Restricted Rights Notice
(Applicable to licenses to the United States government only.)
This software and/or user documentation is/are provided with RESTRICTED AND LIMITED
RIGHTS. Use, duplication, or disclosure by the government is subject to restrictions as set forth in
FAR 52.227-14 (June 1987) Alternate III(g)(3) (June 1987), FAR 52.227-19 (June 1987), or
DFARS 52.227-7013(c)(1)(ii) (June 1988), and/or in similar or successor clauses in the FAR or
DFARS, or in the DOD or NASA FAR Supplement, as applicable. Contractor/manufacturer is
Aprisma Management Technologies, Inc. In the event the government seeks to obtain the software
pursuant to standard commercial practice, this software agreement, instead of the noted
regulatory clauses, shall control the terms of the government's license.

Virus Disclaimer
Aprisma makes no representations or warranties to the effect that the licensed software is virus-
free.
Aprisma has tested its software with current virus-checking technologies. However, because no
antivirus system is 100 percent effective, we strongly recommend that you write-protect the
licensed software and verify (with an antivirus system in which you have confidence) that the
licensed software, prior to installation, is virus-free.

Contact Information
Aprisma Management Technologies, Inc.
273 Corporate Drive
Portsmouth, NH 03801
Phone: 603-334-2100
U.S. toll-free: 877-468-1448
Web site: http://www.aprisma.com
Contents

Introduction................................................................................................. 6
Prerequisites .................................................................................................... 7
ATM Circuit Manager Modeling Schemes ......................................................... 7
  Modeling Scenario #1 .............................................................................. 7
  Modeling Scenario #2 .............................................................................. 8
  Modeling Scenario #3 .............................................................................. 8
Required MIBs .............................................................................................. 8
SpectroGRAPH Models .................................................................................. 9
  ATM_Cloud ................................................................................................. 9
  ATM_Network ............................................................................................ 10
  ATM Interface ............................................................................................ 11
  ATMVcILink ............................................................................................... 11
  UnmanAtmLink .......................................................................................... 12
Modeling Paradigms ..................................................................................... 12
  The Older Methodology .......................................................................... 12
  The New Modeling Paradigm .................................................................... 14
Terminology ................................................................................................... 15

Modeling the ATM Circuits ........................................................................ 17
ATM and Frame Relay ................................................................................... 23

Importing Connectivity Information ............................................................ 25
  Creating a Comma-Delimited Input File ................................................... 25
  Using the wanimport Tool ....................................................................... 26
    Moving the wanimport Tool to the Third-Party Host .............................. 26
    Running the wanimport Tool .................................................................. 28

Using ATM Circuit Manager ......................................................................... 29
  Configuring and Monitoring Link Models ................................................. 29
    Starting the ATM Logical Connection View ........................................ 30
      Managed Links ....................................................................................... 32
      Unmanaged Links ............................................................................... 34
Introduction

This document describes the ATM Circuit Manager application including the modeling scheme by which SPECTRUM models ATM circuits, the functionality provided by this modeling scheme, and how application is used to manage ATM networks.

Asynchronous Transfer Mode (ATM) is based on the transmission of fixed-length (53-byte) cells of data. Contrast this with an Ethernet LAN, which transmits variable-length packets ranging in size from 64 to over 1500 bytes of data. ATM’s use of small, fixed-length data cells allows for improved traffic management and traffic shaping.

ATM is a connection-oriented network communication architecture, which generally means that it transmits data through pre-established virtual channels (circuits), similar to telephone calls. Virtual channels may be established automatically by Switched Virtual Circuit (SVC see Page 15) signaling or they may be set up manually by the network administrator to form Permanent Virtual Circuits (PVCs).

Each ATM cell contains a five-byte header and 48 bytes of payload. The header includes a Virtual Path Identifier (VPI see Page 15) and a Virtual Channel Identifier (VCI see Page 15). These identifiers are used by ATM switches to determine the correct channels to transmit particular cells. Transmission is controlled by statistical multiplexing, which awards bandwidth (channels) to devices ready to send data on a first come, first served basis.

The combination of small, fixed-length data cells and the efficient use of bandwidth (among other things) allow ATM switches to communicate time-critical video and audio data along with other computer data across the ATM network. In an end-to-end transmission across a mixed LAN/ATM/LAN network, packets transmitted by a LAN workstation to an ATM switch are segmented into cells for high speed transmission through ATM channels. At the receiving end, cells are reassembled into packets for use by another LAN workstation.
Prerequisites

You must be familiar with ATM network technology before using the ATM Circuit Manager application and this document. This document is not meant to be an explanation of ATM and networking. In order to set thresholds and determine connections, you must be completely familiar with the service provider's contract and with the ATM network's topology.

ATM Circuit Manager Modeling Schemes

There are three different modeling schemes that can be used to model ATM networks. These schemes are dependent on the type of ATM network that a SPECTRUM customer has. There are three ATM network types:

1. Customers who own and administer the switches that comprise the ATM fabric.
2. Customers who use an ATM service provider to provide wide-area connectivity and have no management access to the ATM switches.
3. Customers who own their own ATM switches and connect to a service provider's network.

Modeling Scenario #1

Customers who own their own ATM switches will use an ATM_Network model to represent their switched fabric. Switch models will appear within the ATM_Network model and will have a Collects association with that model. Client models will have an Is_Adjacent_to association with the ATM_Network model.
Introduction

Modeling Scenario #2
If an ATM service provider is used to provide wide-area connectivity to remote sites, there is no management access to the service provider’s ATM switches. The ATM clients must provide all the data to monitor the ATM circuits. To facilitate the modeling of this type of network, a new SPECTRUM model type, ATM_Cloud, has been introduced. All ATM interfaces that connect to the service provider’s network will have a Connects_to association with the ATM_Cloud model. This modeling association can be established manually by copying and pasting the ATM_Cloud model onto those interfaces.

Modeling Scenario #3
If the SPECTRUM user owns their own local ATM switches but still connects to a service provider’s network for wide-area access, interface models of ATM switches will be connected to the ATM_Cloud model. SPECTRUM intelligence will automatically create VPL (Virtual Path Link see Page 15) and VCL (Virtual Channel Link see Page 16) models to represent the VPLs and VCLs that run over an interface connected to the ATM_Cloud model.

Required MIBs
When managing physical connections, SPECTRUM queries the routers in the network for interface status and statistics. When managing virtual connections, SPECTRUM queries the network’s switches and their clients for VCL and VPL status and statistics.

