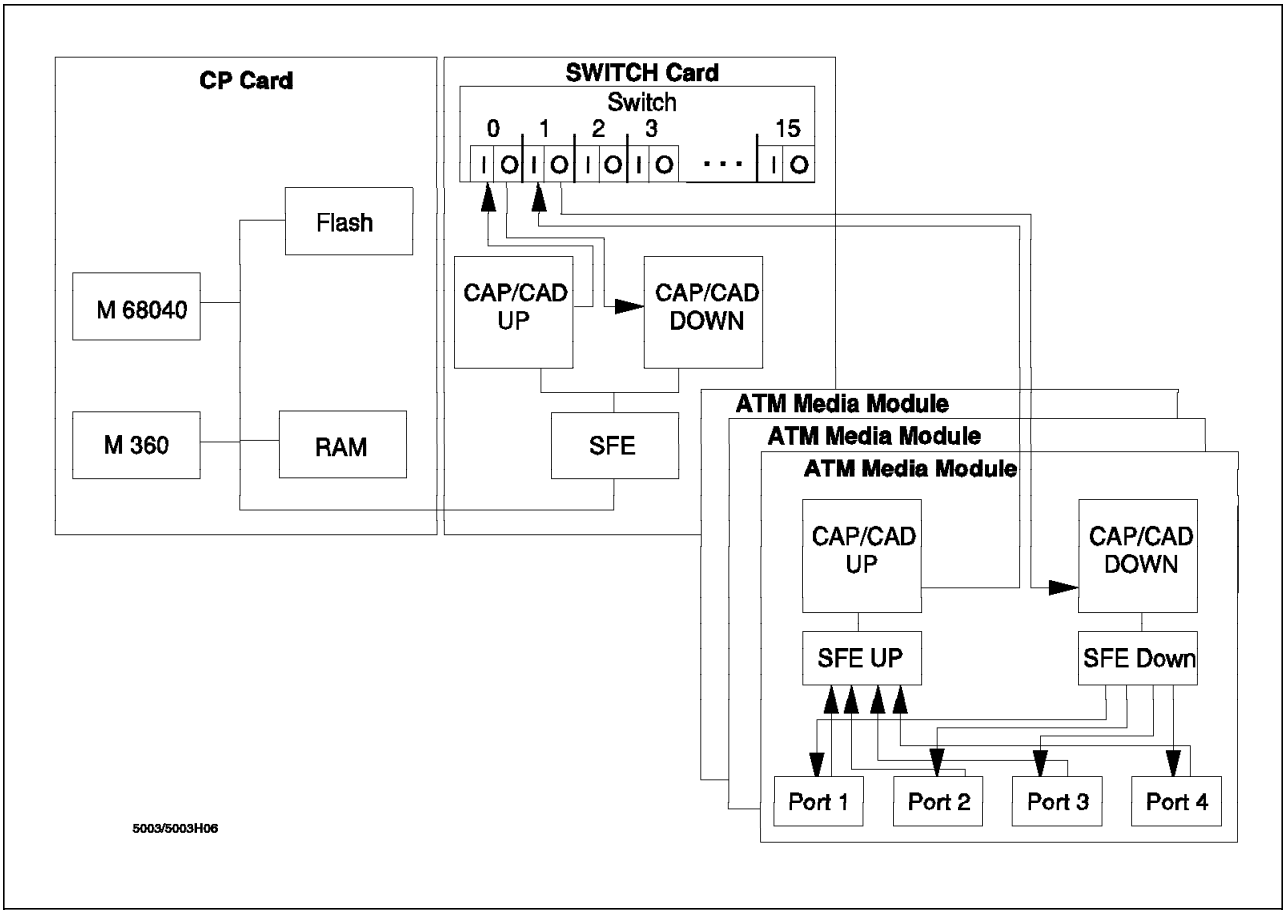



*Figure 92. Connection of the Control Point to the ATM Subsystem*

CAP/CAD stands for Common ATM Processor/Common ATM Datamover. It is explained fully in 6.1.2, “CAP/CAD” on page 205. The functions that it performs are as follows:

- Cell routing for point-to-point and point-to-multipoint connections
- Traffic management
- Accounting

© Copyright IBM Corp. 1996

The control point processor is a Motorola M68040 processor that resides on the control point card. It also has flash memory from which it loads the bootstrap code. The flash memory also holds the operational code that is run from RAM. A real-time multitasking operating system called pSOS is used by the control point function. The control point card performs all the other functions that the CAP/CAD modules do not perform, including:

- Topology and route selection
- Address mapping
- Resource management
- Signalling entities

The control point performs operations on the rest of the ATM subsystem by sending cells to control the rest of the 8260 ATM subsystem via an internal port connected to the switch (port 0).

The switch card performs exactly what its name implies: switching of cells. Although the switch has 16 ports, only 15 of them are used. Port 0 is used as an internal port that connects to the control point. Port 1 to Port 8 connect to slots 1 to 8 of the hub, respectively. Port 9 to Port 14 connect to slots 12 to 17 of the hub, respectively.

The switch card also has CAP/CAD modules like all the ATM media modules. The switch card's CAP/CAD modules are used to connect the control point to port 0 of the switch. The control point can only access the rest of the ATM subsystem via Port 0, which is why the control point uses cells to perform any operations on the ATM media modules.

### 6.1.1 Internal Cell Format

The IBM 8260 ATM does not actually switch ATM cells in their original 53-byte form. Instead it uses an internal 64-byte cell format. Figure 93 on page 205 shows a diagram of how the internal cell is organized.

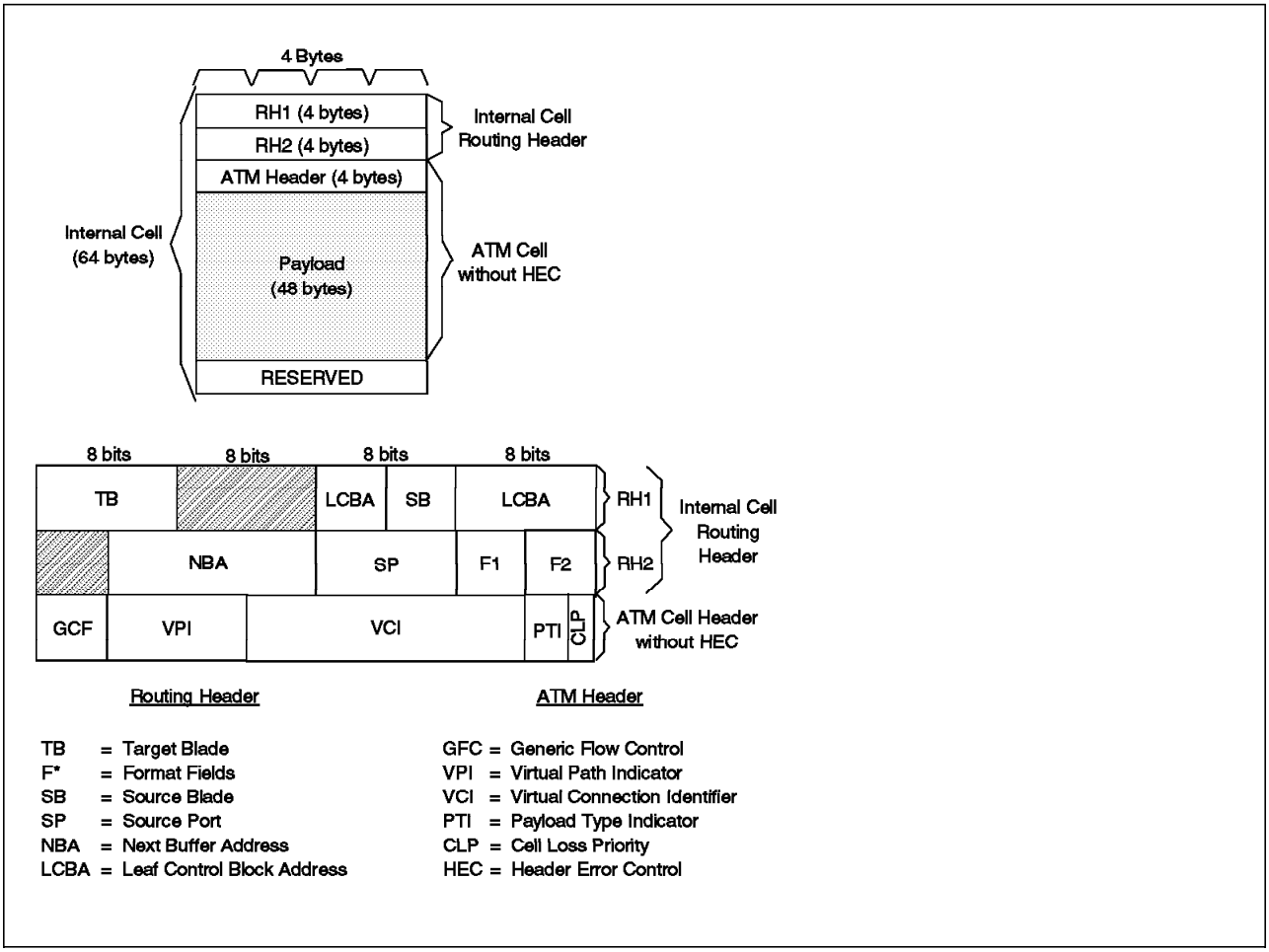



*Figure 93. Internal Cell Format*

## 6.1.2 CAP/CAD

All CAP/CAD functions are implemented on each ATM blade. Figure 94 on page 206 is a diagram of how the CAP/CAD functions on a 4-port 100-MBps ATM media module are arranged. Although the diagram describes a particular media module, it is applicable to other media modules except that the number of ports and the physical interface may be different.

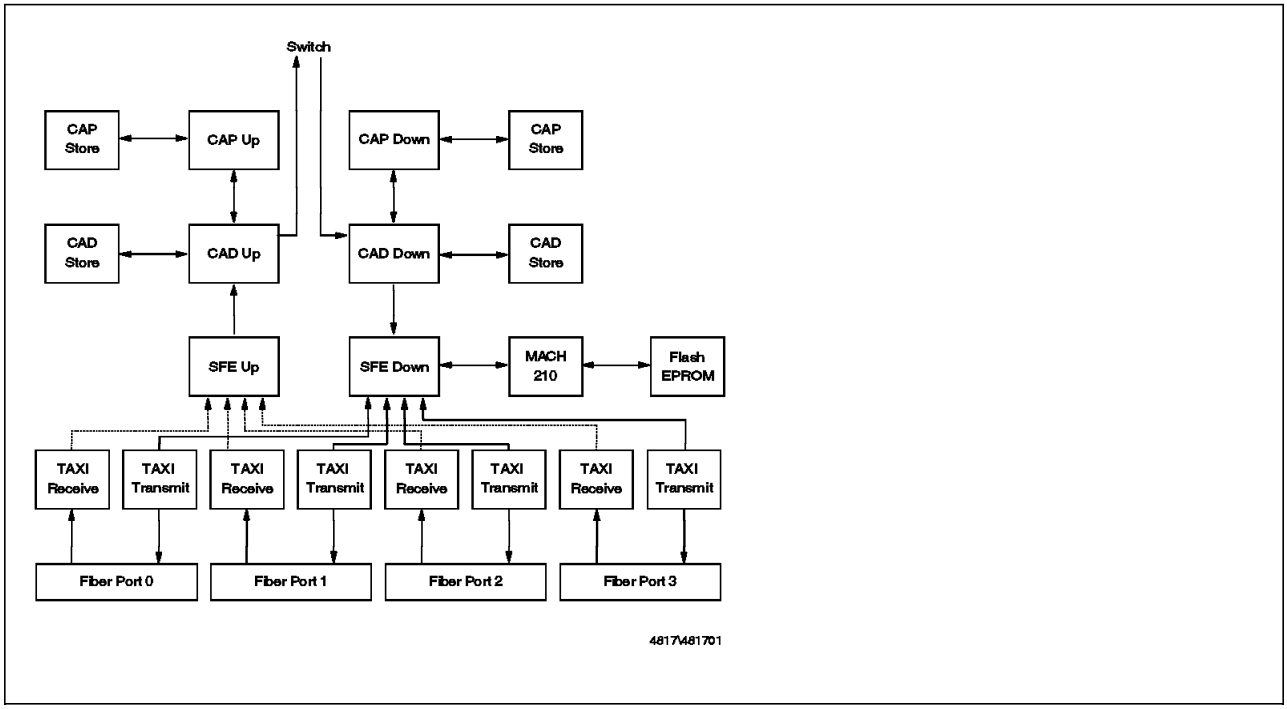



*Figure 94. A-CPSW components View with ATM Media Modules*

Figure 94 gives a generic component diagram of the 100-Mbps media modules that are connected to the A-CPSW Module. You will notice that there are basically three main components in the blades:

- Specific Front End (SFE)
- Common ATM Datamover (CAD)
- Common ATM Processor (CAP)

There are actually two sets of the three components in each blade. One set handles the inbound cells that come up from the ports to be dispatched to the switch. These modules are called SFEup, CADup and CAPup. The other set handles the outbound cells that are received from the switch to be sent back out of the ports as ATM cells. These modules are called SFEdown, CADdown and CAPdown. In a particular blade, there are also memory modules, called CADstore and CAPstore, which the CAD and CAP modules use. Along with connections between the various modules, there are also connections to the control point processor.

SFE handles the ATM front-end concentration and dispatch. In this module it concentrates on the four 100-MBps ports. Its main role is to deliver the cell from whatever type of interface (in this case it is a TAXI interface) to the CAD.

The CAD function prepares the cell for transmission to the switch. CAD builds the internal cell in CADstore according to instructions given by the CAP.

The CAP module handles the cell routing, queuing, scheduling and traffic management. It determines what the routing header for the internal cell should be and gives the information to the CAD to build the cell.

### 6.1.2.1 CAP/CAD Cell Routing

The following describes the process of how cells are switched from one port to another port in the ATM hub. The best way to understand this process is to follow a cell as it enters one port and exits another to see what actually happens as it goes through the various components. Please refer to 6.1.1, “Internal Cell Format” on page 204, as this section assumes that you are already familiar with the internal cell format.

***Point to Point:*** The following steps describe what happens to a cell in a point-to-point connection:

1. Receive the cell.
   a. For every port, the CADup module prepares in advance the address of the next cell assembly buffer, which is the location where the internal cell will be built in CADstore.
   b. An ATM cell is received by the SFEup module.
   c. The Header Error Control (HEC) in the ATM cell header is checked. In case of error, the whole cell is discarded. Otherwise, the HEC is stripped and the remaining 52 bytes are delivered to CADup.

      **Note:** The connection from SFEup to CADup is 32 bits wide (4 bytes) so the 52-byte ATM cell is transferred in 13 x 4-byte blocks.

      When the first 4-byte block of a cell gets transferred, one of the control lines is raised to indicate the beginning of the cell.

      Port lines between SFEup and the CADup indicate from which port the cell is coming. By using these port lines, the 4-byte blocks transferred can be mixed in CADup, whatever its original port. For example, SFEup can deliver a 4-byte block from Port 1, then deliver a 4-byte block from Port 2, then deliver a 4-byte block from Port 1 and so on. This ensures that no time is wasted in delivering data from a port that has no cell.
   d. The next steps depend on the status of the control line cell indicator.
      - **It is not the first 4-byte block of a cell:**

        SFEup places each 4-byte block of the cell in the CADup, which stores it in CADstore using the address of the next cell assembly buffer prepared previously.
      - **It is the first 4-byte block of a cell:**

        CADup stores the third 4-byte block in the CADstore, skipping the first 8 bytes. The first 8 bytes are reserved for the routing header in the internal cell.
      - **In every case:**

        A 4-bit register, used to point to the lower address where the next 4-byte block location should be memorized, is updated. This 4-bit register is used as a displacement pointer from the cell assembly buffer.
   e. CADup writes source port (SP) parameters in the routing header (RH).

      When a cell is completely assembled in the CADStore, CADup put the cell assembly buffer address in a general queue (GQ), which buffers all assembled cells coming from all module ports. The GQ allows all cells coming from any port received at the same time to be fully completed and assembled consecutively within an SFE/CADup interface cycle.

In the general queue, each cell will then be dequeued on a first-in/first-served basis.

CADup sends a copy of the first 4-byte block with the source port (SP) parameters to CAPup, then prepares the address of the next assembly cell buffer for this port. The address is determined from the port number, which points to a register where these 4 bytes should be placed.

2. Prepare the routing header (RH).

   a. The first 4-byte block of a cell is the first 4 bytes of the ATM cell header, which contains the VPI/VCI. When CAPup receives the first 4-byte block with source port (SP) parameters it now has all the information it needs to identify a particular connection: SP, VPI and VCI. From these three values, CAPup determines the inbound leaf control block address (LCBAup), which is the pointer to the leaf control block (LCB) for this connection.

   b. The LCB contains target blade (TB) parameters.

      TB, LCBAup, source blade (SB), and RB/NRB connection parameters are given to CADup to be written to the internal cell header in CADstore.

      CAPup, which knows the address of the beginning of this cell, forwards the cell header address to CADup to ensure that the information is written at in the correct place in CADstore.

      In the case of an unknown SP/VP/VC, the cell is released by CAPup by sending to CADup the cell buffer address, which can be used for another data movement.

      CAPup also performs ″Smart Discard″ on NRB AAL5 frame flows, which purge cells on an AAL5 frame basis in the case of NRB node congestion.

3. Place the cell in the queue.

   CADup puts the cell in the appropriate output queue (with the RB/NRB indication) so that prioritization of traffic can occur. There is an RB queue and an NRB queue.

   The cell is now ready to be given to the Switch-on-a Chip.

4. Switch the cell.

   a. When the Switch-on-a Chip indicates to CADup to give the next cell, CADup gives the first cell from the appropriate queue based on its priority mechanism (RB over NRB Queue).

   b. The cell is delivered to the Switch-on-a Chip, and the pointers of that queue are updated.

   c. The Switch-on-a Chip switches the cell based on the target blade (TB).

5. Receive the cell into the target blade.

   a. CADdown has prepared a location in advance for the next cell.

   b. CADdown receives the cell into CADstore in the general queue.

   c. CADdown dequeues the cell and sends CAPdown a copy of the RH, which contains the LCBAup and the source blade.

6. Place the cell in the correct output queue and prepare it for transfer to the SFEdown module.

   a. Using SB and LCBAup, CAPdown determines the LCBAdown. LCBAdown points to the LCB for the connection in the outbound blade.

The LCB has VPI/VCI out, target port (TP), RB/NRB and multicast indications. For performance reasons. a part of the LCB is set in a CADstore shadow zone.

b. LCBAdown, TP, NRB/RB, and multicast indications from the LCB are given to CADdown.

CADdown queues the cell in the corresponding target port queue (One RB, one NRB per port) with the indication received from CAPdown. and prepare it for transfer to SFEdown.

7. Prepare and send a new ATM cell.

   a. When SFEdown asks for the next cell of a port, CADdown moves the contents of LCBshadow, which has VPI/VCI out and the type of swapping (SWAP_TYPE) to be performed, plus the 52-byte cell to SFEdown.

   b. SFEdown modifies the header based on SWAP_TYPE. SWAP_TYPE indicates if only the VP needs to be swapped, if both the VP and VC need to be swapped or neither need to be swapped. The **Payload Type indicator**(PTI) field
   is always retrieved from the incoming header.

   c. SFE generates HEC.

   d. SFE presents the cell to TAXI.

***Point to Multipoint:*** In a point-to-multipoint (multicast) connection, the process is very similar. Steps remain the same until the cell is given to the Switch-on-a Chip. The TB field is now indicating that this cell is part of a multicast connection by having the first bit of the TB (target blade) set to 1. The other 7 bits form the Multicast ID (Mid).

In a point-to-point connection, the first bit is set to 0 and the other 7 bits indicate the target blade where the Switch-on-a Chip must switch the cell to.

The following describes the steps just after the Switch-on-a Chip has received the multicast cell:

1. Switch the cell.

   a. The Switch-on-a Chip recognizes that the TB is actually a Multicast ID; thus, using the Mid as a pointer, it looks at its switch multicast tree table to get 16 bits. Each bit corresponds to a blade. If the bit is on, then that blade is part of the multicast tree.

   b. The Switch-on-a Chip switches the cell to the target blades based on the multicast tree table.

2. Receive the cell into the target blade.

   This step is the same as in a point-to-point connection described earlier.

3. Place the cell in the correct output queue and prepare for transfer to SFEdown.

   a. Using SB and LCBA, CAPdown determines LCBAdown. Since this is a multicast connection, LCBAdown actually points to a chain of LCBs. Each LCB in the chain represents the branches on the multicast tree on this blade. Each LCB in the chain has VPI/VCI out, SWAP_TYPE and target port (TP) and last multicast (LAST_MC) indication. There is also a shadow of the LCB chain in CADstore for performance reasons.

b. The same steps as in the unicast case applies. But when the cell has been sent to SFEdown, the CADdown will re-enqueue this cell in the general queue so that CAP will reprocess this cell with the next LCB in the chain. This is done till CAPdown finds the LAST_MC indication in the LCB.

4. Prepare and send a new ATM cell.

# Chapter 7. Nways Campus ATM Manager

This chapter describes the management functions that are available with an 8260 ATM network. It provides a brief overview of the MIBs that are available, functions of ATM Campus Manager and an explanation of how to perform some of the functions that we have found useful in configuring the 8260 ATM network. For detailed information about ATM campus network management, please refer to *ATM Campus Network Management,* SG24-5006.

## 7.1 Management Information Bases (MIBs)

The A-CPSW provides full SNMP support with the use of standard SNMP commands: get, getnext, set and traps. Below is a list of all the MIBs an 8260 ATM network supports that any SNMP-based management can use:

- MIB-II

  The 8260 ATM subsystem fully supports this MIB. For the purposes of the system group, ATM is treated as a data link protocol. The interface group describes the ATM cell layer interface. This group only concerns itself with the ATM cell layer as a whole and not the individual connections. Here the amount of traffic that was transmitted and received can be found. Also the number of cells dropped due to an incorrect HEC and invalid ATM cell header will be found.

- IETF AToMIB

  This MIB is described in RFC 1695. It describes objects used for managing ATM-based interfaces, devices, networks and services. The following are descriptions of the various groups.

  – The ATM Interface Configuration Group

    This group describes the type of ATM traffic on a particular interface. It contains ATM interface configuration parameters such as the status of the interface, maximum number of VPCs and VCCs supported on an interface, the number of configured VPCs and VCCs, the number of active VPI and VCI bits, VPI/VCI of ILMI (if at all) and the ATM Address type.

  – The DS3 PLCP Group

    This group has configuration and state information for those ATM interfaces that use DS3 for carrying ATM cells. Since the 8260 ATM networks do not use DS3 interfaces, this group is not used.

  – The ATM Traffic Descriptor Parameter Group

    This group has information relating the ATM traffic parameters, including the QoS class.

  – ATM Virtual Path Link (VPL) Group

    This group contains configuration and state information of a bidirectional VPL. Here VPs can be created, deleted or modified.

  – ATM Virtual Channel Link (VCL) Group

    This group contains configuration and state information of a bidirectional VCL. VCs can be created, deleted or modified here. Also, information

© Copyright IBM Corp. 1996

can be found on the AAL that is in use on a VC; specific information can be found if AAL5 is used, such as the type of data encapsulation.

  - The Virtual Path (VP) Cross Connect Group

    This group contains configuration and state information of all point-to-point and point-to-multipoint VP cross connects. In other words it gives information on the VP swapping table. With this group VP cross-connects can be established and removed.

  - The Virtual Connection (VC) Cross Connect Group

    This group performs the same functions as in the VP Cross Connect Group except for VCs.

  - The AAL5 Virtual Channel Connection Performance Statistics Group

    This group contains the AAL5 performance statistics of a VCC.

- OSPF MIB

  Since the TRS function uses OSPF with very few modifications to the original code, the 8260 ATM network supports the OSPF MIB unchanged.

- ILMI MIB

  This MIB is defined by ATM Forum in V3.0 of the UNI specification. Following is a brief description of the groups defined in these MIBs:

  - Physical Port Group

    This group gives information on a particular port such as the status, transmission types (for example 4B/5B encoding at 100 Mbps or Sonet STS-3c at 155.52 Mbps) and cable type.

  - ATM Layer Group

    This group has the maximum number of supported VPs and VCs on the UNI, the number of VPs and VCs configured on the UNI and the number of active VP and VC bits on the interface.

  - ATM Statistics Group

    Here you fill find the number of cells received, dropped and transmitted on the UNI.

  - Virtual Path Group

    This group gives information on the VPs on the UNI. This includes status, traffic shaper parameters, policing parameters and QoS.

  - Virtual Channel Group

    This group performs the same functions as the Virtual Path Group except for VCs.

  - Network Prefix Group

    This group has information on the network prefix in use on the user side of the UNI and its validity.

  - Address Group

    This group has information on the ATM address in use on the user side of the UNI and its validity.

- IBM Hub-Specific MIB Extensions

  - Traps Control Group

This group allows you to configure what traps are sent.

- Switch Control Group

  This group determines which slots are controlled by the switch.

- ATM Modules Group

  This group gives details on the modules such as the maximum number of supported VPs and VCs, the number of VPs and VCs in use and the type of module.

- ATM Port Group

  Information can be found here on the number of ports on a module, cable type, status and what interfaces it supports (UNI, NNI or SSI).

- The ATM Interfaces Group

  This group maps for each ATM port the MIB-II interface index and the physical slot/port numbers.

- Cross Connect Group

  Information on the label swapping tables for VPs and VCs is stored here.

- Neighbor Devices Group

  Here information can be found on the ATM devices connected on ports such as the IP address and description.

- TFTP Group

  This group controls the parameters for TFTP download functions.

- Statistics Group

  Statistics for individual VP and VC connections are found here.

- IBM Signalling Extensions

  The IBM MIB extension defines ATM signalling support on the device. Below is a brief list of the information that can be accessed via this MIB:

  - Number of supported signalling channels.
  - Range of reserved VPs and VCs.
  - VPI/VCI used for the signalling channel on a port.
  - The state of the Q93B interface.
  - Q93B statistics such as the number of call attempts and rejections.
  - Information about Q93B calls in progress such as calling and called party.
  - Details of cleared calls including the ATM interface involved, called party and calling party, date and time, cause of clearing, QoS requested and the bandwidth requested.
  - Details and statistics on the SAAL.

- IBM PVC management MIB Extensions
- IBM ATM Statistics MIB Extensions

## 7.2 ATM Campus Manager Overview

Nways Campus Manager for ATM (referred to as ATMC in this publication) is a program that runs under SystemView for AIX. Although this product is not mandatory to configure and set up an 8260 ATM network, it is highly recommended because the information and functions that it provides will make it significantly easier to work with the ATM network.

ATMC provides automatic discovery of ATM nodes and physical links. It will automatically place the devices in the ATM hierarchy of ATM network, clusters and nodes. All the physical interfaces of a node can be viewed including the internal port. Any changes to the topology are automatically reflected by changes in the network map. ATMC also allows you to monitor and configure the ATM resources. These include the physical ports, PVCs, SVCs, VPs and VCs. It is able to tell you the characteristics of the ATM device that is connected to a particular port.

ATMC interoperates with SytemView for AIX to provide:

- Display of traps
- Color coding of status information
- Logging of call failures

ATMC allows you to perform connection tracking functions:

- List and delete SVCs
- Show the characteristics of SVCs such as the calling party, called party and QoS parameters
- List, delete and create PVCs
- Show the characteristics of a PVC
- Track the connection including the VPI/VCI labels of each segment of the connection and what physical ports the connection goes through

ATMC allows you to collect statistical information and display it in a more readable graphical format. Below is a list of some of the different types of information that it can collect:

- Logging of calls

  All calls, that is calls in progress and calls that have been cleared, on a node can be logged with information such as calling number, called number, creation time, clear time and clear cause.

- Traffic

  Statistics on an interface′s traffic can be gathered with information such as received cells, transmitted cells, discarded cells and invalid cells.

- Bandwidth

  Information about the amount of bandwidth that is utilized on a port can be found.

- Q.2931 status

  Information on the incoming and outgoing calls in progress can be collected.

- SAAL Errors

Information on the various errors detected by SAAL can be collected.

### 7.2.1 ATMC Prerequisites

This section lists the recommended hardware and software requirements for the installation and operation of ATMC.

**Hardware Requirements**

- RISC System/6000 or POWERserver (minimum 33 MHz CPU)
- An IBM compatible mouse
- Color display
- Min. 48 MB of RAM
- Min. 20 MB of free space disk
- Min. 144 MB of paging space
- A tape drive (8 mm or 1/4 in.) for installation
- A connection for running TCP/IP between 8260 and ATMC Manager
- Classical IP over ATM

**Software Requirements for ATMC Manager**

- AIX V3.2.4 or higher
- One of the following:
  - AIX NetView/6000 V2
  - NetView for AIX V3
  - NetView Entry for AIX V3
- AIXwindows Environment/6000 V11 R4 or R5
- OSF/Motif V1.1 or 1.2
- X-Window System V11 R4 or R5

**Note**

For getting a graphical display of the 8260s with its modules, it is also desirable to have the Nways Hub Manager for 8260 application installed in the manager workstation.

## 7.3 Using ATM Campus Manager

This section describes how you can use ATMC to manage your 8260-based ATM subsystem.

### 7.3.1 ATMC Manager Views

When you navigate through ATMC, you get the following submaps in a hierarchical level:

- Root Submap
- ATMC Campus Submap
- ATMC Cluster Submap

- ATMC Switch Submap

If you have Nways Hub Manager installed in your workstation, you can use it to get into ATMC. Figure 95 shows how to navigate through ATMC and its connections with Nways Hub Manager and the IP Internet applications of NetView/6000.

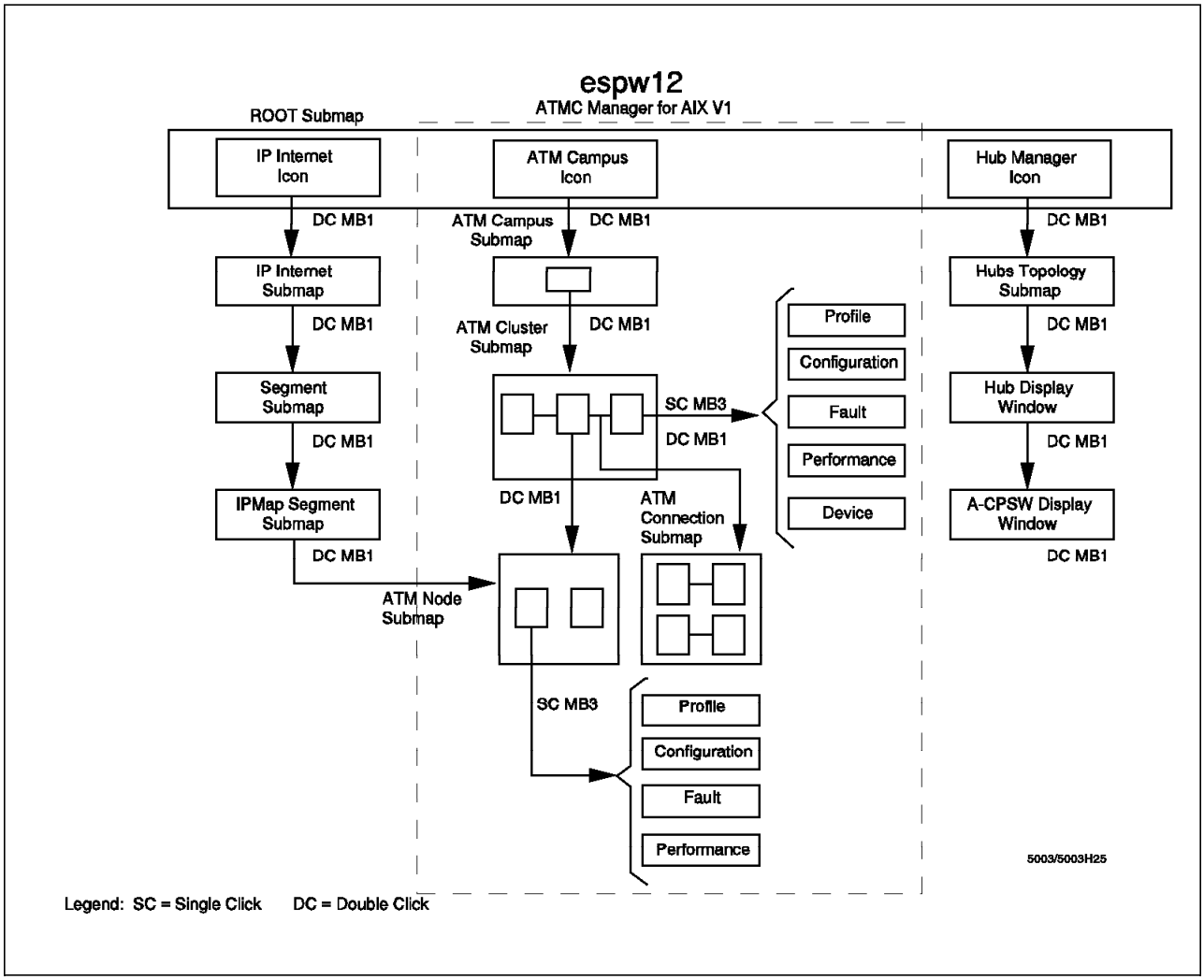



*Figure 95. Navigating ATMC*

### 7.3.1.1 Root Submap

The NetView for AIX Root Submap shown in Figure 96 on page 217 is the access point when using ATMC. From the Root Submap you can:

- Manage the ATM Campus

  When the ATM campus is managed, each node of the ATM campus will be polled every *polling interval* seconds.

- Unmanage the ATM Campus

  An unmanaged ATM campus is not managed by the ATMC manager, meaning that none of the nodes in this campus will be polled by the ATMC manager.

- Explode the ATM Campus icon

This allows you to display the ATM Cluster-level view in the ATM Campus Submap.

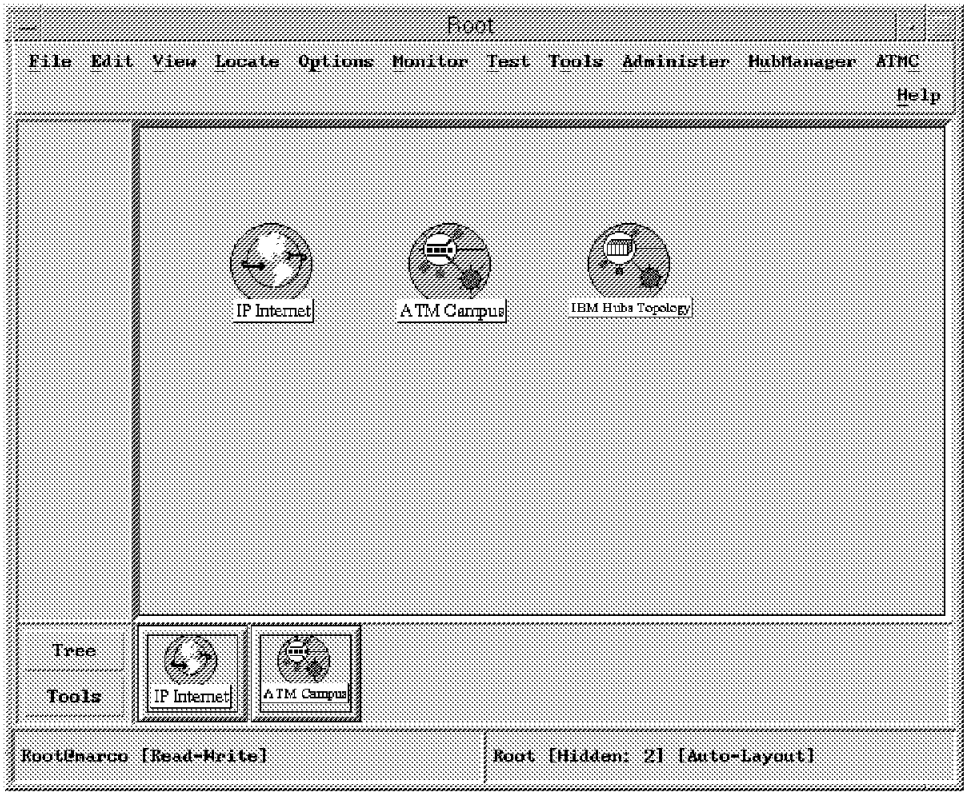



*Figure 96. Root Submap*

### 7.3.1.2 ATMC Campus Submap

The ATMC Campus Submap, as shown in Figure 97 on page 218 displays all the clusters in your ATM campus. From this submap you can choose to have a cluster be managed or unmanaged by ATMC. When a cluster is managed, it can be exploded to display the ATM Node-level in the ATM Cluster Submap.

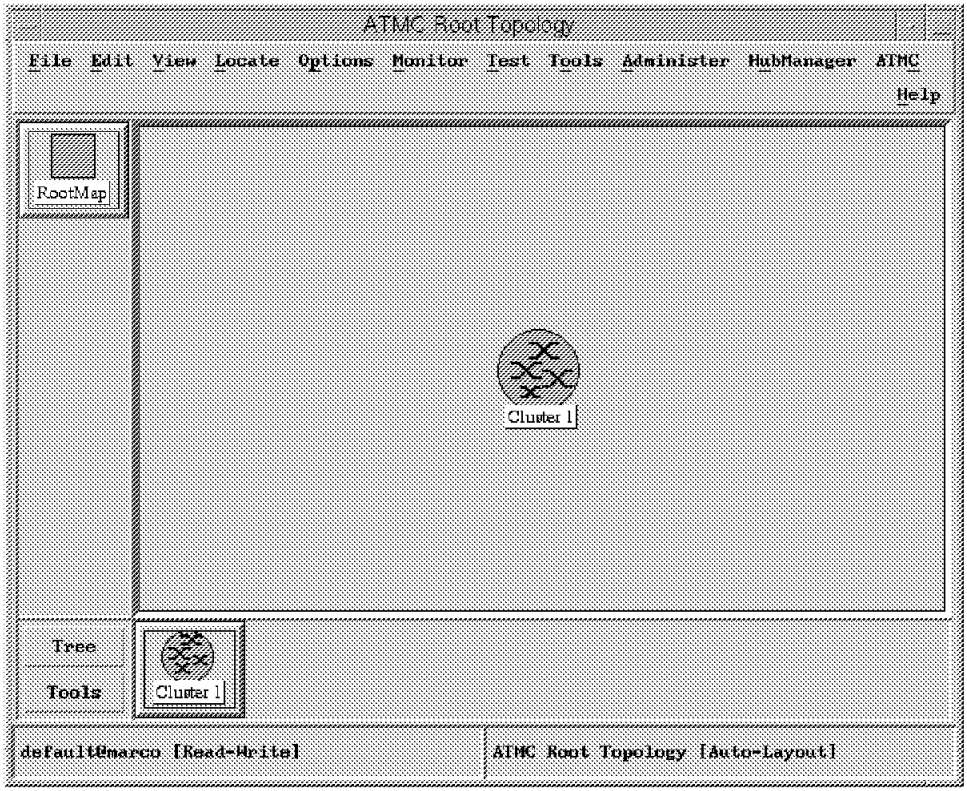



*Figure 97. ATMC Campus Submap*

### 7.3.1.3 ATM Cluster Submap

The ATM Cluster Submap shown in Figure 98 on page 219 displays the node-level view and contains the icons representing the 8260 ATM nodes and the ATM physical links between them. From this submap you can manage or unmanage ATM nodes. Also, from the ATMC menu you can choose the following for each node:

- Profile
- Configuration
- Fault
- Performance
- Device

These options are explained in detail in 7.3.3, “Displaying 8260 Node Related Information” on page 221.

