

Applying Cisco Troubleshooting Tools

This chapter introduces some powerful troubleshooting tools that are built into the Cisco IOS. As with other tools, it is important that you identify when to use them and what information they reveal. Because some of these tools have an impact on the way routers operate and may impede the routers' utmost performance, it is essential to use them with care. To better understand the output of these commands, and to recognize what router internal operations they affect, this chapter discusses router internal components and operations. As each tool/command is introduced, its usefulness is described and tips are given on how to use it effectively.

The Cisco IOS troubleshooting commands help you gather valuable information about the state of the network and its devices. The gathered facts help eliminate some of the possibilities, at the same time strengthening the likelihood of the hypotheses that you may have formed. Again, due to the impact that certain troubleshooting commands (debug, for example) have on some of the router's internal operations—and ultimately on the router's performance—these commands have to be used selectively, properly, and temporarily.

“Do I Know This Already?” Quiz

If you wish to evaluate your knowledge of the contents of this chapter before you get started, answer the following questions. The answers are provided in Appendix A, “Answers to Quiz Questions.” If you are having difficulty providing correct answers, you should thoroughly review the entire chapter. If all or most of your answers are correct, you might want to skim this chapter for only those subjects you need to review. You can also use the “Foundation Summary” section to quickly review topics. Once you have completed the chapter, you should reevaluate yourself with the questions in the “Q&A” section at the end. Finally, use the companion CD-ROM to evaluate your knowledge of the topics and see if you need a review.

- 1 Briefly explain why Cisco IOS troubleshooting commands/tools need proper handling.

2 What does proper handling of troubleshooting tools entail?

3 Provide a generic explanation for route caching (or fast switching) and the motivation behind it.

4 Which of the route caching methods are not enabled by default? And from which configuration mode (prompt level) can they be enabled?

5 With regard to speed and switching optimization, how did Cisco Systems improve the Cisco 7500 routers (in comparison to the 7000 series)?

6 Briefly describe the advantages of Netflow switching. Also specify whether there should be any precautions with respect to enabling Netflow switching.

7 Provide at least three examples of operations or packet types that are process switched.

- 8 Before you activate Debug, what are some of its characteristics that you should consider?

- 9 Before you enable debugging on a router, you are encouraged to check the router’s CPU utilization. What is the command that allows you to do that? If the utilization is above 50%, are you encouraged to debug packets or to debug events?

- 10 What is the default setting (for example, enabled/disabled, default destination) for message logging?

- 11 What information does the output of the **show logging** Cisco IOS exec command display?

- 12 The outputs of the **show memory** and the **show processes [cpu]** commands will most likely be asked for in which situation (loss of functionality, crash, or performance degradation)?

- 13 If the output of **show buffer** command displays a large number of misses, increasing the value of which one of the buffer management parameters (Permanent, Min-Free, Max-Free, Initial) will most likely remedy the situation?

- 14 The **show processes** command's output provides two numbers separated by a slash (for example, 4%/4%) for the CPU utilization over the last five seconds. How are those numbers interpreted?

- 15 Which command causes the router to attempt to produce a core dump when it crashes?

Foundation Topics

System Impact of Cisco Troubleshooting Tools

After completing the problem definition step, your next step (following the systematic troubleshooting process) is to start gathering detailed facts about the behavior of the devices and protocols of the production network. This task usually entails using several IOS troubleshooting tools and commands. Despite their importance in terms of the valuable information they provide, these tools inevitably utilize some processing cycles and memory of the router. Furthermore, they may disable or at least have a negative effect on some of router's internal (optimized) operations (for example, fast switching).

If you are responsible for fixing a production network's problems, you need to be familiar with troubleshooting tools. In other words, you have to know which tools you need, how to interpret each tool's output, and, very importantly, how to use each tool properly. Proper usage of a tool means that you should use it with appropriate focus and selectiveness, yet to an extent that you will gather the desired information. You must stop using these tools immediately after you attain your objective, thus you should limit the period of time during which these tools are used.

Several of the Cisco IOS **show** commands display information about the status of the router, its interfaces, and the rate of utilization of router resources. **debug** is a powerful command for finding out which packets are generated, received, and forwarded by a router. Several parameters of the **debug** command help focus the output on what you are interested in seeing, and hence give you a great insight on the current events and how the router handles them. But **debug** lowers a router's performance substantially, and that is why you need to give this command special attention.

