



The Cisco Certified Network Associate Curriculum

Chapter 4

Router Components

Overview

Now that you have an understanding of the router command line interface, it is time to examine the router components that ensure efficient and effective delivery of data on a network. In this chapter, you will learn the correct procedures and commands to access a router, examine and maintain its components, and test its network connectivity.

After completing this chapter, you will be able to perform tasks related to:

4.1. Router Components

4.2. Router Show Commands

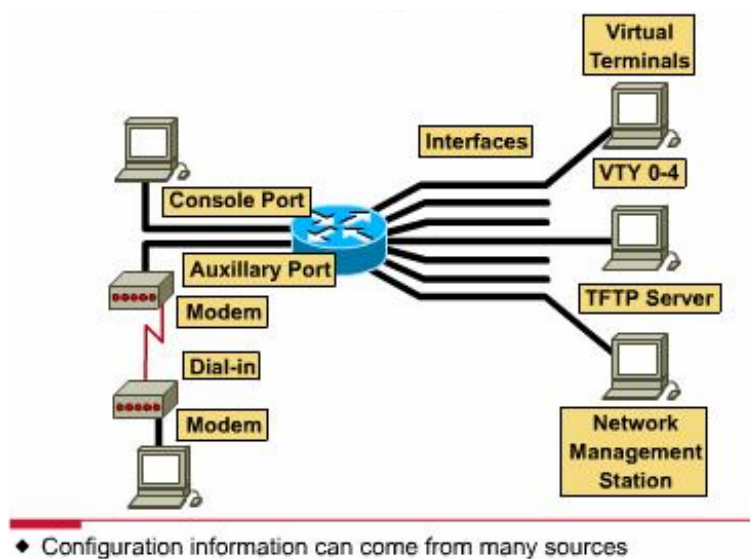
4.3. Router's Network Neighbors

4.4. Basic Networking Testing Commands

4.5. Lab: Troubleshooting Tools Challenge

External router configuration sources

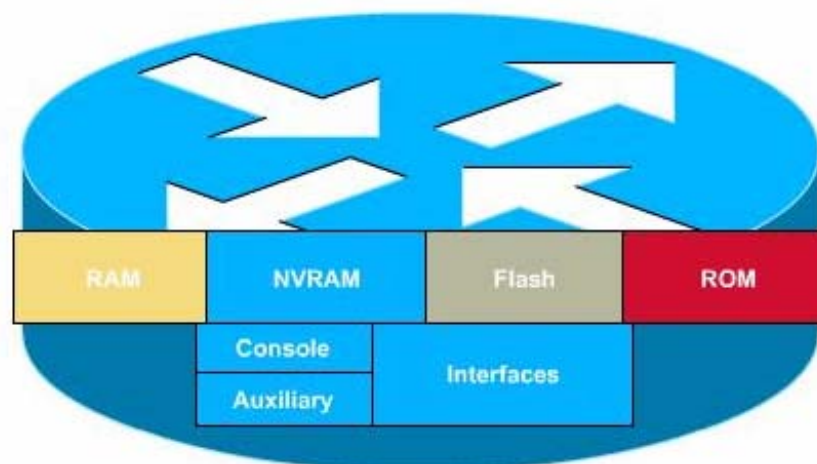
In this section, you will learn about the router components that play a key role in the configuration process. Knowing which components are involved in the configuration process gives you a better understanding of how the router stores and uses your configuration commands. Being aware of the steps that take place during router initialization will help you determine what and where problems may occur when you start up your router. You can configure a router from many external locations as shown in the Figure, including the following: from the console terminal (a computer connected to the router through a console port) during its installation via modem by using the auxiliary port from Virtual Terminals 0-4, after it has been installed on the network from a TFTP server on the network from a Network Management Station on the network



Internal router's configuration components

The internal architecture of the Cisco router supports components that play an important role in the startup process, as shown in the Figure. Internal router configuration components are as follows: *RAM/DRAM* -- stores routing tables, ARP cache, fast-switching cache, packet buffering (shared RAM), and packet hold queues; RAM also provides temporary and/or running memory for a router's configuration file while the router is powered; RAM content is lost during a power down or restart *NVRAM* -- non-volatile RAM stores the router's backup/startup configuration file;

