CHAPTER 4

Maintenance and Troubleshooting Overview

This chapter contains an overview of maintenance and troubleshooting concepts for the elements of the Cisco Media Gateway Controller (MGC) node. It includes overall maintenance and system troubleshooting strategies, and reviews available troubleshooting tools.

Although maintenance and troubleshooting are described separately in this chapter, they are associated activities. Hence, several of the maintenance and troubleshooting chapters in this guide frequently refer to each other.

This chapter includes the following sections:

- Maintenance Strategy Overview, page 4-1
- Troubleshooting Strategy Overview, page 4-2

Maintenance Strategy Overview

Maintenance usually consists of the following tasks for each element of the Cisco MGC node, performed in the order listed:

- Checking equipment status. Determining the current status involves three basic activities:
  - Reading LEDs—Most Cisco products include light-emitting diode (LED) indicators on the front or rear panels and, in some cases, on both panels. These LEDs indicate the status of the equipment. The specific meaning of each LED on each product is described in the maintenance sections for the individual elements of the Cisco MGC node.
  - Issuing Status Queries—You can query the status of the system using various commands. The commands that can be used to determine the status of the devices in your system are described in the maintenance sections for the individual elements of the Cisco MGC node.
  - Using a GUI NMS—Using a network management system (NMS) with a graphical user interface (GUI), such as CiscoWorks2000 or Cisco WAN Manager, to determine the operational status of system devices is described in detail in the maintenance sections for the individual elements of the Cisco MGC node.
- Removing the device from the system—Procedures for removing defective devices from the system with as little impact on the system as possible are described in the maintenance sections for the individual elements of the Cisco MGC node.
- Replacing the complete device—Reinstating a device into the system using a new or repaired model, again with as little impact on the system as possible, is described in the maintenance sections for the individual elements of the Cisco MGC node.

- Replacing hardware components—Swapping out components of a device is a maintenance task used for replacing defective components and for upgrading hardware. The maintenance chapters for each element of the Cisco MGC node include sections describing how to replace the field-replaceable components of that device.

Troubleshooting Strategy Overview

The Cisco MGC node supports connections to external switches and to internal components, such as media gateway controllers, signal processors, and trunking gateways. Because the Cisco MGC node functions in a complex environment involving numerous connections, links, and signaling protocols, when connectivity and performance problems occur, they can be difficult to resolve.

Troubleshooting usually consists of determining the nature of a problem and then isolating the problem to a particular device or component. When a problem has been isolated and identified, troubleshooting also consists of fixing the problem, usually by replacing the device or some component of the device. The goal of this is to provide you with a general troubleshooting strategy, as well as information about the tools available for isolating and resolving connectivity and performance problems.

Symptoms, Problems, and Solutions

Problems in a system are characterized by certain symptoms. These symptoms can be general (such as a Cisco SLT being unable to access the SS7 network) or specific (routes not appearing in a routing table).

You can determine the cause of a symptom by using specific troubleshooting tools and techniques. After identifying the cause, you can correct the problem by implementing a solution consisting of a series of actions.

General Problem-Solving Model

A systematic approach works best for troubleshooting. Define the specific symptoms, identify all potential problems that could be causing the symptoms, then systematically eliminate each potential problem (from the most likely to the least likely) until the symptoms are no longer present.

Figure 4-1 illustrates the process flow for this general approach to problem-solving. This process is not a rigid outline for troubleshooting. It is a guide you can use to troubleshoot a problem successfully.