Cisco Routers' Routing Processes and Switching Processes

Routing and switching processes are two of the most essential tasks performed by routers. Some people in the internetworking field have a little difficulty with the idea that routers perform switching, but of course everybody is comfortable with the fact that routers perform routing. Nonetheless, as you will see, the distinctions between these operations will be quite useful in optimizing the router's performance. In the following paragraphs the concepts of routing and switching are defined. A discussion then follows about the benefit of distinguishing among these tasks while troubleshooting and about methods for examining the distinct processes and components of the routing and switching process.

Switching is commonly defined as the process that takes charge of moving data units (frames or packets) through the anatomy of internetworking devices. From the time data

units arrive at an interface until they leave the router, several issues need to be constantly addressed. Where is the data unit stored? What type of information should accompany the data unit? Where does the data unit go next? How is the next destination determined? And which statistics need to be collected about this data unit? As you can imagine, the mere task of moving a data unit (a packet, in the context of routers) from one place to the next is one of the simplest and least resource-consuming internetworking tasks.

Routing can be simply defined as the operation that attempts to select an output interface and perhaps a next hop for a packet based on the packet's destination address. Different routing processes perform the task of routing different protocols' packets. For example, the IP routing process, which is enabled by default, handles routing of IP packets. The routing process makes its routing decision by consulting its routing table, which it builds and maintains dynamically. The sources of information of a routing process for building its routing table are

- 1 The network segments that the router is actively connected to.
- 2 The usable static routes available in the router configuration.
- 3 The dynamic routing entries that the routing protocols offer.
- 4 The routing policies or restrictions that are imposed.
- 5 The usable default routes available.

Imagine that a router (call it R1) is receiving a bunch of packets from its ethernet 0 interface and all of these packets have the same destination address. When the router receives the first layer 3 packet (call it P1), the data link header (layer 2 frame) is discarded and the packet stored in the E0 interface's buffer. Next, the packet will be moved through R1's internal bus into main memory, then stored in a packet buffer where it will wait its turn to be examined by the routing process. It is important to note that the packet is accompanied by additional information such as the interface the packet entered the router from. When it is P1's turn, the routing process uses the destination address of the packet and the routing table to select an interface for the packet to leave the router from. Finally, a frame appropriate to that interface must be created. This entails building a frame header, usually composed of layer 2 addressing and perhaps other information such as protocol type.

Notice that during this process the packet is moved by the switching process from one place to another (from the interface internal buffer to the main memory, for instance) several times. Hence, the first packet (P1) must be handled by both the routing and the switching processes. Performing both routing and switching tasks on a packet is called process switching. Now think about the second packet (call it P2) that has the same destination as P1. Does P2 have to be routed and switched (i.e., process switched) as well? The answer is usually negative.

To enhance efficiency and speed, the experience gained from the effort spent (through the routing task) on the first packet (P1, in our example) can be reused. P1's destination address and the outgoing interface (selected by the routing process) can be stored in a place referred

to as a switching cache, and can be reused for quick processing of the subsequent packets (P2, P3, and so on, which have the same destination address as P1). Inside the switching cache, the router needs to store the first packet's destination address along with the output interface number. The information stored in the switching cache is used to quickly discover the output interface for the following packets (destined for the same network as P1), without having to perform the time-consuming task of routing on each of them.

You must remember that the router also caches information such as MAC addresses (in the ARP cache) so that it does not have to generate an ARP request for every packet. Hence, when dealing with a number of packets (with identical destination addresses) arriving at the router one after the other, the router must perform both the routing and the switching task on the first packet.

To process the following packets, using the information kept in the switching cache, the router can skip the routing task. This is commonly referred to as route caching (and loosely speaking, also called fast switching). Please keep in mind that there are more operations and tasks involved in the processing of each packet, most of which have to be performed for each packet individually. For example, security tasks (checking access lists) and accounting/queuing tasks still need to be performed for each packet (except in case of Netflow switching, which is presented later in this chapter).

Since the routing task is more resource consuming, is more complex, and introduces a longer latency, skipping this operation on all the packets except the first (all with the same destination address) is very advantageous and efficient. When a packet (such as P2) is not process switched (not subjected to routing), the packet is said to be fast switched. The switch cache, where the network layer destination address and the corresponding selected output interface (based on processing of the first packet) are stored, can be on one or more of the router's components. Where that information is stored depends on the type of router and its components, and based on that, the cache is referred to by a special term.