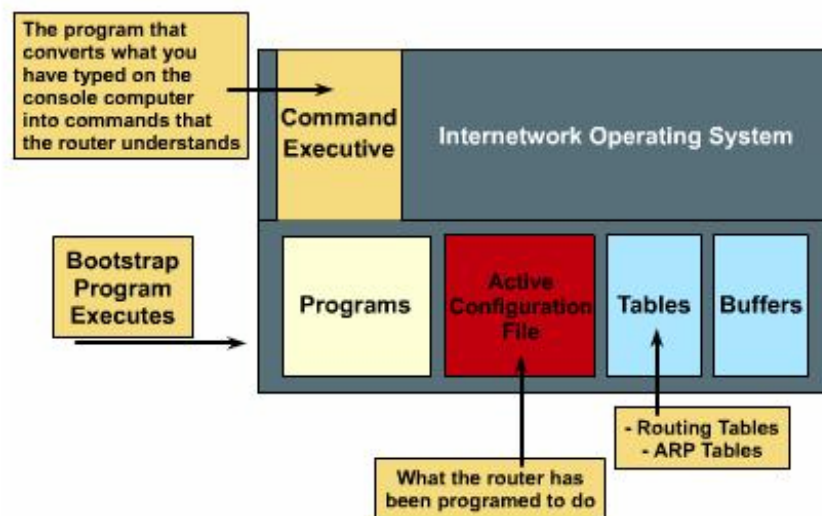
NVRAM content is retained during power down or restart *Flash* -- erasable, reprogrammable ROM that holds the operating system image and microcode; Flash memory enables software updates without removing and replacing processor chips; Flash content is retained during power down or restart; Flash memory can store multiple versions of IOS software *ROM* -- contains power-on diagnostics, a bootstrap program, and operating system software; software upgrades in ROM require removing and replacing pluggable chips on the CPU *Interfaces* -- network connections on the motherboard or on separate interface modules, through which packets enter and exit a router



RAM for working storage in the router

RAM is the working storage area for a router. When you turn a router on, the ROM executes a bootstrap program. This program performs some tests, and then loads the Cisco IOS software into memory. The command executive, or EXEC, is one part of the Cisco IOS software. EXEC receives and executes commands you enter for the router. As shown in the Figure, a router also uses RAM to store an active configuration file and tables of network maps and routing address lists. You can display the configuration file on a remote or console terminal.

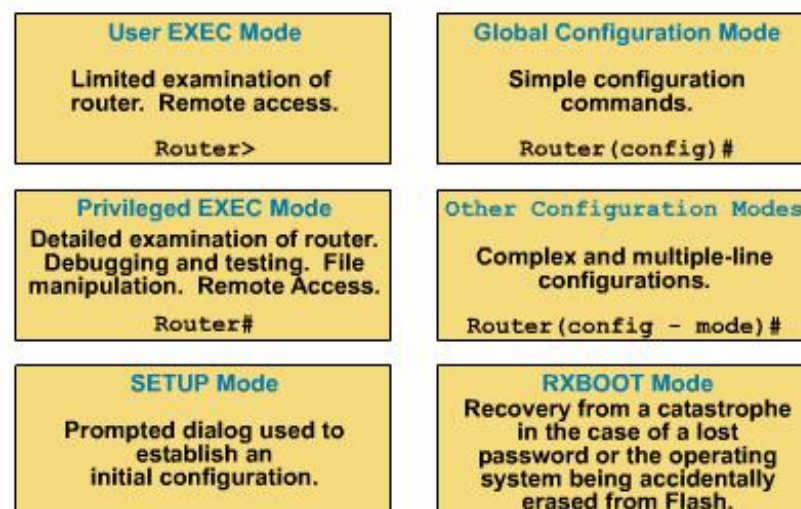
A saved version of this file is stored in NVRAM. It is accessed and loaded into main memory each time a router initializes. The configuration file contains global, process, and interface information that directly affects the operation of a router and its interface ports. An operating system image cannot be displayed on a terminal screen. An image is usually executed from the main RAM and loaded from one of several input sources. The operating software is organized into routines that handle the tasks associated with different protocols, such as data movement, table and buffer management, routing updates, and user command execution.



Router modes

Whether accessed from the console or by a Telnet session through a TTY port, a router can be placed in several modes. (see Figure) Each mode provides different functions: *user EXEC mode* -- This is a look-only mode in which the user can view some information about the router, but cannot make changes. *Privileged EXEC mode* -- This mode supports the debugging and testing commands, detailed examination of the router, manipulation of configuration files, and access to configuration modes.