If a node is managed, you can explode it to display the interface-level view in the ATM Node Submap.

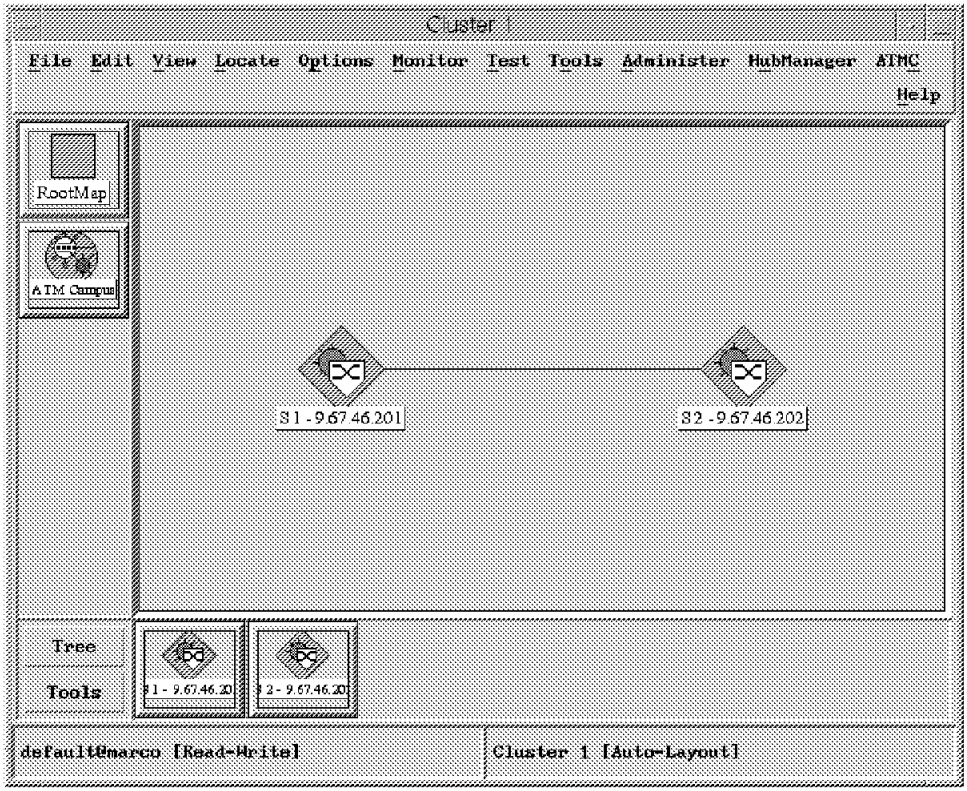



*Figure 98. ATM Cluster Submap*

### 7.3.1.4 ATM Node Submap

The ATM Node Submap shown in Figure 99 on page 220 displays the interface-level view and contains icons representing the physical ATM ports of the 8260 and the ATM node internal interface. The interface number shown for each port is the slot.port.

From this submap you can manage/unmanage a specific interface or all interfaces. Also, you can select the following items from the ATMC menu for each interface:

- Profile
- Configuration
- Fault
- Performance

These options are explained in 7.3.4, “Displaying ATM Interface Related Information” on page 223.

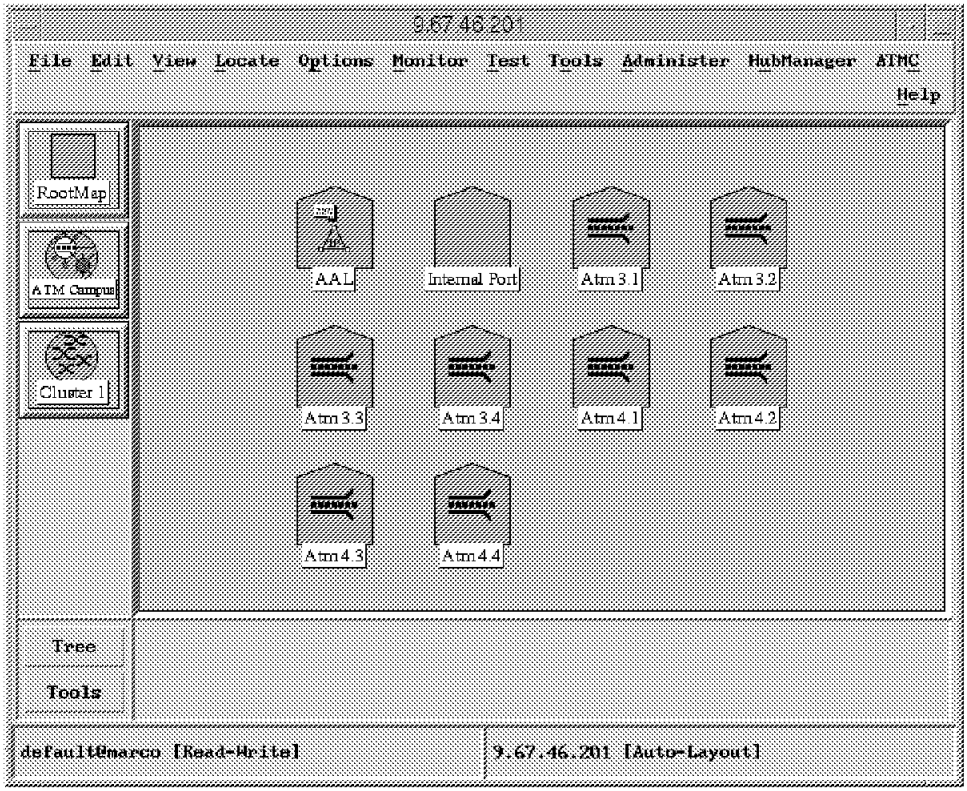



*Figure 99. ATM Node Submap*

## 7.3.2 Accessing ATMC from Other SystemView for AIX Applications

As shown in Figure 95 on page 216, you can have access to ATMC through the IP Internet and Hub Manager Applications.

### 7.3.2.1 Accessing ATMC from IP Internet

You can access ATMC from the IP Internet application by doing the following:

1. From the Root Submap select the **IP Internet** icon and double-click on it to enter the IP Internet Submap.
2. Select the specific segment which has the IP addresses of your ATM network. This will take you to the IP Segment Submap.
3. Double-click on the icon of the ATM node you want to display. This will take you to the ATMC Node Submap.

### 7.3.2.2 Accessing ATMC from Hub Manager

If you have installed Hub Manager in your manager workstation, you can access ATMC by doing the following:

1. From the Root Submap select the **Hub Manager** icon and double-click on it. This takes you to the Hub Topology Submap.
2. Select the specific 8260 you want to display. This takes you to the hub display window, as shown in Figure 100 on page 221.

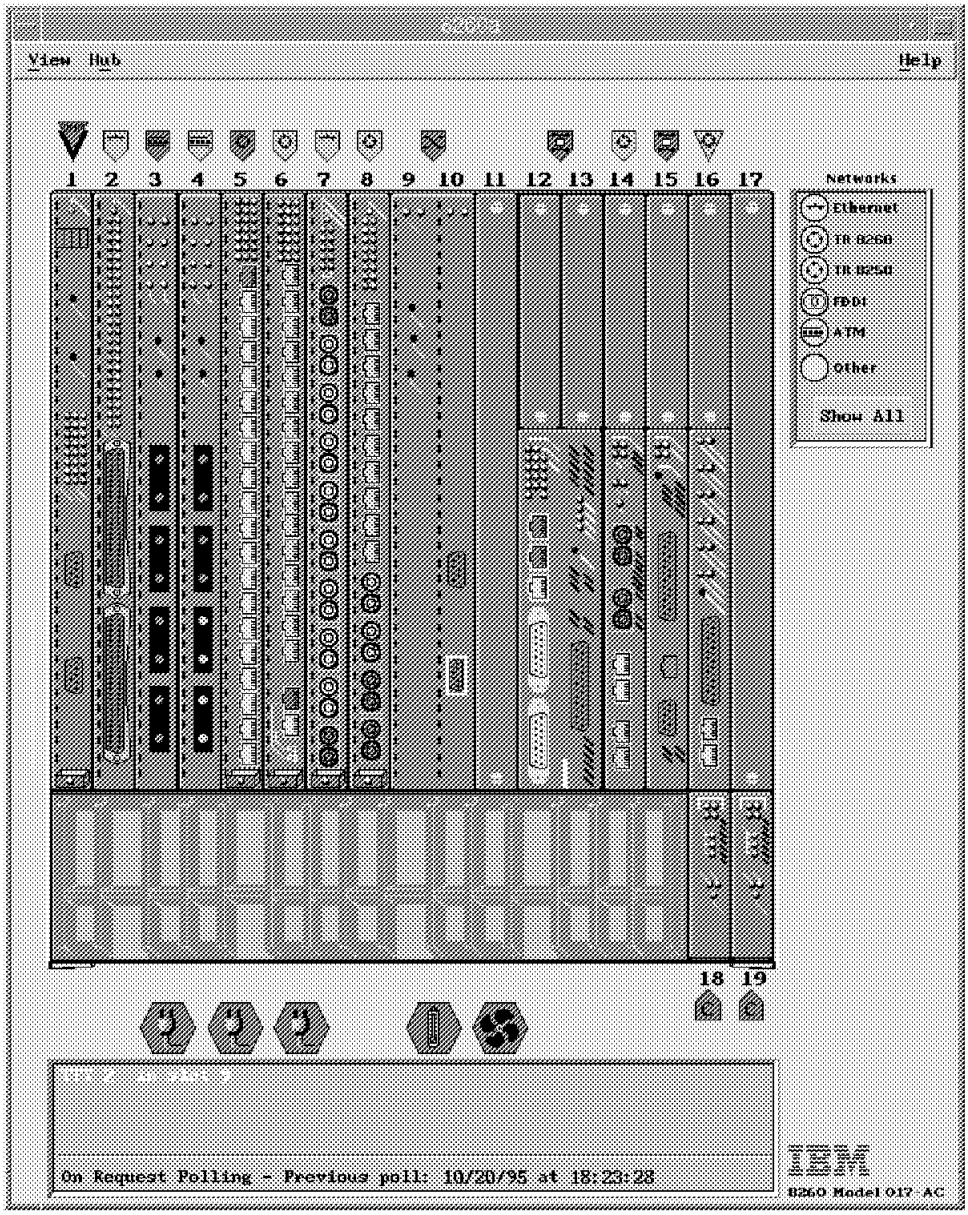



*Figure 100. Hub Display Window*

3. Double-click on the icon above the A-CPSW module. This will take you to the A-CPSW module window.
4. Double-click on the icon in the lower right corner of the A-CPSW module window to access the ATMC Cluster Submap.

### 7.3.3 Displaying 8260 Node Related Information

As explained in 7.3.1.3, “ATM Cluster Submap” on page 218, you can access the following 8260 information from the ATM Cluster Submap:

- Profile
- Configuration
- Fault
- Performance

The following sections provide you with more details about these options.

### 7.3.3.1 8260 Node Profile

This panel, as shown in Figure 101, allows you to modify the following:

- Contact Person
- Administratively assigned name
- Location

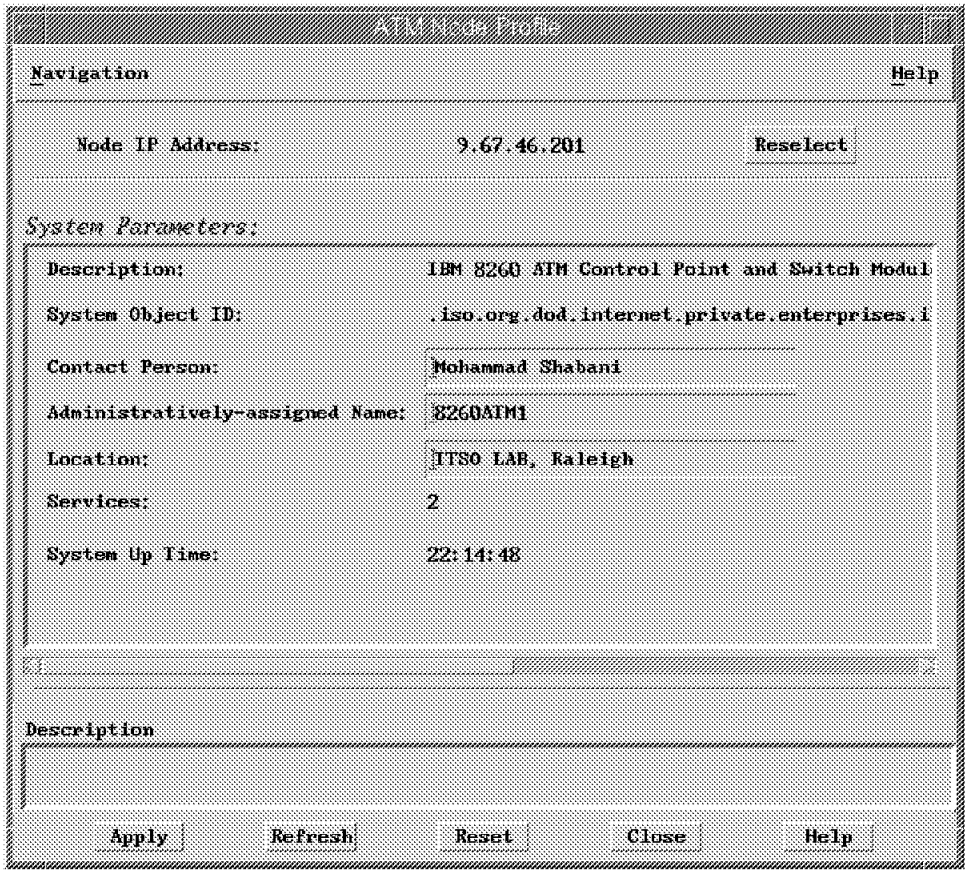



*Figure 101. Displaying 8260 Node Profile*

### 7.3.3.2 Node Configuration

This panel, as shown in Figure 102 on page 223, can be used to:

- Display configuration information

  This is explained in detail in 7.3.4, “Displaying ATM Interface Related Information” on page 223.

- Lock and unlock selected ATM nodes

  This is done to ensure that the operator cannot unintentionally disable the port for exchanging network management information between the ATMC Manager station and the hub.

- List the interfaces on the selected ATM node
- From the services option in the menu bar you can select any of two items:
  - Trace and Dump (System Trace, TRS Trace, System Dump, TRS Dump)
  - File Transfer

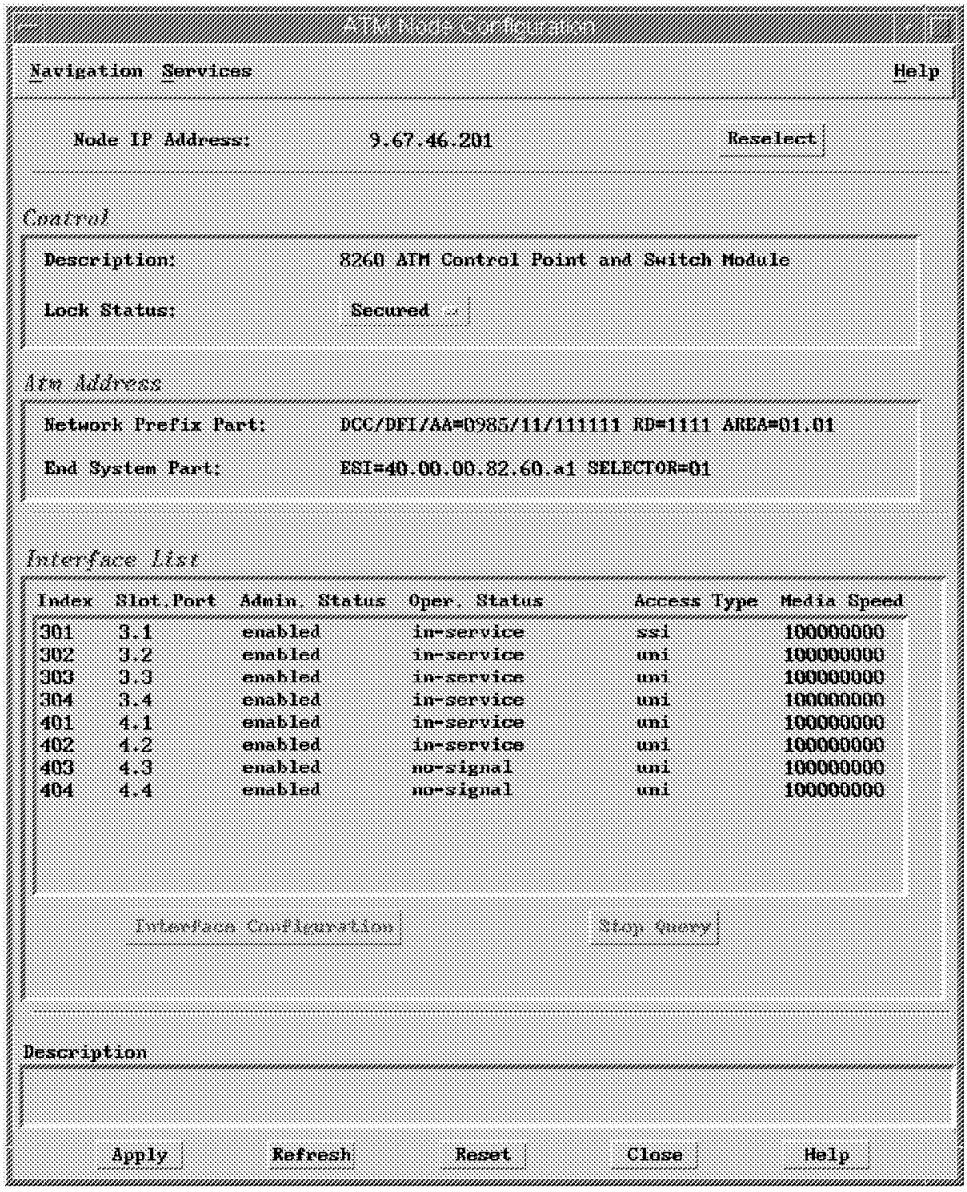



*Figure 102. Displaying 8260 Node Configuration*

#### 7.3.3.3 8260 Fault

This option allows you to display the events received from the IP address of the selected 8260.

#### 7.3.3.4 8260 Performance

This option lets you collect and view statistics about network performance on the 8260.

### 7.3.4 Displaying ATM Interface Related Information

As explained in 7.3.1.4, “ATM Node Submap” on page 219, from the ATM Node Submap you can access the following information for a desired ATM interface:

- Profile
- Configuration
- Fault

- Performance
- Device

The following sections provide a brief description of these options.

### 7.3.4.1 ATM Interface Profile

This panel displays the profile of the selected ATM interface.

### 7.3.4.2 ATM Interface Configuration

This panel, shown in Figure 103, allows you to modify the following:

- Desired state (enabled or disabled)
- ATM access type (UNI, SSI, NNI)

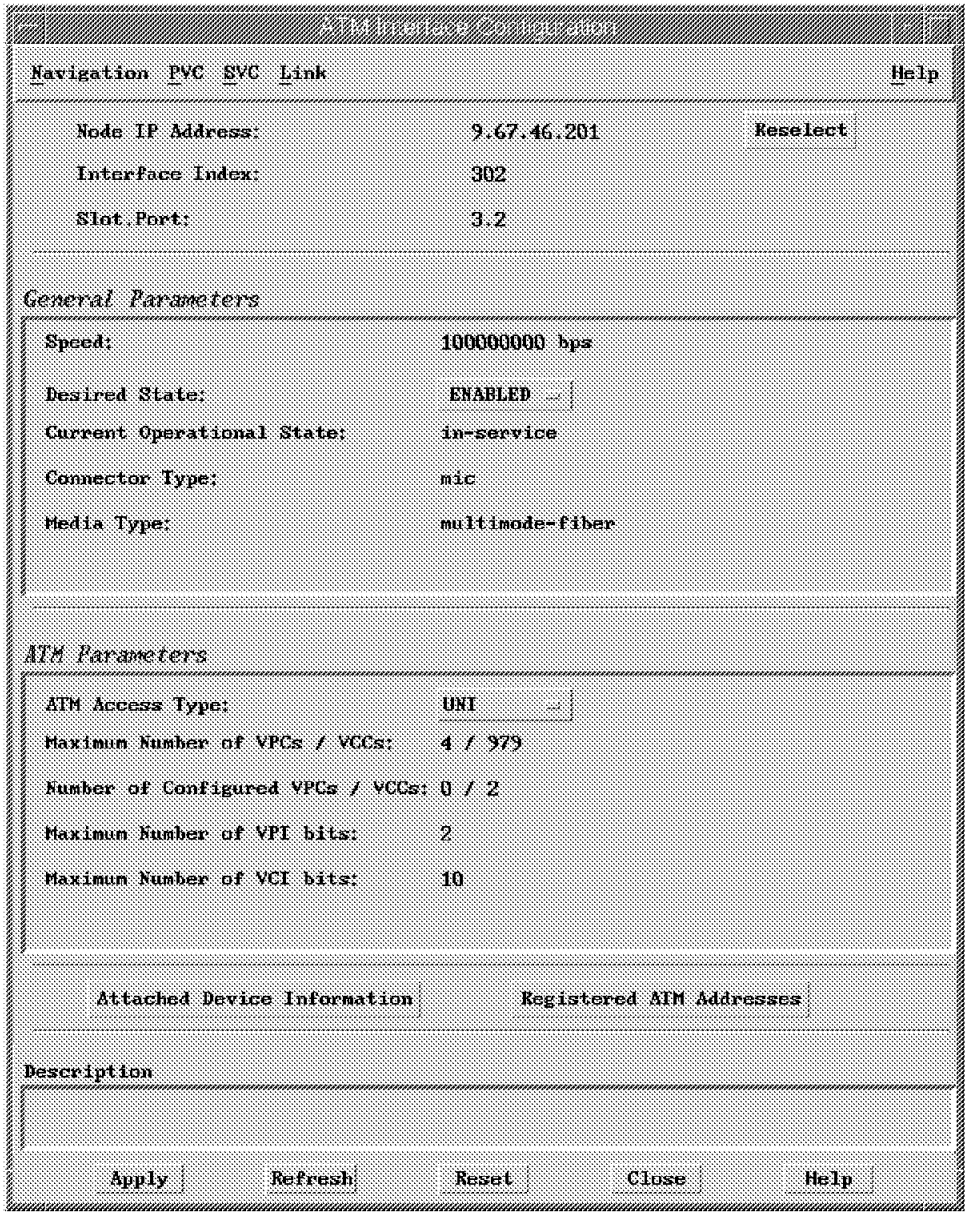



*Figure 103. Displaying ATM Interface Configuration*

The upper menu in the ATM interface configuration panel allows you to display information about the ATM connections in the selected interface (SVCs, PVCs,

and Links). These options are explained in 7.3.6, “Displaying SVCs” on page 227, 7.3.9, “Displaying PVCs” on page 232, and 7.3.11, “Displaying Links on an Interface” on page 235.

Also, from the ATM configuration panel you can get information about the ATM-attached device and its registered ATM addresses. Figure 104 shows the information you obtain by clicking on **Attached Device Information**.


*Figure 104. Displaying Interface Attached Device*

### 7.3.4.3 ATM Interface Fault

This option allows you to display the events received by the ATM node that contains this interface.

### 7.3.4.4 ATM Interface Performance

This option allows you to access panels for displaying statistics of the selected interface. Figure 105 displays a plot of the interface traffic against elapsed time.

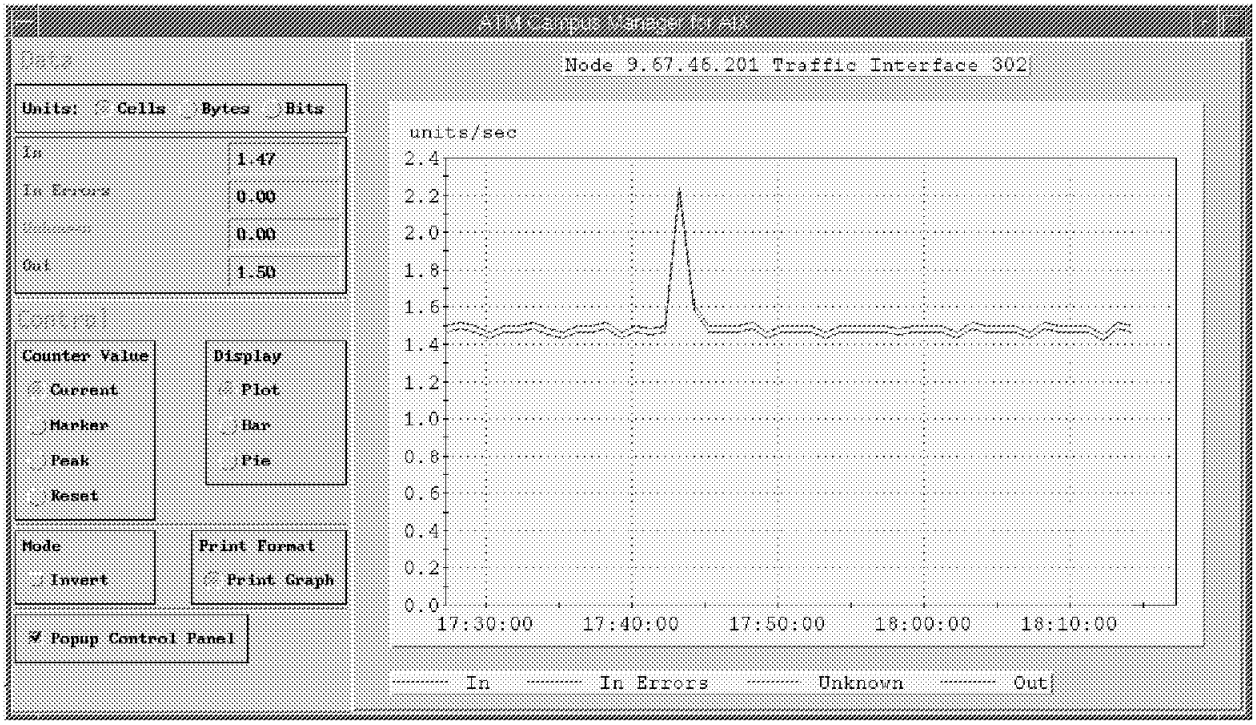



*Figure 105. Displaying ATM Interface Statistics*

## 7.3.5 Displaying Registered ATM Stations

When an ATM endsystem is connected to the ATM switch, it will use the ILMI process to acquire its ATM address and register its ATM address with the ATM switch. You may display the registered addresses on a per-UNI-interface basis.

To display the ATM registered address for an interface do the following:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on an **ATM Cluster** icon in the ATM Campus Submap.
- Double-click on an **ATM Node** icon in the ATM Cluster Submap.
- Select **ATMC-Configuration** from the menu bar. Select an interface in the interface list section of the ATM Node Configuration panel and click on the **Interface Configuration** push button.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM interface icon in the ATM Node Submap.
- Select **Registered ATM Addresses** in the ATM Interface Configuration panel. A panel similar to Figure 106 on page 227 will be displayed.

  If you have a problem in establishing a connection, we recommend you first check that all the parties involved have successfully registered with the ATM switch.

Note that when an 8282 is connected to the ATM switch, there will be one registered ATM address for the 8282 itself, as well as registered ATM addresses for each station attached to the 8282. Therefore, for such an interface, you may see up to 13 registered ATM addresses. On the other hand, for TURBOWAYS 100 and TURBOWAYS 155 adapters, which connect directly to the ATM switch, there will be a single registered ATM address.

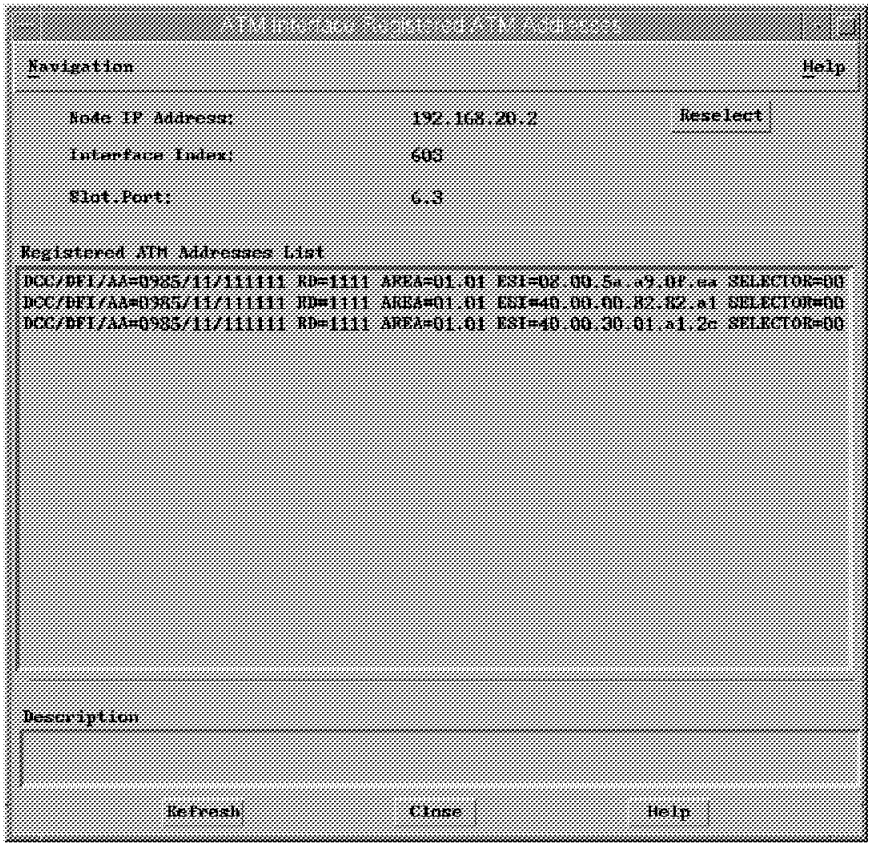



*Figure 106. Displaying Registered ATM Addresses*

## 7.3.6 Displaying SVCs

Once the endsystems are registered with the ATM switch, a point-to-point SVC is used to connect two remote devices together, whereas, a point-to-multipoint SVC is used to connect several remote endpoints (leaves) to a single endpoint (root).

The switched virtual connections (SVCs) that are currently set up in your network can be listed on a per-UNI-interface basis.

On a given interface, each SVC is uniquely identified by a unique call reference (negotiated between the ATM switch and the device that issued the CALL SETUP), virtual path identifier (VPI), and virtual channel identifier (VCI).

Note that the point-to-multipoint connections share the same call reference. However, they are listed as separate lines.

To display the ATM SVC List panel do the following:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.

- Double-click on the **ATM Node** icon in the ATM Cluster Submap.
- Select **ATMC-Configuration** from the menu bar. Select an interface in the interface list section of the ATM Node Configuration panel and click on the **Interface Configuration** push button.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM interface icon in the ATM Node Submap.
- Select **SVC-List** in the menu bar of the ATM Interface Configuration panel. A panel similar to Figure 107 will be displayed.

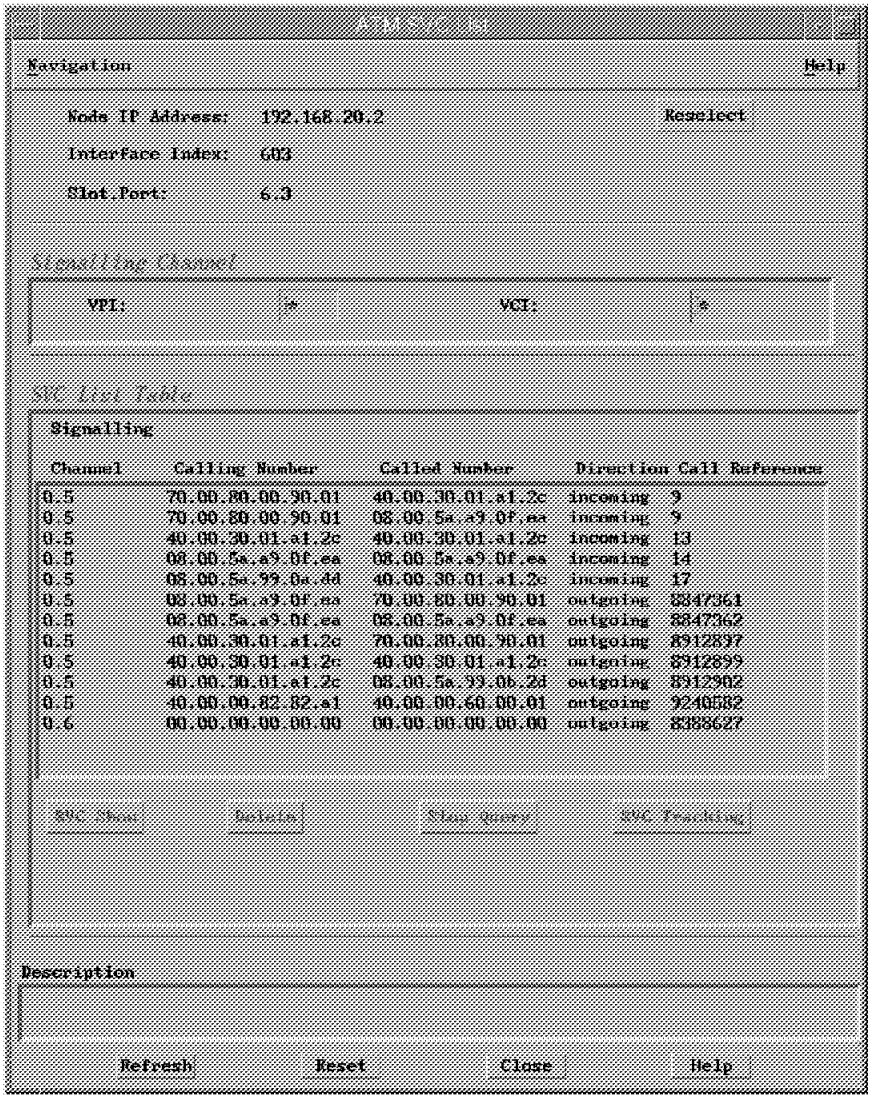



*Figure 107. Displaying Switched Virtual Circuits*

To function properly in a LAN emulation environment, every ATM station and ATM-LAN bridge must establish a Default VCC with the LE server. Additionally, they may establish Direct VCCs with each other to exchange frames.

In a Classical IP over ATM environment, every ATM station must establish a VCC with the ARP server. Additionally, they must establish a VCC with every station that they communicate with.

You may use the ability to display SVCs to do the following:

1. Verify that in a LAN emulation environment, individual endsystems have established a Default VCC with the LAN emulation server.
2. Verify that in a Classical IP over ATM environment, individual endsystems have established a VCC with the ARP server.
3. Verify that in a LAN emulation environment, the individual endstations and bridges are added to the General Multicast VCC, and the bridges are added to the Bridge Multicast VCC.
4. Determine if a Direct VCC between the stations has been established. This Direct VCC is required in a Classical IP over ATM network for the stations to be able to communicate with each other. But, the Direct VCC is optional in the IBM′s LAN emulation environment.

### 7.3.7 Displaying SVC Characteristics

You can select a given SVC from the ATM SVC List screen and select the **SVC Show** button. When you do this you will see a screen similar to Figure 108 on page 230. This screen gives you more details on an individual SVC including the QoS parameters, the VPI/VCI assigned at that interface for the SVC, as well as the calling and called numbers.

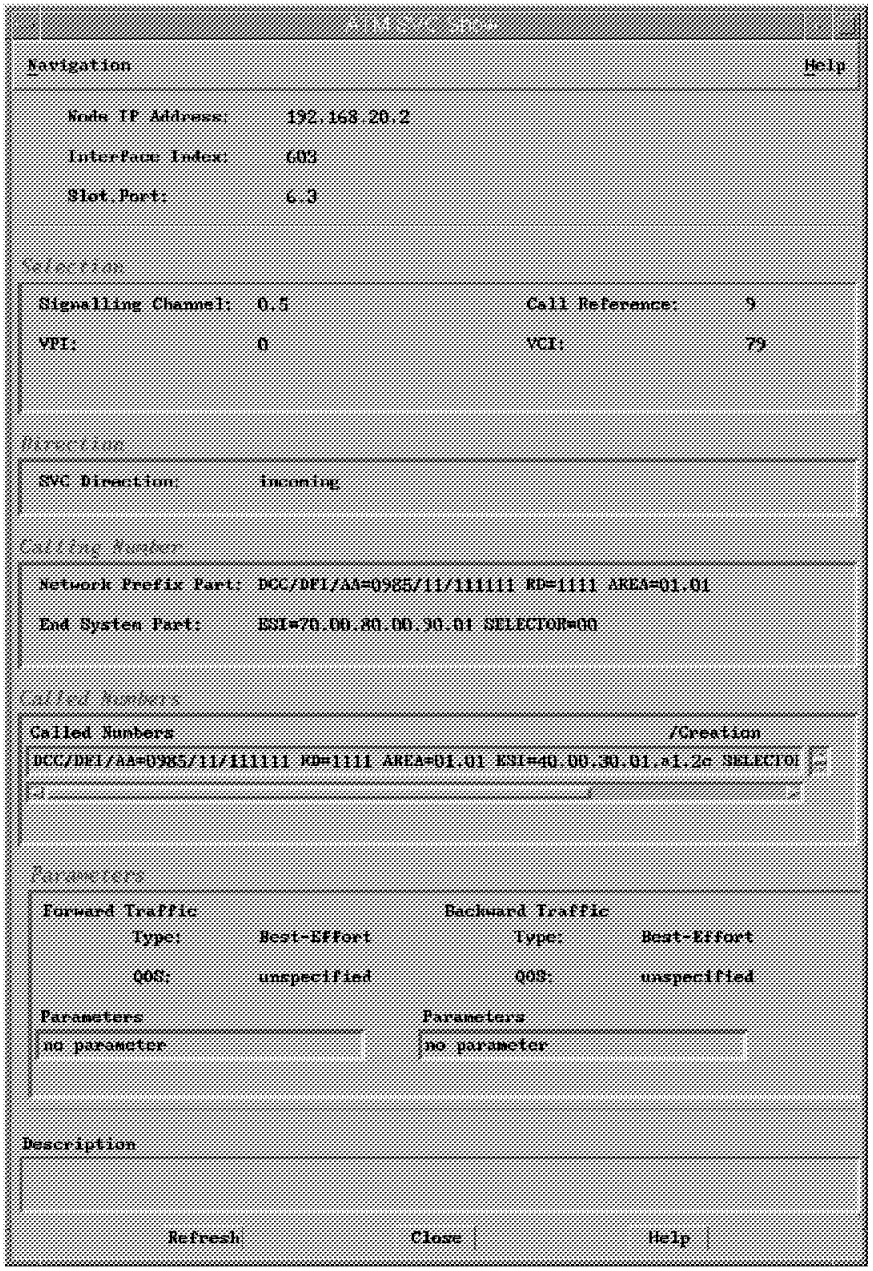



*Figure 108. Switched Virtual Circuits Characteristics*

## 7.3.8 Creating a PVC

A PVC is created between two endpoints. An endpoint is defined as:

- An ATM node
- An interface index or a slot.port number

A PVC is created by:

- Specifying a source endpoint (which will become the primary, or root, side of the PVC)
- Specifying a destination endpoint (the secondary side)
- Providing the PVC characteristics such as the bandwidth allocated, if any

The secondary side of the PVC may be connected to the same ATM switch as the primary node or to a different ATM switch. If the primary and secondary sides are not connected to the same ATM switch, there must be a physical path between the two ATM switches. Otherwise, the PVC creation will fail due to lack of a path.

