CHAPTER 1

Troubleshooting Overview

Internetworks come in a variety of topologies and levels of complexity—from single-protocol, point-to-point links connecting cross-town campuses to highly meshed, large-scale wide-area networks (WANs) traversing multiple time zones and international boundaries. The overall trend is toward increasingly complex environments, involving multiple media, multiple protocols, and sometimes interconnection to "unknown" networks. As a result, the potential for connectivity and performance problems in internetworks is often high, even when all elements of an environment appear to be fully operational. The objective of this publication is to help you identify potential problem sources in your internetwork and then to resolve problems that arise.

Focus on Symptoms, Causes, and Actions

Failures in internetworks are characterized by certain *symptoms* (such as clients being unable to access specific servers). Each symptom can be diagnosed based on *problems* or *causes* by using specific troubleshooting tools. Once identified, each cause can be remedied by implementing a series of *actions*.

Use this manual as a starting point to develop a problem-solving process for your internetwork. This publication aims to integrate the process of symptom definition, problem identification, and action implementation into an overall troubleshooting model. It illustrates how problems can be detected and diagnosed within the context of case environments.

What This Guide Is Not

With these broad objectives stated, it is equally important to outline topics that are beyond the scope of this publication.

- This publication is not to intended to be the last word in troubleshooting. It does not guide you through every possible error condition, obscure anomaly, or subtle protocol problem. Instead, *Troubleshooting Internetworking Systems* is a roadmap that illustrates the *common* pitfalls and problems most frequently encountered by internetwork administrators.
- *Troubleshooting Internetworking Systems* is not a maintenance and repair guide; nor is it a reference guide. Refer to your hardware installation and maintenance publication for additional details regarding maintenance of router hardware. Refer to the *Router Products Configuration Guide* and *Router Products Command Reference* publications for configuration command details.

  This publication recommends actions for resolving a spectrum of common internetworking problems. In general, it assumes that routers are operational. However, several brief tables provided later in this chapter summarize typical router hardware problems.

- Finally, *Troubleshooting Internetworking Systems* is not a *network* troubleshooting publication. Although suggestions about troubleshooting certain media (including Ethernet, FDDI, serial, and Token Ring) are provided, the focus of the publication is not on troubleshooting media, per se. Several commercially available publications provide this information, such as *LAN Troubleshooting Handbook* by Mark Miller. Appendix E, "References and Recommended Reading," suggests some others.

What, then, does that leave? The discussions that follow outline how you can use this publication to resolve common *internetworking* problems.

The remainder of this overview addresses the following topics:

- Using this publication
- Using router diagnostic tools
- Using CiscoWorks to troubleshoot your internetwork
- Using third-party troubleshooting tools

Using This Publication

Troubleshooting Internetworking Systems focuses on identifying failure symptoms and their associated causes, detecting and isolating those causes, and then resolving problems through specific actions. The symptom discussions and scenarios provided concentrate on issues pertaining to *router* configuration and the interoperation of nodes within a multivendor internetwork.

Within this context, use *Troubleshooting Internetworking Systems* as a guide to do the following:

- Identify possible problem causes when your internetwork is down or slow
- Get direction about resolving problems
- See what kinds of problems have been encountered and resolved in the past
- Avoid falling into the same traps
- Develop your own processes for troubleshooting

To support these activities, this guide uses three key organizational elements (defined in the discussions that follow):

- General problem-solving model
- Symptom modules
- Troubleshooting scenarios

In addition, this overview provides guidelines for the following tasks:

- Using this publication to troubleshoot problems
- Using this publication as a tutorial

