Troubleshooting Methodologies

17.6.1

Basic Troubleshooting Approaches

In the previous two topics, you learned about some utilities and commands that you can use to help identify problem areas in your network. This is an important part of troubleshooting. There are many ways to troubleshoot a network problem. This topic details a structured troubleshooting process that can help you to become a better network administrator. It also provides a few more commands to help you resolve problems. Network problems can be simple or complex, and can result from a combination of hardware, software, and connectivity issues. Technicians must be able to analyze the problem and determine the cause of the error before they can resolve the network issue. This process is called troubleshooting.

A common and efficient troubleshooting methodology is based on the scientific method.

The table shows the six main steps in the troubleshooting process.

Step	Description
Step 1. Identify the Problem	• This is the first step in the troubleshooting process. • Although tools can be used in this step, a conversation with the user is often very helpful.
Step 2. Establish a Theory of Probable Causes	• After the problem is identified, try to establish a theory of probable causes. • This step often yields more than a few probable causes to the problem.
Step 3. Test the Theory to Determine Cause	• Based on the probable causes, test your theories to determine which one is the cause of the problem. • A technician will often apply a quick procedure to test and see if it solves the problem. • If a quick procedure does not correct the problem, you might need to research the problem further to establish the exact cause.
Step 4. Establish a Plan of Action and Implement the Solution	After you have determined the exact cause of the problem, establish a plan of action to resolve the problem and implement the solution.
Step 5. Verify Solution and Implement Preventive Measures	• After you have corrected the problem, verify full functionality. • If applicable, implement preventive measures.
Step 6. Document Findings, Actions, and Outcomes	• In the final step of the troubleshooting process, document your findings, actions, and outcomes. • This is very important for future reference.

To assess the problem, determine how many devices on the network are experiencing the problem. If there is a problem with one device on the network, start the troubleshooting process at that device. If there is a problem with all devices on the network, start the troubleshooting process at the device where all other devices are connected. You should develop a logical and consistent method for diagnosing network problems by eliminating one problem at a time.

17.6.2

Resolve or Escalate?

In some situations, it may not be possible to resolve the problem immediately. A problem should be escalated when it requires a manager decision, some specific expertise, or network access level unavailable to the troubleshooting technician.

For example, after troubleshooting, the technician concludes a router module should be replaced. This problem should be escalated for manager approval. The manager may have to escalate the problem again as it may require the approval of the financial department before a new module can be purchased.

A company policy should clearly state when and how a technician should escalate a problem.

17.6.3

The debug Command

OS processes, protocols, mechanisms and events generate messages to communicate their status. These messages can provide valuable information when troubleshooting or verifying system operations. The IOS **debug** command allows the administrator to display these messages in real-time for analysis. It is a very important tool for monitoring events on a Cisco IOS device.

All **debug** commands are entered in privileged EXEC mode. The Cisco IOS allows for narrowing the output of **debug** to include only the relevant feature or subfeature. This is important because debugging output is assigned high priority in the CPU process and it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems.

For example, to monitor the status of ICMP messages in a Cisco router, use **debug ip icmp**, as shown in the example.

```
R1# debug ip icmp
ICMP packet debugging is on
R1#
R1# ping 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
R1#
*Aug 20 14:18:59.605: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology BASE, dscp 0 topoid 0
*Aug 20 14:18:59.606: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology BASE, dscp 0 topoid 0
*Aug 20 14:18:59.608: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology BASE, dscp 0 topoid 0
*Aug 20 14:18:59.609: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology BASE, dscp 0 topoid 0
*Aug 20 14:18:59.611: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology BASE, dscp 0 topoid 0
R1#
```

To list a brief description of all the debugging command options, use the **debug ?** command in privileged EXEC mode at the command line.