**Note:** The path between the two 8260 ATM switches must be an SSI connection. Currently, the 8260 does not support PVCs over an NNI connection.

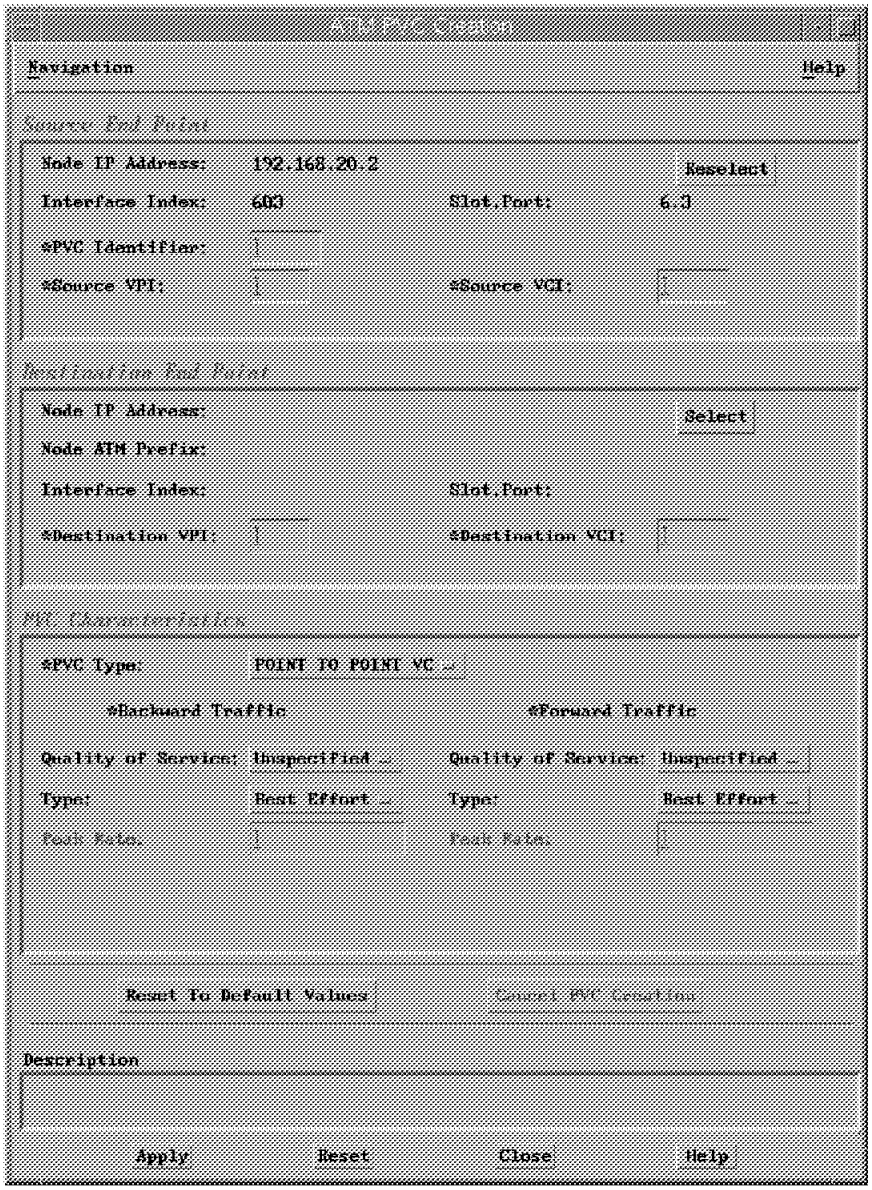



*Figure 109. Creating Permanent Virtual Circuits*

The PVC Creation panel (as shown in Figure 109) can be displayed using the following procedure:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.

- Select **ATMC-Configuration** from the menu bar and then select an interface in the interface list section of the ATM Node Configuration panel and click on the **Interface Configuration** push button.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM interface icon in the ATM Node Submap.

- Select **PVC-Create** from the menu bar of the ATM Interface Configuration panel.

  **Note:** This panel can also be used to create a point-to-point permanent virtual path (PVP).

### 7.3.9 Displaying PVCs

The permanent virtual connections (PVCs) currently defined in the IBM 8260 ATM node can be listed on a per-interface basis.

On a given interface, each PVC is uniquely defined by its PVC identifier, which is allocated when the PVC is created.

The PVC endpoint associated with the interface where the PVC was first created is the primary, or root, side. The other endpoint is the secondary (or leaf side) of the PVC.

To display the ATM PVC List panel do the following:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.
- Double-click on the **ATM Node** icon in the ATM Cluster Submap.
- Select **ATMC-Configuration** from the menu bar and then select an interface in the interface list section of the ATM Node Configuration panel and click on the **Interface Configuration** push button.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM Interface icon in the ATM Node Submap.

- Select **PVC-List** in the menu bar of the ATM Interface Configuration panel. A panel similar to Figure 110 on page 233 will be displayed, providing you with information about the PVC.

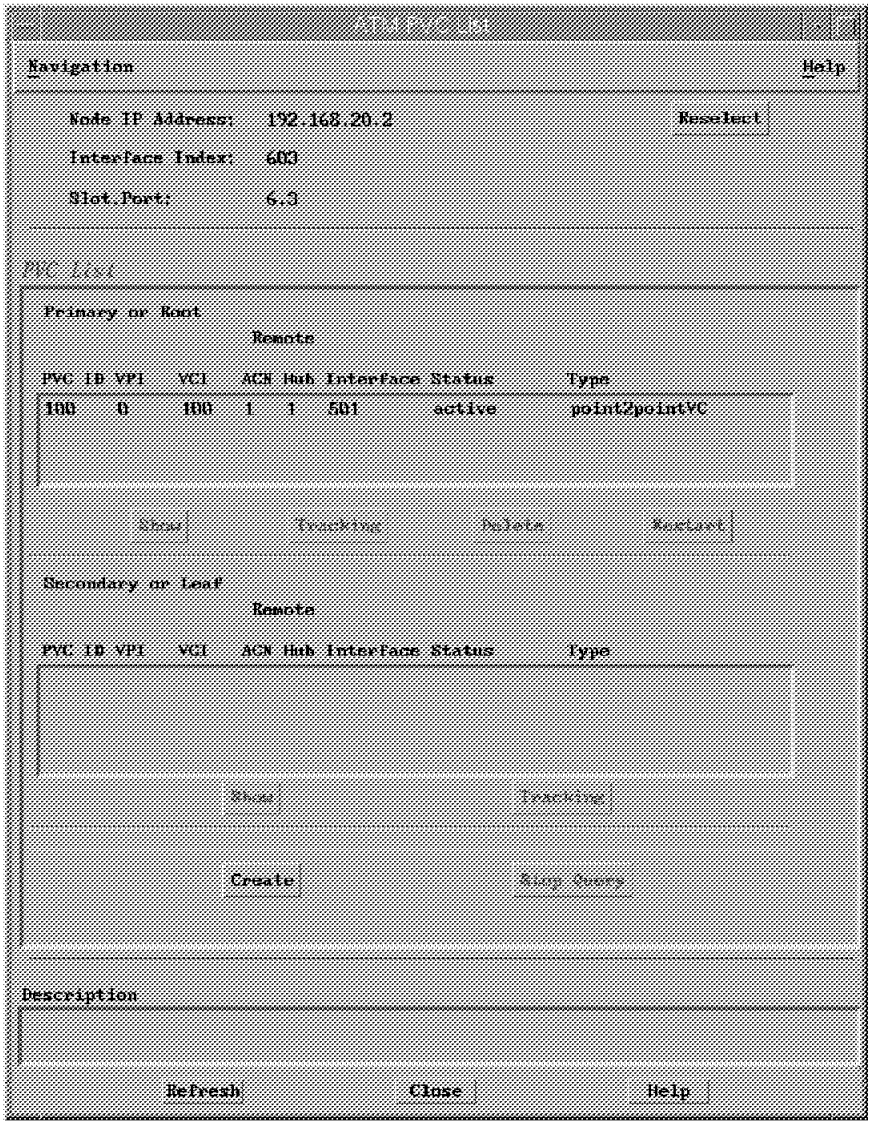



*Figure 110. Displaying Permanent Virtual Circuits*

## 7.3.10 Displaying PVC Characteristics

The characteristics of a PVC can be shown at any time, regardless of whether the PVC is active or only defined.

**Note**

The following two kinds of panels can be opened for a PVC:

- PVC Primary Show
- PVC Secondary Show

To display the ATM PVC Show panel do the following:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.
- Double-click on the **ATM Node** icon in the ATM Cluster Submap.

- Select **ATMC-Configuration** from the menu bar and then select an interface in the interface list section of the ATM Node Configuration panel and click on the **Interface Configuration** push button.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM Interface icon in the ATM Node Submap.
- Select **PVC-List** in the menu bar of the ATM Interface Configuration panel.
- Select a PVC in the PVC List Table, then click on the **Show** push button. A panel similar to Figure 111 will be displayed.

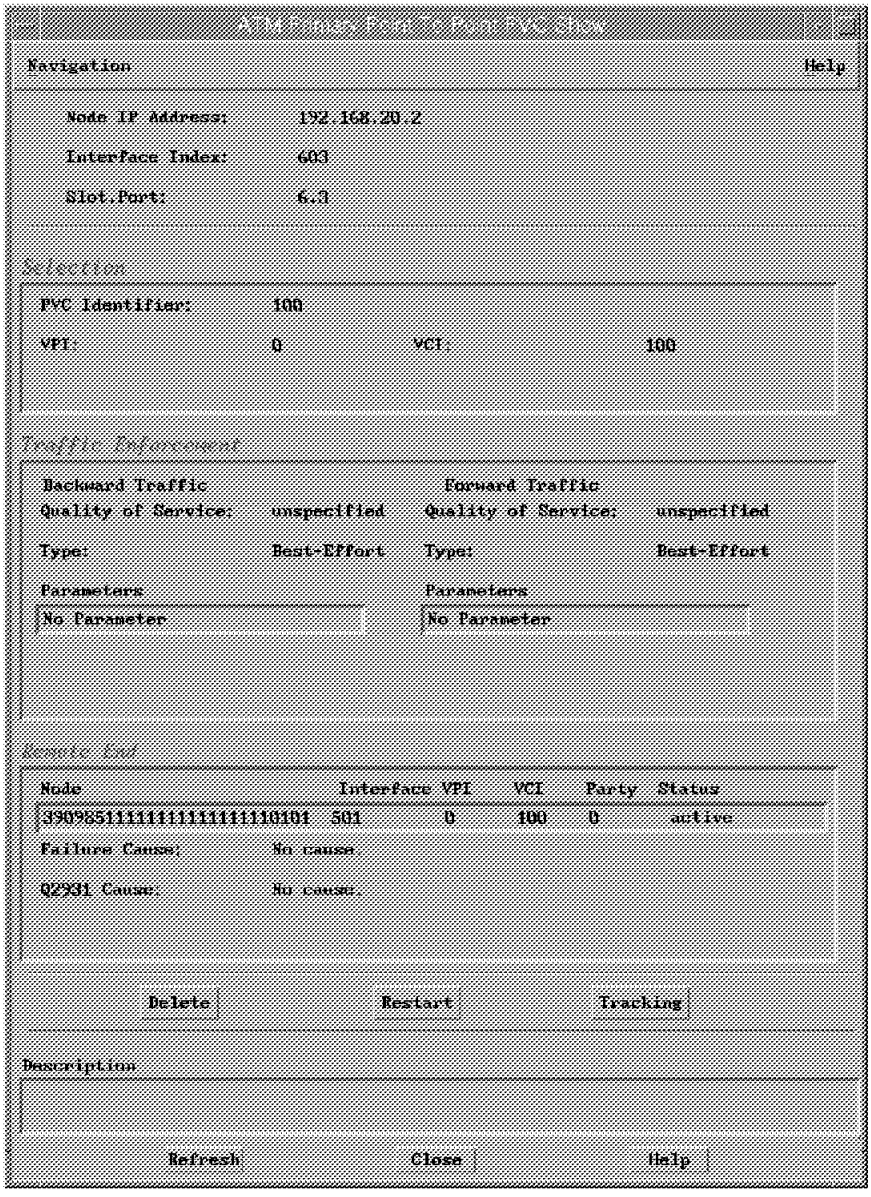



*Figure 111. Displaying Permanent Virtual Circuits*

### 7.3.11 Displaying Links on an Interface

The following are the two different types of links in an ATM network:

- Physical link

  A physical link provides the attachment between two adjacent devices. This, for example, can be the fiber link connecting two 8260 ports together, or the fiber link connecting the 8260 port to a TURBOWAYS 155 ATM adapter.

- Virtual link

  For two ATM devices to be able to communicate with each other, they need to establish a virtual channel connection (VCC) or a virtual path connection (VPC).

  A VCC is defined as a concatenation of two or more virtual channel links (VCLs) extending from one end user to another through the ATM network. The virtual channel link is, therefore, defined as a logical entity that allows the transportation of ATM cells between two adjacent ATM devices. For example, in Figure 112 on page 236, the VCC between End User A and End User B consists of the following:

  1. A VCL from End User A to ATM Switch A
  2. A VCL from ATM Switch A to ATM Switch B
  3. A VCL from ATM Switch B to End User B

  A VCL is identified by a virtual channel identifier (VCI).

  A VPC is defined as a concatenation of two or more virtual path links (VPLs) extending from one end user to another through the ATM network. The virtual path link is, therefore, defined as a logical entity that allows the transportation of ATM cells between two adjacent ATM devices. For example, in Figure 112 on page 236, the VPC between End User X and End User Y consists of the following:

  1. A VPL from End User X to ATM Switch A
  2. A VPL from ATM Switch A to ATM Switch B
  3. A VPL from ATM Switch B to End User Y

  A VPL is identified by a virtual path identifier (VPI).

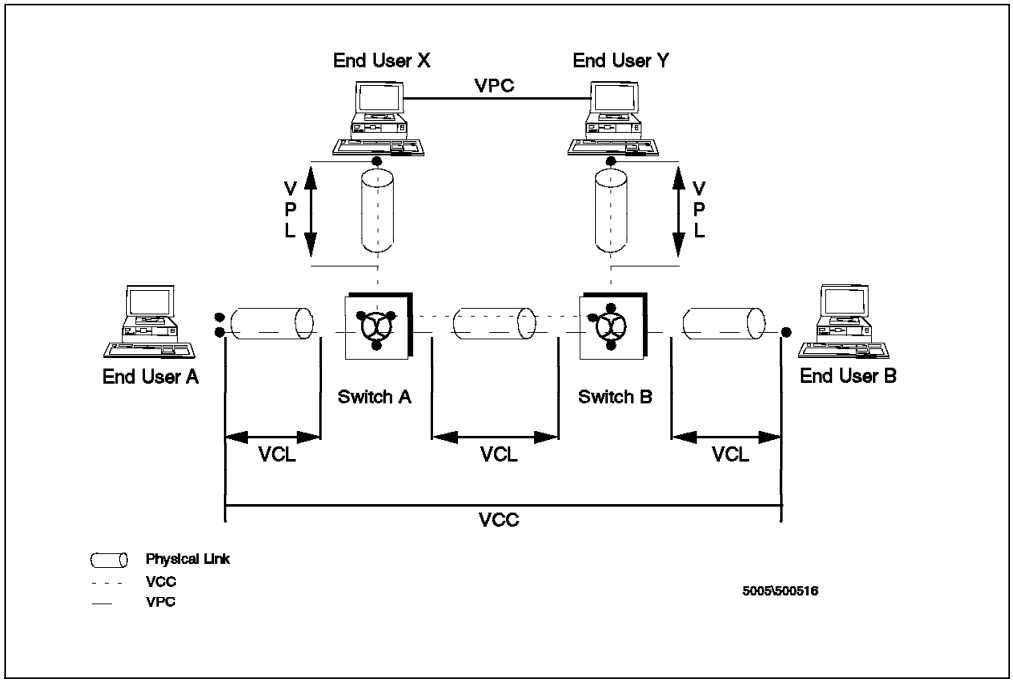



*Figure 112. Virtual Links and Virtual Connections*

ATM Campus Manager for AIX allows you to display the following type of link information, on a per-interface basis:

- Physical Link

  Provides you with the following information about the attached physical link to an 8260 port:

  – Source of the bit clock

  – Type of the scrambling used on the link

- Logical link

  Provides you with the following information about the link attached to an 8260 port:

  – ILMI parameters

  – Q.2931 parameters for the signalling channel

- Virtual links

  Provides information about the virtual channel links and virtual path links currently used on 8260 ports.

  The lists of virtual links actually shown may be filtered based on the VPI value, the VCI values, or both.

## 7.3.12 Displaying Physical Links

To display physical link information about an interface do the following:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.
- Double-click on the **ATM Node** icon the ATM Cluster Submap.

- Select **ATMC-Configuration** from the menu bar and then select an interface in the interface list section of the ATM Node Configuration panel and click on the **Interface Configuration** push button.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM interface icon in the ATM Node Submap.

- Select **Link-Physical Link** in the menu bar of the ATM Interface Configuration panel. A panel similar to Figure 113 will be displayed.

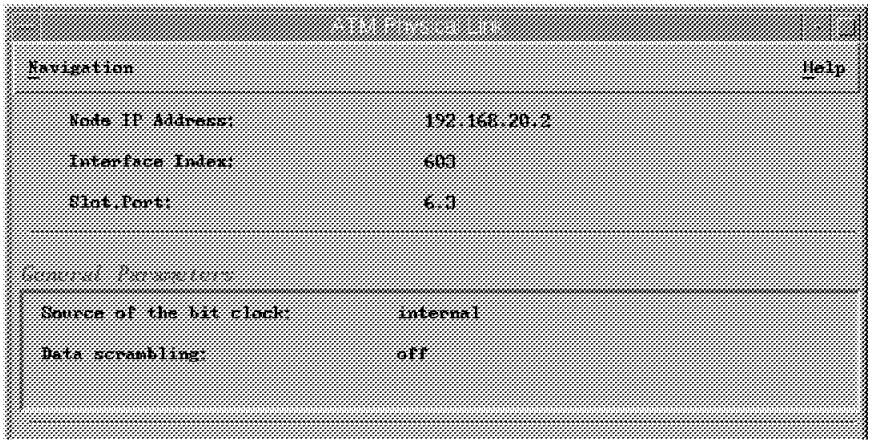



*Figure 113. Displaying Physical Links*

### 7.3.13 Displaying Logical Links

To display logical link information about an interface do the following:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.
- Double-click on the **ATM Node** icon in the ATM Cluster Submap.
- Select **ATMC-Configuration** from the menu bar and then select an interface in the interface list section of the ATM Node Configuration panel and click on the **Interface Configuration** push button.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM Interface icon in the ATM Node Submap.

- Select **Link-Logical Link** in the menu bar of the ATM Interface Configuration panel. A panel similar to Figure 114 on page 238 will be displayed.

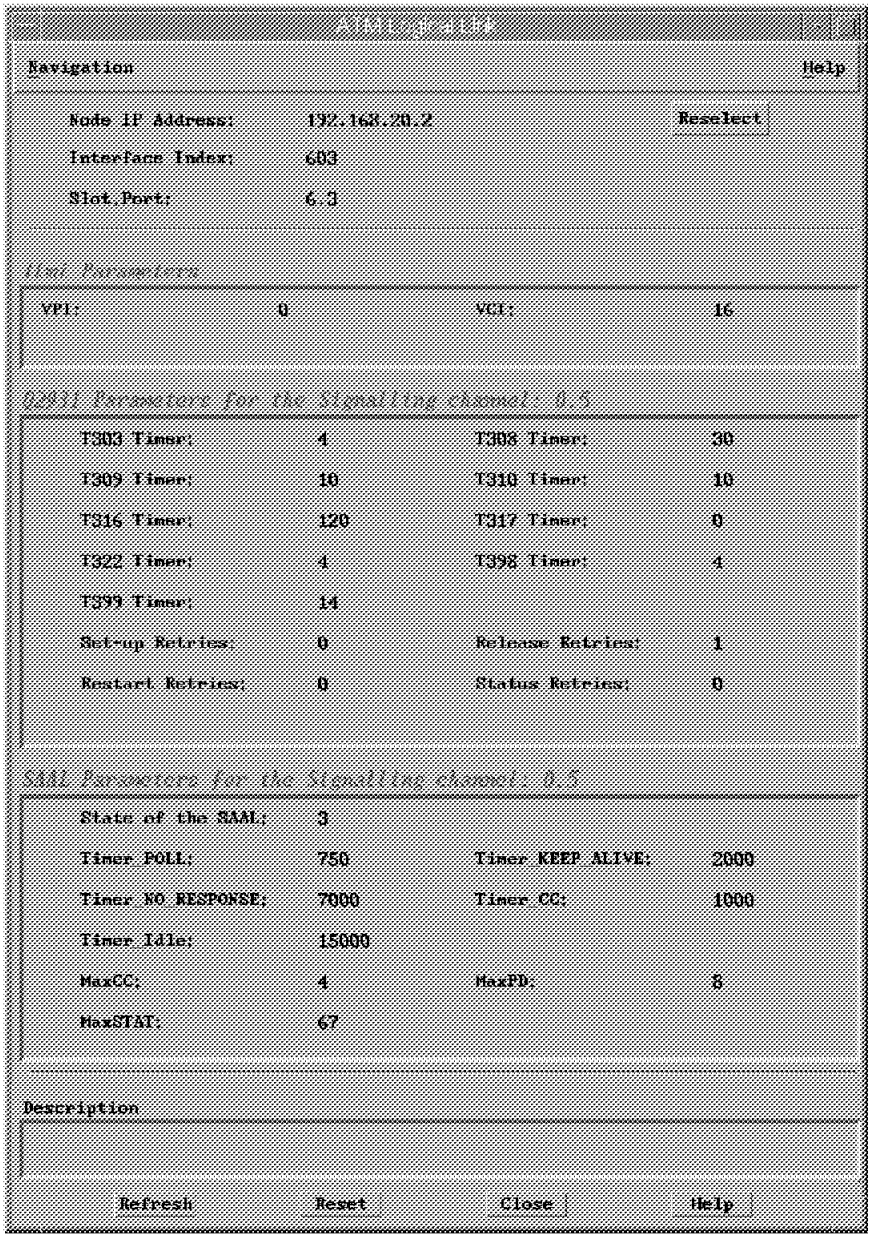



*Figure 114. Displaying Logical Links*

## 7.3.14 Displaying Virtual Links List

To display the list of virtual links used on an 8260 port do the following:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on an **ATM Cluster** icon in the ATM Campus Submap.
- Double-click on an **ATM Node** icon in the ATM Cluster Submap.
- Select **ATMC-Configuration** from the menu bar and then select an interface in the interface list section of the ATM Node Configuration panel and click on the **Interface Configuration** push button.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM Interface icon in the ATM Node Submap.

- Select **Link-Virtual Link List** in the menu bar of the ATM Interface Configuration panel. A panel similar to Figure 115 on page 239 will be displayed.

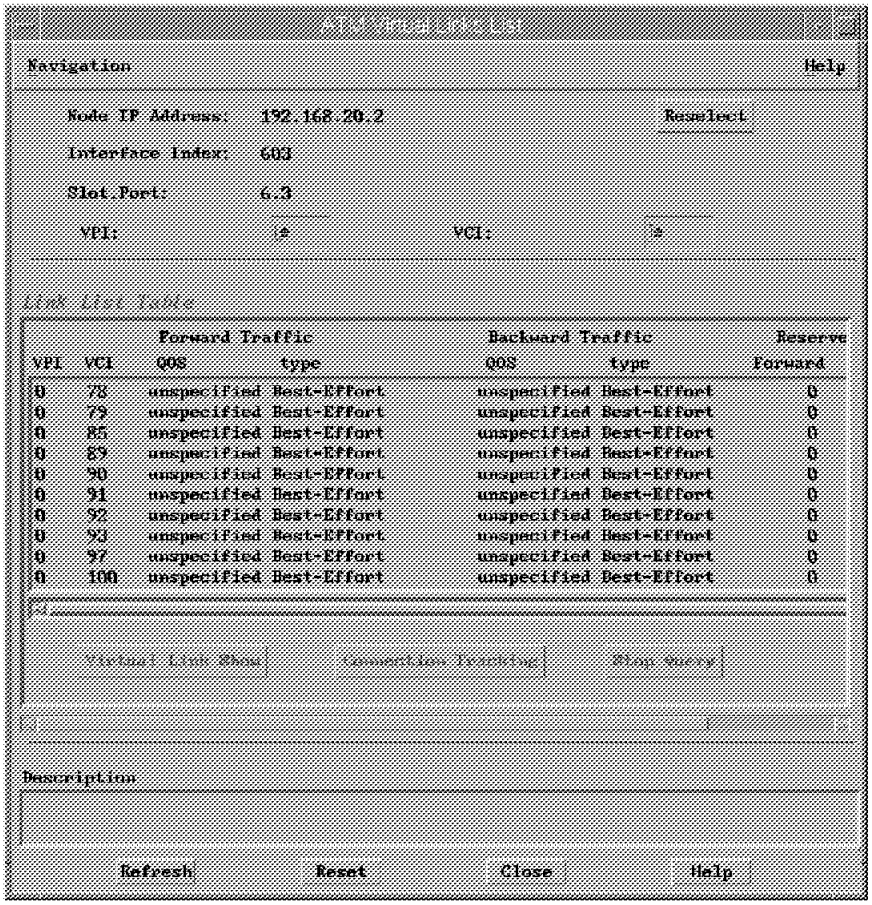



*Figure 115. Displaying Logical Links*

## 7.3.15 Displaying Characteristics of a Virtual Link

To display the ATM Virtual Links Detail panel do the following:

- Select a virtual link from the ATM Virtual Link List and then click on the **Virtual Link Show** button. A panel similar to Figure 116 on page 240 will be displayed.

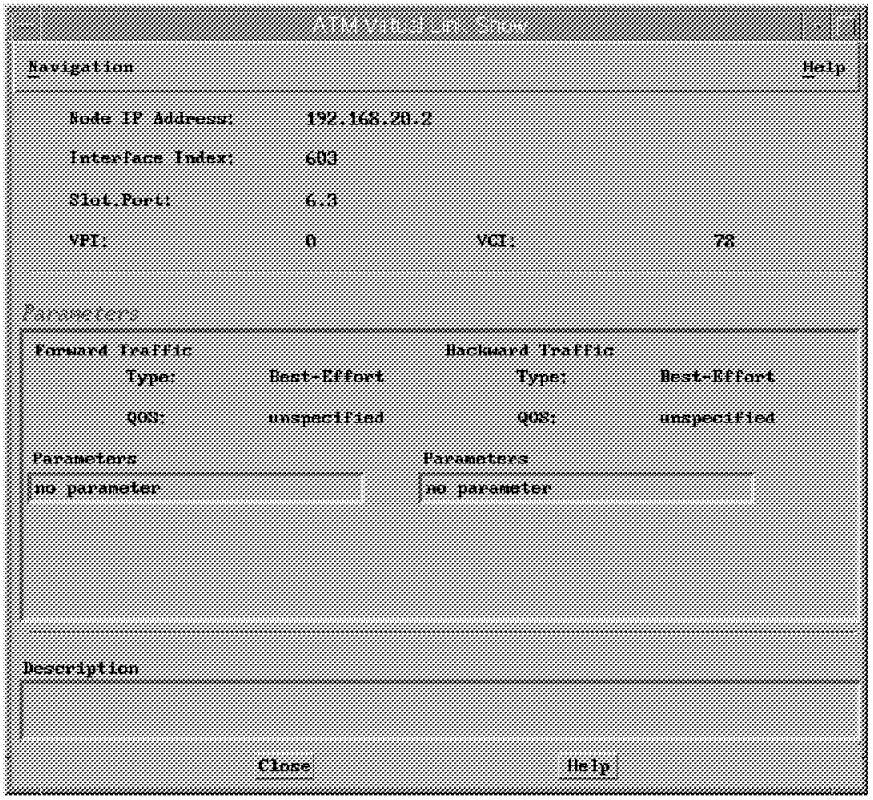



*Figure 116. Displaying Virtual Link Characteristics*

## 7.3.16 Tracking Connections

Tracking a connection in the network consists of identifying the endpoints of a connection and the intermediate nodes used by the connection.

To track a connection, the ATM Campus Manager must be given a connection identifier. The connection may be a PVC or an SVC, and the identifier will be the following:

- For an SVC - An IBM 8260 node, an interface on this node, a signalling channel and a call reference number.
- For a PVC - An IBM 8260 node, an interface on this node and a PVC identifier.

## 7.3.17 Tracking an SVC

When one of the endpoints of an SVC is known, that is selected in the SVC List section, it is possible to know the other endpoint and all intermediate nodes used by this SVC.

In the case of a point-to-multipoint SVC, the root of the SVC and all the leaves along all the intermediate nodes are found.

An SVC is tracked from the SVC List panel as follows:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.
- Double-click on the **ATM Node** icon the ATM Cluster Submap.

- Select **ATMC-Configuration** from the menu bar and then select an interface in the interface list section of the ATM Node Configuration panel and click on the **Interface Configuration** push button.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM Interface icon in the ATM Node Submap.
- Select **SVC-List** in the menu bar of the ATM Interface Configuration panel.
- Select an SVC in the SVC List Table. Then click on the **SVC Tracking** push button. A panel similar to Figure 117 will be displayed.

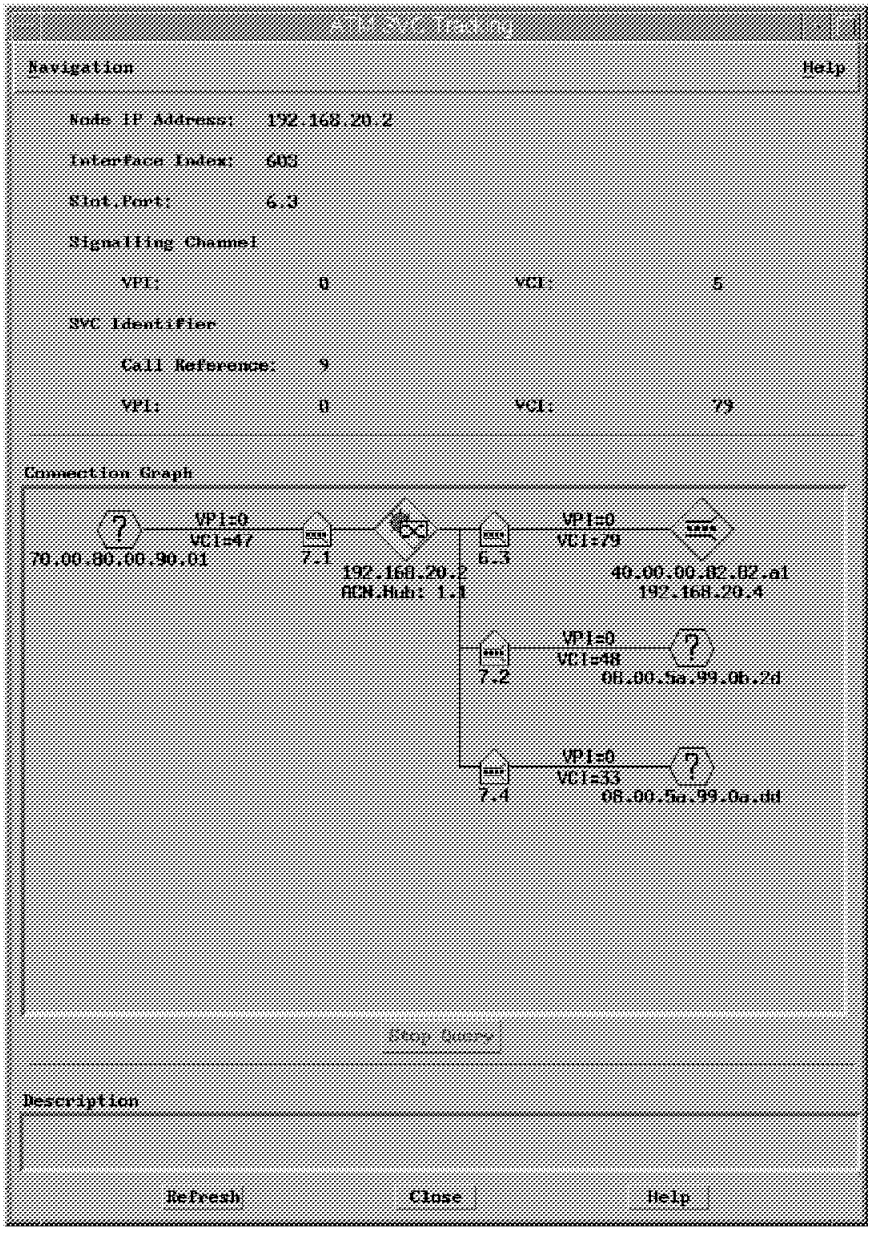



*Figure 117. Tracking Switched Virtual Connections*

### 7.3.18 Tracking a PVC

A PVC is tracked from the PVC List panel as follows:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.
- Double-click on the **ATM Node** icon in the ATM Cluster Submap.
- Select **ATMC-Configuration** from the menu bar and then select an interface in the interface list section of the ATM Node Configuration panel and click on the **Interface Configuration** push button.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM Interface icon in the ATM Node Submap.
- Select a PVC in the PVC List Table. Then click on the **PVC Tracking** push button. A panel similar to Figure 118 will be displayed.

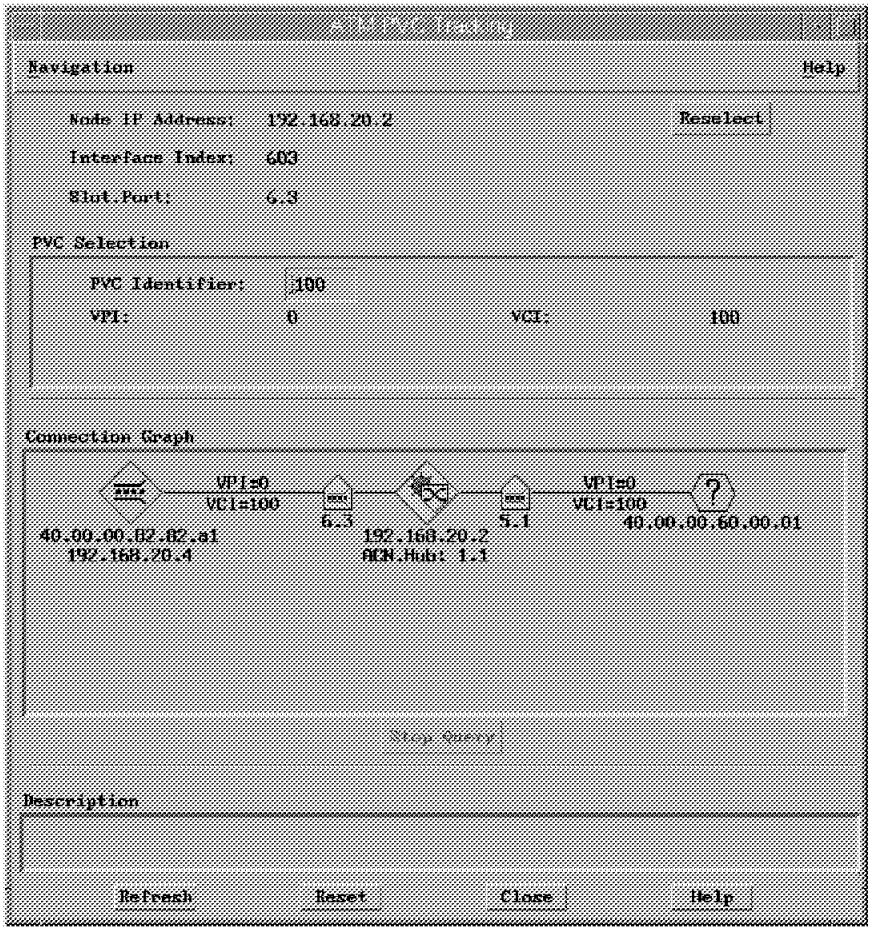



*Figure 118. Tracking Permanent Virtual Connections*

### 7.3.19 Tracking a Virtual Connection

When one virtual link is known, that is, selected in the Virtual Link section, it is possible to know the endpoints and all intermediate nodes used by this connection.

If this virtual link belongs to a point-to-multipoint connection, the root of the connection and all the leaves along with the intermediate nodes are found.