Switching in 7000, 7500, 4000, 3000, and 2500 Series Routers

This section discusses the switching options and their initialization on different router models. As you will notice, most of the methods are performing the same task, but some are faster than others due to highly specialized internal techniques and microchips.

It is important to know that all the switching features, except fast switching, need to be turned on manually. Enabling a specific type of switching is an interface configuration task and must be done for each protocol individually. Of course the type of switching that is available is dependent upon the router.

Switching in 7000 Series Routers

The 7000 series routers, similar to other Cisco routers, have a fast switching option that is enabled by default. Fast switching is performed using a Fast Switch Cache in the Route Processor. Two major components that participate in the routing and switching operations are RP (Route Processor) and SSP (Silicon Switch Processor). The early models of the 7000 series had RP and SP (Switch Processor). The SP (in the earlier models) only had an Autonomous Switch Cache. The SSP was introduced later and is equipped with both an Autonomous Switch Cache and a Silicon Switch Cache (see Figure 4-1). Autonomous and silicon switching are not enabled by default. The administrator of a router may enable either or both of these switching options on a per-protocol basis at each interface. The commands for enabling and disabling fast switching, autonomous switching, and silicon switching are:

Fast switching:

```
Router(config-if)# [no] [protocol] route-cache
```

Autonomous switching:

```
Router(config-if)# [no] [protocol] route-cache cbus
```

Silicon switching:

```
Router(config-if)# [no] [protocol] route-cache sse
```

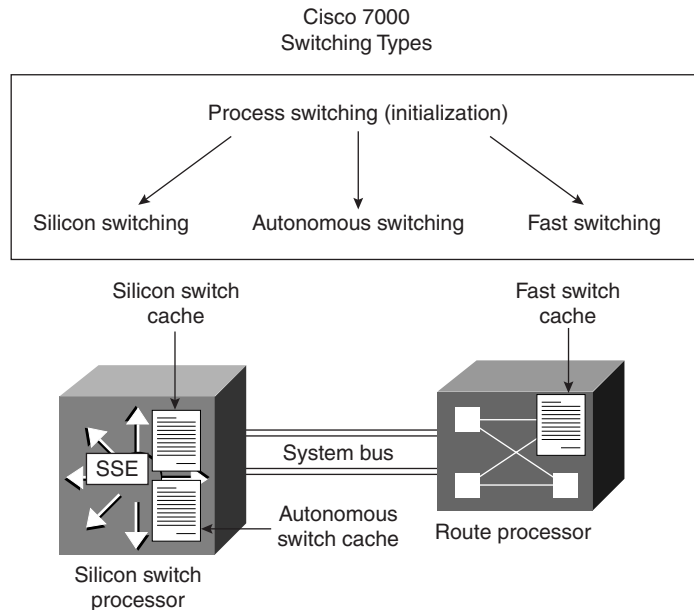
NOTE

You must specify a protocol name, such as **ip** or **ipx**.

Assume that fast switching, autonomous switching, and silicon switching are all enabled on all of the interfaces of a router (call it R1) for all protocols, including IP. Now, for the sake of the example, take a look at the sequence of tasks and operations executed by R1 to process some packets (call them P1, P2, and so on) entering into the router via its ethernet 0 interface. When the frame carrying P1 is accepted by the ethernet 0 interface (E0 of R1), the header (layer 2 frame) is discarded and the encapsulated packet (P1) is kept in the interface buffer.

The E0 interface must wait its turn to get a chance to send P1 through the router's bus (CxBus in the case of 7000 series routers operating at 533Mbps) to the SSP. When P1 arrives at the SSP, it is stored in a buffer, and then its destination address is first checked against the entries in the Silicon Switch Cache (to find a match).

Since P1 is the first packet from the sequence of packets with the same destination (P1, P2, and so on), there won't be a match in the SSP's Silicon Switch Cache for P1. There will be no entries matching P1's destination address in the Autonomous Switch Cache, either.

Figure 4-1 Routing and Switching in 7000 Routers

Next, P1's header is moved through a system bus (operating at only 153 Mbps) to the Route Processor for a quick search and match against the entries in the Fast Switch Cache (residing in the RP). Again, there won't be a match. Now the entire packet (*not* only the packet header) must be moved from the SSP to the RP, stored in a buffer, and in turn be looked after by the routing process. Hence, the entire P1 packet waits its turn in a buffer (within RP's memory) until the routing process gets a chance to administer it.