The following steps describe the problem-solving process outlined in Figure 4-1 in more detail:

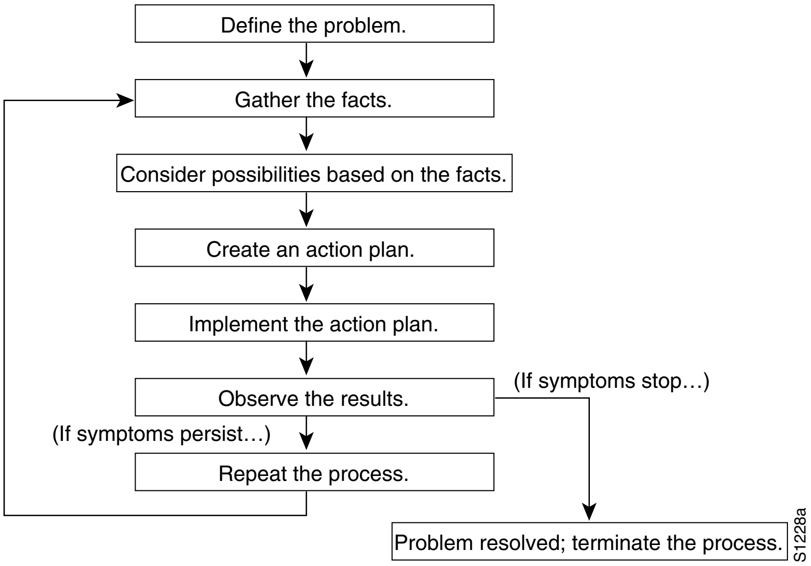
Step 1 When analyzing a problem, draft a clear problem statement. Define the problem in terms of a set of symptoms and the potential causes behind those symptoms.

For example, the symptom might be that the EQPT FAIL alarm has become active. Possible causes might be physical problems, a bad interface card, or the failure of some supporting entity (for example, layer 1 framing).

Step 2 Gather the facts you need to help isolate the symptoms and their possible causes.

Ask questions of affected users, network administrators, managers, and other key people. Collect information from sources such as network management systems, protocol analyzer traces, output from router diagnostic commands, or software release notes.

Figure 4-1 General Problem-Solving Model



Step 3 Consider possible causes based on the facts you have gathered. You can also use these facts to eliminate potential causes from your list.

For example, depending on the data, you might be able to eliminate hardware as a cause, allowing you to focus on software. At every opportunity, try to narrow the number of potential causes so that you can create an efficient plan of action.

Step 4 Create an action plan based on the remaining potential causes. Begin with the most likely cause, and devise a plan in which only one variable at a time is manipulated.

This approach allows you to reproduce the solution to a specific problem. If you alter more than one variable simultaneously, identifying the change that eliminated the symptom becomes more difficult.

Step 5 Perform each step of the action plan carefully, and test to see if the symptom disappears.

Step 6 Whenever you change a variable, gather the results. You should use the same method of gathering facts that you used in Step 2.

Analyze the results to determine if the problem has been resolved. If it has, then the process is complete.

Step 7 If the problem has not been resolved, you must create an action plan based on the next most likely problem in your list. Return to Step 2 and continue the process until the problem is solved.

Before trying out a new cure, make sure to undo any "fixes" you made in implementing your previous action plan. Remember that you want to change only one variable at a time.

Note If you exhaust all of the common causes and actions (those outlined in this chapter and those that you have identified for your environment), your last recourse is to contact the Cisco Technical Assistance Center (TAC). Refer to the "Obtaining Technical Assistance" section on page xviii for more information about contacting the Cisco TAC.

System Troubleshooting Tools

This section presents information about the wide variety of tools you can use to troubleshoot the system.

Alarms

The Cisco MGC software generates alarms to indicate problems with processes, routes, linksets, signaling links, and bearer channels. For more information on troubleshooting using alarms, refer to Chapter 8, "Troubleshooting the Cisco MGC Node." Refer to the *Cisco Media Gateway Controller Software Release 7 Software Messages Reference Guide* for detailed information on the system alarms.