Management of logical connections requires that the devices making the connections support the necessary SNMP MIBs. To fully utilize the performance monitoring and threshold alarming functionality of the ATM Circuit Manager, the following MIBs must be supported by the managed devices:

- RFC1695 which provides up/down status information for each ATM link. The RFC1695 MIB is necessary for fault notification.
• ATM2 MIB which provides in cells/out cells counter statistics for each ATM link. Without the cell counters inherent in this MIB, no thresholds can be set for a circuit and no performance information will be available.

Note: The ATM2 MIB is currently an internet draft that is not yet an RFC. However, the ATM2 MIB is necessary to monitor the performance of the VCLs/VPLs.

• In addition, SPECTRUM supports the Cisco CA-CONN-MIB (found on Lightstream switches), and the Fore Switch MIB. Either of these MIBs provide the statistics necessary to monitor the performance of the ATM links.

SpectroGRAPH Models

This section provides descriptions of the new model types and corresponding models that have been developed to manage ATM circuits with SPECTRUM. Using these new models to manage your ATM network is described in detail in the Using ATM Circuit Manager (Page 29) section of this document.

ATM_Cloud

The ATM_Cloud model type is used in modeling situations where an ATM service provider is used to provide wide-area connectivity to remote sites and there is no management access to ATM switches. The ATM clients must provide all the data to monitor the ATM circuits. The ATM_Cloud model type also provides access to the ATM Logical Connection view.
**ATM_Network**

The ATM_Network model type represents a completely managed ATM switched fabric. It also provides access to the ATM Logical Connection view.
**ATM Interface**

This model provides access to a Device Topology view containing the sub-interface models. *Connects_to* associations with the ATM Interface model represent the physical, interface-to-device, or interface-to-ATM_Cloud connectivity.

**ATMVclLink**

This "link" model represents an endpoint of a virtual connection. Each VCL or VPL acts, in many ways, like an interface model. SPECTRUM retrieves circuit status and statistics for the connection by monitoring these endpoint models. SPECTRUM logs statistics from these models which can then be used to generate reports. SPECTRUM generates alarms based on the status of these models. For example, if the load of an ATM "link" model exceeds a predefined threshold, SPECTRUM alerts the user with an alarm. Management of PVPs (Permanent Virtual Paths see Page 15) and PVCs (Permanent Virtual Circuits see Page 15) in SPECTRUM is achieved by polling and logging attributes of the VPL and VCL models.
**UnmanAtmLink**

The UnmanAtmLink model represents an unmanageable ATM link. Some ATM paths or circuits may be manageable from only one end-point. The device on one side may not have an SNMP Agent or may be inaccessible for some other reason. For these links, the UnmanAtmLink model can be created and associated to the end-point that does have management. This allows SPECTRUM to manage this type of link.

**Modeling Paradigms**

**The Older Methodology**

Before ATM Circuit Manager, it was impossible to accurately represent the connectivity of switches and/or routers in a service provider's network. In a fully or partially meshed network, each physical ATM Interface may have logical connections with many other ATM devices. However, previous SPECTRUM modeling functionality only allowed a single connection per interface model.

For this reason, many SPECTRUM users created a WA_Link model to represent the ATM service provider's network and all ATM interfaces were connected to a WA_Segment model collected by the WA_Link model (Figure 1). This provided an adequate representation of the physical connectivity unless there were managed switches between some of the clients. The true data relay paths between the clients were not known and proper fault isolation could not be guaranteed.
Figure 1: The Older Methodology
The New Modeling Paradigm

The ATM Circuit Manager creates models to represent each virtual path link and virtual channel link. Each virtual link model may be associated with another link model to indicate the “logical connectivity” of the interface through the ATM Logical Connection view. The physical connectivity is still indicated by a Connects_to association with the ATM interface model on the left side of the view and the connected device model on the right.

Figure 2: The New Modeling Paradigm
Terminology

Path
A large communications pipe that pre-allocates bandwidth and allows for greater flexibility in establishing PVCs. A defined amount of bandwidth is leased from a service provider and you can establish as many PVCs as necessary within the limits of that bandwidth.

Channel
A data transmission link between two or more points.

Permanent Virtual Path (PVP)
A logical communications path that has a defined amount of leased bandwidth. This path is maintained at all times even if it is not always in use.

Permanent Virtual Circuit (PVC)
A logical connection that is manually created by a network administrator. This connection is maintained at all times even if it is not always in use. PVCs can exist without being part of a PVP.

Switched Virtual Circuit (SVC)
A temporary connection that is established and maintained only for the duration of a data transfer session.

Virtual Path Identifier (VPI)
The field of a cell header that contains the address of the virtual path.

Virtual Channel Identifier (VCI)
The field of a cell header that contains the address of the virtual channel.

Virtual Path Link (VPL)
A unidirectional method of transport for ATM cells that begins at the point where a VPI value is assigned and ends at the point where the VPI value is translated or removed.
Virtual Channel Link (VCL)
A unidirectional method of transport for ATM cells that begins at the point where a VCI value is assigned and ends at the point where the VCI value is translated or removed.
Modeling the ATM Circuits

This section describes the procedures for modeling an ATM network in SPECTRUM.

As previously stated, there are three possible modeling schemes depending on the user's ATM environment:

1. The entire switched fabric is owned and managed by the customer (Figure 3)
2. All channels are leased through a service provider's network (Figure 5)
3. There is a local area ATM network owned and managed by the customer and additional wide-area channels that are leased through a service provider's network (Figure 6)

There are three procedures for modeling an ATM network with SPECTRUM:

1. Creating and resolving all models manually.
2. Using the discovery process provided by the ACMAsciimodelingApp (Page 41).
3. Incrementally adding new circuits if you already have a fully-meshed, switched network.

If the customer is managing the switched fabric, there will be one or more ATM_Network models in the SPECTRUM topology. The ATM switches will be collected by the ATM_Network models and the ATM clients will be adjacent to the ATM_Network model (Figure 3).

Auto Discovery maps the physical connectivity between ATM switches and places these switch models inside ATM_Network container models. Auto Discovery also places SW_Link models between connected ATM Interface models. If AutoDiscovery does not fully map the physical ATM connectivity of your network, including switch-to-switch and router-to-switch connections, you must do so manually or by re-running AutoDiscovery. It is also possible to import connectivity information via a
comma-delimited ASCII text file. For more information on doing this, see *Importing Connectivity Information* on Page 25.

In order to model the ATM circuits correctly, the physical connectivity must be modeling first. If you model your ATM network manually, you can connect the ATM Interface models directly to the switch or client models but they must be collected by an ATM_Network model to get the benefit of the Logical Connection view.

The ATM circuits go from one client, through the ATM_Network, to another client. The links within the switched fabric itself will not be managed, unless the user specifically requests it.