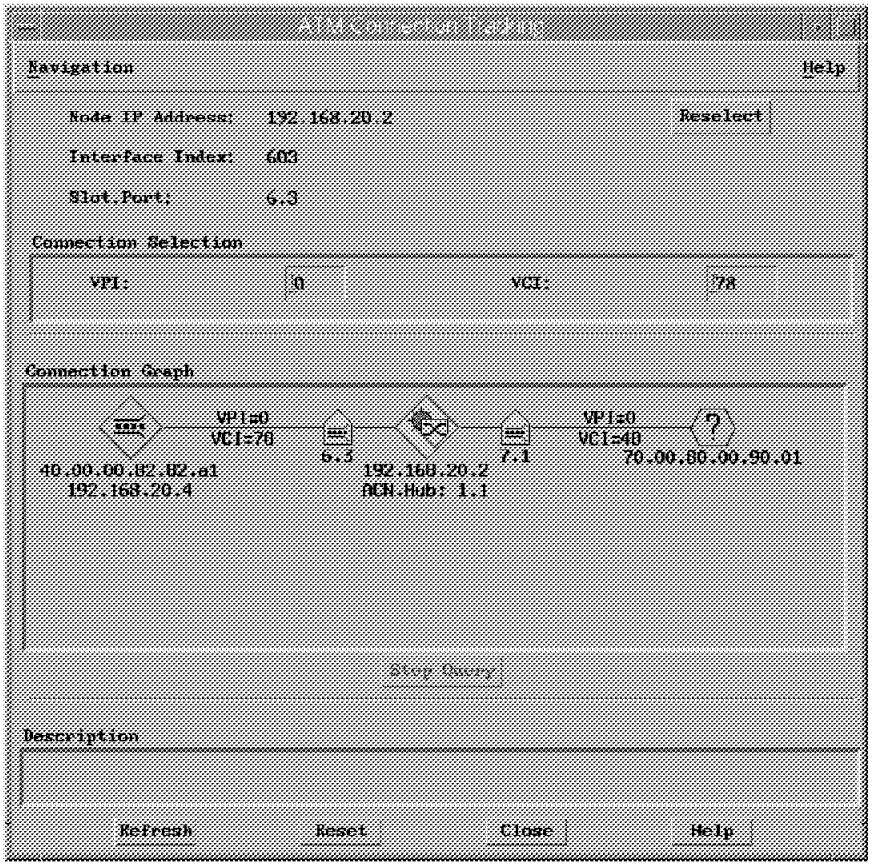



*Figure 119. Tracking Virtual Connections*

To display the Connection Tracking panel do the following:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon the ATM Campus Submap.
- Double-click on the **ATM Node** icon the ATM Cluster Submap.
- Select **ATMC-Configuration** from the menu bar and then select an interface in the interface list section of the ATM Node Configuration panel and click on the **Interface Configuration** push button.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM Interface icon in the ATM Node Submap.
- Select **Link-List** in the menu bar of the ATM Interface Configuration panel.
- Select a line in the Link List Table section of the ATM Virtual Links List panel. Then click on the **Connection Tracking** push button. A panel similar to Figure 119 will be displayed.

## 7.4 Managing Faults Using ATMC

The IBM 8260 provides you with the ability to enable recording of system traces and topology and route service (TRS) traces. When a trace is active, the messages are stored in a flat ASCII file in the IBM 8260 Control Point and Switch module.

Additionally, the 8260 allows you to take a *TRS dump* (a dump of the topology and route services microcode in the CPSW). When a dump is taken, the requested data is stored in a flat ASCII file in the 8260 CPSW module.

The 8260 stores the information associated with any errors in the 8260 in an *error log*. The error log is an ASCII file that is stored in the 8260 CPSW.

The trace, dump and error log files can be retrieved from the CPSW module, using the trivial file transfer protocol (TFTP), for further analysis.

**Note:** The dumps, traces and error log can be used by the IBM service personnel when dealing with errors associated with the 8260.

The 8260 ASCII console provides you with the ability to enable/disable tracing and dumping. It also allows you to transfer the resulting files, as well as the error log, to another station using trivial file transfer.

ATM Campus Manager also provides facilities for you to enable/disable recoding of traces in the 8260 and transfer them using TFTP.

### 7.4.1 Trace and Dump Recoding

To enable/disable traces and dump recoding do the following:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.
- Select **ATMC-Configuration** from the menu bar.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM Node icon in the ATM Cluster Submap.
- Select **Services-Traces and Dumps** from the menu bar of the ATM Node Configuration panel. The resulting panel, as shown in Figure 120 on page 245, can be used to start/stop traces and dumps.

  Traces can be started by setting the value of the System and Topology and Route Service fields to On and then selecting the **Apply** button.

  A dump is taken when the Topology and Route Service field is set to start and the **Apply** button is selected.

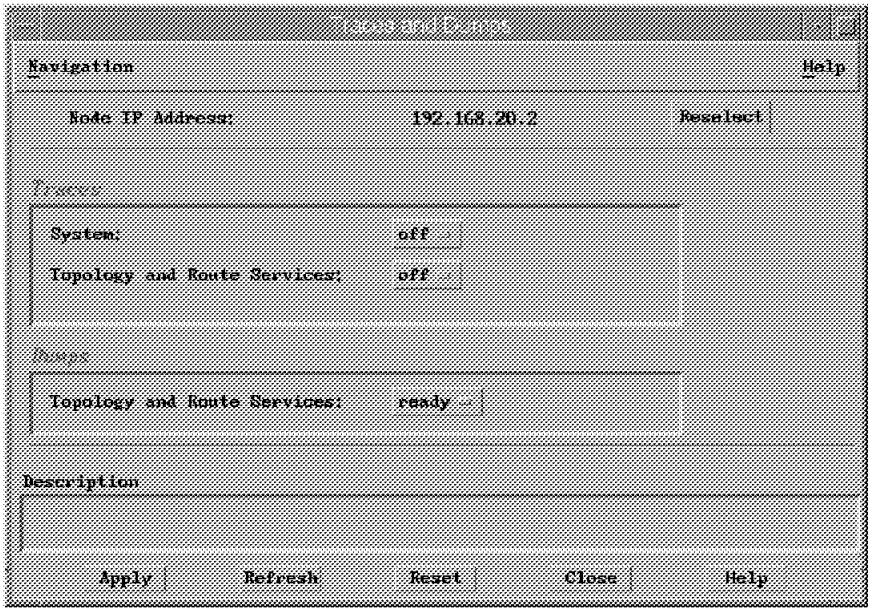



*Figure 120. Traces and Dumps Panel*

### 7.4.2 Transferring Files

To initiate TFTP to transfer the trace, dump and error log to a workstation do the following:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.
- Select **ATMC-Configuration** from the menu bar.

  Alternatively, you may select **Configuration** from the context menu displayed when you click mouse button 3 on the ATM Node icon in the ATM Cluster Submap.
- Select **Services-File Transfer** from the menu bar of the ATM Node Configuration panel.

  The resulting panel, as shown in Figure 121 on page 246, can be used to initiate the file transfer:

  – Set the Action field to Upload.

  – Set the Server IP Address field to the IP address of the workstation to which the files are transferred.

  – Set the File Name field to the directory path and the file name used in the workstation.

  – Set the File Type field to the type of file being uploaded. The following is the list of file types:

    - errorlog
    - systemTrace
    - trsTrace
    - trsDump

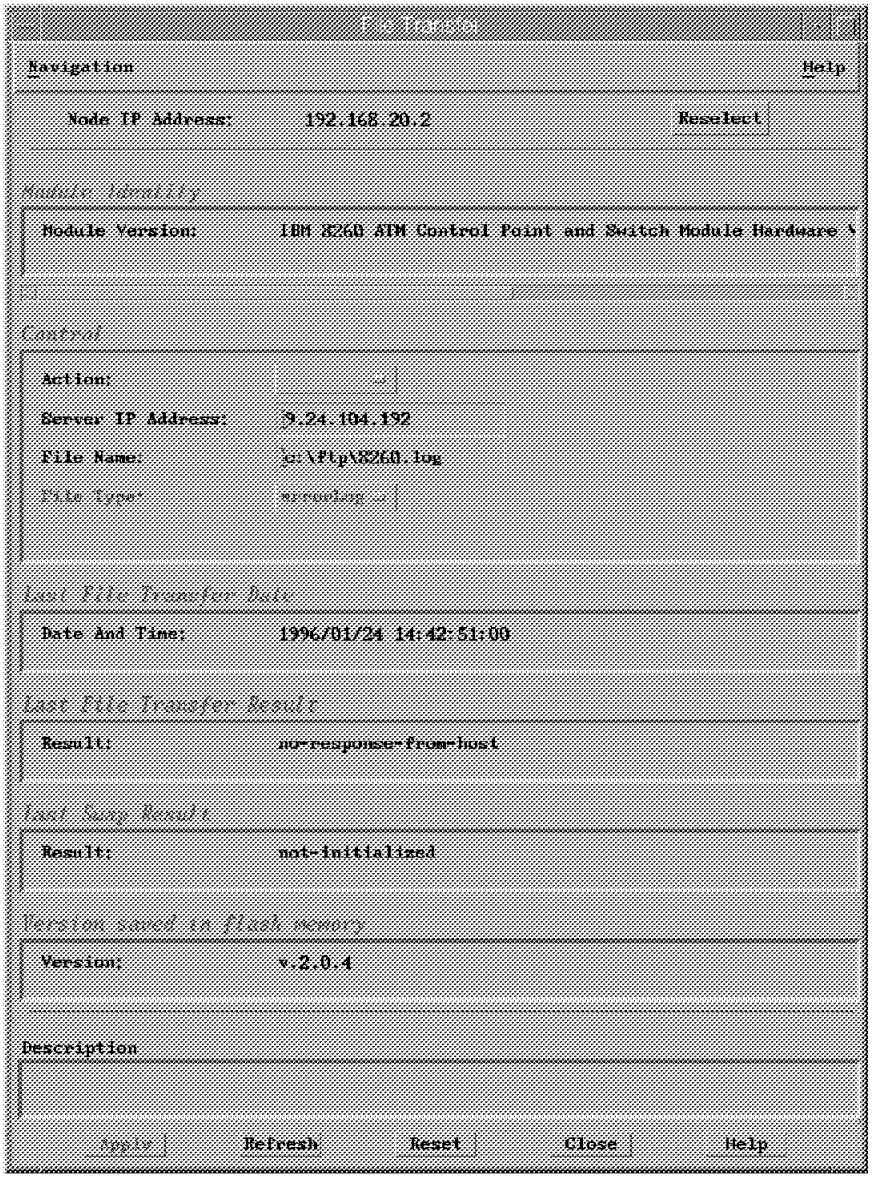



*Figure 121. File Transfer*

## 7.5 Monitoring Resources Using ATMC

The resources and the type of data that can be monitored using ATM Campus Manager for AIX are as follows:

- The 8260 ATM Switch Node
  - Logged call

    The IBM 8260 keeps a log of the following information for each SVC:
    - Internal call index
    - Interface number
    - Calling number
    - Called number

- Creation date
- Clear time
- Clear cause

- ATM Port on the 8260
  - Traffic

    Provides you with the counters for the traffic sent and received on each port. The following counters can be measured for each port on the 8260:

    - Received cells
    - Received cells in error
    - Unknown received cells
    - Transmitted cells
  - Bandwidth

    Provides you with the information about the maximum supported bandwidth and the bandwidth currently in use for each port.
  - Q.2931 Status

    Provides you with the following information:

    - Incoming calls in progress
    - Outgoing calls in progress
  - SAAL errors

    Provides you with the information about the errors detected by the Signalling Adaptation Layer (SAAL).

The following sections provide you with information on how to display the above information.

### 7.5.1 ATM Monitor

The Monitor panel of the ATM Campus Manager for AIX allows you to display an overview of the load on the ATM switch and on selected ATM ports. To access the ATM Monitor panel (as shown in Figure 122 on page 248) do the following:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.
- Select **ATMC-Monitor** from the menu bar or the context menu that is displayed when you click mouse button 3 on an ATM node icon in the ATM Cluster Submap. A panel similar to Figure 122 on page 248 will be displayed. The selection area on this panel allows you to choose one of the following three modes:
  - Top 5 In Traffic

    This mode results in the total load on the ATM switch to be monitored, as well as the five ports that receive the highest amount of traffic.
  - Top 5 Out Traffic

    This mode results in the total load on the ATM switch to be monitored, as well as the five ports that transmit the highest amount of traffic.

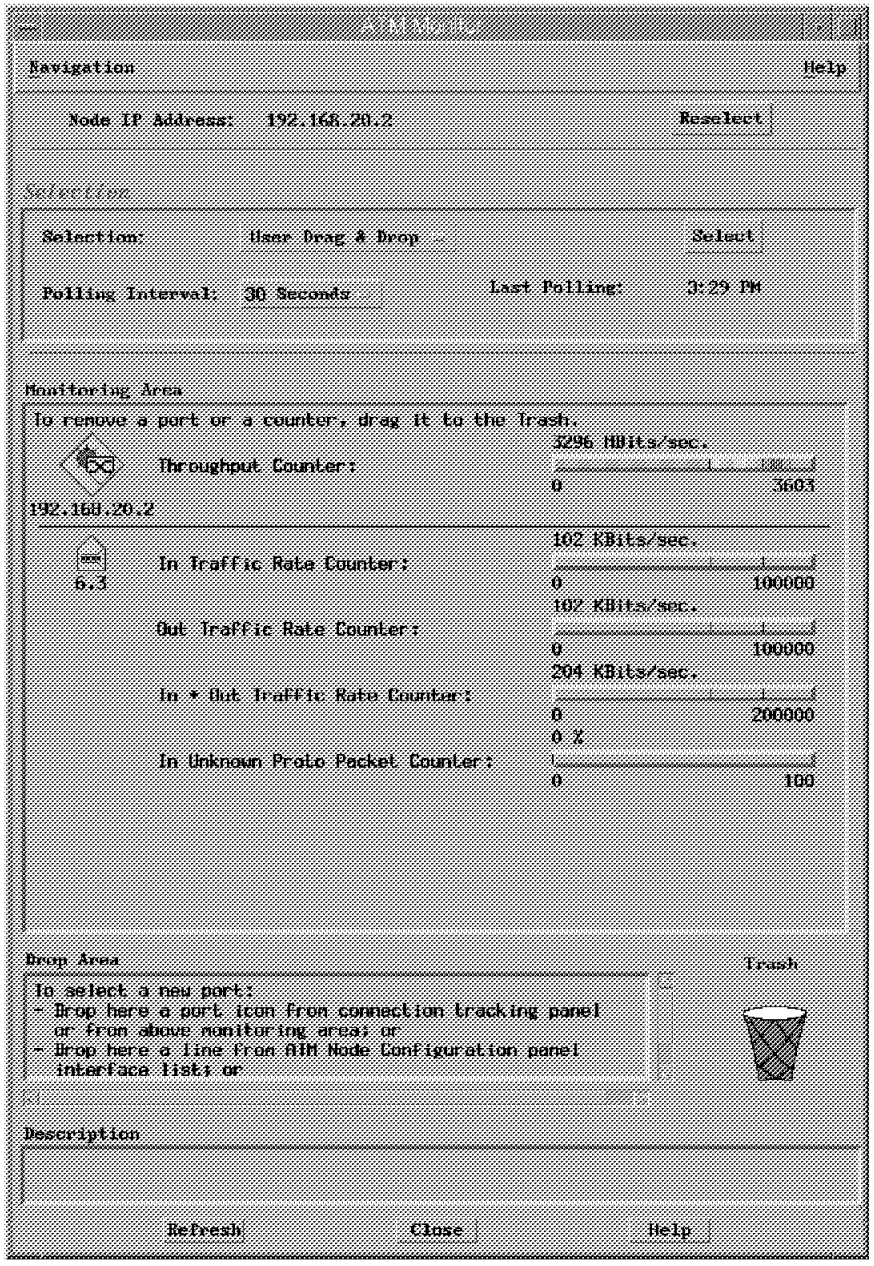



*Figure 122. ATM Performance Control*

– User Drag and Drop

  This mode automatically enables the monitoring of the total load on the ATM switch. Additionally, it allows you to choose which ports are to be monitored and which counters are to be displayed by using a drag and drop function.

  To use the drag and drop function, press mouse button 2 when the cursor is over a port listed in ATM Node Configuration panel and drag the selected ATM port into the Drop Area at the bottom of the ATM Monitor panel. This will result in the selected port and available counters for it to be displayed in the Drop Area. Once a port is in the Drop Area, you can start monitoring various counters by dragging the counters (using mouse button 2) into the Monitoring Area. When you no longer wish to monitor a port or a counter on that port, drag the port or

the counter (using mouse button 2) into the Trash Area to stop the monitoring.

### 7.5.2 ATM Performance Control

The ATM Campus Manager performance function displays statistical information for selected ATM resources.

The resources and categories currently defined for monitoring must be listed in the ATM Performance Control panel which can be accessed as follows:

- Double-click on the **ATM Campus** icon in the Root Submap.
- Double-click on the **ATM Cluster** icon in the ATM Campus Submap.
- Select the ATM switch or the concentrator to be monitored from the ATM Campus Cluster Submap and then select **ATMC-Performances** from the menu bar.

  Alternatively, you may select **Performance** from the context menu displayed when you click mouse button 3 on the ATM Switch or ATM Concentrator icon in the ATM Cluster Submap. A panel similar to Figure 123 on page 250 is displayed.

The ATM Performance Control panel allows you to select a resource and category shown in the Selection area. Note that the resource selection section allows you to select the node or the interface to be monitored. The category section contains a list of MIB variables grouped by similarity. The categories permitted for each resource type are the following:

- Node
  - Call logging
- Interface
  - Traffic
  - Bandwidth
  - Q.2931_Errors
  - Q.2931_Calls
  - SAAL

The following sections provide a brief overview of how this information may be used to manage the campus ATM network.

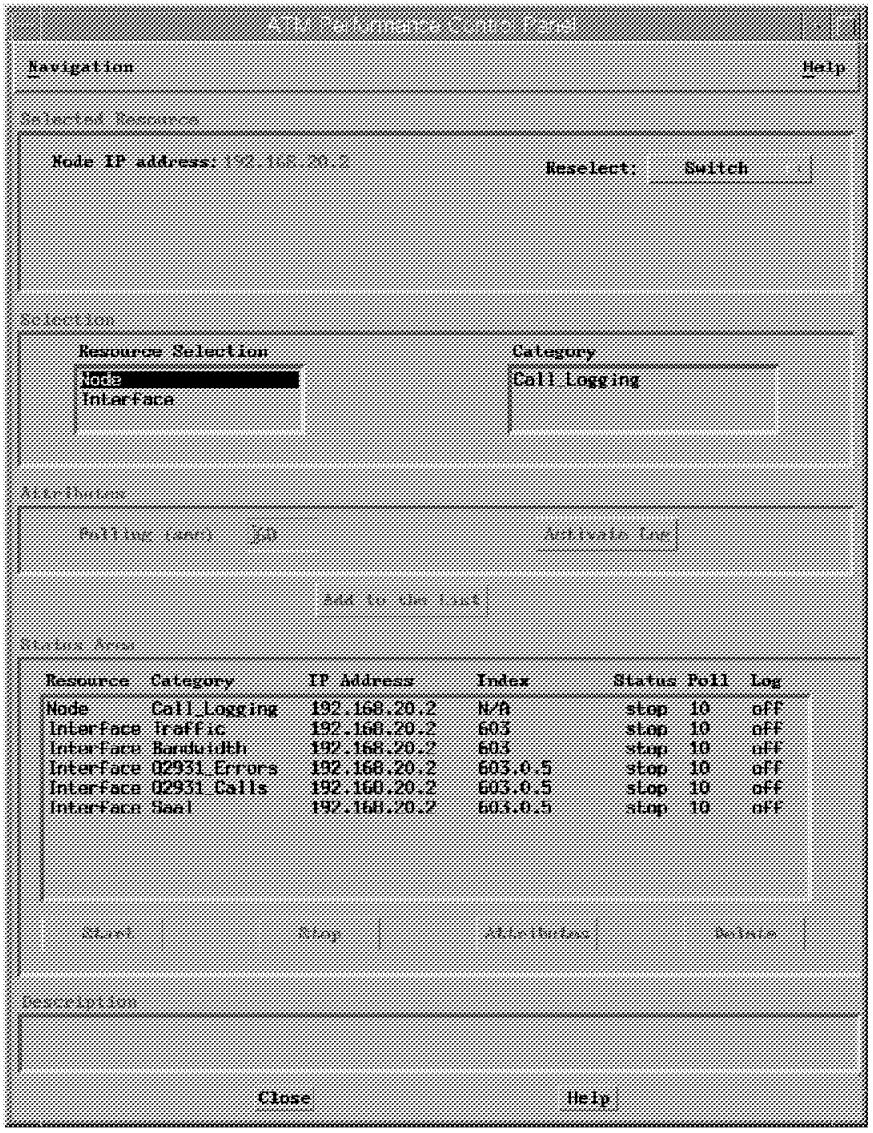



*Figure 123. ATM Performance Control*

## 7.5.3 Graphing Traffic

One of the more interesting things that you can do from the Control panel is to graph the traffic or bandwidth for any given interface (slot.port) within an individual node. The graph can be displayed in units of cells, bytes or bits. Figure 124 on page 251 and Figure 125 on page 251 provide an example of the information displayed.

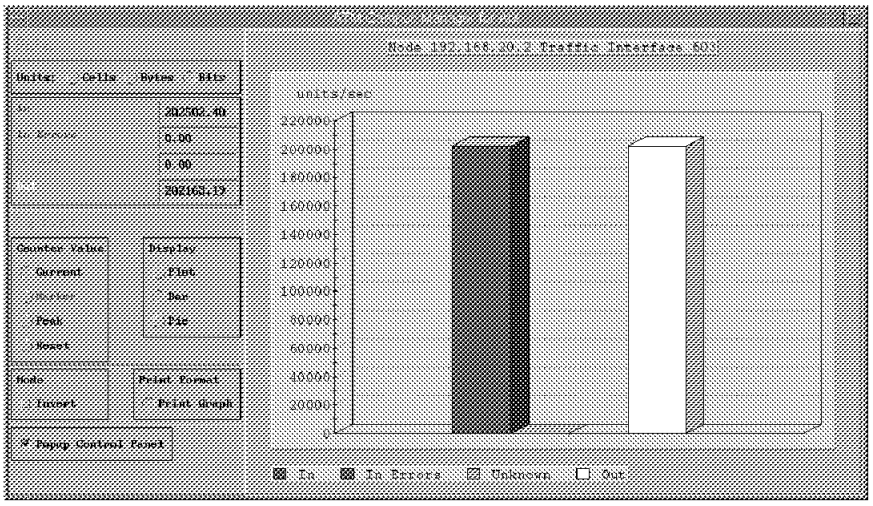



*Figure 124. Graphing Traffic*

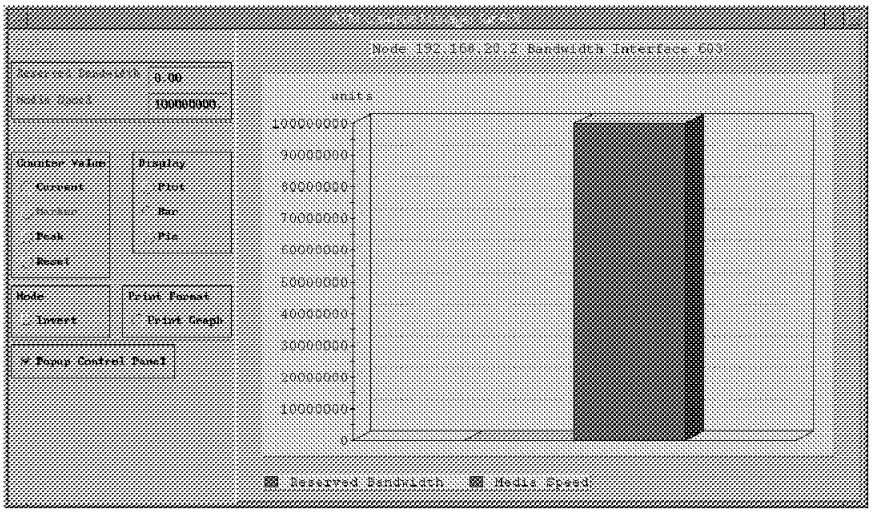



*Figure 125. Graphing Reserved Bandwidth*

## 7.5.4 Tracking Logged Calls

In an ATM environment you will be called upon to do problem determination on why two stations cannot establish a connection with each other. One of the tools that you could use for that purpose is the Call Logging function of the Performances selection. This allows you to see all calls that have have been terminated.

The calls are date/time stamped and have a cause code associated with each termination. You will also see two fields displayed, the calling number and the called number. This represents the endsystem ID of the ATM device placing the call and the endsystem ID of the ATM device that is being called. You will also see the cause. By selecting a particular call and selecting the **Details** button, you will see more information on the terminated call.

In the Logged Call Details panel, the full 20-byte ATM address of both the calling and called device is displayed, as well as the cause field with an English

explanation. You will also notice that any quality of service QoS) parameters used for the call are also listed.

When the ATM Cause Code explanation is not clear enough, you can refer to Annex E of the UNI V3.0 or V3.1 specifications.

Figure 126 shows an example of a call logging detail panel.


*Figure 126. Graphing Reserved Bandwidth*

# Appendix A. Introduction to ATM and Campus ATM Terminology

This appendix provides a brief introdunction to ATM and campus ATM terminology.

## A.1 ATM and the B-ISDN Protocol Reference Model

Broadband Integrated Services Digital Network (B-ISDN) is the technology that is being developed to give users a single service to support voice, video and data.

ATM was chosen as the transfer mode technology to deliver B-ISDN. Consequently you may see the term B-ISDN and ATM used interchangeably.

Part of the initial ITU-T recommendation on B-ISDN included the B-ISDN Protocol Reference Model, which specifies a layered architecture that defines basic principles and characteristics of B-ISDN. The model also defines a series of planes. These are referred to in ATM standards (for example, those produced by the ATM Forum).

The *user plane* provides for the transfer of user application information. It contains a physical layer, an ATM layer and multiple ATM adaptation layers required for different service users; these include constant bit rate (CBR) and variable bit rate (VBR) services.

The *control plane* deals with connection setup and release, and other connection control functions necessary for providing switched services. The control plane structure shares the physical and ATM layers with the user plane, and also includes ATM adaptation layer (AAL) procedures and higher-layer signalling protocols.

The *management plane* provides specific functions and also has the capability to exchange information between the user plane and the control plane. It contains two sections: layer management and plane management. Layer management performs layer specific management functions while the plane management performs management and coordination function related to the complete system.

*Asynchronous Transfer Mode* (ATM) is the new transmission technology developed to be used from desktop to desktop across worldwide information networks. ATM technology was developed to handle different types of information, including voice, video and data traffic. It is radically and fundamentally different from previous technologies, using cell switching instead of packet switching or shared-media solutions.

One of the driving forces behind ATM was the need to exploit evolving telecommunications capabilities. These capabilities are characterized by the following:

- Very high-speed communications links (up to 10 Gbps) are available using fiber technology.
- Extremely low error rates - Compared with traditional copper communications facilities, optical fiber is a million times better.

© Copyright IBM Corp. 1996

- The bandwidth of fiber cable is not limited in comparison to the networking equipments using it. The installation of fiber cables is, therefore, a very economical, long-term investment.

At the same time high bandwidth requirements have been evolving in existing LAN networks. This evolution is associated principally with the increasing day-to-day use of client/server technology, which has led to the following commonly encountered problems for network managers:

- In distributed client/server environments, servers may require bandwidth beyond the limit of current LAN technologies.
- Applications require more and more bandwidth, so in order to maintain quality of service, fewer and fewer workstations can be connected to the same shared LAN segment.
- High segmentation of a large network into segments and subnets solves the problems described above, but it increases the complexity of the network, the latency and delay of the overall network, and the costs of the network in installation, management and maintenance time.
- Multimedia applications require isochronous communication that is not adequately handled by shared-media LANs.

Developing ATM switching technologies has also meant that existing communications practices and protocols are no longer efficient:

- The network and individual switches must be capable of switching data at the full link speed, so existing software switching methods are no longer possible.
- There is no time for error recovery at the physical layer, and low error rates on links make it impractical at the network level. Therefore, it is more effective to check data integrity at the application level and retransmit the whole message if required.
- Existing flow and congestion control mechanisms require very significant processing time, which is no longer available in very fast networks. Flow and congestion controls for data within conventional networks could be replaced by controlling the flow of data entering the network.

ATM has addressed the application, telecommunications and protocol issues highlighted above through the development of a series of standards currently being implemented by vendors. In summary, some of the benefits of this ATM technology are as follows:

- ATM handles different types of traffic (voice, video, data, image, multimedia, etc.) in an integrated way.
- ATM can be used in both LAN and WAN environments, providing a basis for end-to-end internetworking across enterprises.
- ATM uses new hardware switching technology, which allows very fast campus networks to be built.
- ATM is a very cost-effective technology for building a campus network, because users can connect to the network using adapters that support bandwidth according to their individual requirements. Workstations can be connected with low-speed adapters, while servers and backbone switches can use high-speed connections.

- ATM is open - defined by a consortium of vendors and users (the ATM Forum) and standardized by the International Telecommunication Union - Telecommunication (ITU-T, formerly CCITT).

### A.1.1 ATM Network Characteristics

From a design point of view it is important to understand the following fundamental characteristics of an ATM network:

***Connection-Oriented:*** An ATM network uses connection-oriented technologies. This means that there is no way to send data through the ATM network before a connection is established. A connection may be either *switched*, in which case it is built call-by-call by the signalling procedure, or *permanent*, which is preestablished in the network based on preconfigured information.

***Connectionless Operation:*** The majority of pre-ATM networks operate in connectionless mode. Several methods have been defined to emulate connectionless data transfer through connection-oriented ATM networks for compatibility with existing applications. These solutions operate above the ATM transport layer and rely either on the ATM connection being established by the first cell of the data or on using predefined permanent ATM connections.

***Guaranteed In-Sequence Delivery:*** Because ATM cells are delivered over a virtual connection, each cell travels along the same route, and all cells will be transferred to the destination endstation in the same order that they were presented to the network. This restricts the network to using a single path for cells on a single virtual connection, even though other existing physical channels may be available or underloaded.

***Broadcast and Multicast:*** Although ATM is connection-oriented, point-to-multipoint connection types (multicast) are defined. These can be used to send data simultaneously to more than one endsystem and use a tree structure as illustrated in Figure 127 on page 256.

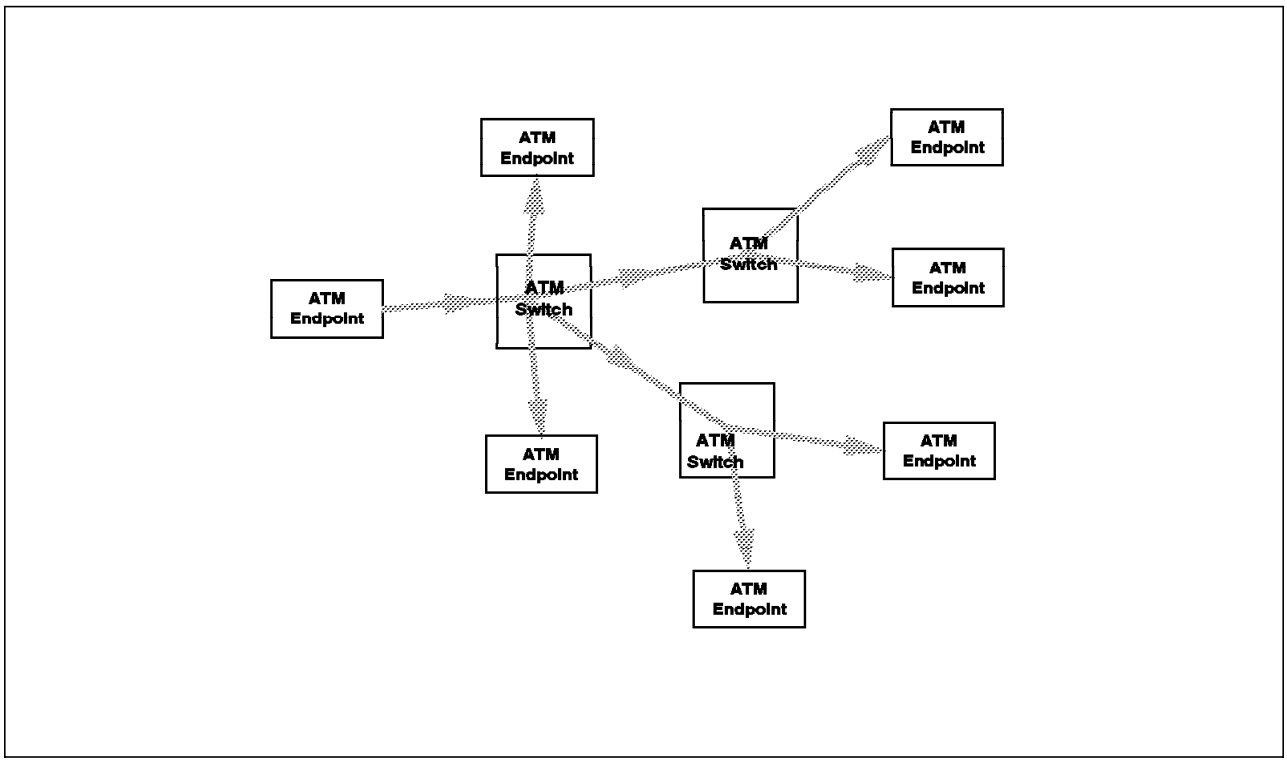



*Figure 127. Multicast Tree*

Point-to-multipoint connections are first established as a single point-to-point connection between the *root* endsystem and one *leaf*. Once this connection is established, a second leaf is connected to it using the optimal route from the established connection. This algorithm is used until the last leaf is connected to the tree.

Some characteristics of a multicast connection are:

- Communication is available from the root to the leaf.
- Data may be sent from the leaf to the root, but does not allow leaf-to-leaf communication over this connection.
- The multicast tree may be set up by signalling or by the network administrator as a permanent connection.

***Quality of Service (QoS):*** Each ATM virtual connection has quality of service characteristics associated with it. During congestion, when a network cannot recover from an overload, it discards only the cells marked as low priority. The network can select which cells to discard depending on the QoS characteristics of the virtual connection.

The QoS parameters defined by the ITU-T are as follows:

- Cell Transfer Delay (Network Latency)
- Cell Delay Variation (Jitter)
- Cell Transfer Capacity (Speed - average and peak allowed rates)
- Cell Error Ratio
- Cell Loss Ratio
- Cell Misinsertion Rate

Each virtual path also has a QoS associated with it. The QoS of a virtual connection within a virtual path may be lower than that of the virtual path, but cannot be higher.

***Cell Loss and Cell Discard:*** Cells may be lost or discarded by an ATM network. The network does not detect the loss of cells and does not signal the user when it has discarded cells from a particular connection.

Some variable bit rate applications for voice and video can produce two types of cells. Standard cells contain basic information, and optional cells contain information on an alternate quality of service. If the user equipment can mark the optional cells, it can avoid the loss of essential information during network congestion.

In fast, cell-based networks, congestion is handled by discarding cells, and recovery is accomplished by retransmission of the full block rather than individual cells.

***Congestion Control:*** ATM networks do not have flow control of the kind found in traditional packet switching networks. This is because traditional windowed link protocols are no longer effective at high link speeds.

In ATM, the parameters for a connection (for example, requested bandwidth and QoS) are examined before the connection is established, and the connection is only allowed if the network can support the desired parameters.

The network allocates resources on a statistical basis. It allows for the possibility that demand may exceed available network resources, in which case the network will discard cells.

***Input Rate Policing:*** At the entry point of the network, the ATM switch monitors the rate of data arrival for a virtual connection or virtual path according to the negotiated QoS parameters for the connection. It will take action to prevent an ATM endpoint from exceeding its allowed limits using a technique called back-pressure.

In the case of overload, depending upon the network configuration, the network may either:

- Discard cells received over the allowed maximum rate.
- Mark the overloaded cells with CLP.

***End-to-End Data Integrity:*** No end-to-end data integrity is provided by the ATM transport layer. This function is the responsibility of end-user equipment or of a higher-layer protocol.

***Priorities:*** There is no priority defined within an ATM network. CLP is not a real priority mechanism, despite its name, as it simply defines which cells may be discarded in the event of congestion.

## A.1.2 The Structure of an ATM Network

The conceptual structure of an ATM network is shown in Figure 128. The main components of the network are described in the following sections.

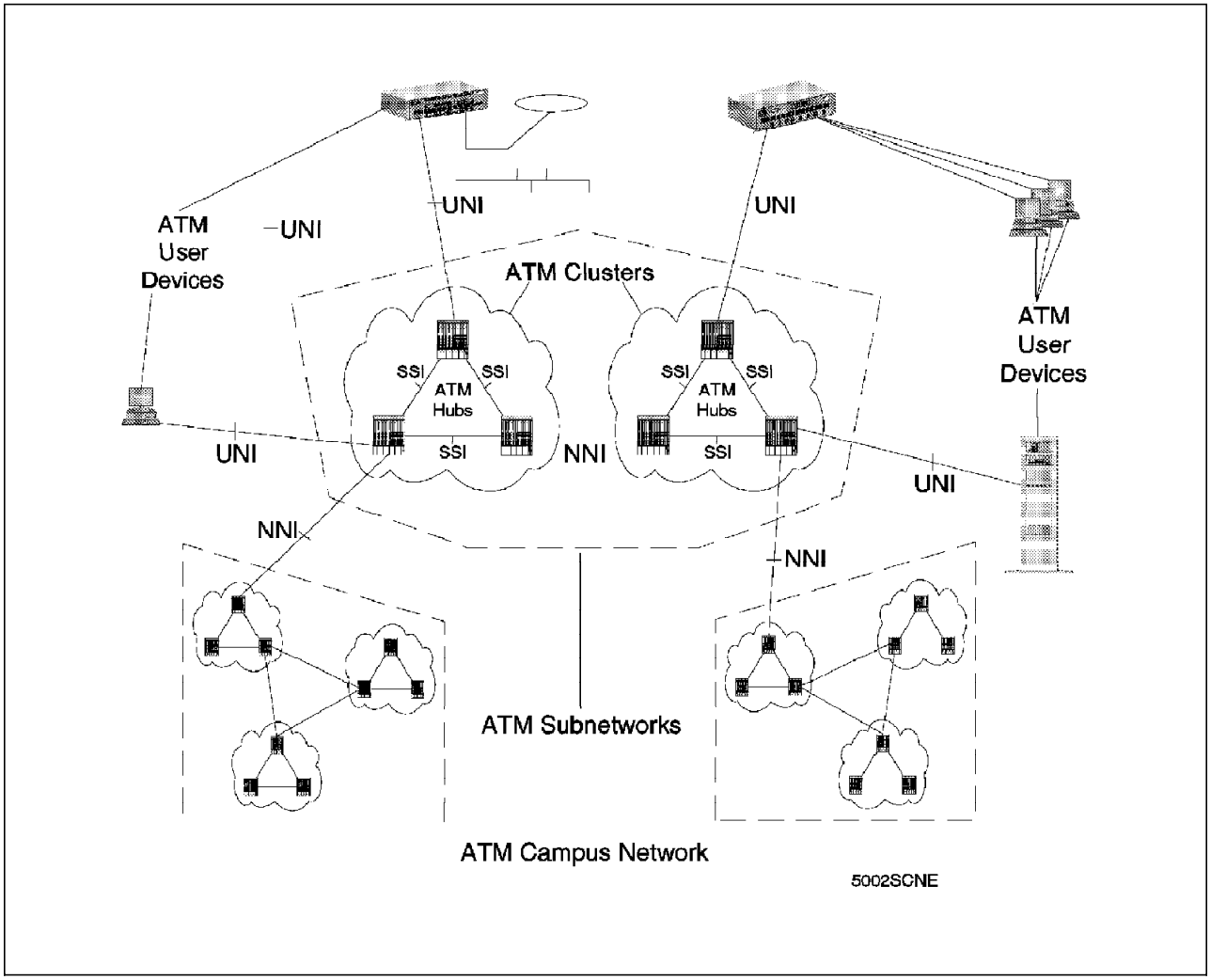



*Figure 128. ATM Network Structure Overview*

***ATM Network:*** An ATM network is a set of interconnected ATM subnetworks controlled as one administrative domain. Networks can be either private or public. Individual networks can be interconnected to each other.