Call Traces

The Cisco MGC generates call traces that capture call-processing activity by following the call from a specified destination through the Cisco MGC software engine to see where it fails. Call failure location is determined using the following information provided in the call trace:

- The protocol data units (PDUs) that the Cisco MGC receives
- How the Cisco MGC decodes the PDU
- The PDUs that the Cisco MGC sends out

The results of call traces are signal flow diagrams that you can use for troubleshooting. Call traces are typically used to capture system activity as part of a procedure to clear an alarm. For more information on using call traces, refer to Chapter 8, "Troubleshooting the Cisco MGC Node."

System Logs

The Cisco MGC software continuously generates log files of various system information, including operational measurements (OMs) and alarm records. You can use these logs to obtain statistical information about the calls processed by the system and network events such as delays or service-affecting conditions. The Cisco MGC generates the following types of logs:

- Platform logs containing information useful for tracking configuration errors and signaling link and call instantiation problems.
- Command/response logs containing Man-machine language (MML) command history.
- Alarm logs containing alarm information.
- Measurement logs containing system measurements data.
- Call record logs containing call-processing data.

System logs can be read using the various viewers within the Cisco MGC viewer toolkit. For more information on the viewers that comprise the Cisco MGC toolkit, refer to "Using the Cisco MGC Viewer Toolkit" section on page 3-102.

Refer to Appendix A, "Configuring Cisco MGC Report Files,"for more information on system log files.

MML Queries

MML is the command line interface method for configuring and managing the Cisco MGC. You can use it to retrieve information about system components, and to perform logging and tracing. Refer to the *Cisco Media Gateway Controller Software Release 7 Software MML Command Reference Guide* for more information.

Cisco Internetwork Management Tools

The following Cisco internetwork management products provide design, monitoring, and troubleshooting tools to help you manage your Cisco MGC node:

- CiscoWorks2000
- Cisco WAN Manager
- Cisco Media Gateway Controller Node Manager (CMNM)

CiscoWorks2000

CiscoWorks2000 is a series of SNMP-based internetwork management software applications. CiscoWorks applications are integrated on several popular network management platforms. The applications build on industry-standard platforms to provide tools for monitoring device status, maintaining configurations, and troubleshooting problems.

Some of the applications included in CiscoWorks2000 that are useful for troubleshooting are:

- Device Monitor—Monitors specific devices for environmental and interface information.
- Health Monitor—Displays information about the status of a device, including buffers, CPU load, memory available, and protocols and interfaces being used.
- Show Commands—Enables you to view data similar to output from router show EXEC commands.
- Path Tool—Collects path utilization and error data by displaying and analyzing the path between devices.
- Device Polling—Extracts data about the condition of network devices.
- CiscoView—Provides dynamic monitoring and troubleshooting functions, including a graphical display of Cisco devices, statistics, and comprehensive configuration information.
- Offline Network Analysis—Collects historical network data for offline analysis of performance trends and traffic patterns.
- CiscoConnect—Allows you to provide Cisco with debugging information, configurations, and topology information to speed resolution of network problems.

CiscoWorks2000 can be used to manage a variety of Cisco products. Within the Cisco MGC node, CiscoWorks2000 can be used for management of the Cisco SLTs and the Cisco Catalyst 5500 MSRs. Refer to the CiscoWorks2000 documentation for more information.

Cisco WAN Manager

Cisco WAN Manager is part of the Cisco Service Management System of provisioning and management tools for service provider and large enterprise networks. Working at the network and element management level, WAN Manager provides fault-management capabilities handled through the Event Browser, CiscoView, and Configuration Save and Restore features.

You can use Cisco WAN Manager to perform search, sort, and filter operations and to tie events to extensible actions. For instance, Cisco WAN Manager can page someone upon receiving a certain type of SNMP trap. It supports alarm hierarchies that report the root cause of problems to operators and higher-level systems.

Configuration Save and Restore saves a snapshot of the entire network configuration. For disaster recovery, operators can selectively restore configurations of any element, from a single node up to the entire network. This restoration ability significantly reduces recovery time when a catastrophic failure occurs.