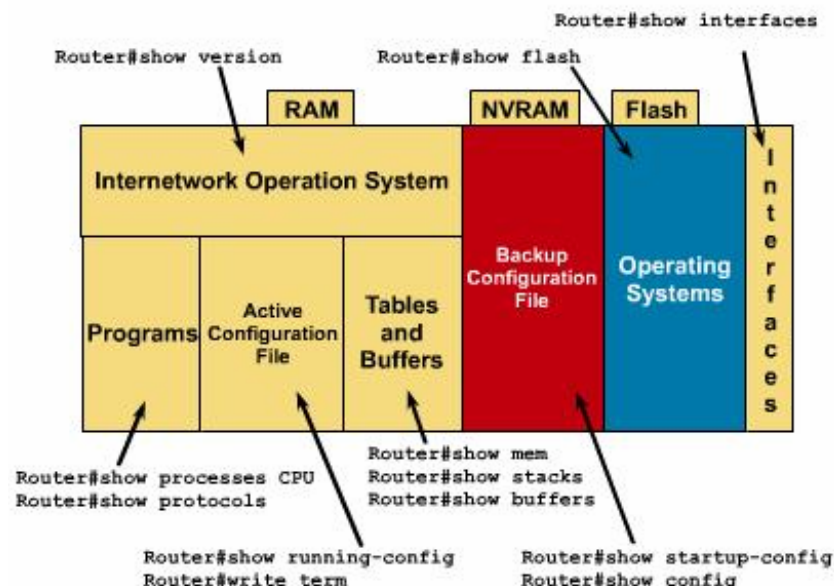
Setup mode -- This mode presents an interactive prompted dialog at the console that helps the new user create a first-time basic configuration. *Global configuration mode* -- This mode implements powerful one-line commands that perform simple configuration tasks. *Other configuration modes* -- These modes provide more detailed multiple-line configurations. *RXBOOT mode* -- This is the maintenance mode that you can use, among other things, to recover from lost passwords.



Examining router status by using router status commands

In this section, you will learn basic commands that you can issue to determine the current status of a router. These commands help you obtain vital information you need when monitoring and troubleshooting router operations. It is important to be able to monitor the health and state of your router at any given time. As shown in the Figure, Cisco routers have a series of commands that allow you to determine whether the router is functionally correct or where problems have occurred.

Router status commands and their descriptions are shown below. **show version** -- displays the configuration of the system hardware, the software version, the names and sources of configuration files, and the boot image
show processes -- displays information about the active processes
show protocols -- displays the configured protocols; shows the status of all configured Layer 3 protocols
show memory -- shows statistics about the router's memory, including memory free pool statistics
show stacks -- monitors the stack use of processes and interrupt routines and displays the reason for the last system reboot
show buffers -- provides statistics for the buffer pools on the router
show flash -- shows information about the Flash memory device
show running-config (write term on Cisco IOS Release 10.3 or earlier) -- displays the active configuration file
show startup-config (show config on Cisco IOS Release 10.3 or earlier) -- displays the backup configuration file
show interfaces -- displays statistics for all interfaces configured on the router



The show running-config and show startup-config commands

Among the most used Cisco IOS software EXEC commands are `show running-config` and `show startup-config`. They allow an administrator to see the current running configuration on the router or the startup configuration commands that the router will use on the next restart. (**Note:** The commands, `write term` and `show config`, used with Cisco IOS Release 10.3 and earlier, have been replaced with new commands.

The commands that have been replaced continue to perform their normal functions in the current release but are no longer documented. Support for these commands will cease in a future release.) You can recognize an active configuration file by the words `current configuration` at the top. You can recognize a backup configuration file when you see a message at the top that tells you how much non-volatile memory you have used.

```
Command Output
Router#show running-config
Building configuration...

Current configuration:
!
version 11.1
!
-- More --
```


The show interfaces, show version, and show protocols commands

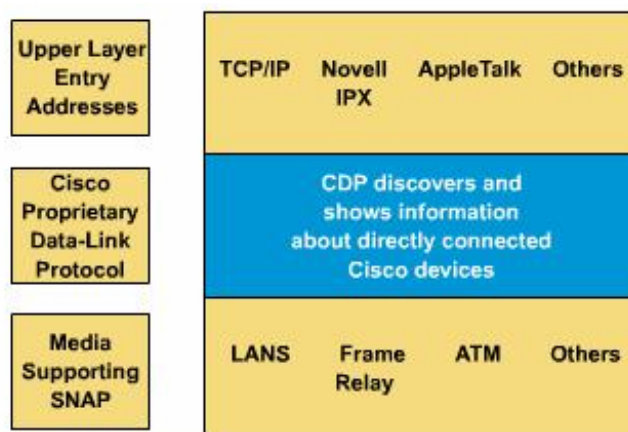
The **show interfaces** command displays configurable parameters and real-time statistics related to all interfaces configured on the router (see Figure). The **show version** command displays information about the Cisco IOS software version that is currently running on the router (see Figure). You use the **show protocols** command to display the protocols configured on the router. This command shows the global and interface-specific status of any configured Level 3 protocols (for example, IP, Decnet, IPX, and AppleTalk).

```
Command Output
Router#show interfaces
Serial0 is up, line protocol is up
Hardware is MK5025
Internet address is 183.8.64.129, subnet mask is 255.255.255.128
MTU 1500 bytes, BW 56 kbit, DLY 20000 usec, rely 255/255, load 9/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
Last input 0:00:00, output 0:00:01, output hang never
Last clearing of show interfaces counters never
Output queue 0/40, 0 drops, input queue 0/75, 0 drops
Five minute input rate 1000 bits/sec, 0 packets/sec
331885 packets input, 62400237 bytes, no buffer
Received 230457 broadcasts, 0 runts, 0 giants
3 input errors, 3 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
403591 packets output, 66717279 bytes, 0 underruns
0 output errors, 0 collisions, 8 interface resets, 0 restarts
45 carrier transitions
```