***ATM Subnetwork:*** An ATM subnetwork comprises a single ATM cluster or a set of ATM clusters that are operating as a single routing domain. Clusters within the subnetwork are interconnected by NNI interfaces and share the same routing domain number (RDN) in their ATM addresses.

***ATM Cluster:*** An ATM cluster comprises a single ATM subsystem or a set of ATM subsystems interconnected by SSI interfaces. Each subsystem in an ATM cluster is configured to have the same ATM cluster number (ACN) in its ATM address.

***ATM Subsystem:*** An ATM hub is an example of an ATM subsystem, which typically carries out the switching function in a network. Each subsystem in an ATM cluster is identified by a unique hub number (HN) in its ATM address.

***ATM User Device:*** An ATM user device is the endsystem that encapsulates data into ATM cells and forwards them to a local switch for forwarding over the network. Each user device has a unique ATM address comprising a network prefix, routing domain, cluster number, and hub number, in addition to a local n-byte suffix. Typically the user device learns the remainder of its address (all but the local suffix) during initialization using facilities of the user device to switch interfaces, called the UNI interface.

### A.1.3 ATM Connections

ATM differs from existing LAN networks, because it uses connection-oriented technology. The connection, in ATM terminology, is a point-to-point or point-to-multipoint link from one endsystem to another across a series of ATM switches in a network.

This connection-oriented technology simplifies the routing of cells across the ATM network. Station destination and source addresses do not need to be carried in each ATM cell. Only the connection identifier is required by each ATM switch to route the cell correctly. Because the information between endsystems is sent over the single route described by the connection identifier, the information is received in the same order as it was sent. This in-sequence delivery is required especially for voice and video traffic and also simplifies the processing of data traffic.

Before data can be transferred, a virtual connection needs to be established between the endsystems either by using a predetermined fixed path or by means of the signalling protocol. The connection may, therefore, be permanently established by the network operator (permanent virtual connection, or PVC), or temporarily established on request from an ATM endsystem by signalling (switched virtual connection, or SVC).

To handle each connection, ATM uses the concept of virtual paths and virtual channels. These are described in the following sections.

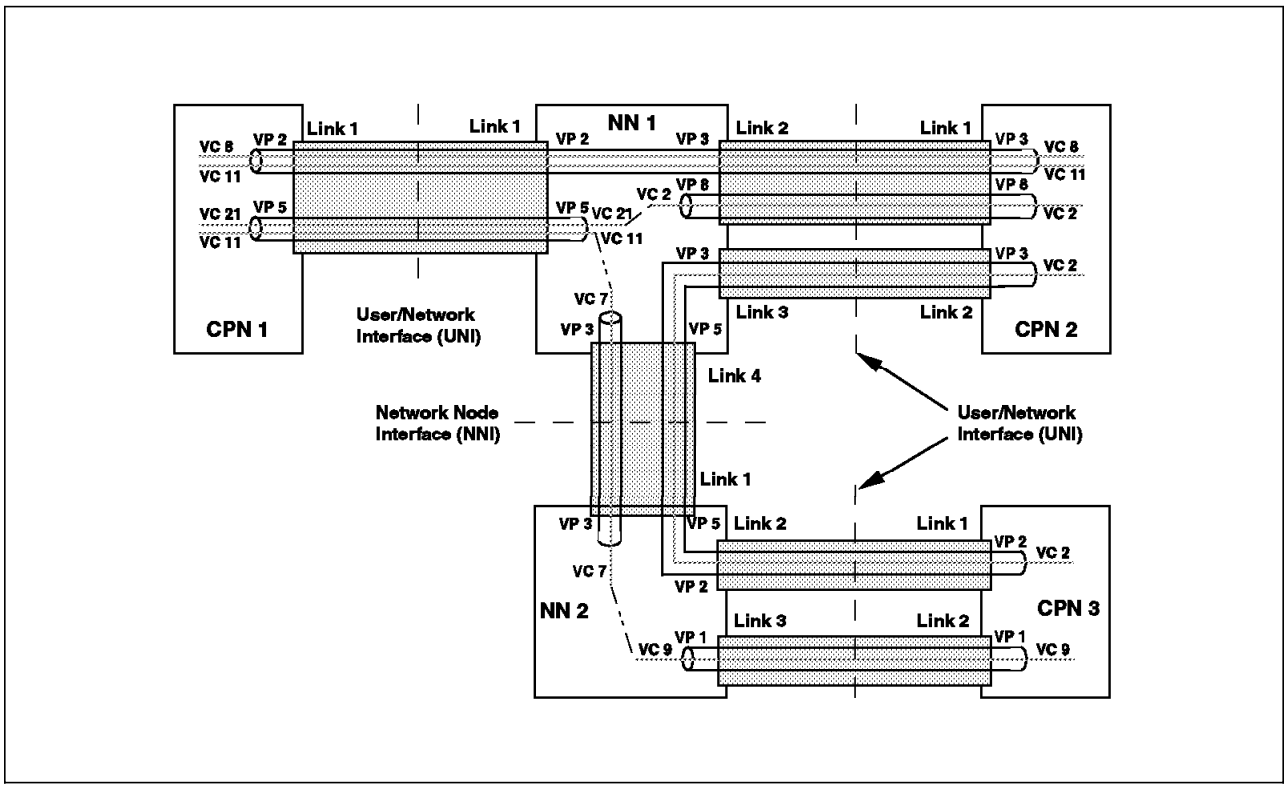



*Figure 129. Routing Concept in an ATM Network*

***Virtual Path (VP) and Virtual Path Indicator (VPI):*** As illustrated in Figure 129, a virtual path is an aggregate route through a network representing a group of virtual channels (VCs). VPs may exist between:

- ATM endsystems
- ATM endsystems and ATM switches
- ATM switches

The virtual path indicator is the indication of the virtual path to be used by a cell and is contained within each cell in the network.

***Virtual Path Link (VPL):*** A virtual path link exists between the points where a VPI value is assigned, translated or determined. Typically these points would be switches in the ATM network.

***Virtual Path Connection (VPC):*** A virtual path connection is the concatenation (sequence) of VPLs that extends between virtual path terminations.

***Virtual Path Switch (VPS):*** A virtual path switch is the processing function that connects VPLs to form VPCs. This function translates VPI values (label swapping) and directs cells to the correct output link at a particular ATM switch.

***Virtual Path Terminator (VPT):*** The virtual path terminator is a processing function also. It terminates each VP and makes the associated VCs available for separate and independent connection routing.

***Virtual Path Connection Identifier (VPCI):*** This identifier of a VP connection is returned by the ATM network when call setup is performed by a user device. It is 16 bits long and is used by the signalling protocol instead of the VPI, which is unique only within a single ATM link.

***Virtual Channel(VC) and Virtual Channel Indicator (VCI):*** A virtual channel is defined in ATM as a unidirectional connection between user devices.

The virtual channel indicator (VCI) is the indication of the virtual channel to be used by a cell and is contained within each cell in the network.

***Virtual Channel Connection (VCC):*** A virtual channel connection is the end-to-end connection along which a user device sends data. This is because, strictly speaking, virtual channels are unidirectional; a VCC would normally consist of two virtual channels to provide full-duplex data transfer.

***Virtual Channel Link (VCL):*** A virtual channel link exists between the points where a VCI value is assigned, translated or determined. Typically these points would be switches in the ATM network.

A virtual channel link is a separately defined data flow within a link or virtual path. A virtual channel connection (VCC) through a network is a sequence of interconnected (concatenated) VCLs.

The relationship between links, VPs and VCs is summarized in Figure 130 on page 262 and Figure 131 on page 262.

***Virtual Channel Switch (VCS):*** The virtual channel switch is the VC switching function shown in Figure 131 on page 262, where VCLs are connected together to form VCCs. To do this, they terminate VPCs and translate VCI values.

There are limits on the number of VPs, VCs, etc. within an ATM network, as follows:

- The maximum number of VPs on links is determined by the number of bits allocated to address the VPs in the cell header (VPI). This is either 8 or 12 bits.
- The maximum number of VCs within a VP is determined by the number of bits allocated to address the VCs in the cell header (VCI). This is 16 bits.
- There may be additional limits imposed by the capacity of specific ATM equipment.

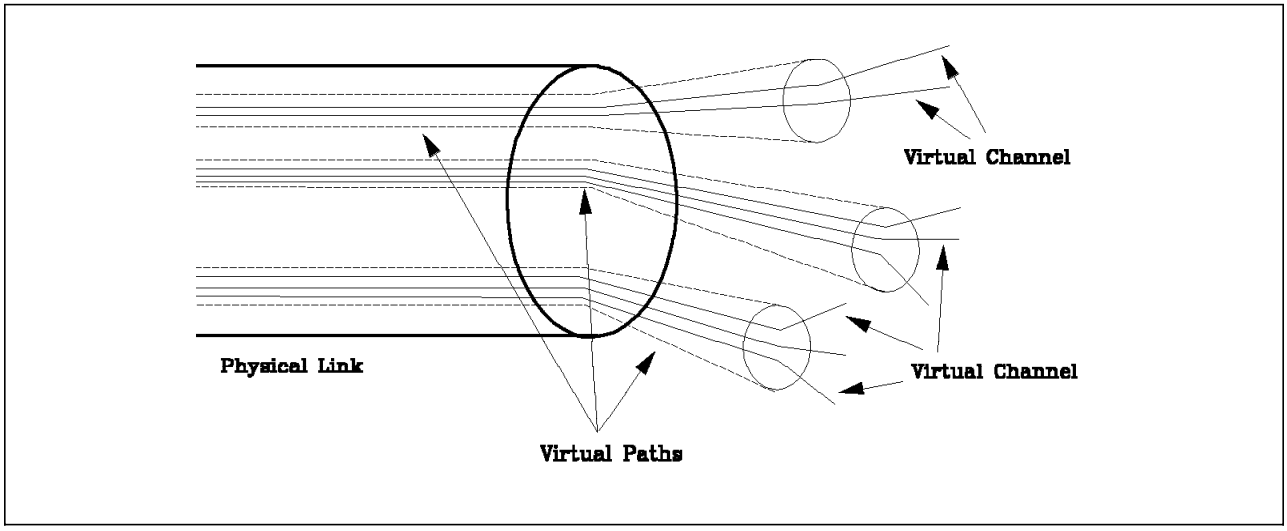



*Figure 130. Link, Virtual Path and Virtual Channel Relationship*

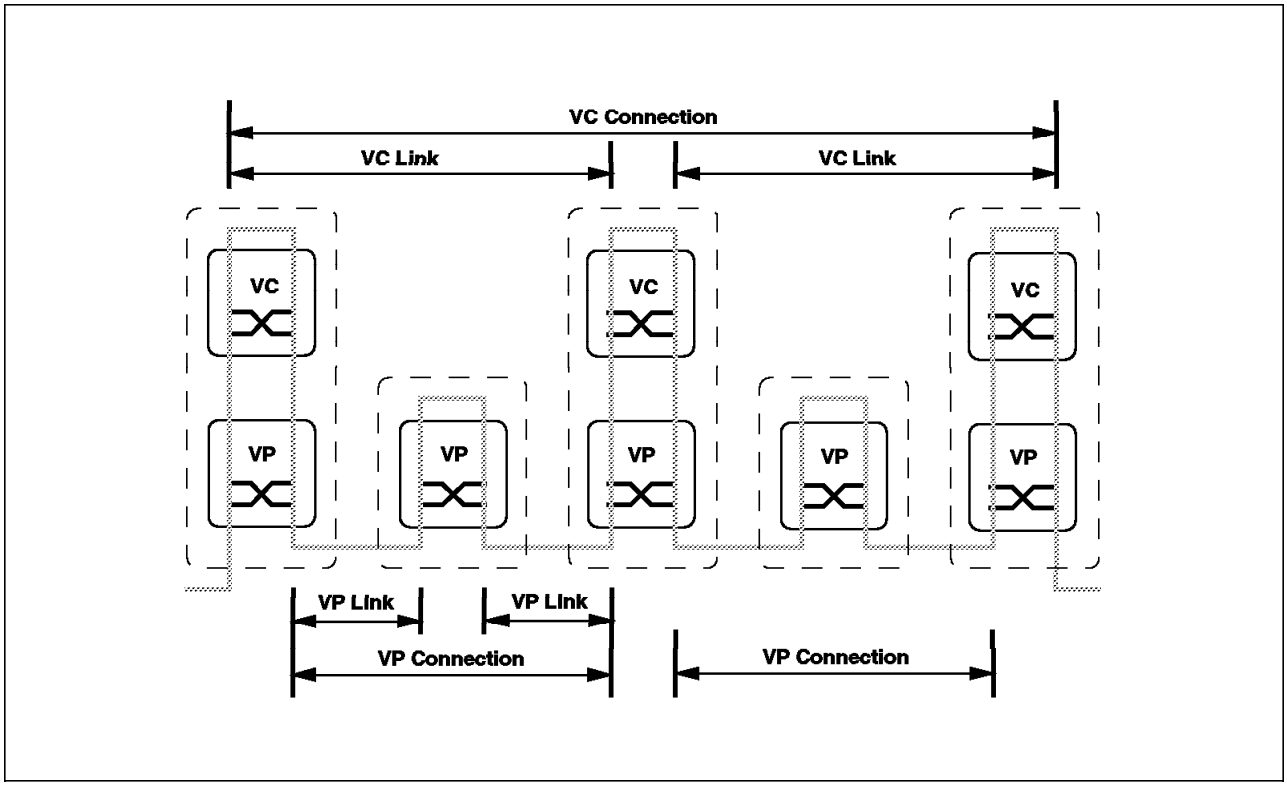



*Figure 131. VP and VC Concepts*

## A.1.4 Routing/Switching ATM Cells

An ATM cell is transmitted along a virtual channel connection according to the routing information contained in its header. This routing information is swapped at every switch along the path of the connection, to enable the routing of the cell to the next switch along the connection. This process is referred to as *label swapping*.

The routing information in the cell consists of the VPI and the VCI fields in the cell header. The definition of the VPI/VCI mapping is established when the connection is established. The mapping for the connection is held at every intermediate switch and consists of VPI/VCI input fields mapped to VPI/VCI output and port output fields.

This means that each VPI/VCI pair is associated with a particular port on a switch and each VPI/VCI associates a cell with an input link and a corresponding output link. Figure 132 shows an example of mapping VPI/VCI. Based on the VPI/VCI in the header of the cell, an ATM switch can identify the output link across which the cell is to be routed and give the new link identifiers to the cell.

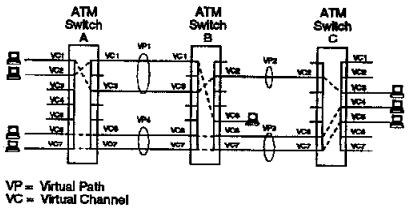



*Figure 132. VPI and VCI Switching in an ATM Switch*

Switching of all virtual channels within a virtual path can take place inside a switch. In this case the switch unpacks the virtual path into unique virtual channels. Using its routing table, the switch groups them together again, forming new outgoing virtual paths as shown in the top portion of Figure 133.

The bottom portion of Figure 133 shows an example where the VPI is switched without unpacking it.

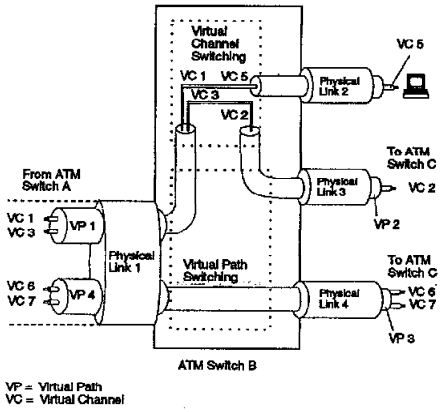



*Figure 133. VPI and VCI Mapping Across an ATM Switch*

## A.1.5 ATM Cells and Cell Format

In ATM networks information is transmitted in cells. Cells are fixed-length packets. Each packet is 53 bytes long; 48 bytes are the *payload* with a 5-byte *header*. The header contains control information, including the VPI/VCI route identifier that defines cell route information. The header is error checked to

prevent errors from being propagated over the network. The payload is user information that is not protected by error checking at the ATM network level.

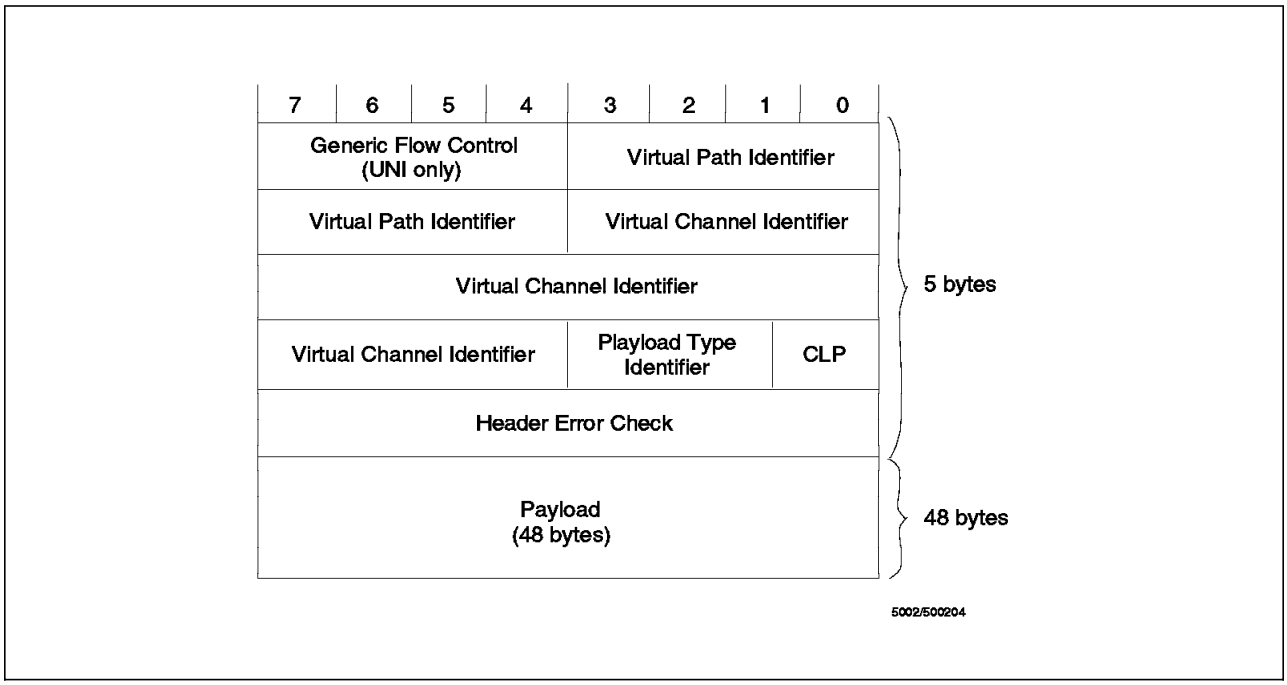



*Figure 134. ATM Cell*

The cell formats for the user network interface (UNI) and the network node interface (NNI) are illustrated in Figure 135 and Figure 136 on page 265.

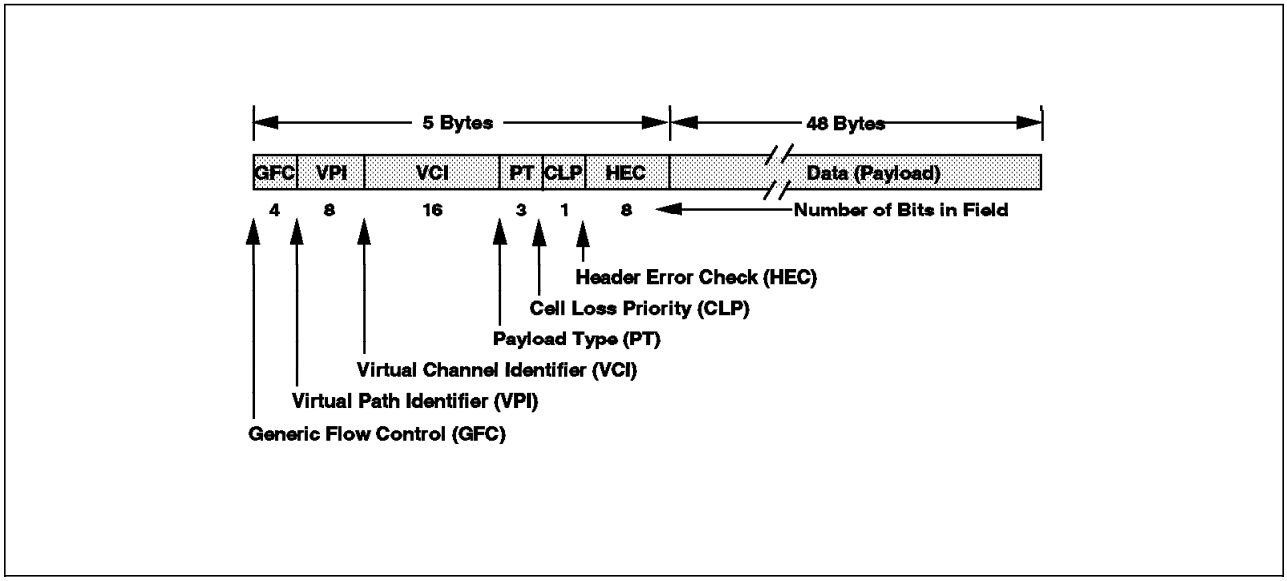



*Figure 135. ATM Cell Format at the User Network Interface (UNI)*

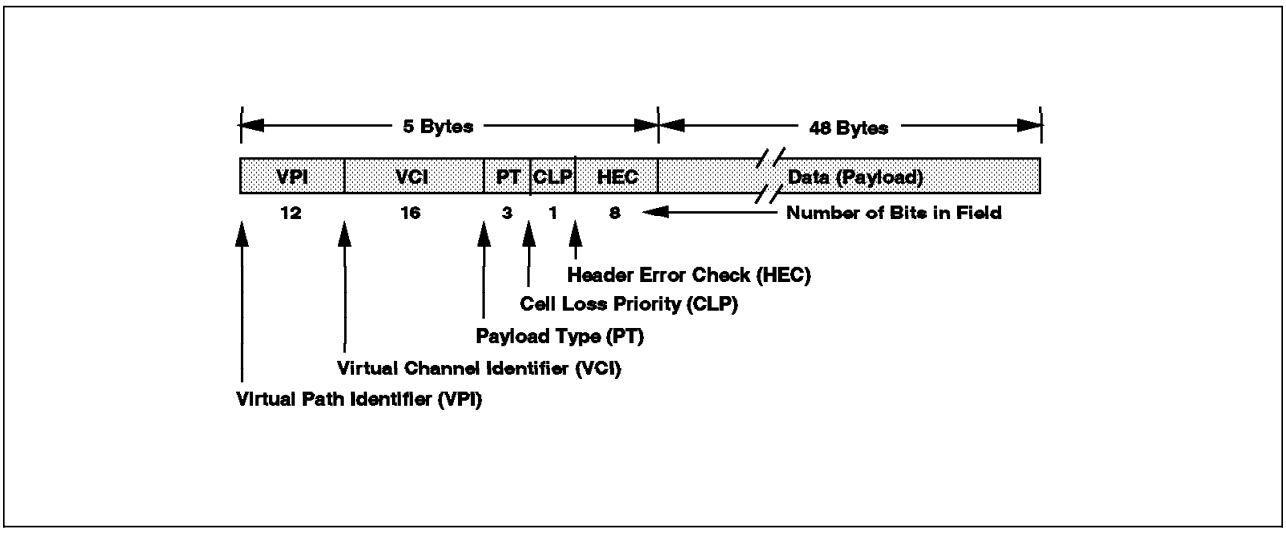



*Figure 136. ATM Cell Format at the Network Node Interface (NNI)*

The fields within the cell are as follows:

***Cell Size:*** An ATM cell always contains 48 bytes of data, with a 5-byte header. The cell size was defined by the ITU-T as a compromise between data and voice requirements.

***Generic Flow Control (GFC):*** Currently, use of the GFC has not yet been standardized, but it is defined as having the same length as in the UNI cell.

**Note:** This field is missing in the NNI cell header so the flow control can only be a procedure local to the user device.

***VPI and VCI:*** These two fields, Virtual Path Identifier (VPI) and Virtual Connection Identifier (VCI) are the most important fields in the cell header. Together they identify the logical connection (the virtual connection) over which the cell is travelling. Some VPI/VCI values are reserved for signalling (for example, connection establishment) and for maintenance and resource management.

If the VPI and VCI values are set to zero, it means that the cell is empty. Empty cells may be required in a network to maintain physical link protocols.

***Payload Type (PT):*** This is a three-bit field. Bit 0 determines whether the cell is user data (Bit 0 = 0) or operations, administration and maintenance (Bit 0 = 1).

Bit 1 is used when the cell carries user data. Bit 1 = 1 means that congestion was experienced somewhere along the route passed by the cell.

Bit 2 is used by higher-layer processing. Bit 2 = 1 means that the cell is the last cell of the user data frame. If Bit 2 = 0, the cell is the first or the middle cell of the frame.

***Cell Loss Priority (CLP):*** When set, this bit indicates that the cell is a low-priority cell. When the system needs to discard cells in a congested situation, these cells should be discarded first.

***Header Error Check (HEC):*** This field allows the correction of all single-bit errors or the detection of multi-bit errors in the header part of the cell.

### A.1.6 ATM Signalling

ATM signalling is the process used for dynamic setup and clearing of ATM switched virtual connections (VPCs and VCCs) at the UNI interface.

Permanent virtual connections are established by a network operator. If a connection is permanent, and if the network fails and is restarted, the circuit is reestablished. In case of recovery from failure, the network will not reestablish lost switched virtual connections.

The key elements of ATM signalling are:

- Signalling takes place on separate VCCs from those used by user data. The same principle is used as in narrowband ISDN where the D channel is used to initiate connections.
- There is a method to set up additional signalling channels besides the predefined channels (although this is not used in current ATM implementations).
- Point-to-point signalling is the default signalling method, but broadcast signalling may be implemented in the future. Point-to-multipoint connections are established using point-to-point signalling.
- Class B (AAL-2) services have not yet been defined, so it is not supported by the signalling protocol.
- The Class D protocol is also not supported, because clients must have connections to connectionless servers. Therefore, the calling procedure is not required.
- Connection setup by third-party equipment, which is not involved in the data transfer, is not supported.

The routing framework developed for the routing of ATM connections is based on an extension to the OSPF (open shortest path first) routing protocol known as widest path OSPF. In this modification, the cost of a link is defined as its available bandwidth. This algorithm will select the path providing the widest bandwidth for the connection, instead of the shortest as defined in *traditional* OSPF.

These characteristics allow the connection path to be computed in advance (precomputation); hence, the route computation process is independent of the connection setup process. During connection setup, the only checking required is whether there is available bandwidth for the new connection according to its QoS characteristics. Using precomputed routes, connection setup time will be shorter and a higher connection setup rate can be provided by the ATM network.

The following functions and subfunctions are defined for ATM signalling:

1. Call Establishment
    - Setup
    - Call Processing
    - Connect
    - Connect Acknowledge
2. Call Clearing
    - Disconnect

- Release
- Release Complete

3. Status
   - Status Enquiry
   - Status
4. Point-to-Multipoint Messages
   - Add Party
   - Add Party Acknowledge
   - Add Party Reject
   - Drop Party
   - Drop Party Acknowledge

Typical connect and disconnect procedures are shown in Figure 137 and Figure 138 on page 268.

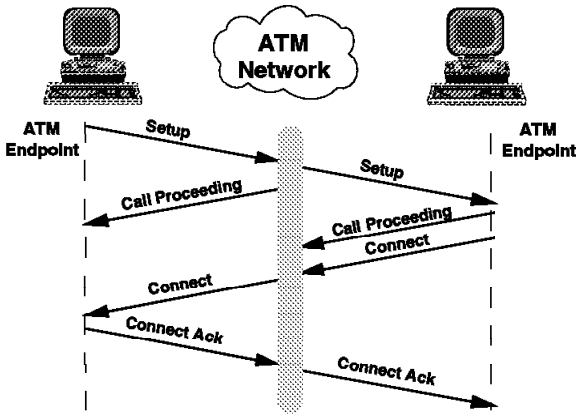



*Figure 137. ATM Switched Circuit Setup Protocol*

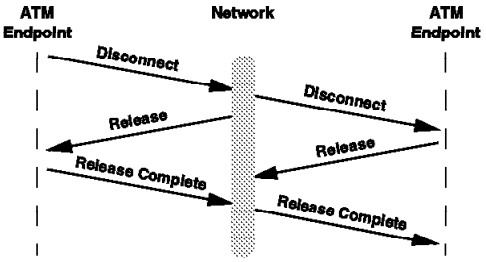



*Figure 138. ATM Switched Circuit Disconnect Protocol*

## A.1.7 ATM Address Format

The address formats used in ATM are shown in Figure 139.

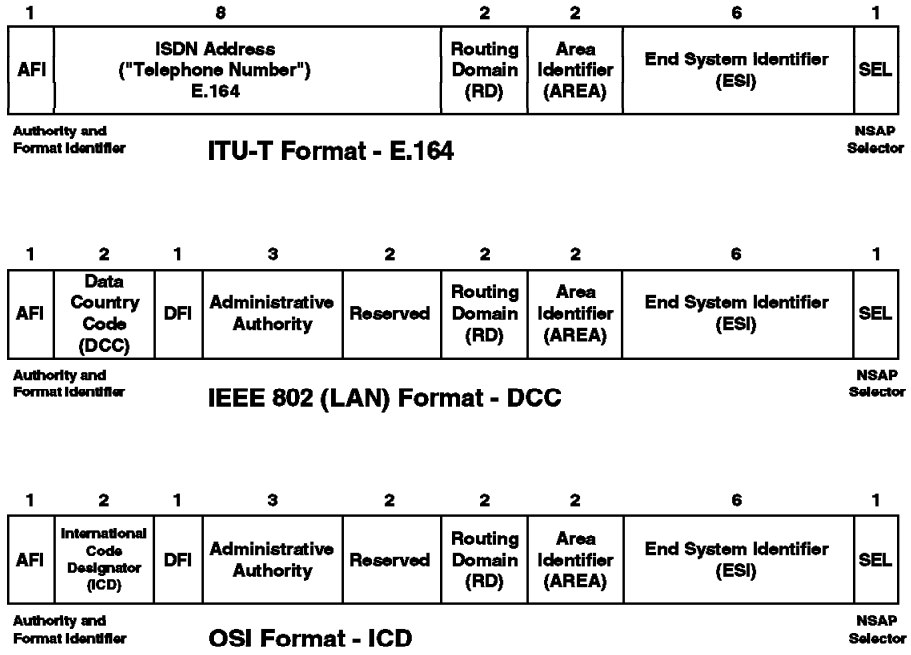



*Figure 139. ATM Address Formats*

The address must be separated into two distinct parts:

- The Endsystem Address identifies the endsystem uniquely in a cluster. The field is 7 bytes long and consists of an ATM endsystem address (similar to a LAN MAC address) and a single byte endsystem identifier (ESI) that identifies a subcomponent of an endsystem.

- The remaining 13 bytes contain the network part of the address that includes fields for the hub number, cluster number, and routing domain, as well as the standard network prefix.

There are three different formats for ATM addresses, each controlled by a different authority.

***ITU-T (E.164) Format:*** This format is essentially the same as telephone style addressing. It is specified by the ITU-T and will be used by public (carrier provided) ATM networks.

***DDC (Data Country Code) Format:*** This format carries LAN addresses as specified in IEEE 802 recommendations.

***IDC Format:*** This format is specified by the ISO for OSI.

The ATM Forum specifies that equipment in a private network must support all three formats.

### A.1.8 ATM Adaptation Layers (AALs)

The network characteristics required by various types of traffic over an ATM network are provided by an ATM adaptation layer, which is found in each end system and, in special forms, in switches.

The ITU-T has defined four different generic service classes of network traffic, each of which must be treated differently by an ATM network. These classes are designated Class A to Class D and four different types of ATM Adaptation Layers (AALs) have been defined to realize the necessary network characteristics to handle them. The relationship between these is shown in Figure 140 on page 270, and the functions of the appropriate layers are shown in Figure 141 on page 270.

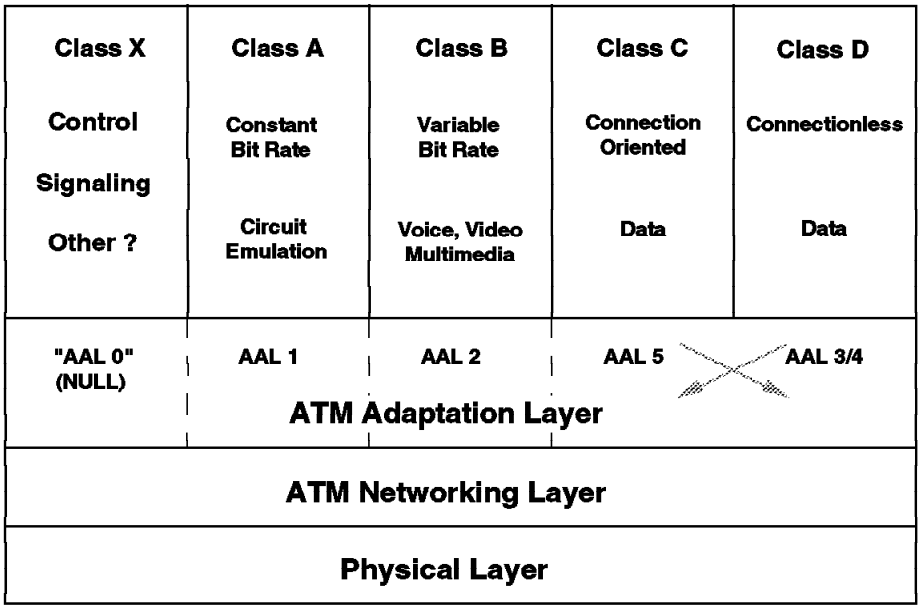



*Figure 140. Service Classes and AAL Types*

| *Layer Name* | | *Functions* | |
|---|---|---|---|
| Higher Layers | | Higher Layers Functions | |
| AAL | Convergence sublayer | Service specific (SSCS)<br>Common part (CPCS) | Layer Management |
| | SAR sublayer | Segmentation and reassembly | |
| ATM | | Generic flow control<br>Cell header generation/extraction<br>Cell VPI/VCI translation<br>Cell multiplexing/demultiplexing | |
| Physical | Transmission convergence (TC) sublayer | Cell rate decoupling<br>Cell delineation<br>Transmission frame generation/recovery | |
| | Physical medium dependent (PMD) sublayer | Bit timing<br>Physical medium | |

5002/500205

*Figure 141. Composition of the ATM Sublayers*

The following sections provide a short description of these service classes.

### A.1.8.1 Class A (Circuit Emulation)

This service emulates a leased line and is used for traffic that has a constant bit rate (for example, voice and video).

The characteristics of Class A are:

- A constant bit rate at source and destination
- A timing relationship between source and destination
- A connection between end users of the service

To realize these functions, the adaptation layer must perform the following services:

- Segmentation and reassembly of data frames into cells
- Handling (by buffering) of cell delay variations
- Detection and handling of lost, discarded, misrouted or duplicated cells
- Recovery of the source clock frequency
- Detection of bit errors in the user information field

### A.1.8.2 Class B (Variable Bit Rate Services)

This service is intended for isochronous voice and video traffic, which may be coded as variable rate information, and requires a timing relationship between the ends of the connection. The service is strictly connection-oriented.

The services provided by the class B adaptation layer are:

- Transfer of variable rate information between endpoints.
- Transfer of timing between source and destination.

No indication is provided of lost or corrupted information.

Some video applications (such as videoconferencing) require synchronization between voice and video, while others can be transmitted in multicast mode (just like a film) and are not sensitive to network delay.

The design and control of a network for Class B traffic is very challenging because of the unpredictable bandwidth required, and because the data rates are often near to the peak rate capability of the network.

### A.1.8.3 Class C (Connection-Oriented Data)

Class C traffic is traditional data traffic, such as SNA and X.25. The service offered for it is connection-oriented and it supports variable rate information flow.

The services provided by the Class C adaptation layer are:

- Segmentation and reassembly of frames into cells
- Detection and signalling of errors in user data frames
- Possible multiplexing and demultiplexing of multiple end-user connections into a single ATM connection

### A.1.8.4 Class D (Connectionless Data)

The Class D service is connectionless and also supports variable rate information flow. It is intended to support connectionless protocols such as TCP/IP.

The services provided by the Class D adaptation layer are:

- Segmentation and reassembly of frames into cells
- Detection and signalling of errors in user data frames
- Multiplexing and demultiplexing of multiple end-user connections into a single ATM connection
- Network layer addressing and routing

### A.1.8.5 Class X (User-Defined)

This is a connection-oriented ATM transport service where the network characteristics are user defined. Only the required bandwidth and the QoS parameters are used by the network.

As shown in Figure 140 on page 270, the different service classes are represented by the appropriate ATM adaptation layer types.

### A.1.8.6 AAL-0

This is the null AAL and corresponds to a process that connects the AAL service interface directly to the ATM networking service.