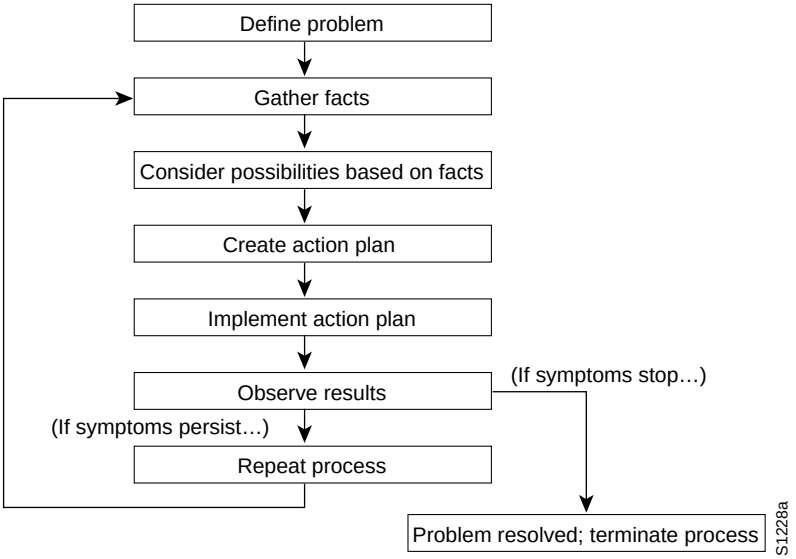
General Problem-Solving Model

Before embarking on your troubleshooting effort, be sure to have a *plan* in place to identify prospective problems, isolate the likely causes of those problems, and then systematically eliminate each potential cause.

The problem-solving model that follows is not a rigid "cookbook" for solving internetworking problems. It is a foundation from which you can build problem-solving plans to suit your particular environment.

Figure 1-1 illustrates process flow for the general problem-solving model described in the steps that follow.

Figure 1-1 General Problem-Solving Flow Diagram



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

Step 1 Define problems in terms of a set of *symptoms* and associated *causes*.

Make a clear problem statement. You must recognize and define the problem/failure mode by identifying any associated general symptoms and then identifying the possible kinds of problems that result in the listed symptoms.

For example, certain hosts might not be responding to service requests from certain clients (a symptom). Possible causes include a misconfigured host, bad interface cards, or missing router commands.

Step 2 Gather facts.

After you list your symptoms and identify possible causes, collect facts. Fact gathering might involve obtaining network analyzer traces, serial line traces, stack dumps, core dumps, and output from a variety of **show** and **debug** privileged EXEC commands. The definition of the problem will point to a more specific set of data to gather.

Step 3 Consider possibilities based on facts.

Armed with a working knowledge of the product, you should be able to eliminate entire classes of problems associated with system software and hardware. This way, you can narrow the scope of interest to only those portions of the product, media, or host problems that are relevant to the specific problem or failure mode.

Step 4 Create an action plan.

The action plan should be based on the set of possibilities you just derived. Your action plan must limit manipulation to *one* variable at a time. This approach allows you to reproduce a given solution to a specific problem. If you alter more than one variable simultaneously, you might solve the problem, but identifying the specific change that eliminated the symptom becomes more difficult.

Step 5 Implement the action plan.

This phase consists of executing the action plan you just created. It is important to be very specific in creating the action plan (that is, identify a specific set of steps and then carefully implement each step).

Step 6 Observe the results of each action.

After having manipulated a variable in an attempt to find a solution to a problem, be sure to gather results based on this action plan (obtain relevant traces, capture **debug** privileged EXEC command data, examine output of **show** EXEC commands, and so forth). This data can be used to fine-tune the action plan until the proper solution is achieved. It is during this phase that you must determine whether the problem has been resolved. This is the exit point of the loop shown in Figure 1-1.

Step 7 Narrow possibilities based on results.

In order to reach a point where you can exit this problem/solution loop, you must strive to make continuous progress toward a smaller set of possibilities, until you are left with only one.

Step 8 Repeat the problem-solving process.

After narrowing your possibility list, repeat the process, starting with a new action plan based on a new (possibly shorter or longer) list of possibilities. Continue the process until a solution is found. Problem resolution can consist of several modifications to hosts, routers, or media.

Note If you exhaust all the common causes and actions (either those suggested here or ones that you have identified for your environment), your last recourse is to contact your router technical support representative. Appendix A, “Technical Support Information List,” outlines information needed by technical support representatives to troubleshoot internetworking problems. One objective of this publication is to help you develop your own processes for gathering data, resolving problems, and preventing problems from recurring (with a minimum of downtime and external intervention).

Symptom Modules

The *symptom modules* in this publication are *not* comprehensive case studies, but instead are brief snapshots of likely problems associated with a specific symptom. Use them as tools for compiling lists of candidate problems (by symptom). The connectivity and performance chapters are organized around the symptom modules. These chapters are not meant to be read from beginning to end; rather, specific information in these symptom-oriented chapters is intended to be used as needed.

Each symptom module includes a brief summary statement and a table listing possible causes. A series of suggested actions is provided for each listed cause to help you determine whether the specific cause is actually the source of the symptom and then to resolve the problem.