Gaining access to other routers by using Cisco Discovery Protocol (CDP)

Cisco Discovery Protocol (CDP) provides a single proprietary command that enables network administrators to access a summary of what the configurations look like on other directly-connected routers. CDP runs over a data link layer that connects lower physical media and upper network layer protocols, as shown in the Figure. Because it operates at this level, CDP devices that support different network layer protocols can learn about each other.

(Remember that a data link address is the same as a MAC address.) When a Cisco device that is running Cisco IOS (Release 10.3 or later) boots up, CDP starts up automatically, which then allows the device to detect neighbouring Cisco devices that are also running CDP. Such devices extend beyond those using TCP/IP, and include directly-connected Cisco devices, regardless of which Layer 3 and 4 protocol suite they run.

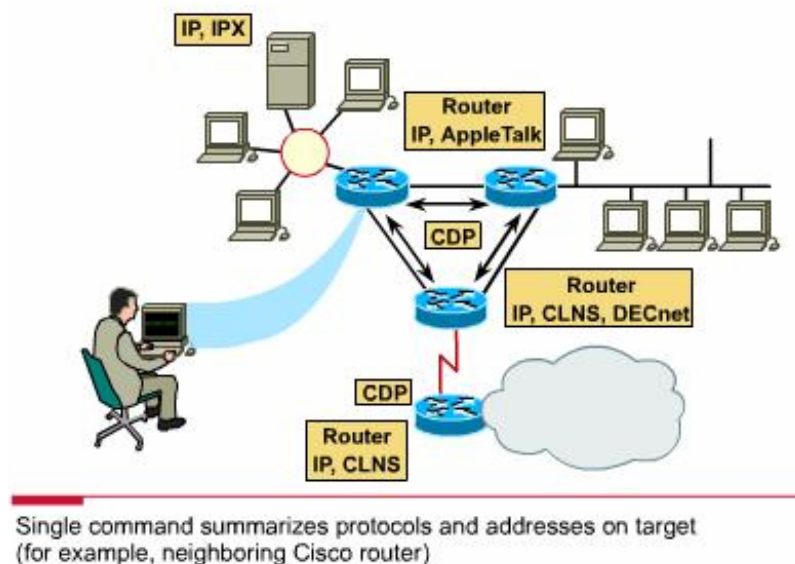


Media and Protocol Interactions

Showing CDP neighbour entries

The primary use of CDP is to discover platforms and protocols on your neighbouring devices. Use the **show cdp neighbours** command to display the CDP updates on the local router. The Figure displays an example of how CDP delivers its collection of information to a network administrator. Each router that is running CDP exchanges information regarding any protocol entries with its neighbours. The administrator can display the results of this CDP information exchange on a console that is connected to a router configured to run CDP on its interfaces.

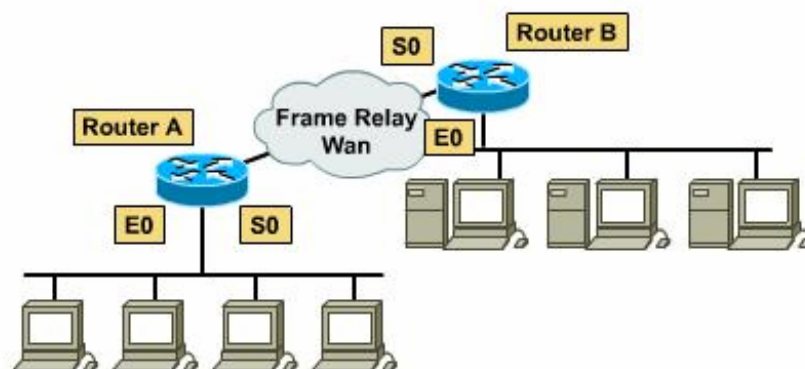
The network administrator uses a **show** command to display information about the networks directly connected to the router. CDP provides information about each CDP neighbour device. Values include the following: *device identifiers* -- e.g. the router's configured host name and domain name (if any) *address list* - at least one address for SNMP, up to one address for each supported protocol *port identifier* -- e.g. Ethernet 0, Ethernet 1, and Serial 0 *capabilities list* -- e.g. if the device acts as a source route bridge as well as a router *version* -- information such as that provided by the local command **show version** *platform* -- the device's hardware platform, e.g. Cisco 7000 Notice that the lowest router in the figure is not directly connected to the administrator's console router. To obtain CDP information about this device, the administrator would need to Telnet to a router that is directly connected to this target



A CDP configuration example

CDP begins automatically upon a device's system startup. The CDP function normally starts by default when a Cisco product boots up with Cisco IOS Release 10.3 or later. Only directly connected neighbours exchange CDP frames. A router caches any information it receives from its CDP neighbours. If a subsequent CDP frame indicates that any of the information about a neighbour has changed, the router discards the older information and replaces it with the new information.