### A.1.8.7 AAL-1

AAL-1 is used for Class A (constant bit rate) traffic. In practice this may take the form of data in an SDH or PDH frame where the frame rate is constant, but data exists in a specific part of the frame so that it arrives at the network in short periodic bursts.

When these cells are transported through the network, dependent upon the current load on the network, they may suffer relative delay. This delay is not consistent and can introduce jitter. The receiver has to buffer the received data to avoid problems of overrun or underrun data frames, resulting in additional delay in the data stream.

Figure 142 shows the format of AAL-1 cell.

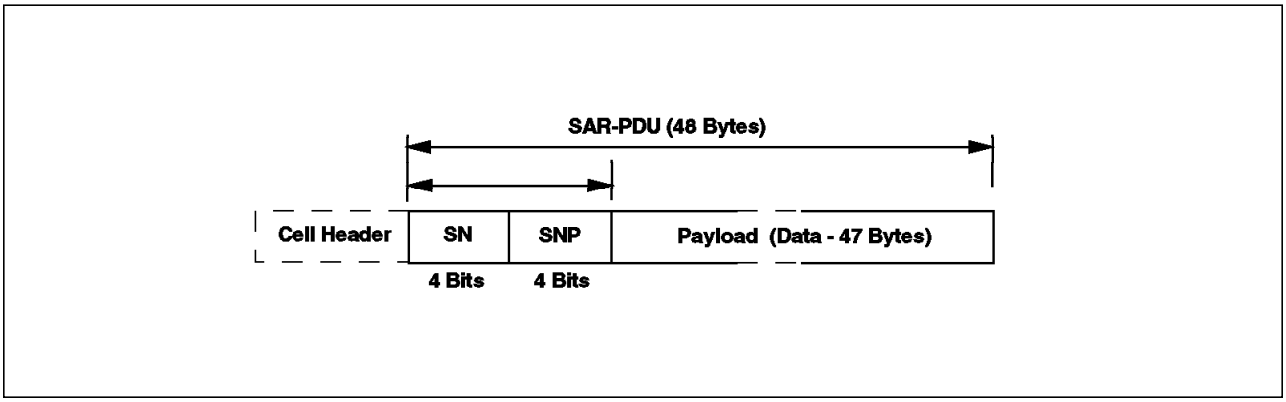



*Figure 142. Cell Format for AAL-1*

There are two additional control fields in the cell, but because there is no additional space in the cell header, they are located at the beginning of the payload. Each field is 4 bits long.

***Sequence Number (SN):*** This field contains a three-bit sequence number field that operates in cyclic fashion, with a 1-bit CS indicator (CSI bit), which is used depending upon the type of the traffic.

***Sequence Number Protection (SNP):*** This field is CRC protection for the SN field.

In order to minimize the delay caused by assembly and playout, it is possible to send cells that are not full. This fact must be specified between the endsystems at the time of circuit establishment.

### A.1.8.8 AAL-2

AAL-2 is used for Class B traffic. AAL-2 processes data streams like AAL-1. The only difference is the variable bit rate due to compression. One of the key problems of AAL-2 is handling the skew between voice and video information.

AAL-2 is currently absent from the draft standards; the detailed description has been delayed because of the previously mentioned problems.

### A.1.8.9 AAL-3/4

AAL-3/4 is used for Class C and Class D traffic.

It is a relatively complex, high-function AAL that will offer assured data delivery.

When the AAL detects corruption of a data frame, it will be able to retransmit the data frame, although the details of this operation have not been defined.

In blocking mode, short data frames are blocked into a longer data unit for transmission through the network, and the multiplexing of several AAL-to-AAL connections onto a single VCC connection is possible.

Point-to-multipoint connections are also possible using AAL-3/4.

In this mode, several additional control information bits are placed in the payload field. The format of the cell for AAL-3/4 is shown in Figure 143 on page 274.

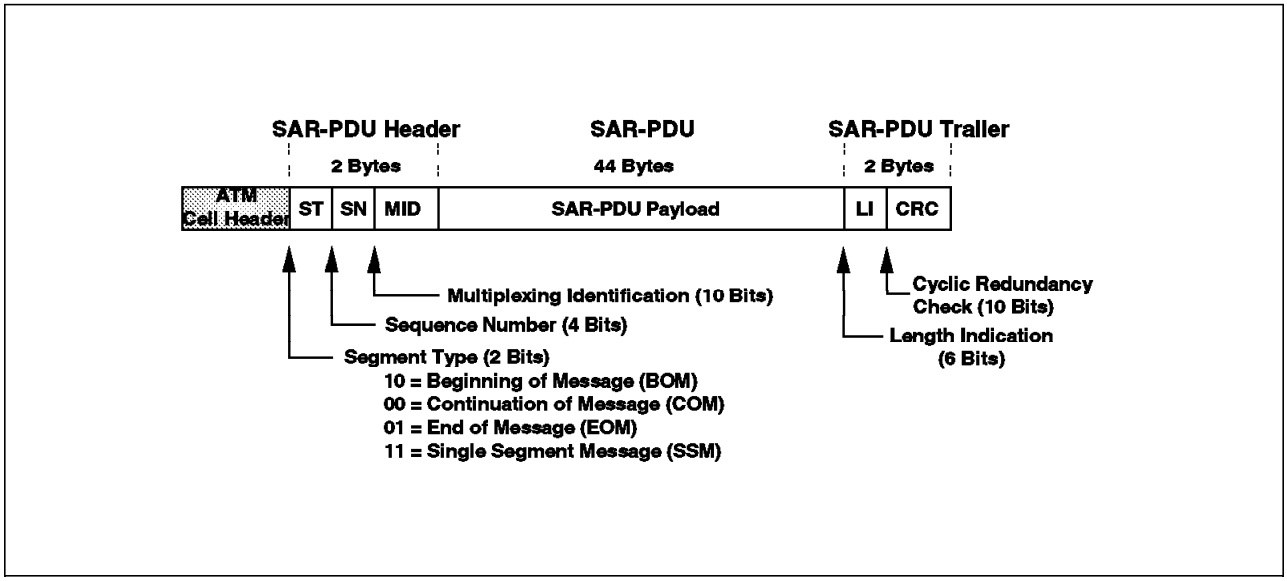



*Figure 143. Cell Format for AAL-3/4*

The fields have the following meaning:

***Segment Type (ST):*** These two bits indicate where the content of this cell is located in the data frame.

***Sequence Number (SN):*** This 4-bit field is a module four counter to verify the correct sequence of the cells.

***Multiplexing Identification Field (MID):*** In the case of multiplex mode, this field indicates the connection to which the cell belongs.

***Length Indicator (LI):*** This field specifies how much data is in the variable part of the cell. All cells except the last cell of the data frame should be full.

***Cyclic Redundancy Check (CRC):*** This is a polynomial CRC to protect the whole of the cell payload, except for the CRC itself.

### A.1.8.10 AAL-5

This AAL is often called a Simple and Efficient Adaptation Layer (SEAL). It is designed to operate significantly more efficiently than AAL-3/4 but, with the exception of connection multiplexing, has the same functions as AAL-3/4.

In the case of AAL-5, the whole user frame is protected by CRC, and there is no length field in every cell as in the case of AAL-3/4. In this way, the whole 48 bytes of the payload can be used for data transmission. AAL-5, therefore, can recognize corrupted or missing data only when the whole data frame is received at the destination endsystem.

## A.2 ATM Network Interfaces

ATM interfaces define interoperability and connectivity between different components of an ATM network.

They have been defined by both the ITU-T and the ATM Forum. The ITU-T has been working on the definition of the public UNI and NNI. The interface

specifications defined by the ATM Forum include user node interface (UNI), private network-to-network interface (P-NNI) and the data exchange interface (DXI).

An ATM network comprises a set of endsystems and a set of intermediate nodes (switches), all linked by a set of point-to-point ATM links. The different types of interfaces used over these links are shown in Figure 128 on page 258 and are described in the following sections.

### A.2.1 ATM Physical Interfaces

***SONET and SDH:*** Synchronous Optical Network (SONET) is a US standard for the internal operation of PTT optical fiber networks. It relates closely to a system called Synchronous Digital Hierarchy (SDH), which has been adopted by the ITU-T for the internal operation of carrier (PTT) optical fiber networks worldwide.

Traditionally PTT networks have been built by using a cascade of bandwidth multiplexors at each end of the high-speed connection. This resulted in more and more stages of multiplexing to provide faster links, the internals of which were generally proprietary. For example, the US used a different structure from Europe, and both the US and Europe used different structures from Japan.

Both SONET and SDH (which was developed from it) eliminate the problems illustrated above by providing a standardized method of internal operation and management and worldwide compatibility, while enabling existing speeds to be accommodated. They permit many levels of multiplexing and demultiplexing to be achieved in a single step and allow many different speed channels to be carried through the system. With this new scheme, it is possible to access low-bandwidth channels without having to demultiplex the whole bandwidth stream.

The basic structure in SONET is a frame of 810 bytes, which is sent every 125 microseconds. This allows a single byte within a frame to be a byte in a 64-Kbps digital voice channel. Since the minimum frame size is 810 bytes, the minimum speed at which SONET will operate is 51.84 Mbps, which is calculated as follows:

810 bytes x 8000 frames/second x 8 bits = 51.84 Mbps

The basic SONET frame is called the Synchronous Transport Signal Level 1 (STS1), as shown in Figure 144 on page 276.

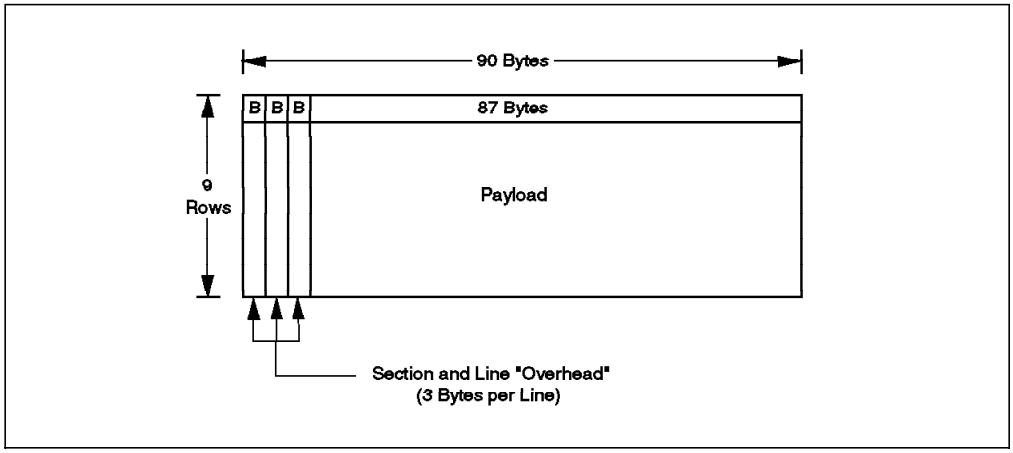



*Figure 144. SONET STS1 Frame Structure*

It is conceptualized as containing nine rows of 90 columns each with the following attributes:

- The first three columns of every row are used for administration and control of the multiplexing system. They are called *overhead* in the standard but are very necessary for successful systems operation.
- The frame is transmitted row by row, from the top left of the frame to the bottom right. One frame is transmitted every 125 microseconds.
- It is important to remember that the representation of the structure as a two-dimensional frame is purely a method of showing a repeating structure. In reality it is only a string of bits with a defined repeating pattern.

Multiple STS1 frames can be byte multiplexed together to form higher speed signals. When this is done, they are called STS2, STS3, etc., where the numeral suffix indicates the number of STS1 frames that are present. For example, STS3 is three times an STS1 or 155.52 Mbps.

For more information regarding the details of the SONET frame structure, please refer to Appendix B3 in *ATM Technical Overview*, SG24-4625.

***SONET LITE:*** SONET LITE is not an officially accepted term, but is a convenient description of the underlying technology upon which it is based. It is the standard proposed by the ATM Forum for use with multimode fiber in the local LAN environment, and is based on the SONET STS3c standard.

The term LITE comes from the fact that most of the management information flows of STS3c have been replaced. This is possible because of the relatively short distances involved in LAN environments, which make them inherently more reliable than WAN connections. The approach of minimizing some of the frame overheads significantly reduces the cost of implementation, making it an attractive implementation option.

## A.2.2 The UNI Interface

The UNI specification defines the interface between ATM endsystems (such as terminals, routers, bridges, servers, or concentrators equipped with an ATM adapter) and the ATM network. The UNI comes in two parts: the private UNI, and the public UNI.

The private UNI defines interfaces between an endsystem and a switch owned by an organization. An analogy would be the IEEE 802.3 specification.

The public UNI defines interfaces between an endsystem and a service provider's equipment. An analogy would be the specification of the interface to public X.25.

The two interfaces have different controlling organizations: the private by the ATM Forum and the public by ITU-T.

The major differences between the two UNI specifications are as follows:

- Link Types Allowed

  Some of the link types allowed by the private UNI use protocols that work over very short distances (for example, 100 meters). These would be obviously inapplicable to a public network interface.

- Addressing Format

  Public ATM networks will use the E.164 addresses (similar to those used for telephone numbers), while private networks will probably use addressing from LAN or OSI environments.

In the campus environment, the private UNI is used and is supported by the IBM 8260 hub. The cell format used by this interface is shown in Figure 135 on page 264.

Basic signalling information (for example, connection setup) across the UNI uses the VPI=0, VCI=5 channel. The Interim Local Management Interface (ILMI) protocol (which is part of the UNI specification) is used by switches to communicate their network address prefixes to endsystems, and for the end systems to communicate the ESI portion of their network address for the switches at initialization time. The ILMI protocol uses the VPI=0, VCI=16 channel for address registration.

***Generic Flow Control (GFC) at the UNI:*** The header of the UNI cell contains a field named GFC. Using this field, one-way flow control is defined from the ATM endsystem to the ATM switch. There is no control in the opposite direction.

According to the UNI standard, three queues may be defined at the endsystem: one for uncontrolled traffic, and two for controlled traffic. Usually, only one queue is used for controlled traffic. This concept is illustrated in Figure 145 on page 278.

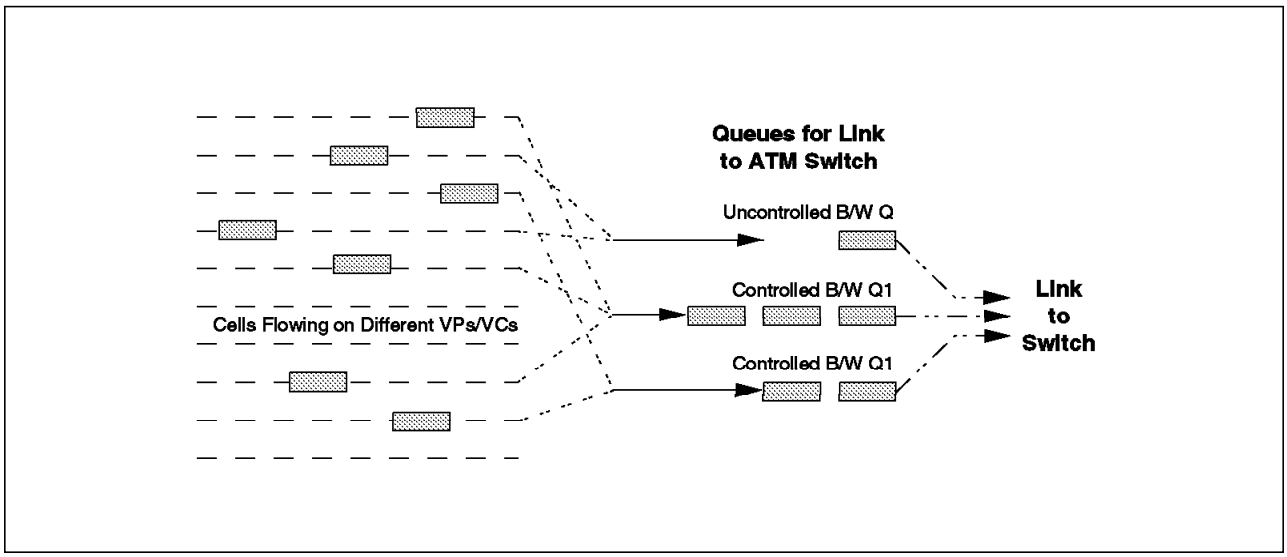



*Figure 145. GFC Operation*

***Controlled Traffic:*** This is the traffic for which the GFC mechanism is defined. It is usually all the nonreserved bandwidth (NRB) traffic on the interface. Controlled traffic is distinguished in the cell header by the presence of a non-zero GFC field.

***Uncontrolled Traffic:*** This traffic is not subject to GFC control and is treated as having a higher priority than the controlled traffic. It would normally be RB traffic for which there is a reserved bandwidth.

The flow control mechanism uses windowing. Each queue has a window that represents the number of cells that it is allowed to send before the network must respond giving permission to continue. The window is maintained as a counter in the endsystem. Each time a cell is sent, the counter is decremented and the endsystem is allowed to send cells as long as the value of the counter is not zero. If the counter reaches zero, the endsystem must stop transmitting until the counter has been reset to an initial value. During normal operation, the switch sends reset signals fairly often so that the counter never reaches zero.

***GFC field towards the network (outbound):***

- Bit 0 is unused.
- Bit 1 indicates that the cell is flow controlled by Q1.
- Bit 2 indicates that the cell is flow controlled by Q2. (In the case where only one queue exists, this bit is always zero.)
- Bit 3 indicates if the equipment is controlled (1) or not controlled (0).

***GFC field away from the network (inbound):***

- Bit 0 means HALT (1) or NOHALT (0). If HALT, the network is unable to receive input from the endsystem, even uncontrolled reserved bandwidth traffic.
- Bit 1, when set, means reset the counter of Q1.
- Bit 2, when set, means reset the counter of Q2. If Q2 does not exist, it must be 0.
- Bit 3 is reserved for future use.

# Appendix B. IBM LAN Emulation over ATM

This chapter discusses the specification of IBM′s proprietary LAN emulation architecture.

In order to use the vast base of existing LAN application software over an ATM network, it is necessary to define an ATM service, called *LAN emulation*. This service emulates the services of existing LANs, such as token-ring or Ethernet across an ATM network, and can be supported via a software layer in endsystems. The objective of LAN emulation is to replace and/or complement the existing LAN system with an ATM-based system with as little change to the workstation software as possible.

If such a LAN emulation service is provided for an ATM network, then endsystems (for example, workstations and servers) could connect to the ATM network while the software applications interact as if they were attached to a traditional LAN. Also, this service would support interconnection of ATM networks with traditional LANs by means of today′s bridging, switching, and/or routing methods. This will allow interoperability between software applications residing on ATM-attached endsystems and on traditional LAN endsystems.

The LAN emulation service will be important to the acceptance of ATM, since it provides for the migration of existing LAN applications to the ATM environment. Networking customers have indicated in recent surveys that coexistence with existing networks and transition to ATM need to be addressed. Likewise, the surveys have concluded that ATM for workgroup LANs and LAN backbones are the top planned uses. Customers expect to continue to use existing LAN applications as they migrate to ATM.

In practice, emulated LANs over ATM are not going to exist in isolation. They will need to interface to existing LANs in various ways via the well-established switching, bridging and routing methods. Also, they will need to provide the customers with the ability to reuse the existing software in the ATM-attached endsystems.

To emulate a LAN service, different types of emulation can be defined, ranging from emulation of the MAC service (that of IEEE 802.x LANs) to emulation of the services of network and transport layers. There are many potential ways of achieving this (at different levels in the protocol stack), but the most attractive appears to be LAN emulation at the MAC level. This approach allows the widest possible coverage of existing applications and potentially the cleanest and most trouble-free software interface. This book concentrates on MAC service emulation.

The challenge in developing an emulated LAN architecture that can meet the above objectives is to accommodate the differences between the following environments:

1. LANs are connectionless versus the connection-oriented approach of ATM, which requires that a connection be established between endsystems before data can be exchanged.
2. Broadcast and multicast operations are easy and natural on a LAN, because a frame sent on the medium will be received by all other users. Due to the

© Copyright IBM Corp. 1996

connection-oriented approach with ATM, however, broadcasting to all addresses is not as easily achieved and a different approach is required.

3. LAN MAC addresses, based on manufacturing serial numbers, are independent of the network geography. more ways than just the format. The ATM address of an endsystem will be determined by the ATM switch to which the endsystem is connected. Moving the endsystem from one ATM switch to another will result in a change in its ATM address, whereas LAN addresses are device serial numbers burned into the LAN adapter card at the time of manufacture (although an optional ability exists for the user to specify a locally administered LAN address). Any LAN emulation system will need to make use of real LAN addresses for some functions. Therefore, a database that allows mapping from LAN addresses to ATM addresses must exist somewhere.

Other approaches to using ATM for the existing LAN services are documented in IETF RFC1483, ″Multiprotocol Encapsulation Over ATM Adaptation Layer 5,″ and RFC1577, ″Classical IP and ARP Over ATM″. The ATM Forum Technical Committee is also working toward a Multiprotocol Over ATM (MPOA) system to run multiple internetwork layer protocols over ATM. The fundamental purpose of the MPOA service is to provide end-to-end internetworking layer connectivity across an ATM fabric, including the case where some internetworking layer hosts are attached directly to the ATM network, and some are attached to legacy subnetwork technologies. Because the work done in this area is not yet finalized by the ATM Forum Committee, and IBM is not yet making products available that implement this architecture, the subject of MPOA will not be covered in this book.

The deployment of ATM into the Internet community is just beginning and will take many years to complete. During the early part of this period, this is expected to follow traditional IP subnet boundaries for the following reasons:

- Administrators and managers of IP subnetworks will tend to initially follow the same models as they currently have deployed. The mindset of the community will change slowly over time as ATM increases its coverage and builds its credibility.
- Policy administration practices rely on the security, access, routing, and filtering capability of IP Internet gateways (that is, firewalls). ATM will not be allowed to get around these mechanisms until ATM provides better management capability than the existing services and practices.
- Standards for global IP over ATM will take some time to complete and implement.

**Important**

Before the ATM Forum Technical Committee produced its standard for LAN emulation over ATM, IBM was already researching and developing technologies in this field. These were presented to the ATM Forum Committee for inclusion in the final specification of the ATM Forum′s LAN emulation.

The ATM Forum LAN emulation specification resulted as a combination of many vendors′ efforts and represents the input from all those involved. Many of IBM′s submissions were accepted as part of the standard.

In order to provide the customers with products that can be used to implement ATM networks using LAN emulation and in view of the fact that IBM had developed a working architecture for LAN emulation before the ATM Forum had completed its specification, IBM decided to offer a set of interim products that implemented the IBM proprietary LAN emulation architecture.

IBM′s intention has always been that these products will be phased-out in favor of (or upgraded to) products conforming to the ATM Forum′s LAN emulation specification as soon as possible. In fact, IBM has already started shipping certain products that are ATM Forum-compliant. (For example, the LAN emulation client implemented in the 8260 A-CPSW module is based on the LAN emulation specification from ATM Forum.) Therefore, customers installing LAN emulation over ATM networks using the IBM products implementing the IBM proprietary LAN emulation architecture should plan their migration strategy to IBM products implementing ATM Forum′s specification. IBM representatives will be able to provide the customers with information about the availability dates for IBM products using ATM Forum-compliant specifications.

## B.1 IBM′s Proprietary LAN Emulation over ATM

This section defines the components and functions of IBM′s proprietary LAN emulation service. The major components of this service are described in the following subsections.

### B.1.1 LAN Emulation Layer

Figure 146 on page 282 shows the interconnection of two ATM stations (for example, workstations or servers) over an ATM network using the LAN emulation service. Inside the ATM stations, the LAN emulation (LE) function is provided by a so-called LAN emulation layer (LE layer). The users of the LE layer are protocols (such as NetBIOS, IPX, IEEE 802.2 LLC) written for traditional LANs, such as Ethernet or token-ring.

The LE layer shields these protocol stacks from the characteristics of the ATM network and gives them the illusion of being attached directly to a traditional LAN. Therefore, by using the services of the LE layer, no changes to the LAN applications are required for running them over the ATM network. Since we are emulating the MAC services of traditional LANs, each LE layer is identified by a MAC address that can be either globally or locally administered. The LE drivers shipped for the IBM ATM adapters allow you to choose whether the emulated LAN is Ethernet or token-ring. In case of Ethernet, the drivers are capable of

supporting both 802.3 and DIX simultaneously. Note that an endsystem using these drivers can be a member of a single emulated LAN only.

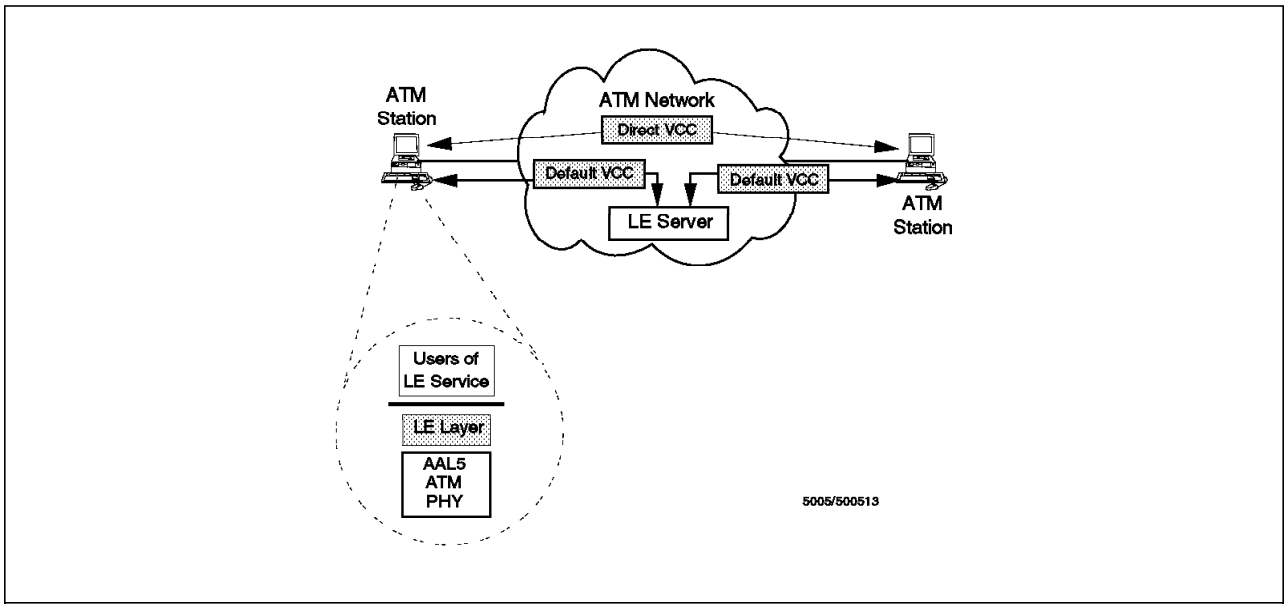



*Figure 146. ATM Station-to-ATM Station Using LAN Emulation*

Figure 147 on page 283 shows the interconnection of an ATM station with a station attached to a traditional LAN. The new component in this configuration is the ATM bridge, which also contains an LE layer similar in function to the LE layer in an ATM endstation. This bridge LE layer also provides its user, in this case the MAC relay function of the bridge, with the illusion of having a traditional LAN on the ATM port. This means that today′s bridging methods, such as those defined in IEEE 802.1d, can be employed without any modification.

The IBM 8281, which is the product implementing such a function, provides up to four LAN ports and one ATM port. The ATM port can be configured to emulate either an Ethernet (IEEE 802.3 and DIX) or token-ring (IEEE 802.5) LAN, but not both at the same time. The LAN ports can also be configured to be either an Ethernet (IEEE 802.3 and DIX) or token-ring (IEEE 802.5) LAN. Note that all the LAN and ATM ports on the IBM 8281 must be configured for the same LAN type (no mixing of the LAN type is allowed on the 8281). This means that the IBM 8281 cannot be used as a translational bridge between different LAN types (emulated or real) but can be configured to perform the one of the following functions:

- Bridging from Ethernet LAN ports to an ATM port emulating Ethernet
- Bridging from token-ring LAN ports to an ATM port emulating token-ring

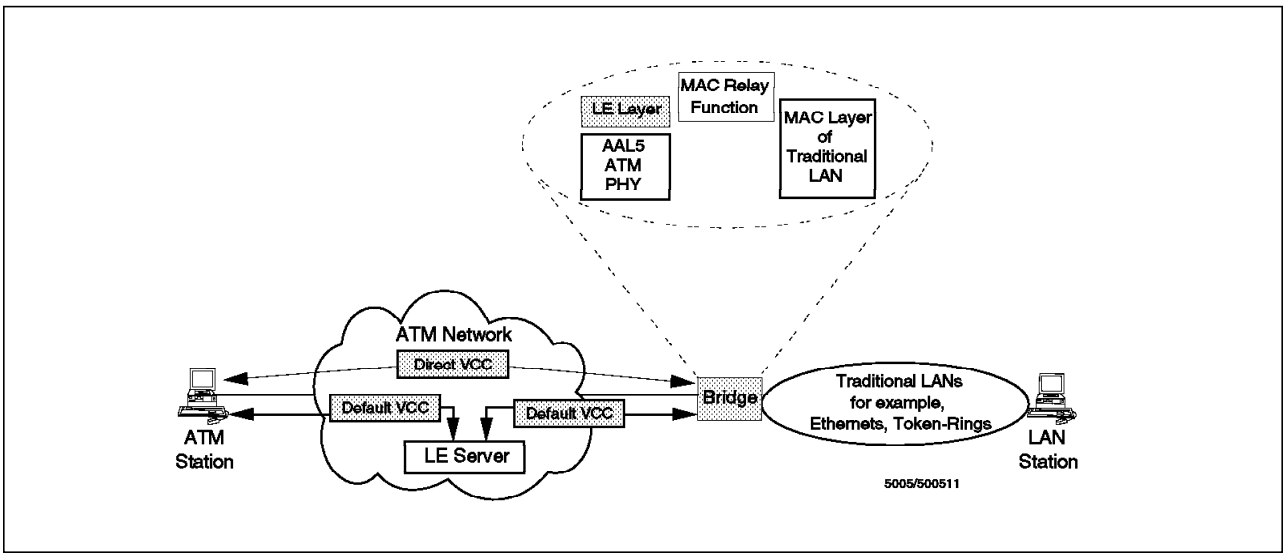



*Figure 147. ATM Station-to-LAN Station Using LAN Emulation*

Figure 148 shows the conventional LAN-to-LAN interconnection over ATM using LAN emulation services. There is no difference between the bridge component shown here and the one in the second configuration. Similarly, there is no difference in the LE server or the VCCs required for connection over the ATM network.

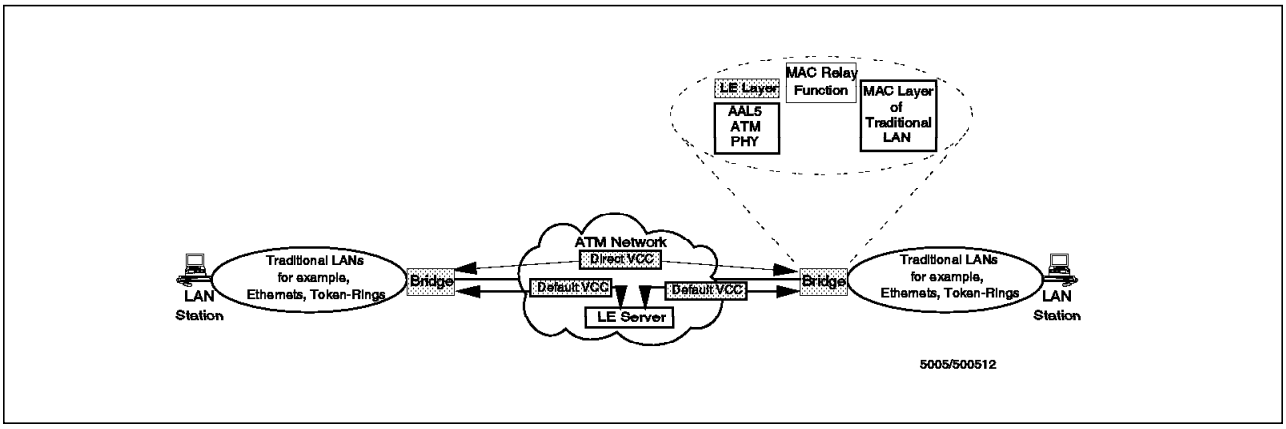



*Figure 148. LAN Station-to-LAN Station Using LAN Emulation*

## B.1.2 LAN Emulation Server

The LAN emulation service is provided by a LAN emulation server (LE server).

The LE server provides the LE layers with a connectionless service for the transfer of LAN messages between destinations. That is, the LE server acts as the shared medium for an emulated LAN in the inherently connection-oriented ATM network. Additionally, the LE server provides multicast and directory (address server) services for the LE layers.

There may be multiple instances of emulated LANs on a single ATM network. These instances can be a mixture of 802.3 and 802.5 LANs. For each LAN that is emulated over an ATM network, a separate LE server is required. The scope of an emulated LAN is thus defined by the set of ATM-attached endsystems and bridges that receive services from a particular LE server.

### B.1.3 Default VCC

The components of an emulated LAN exchange LAN emulation control messages through Default VCCs, and encapsulated LAN frames through either Default or Direct VCCs. A Default VCC is a point-to-point VCC connecting a LAN emulation layer in an endstation or ATM-LAN bridge to its corresponding LAN emulation server. The LE drivers for the IBM ATM adapters require that the Default VCCs be provided using switched virtual connections (SVCs). The Default VCCs only use non-reserved bandwidth connections.

The Default VCCs are required because of the following:

1. All LE control messages are passed between an LE layer and its LE server through a Default VCC.
2. The Default VCC is the default path for passing encapsulated LAN frames between LE layers if no Direct VCC is available between the LE layers that are exchanging information.

Default VCCs are always bidirectional; that is, a single Default VCC can be used for both sending and receiving information.

### B.1.4 Direct VCC

A Direct VCC is a point-to-point VCC connecting two LAN emulation layers directly. Direct VCCs are always switched; this means that signalling protocols are required for their establishment, maintenance, and release.

### B.1.5 Using Default and/or Direct VCC

The main function of the LE layer inside an ATM endsystem is the transfer of LAN frames (issued by higher layers) to their destination. LAN frames are transferred through the ATM network over either a Default or a Direct VCC, as indicated in the following:

1. Default VCCs: The first possibility is to send the LAN frames to the LE server using the Default VCC. The LE server will then forward them to their destinations based on address information included in the header of the frames.
2. Direct VCCs: The second possibility is to send the LAN frames to the destination LE layer using a Direct VCC. Direct VCCs are established and maintained by the LE layer as long as they are needed. The ATM addresses of the destination LE layers needed to set up the Direct VCCs are supplied by the LE server.

The decision on which VCC to employ depends on whether the sending LE layer knows the ATM address of the receiving LE layer or not. If it does not, then it sends the LAN frame over the Default VCC. If, however, the sending LE layer knows the ATM address of the receiving LE layer, then the following will occur:

- If a Direct VCC to that LE layer already exists, the frame will be sent directly to the receiving LE layer over the available VCC.
- Otherwise, frames will continue to be sent via the Default VCC until a Direct VCC has been established; subsequent frames will be sent directly to the receiving LE layer over the newly established Direct VCC.

### B.1.6 General Multicast VCC and Bridge Multicast VCC

Since we are emulating the MAC services of existing LANs, the LAN emulation service must also support the use of multicast MAC addresses (that is, broadcast, group, or functional MAC addresses). This is done by establishing a General Multicast VCC and Bridge Multicast VCC. Both of these VCCs take advantage of the point-to-multipoint VCC support provided by the IBM 8260.

A General Multicast VCC is a VCC from the LE server to the LAN emulation layer in endsystems (ATM endstations and bridges). The LE server transfers all frames with a multicast address to an endsystem via the General Multicast VCC. Multicast messages are sent to all stations, relying on filtering in those stations as is done in existing shared media LANs. In the case of the group and functional addresses, the filtering in each endsystem (as is currently done on 802.3 and 802.5 LANs) is used to discard the frames that are not intended for that endsystem.

A Bridge Multicast VCC is a VCC from an LE server to the LAN emulation layer in ATM-LAN bridges. For example, this VCC is used by the LE server to forward the individual address frames to destinations that are connected to an 802.3 LAN and are accessed through a transparent bridge.

### B.1.7 Functions of the LAN Emulation Service

The LAN emulation service, as provided through the LE layer in ATM endsystems (that is, ATM stations and bridges to existing LANs) and the LE server, comprise the following five functions:

1. Initialization
2. Address registration
3. Address management and resolution
4. LAN frame transmit
5. LAN frame receive

These functions are defined in the following paragraphs.

#### Initialization

The initialization function provides an endsystem with the capability to obtain access to the Default VCC, connecting the LE layer in an endstation or ATM-LAN bridge to the LE server for the exchange of control and user information.

#### Address Registration

The address registration function provides the MAC addresses and/or route descriptor information to the LE layer in ATM endsystems for local filtering of incoming LAN frames.

#### Address Management and Resolution

The address management and resolution function provides a method by which an ATM endsystem learns a destination ATM address so that it can establish a Direct VCC for the transmission of LAN frames. This method includes mechanisms for learning the ATM address of a target station, mapping the MAC address (or route descriptor field in the LAN frame) to an ATM address, storing the mapping in a table, and managing the table.

The LAN emulation layer in ATM endstations and bridges will maintain an association (or mapping) between the destination of LAN frames and the Direct VCC (and ATM address) of the LAN emulation layer that provides connectivity to the destination. This mapping is required in order to set up and use a Direct VCC to another ATM endstation or ATM-LAN bridge attached to the ATM network. The mapping is referred to as the address cache (also known as Destination Address Association Table-DAAT) for the endsystem. The address management and resolution function in the endsystem manages and updates the address cache and provides information to the LAN frame transmit function, when required to transmit a LAN frame.

Similarly, this function provides the LE server with a means to support the use of a Direct VCC by an ATM endsystem. This includes mechanisms for mapping the MAC address (or route descriptor field in the LAN frame) to an ATM address, storing the mapping in a table, managing the table, and providing the mapping to ATM endsystems.

Address management in the LE server includes the building and maintenance of address tables that are used by the server to manage the flow of messages between endsystems and providing address resolution services. The LE server tables, kept in cache memory, associate the ATM address of the endstations or ATM-LAN bridges with virtual circuits connecting these endsystems to the LE server. These connections may be point-to-point (Default VCC) connections (for example, to an ATM endstation or an ATM-LAN bridge), broadcast (for example, to all endsystems), or multicast (for example, bridge multicast).