The Cisco WAN Manager Trivial File Transfer Protocol (TFTP) statistics collection facility offers the ability to obtain extensive usage and error data across machines and platforms.

A wide range of statistics are available at the port and virtual channel level including:

- Connection statistics
- Circuit line statistics
- Packet line statistics
- Frame Relay port statistics
- Network statistics
- Physical layer statistics
- Protocol layer statistics

The Cisco WAN Manager application can be used to manage a variety of Cisco products. Within the Cisco MGC node, the Cisco WAN Manager can be used for management of the Cisco SLTs and the Cisco Catalyst 5500 MSRs. Refer to the Cisco WAN Manager documentation for more information.

Cisco Media Gateway Controller Node Manager

The Cisco Media Gateway Controller Node Manager (CMNM) is an element management system based on the Cisco Element Management Framework (CEMF). It is responsible for managing the Cisco MGC node, including Cisco MGC(s), LAN switch(es), and Cisco SLTs.

NMS design divides network management into five discrete areas: fault, configuration, accounting, performance, and security. The CMNM provides fault and performance management of the Cisco MGC, as well as flow-through provisioning of the Cisco MGC and its subcomponents. In addition, CMNM also provides fault and performance management of the Cisco SLT and LAN switch. CMNM uses the Cisco Voice Service Provisioning Tool to provide configuration of the Cisco MGC and uses CiscoView for configuration of the Cisco SLT and the LAN switch.

Security and some accounting features are provided directly by the CEMF platform. CMNM does not provide any security or accounting features beyond what is natively supported by the CEMF. CMNM is designed to be used on a standalone basis with a customer operations support system or a Cisco-based NMS such as the Voice Network Manager (VNM).

For more information on CMNM, refer to the *Cisco MGC Node Manager User's Guide*.

Cisco SLT Diagnostic Commands

Cisco SLTs provide the following integrated IOS command types to assist you in monitoring and troubleshooting systems:

- **show**
- **debug**
- **ping**
- **trace**

Show Commands

The show commands are powerful monitoring and troubleshooting tools. You can use the show commands to perform a variety of functions:

- Monitoring router behavior during initial installation
- Monitoring normal network operation
- Isolating problem interfaces, nodes, media, or applications
- Determining when a network is congested
- Determining the status of servers, clients, or other neighbors

Some of the most commonly used status commands include:

- **show interfaces**—Displays statistics for network interfaces using the following commands:
  - **show interfaces ethernet**
  - **show interfaces fddi**
  - **show interfaces atm**
  - **show interfaces serial**
- **show controller t1**—Displays statistics for T1 interface card controllers
- **show running-config**—Displays the router configuration currently running
- **show startup-config**—Displays the router configuration stored in nonvolatile RAM (NVRAM)
- **show flash**—Displays the layout and contents of Flash memory
- **show buffers**—Displays statistics for the buffer pools on the router
- **show memory**—Shows statistics about the router's memory, including free pool statistics
- **show processes**—Displays information about the active processes on the router
- **show stacks**—Displays information about the stack utilization of processes and interrupt routines, as well as the reason for the last system reboot
- **show version**—Displays the configuration of the system hardware, the software version, the names and sources of configuration files, and the boot images

For details on using and interpreting the output of specific **show** commands, refer to the Cisco IOS command reference for the release you are using.

Using Debug Commands

The **debug** privileged EXEC commands can provide a wealth of information about the traffic being seen (or not seen) on an interface, error messages generated by nodes on the network, protocol-specific diagnostic packets, and other useful troubleshooting data.

Caution Exercise care when using **debug** commands. These commands are processor-intensive and can cause serious network problems (degraded performance or loss of connectivity) if they are enabled on an already heavily loaded router. When you finish using a **debug** command, remember to disable it with its specific **no debug** command, or use the **no debug all** command to turn off all debugging.