Troubleshooting Scenarios

The *troubleshooting scenarios* combine the problems and actions presented in symptom modules with the methods outlined in the section "General Problem-Solving Model" within a context of integrated *case studies*.

Each scenario outlines a set of "observed" symptoms, an internetworking environment, and a list of likely problems for each symptom. Scenarios focus on the process of problem diagnosis (discovery), isolation, and resolution. Not all symptoms discussed in this publication are explored in the scenarios. Instead, selected multiple symptoms are addressed per scenario. An effort has been made to choose common, realistic problems.

Using This Publication to Troubleshoot Specific Symptoms

When using this publication to troubleshoot your internetwork, follow these general steps:

Step 1 Identify symptoms encountered on your internetwork.

Step 2 Eliminate hardware as a possible problem by either fixing any hardware problems or ruling out hardware as a possible cause. (For hardware troubleshooting details, refer to the "Troubleshooting Router Startup Problems" chapter.)

Step 3 Each of the "Troubleshooting Connectivity" chapters offers a "Connectivity Symptoms" section which contains individual *symptom modules* that describe a symptom, possible causes for the symptom, and suggested actions to take to resolve each cause. To identify symptoms similar to those you are experiencing, refer to the chapters that address the technologies or protocols used in your internetwork.

Step 4 Within the appropriate symptom modules, evaluate the problems listed and compare them to your internetworking environment. Note those problems that could apply to your situation.

Step 5 Systematically apply actions for each suspected problem until all symptoms are eliminated, or the possible cause list is exhausted.

Step 6 If problems persist after all of the suggested actions are performed, contact your technical support representative.

Using This Publication as a Tutorial

When using this guide as a tutorial, associated activities are a little less structured than when using it to troubleshoot a specific problem. Nonetheless, you can think of the learning process as a series of steps, as follows:

Step 1 Review the section "General Problem-Solving Model" earlier in this chapter to see recommendations for approaching the troubleshooting process.

Step 2 Read through the troubleshooting scenarios presented in the "Troubleshooting Connectivity" chapters and those in the "Performance Problem Scenarios" chapter.

Step 3 Characterize similarities or differences between these scenarios and your own internetworking environment.

Step 4 Review the symptom modules associated with protocols or technologies implemented in your internetwork.

Step 5 Develop a list of possible symptoms and problems that you encounter in your internetwork. Be as specific as possible. Keep this list on hand in a troubleshooting binder.

Step 6 When similar symptoms occur, use this list to start the troubleshooting process. Remember to modify your problem-solving procedures as you find subtleties associated with your implementation. The key to developing an effective response to problems in your environment is being able to identify the causes of those problems and then implement an action plan. Whatever you can do to preempt time spent in diagnosis will pay off in terms of reducing downtime.

Step 7 Periodically revisit this process to accommodate changes to your internetwork.

Using Router Diagnostic Tools

The following tools are universally applicable when gathering information to troubleshoot problems in router-based internetworks:

- **show** EXEC commands (Although many of these commands are user-accessible, other relevant **show** commands for troubleshooting are privileged EXEC commands.)
- **debug** privileged EXEC commands
- **ping** (Echo Request/Echo Reply) EXEC command
- **trace** EXEC command
- **exception dump** global configuration command and **write core** privileged EXEC command

The discussions that follow summarize using these tools. Appendix C of this publication, "Creating Core Dumps," describes the **exception dump** and **write core** commands. The *Debug Command Reference* publication defines the **debug** commands for protocols and technologies discussed in this publication. The *Router Products Command Reference* publication details the **show**, **ping**, and **trace** commands.

Using show Commands

The **show** commands are among your most important tools for understanding the status of a router, detecting neighboring routers, monitoring the network in general, and isolating problems in your internetwork.

These commands are essential in almost any troubleshooting and monitoring situation. Use **show** commands for the following activities:

- 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

For some protocols, such as Novell IPX and AppleTalk, the methodical use of **show** commands is one of the most reliable ways to create a topology map of your internetwork. To create a topology map, use the **show** commands as follows:

Step 1 Use the appropriate **show** *protocol* **route** EXEC command (such as **show novell route**) to determine which neighbors are directly connected.

Step 2 Record the names and network addresses of all directly connected neighbors.

Step 3 Open a connection to each of these directly connected neighbors and obtain the output of the **show** *protocol* **route** command for those neighbors.