Use the command `show cdp interface`, as shown in Figure , to display the values of the CDP timers, the interface status, and the encapsulation used by CDP for its advertisement and discovery frame transmission. Default values for timers set the frequency for CDP updates and for aging CDP entries. These timers are set automatically at 60 seconds and 180 seconds, respectively. If the device receives a more recent update, or if this hold-time value expires, the device must discard the CDP entry.



Showing CDP entries for a device and CDP neighbours

CDP was designed and implemented as a very simple, low-overhead protocol. A CDP frame can be small yet retrieve a lot of useful information about neighbouring routers. You use the command `show cdp entry {device name}` to display a single cached CDP entry. Notice that the output from this command includes all the Layer 3 addresses present in the neighbour router, Router B. An administrator can view the IP addresses of the targeted CDP neighbour (Router B) with the single command entry on Router A. The hold-time value indicates the amount of elapsed time since the CDP frame arrived with this information.

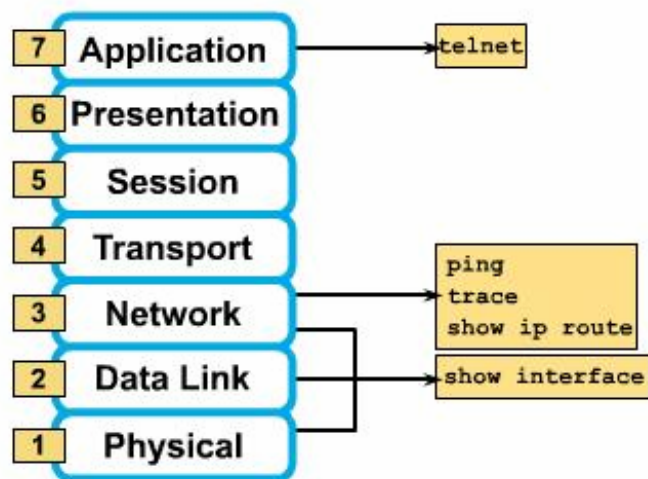
The command includes abbreviated version information about Router B. You use the command `show cdp neighbours`, as shown in Figure , to display the CDP updates received on the local router. Notice that for each local port, the display shows the following: neighbour device ID local port type and number decremental hold-time value, in seconds neighbour device capability code neighbour hardware platform neighbour remote port type and number To display this information as well as information like that from `show cdp entry`, you use the optional `show cdp neighbours detail`.

```
Command Output
RouterA#show cdp entry routerB
-----
Device ID: routerB
Entry address(es):
  IP address: 198.92.68.18
Platform: 2501. Capabilities: Router
Interface: Ethernet0, Port ID (outgoing port): Ethernet0
Holdtime: 155 sec

Version
IOS (tm) GS Software (GS3), 11.2(13337)[asastry 161]
Copyright (c) 1986-1996 by Cisco Systems, Inc.
Compiled Tue 14-May-96 1:04
```

Testing process that uses the OSI model

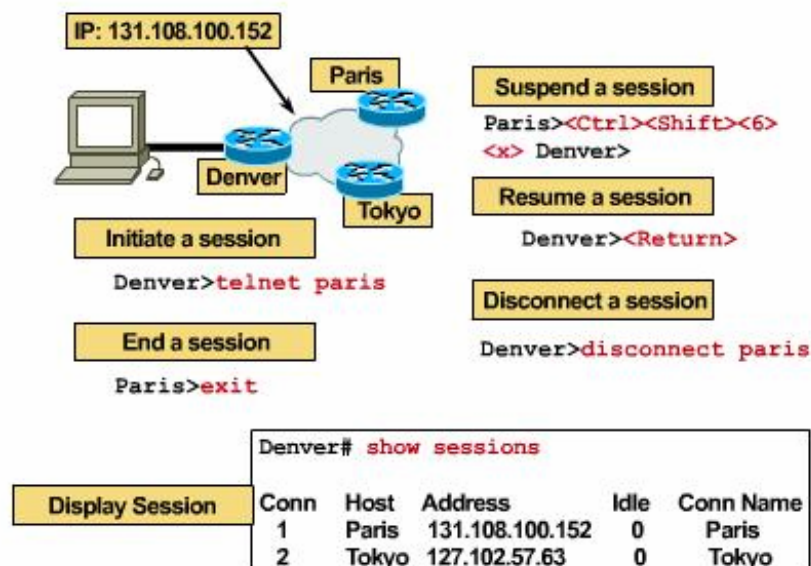
The most common problems that occur on IP networks result from errors in the addressing scheme. It is important to test your address configuration before continuing with further configuration steps. Basic testing of a network should proceed in sequence from one OSI reference model layer to the next. Each test presented in this section focuses on network operations at a specific layer of the OSI model. As shown in the Figure, **telnet**, **ping**, **trace**, **show ip route**, **show interfaces** and **debug** are commands that allow you to test your network.