**Figure 3: Modeling a Completely Owned Network**
Every ATM interface that connects to the service provider's network must be connected manually to the ATM_Cloud model. This can be done by copying the ATM_Cloud and pasting it onto the ATM interface in the Device Topology view of each device.

1. In the Topology view, select **Edit** from the File menu.
2. Highlight the ATM_Cloud model.
3. Select **Copy** from the Edit menu.
4. Select **Close** Edit from the File menu.
5. Navigate into the ATM device's Device Topology view.
6. Select **Edit** from the File menu.
7. Select **Paste** from the Edit menu.
8. Paste the ATM_Cloud model onto the ATM Interface model.

**Note:** If the customer is leasing channels through a service provider's network, there will be an ATM_Cloud model in the SPECTRUM topology.
Figure 4: Pasting the ATM_Cloud Model into a Device Topology View
It is possible for a customer to have a local ATM network and use ATM wide-area services from a service provider. In this case, interfaces from some of the switches may be connected to the ATM_Cloud and other switches go from client to client through the ATM_Network. This will make the ATM_Network adjacent to the ATM_Cloud. This is the only scenario in which the virtual link of a switch will be managed as an endpoint of a circuit. (In other cases, the virtual links of the clients, not switches are managed).
Figure 6: Modeling a Completely Owned ATM LAN and Leased ATM Wide-Area Links
ATM and Frame Relay

It is possible to have a hybrid ATM network with both ATM and Frame Relay interfaces. A customer may be using a completely leased network through a service provider and, depending on the type of applications they are using, there could be a mixture of ATM and Frame Relay interfaces on either side of the leased network.

For example, a customer could be running a graphical medical software application which would require using ATM and ATM interfaces in one part of their network. However, a remote office might be using Frame Relay for communications and would not require ATM. Signals transmitted from an ATM interface going through the service provider’s network would be converted to Frame Relay by a translational bridge before being received by the Frame Relay interface or Frame Relay signals could be converted to ATM (Figure 7).

Note:
To use Frame Relay with the ATM Circuit Manager Application, you must have purchased and installed the SPECTRUM Frame Relay Manager Application. Contact SPECTRUM Support for more information.
The modeling procedure for this scenario would be identical to the procedure for modeling a completely leased network, as described previously.
Importing Connectivity Information

This section describes using the wanimport tool to import ATM connectivity information contained in a comma-delimited ASCII text file.

ATM connectivity information can be imported into SPECTRUM using the wanimport tool. To use this tool, you must create a comma-delimited ASCII text file that defines the connections you would like to create. The wanimport tool is then executed referencing this file and the appropriate connections between port models are created within the SpectroSERVER database.

Creating a Comma-Delimited Input File

A comma-delimited ASCII text file is used to define the connections that will be imported into SPECTRUM. This file can specify connections between two ATM circuits, two Frame Relay circuits, or an ATM and a Frame Relay circuit. You have the option to specify that a live pipe be created in the SpectroGRAPH to represent the connection. Multiple connections can be specified in the same input file. The device models involved in these connections must already exist in SPECTRUM.

Following is the format for the input file:

\(<Device\_IP>, <OID>, <Device\_IP>, <OID>, <CircuitName>, <CircuitID>, <Pipe>\)

Device_IP is the IP address of each device involved in the connection. This parameter is required for each device.

OID is the OID instance of frCircuitTable, atmVclTable or atmVpiTable to specify the circuit link on the device. This parameter is required for each device.
CircuitName is an optional parameter specifying the name of the circuit involved.

CircuitID is an optional parameter specifying the ID of the circuit involved.

Pipe is an optional parameter with two possible values, CREATE_PIPE or NO_CREATE_PIPE. If the value is set to CREATE_PIPE, live pipes will be created between the connections specified. If the value is set to NO_CREATE_PIPE, live pipes will not be created between the connections specified. If no value is specified for this parameter, a default value of CREATE_PIPE is assumed.

The following example shows a line from an input file that specifies the connection between a Frame Relay circuit and an ATM circuit. In this example a pipe is created between the two ports.

172.19.57.93, 4.161, 10.253.32.225, 5.0.17, FR<->ATM, 12345, CREATE_PIPE

This example shows a line from an input file that specifies the connection between the ATM circuits. In this example a pipe is not created.

10.253.9.12, 8.0.35, 10.253.32.225, 5.0.37, atm link, NO_CREATE_PIPE

Using the wanimport Tool

Once you have created the input file, use the wanimport tool to send the connectivity data into the SpectroSERVER database. The wanimport tool is a command line utility that is located in SPECTRUM’s SS-Tools directory. This tool can be run from the SpectroSERVER or from a third-party host. In order to run the tool from a third-party host, follow the instructions on moving the tool outlined below.

Moving the wanimport Tool to the Third-Party Host

To run the wanimport tool from the third-party host, you must move the wanimport tool and all of its support files to the third-party host.
SPECTRUM provides a script that packs up the wanimport tool so that it can be sent via FTP to the third-party host. The script ensures that the relative directory structure of the tool and its support files is retained when the files are moved.

The following steps show you how to move these files:

1. On the SpectroSERVER, check to make sure the environmental variable SPECROOT is set to the SPECTRUM installation directory path.

2. Run the script that packs up the tool and its support files. The script can be found in the SPECTRUM's SS-Tools directory and is called packtool.pl.

   To run the script from the Bash shell or other Unix shell, type:

   ```
   <Spectrum_Installation_Path>/SS-Tools/packtool.pl wanimport
   ```

   Where `<Spectrum_Installation_Path>` is the directory structure where Spectrum is installed on your SpectroSERVER.

3. The script generates an executable file called wanimport_bundle (Unix) or wanimport_bundle.exe (Windows) that contains the wanimport tool and all of its support files.

4. On the third-party host, create a new directory to unpack the wanimport tool and its support files, i.e. `/disk/Spectrum`.

5. FTP the wanimport_bundle or wanimport_bundle.exe file from the SpectroSERVER to the third-party host, and place it in the directory created in step 4. Be sure to use binary mode during the FTP process.

6. Once the file is on the third-party host, execute the file from the DOS, Bash, or other Unix shell. The wanimport tool and its support files will unpack into the appropriate directory structure. The wanimport tool can now be run from this host machine.

   **Note:** Both servers involved in this process must be running the same operating system. You cannot pack the tool on a Windows server and unpack it on a Unix machine or vise versa.
Running the wanimport Tool

In order to run the wanimport tool from a machine other than the SpectroSERVER, you must set the SPECROOT (Unix) or SPECPATH (Windows) environmental variable on this machine equal to the path to the directory where you placed the wanimport_bundle or wanimport_bundle.exe file.