Step 4 Continue this process for all routers in your internetwork.

The resulting map reflects all paths to the routers in your internetwork.

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. But be aware that these commands often generate data that is of little use for a specific problem.

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

Note You can use the **terminal monitor** privileged EXEC command to copy **debug** command output and system error messages to your current terminal display—as well as to the console terminal. This permits you to establish a Telnet connection to the router and view **debug** command output remotely, without being connected through the console port.

This publication refers to specific **debug** commands that are useful when troubleshooting specific problems. Complete details regarding information provided in **debug** command output are provided in the *Debug Command Reference* publication. However, the *Debug Command Reference* does not document *every* **debug** command that exists in the router code, but only those identified as particularly useful for troubleshooting specific media and protocols.

Caution The use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router when internetworks are experiencing high load conditions. When you finish using a **debug** command, remember to disable it with its specific **no debug** command or with the **no debug all** command (the **undebug** command is also accepted).

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* publication.

Using ping and trace Commands

Two of the most useful internetworking diagnostic tools are the **ping** and **trace** EXEC commands. The *ping* capability provides a simple mechanism to determine whether packets are reaching a particular destination. Routers from other manufacturers may not forward pings, and some hosts may not reply normally, but even an error packet (ERPDU) response can be useful because it confirms the reachability of the host.

The *trace* capability allows you to determine the specific path taken to a destination and where packets are stopping. Together, these functions may be two of the most important troubleshooting tools available.

Both the **ping** and **trace** commands are available as both user-accessible EXEC commands and as privileged EXEC commands. Depending on the situation, the user-accessible EXEC command may be adequate for testing connectivity. However, if you intend to perform any custom tests, use the privileged EXEC command versions.

Note The **ping** and **trace** commands are protocol specific. The **ping** command can be used with AppleTalk, Banyan VINES, IP, ISO CLNS, Novell IPX, and XNS internetworks, and only routers running one of those protocols will respond. AppleTalk, Banyan VINES, IP, and ISO CLNS support the **trace** function. To use the **trace** command, one of these protocols must be enabled for routing, and only nodes running the specific protocol will respond.

Using Core Dumps

The **exception dump** global configuration command and **write core** privileged EXEC command are among the more obscure (although useful) diagnostic commands available in your router toolkit. When the system software fails, analyzing a *core dump* (produced by the **exception dump** command) is sometimes the only way to determine what happened. The **write core** command is useful if the router is malfunctioning, but has not crashed.

Caution Use these commands only in coordination with a qualified technical support representative. The resulting binary file must be directed to a specific UNIX syslog server and subsequently interpreted by qualified technical personnel. Appendix C, “Creating Core Dumps,” briefly describes the process.

Developing a Strategy for Isolating Problems

One important consideration to remember when troubleshooting broken interconnections is that *normally* everything does not break at the same time. As a result, when trying to isolate a problem, you can typically work out from an operational node to the point of failure. The following basic steps should help when you are trying to isolate the source of connection disruption:

Step 1 First, determine whether the local host that is experiencing connectivity problems is properly configured.

Step 2 For AppleTalk, Banyan VINES, IP, ISO CLNS, and Novell IPX internetworks, use the **ping** or **trace** EXEC commands (as applicable) to determine whether the routers and bridges through which the local host must communicate can respond. Start with the most local router or bridge and progressively "ping out" through the internetwork.

Step 3 If you cannot get through a particular router, examine the configuration of the router and use the various **show** commands to determine the state of that router.

Step 4 If you access all the routers in the path, check the configuration of the remote host (or get the help of someone to do so).

Step 5 Use the appropriate **show** *protocol* **route** command to see if the hosts in question appear in the routing tables. Use other protocol-specific **show** commands to check for anomalies.

Using CiscoWorks to Troubleshoot Your Internetwork

The CiscoWorks product is a set of router management applications that allows you to manage your internetwork from a central location. You can use CiscoWorks software to monitor and troubleshoot complex internetworks. Because CiscoWorks uses the Simple Network Management Protocol (SNMP), it can monitor and control any SNMP device on an internetwork. The CiscoWorks software comprises five different applications: configuration management, fault management, accounting management, performance management, and security management.