Testing the application layer by using telnet

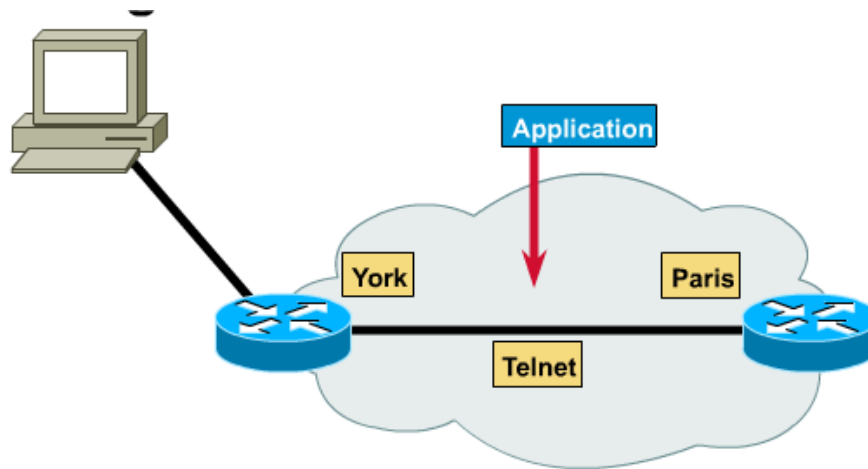
Another way to learn about a remote router is to connect to it. Telnet, a virtual terminal protocol that is part of the TCP/IP protocol suite, allows connections to be made to hosts. You can set a connection between a router and a connected device. Telnet allows you to verify the application-layer software between source and destination stations. This is the most complete test mechanism available. A router can have up to five simultaneous incoming Telnet sessions. Let's begin testing by initially focusing on upper-layer applications. As shown in Figure , the `telnet` command provides a virtual terminal so administrators can use Telnet operations to connect with other routers running TCP/IP. With Cisco's implementation of TCP/IP, you do not need to enter the command `connect` or `telnet` to establish a Telnet connection.

If you prefer, you can just enter the learned host name. To end a Telnet session, use the EXEC commands `exit` or `logout`. The following list shows alternative commands for the operations listed in the figure: Initiate a session from Denver: `Denver> connect Paris` `Denver> Paris` `Denver> 131.108.100.152` Resume a session (enter session number or name): `Denver> 1` `Paris>` End a session: `Paris> exit` As you have already learned, the Telnet application provides a virtual terminal so that you can connect to other hosts that are running TCP/IP. You can use Telnet to perform a test to determine whether or not you can access a remote router.



As is shown in Figure , if you can successfully use Telnet to connect the York router to the Paris router, then you have performed a basic test of the network connection. If you can remotely access another router through Telnet, then you know that at least one TCP/IP application can reach the remote router. A successful Telnet connection indicates that the upper-layer application (and the services of lower layers, as well) function properly. If we can Telnet to one router but not to another router, it is likely that the Telnet failure is caused by specific addressing, naming, or access permission problems. These problems

can exist on your router or on the router that failed as a Telnet target. The next step is to try **ping**, which is covered in this section. This command lets you test end-to-end at the network layer.

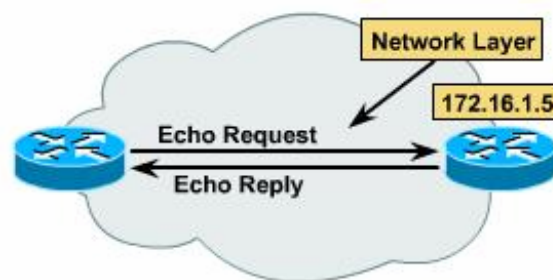


◆ Can the remote router be accessed?

Testing the network layer using the ping command

As an aid to diagnosing basic network connectivity, many network protocols support an echo protocol. Echo protocols are used to test whether protocol packets are being routed. The `ping` command sends a packet to the destination host and then waits for a reply packet from that host. Results from this echo protocol can help evaluate the path-to-host reliability, delays over the path, and whether the host can be reached or is functioning.

In the Figure, the `ping` target 172.16.1.5 responded successfully to all five datagram's sent. The exclamation points (!) indicate each successful echo. If you receive one or more periods (.) instead of exclamations on your display, the application on your router timed out waiting for a given packet echo from the `ping` target. You can use the `ping user EXEC` command to diagnose basic network connectivity. The `ping` uses the ICMP (Internet Control Message Protocol).