For example, if you are working in the Unix environment and placed the wanimport_bundle file in /disk/Spectrum as in step 4 in Moving the wanimport Tool to the Third-Party Host (Page 26), then you must set SPECROOT=/disk/Spectrum. If you are working in the Windows environment and placed the wanimport_bundle.exe file in C:\disk\Spectrum, then you must set SPECPATH=C:\disk\Spectrum.

The import tool command takes 4 arguments, two are required and two are optional.

**Windows Syntax:**

wanimport.bat -vnm <vnm_name> -i <input_file>
[-o <outputfile>] [-debug]

**Unix Syntax:**

wanimport -vnm <vnm_name> -i <input_file> [-o <outputfile>] [-debug]

The vnm_name argument is the name of the SpectroSERVER host. This argument is required.

The input_file is the name of the comma-delimited input file containing the connectivity information. This argument is required.

The -o argument logs the error information to the file named in the outputfile parameter. If this option is not used, the error information is logged to a file named inputfile.log, where inputfile is the name of the comma-delimited input file.

The debug argument indicates that you would like to create a debugging output file during the import process. This argument is optional.
Using ATM Circuit Manager

This section describes managing ATM Circuits with SPECTRUM. It provides information on setting up thresholds, monitoring network performance, and diagnosing some common network problems.

SPECTRUM management of PVPs and PVCs on ATM switches and ATM clients consists of:

- Performance monitoring and reporting
- Load threshold alarming
- Circuit fault notification
- Service Provider/Customer tracking

Configuring and Monitoring Link Models

This section describes managing link models through the VCL Quality of Service and VCL Threshold views. VCL Link models are displayed in the ATM Logical Connection view or the sub-interface view of an ATM Interface model. VCL Link models allow you to:

- View the quality of service information through the VCL Quality of Service Information view
- Monitor real-time performance of the channel through the Performance view.
- Set thresholds through the VCL Threshold view.

Note: To use the Performance and Threshold views, your device must support the ATM2 MIB or one of the supported proprietary MIB extensions. Without the cell counters inherent in this MIB, no thresholds can be set for a circuit and no performance information will be available.
The Logical Connection view displays all of the managed virtual circuits that go through the switched fabric represented by the ATM_Cloud or ATM_Network model from which it was launched.

**Starting the ATM Logical Connection View**

The ATM Logical Connection view is accessed from the ATM_Cloud or ATM_Network model. Access the ATM Logical Connection view, as follows:

1. Highlight the ATM_Cloud or ATM_Network icon
2. Access the Icon Subviews menu by selecting Icon Subviews from the View menu or by clicking on the ATM_Cloud or ATM_Network icon with the right mouse button
3. Select ATM Logical Connection View from the Icon Subviews menu

**Caution:**

If you are monitoring Cisco devices with the ATM Circuit Manager, and are making use of SPECTRUM’s Live Pipes feature or the PortPollStatus attribute to monitor connectivity, it is recommended that you use Cisco’s Operation, Administration, and Maintenance (OAM) feature. Turning on this feature will ensure that you are able to detect communication problems that occur on a permanent virtual connection (PVC) where the network connectivity is lost, but the PVC remains up on the end devices. In this instance, if OAM is not configured and a CiscoATMvclLnk goes down, no alarm will be generated.

For more information on this issue please refer to the ATM technical tips on Cisco’s web site.
Figure 8: The ATM Logical Connection View
Managed Links

A managed virtual circuit is a circuit whose endpoint models in SPECTRUM are associated by the Links_with relation. By default, none of the circuits through the switched fabric are managed.

This is mostly because there is no way of autodiscovering the circuits that go through a service provider’s network because it is not possible to have SNMP contact with a service provider’s switches. The Logical Connection view provides management of these circuits by allowing a user to establish a Links_with association between any two virtual link models of interfaces adjacent to the ATM_Cloud or ATM_Network model. This the normal method of indicating a virtual circuit to SPECTRUM. SPECTRUM will use this information during the fault isolation process.

After the clients, switches, and containers have been modeled and connected in the SPECTRUM topology, you can launch the Logical Connection view from the ATM_Network or ATM_Cloud icons. The Logical Connection view initially indicates that there are no connected virtual links. To provide management for the clients' virtual circuits, do the following:

1. Click on the "Add" toolbar button to access the Add Link view (Figure 9).

All adjacent interfaces will be displayed. To establish the Links_with association between two virtual links:

2. Select the interface on the left side and the interface on the right side.
3. Select the link on the left side and the link on the right side.
4. Click the Add button.

The Links_with association will be added to the virtual link models and the connection will be displayed in the Logical Connection view.
Figure 9: The Add Link View
Unmanaged Links

It is possible for a SPECTRUM user to be responsible for a circuit, but not have SNMP contact with the device on one side. If this is the case, the Unmanaged Link option provides a mechanism to manage the circuit from a single endpoint.

There are two methods by which a user would create UnmanagedAtmLink models:

1. The first and primary way is though the use of the ATM Logical Connection view. In this view, the user will be able to select a currently-modeled, manageable ATM link and create a connection between it and an unmanaged ATM link.

2. The second method is using the ACMAscciiModelingApp application. This application uses an ascii file containing ATM link data to model a customer's network.

Using the ATM Logical Connection View to Create Unmanaged Link Models

At the bottom of the right hand VPL/VCL list, there is a selection for "Unmanaged Link". To create Unmanaged Link models:

1. Select an ATM Interface and link model on the left side of the Add Link view.
   This will be the single-point of management for this connection.

2. Highlight "Unmanaged Link" on the right side of the ATM Logical Connection view

3. Click on the **Add** button

You will be asked to provide some additional information about this endpoint. The information entered will be displayed in this view, but used only for display purposes. The only required field is model name.
When the "Unmanaged Link" option is used, the Logical Connection view creates a new model of type **UnmanAtmLink** and associates this model with the ATM_Network/ATM_Cloud model via the *Contains* relation. A *Links with* association is then created between the "left side" link model and the new UnmanAtmLink model.

**Controlling the Creation of Link Models**

By default, virtual links are NOT automatically modeled for ATM switches. Management of these circuits is mainly done via SNMP communication with the ATM clients. An exception to this rule is when an interface of an ATM switch is connected to an ATM_Cloud model. When this occurs, the virtual links associated with that interface are modeled. These virtual links are necessary to resolve the link-to-link connectivity across an ATM_Cloud.