***LE Server Address Cache:*** The address cache in the LE server includes a table, named the Server Address Association Table (SAAT), that relates all registered MAC addresses to their respective ATM endstations and ATM-LAN bridges. A registered endsystem is one that has used the ATM and LAN emulation address registration procedures.

**LAN Frame Transmit**

The LAN frame transmit function in ATM endsystems consists of LAN frame forwarding as well as VCC control. LAN frame forwarding includes identification of the VCC (Default or Direct) over which a LAN frame is to be sent using the address management table and encapsulation of the frame. VCC control consists of the establishment, release, and maintenance of Direct VCCs.

Similarly, the LAN frame transmit function of the LE server determines the outgoing VCC over which a frame is to be forwarded by querying the address management table. This function also may trigger the address resolution function, which provides the originating endsystem of a LAN frame with the ATM address of the destination so that a Direct VCC may be employed.

**LAN Frame Receive**

For an ATM station, the LAN frame receive function determines whether an incoming data frame is to be received by this endstation, according to the LAN frame filter controls specified by the higher layers. Once a data frame is received, this function performs the decapsulation of the frame before delivering it to the user. For frames received via a Direct VCC, this function may inform the address management and resolution function of that VCC and its associated source MAC address so that the LAN frame transmit function can establish a

Direct VCC for sending LAN frames in the opposite direction. For a bridge, the LAN frame receive function is the same as that of an endstation.

For the LE server, the LAN frame receive function delivers every frame it receives to the LAN frame transmit function. The LE server does not alter the contents of received LAN frames.

## B.1.8 IBM LAN Emulation Message Flows

This section details message flows between ATM attached stations using IBM′s LAN emulation. For details of flow, when using ATM-LAN bridges, please refer to 3.5.3, “ATM-LAN Bridge Module and LAN Emulation” on page 53.

### B.1.8.1 802.3 ATM Station-to-802.3 ATM Station

In this section, we examine the message flow between two ATM-attached stations (emulation 802.3) when they use the IBM′s LAN emulation protocol. The message flow between these two stations is shown in Figure 149.

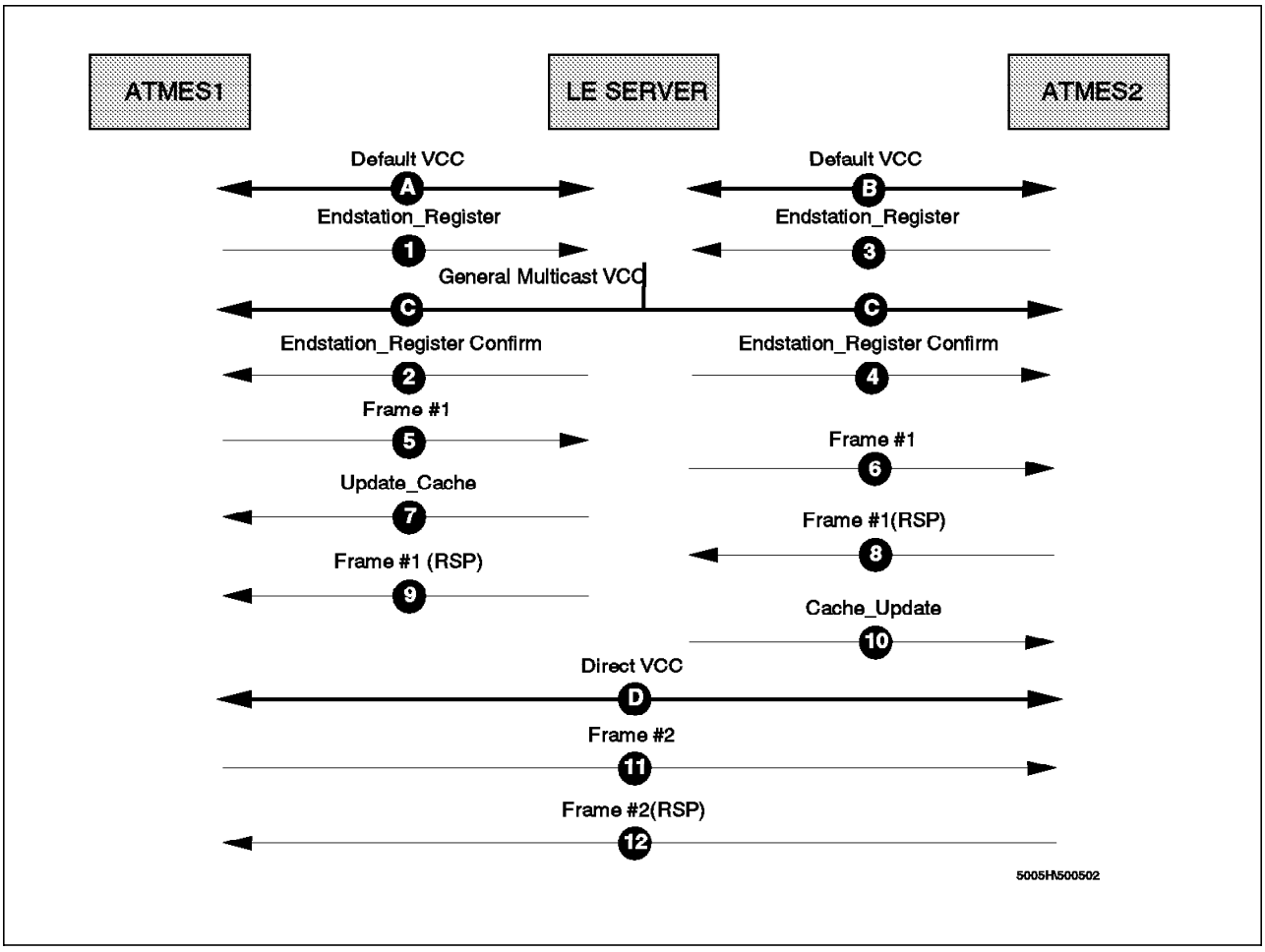



*Figure 149. 802.3 ATM Station-to-802.3 ATM Station Message Flow*

It is assumed that the ATM endsystems (ATMES1 and ATMES2) and the LAN emulation server have all completed their ILMI process and have initialized at both the physical and ATM layers. They already know their ATM address. Also, ATMES1 and ATMES2 know the ATM address of the LE server because this is a configuration parameter specified by the administrator as part of the configuration of ATMES1 and ATMES2.

1. ATMES1 establishes the Default VCC **A** with the LE server.
2. ATMES1 sends a REGISTER_ENDSTATION **1** to the LE server using the Default VCC **A**.

   This frame contains information including the type of frame, registration or deregistration, type of multicast service required, and whether ATMES1 will establish Direct VCCs. It may optionally include the ATM address of ATMES1.
3. The LE server updates its SAAT forwarding/filtering table.
4. The LE server adds ATMES1 as a leaf to the General Multicast VCC **C**.
5. The LE server issues a REGISTER_ENDSTATION_CONFIRM **2** to ATMES1 using the Default VCC **A**.

   This frame contains information including a list of all successfully registered addresses, the emulated LAN identifier (ELID) and the originator identifier (OID).
6. ATMES2 establishes the Default VCC **B** with the LE server.
7. ATMES2 sends a REGISTER_ENDSTATION **3** to the LE server using the Default VCC **B**.

   This frame contains information including the type of frame, registration or deregistration, type of multicast service required, and whether ATMES2 will establish Direct VCCs. It may optionally include the ATM address of ATMES1.
8. The LE server updates its SAAT forwarding/filtering table.
9. The LE server adds ATMES2 as a leaf to the General Multicast VCC **C**.
10. The LE server issues a REGISTER_ENDSTATION_CONFIRM **4** to ATMES2 using the Default VCC **B**.

    This frame contains information including a list of all successfully registered addresses, the emulated LAN identifier (ELID) and the originator identifier (OID).
11. When the LE layer in ATMES1 receives a frame, from the upper-layer protocol, destined to the MAC address of ATMES2, it examines its Destination Address Association Table (DAAT) to determine whether it knows the ATM address of the target. In this scenario, it is assumed that the address is not in the ATMES1′s DAAT. In this case, ATMES1 sends this frame **5** to LE server over its Default VCC **A**.

    **Note:** In our discussion we have assumed that the upper-layer protocol already knows the MAC address of the destination. Normally, a broadcast frame is used by the upper-layer protocol to obtain this MAC address. For the sake of simplicity, we have not shown this step in our flows.
12. The LE server examines its SAAT to determine the ATM address of the requested target. In this case, since ATMES2 is registered with the LE server, the LE server forwards the data frame **6** to the target (ATMES2) over the Default VCC **B**.
13. The LE server also sends an Update_Cache **7** to the ATMES1 with the ATMES2′s MAC-to-ATM address mappings so that ATMES1 can update its DAAT.
14. ATMES1 updates its DAAT.

15. ATMES2 sends the reply 8 to the LE server because it does not find the address of ATMES1 in its DAAT.
16. The LE server forwards the reply 9 to ATMES1.
17. The LE server also sends an Update_Cache message 10 to ATMES2 with the ATMES1′s MAC-to-ATM address mapping so that ATMES2 can update its DAAT.
18. ATMES2 updates its DAAT.
19. When the second frame is to be sent from ATMES1 to ATMES2, ATMES1 establishes a Direct VCC D with ATMES2 and sends the frame 11 over these connections.
20. The response to this message 12 will be sent by ATMES2 over the Direct VCC D.
21. All subsequent unicast frames between ATMES1 and ATMES1 are sent over the Direct VCC D as well.

### B.1.8.2 802.5 ATM Station-to-802.5 ATM Station

In this section we examine the message flow between two ATM-attached stations (emulation 802.5) when they use IBM′s LAN emulation protocol. The message flow is shown in Figure 150.

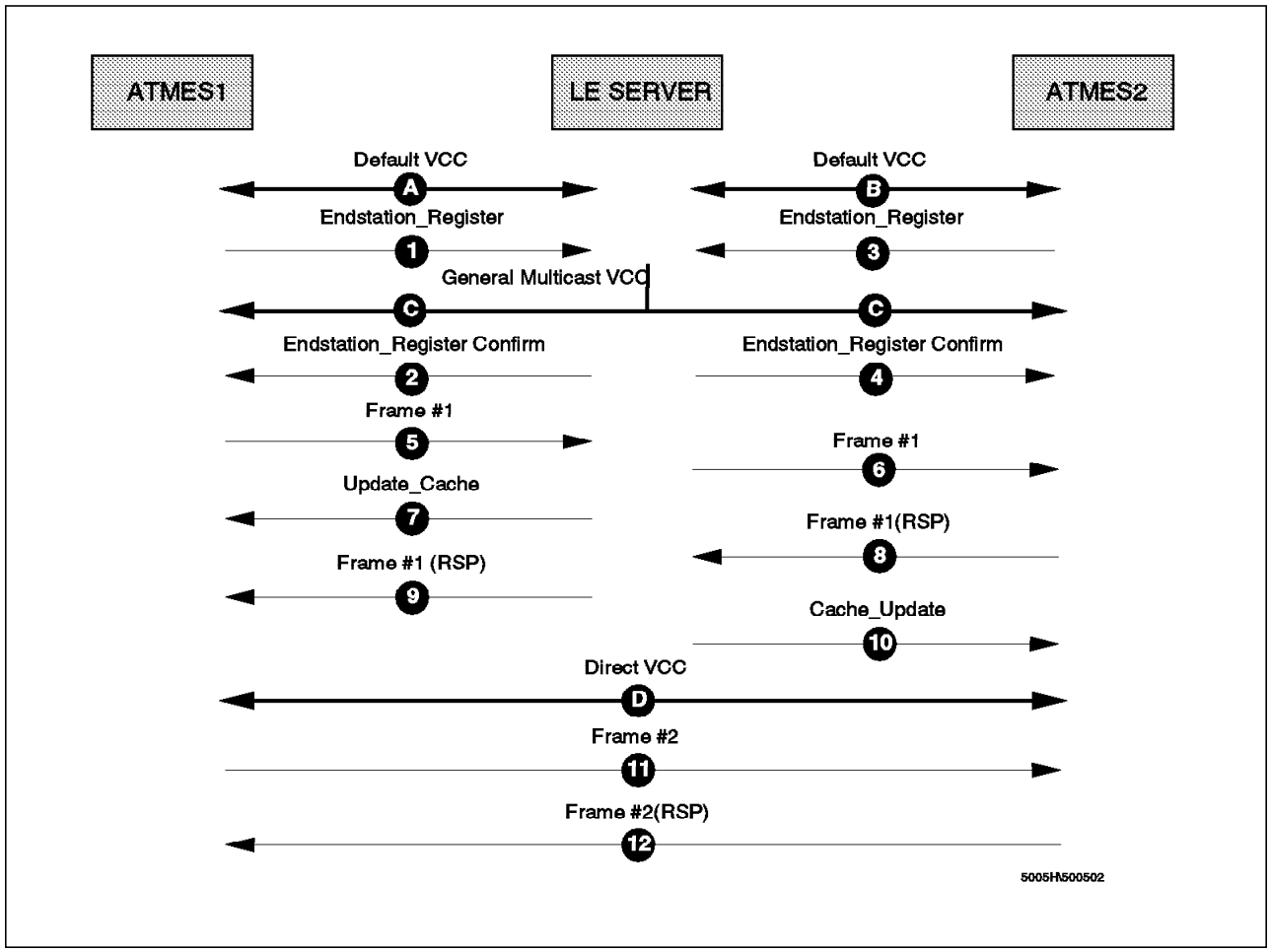



*Figure 150. 802.5 ATM Station-to-802.5 ATM Station Message Flow*

It is assumed that the ATM endsystems (ATMES1 and ATMES2) and the LAN emulation server have all completed their ILMI process and have initialized at both the physical and ATM layers. They already know their ATM address. Also, ATMES1 and ATMES2 know the ATM address of the LE server, as this is a configuration parameter specified by the administrator as part of the configuration of ATMES1 and ATMES2.

1. ATMES1 establishes the Default VCC **A** with the LE server.
2. ATMES1 sends a REGISTER_ENDSTATION **1** to the LE server using the Default VCC **A**.

   This frame contains information including the type of frame, registration or deregistration, type of multicast service required, and whether ATMES1 will establish Direct VCCs. It may optionally include the ATM address of ATMES1.
3. The LE server updates its SAAT forwarding/filtering table.
4. The LE server adds ATMES1 as a leaf to the General Multicast VCC **C**.
5. The LE server issues a REGISTER_ENDSTATION_CONFIRM **2** to ATMES1 using the Default VCC **A**.

   This frame contains information including a list of all successfully registered addresses, the emulated LAN identifier (ELID) and the originator identifier (OID).
6. ATMES2 establishes the Default VCC **B** with the LE server.
7. ATMES2 sends a REGISTER_ENDSTATION **3** to the LE server using the Default VCC **B**.

   This frame contains information including the type of frame, registration or deregistration, type of multicast service required, and whether ATMES2 will establish Direct VCCs. It may optionally include the ATM address of ATMES1.
8. The LE server updates its SAAT forwarding/filtering table.
9. The LE server adds ATMES2 as a leaf to the General Multicast VCC **C**.
10. The LE server issues a REGISTER_ENDSTATION_CONFIRM **4** to ATMES2 using the Default VCC **B**.

    This frame contains information including a list of all successfully registered addresses, the emulated LAN identifier (ELID) and the originator identifier (OID).
11. When the LE layer in ATMES1 receives a frame from the upper-layer protocol destined to the MAC address of ATMES2, it examines its Destination Address Association Table (DAAT) to determine whether it knows the ATM address of the target. In this scenario, it is assumed that the address is not in the ATMES1′s DAAT. Therefore, ATMES1 sends an all route explorer (ARE) frame addressed to ATMES2 **5** to the LE server over its Default VCC **A**.

    **Note:** In our discussion we have assumed that the upper-layer protocol already knows the MAC address of the destination. Normally, a broadcast frame is used by the upper-layer protocol to obtain this MAC address. For the sake of simplicity, we have not shown this step in our flows.
12. The LE server examines its SAAT to determine the ATM address of the requested target. In this case, since ATMES2 is registered with the LE

server, the LE server forwards the data frame 6 to the target (ATMES2) over the Default VCC B.

**Note:** Although ATMES1 has sent the frame as an ARE, the LE server does not broadcast the frame and sends it directly to ATMES2.

13. The LE server also sends an Update_Cache 7 to the ATMES1 with the ATMES2′s MAC to ATM address mappings so that ATMES1 can update its DAAT.
14. ATMES1 updates its DAAT.
15. ATMES2 sends the reply 8 to the LE server because it does not find the address of ATMES1 in its DAAT.
16. The LE server forwards the reply 9 to ATMES1.
17. The LE server also sends an Update_Cache message 10 to ATMES2 with the ATMES1′s MAC-to-ATM address mapping so that ATMES2 can update its DAAT.
18. ATMES2 updates its DAAT.
19. When the second frame is to be sent from ATMES1 to ATMES2, ATMES1 establishes a Direct VCC D with ATMES2 and sends the frame as a specifically routed frame (SRF) 11 over this connection.
20. The response from ATMES2 12 will be sent over the Direct VCC D.
21. All subsequent unicast frames between ATMES1 and ATMES2 are sent over the Direct VCC D.

# Appendix C. ATM Forum′s LAN Emulation

This chapter provides a brief introduction to LAN emulation as defined by the ATM Forum.

## C.1 ATM Forum′s LAN Emulation over ATM

As with IBM′s implementation of LAN emulation over ATM, the basic concept of the ATM Forum′s LAN emulation over ATM is to allow one or multiple virtual LANs over an ATM network to emulate either a token-ring (IEEE 802.5) or Ethernet (IEEE 802.3 and DIX) LAN. Again, these virtual LANs are not restricted to the local environment but can be established across any ATM network. Different virtual LANs implemented over one ATM network are completely independent from each other, and users connected to one virtual LAN *cannot* directly communicate with users connected to a different virtual LAN. A workstation may be connected, of course, to more than one virtual LAN. Such a workstation could then be a bridge or router to provide connectivity to the members of the virtual LANs to which it is connected.

Communication between different LAN emulation (LE) components can be performed using either switched or permanent VCCs or a mixture of both. When using permanent VCCs, of course, there is the burden for the system administrator to predefine all the necessary connections.

### C.1.1 LAN Emulation Components

An emulated LAN comprises the following components:

- One LAN emulation server (LES)
- One LAN emulation configuration server (LECS)
- One broadcast and unknown server (BUS)
- LAN emulation clients (LECs), such as user workstations, bridges, or routers

Users connect to the virtual LAN via LE clients, which request services through the LAN emulation user-to-network interface (LUNI). The three components (LE server, LECS, and BUS) may be distributed over different physical systems or may be grouped together in one system, but logically they are distinct functions. The LAN emulation services may be implemented in ATM intermediate systems (for example, switches) as part of the ATM network, or in one or several ATM endsystems.

As illustrated in Figure 151 on page 294, each LEC has to support a variety of VCCs across the LUNI for transport of control and data traffic.

© Copyright IBM Corp. 1996

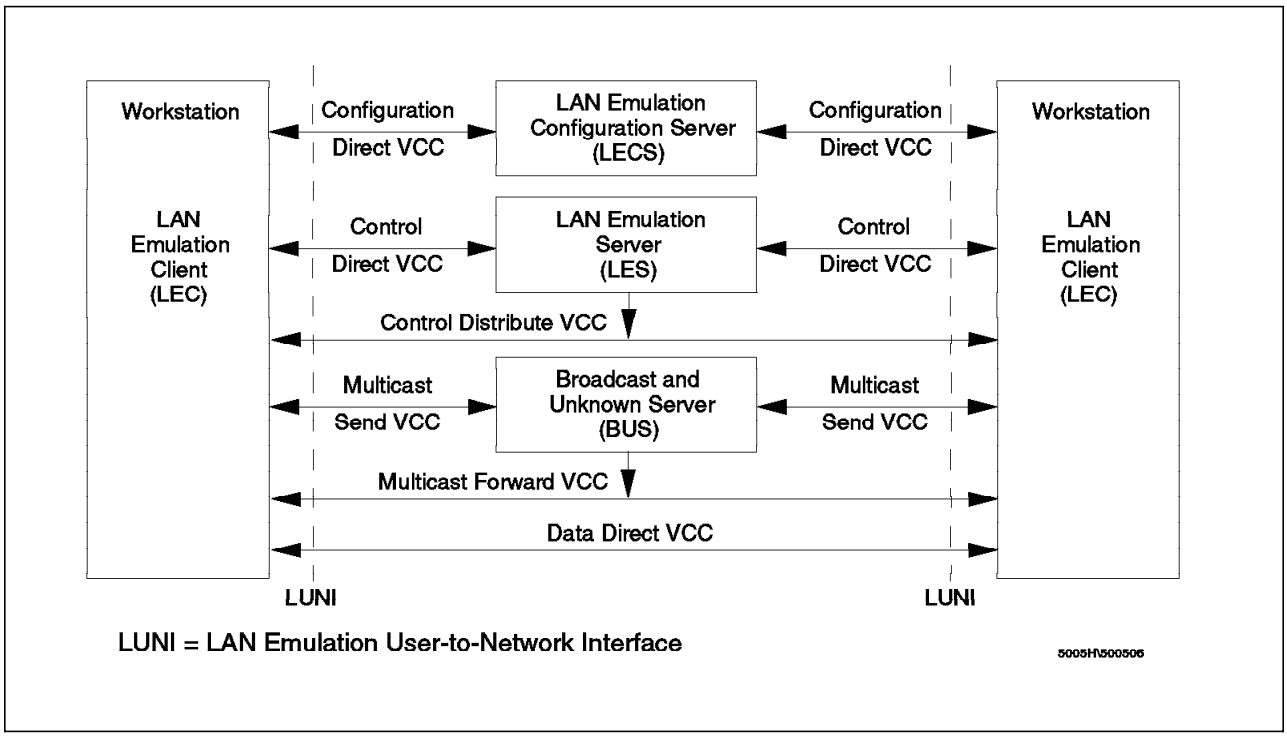



*Figure 151. LAN Emulation Components*

### C.1.1.1 LAN Emulation Server (LES)

The basic function of the LE server is to provide directory and address resolution services to the LECs of the emulated LAN. Each emulated LAN must have an LE server. An LE client registers the LAN address(es) it represents with the LE server. When an LE client wants to establish a direct connection with a destination LEC, it gets the destination's MAC address from the higher-layer protocols and has to ask the LE server for the destination's ATM address. The LE server will either respond directly (if the destination client has registered that address) or forward the request to other clients to find the destination.

An emulated token-ring LAN cannot have members that are emulating an Ethernet LAN (and vice versa). Thus, an instance of an LE server is dedicated to a single type of LAN emulation. The problems of translation bridging between different LAN types is not addressed in the ATM Forum's LAN emulation.

The LE server may be physically internal to the ATM network or provided in an external device, but logically it is always an external function that simply uses the services provided by ATM to do its job.

### C.1.1.2 LE Configuration Server (LECS)

The LECS assigns the individual LE clients to the different virtual LANs that can exist on top of the ATM network. During initialization, an LE client requests the ATM address of the LE server for the virtual LAN to which it should be connected. An LE client is not required to request this information from the LECS; an LE server's ATM address may be configured (system defined) in the LE client.

Using an LECS to assign clients to the different virtual LANs allows for central configuration and administration of multiple virtual LANs in an ATM network.

The LECS could make its decision to assign an LE server, for example, based on a client's ATM or MAC address according to a defined policy, or simply based on a system-defined database.

### C.1.1.3 Broadcast and Unknown Server (BUS)

The BUS handles data that, on a real LAN, is sent to the broadcast MAC address (X'FFFFFFFFFFFF'), all multicast traffic, and unicast data frames that are sent by a client before the destination ATM address has been resolved (by the LE server) and a Default VCC established. The BUS will then forward the data frames either over a Multicast Send VCC (the return path) or over the Multicast Forward VCC to the destination(s). Of course, a client could send unicast frames without ever trying to establish a direct connection to the destination; however, the ATM Forum's LAN emulation specification states that *if the client has insufficient resources to set up a Data Direct VCC, it must not continue to send frames to the BUS that it should have sent on the Data Direct VCC. In this case, it should tear down an existing Data Direct VCC to another client in order to free up resources to allow the new Data Direct VCC to be set up.*

The BUS works in a store-and-forward fashion, such that a frame's AAL-5 cell train has to be completely received by the BUS before the frame can be forwarded to its destination(s). This means that cells of different frames must not be intermixed. Also, all data frames sent through the BUS have to pass over the ATM network twice. This is not very effective when sending large files. Nevertheless, the BUS is essential for the functioning of the virtual LAN; it is the BUS that presents the image of a shared-media LAN over an ATM network.

### C.1.1.4 LAN Emulation Client (LEC)

Each workstation connected to the virtual LAN has to implement the LE layer (also called LE entity), which performs data forwarding and control functions such as address resolution and establishment of the various VCCs. The LE layer functions could be implemented completely in software, in hardware on a specialized LAN emulation ATM adapter, or in a combination of both. The layered structure of the LEC is shown in Figure 152.

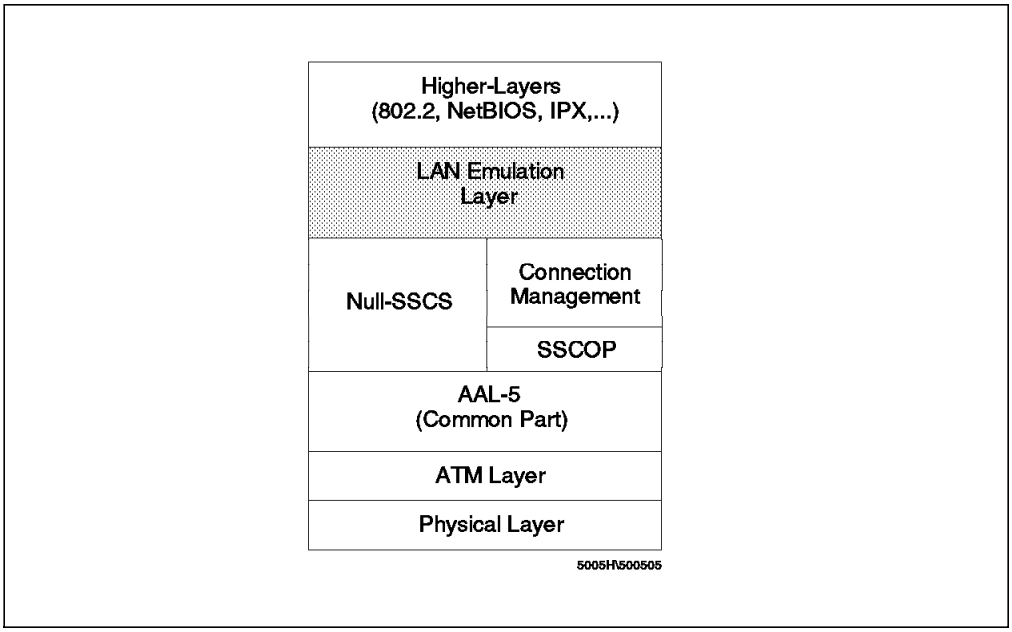



*Figure 152. LAN Emulation Client Functional Layers*

The LE layer provides the interface to existing higher-layer protocol support (such as IPX, IEEE 802.2 LLC, NetBIOS, etc.) and emulates the MAC-level interface of a real shared-media LAN (802.3/Ethernet or token-ring). This means that no changes are needed to existing LAN application software to use ATM services. The LE layer implements the LUNI interface when communicating with other entities in the emulated LAN.

The primary function of the LE layer is to transfer LAN frames (arriving from higher layers) to their destination either directly or through the BUS.

A separate instance of the LE layer is needed in each workstation for each different LAN or type of LAN to be supported. For example, if both token-ring and Ethernet LAN types are to be emulated within a single station, then you need two LE layers. In fact, they will probably only be different threads within the same copy of the same code, but they are logically separate LE layers. Separate LE layers would also be used if one workstation needed to be part of two different emulated LANs, both emulating the same LAN type (for example, token-ring). Each separate LE layer needs to have a different MAC address and must be attached to its own LE server, but it can share the same physical ATM connection (adapter).

## C.1.2 LAN Emulation VC Connections

Data transfer in the LE system (consisting of control messages and encapsulated LAN frames) uses a number of different ATM VCCs as illustrated in Figure 151 on page 294.

### C.1.2.1 Configuration and Control Connections

Control VCCs connect an LE client to the LE configuration server and the LE server, but they are never used for user data traffic. These connections may be permanent or switched and are established when an LE client connects to the virtual LAN.

**Configuration Direct VCC**

A bidirectional, point-to-point configuration Direct VCC may be established between an LE client and the LECS to obtain configuration information (for example, the LE server's ATM address).

**Control Direct VCC**

A bidirectional, point-to-point control Direct VCC must be established (and kept active) between each LE client and the LE server. This is used for the exchange of control traffic (for example, address resolution) between the LE client and the LE server.

**Control Distribute VCC**

The LE server may optionally establish a unidirectional control distribute VCC to distribute control information (for example, query for an unregistered MAC address) to all LE clients connected to the virtual LAN. This can be a point-to-point VCC to each LE client. If the ATM supports point-to-multipoint connections, then the LE server might instead establish one point-to-multipoint VCC to all LECs (of course, the clients will be added or deleted as leaves on this point-to-multipoint tree as they enter or leave the virtual LAN).

### C.1.2.2 Data Connections

Data connections are Direct VCCs from an LE client to other LE clients and to the BUS. They are used to carry user data traffic and never carry control traffic (except for a flush message for cleanup).

**Data Direct VCC**

For unicast data transfer between endsystems, Data Direct VCCs are set up through ATM signallng as bidirectional, point-to-point connections once the LE client has received the destination′s ATM address from the LE server.

As long as a Data Direct VCC has not been established (the protocol flows with the LE server may take some time), an LE client may send initial data frames through the BUS, but as soon as a Data Direct VCC is established, it has to be used and no data must be sent through the BUS. Since the LAN frames can be exchanged between two LE clients through either the BUS or using the Direct VCC (that is, there are two possible paths between the clients to exchange LAN frames), careful control is needed to ensure that when the Direct VCC becomes available, frames are not delivered out of sequence to the destination.

Data Direct VCCs stay in place until one of the partner LECs decides to end the connection based on installation options defining relevant timeouts.

**Multicast Send VCC**

During initialization, an LEC has to establish a bidirectional, point-to-point Multicast Send VCC to the broadcast and unknown server (the BUS′s ATM address is provided to the LEC by the LE server) and must keep this VCC established while being connected to the virtual LAN. This VCC is used by the LEC to send broadcast and multicast data frames. It is also used by the LE clients for sending unicast frames until a Data Direct VCC is established between the LE client and its partner. The BUS may use this VCC to send data (including multicast) to the LEC.

**Multicast Forward VCC**

When an LE client establishes its Multicast Send VCC to the BUS, the BUS learns about the new member of the virtual LAN. The BUS then will initiate signallng for the unidirectional Multicast Forward VCC to the LEC, which is used to forward data frames from the BUS to the LECs. This VCC can be either point-to-point or point-to-multipoint (of course, a Point-to-Multipoint VCC is more effective for multicast operations).

Every LEC must be able to receive data frames from the BUS (both over the Multicast Send VCC or the Multicast Forward VCC) but will not receive duplicates as the ATM Forum LAN emulation specification prevents the BUS from sending duplicate frames on these VCCs.

## C.1.3 LE Service Operation

In operation, the LAN emulation service performs the following functions:

**Initialization**

During initialization, the LE client discovers its own ATM address from the ATM switch, which is needed if the client is to later set up Direct VCCs. It obtains the LE server′s ATM address from the LECS and

establishes the Control VCCs with the LE server and the BUS. The BUS address is provided to the LE client by the LE server.

For more details of this function, refer to C.1.3.1, “Initialization” on page 299.

**Address Registration**

Clients use this function to provide address information to the LE server. An intelligent LE server may respond to address resolution requests if LE clients register their LAN destinations (MAC addresses, or for source routing IEEE 802.5 LANs only, route descriptors) with the LE server. The LAN destinations may also be unregistered as the state of the client changes. A client must either register all LAN destinations for which it is responsible, or join as a proxy.

For more details of this function, refer to C.1.3.2, “Address Registration” on page 303.

**Address Resolution**

This is the method used by an ATM client to associate a LAN destination with the ATM address of another client or the BUS. Address resolution allows clients to set up Data Direct VCCs to carry frames. This function includes mechanisms for learning the ATM address of a target station, mapping the MAC address to an ATM address, storing the mapping in a table, and managing the table.

For the server, this function provides the means for supporting the use of Direct VCCs by endstations. This includes a mechanism for mapping the MAC address of an endsystem to its ATM address, storing the information, and providing it to a requesting endstation.

For more details on this function, refer to C.1.3.3, “Address Resolution” on page 304.

**Connection Management**

In SVC environments, the LAN emulation client, the LAN emulation server, and BUS set up connections between each other using UNI signalling. This function is beyond the scope of this book.

**Data Transfer**

To transmit a frame, the sending LE layer must do the following:

- Decide on which of its VCCs (to destination LE client or BUS) a frame is to be transmitted
- Encapsulate the frame (AAL-5 is used)

It must also decide when to establish and release Data Direct VCCs. To do this it may need to access the LE server for address resolution purposes.

For more details of this function, refer to C.1.3.4, “Data Transfer” on page 305.

**Frame Ordering**

A sending LAN emulation client and a receiving client may have two paths between them for unicast frames, one via the BUS and one via a Data Direct VCC between them. A client is expected to use only one path at a time for a specific LAN destination, but the choice of paths may change over time. Switching between those paths introduces the

potential for frames to be delivered out of order to the receiving client. The out-of-order delivery of frames between two LAN endsystems is uncharacteristic of LANs, and undesirable in an ATM emulated LAN. The Flush protocol is therefore provided to ensure the correct delivery of unicast data frames.

For more details of this function, refer to C.1.3.5, “Frame Ordering” on page 305.

### C.1.3.1 Initialization

The initialization of a LAN emulation client comprises the definition of the initial state and then the following five operational phases:

1. LAN Emulation Configuration Server Connect Phase
2. Configuration Phase
3. Join Phase
4. Initial Registration Phase
5. Broadcast and Unknown Server Connect Phase

These phases must be completed in the specified order. If a LAN emulation client (LE client) is to achieve full interoperability, all of these phases must be completed successfully.

***Initial State:*** The initial state of a client is an implementation issue, but certain parameters are the subject of range constraints. Parameters have a minimum and maximum value and also a default value. If any parameter falls outside of its range, the result may be a poorly functioning or possibly a non-functioning emulated LAN.

If the initialization phase terminates abnormally, the LAN emulation client must return to the initial state and inform layer management.

***LAN Emulation Configuration Server (LECS) Connect Phase:*** In the LECS connect phase, the LE client establishes its session with the LAN emulation configuration server.

The LE client issues an ILMI Get or GetNext to obtain the ATM address of the LECS for that user-to-network interface (UNI). The LE client then attempts to establish a Configuration Direct VCC to the ATM address it has received from the ILMI process. If the connection to the ATM address of the LECS fails, the LE client issues an ILMI Get or GetNext request to determine if an additional LECS ATM address is available. If an address is returned, the LEC will then attempt to establish the Configuration Direct VCC to that address. However, in the event that an address is not returned, or the LE client cannot establish a Configuration Direct VCC, the LE client will use the well-known LECS ATM address X′47007900000000000000000000A03E00000100′ to open the Configuration Direct VCC to the configuration service. If the LE client cannot establish a VCC to the well-known ATM address of the LECS, then the well-known VCC of VPI=0 and VCI=17 (decimal) is used for the Configuration Direct VCC.

Once the LE client has established the Configuration Direct VCC, the LECS connect phase is complete.

**Note**

If the use of the LECS is not required, the LE client may execute a *null* LECS connect by using a preconfigured SVC or PVC to the LE server. If the LE client does not use the LECS connect procedures, it will also be unable to use the configuration phase procedures.

***Configuration Phase:*** In the configuration phase, the LE client obtains the ATM address of the LE server and optionally other configuration parameters necessary to prepare the LE client to enter the join phase (see "Join Phase" on page 300).

There are the following two types of control frames used in the configuration phase:

- LE_CONFIGURE_REQUEST

  This is issued by the LE client to the LECS along the Configuration Direct VCC to obtain configuration information.

- LE_CONFIGURE_RESPONSE

  This is issued by the LECS in response to the LE_CONFIGURE_REQUEST.

This phase can be performed by using static parameters configured into the LE client or by using the LE configuration protocol, which retrieves the parameters from the LECS. By using the LE configuration protocol, the LE client can be assigned to different emulated LANs and will learn the operating parameters of those LANs. The LECS will forward these operating parameters, as well as the ATM address of the LE server, to the LE client using the LE_CONFIGURE_RESPONSE.

The frames LE_CONFIGURE_REQUEST and LE_CONFIGURE_RESPONSE flow between the LE client and the LECS. Upon a successful conclusion of these stages, the LE client will have the following information copied to its variables:

- C2 LAN type
- C3 Maximum frame size
- C5 Emulated LAN name
- C9 Target ATM address - The address of the LE server

However, if the LE client does not receive an LE_CONFIGURE_RESPONSE within the specified time period (configurable), it can retry until the configured number of retries is exhausted, at which time the configuration phase fails. The client will then return to the beginning of the initialization phase.

***Join Phase:*** In the join phase, the LE client establishes its connection with the LE server. It will determine the operating parameters of the emulated LAN and is permitted, though not required, to register one MAC address/ATM address with the LE server.

The join protocol uses the following two type of frames:

- LE_JOIN_REQUEST

  This is sent by the LE client to the LE server as a request to be permitted to join an emulated LAN. It contains the following information on LE client variables:

  – C1 The primary ATM address

  - C2 The LAN type
  - C3 The maximum data frame size
  - C4 Whether the LE client is acting as a proxy for other unicast MAC addresses
  - C5 The name of the emulated LAN
  - C6 This parameter is optional; it is the local unicast MAC address

- LE_JOIN_RESPONSE

  This is sent by the LE server to the LE client in response to the LE_JOIN_REQUEST. It contains the following information:

  - C2 The LAN type
  - C3 The maximum data frame size
  - C5 The name of the emulated LAN
  - C14 The LE client identifier (LECID)

The LE client will set up a Control Direct VCC with the LE server or use a predefined Control Direct PVC if the LE client has been unable, or fails, to set up an SVC connection. The Control Direct VCC is a point-to-point, bidirectional connection. If the LE client is unable to establish either of these virtual circuits, it terminates the join phase. Once the Control Direct VCC is established, the LE client sends an LE_JOIN_REQUEST containing the information discussed above. Having sent the LE_JOIN_REQUEST, and if no LE_JOIN_RESPONSE has been received, the LE client will accept any request by the LE server to establish a Control Distribute VCC. However, if a response has been received, the LE client will assume no Control Distribute VCC is required and has the option to refuse an attempt by the LE server to make this connection. Further, if the LE client receives no response within the configured time allowed (the default is 120 seconds), it can retry until the configured number of retries is exhausted, at which time the join phase fails.

