CHAPTER 2

Troubleshooting Tools

This chapter presents information about the wide variety of tools available to assist you in troubleshooting your internetwork. This includes information on using router diagnostic commands, Cisco network management tools, and third-party troubleshooting tools.

Using Router Diagnostic Commands

Cisco routers provide numerous integrated commands to assist you in monitoring and troubleshooting your internetwork. The following sections describe the basic use of these commands:

- The **show** commands help monitor installation behavior and normal network behavior, as well as isolate problem areas.
- The **debug** commands assist in the isolation of protocol and configuration problems.
- The **ping** commands help determine connectivity between devices on your network.
- The **trace** commands provide a method of determining the route by which packets reach their destination from one device to another.

Using show Commands

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

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

The following are some of the most commonly used **show** commands:

- **show version**—Displays the configuration of the system hardware, the software version, the names and sources of configuration files, and the boot images.
- **show running-config**—Displays the router configuration currently running.
- **show startup-config**—Displays the router configuration stored in nonvolatile RAM (NVRAM).

- **show interfaces**—Displays statistics for all interfaces configured on the router or access server. The resulting output varies, depending on the network for which an interface has been configured.
- **show controllers**—Displays statistics for interface card controllers.
- **show flash**—Displays the layout and contents of Flash memory.
- **show buffers**—Displays statistics for the buffer pools on the router.
- **show memory summary**—Displays memory pool statistics and summary information about the activities of the system memory allocator, and gives a block-by-block listing of memory use.
- **show process cpu**—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 cdp neighbors**—Provides a degree of reachability information of directly connected Cisco devices. This is an extremely useful tool to determine the operational status of the physical and data link layer. Cisco Discovery Protocol (CDP) is a proprietary data link layer protocol.
- **show debugging**—Displays information about the type of debugging that is enabled for your router.

You can always use the **?** at command line for a list of subcommands.

Like the **debug** commands, some of the **show** commands listed previously are accessible only at the router's privileged exec mode (enable mode). This will be explained further in the "Using **debug** commands" section.

Hundreds of other **show** commands are available. For details on using and interpreting the output of specific **show** commands, refer to the Cisco Internetwork Operating System (IOS) command references.

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. To access and list the privileged exec commands, enter this code:

Router> enable

Password: XXXXXX

Router# ?

Note the change in the router prompts here. The # prompt (instead of the normal > prompt) indicates that you are in the privileged exec mode (enable mode).

Caution Exercise care when using **debug** commands. Many **debug** commands are processor-intensive and can cause serious network problems (such as 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).

Use **debug** commands to isolate problems, not to monitor normal network operation. Because the high processor overhead of **debug** commands can disrupt router operation, you should use them only when you are looking for specific types of traffic or problems, and have narrowed your problems to a likely subset of causes.

Output formats vary with each **debug** command. 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 Use 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. The **enable** exec command places the router in the privileged exec mode. After entering the **enable** password, you receive a prompt that consists of the router name with a **#** symbol.

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

By redirecting output to your current terminal display, you can view **debug** command output remotely, without being connected through the console port.

If you use **debug** commands at the console port, character-by-character processor interrupts are generated, maximizing the processor load already caused by using **debug**.

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 is described in the *Debug Command Reference*.

This book refers to specific **debug** commands that are useful when troubleshooting specific problems. Complete details regarding the function and output of **debug** commands are provided in the *Debug Command Reference*.

In many situations, using third-party diagnostic tools can be more useful and less intrusive than using **debug** commands. For more information, see the section "Third-Party Troubleshooting Tools," later in this chapter.

Using the ping Commands

To check host reachability and network connectivity, use the **ping** command, which can be invoked from both user exec mode and privileged exec mode. After you log in to the router or access server, you are automatically in user exec command mode. The exec commands available at the user level are a subset of those available at the privileged level. In general, the user exec commands enable you to connect to remote devices, change terminal settings on a temporary basis, perform basic tests, and list system information. The **ping** command can be used to confirm basic network connectivity on AppleTalk, ISO Connectionless Network Service (CLNS), IP, Novell, Apollo, VINES, DECnet, or XNS networks.

For IP, the **ping** command sends Internet Control Message Protocol (ICMP) Echo messages. ICMP is the Internet protocol that reports errors and provides information relevant to IP packet addressing. 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. To enter **ping** extended command mode, enter **yes** at the extended commands prompt of the **ping** command.

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

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

Using the trace Commands

The **trace** user exec command discovers the routes that a router's packets follow when travelling 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 works by using 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, **trace** 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.

As with **ping**, it is a good idea to use the **trace** command when the network is functioning properly to see how the command works under normal conditions and 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*.

Using Cisco Network Management Tools

Cisco offers the CiscoWorks 2000 family of management products that provide design, monitoring, and troubleshooting tools to help you manage your internetwork.

The following internetwork management tools are useful for troubleshooting internetwork problems:

- CiscoView provides dynamic monitoring and troubleshooting functions, including a graphical display of Cisco devices, statistics, and comprehensive configuration information.
- Internetwork Performance Monitor (IPM) empowers network engineers to proactively troubleshoot network response times utilizing real-time and historical reports.
- The TrafficDirector RMON application, a remote monitoring tool, enables you to gather data, monitor activity on your network, and find potential problems.
- The VlanDirector switch management application is a management tool that provides an accurate picture of your VLANs.