**Link Modeling Options View**

The ATMClientApp model’s ATM Link Modeling Options view allows you to control the creation of VCL and VPL models. If you set the *Create Link Models* option in this view to **TRUE** and the *Create VCL Models* option to **FALSE**, this indicates that VPL models should exist for this device. If you set both the *Create Link Models* option and the *Create VCL Models* option to **TRUE**, this indicates that both VCL and VPL models exist created. If both of these options are set to **FALSE**, neither VCL nor VPL models exist for this device.

To implement the selections in this view, you must click the **Save** button and then click the **Reconfigure Now** button. SPECTRUM will create or destroy the appropriate VPL and VCL models for the device depending on the specifications outlined above. The actual creation of virtual link models is done by reading the VPL and VCL tables to determine which links exist on a device.

**ATM and Frame Relay Virtual Links**

In a situation where there is a mixture of ATM and Frame Relay interfaces on either side of the leased service provider’s network, the Logical Connection view can provide management of either type of circuit by
allowing a user to establish a \textit{Links\_with} association between the ATM and Frame Relay virtual link models.

\begin{quote}
\textbf{Note:} To use Frame Relay with the ATM Circuit Manager Application, you must have purchased and installed the SPECTRUM Frame Relay Manager Application. Contact SPECTRUM Support for additional information.
\end{quote}

To provide management for the ATM and Frame Relay virtual circuits, do the following:

1. Click on the "Add" toolbar button to access the Add Link view.
   All adjacent interfaces will be displayed. To establish the \textit{Links\_with} association between the ATM and the Frame Relay virtual links:
2. Select the interface on the left side and the interface on the right side.
3. Select the link on the left side and the link on the right side.
4. Click the \textbf{Add} button.

The \textit{Links\_with} association will be added to the ATM and Frame Relay virtual link models and the connection will be displayed in the Logical Connection view. The Frame Relay side of the link will display a DLCI Port model. See the \textit{Frame Relay Manager User's Guide} for information on the DLCI Port model.

\section*{Remote Ping}

If you have Cisco routers on your ATM network, the ATM Circuit Manager initiates remote pings to determine the status of ATM PVCs through the CiscoPingApp application. Remote pings are initiated from one router to another. An inference handler examines the ATM connections for a particular router, and instructs that router to ping the IP addresses of the routers on the other side of the ATM PVC. If the ping fails, an event with an event code of \texttt{0x02dc0001} is sent to the ATM link model that represents this router's side of the ATM PVC. This results in a red alarm.
being generated on this model with the following probable cause information.

REMOTE PING FAILURE MAY INDICATE A PVC FAILURE

SYMPTOMS:
The SpectroSERVER initiated a remote ping from one router to another over an ATM PVC. Not all ICMP echos were received back by this router.

PROBABLE CAUSES:
The PVC connecting this router to the IP address that was pinged may be down.

RECOMMENDED ACTIONS:
1) Check the Event tab to see what IP address was pinged.
2) Verify that all PVCs on this device are operating normally.

Using the Event Configuration Editor, it is possible to change the severity of this alarm or stop this alarm from being generated. For instructions on using the Event Configuration Editor, refer to the Event Configuration Editor User’s Guide.

Table 1 shows the remote ping application attributes that can be modified.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Default Setting</th>
<th>Attribute ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnableRemotePing</td>
<td>True</td>
<td>c4063f</td>
<td>Enables/Disables remote pinging. If set to “False” pinging is disabled.</td>
</tr>
</tbody>
</table>
EnableRemotePing

When set to “False” disables remote pinging from the router. If the ATM network has redundant paths set up or if the OSPF (Open Shortest Path First) routing protocol is being used, remote pinging may return information that is not completely useful in determining the health of the network and, therefore, will use up unnecessary bandwidth.

PingInterval

NumberOfPingPackets

PingFailuresAllowed

The three attributes above would be used in conjunction to increase the frequency of remote pinging and the speed of any network fault detection.

For example: if “PingInterval” was lowered to 60, “NumberOfPingPackets” remained set at “3”, and “PingFailuresAllowed” was decreased to “0”, ping requests would be initiated more frequently and no failure of any of these requests would be allowed. This would result in the ATM network being
more closely monitored for remote link problems and, if problems were discovered, alarms would be generated more quickly.
To Change the Remote Ping Configuration

To change the configuration of the remote ping application, you can use the Command Line Interface (CLI) tool. You must determine the model handle of the CiscoPingApp of interest and update the model with the new configuration. Do the following:

1. From the command line in SPECTRUM’s \texttt{vnms} directory, enter the following to determine the model handle and press \textbf{Return}:
   
   \begin{verbatim}
   show models | grep CiscoPingApp
   \end{verbatim}

   The system will return information similar to the following:

<table>
<thead>
<tr>
<th>Model Handle</th>
<th>Model Name</th>
<th>Model Type Handle</th>
<th>Model Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x6f002f3</td>
<td>bird</td>
<td>0x2dc000</td>
<td>CiscoPingApp</td>
</tr>
<tr>
<td>0x6f0015b</td>
<td>montana</td>
<td>0x2dc000</td>
<td>CiscoPingApp</td>
</tr>
<tr>
<td>0x6f0001e</td>
<td>spec203</td>
<td>0x2dc000</td>
<td>CiscoPingApp</td>
</tr>
<tr>
<td>0x6f00018</td>
<td>dowland</td>
<td>0x2dc000</td>
<td>CiscoPingApp</td>
</tr>
</tbody>
</table>

2. Using the following format, change the configuration value(s) on the CiscoPingApp model of interest and press \textbf{Return}:

   \begin{verbatim}
   update model <model name> attrid=<attribute ID>, val=<value>
   \end{verbatim}

   \textbf{For example:}

   \begin{verbatim}
   update model <dowland> attrid=c40642, val=0
   \end{verbatim}
ACMAsciimodelingApp

To facilitate the modeling of hundreds of virtual circuits, an ASCII file import application is included with the ATM Circuit Manager. This application takes a text file as a parameter and automatically establishes the Links_with association between the link models indicated in the file. The application assumes that the physical connectivity modeling has already been done, including connecting all appropriate interfaces to an ATM_Cloud (if a service provider's network is used). The format of the text file must be as follows:

link_A.ip,link_A.oid,link_B.ip,link_B.oid,circuit name,circuit id

Circuit name and circuit id are optional parameters. The oid must be the two-term index of a VPL or the three-term index of a VCL. If no model representing link_B is found, the application will create an UnmanAtmLink and associate it with link_A. This circuit will then be managed by a single endpoint - link_A.

If the circuit name and circuit id parameters are specified, this information will be written to the existing models (or to the newly created UnmanAtmLink model).