To turn off a specific debugging feature, add the **no** keyword in front of the **debug** command:

```
Router# no debug ip icmp
```

Alternatively, you can enter the **undebug** form of the command in privileged EXEC mode:

```
Router# undebug ip icmp
```

To turn off all active debug commands at once, use the **undebug all** command:

```
Router# undebug all
```

Be cautious using some **debug** command. Commands such as **debug all** and **debug ip packet** generate a substantial amount of output and can use a large portion of system resources. The router could get so busy displaying **debug** messages that it would not have enough processing power to perform its network functions, or even listen to commands to turn off debugging. For this reason, using these command options is not recommended and should be avoided.

17.6.4

The terminal monitor Command

Connections to grant access to the IOS command line interface can be established in the following two ways:

- **Locally** - Local connections (i.e., console connection) require physical access to the router or switch console port using a rollover cable.
- **Remotely** - Remote connections require the use of Telnet or SSH to establish a connection to an IP configured device.

Certain IOS messages are automatically displayed on a console connection but not on a remote connection. For instance, **debug** output is displayed by default on console connections. However, **debug** output is not automatically displayed on remote connections. This is because **debug** messages are log messages which are prevented from being displayed on vty lines.

In the following output for instance, the user established a remote connection using Telnet from R2 to R1. The user then issued the **debug ip icmp** command. However, the command failed to display **debug** output.

```
R2# telnet 209.165.200.225
Trying 209.165.200.225 ... Open
 Authorized access only!
User Access Verification
Password:
R1> enable
Password:
R1# debug ip icmp
ICMP packet debugging is on
R1# ping 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
R1#
! No debug output displayed>
```

To display log messages on a terminal (virtual console), use the **terminal monitor** privileged EXEC command. To stop logging messages on a terminal, use the **terminal no monitor** privileged EXEC command.

For instance, notice how the **terminal monitor** command has now been entered and the **ping** command displays the **debug** output.

```
R1#  terminal monitor
R1# ping 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
R1#
*Aug 20 16:03:49.735: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology BASE, dscp 0 topoid 0
**Aug 20 16:03:49.737: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology BASE, dscp 0 topoid 0
**Aug 20 16:03:49.738: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology BASE, dscp 0 topoid 0
**Aug 20 16:03:49.740: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology BASE, dscp 0 topoid 0
**Aug 20 16:03:49.741: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology BASE, dscp 0 topoid 0
R1# no debug ip icmp
ICMP packet debugging is off
R1#
```

Note: The intent of the **debug** command is to capture live output for a short period of time (i.e., a few seconds to a minute or so). Always disable **debug** when not required.

17.6.5

Check Your Understanding - Troubleshooting Methodologies

Check your understanding of troubleshooting methodologies by choosing the BEST answer to the following questions.

1. A technician is troubleshooting a network problem and has just established a theory of probable causes. What would be the next step in the troubleshooting process?
   - ○ Document findings, actions, and outcomes.
   - ○ Establish a plan of action and implement the solution.
   - ○ Identify the problem.
   - ◉ Test the theory to determine cause.
   - ○ Verify solution and implement preventive measures.

2. A technician is troubleshooting a network problem. After troubleshooting, the technician concludes that a switch should be replaced. What should the technician do next?
   - ○ Email all users to let them know they are replacing a switch.
   - ◉ Escalate the trouble ticket to the manager to approve the change.
   - ○ Purchase a new switch and replace the defective one.
   - ○ Resolve the problem.

3. A technician is using the **debug ip icmp** privileged EXEC command to capture live router output. Which commands would stop this **debug** command on a Cisco router? (Choose two.)
   - ☐ **debug ip icmp off**
   - ☐ **no debug debug ip icmp**
   - ☑ **no debug ip icmp**
   - ☑ **undebug all**
   - ☐ **undebug debug ip icmp**

4. A technician has established a remote connection to router R1 to observe **debug** output. The technician enters the **debug ip icmp** command then pings a remote destination. However, no output is displayed. Which command would the technician have to enter to display log messages on a remote connection?
   - ○ **monitor debug output**
   - ○ **monitor terminal**
   - ◉ **terminal monitor**
   - ○ **terminal monitor debug**

Check

Show Me

Reset