CiscoView

CiscoView graphical management features provide dynamic status, statistics, and comprehensive configuration information for Cisco internetworking products (switches, routers, hubs, concentrators, and access servers). CiscoView aids network management by displaying a physical view of Cisco devices and color-coding device ports for at-a-glance port status, allowing users to quickly grasp essential information. Features include the following:

- Graphical displays of Cisco products from a central location, giving network managers a complete view of Cisco products without physically checking each device at remote sites
- A continuously updated physical view of routers, hubs, switches, or access servers in a network, regardless of physical location
- Updated real-time monitoring and tracking of key information and data relating to device performance, traffic, and usage, with metrics such as utilization percentage, frames transmitted and received, errors, and a variety of other device-specific indicators
- The capability to modify configurations such as trap, IP route, virtual LAN (VLAN), and bridge configurations

Internetwork Performance Monitor

IPM is a network management application that enables you to monitor the performance of multiprotocol networks. IPM measures the response time and availability of IP networks on a hop-by-hop (router-to-router) basis. It also measures response time between routers and the mainframe in Systems Network Architecture (SNA) networks.

Use IPM to perform the following tasks:

- Troubleshoot problems by checking the network latency between devices
- Send Simple Network Management Protocol (SNMP) traps and SNA alerts when a user-configured threshold is exceeded, when a connection is lost and re-established, or when a timeout occurs
- Analyze potential problems before they occur by accumulating statistics, which are used to model and predict future network topologies
- Monitor response time between two network end points

The IPM product is composed of three parts: the IPM server, the IPM client application, and the response time reporter (RTR) feature of the Cisco IOS software.

The TrafficDirector RMON Application

The *TrafficDirector* advanced packet filters let users monitor all seven layers of network traffic. Using Cisco IOS embedded RMON agents and SwitchProbe standalone probes, managers can view enterprise-wide network traffic from the link, network, transport, or application layers. The TrafficDirector multilayer traffic summary provides a quick, high-level assessment of network loading and protocol distributions. Network managers then “zoom in” on a specific segment, ring, switch port, or trunk link and apply real-time analysis and diagnostic tools to view hosts, conversations, and packet captures.

TrafficDirector threshold monitoring enables users to implement a proactive management environment. First, thresholds for critical Management Information Base (MIB) variables are set within the RMON agent. When these thresholds are exceeded, traps are sent to the appropriate management station to notify the network administrator of an impending problem.

The VlanDirector Switch Management Application

The *VlanDirector* switch management application simplifies VLAN port assignment and offers other management capabilities for VLANs. VlanDirector offers the following features for network administrators:

- Accurate representation of the physical network for VLAN design and configuration verification
- Capability to obtain VLAN configuration information on a specific device or link interface
- Discrepancy reports on conflicting configurations
- Capability to troubleshoot and identify individual device configurations that are in error with system-level VLANs
- Quick detection of changes in VLAN status of switch ports
- User authentication and write protection security

Third-Party Troubleshooting Tools

In many situations, third-party diagnostic tools can be more useful than commands that are integrated into the router. For example, enabling a processor-intensive **debug** command can be disastrous in an environment experiencing excessively high traffic levels. However, 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. The following are some typical third-party troubleshooting tools used for troubleshooting internetworks:

- Volt-ohm meters, digital multimeters, and cable testers are useful in testing the physical connectivity of your cable plant.
- Time domain reflectors (TDRs) and optical time domain reflectors (OTDRs) are devices that assist in the location of cable breaks, impedance mismatches, and other physical cable plant problems.
- Breakout boxes, fox boxes, and BERTs/BLERTs are useful for troubleshooting problems in peripheral interfaces.
- Network monitors provide an accurate picture of network activity over a period of time by continuously tracking packets crossing a network.
- Network analyzers such as sniffers decode problems at all seven OSI layers and can be identified automatically in real time, providing a clear view of network activity and categorizing problems by criticality.

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 measure parameters such as AC and DC voltage, current, resistance, capacitance, and cable continuity. They are used to check physical connectivity.

Cable testers (scanners) also enable you to check physical connectivity. Cable testers are available for shielded twisted-pair (STP), unshielded twisted-pair (UTP), 10BaseT, and coaxial and twinax cables. A given cable tester might be capable of performing any of the following functions:

- Test and report on cable conditions, including near-end crosstalk (NEXT), 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. Because of the relatively high cost of this cable and its installation, fiber-optic cable should be tested both before installation (on-the-reel testing) and after installation. Continuity testing of the fiber 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.

TDRs and OTDRs

At the top end of the cable testing spectrum are *TDRs*. 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. Some TDRs can also 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 take 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.

Breakout Boxes, Fox Boxes, and BERTs/BLERTs

Breakout boxes, *fox boxes*, and *bit/block error rate testers* (*BERTs/BLERTs*) are digital interface testing tools used to measure the digital signals present at PCs, printers, modems, the channel service unit/digital service unit (CSU/DSU), and other peripheral interfaces. These devices can monitor data line conditions, analyze and trap data, and diagnose problems common to data communication systems. Traffic from data terminal equipment (DTE) through data communications equipment (DCE) can be examined to help isolate problems, identify bit patterns, and ensure that the proper cabling has been installed. These devices cannot test media signals such as Ethernet, Token Ring, or FDDI.

Network Monitors

Network monitors continuously track packets crossing a network, providing an accurate picture of network activity at any moment, or a historical record of network activity over a period of time. They 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, the number 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. This data can be used to create profiles of LAN traffic as well as to assist in locating traffic overloads, planning for network expansion, detecting intruders, establishing baseline performance, and distributing traffic more efficiently.

Network Analyzers

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:

- Filter traffic that meets certain criteria so that, for example, all traffic to and from a particular device can be captured
- Time stamp-captured data
- Present protocol layers in an easily readable form
- Generate frames and transmit them onto the network
- Incorporate 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