Note Output formats vary among **debug** commands. Some generate a single line of output per packet, and others generate multiple lines of output per packet. Some generate large amounts of output, and others generate only occasional output. Some generate lines of text, and others generate information in field format.

To minimize the negative impact of using **debug** commands, follow this procedure:

Step 1 Enter the **no logging console** global configuration command on your router. This command disables all logging to the console terminal.

Step 2 Telnet to a router port and enter the **enable** EXEC command.

Step 3 Enter the **terminal monitor** command on your router to copy **debug** command output and system error messages to your current terminal display.

This permits you to view **debug** command output remotely, without being connected through the console port. Following this procedure minimizes the load created by using **debug** commands because the console port no longer has to generate character-by-character processor interrupts.

If you intend to keep the output of the **debug** command, spool the output to a file. The procedure for setting up such a **debug** output file, as well as complete details regarding the function and output of **debug** commands is provided in Chapter 10, “Debug Command Reference,” in the *Troubleshooting Internetworking Systems* manual.

Note In many situations, third-party diagnostic tools can be more useful and less intrusive than the use of **debug** commands. For more information, see the “Third-Party Troubleshooting Tools” section on page 4-9.

Using the Ping Command

To check host accessibility and network connectivity, use the **ping** EXEC (user) or privileged EXEC command.

For IP, the **ping** command sends ICMP Echo messages. If a station receives an ICMP Echo message, it sends an ICMP Echo Reply message back to the source. The extended command mode of the **ping** command permits you to specify the supported IP header options. This allows the router to perform a more extensive range of test options.

It is a good idea to use the **ping** command when the network is functioning properly under normal conditions so that you have something to compare against when you are troubleshooting.

For detailed information on using the **ping** and extended **ping** commands, refer to the *Cisco IOS Configuration Fundamentals Command Reference*.

Using the Trace Command

The **trace** user EXEC command discovers the routes a router's packets follow when traveling to their destinations. The **trace** privileged EXEC command permits the supported IP header options to be specified, allowing the router to perform a more extensive range of test options. The **trace** command uses the error message generated by routers when a datagram exceeds its time-to-live (TTL) value.

First, probe datagrams are sent with a TTL value of 1. This causes the first router to discard the probe datagrams and send back "time exceeded" error messages. The **trace** command then sends several probes and displays the round-trip time for each. After every third probe, the TTL is increased by 1.

Each outgoing packet can result in one of two error messages. A "time exceeded" error message indicates that an intermediate router has seen and discarded the probe. A "port unreachable" error message indicates that the destination node has received the probe and discarded it, because it could not deliver the packet to an application. If the timer goes off before a response comes in, the **trace** command prints an asterisk (*).

The **trace** command terminates when the destination responds, when the maximum TTL is exceeded, or when the user interrupts the **trace** with the escape sequence. It is a good idea to use the **trace** command when the network is functioning properly under normal conditions so that you have something to compare against when troubleshooting.

For detailed information on using the **trace** and extended **trace** commands, refer to the *Cisco IOS Configuration Fundamentals Command Reference*.

Third-Party Troubleshooting Tools

In many situations, third-party diagnostic tools can be more useful than system commands that are integrated into the router. For example, the enabling of a processor-intensive **debug** command can contribute to the overloading of an environment that is already experiencing excessively high traffic levels. Attaching a network analyzer to the suspect network is less intrusive and is more likely to yield useful information without interrupting the operation of the router.

Some useful third-party tools for troubleshooting internetworks include:

- Volt-ohm meters, digital multimeters, and cable testers
- Breakout boxes, fox boxes, bit error rate testers (BERTs), and block error rate testers (BLERTs)
- Network analyzers and network monitors
- Time domain reflectometers (TDRs) and optical time domain reflectometers (ODTRs)

Volt-Ohm Meters, Digital Multimeters, and Cable Testers

Volt-ohm meters and digital multimeters are at the lower end of the spectrum of cable testing tools. These devices can measure basic parameters such as AC and DC voltage, current, resistance, capacitance, and cable continuity. They are used primarily to check physical connectivity.