To use this application, open a shell window, and change directories to SPECTRUM’s SG-Support/CsScript directory. Run the application with the following syntax:

./ACMAsciimodelingApp <SpectroSERVER machine name> <input file> <output file>
If not specified, the application will create an output file name based on the date and time of execution. The output file contains information about which circuits were created successfully, and which failed. Sample output follows:

Found ATM link 4.0.16 at IP 172.19.59.56
Found ATM link 6.0.16 at IP 172.19.56.77
Successfully created ATM circuit!
Found ATM link 0.0.15 at IP 192.168.112.227
Found ATM link 0.0.15 at IP 192.168.112.228
Successfully created ATM circuit!
Found ATM link 0.0.14 at IP 192.168.112.227
Unable to find ATM link 5.0.47 at IP 172.12.47.134, but found existing circuit
Did not replace existing ATM circuit!
ATM Link Threshold View

ATM Link models have a Threshold view that allows the user to establish levels of activity that will generate alarms.

Figure 10: The Threshold View
The following thresholds can be set in this view:

**Received Load**
The average number of bits received by the ATM Link model since the last poll.

**Transmit Load**
The average number of bits transmitted by the ATM Link model since the last poll.

**Receive CPS**
The average number of cells per second received by the ATM Link model since the last poll.

**Transmit CPS**
The average number of cells per second transmitted by the ATM Link model since the last poll.

By default, the high threshold values for load are set to “90%” and reset at “80%” and the low threshold values are set and reset at “0” (disabled). The threshold values are recalculated at every poll cycle and represent the average number per poll.

The “Set” field for each attribute is the high threshold that, if exceeded, will generate an alarm for that attribute. The “Reset” field for each attribute is the low threshold that, if gone below, automatically clears the alarm for that attribute.

**Caution:**
Do not set the “Reset” field to “0.” If the “Reset” field is set to “0” and the “Set” field is crossed, the subsequent alarm will never be cleared automatically. If this alarm is cleared manually by the user, the SPECTRUM threshold intelligence is not reset and the alarm for that model will not be generated again. To reset the threshold intelligence, the SpectroSERVER must be restarted.

To access the Threshold view, highlight the ATM Link model in the ATM Logical Connection view and select **Threshold** from the Icon Subviews menu.
Setting Up Thresholds

To set thresholds for an ATM Link model, do the following:

1. Access the ATM Logical Connection view.
2. Select **Threshold** from the left-side Model (A) Icon Subviews menu.
3. In the “Set” field for the desired attribute, enter the high threshold.
4. In the “Reset” field for the desired attribute, enter the low threshold.
5. Select **Save All Changes** from the File menu.

Figure 11 describes the functionality provided by the high and low threshold feature for ATM Link models.
**ATM VCL QoS Information View**

The VCL QoS (Quality of Service) Information view can also be accessed from an ATM link model. It provides receive and transmit parameters for this connection, the QoS class being used for the connection, and the bandwidth parameters used to calculate load.

1. Access the ATM Logical Connection view
2. Highlight an ATM Link model
3. Select **QoS Parameters** from the Icon Subviews menu
Figure 12: The VCL QoS Information View
QoS Classes

**Constant Bit Rate (CBR)** is a connection which supports applications that transmit at a fixed bandwidth. The amount of bandwidth is described by the Peak Cell Rate parameter. Constant Bit Rate works well for circuit emulation.

**Variable Bit Rate (VBR)** supports connections that require a variable bandwidth. The Peak Cell Rate, Sustained Cell Rate, and Maximum Burst Size parameters describe this type of QoS connection. Variable Bit Rate works well for voice and video compression transmissions.

**Unspecified Bit Rate (UBR)** is a connection which supports an open-ended bit rate and provides a “best effort” quality of service. Unspecified Bit Rate works well in the transmission of LAN data. It will use whatever bandwidth is available. If network congestion occurs, the data is placed in a buffer. If the buffer gets too full, the data is discarded.

**Available Bit Rate (ABR)** is an Unspecified Bit Rate connection with the addition of flow control protocols which attempt to prevent network congestion from occurring. It is still a “best effort” quality of service and is used for the transmission of LAN data.

Receive QoS Parameters

**QoS Class**
The QoS class used for this connection.

**QoS Peak Cell Rate**
The maximum number of cells per second the connection can receive from the network.

**QoS Sust Cell Rate**
The average number of cells per second the connection can receive from the network.

**QoS Max Burst Size**
The maximum length of time that the connection can receive cells from the network at the peak cell rate.
**QoS Tagging**
If “on”, tagging has been enabled. Tagging is the process of marking the CLP (Cell Loss Priority) bit of cells in an ATM network because they do not conform to the subscribed QoS contract. This identifies these cells as having a lower priority and would be the first cells to be dropped by the network in traffic congestion situations.

**QoS CLPO Peak Cell Rate**
The maximum number of cells per second with the CLP bit set that the connection can receive from the network.

**QoS CLPO Sust Cell Rate**
The average number of cells per second with the CLP bit set that the connection can receive from the network.

**QoS CLPO Max Burst Size**
The maximum length of time that the connection can receive cells with the CLP bit set from the network at the CLP peak cell rate.

**QoS Cell Delay VT**
The maximum time delay variation tolerance between the arrival one cell from the network and the arrival of the cell immediately following it. This is typically very low for CBR and VBR connections and very high for ABR and UBR connections.

**Transmit QoS Parameters**

**QoS Class**
The QoS class used for this connection.

**QoS Peak Cell Rate**
The maximum number of cells per second the connection can receive from the network.

**QoS Sust Cell Rate**
The average number of cells per second the connection can transmit on to the network.
**QoS Max Burst Size**
The maximum length of time that the connection can transmit cells on to the network at the peak cell rate.

**QoS Tagging**
If “on”, tagging has been enabled. Tagging is the process of marking the CLP (Cell Loss Priority) bit of cells in an ATM network because they do not conform to the subscribed QoS contract. This identifies these cells as having a lower priority and would be the first cells to be dropped by the network in traffic congestion situations.

**QoS CLPO Peak Cell Rate**
The maximum number of cells per second with the CLP bit set that the connection can transmit on to the network.

**QoS CLPO Sust Cell Rate**
The average number of cells per second with the CLP bit set that the connection can transmit on to the network.

**QoS CLPO Max Burst Size**
The maximum length of time that the connection can transmit cells with the CLP bit set on to the network at the CLP peak cell rate.

**QoS Cell Delay VT**
The maximum time delay variation tolerance between the transmission one cell on to the network and the transmission of the cell immediately following it. This is typically very low for CBR and VBR connections and very high for ABR and UBR connections.