If the LE server does not receive the LE_JOIN_REQUEST on a new Control Direct VCC within a set time (the default is 120 seconds), it has the option of terminating the LE client's membership of the emulated LAN.

When an LE_JOIN_REQUEST is received, the LE server validates the request, in which case it has the option of setting up a Control Distribute VCC to the LE client. For the request to be successful, it must contain either an exact match with the LE server's LAN type or unspecified in the LAN type (LE client variable C2). The maximum frame size (LE client variable C2) must be either unspecified or greater than or equal to the LE server's maximum frame size.

The LE server will check for duplicate MAC addresses and duplicate ATM addresses already registered. If either are found, the join request will fail and no LE client identifier will be assigned. Once the conditions are met for a successful join, the LE server will issue an LE_JOIN_RESPONSE to the LE client, and the join phase of initialization is completed.

***Initial Registration Phase:*** When the LE client sends an LE_JOIN_REQUEST, it has the option of including one MAC address in LE client variable C6. If this occurs, the LE server will register this MAC address with the ATM address mapping as part of the join phase. The LE server will check for duplicate MAC

addresses and duplicate ATM addresses already registered. If either are found, the registration request will fail.

Following a successful join phase, the LE client has the option of registering more LAN destinations using the LE registration phase and the LE client variables C6 (or route descriptor fields using variable C8 if the LE client is a source-route bridge).

**Important**

The registration phase is optional. However, if the registration is performed, the LE client must register all of its local unicast MAC addresses; if the LE client is a token-ring emulated client, it must also register all of its route descriptors. All registrations must be complete before the LE client reaches an operational state. Once in the operational state, the LE client must not have in its variable C6 or C8 any LAN destination that has not successfully been registered with the LE server. Therefore, an LE client with only one unicast MAC address need not use the registration protocol, since it may implicitly register one MAC address during the join phase. This is because a join with a MAC address is functionally equivalent to a join without a MAC address, followed by a register with a MAC address.

Also, note that an LE client must either register all LAN destinations for which it is responsible or join as a proxy.

***Broadcast and Unknown Server Connect Phase:*** In the broadcast and unknown server (BUS) connect phase, the LE client establishes its connection with the BUS. In order to determine the ATM address of the BUS, the LE client issues an LE_ARP_REQUEST to the LE server to resolve the broadcast MAC address. The LE server will respond with the LE_ARP_RESPONSE containing the ATM address of the BUS.

The LE client uses this address to establish a bidirectional Multicast Send VCC. This VCC is used by the LE client to send all broadcast and multicast destination packets to the BUS. When the Multicast Send VCC is established, the BUS automatically establishes the Multicast Forward VCC. This VCC is unidirectional and can be either point-to-point or point-to-multipoint. It is used by the BUS to send multicast frames to LE clients. If the Multicast Send VCC cannot be established, the LE client may be removed from the emulated LAN.

If either party detects the release of the Multicast Send VCC, it will automatically release the Multicast Forward VCC. The LE client has the option of attempting to reestablish the connection for a configurable number of attempts. If these attempts fail, the LE client will terminate its membership of the emulated LAN.

If the LE client detects the intentional release of the Multicast Forward VCC, it will terminate its membership of the emulated LAN without any attempt at recovery. If it detects that this release was accidental, it may attempt to recover the connection for a configurable number of attempts after which it will terminate its membership of the emulated LAN.

In the case of the BUS detecting that the Multicast Forward VCC has been released, it will release that LE client's Multicast Send VCC and will make no attempt to reestablish the Multicast Forward VCC.

### C.1.3.2 Address Registration

The Address Registration protocol is used by an LE client wishing to register additional LAN destination and ATM address pairs not registered during the join phase.

The registration procedure can occur at any time after successfully joining an emulated LAN and is optional. The following are the four types of registration protocol frames:

- LE_REGISTER_REQUEST

  This frame is sent by the LE client to the LE server. It contains a request to register one LAN destination-ATM address pair.
- LE_REGISTER_RESPONSE

  This frame is sent by the LE server in response to the LE_REGISTER_REQUEST. It contains confirmation of a successful registration.
- LE_UNREGISTER_REQUEST

  This frame is sent by the LE client to the LE server. It contains a request to remove the registration of one LAN destination-ATM address pair.
- LE_UNREGISTER_RESPONSE

  This frame is sent by the LE server in response to the LE_UNREGISTER_REQUEST. It contains confirmation of a successful removal of the registration.

If an LE client has only one unicast MAC address, it need not use the registration protocol because every LE client can implicitly register one MAC address during the join phase. It should also be noted that a join with a MAC address is functionally equivalent to a join without a MAC address followed by a register with a MAC address.

All (un)register requests are issued over the Control Direct VCC. The LE server will respond to (un)registration requests over either the Control Distribute VCC or the Control Direct VCC. The LE server checks for duplicate MAC addresses and duplicate ATM addresses already registered. The rules governing registrations are as follows:

- An ATM address can only be associated with one LECID.
- An LECID can only be associated with one ATM address.
- An ATM address/LECID mapping can have multiple MAC addresses associated with it.
- A MAC address cannot be registered by more than one ATM address.
- An ATM address/LECID mapping cannot register a MAC address already associated with another ATM address/LECID mapping.

If a registration request does not fully comply with these rules then it will fail.

If an LE client requests the unregistration of a mapping it did not register, the LE server will send a successful response but will not actually unregister a mapping registered by another LE client.

### C.1.3.3 Address Resolution

The address resolution protocol is used by the LE client to associate a MAC destination address with the ATM address of another LE client. Address resolution makes the establishment of Data Direct VCCs possible so that data can be transferred directly between ATM endsystems.

The following are four types of address resolution frames:

- LE_ARP_REQUEST

  This frame is sent by an LE client to determine the ATM address of a given MAC address or route descriptor.
- LE_ARP_RESPONSE

  This frame is sent by the LE server in response to the LE_ARP_REQUEST to provide the information requested.
- LE_NARP_REQUEST

  This frame is sent by the LE client to advertise changes in remote address bindings.
- LE_TOPOLOGY_REQUEST

  This frame is sent by either the LE server or the LE client to indicate that network topology changes are in progress.

***LE_ARP Procedure:*** When the LE client has a frame to transmit to an unknown MAC destination address, it issues an LE_ARP_REQUEST on its Control Direct VCC to the LE server.

When the LE server receives the LE_ARP_REQUEST, it can take the following actions:

1. If the MAC destination address is known to the LE server, it can issue an LE_ARP_RESPONSE. This response will contain the ATM address of the LE client responsible for the LAN destination. If the LE_ARP_REQUEST contains the broadcast MAC address, the LE server responds with the ATM address of the BUS.
2. If the LAN destination is unknown to the LE server, it will do the following:
   a. Forward the LE_ARP_REQUEST to all LE clients using either the Control Direct VCC or the Control Distribute VCC
   b. Forward the LE_ARP_REQUEST to those LE clients that registered as proxy agents using either the Control Direct VCC or the Control Distribute VCC

If the LE server has forwarded the LE_ARP_REQUEST and then receives an LE_ARP_RESPONSE from an LE client, it adds the new mapping to its LE ARP cache and forwards the response to the LE client that originated the request. The LE client adds the new mapping to its LE ARP cache.

***LE_NARP Procedure:*** When an LE client believes that the mapping between a target LAN destination and target ATM address is no longer valid, it has the option of issuing an LE_NARP_REQUEST. This only applies to remote LAN-ATM address mappings and usually occurs because the LE client is now representing the target LAN destination at its source ATM address.

***LE_TOPOLOGY Procedure:*** When the topology of the network changes, either the LE client or the LE server will issue the LE_TOPOLOGY_REQUEST frame to inform the other members of the emulated LAN that these changes are underway. When an LE client receives an LE_TOPOLOGY_REQUEST, it will not use any entries for non-local LAN destinations from its LE ARP cache. If the LE server receives an LE_TOPOLOGY request, it will forward this to all LE clients.

If the LE client is an IEEE 802.1D transparent bridge, it issues an LE_TOPOLOGY_REQUEST for every configuration BPDU that it issued to the BUS. These LE clients also have the option of basing the LAN emulation topology change state on the spanning tree configuration BPDU instead of those received in the LE_TOPOLOGY_REQUEST.

### C.1.3.4 Data Transfer

Once the LE client has established the ATM location of the other party, it will establish a Data Direct VCC with that client. The calling client may already be sending unicast data frames to the BUS to forward to the client. If this is the case, the LE client will issue an LE_FLUSH_REGISTER (see C.1.3.5, “Frame Ordering” on page 305). The LE client will examine all frames received on any Data Direct VCC and if it finds any frames with its own LECID it will discard them. All frames will be filtered by LECID to allow only those frames required by the higher layer to be forwarded.

If the LE client receives a connection request from a client to which it already has a connection, it will accept the request, but will send frames only on the VCC that was initiated by the numerically lower ATM address. This may cause the duplicate VCC to be aged out. If the LE client detects that the Control Direct VCC or the Control Distribute VCC is released at any time other than the join phase, then the LE client must terminate its membership of the emulated LAN.

If the BUS receives a valid data frame from an LE client over a Multicast Send VCC, it will forward it using either the Multicast Forward VCC or the Multicast Send VCC.

***Delivery of Token-Ring Frames:*** When the LE client has a frame to send, it examines both the frame′s destination MAC address and the routing information field to determine where to send the frame and if it is required to issue an LE_ARP_REQUEST. An LE client emulating token-ring must support address resolution of route descriptors. If the location of the target LAN destination is unknown, it must send an LE_ARP_REQUEST to the LE server. It may also send the frame to the BUS.

### C.1.3.5 Frame Ordering

The LE client is able to send unicast frames to the same MAC address using the BUS and using a Data Direct VCC at different times. A mechanism is required to ensure that there is no possibility of delivering frames out of order by having two paths. That mechanism is known as the Flush protocol.

The flush message is a special frame identifiable as a non-data frame by having a reserved value X′FF00′ in the LAN emulation data frame header in place of the LECID of the sender.

The Flush protocol uses the following two frame types:

- LE_FLUSH_REQUEST

This frame is sent by the LE client to either the BUS using the Multicast Send VCC, or another LE client using the Data Direct VCC. It is used to ensure that all data frames in transit on the path have reached their destination LE client.

- LE_FLUSH_RESPONSE

  This frame is sent in direct response to the LE_FLUSH_REQUEST by the called LE client. The frame is sent to the LE server using the Control Direct VCC or the Control Distribute VCC for forwarding to the LE client who initiated the flush.

The Flush protocol is comprised of mandatory rules that must be applied to any component of the emulated LAN, whether they implement Flush protocol or not, and optional rules that must be applied if an LE client implements flush. The mandatory rules are discussed below.

The LE client sends LE_FLUSH_REQUEST over either the Data Direct VCC or the Multicast Send VCC. The client that receives the LE_FLUSH_REQUEST will always send the LE_FLUSH_RESPONSE to the LE server using the Control Direct VCC. The LE server will then forward this response to the LE client which originated the request. If the BUS receives an LE_FLUSH_REQUEST for another LE client, it will forward the request to that client using either the Multicast Send VCC or Multicast Forward VCC.

If an LE client chooses to implement flush, there are certain mandatory rules. These are discussed below.

The LE client sends an LE_FLUSH_REQUEST on either the Data Direct VCC or the Multicast Send VCC. This request must contain a transaction identifier, the source ATM address and the ATM address of the target LE client. The sending LE client cannot reuse the Data Direct VCC for the same LE client until it has received the LE_FLUSH_RESPONSE with matching transaction identifier. During this period, the sending LE client can either hold or discard data frames destined for a LAN destination. If the LE client does not receive a response within the required time, it will either discard any frames it is holding for that target or will send them down the old path. It can then issue another LE_FLUSH_REQUEST containing a new transaction identifier. Once the reply is received, it will send all held data frames on the new path before sending any additional frames.

### C.1.3.6 Termination Phase

In the preceding section where it is indicated that the LE client will terminate the phase and all the SVCs associated with the client (including all Control VCCs, Data Direct VCCs, and all VCCs to and from the BUS) must be released, the LE server and BUS will not attempt to reestablish the VCC for any reason.

# Appendix D. Classical IP over ATM (RFC 1577)

This chapter provides a brief introduction to Classical IP over ATM. For more details, please refer to RFC 1577.

## D.1 Overview of Classical IP over ATM

RFC 1577 defines the operation of IP and the address resolution protocol (ARP) over ATM. It also defines the initial application of ATM within Classical IP networks as a direct replacement for local area networks (Ethernet and token-ring) and for IP links that interconnect routers, either within or between administrative domains. The classical model here refers to the treatment of the ATM host adapter as a networking interface to the IP protocol stack operating in a LAN-based environment.

IBM has been delivering this functionality for the RISC System/6000 since March of 1994.

Some characteristics of the classical model, which are discussed in more detail below, are the following:

- The Logical IP Subnetwork (LIS)
- ATM Address Resolution Protocol - ATMARP
- ATMARP servers
- ATMARP clients
- ATMARP tables
- ATMARP packet formats
- ATMARP/InATMARP packet encapsulation

RFC 1577 does not describe the operation of ATM networks. Any reference to virtual connections, permanent virtual connections, or switched virtual connections applies only to virtual channel connections used to support IP and address resolution over ATM, and thus are assumed to be using AAL5.

### D.1.1 Logical IP Subnetwork Configuration

In the LIS scenario, each separate administrator configures hosts and routers within a closed logical IP subnetwork. Each LIS operates and communicates independently of other LISs on the same ATM network. Hosts connected to ATM communicate directly to other hosts within the same LIS. Communication to hosts outside of the local LIS is provided via an IP router. This router is an ATM endsystem attached to the ATM network that is configured as a member of two or more LISs. This configuration may result in a number of disjoint LISs operating over the same ATM network. Hosts of differing IP subnets communicate via an intermediate IP router even though it may be possible to open a Direct VCC between the two IP members over the ATM network. This is illustrated in Figure 153 on page 308.

© Copyright IBM Corp. 1996

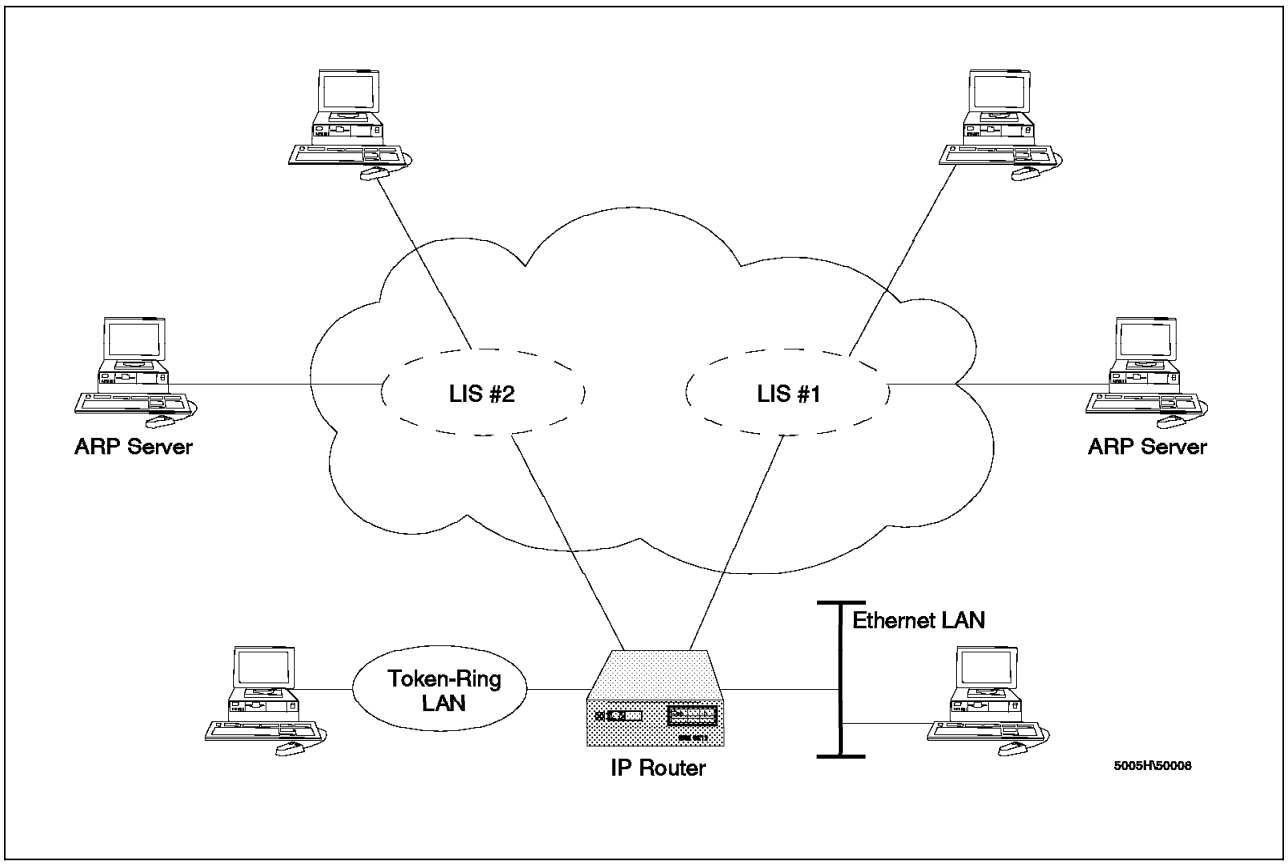



*Figure 153. Logical IP Subnetworks*

**Note**

Currently, the only IBM product that may be used as the IP router in the above configuration is an RS/6000 equipped with ATM and appropriate LAN adapters. Note that a single ATM adapter may be used to provide connections of up to ten LISs. In addition to the IP routing function, the ATM adapter installed in an RS/6000 can be used to provide ARP server functions as well. For example, in the above configuration an RS/6000 equipped with a single ATM adapter, as well as a token-ring and an Ethernet adapter, can be used to provide the IP routing function between LIS#1, LIS#2, token-ring LAN and Ethernet LAN, as well as providing the ARP server function for both LIS#1 and LIS#2.

The following are the requirements for all IP members (hosts, routers) operating in an ATM LIS configuration:

- The same IP network/subnet number and address mask.
- A direct connection to the ATM network.
- All members outside of the LIS are accessed via a router.
- A service for resolving IP addresses to ATM addresses via ATMARP and vice versa via InATMARP when using SVCs.
- A service for resolving VCCs to IP addresses via InATMARP when using PVCs.

- To be able to communicate via ATM with all other members in the same LIS, the virtual connection topology underlying the intercommunication among the members is fully meshed.

## D.1.2 Address Resolution

Address resolution within an ATM logical IP subnet makes use of the ATM address resolution protocol (ATMARP) and the inverse ATM address resolution protocol (InATMARP). ATMARP is the same protocol as the IP ARP protocol with extensions needed to support ARP in a unicast server ATM environment. InATMARP is the same protocol as the original IP InARP protocol but is applied to ATM networks. All IP stations must support these protocols as they are revised in future versions of the RFC. Use of these protocols differs depending on whether PVCs or SVCs are used.

### D.1.2.1 Switched Virtual Connections

To support IP over ATM Switched Virtual Connections (SVCs), a single ATMARP server must be located within the LIS. This server will be responsible for resolving the ATMARP requests of all IP members within the LIS.

To use the server, each member of the LIS must initiate a formal request to the clients in the LIS to initiate the ATMARP registration procedure with the server. To do this, an individual client connects to the ATMARP server using a Point-to-Point VCC. Once the connection between the client and the ATMARP server is established, the server transmits an InARP_REQUEST to determine the IP address of the client. The InARP_REPLY from the client contains the information necessary for the ATMARP server to build its ATMARP table cache. This information is used by the server to generate replies to the ATMARP requests it receives from the clients.

After registration with the ATMARP server, the client may send ATMARP requests to the ATMARP server in order to find out the ATM address of the other clients when it needs to communicate with them. Once the ATM address of the destination is known, the client must establish a Direct Point-to-Point VCC with the destination in order to communicate with it. Note that the ATMARP server is only used as a directory server and will never be used to forward the frames exchanged between two clients.

The ATMARP server mechanism requires that each client be administratively configured with the ATM address of the ATMARP server.

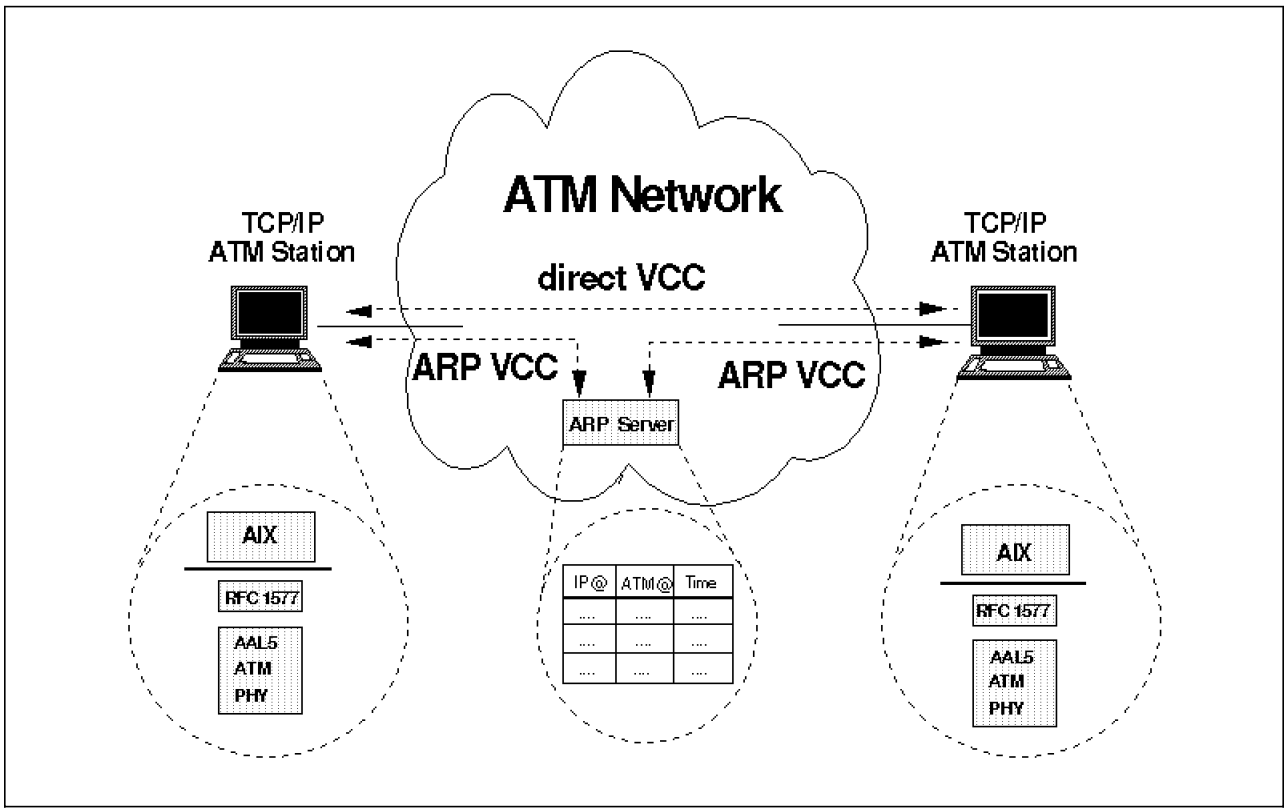



*Figure 154. Classic IP over ATM*

### D.1.2.2 Permanent Virtual Connections

When PVCs are used to connect workstations together, there is no need for an ARP server. However, each IP station must have a mechanism (for example, manual configuration) for determining what PVCs it has and in particular which PVCs are being used with LLC/SNAP encapsulation.

Each member of the LIS is required to use the Inverse ATM Address Resolution Protocol (InATMARP) on those VCs to determine the IP address of the station at the other end of the PVC (InARP_REQUEST). When the requesting station receives the InARP_REPLY, it may complete the ATMARP table entry and use the provided address information.

**Note:** Information learned via InARP_REPLY may be aged or invalidated under certain circumstances. It is the responsibility of each IP station supporting PVCs to revalidate ATMARP table entries as part of the aging process.

## D.1.3 ATMARP Server Operational Requirements

The ATMARP server accepts ATM calls/connections from other ATM endsystems. At call setup, and if the VCC supports LLC/SNAP encapsulation, the ATMARP server will transmit to the originating ATM station an InARP_REQUEST for each logical IP subnet the server is configured to serve. After receiving an InARP_REPLY, the server will examine the IP address and the ATM address. The server will add (or update) the ATM address and IP address map entry and timestamp into its ATMARP table. If the IP address received in the InARP_REPLY duplicates a table entry′s IP address, and the ATM address does not match the table entry′s ATM address and there is an open VCC associated with that table entry, the InARP_REPLY information is discarded and

no modifications to the table are made. ATMARP table entries persist until aged or invalidated. VCC call tear down does not remove ATMARP table entries.

The ATMARP server, upon receiving an ARP_REQUEST, will generate the corresponding ARP_REPLY if it has an entry in its ATMARP table; otherwise it will generate a negative reply (ARP_NAK). The ARP_NAK response is an extension to the ARMARP protocol and is used to improve the robustness of the ATMARP server mechanism. With ARP_NAK, a client can determine the difference between a catastrophic server failure and an ATMARP table lookup failure. The ARP_NAK packet format is the same as the received ARP_REQUEST packet format with the operation code set to ARP_NAK. That is, the ARP_REQUEST packet data is merely copied for transmission with the ARP_REQUEST operation code reset to ARP_NAK.

When the server receives an ATMARP request over a VC, where the source IP and ATM address match the association already in the ATMARP table and the ATM address matches that associated with the VC, the server may update the timeout on the source ATMARP table entry. For example, if the client is sending ATMARP requests to the server over the same VCC that it used to register its ATMARP entry, the server should examine the ATMARP requests and note that the client is still alive by updating the timeout on the client′s ATMARP table entry.

### D.1.4 ATMARP Client Operational Requirements

The ATMARP client is responsible for contacting the ATMARP server to register its own ATMARP information and to gain and refresh its own ATMARP entry/information about other IP members. This means, as noted above, that ATMARP clients must be configured with the ATM address of the ATMARP server. ATMARP clients need to do the following:

1. Initiate the VCC connection to the ATMARP server for transmitting and receiving ARP_REQUEST and InARP_REQUEST packets.
2. Respond to ARP_REQUEST and InARP_REQUEST packets received on any VCC appropriately.
3. Generate and transmit ARP_REQUEST packets to the ATMARP server and process ARP_REPLY and ARP_NAK packets from the server appropriately. ARP_REPLY packets should be used to build/refresh the client′s own ATMARP table entries.
4. Generate and transmit InARP_REQUEST packets as needed and process InARP_REPLY packets appropriately. InARP_REPLY packets will be used by the client to build/refresh its own client ATMARP table entries.
5. Provide an ATMARP table aging function to remove its own old ATMARP tables entries after a convenient period of time.

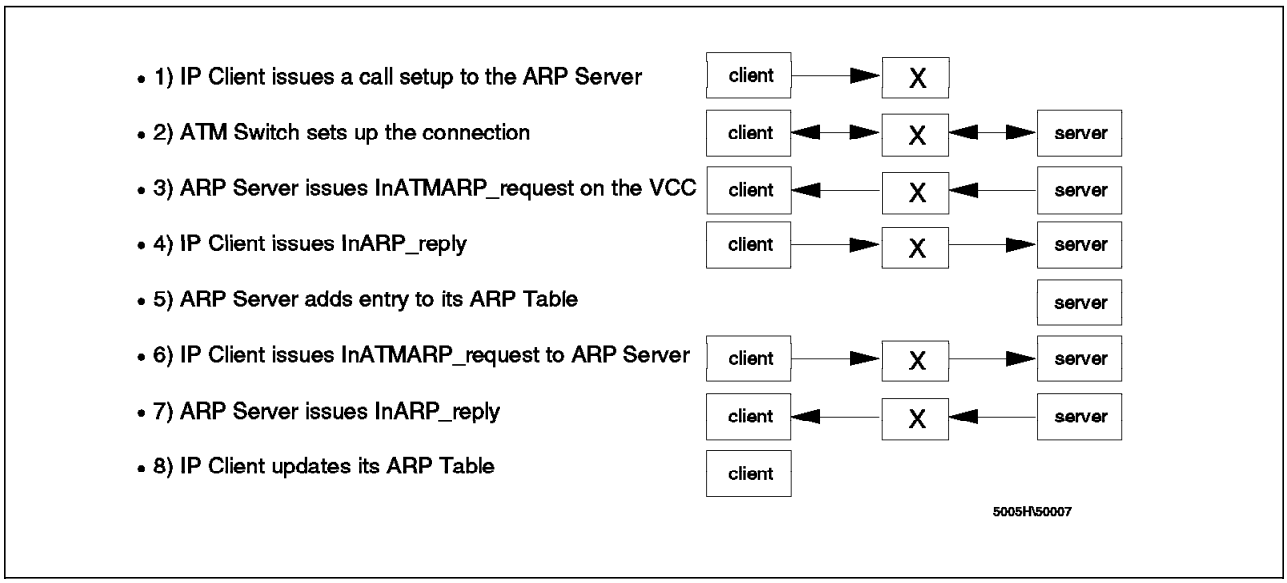



*Figure 155. IP Address Resolution Scenario*

**Note:** If the client does not maintain an open VCC to the server, the client must refresh its ATMARP information with the server at least once every 20 minutes. This is done by opening a VCC to the server and exchanging the initial InATMARP packets.

### D.1.5 ATMARP Table Aging

An ATMARP client or server must know about any open VCCs it has (permanent or switched), their association with an ATMARP table entry, and which VCs support LLC/SNAP encapsulation.

Client ATMARP table entries are valid for a maximum time of 15 minutes.

Server ATMARP table entries are valid for a minimum time of 20 minutes.

Prior to aging an ATMARP table entry, an ATMARP server generates an InARP_REQUEST on any open VCC associated with that entry. If an InARP_REPLY is received, that table entry is updated and not deleted. If there is no open VCC associated with the table entry, the entry is deleted.

When an ATMARP table entry ages, an ATMARP client invalidates the table entry. If there is no open VCC associated with the invalidated entry, that entry is deleted. In the case of an invalidated entry and an open VCC, the ATMARP client revalidates the entry prior to transmitting any non-address resolution traffic on that VCC. In the case of a PVC, the client validates the entry by transmitting an InARP_REQUEST and updating the entry on receipt of an InARP_REPLY. In the case of an SVC, the client validates the entry by transmitting an ARP_REQUEST to the ATMARP server and updating the entry on receipt of an ARP_REPLY. If a VCC with an associated invalidated ATMARP table entry is closed, that table entry is removed.

### D.1.6 ATMARP and InATMARP Packet Format

Internet addresses are assigned independently of ATM addresses. Each host implementation knows its own IP and ATM address(es) and responds to address resolution requests appropriately. IP members also use ATMARP and InATMARP to resolve IP addresses to ATM addresses when needed.

The ATMARP and InATMARP protocols use the same hardware type, protocol type, and operation code data formats as the ARP and InARP protocols. The location of these fields within the ATMARP packet are in the same byte position as those in ARP and InARP packets. A unique hardware type value has been assigned for ATMARP. In addition, ATMARP makes use of an additional operation code for ARP_NAK. The remainder of the ATMARP/InATMARP packet format is different from the ARP/InARP packet format.

### D.1.7 ATMARP/InATMARP Packet Encapsulation

ATMARP and InATMARP packets are to be encoded in AAL-5 PDUs using LLC/SNAP encapsulation.

The LLC value of 0xAA-AA-03 (three octets) indicates the presence of a SNAP header. The OUI value of 0x00-00-00 (three octets) indicates that the following 2 bytes are Ethertype. The Ethertype value of 0x08-06 (two octets) indicates ARP. The total size of the LLC/SNAP header is fixed at eight octets. This aligns the start of the ATMARP packet on a 64-bit boundary relative to the start of the AAL-5 CPCS-SDU.

### D.1.8 IP Broadcast and Multicast Address

ATM does not support broadcast and multicast addressing; therefore, there are no mappings available from IP broadcast/multicast addresses to ATM addresses.

## D.2 Switched Virtual Networking and LAN Emulation

Under IBM′s Switched Virtual Network strategy, the function of LAN emulation will be performed in Multiprotocol Switched Services (MSS). MSS has been developed to relieve the congestion caused by routers performing the distributed routing function. MSS consists of more than just distributed routing; additionally, it consists of enhanced LAN emulation, broadcast management, and VLAN support. This combined functionality should be viewed as a new access service of networking broadband services (NBBS), which provides NBBS functionality all the way to the endstation. More importantly, MSS is based upon ATM Forum′s LAN emulation and multiprotocol standards.

The MSS server will be a hardware-based solution, initially resident in the IBM 8260 or a stand-alone device. It will provide route calculation and directory services, as well as LAN emulation server and broadcast and unknown server (BUS) support. The MSS client will also handle frame forwarding and filtering, have a LAN emulation client, and store a destination cache as opposed to traditional routing tables. The combination of MSS server and MSS client addresses broadcast management within VLANs.

As mentioned earlier, a VLAN is a logical grouping of users and servers independent of their physical location. It may also be described as a single broadcast domain. That is, broadcast traffic must be limited to the members of the VLAN regardless of geographical location. In MSS, VLANs are supported by

protocol and address (both port and network). This, together with broadcast management support, drastically reduces network overhead and simplifies network administration.

Enhanced LAN emulation enables emulated domains to be larger than previous implementations have allowed. Additionally, MSS allows multiple LAN emulation servers per domain and allows a user to be a part of multiple LANs.

The ATM Forum has a subworking group focused on Multiprotocol over ATM (MPOA). This group focuses on layer 3 traffic (IP, IPX, etc.) support over ATM. They have described logical components called MPOA server and MPOA client. These logical components map to a router/MSS server and a router/MSS client.

# Appendix E. Special Notices

This publication is intended to help customers and IBM technical professionals to implement networks using Asynchronous Transfer Mode (ATM) technology. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM 8260 Intelligent Switching hub. See the PUBLICATIONS section of the IBM Programming Announcement for IBM 8260 Intelligent Switching hub for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of

© Copyright IBM Corp. 1996

including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

| | |
|---|---|
| AIX | AIXwindows |
| IBM | NetView |
| Nways | POWERserver |
| RISC System/6000 | RS/6000 |
| SP | SystemView |
| TURBOWAYS | 400 |

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo
are trademarks or registered trademarks of Microsoft Corporation.

Other trademarks are trademarks of their respective companies.

# Appendix F. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## F.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How To Get ITSO Redbooks” on page 319.

- *8260 Multiprotocol Intelligent Hub*, GG24-4370
- *Campus ATM Design Guidelines*, SG24-5002

## F.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

| CD-ROM Title | Subscription Number | Collection Kit Number |
|---|---|---|
| System/390 Redbooks Collection | SBOF-7201 | SK2T-2177 |
| Networking and Systems Management Redbooks Collection | SBOF-7370 | SK2T-6022 |
| Transaction Processing and Data Management Redbook | SBOF-7240 | SK2T-8038 |
| AS/400 Redbooks Collection | SBOF-7270 | SK2T-2849 |
| RISC System/6000 Redbooks Collection (HTML, BkMgr) | SBOF-7230 | SK2T-8040 |
| RISC System/6000 Redbooks Collection (PostScript) | SBOF-7205 | SK2T-8041 |
| Application Development Redbooks Collection | SBOF-7290 | SK2T-8037 |
| Personal Systems Redbooks Collection (available soon) | SBOF-7250 | SK2T-8042 |

## F.3 Other Publications

These publications are also relevant as further information sources:

- *ATM Control Point and Switch Module Installation and User's Guide*, SA33-0326
- *ATM 4-Port 100 Mbps Module Installation and User's Guide*, SA33-0324
- *Nways 8260 ATM 155 Mbps Flexible Concentration Module Installation and User's Guide*, SA33-0358
- *Nways 8260 ATM TR/Ethernet LAN Bridge Module Installation and User's Guide*, SA33-0361

© Copyright IBM Corp. 1996

# How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com.

## How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **ITSO4USA category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

© Copyright IBM Corp. 1996

## How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

| | **IBMMAIL** | **Internet** |
|---|---|---|
| In United States: | usib6fpl at ibmmail | usib6fpl@ibmmail.com |
| In Canada: | caibmbkz at ibmmail | lmannix@vnet.ibm.com |
| Outside North America: | dkibmbsh at ibmmail | bookshop@dk.ibm.com |

- **Telephone orders**

| | |
|---|---|
| United States (toll free) | 1-800-879-2755 |
| Canada (toll free) | 1-800-IBM-4YOU |
| Outside North America | (long distance charges apply) |
| (+45) 4810-1320 - Danish | (+45) 4810-1020 - German |
| (+45) 4810-1420 - Dutch | (+45) 4810-1620 - Italian |
| (+45) 4810-1540 - English | (+45) 4810-1270 - Norwegian |
| (+45) 4810-1670 - Finnish | (+45) 4810-1120 - Spanish |
| (+45) 4810-1220 - French | (+45) 4810-1170 - Swedish |

- **Mail Orders** — send orders to:

| | | |
|---|---|---|
| IBM Publications | IBM Publications | IBM Direct Services |
| Publications Customer Support | 144-4th Avenue, S.W. | Sortemosevej 21 |
| P.O. Box 29570 | Calgary, Alberta T2P 3N5 | DK-3450 Allerød |
| Raleigh, NC 27626-0570 | Canada | Denmark |
| USA | | |

- **Fax** — send orders to:

| | |
|---|---|
| United States (toll free) | 1-800-445-9269 |
| Canada | 1-403-267-4455 |
| Outside North America | (+45) 48 14 2207 (long distance charge) |

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

| | |
|---|---|
| Redbooks Home Page | http://www.redbooks.ibm.com |
| IBM Direct Publications Catalog | http://www.elink.ibmlink.ibm.com/pbl/pbl |

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

## IBM Redbook Order Form

**Please send me the following:**

| Title | Order Number | Quantity |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

- **Please put me on the mailing list for updated versions of the IBM Redbook Catalog.**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number

- Credit card number

Credit card expiration date Card issued to Signature

**We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.**

**DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.**

# Index



© Copyright IBM Corp. 1996

## O



## P



## Q



## R



## S



## T



## U



## V

Printed in U.S.A.