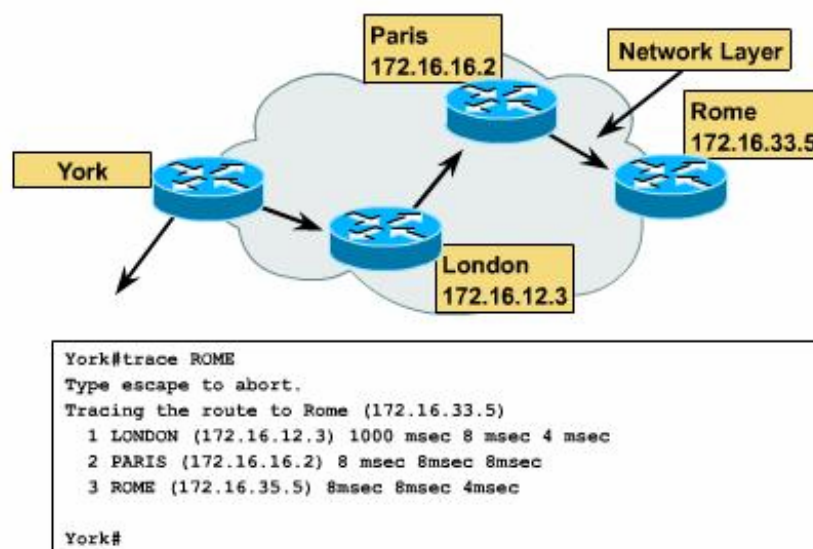


```
Router>ping 172.16.1.5
Type escape sequence to abort.
Sending 5, 100 byte ICMP Echos to 172.16.1.5,
timeout is 2 seconds:
!!!!
Success rate is 100 percent,
round-trip min/avg/max = 1/3/4 ms
Router>
```

Testing the network layer with the trace command

The **trace** command is the ideal tool for finding where data is being sent in your network. The **trace** command is similar to the **ping** command, except that instead of testing end-to-end connectivity, **trace** tests each step along the way. This operation can be performed at either the user or privileged EXEC levels. The **trace** command takes advantage of the error messages generated by routers when a packet exceeds its Time to Live (TTL) value. The **trace** command sends several packets and displays the round-trip time for each.

The benefit of the **trace** command is that it tells which router in the path was the last one to be reached. This is called fault isolation. In this example, we are tracing the path from York to Rome. Along the way the path must go through London and Paris. If one of these routers had been unreachable, you would have seen three asterisks (*) instead of the name of the router. The **trace** command would continue attempting to reach the next step until you escaped using the Ctrl-Shift-6 escape sequence.



Testing network layer with the show ip route command

The router offers some powerful tools at this point in the search. You can actually look at the routing table - the directions that the router uses to determine how it will direct traffic across the network. The next basic test also focuses on the network layer. Use the **show ip route** command to determine whether a routing table entry exists for the target network. The highlight in the graphic shows that Rome (131.108.33.0) is reachable by Paris (131.108.16.2) via the Ethernet1 interface.

```
Command Output
Paris#show ip route

Codes: I - IGRP derived, R - RIP derived, O - OSPF derived
       C - connected, S - static, E - EGP derived, B - BGP derived
       i - IS-IS derived, D - EIGRP derived
       * - candidate default route, IA - OSPF inter area route
       E1 - OSPF external type 1 route, E2 - OSPF external type 2 route
       L1 - IS-IS level-1 route, L2 - IS-IS level -2 route
       EX - EIGRP external route

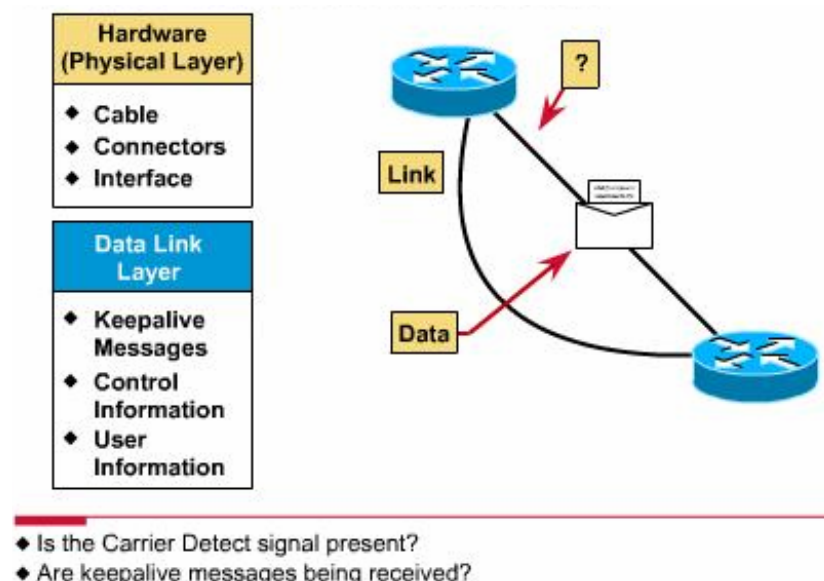
Gateway of last resort is not set

I   144.253.0.0 [100/1300] via 133.3.32.2, 0:00:22 Ethernet1
    131.108.0.0 is subnetted (mask is 255.255.255.0), 3 subnets
I   131.108.33.0 [100/180771] via 131.108.16.2, 0:01:29, Ethernet1
C   131.108.12.0 is directly connected, Ethernet1
C   101.108.16.0 is directly connected, Ethernet0
I   219.100.103.0 [100/1200] via 133.3.32.2, 0:00:22, Ethernet1
```