**Bandwidth Parameters**

**Receive Bandwidth**
The maximum number of bits per second that can be received by this connection.

If the **QoS Class** is set to Variable Bit Rate, the **Receive Bandwidth** is calculated using the **Receive QoS Sustained Cell Rate** parameter. If the
**QoS Class** has any other value, the **Receive Bandwidth** is calculated using the **Receive QoS Peak Cell Rate** parameter.

Receive Bandwidth (bits per second) = (Sustained Cell Rate (cells per second) * 53 (bytes per cell)) * 8 (bits per byte)

or

Receive Bandwidth (bits per second) = (Peak Cell Rate (cells per second) * 53 (bytes per cell)) * 8 (bits per byte)

**Transmit Bandwidth**
The maximum number of bits per second that can be transmitted by this connection.

If the **QoS Class** is set to **Variable Bit Rate**, the **Transmit Bandwidth** is calculated using the **Transmit QoS Sustained Cell Rate** parameter. If the **QoS Class** has any other value, the **Transmit Bandwidth** is calculated using the **Transmit QoS Peak Cell Rate** parameter.

Transmit Bandwidth (bits per second) = (Sustained Cell Rate (cells per second) * 53 (bytes per cell)) * 8 (bits per byte)

or

Transmit Bandwidth (bits per second) = (Peak Cell Rate (cells per second) * 53 (bytes per cell)) * 8 (bits per byte)

**Use PCR in calculation**
This option can be toggled to **TRUE** or **FALSE**. False is the default value, and allows the Receive and Transmit Bandwidth parameters to be calculated as described above.

If this option is set to **TRUE** and the values for Peak Cell Rate are specified, SPECTRUM will always use Peak Cell Rate to calculate the Receive and Transmit Bandwidth parameters. If this option is set to **TRUE** and the values for Peak Cell Rate and Sustained Cell Rate are not specified, the **If Speed** is used as the bandwidth. Click on the **Option Explanation** button for more information about the **Use PCR in calculation** option.
ATM VCL Performance View

Two statistical fields indicating load are displayed in the ATM Link Performance view. They are In Load (the value of the \textit{rcvLoad} attribute) and Out Load (the value of the \textit{xmtLoad} attribute).

\textbf{Figure 13: The ATM VCL Performance View}

![ATM VCL Performance View](image)

The formulae for these attributes are as follows:

\[ \text{rcvLoad} = \frac{\text{rcvCellsPerSecond} \times 100}{\text{rcvBandwidth}} \]
\[ \text{xmtLoad} = \frac{\text{xmtCellsPerSecond} \times 100}{\text{xmtBandwidth}} \]

The \textit{rcvBandwidth} and \textit{xmtBandwidth} attributes are defined by either the Peak Cell Rate (PCR) or Sustainable Cell Rate (SCR) depending on the Quality of Service (QoS) type. For Variable Bit Rate (VBR) circuits, the bandwidth is defined as the SCR. For all other types of service, the
bandwidth is defined as PCR. This means, that for VBR circuits, the load can exceed 100%.

Users who want to always use PCR as bandwidth (even for VBR circuits) can toggle the **Use PCR in calculation** button available in the QoS Information view to **TRUE** (see *Bandwidth Parameters* on Page 50).

The `rcvCellsPerSecond` attribute is calculated by reading the attribute pointed to by `rcvCells_Attr` and the `upTime` attribute over a particular interval, and subtracting the first values from the second values. This gives us a delta of cells received and a delta of micro seconds elapsed. By dividing the delta cells by delta micro seconds, and then multiplying by one hundred, we arrive at the `rcvCellsPerSecond` value.

The `rcvLoad` and `xmtLoad` are graphed in the ATM link model’s Performance view, and can be logged for historical reports.
ATM VCL Service Information View

The ATM VCL Service Information view provides information about service providers associated with the ATM network. All service information is entered by the user and is provided for reference purposes only.

1. Access the ATM Logical Connection view
2. Highlight an ATM Link model
3. Select **Service Information** from the Icon Subviews menu

*Note:* The ATM VCL Service Information view is not supported by the UnmanAtmLink model.
The following information can be entered:

**Provider**
The name of the service provider associated with the ATM network. ATM customers who use multiple carriers can use this field to indicate which carriers provide service for which circuits.
**Customer**
Users who are service providers and manage other companies ATM networks can use this field to indicate who the service provider’s customer is.

**Primary Contact**
The name, phone number, and/or E-mail address of the person to contact if there is a problem with this circuit.

**Secondary Contact**
The name, phone number, and/or E-mail address of a secondary person to contact if there is a problem with this circuit.

**Service Notes**
This area can be used to enter miscellaneous information about this circuit such as Circuit ID or monthly cost.

To access the ATM VCL Service Information view highlight the ATM VCL model and select **Information** from the Icon Subviews menu.

**Application Models and Views**
The device models representing your ATM infrastructure will have various application models associated with them. These application models will vary based on the MIBs that the device supports. For information on these application models and the views that they display, reference the management module guide that pertains to the model representing your device.
Theory of Operations

This section describes the theory and software mechanisms behind the ATM Circuit Manager application.

Modeling Paradigm

In SPECTRUM, physical links between devices are managed by polling the status and performance data of the endpoints, for example, FDDI or HSSI interfaces. There is no difference in managing virtual links than there is in managing physical links except that the virtual link endpoints are Virtual Path Links (VPLs) or Virtual Channel Links (VCLs). These endpoints are modeled for all ATM clients that have supported ATM MIBs. These endpoints are NOT modeled for ATM switches unless an interface of an ATM switch is connected to an ATM_Cloud model. In this case, models are created for the VPLs and VCLs on that interface only.

Like physical interfaces, VPLs and VCLs have objects in a MIB that contain the status, bandwidth, and (depending on the MIB) performance statistics for these links. Whereas physical interfaces (as represented in the MIB-II ifTable) have a single-term index, VPLs have a two-term index and VCLs have a three-term index.

- A VPL index is in the form ifIndex.VPI, where ifIndex is the index of the physical interface the VPL runs on, and VPI is the Virtual Path Identifier - which is merely the name that the device calls the path.

- A VCL index is in the form ifIndex.VPI.VCI, where VCI is the Virtual Channel Identifier - which is merely the name that the device calls the channel. The VPI of a VCL may be “0”, which may indicate that there is no Virtual Path. We will refer to VPLs and VCLs collectively as virtual links.
In SpectroGRAPH, the virtual link models are associated with the lower-layer interface model via the \textit{HASPART} relation. As such, they will not be visible in the Device Topology view of the device, but will be displayed in the Sub-Interface view of the lower layer interface model. The lower layer interface may be a physical ATM Interface model or, in the case of a VCL, may be a VPL model.