Cable testers (scanners) can also be used to check physical connectivity. Cable testers are available for shielded twisted-pair, unshielded twisted-pair, 10BASE-T, and coaxial and twinax cables.

A given cable tester might be also able to perform any of the following functions:

- Test and report on cable conditions, including near-end crosstalk, attenuation, and noise
- Perform TDR, traffic monitoring, and wire map functions
- Display media access control (MAC) layer information about LAN traffic, provide statistics such as network utilization and packet error rates, and perform limited protocol testing (for example, TCP/IP tests such as ping)

Similar testing equipment is available for fiber-optic cable. Due to the relatively high cost of fiber-optic cable and its installation, the cable should be tested both before installation (on-the-reel testing) and after installation. Continuity testing of fiber-optic cable requires either a visible light source or a reflectometer. Light sources capable of providing light at the three predominant wavelengths, 850 nanometers (nm), 1300 nm, and 1550 nm, are used with power meters that can measure the same wavelengths and test attenuation and return loss in the fiber-optic cable.

Breakout Boxes, Fox Boxes, and BERTs/BLERTs

Breakout boxes, fox boxes, and BERTs/BLERTs are digital interface testing tools used to measure the digital signals present at the interfaces of PCs, CSU/DSUs, and other devices. These testing tools can monitor data line conditions, analyze and trap data, and diagnose problems common to communications systems. Traffic from data terminal equipment (DTE) through data communications equipment (DCE) can be examined so that you can isolate problems, identify bit patterns, and ensure that the proper cabling has been installed. These devices cannot test media signals such as those for Ethernet, Token Ring, or FDDI.

Network Monitors and Analyzers

You can use network monitors to continuously track packets crossing a network, thus obtaining an accurate picture of network activity at any moment, or a historical record of network activity over a period of time. Network monitors do not decode the contents of frames. Monitors are useful for baselining, in which the activity on a network is sampled over a period of time to establish a normal performance profile or baseline.

Monitors collect information such as packet sizes, numbers of packets, error packets, overall usage of a connection, the number of hosts and their MAC addresses, and details about communications between hosts and other devices. You can use the data to create profiles of LAN traffic as well locate traffic overloads, plan for network expansion, detect intruders, establish baseline performance, and distribute traffic more efficiently.

A network analyzer (also called a protocol analyzer) decodes the various protocol layers in a recorded frame and presents them as readable abbreviations or summaries, detailing which layer is involved (physical, data link, and so forth) and what function each byte or byte content serves.

Most network analyzers can perform many of the following functions:

- Filtering traffic that meets certain criteria so that, for example, all traffic to and from a particular device can be captured
- Time-stamping captured data
- Presenting protocol layers in an easily readable form
- Generating frames and transmitting them onto the network
- Incorporating an "expert" system in which the analyzer uses a set of rules, combined with information about the network configuration and operation, to diagnose and solve, or offer potential solutions to, network problems

TDRs and OTDRs

TDRs are at the top end of the cable testing spectrum. These devices can quickly locate open and short circuits, crimps, kinks, sharp bends, impedance mismatches, and other defects in metallic cables.

A TDR works by "bouncing" a signal off the end of the cable. Opens, shorts, and other problems reflect the signal back at different amplitudes, depending on the problem. A TDR measures how much time it takes for the signal to reflect and calculates the distance to a fault in the cable. TDRs can also be used to measure the length of a cable or calculate the propagation rate based on a configured cable length.

Fiber-optic measurement is performed by an OTDR. OTDRs can accurately measure the length of the fiber, locate cable breaks, measure the fiber attenuation, and measure splice or connector losses. An OTDR can be used to ascertain the "signature" of a particular installation, noting attenuation and splice losses. This baseline measurement can then be compared with future signatures when a problem in the system is suspected.