In addition to the basic SNMP management functions, the CiscoWorks software provides a fully integrated relational database and uses built-in SunNet Manager (SNM) capabilities to produce a dynamic, user-configurable visual network map. The automatic map-generation features associated with the CiscoWorks Path Tool capabilities can help you visually trace the routes to problem nodes. Tools that can help you isolate connectivity and performance problems are outlined briefly in the following discussions. Refer to the *CiscoWorks User Guide* for complete details about using CiscoWorks to monitor and control your internetwork.

Using CiscoWorks to Troubleshoot Connectivity Problems

Use the following CiscoWorks fault management applications when troubleshooting connectivity problems in your internetwork:

- Device Monitor—Monitors specific devices for environmental and interface information. Sends event information to SNM that causes a glyph to change state.
- Path Tool—Graphically displays a route of the path from a source device to a destination device.
- Environmental Monitor—Graphically displays the temperature and voltage data from an AGS+ router.
- Real-Time Graphs—Monitors the behavior of device interfaces or other network elements suspected of operating in a degraded mode and displays them in a graph.
- Show Commands—Enable you to view data similar to output from router **show** EXEC commands.

- Health Monitor—Provides information about the health of a device with access to several CiscoWorks applications on one window (including Show Commands and Real-Time Graphs) to monitor router activity.
- Contacts—Provides quick access to find your emergency contact person for a particular device.
- Log Manager—Enables you to store, query, and delete messages gathered from CiscoWorks applications and Cisco Systems devices on the internetwork.

Using CiscoWorks to Troubleshoot Performance Problems

Use the following CiscoWorks performance management applications when troubleshooting performance problems in your internetwork:

- Device Polling—Probes and extracts data about the condition of your network devices.
- Polling Summary—Displays polling data, and stops and starts polling.
- Real-Time Graphs—Monitors the behavior of device interfaces or other network elements suspected of operating in a degraded mode and displays them in a graph.
- Path Tool—Graphically displays a route of the path from a source device to a destination device.
- Show Commands—Provide data similar to router **show** EXEC commands output.
- Sybase DWB—Allows you to access the Sybase Data Workbench application to write reports.

Using Third-Party Troubleshooting Tools

This publication emphasizes diagnostic tools provided with the router. However, other troubleshooting tools also are discussed in the symptom modules and scenarios.

In some cases, third-party diagnostic tools can be more useful than integrated tools. For example, enabling a **debug** privileged EXEC command can be disastrous in any environment experiencing excessively high traffic levels. Attaching a network to the suspect network is less intrusive and more likely to yield applicable information without exacerbating load problems for a router.

The following list summarizes some typical third-party troubleshooting tools:

- *Time Domain Reflectometer (TDR)*—A TDR transmits a short pulse of known amplitude and duration down a cable and measures the corresponding amplitude and time delay associated with resultant signal reflections. TDRs are available for all LAN types. Optical TDRs provide a similar test capability for fiber cable.
- *Optical Power Source and Meter*—This device employs an optical power source connected to one end of a fiber cable and a meter placed at the other end to measure optical power. Also called a “light meter,” this device is a cost-effective alternative to an optical TDR.

- *Oscilloscope*—Oscilloscopes graphically display signal voltage per unit of time; commonly used to measure voltages on EIA/TIA-232 and EIA/TIA-422 interfaces.

Note Prior to the acceptance of the EIA/TIA standard by the ANSI committee, these interface standards were referred to as recommended standards RS-232 and RS-422.

- *Breakout Box* —A breakout box displays and monitors status of EIA/TIA-232-D interface leads between data terminal equipment (DTE) and data circuit-terminating equipment (DCE). Breakout boxes are useful for reconfiguring interfaces.
- *Network Analyzer*—Network analyzers (also known as "protocol analyzers" and "LAN analyzers") capture, record, and analyze frames transmitted on a network. Analyzers attach to a network just as any node does. All analyzers support a range of physical interface specifications (including Ethernet, Token Ring, and FDDI), as well as a spectrum of network protocols (including TCP/IP, Novell IPX, IBM SNA, AppleTalk, DECnet, and ISO CLNS).
- *WAN/Serial Line Analyzer*—WAN analyzers generally focus on WAN/serial line analysis, but can include LAN analysis capabilities. WAN analyzers support a range of physical interfaces (such as EIA/TIA-232, EIA/TIA-422, EIA/TIA-449, T1/E1, ITU-T V.35, and ITU-T X.21) and protocols (including HDLC, SDLC, Frame Relay, and ISDN).

Note The ITU-T carries out the functions of the former Consultative Committee for International Telegraph and Telephone (CCITT).