Virtual link models may have \texttt{Links\_with} associations with other virtual link models indicating a logical connection. The \texttt{Links\_with} association between virtual link models is never added by the discovery process. These associations are added manually by using the ATM Logical Connection view application or semi-automatically by providing an ASCII file to the ACMAsciiModelingApp application. Both of these applications have been discussed in detail in previous sections.

Figure 14 shows what an actual SPECTRUM modeling of the ATM environment may look like after some \texttt{Links\_with} associations are made.
In an actual ATM network environment, there would be two switches collected by the ATM_Network, and the port of one switch would be connected to the port of the other via a SwitchLink model. Figure 14 omitted those details to save space and simplify the image.
Fault Suppression

Because ATM link models are derived from the Port model type, their Internal_Link_Status attribute is read every polling interval. This attribute is calculated based on the value of the administrative and operational status of the link. For devices supporting the standard ATM MIB, these objects are atmVclAdminStatus and atmVclOperStatus for channel links, and atmVplAdminStatus and atmVplOperStatus for path links. If the value of Internal_Link_Status is not active, it is assumed that an error condition has occurred. An attempt is made to isolate the problem to either this link model or its “parent” model. This is done by reading the Internal_Link_Status of the parent model (the parent model could be a VPL or an interface).

If the parent model is active, a RED alarm is asserted on this link model. If the parent is NOT active, a GRAY alarm is asserted on this link model. Since the parent model’s Internal_Link_Status attribute has just been read, the model’s intelligence will assert either RED or GRAY if it is active.

Threshold Alarm Generation

High and low thresholds are provided for rcvLoad and xmtLoad. The rcvLoad and xmtLoad values are calculated using SpectroWATCH. These SpectroWATCH expressions are only evaluated for link models that have Polling_Status set to TRUE. Polling_Status is set to FALSE by default to limit management traffic over ATM links.

The rcvLoad and xmtLoad are read every polling interval. If the value of rcvLoad exceeds the value of the hiRcvLoadAlarmSet attribute (or is less than the value of the lowRcvLoadAlarmSet attribute) an appropriate alarm will be asserted. If the value later drops below the hiRcvLoadAlarmReset value (or rises above the lowRcvLoadAlarmReset value), the alarm will be disasserted. An event, indicating that the threshold was violated, will persist.

The threshold alarms for xmtLoad are asserted and disasserted in the same manner as rcvLoad.
Symbols

“best effort” 48

A

ABR 48, 49, 50
ACMAsciiModelingApp 17, 34, 41, 58
Add Link view 32
alarms 43, 60
ATM and Frame Relay 24
ATM cell 6
ATM interface 19
ATM interface model 11, 58
ATM interface model 14
ATM Link 43, 45, 46, 54
ATM link 60
ATM Logical Connection view 34
ATM MIBs 57
ATM_Cloud 8, 9, 35, 57
ATM_Cloud model 8, 19, 21, 30, 41
ATM_Network 59
ATM_Network model 7, 10, 17, 21, 30
ATM2 MIB 9
atmVclAdminStatus 60
ATMVclLink 11
atmVclOperStatus 60
atmVplAdminStatus 60
atmVplOperStatus 60
Available Bit Rate 48

B

bandwidth 46, 48, 52

C

CBR 48, 49, 50
Cell Loss Priority 50
Channel 15
Cisco CA-CONN-MIB 9
Cisco routers 36
CiscoPingApp 36, 40
CLP 50
CLP bit 49, 50
Constant Bit Rate 48
Customer 56

D

Device Topology view 19
DLCI Port model 36

E

EnableRemotePing 37

F

fixed bandwidth 48
Fore Switch MIB 9
Frame Relay 23
Frame Relay interface 35
Frame Relay virtual link models 36
modeling procedures 17
modeling schemes 7, 17

H
high threshold 45
high threshold values 44
high thresholds 45
hiRcvLoadAlarmReset 60
hiRcvLoadAlarmSet 60
hybrid ATM network 23

N
NumberOfPingPackets 38

O
Open Shortest Path First 38
OSPF 38

P
Path 15
PCR 52
Peak Cell Rate 48, 52
Performance view 53
Permanent Virtual Channels 11
Permanent Virtual Circuits 6, 15
Permanent Virtual Circuits (PVCs) 6
Permanent Virtual Paths 11, 15
PingFailuresAllowed 38
PingInterval 38
PingPacketSize 38
Polling_Status 60
Provider 55
PVC 36
PVCs 11, 15
PVP 15
PVPs 11

Q
QoS 52
QoS Cell Delay VT 49, 50

ATM Circuit Manager
QoS Class 48, 49
QoS Classes 48
QoS CLPO Max Burst Size 49, 50
QoS CLPO Peak Cell Rate 50
QoS CLPO Sust Cell Rate 50
QoS connection 48
QoS Max Burst Size 50
QoS Peak Cell Rate 48, 49
QoS Sust Cell Rate 49
QoS Tagging 50
Quality of Service 52

T
Threshold 45
threshold alarms 60
Threshold view 43
Transmit CPS 44
Transmit Load 44
Transmit QoS 49

U
UBR 48, 50
Unmanaged Link model 34
Unmanaged Link option 34
UnmanagedAtmLink model 34
UnmanAtmLink 41, 54
UnmanAtmLink model 12, 35
Unspecified Bit Rate 48
upTime 53

V
Variable Bit Rate 48, 52
VBR 48, 49, 50, 52
VCI 6, 15
VCL 8
VCL index 57
VCL Link model 29
VCL QoS (Quality of Service) Information view 46
VCL Quality of Service 29
VCL Service Information view 54
VCL Threshold 29
VCLs 57
Virtual Channel Identifier 6, 15, 57
Virtual Channel Link 8
virtual channel link 14
Virtual Channel Links 57

S
SCR 52
set thresholds 45
SNMP MIBs 8
Sustainable Cell Rate 52
Sustained Cell Rate 48
SVC 6
SW_Link model 17
Switched Virtual Circuit 6

rcvBandwidth 52
rcvCells_Attr 53
rcvCellsPerSecond 53
rcvLoad 53
Receive Bandwidth 50
Receive CPS 44
Receive QoS 48
Received Load 44
red alarm 36
remote pings 36
RFC1695 8
virtual circuits 32
virtual link models 32
virtual links 57
Virtual Path Identifier 6, 15, 57
Virtual Path Link 8
virtual path link 14
Virtual Path Links 57
VPI 6, 15
VPL 8, 60
VPL index 57
VPL model 58
VPLs 57

W

WA_Link model 12
WA_Segment model 12

X

xmtBandwidth 52
xmtLoad 53