Using the show interfaces serial command to test the physical and data link layers

As shown in Figure , the interface has two pieces, physical (hardware) and logical (software): The hardware -- such as cables, connectors, and interfaces -- must make the actual connection between the devices. The software is the messages -- such as keep alive messages, control information, and user information -- that are passed between adjacent devices. This information is data being passed between two connected router interfaces.

When you test the physical and data link, you ask these questions: Is there a Carrier Detect signal? Is the physical link between devices good? Are the keep alive messages being received? Can data packets be sent across the physical link? One of the most important elements of the **show interfaces serial** command output is display of the line and data link protocol status. Figure indicates the key summary line to check the status meanings. The line status in this example is triggered by a Carrier Detect signal, and refers to the physical layer status. However, the line protocol, triggered by keep alive frames, refers to the data link framing.



The show interfaces and clear counters commands

The router tracks statistics that provide information about the interface. You use the **show interfaces** command to display the statistics as shown in the figure. The statistics reflect router operation since the last time the counters were cleared, as shown in the top highlighted line in the graphic. This graphic shows that it was two weeks and four days earlier. The bottom set of highlights shows the critical counters. Use the **clear counters** command to reset the counters to 0. By starting from 0, you get better picture of the current status of the network.

```
Command Output
Router#show interfaces serial 1

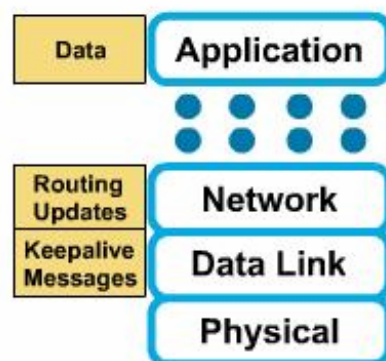
Serial1 is up, line protocol is up

  Hardware is cxBus Serial
  Description: 56Kb Line San Jose - MP
  Internet address is 150.136.190.203, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 56 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 0:00:07, output 0:00:00, output hang never
  Last clearing of show interfaces counters 2w4d
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
    16263 packets input, 1347238 bytes, no buffer
    Received 13983 broadcasts, 0 runts, 0 giants
    2 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 2 abort
    0 input packets with dribble condition detected
    22146 packets output, 2383680 bytes, 0 underruns
    0 output errors, 0 collisions, 2 interface resets, 0 restarts
    1 carrier transitions
```

Checking real-time traffic with debug

The router includes hardware and software to aid it in tracking down problems, on it, or on other hosts in the network. The `debug privileged EXEC` command starts the console display of the network events specified in the command parameter. Use the `terminal monitor` command to forward `debug` output to your Telnet session terminal. In this example, data link broadcasts received by the router are displayed.

Use the `undebug all` command (or `no debug all`) to turn debugging off when you no longer need it. Debugging is really intended for solving problems. (Note: **Be very careful with this tool on a live network**. Substantial debugging on a busy network will slow down the network significantly. **Do not leave debugging turned on**; use it to diagnose a problem, and then turn it off.) By default, the router sends system error messages and output from the `debug` EXEC command to the console terminal. Messages can be redirected to a UNIX host or to an internal buffer. The `terminal monitor` command gives you the capability to redirect these messages to a terminal.



```
Router#debug broadcast
broadcast debugging is on

Ethernet0: Broadcast ARPA, src 0000.00c0.5fa4, dat ffff.ffff.ffff
type 0x0800, data 4500002800000000FF11EA7B, len 60
Serial3: Broadcast HDLC, size 64, type 0x80, fkags 0x8F00
```

Summary

In this chapter, you learned that: The router is made up of configurable components and has modes for examining, maintaining, and changing the components. **show** commands are used for examination. You use CDP to show entries about neighbours. You can gain access to other routers by using Telnet. You should test network connectivity layer by layer. Testing commands include **telnet**, **ping**, **trace**, and **debug**.