The routing process uses the destination address of P1 and the content of its routing table to select the best (if any) interface to send P1 out from. Say TokenRing 0 is selected. Of course an appropriate frame must be created for P1 to be encapsulated into, before it is sent to the TokenRing 0 interface (To0) and ultimately out to the wire. Once these tasks (perhaps including ARP) are completed, the router sends the P1 packet (encapsulated in a Token Ring frame) from the RP to the SSP and then to the T0 interface.

The knowledge gained from processing the P1 packet is saved in the router's switching caches, to be used for processing P2, P3, and so on. Assuming that all switching options are enabled, P1's destination address—the network component of it, to be accurate—along with the destination interface (number), are stored in the Fast Switch Cache (in RP) and the Autonomous and Silicon Switch Caches (in SSP). When P2 arrives from an interface into the SSP and the Silicon Switch Cache is checked, a match is found. The matching entry from the Silicon Switch Cache quickly identifies which interface P2 must exit from (To0 in

this case). Recall that the router keeps information such as the recently used MAC addresses in its ARP cache, as well. As you can imagine, processing of P2 (and P3 and so on) will be much faster than the processing of P1. These packets are not process-switched: they don't interrupt the RP, they don't have to travel through the slow system bus (153 Mbps) to go to the RP, and they don't have to wait in a buffer in RP for their turn to be looked after by the routing process.

In our example, if silicon switching was not enabled on To0, but autonomous switching was, then the match would be found in the Autonomous Switch Cache. If both silicon and autonomous switching were disabled for To0 (notice that it is the output interface that matters), then the match would be found only in the Fast Switch Cache in the RP. Recall that if a match is not found in either of SSP's caches, the header of the packet is moved from the SSP (through the slow system bus) to the RP to do a quick check on the Fast Switch Cache. In a Cisco 7000 router, silicon switching is considered to be roughly 35% faster than autonomous switching, and autonomous switching is roughly 6.6 times faster than fast switching. Finally, fast switching in a Cisco 7000 router is about 12 times faster than process switching. All of the mentioned estimations are based on tests performed using packets of 64 byte in size.

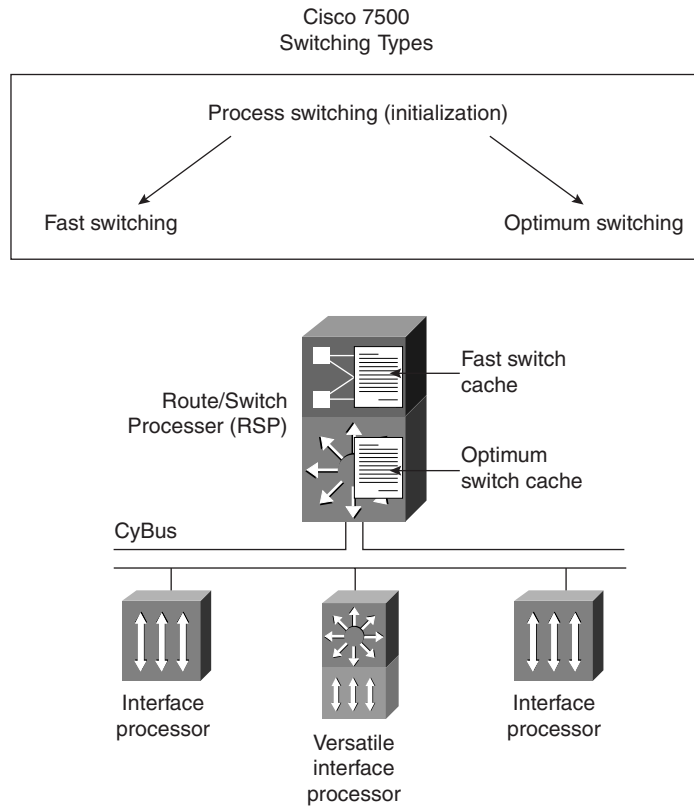
Switching in 7500 Series Routers

To enhance the performance and efficiency of its high-end routers, Cisco Systems made some major improvements in the 7500 series routers:

- The internal bus of the 7500 series router (CyBus) operates at 1 Gbps, which is about twice as fast as the 7000 router's CxBus.
- Instead of having two separate components for routing and switching (SSP and RP) as in 7000 routers, the 7500 router is equipped with one component called the Route/Switch Processor (which eliminates the slow 153 Mbps system bus previously needed to connect the RP and SSP).
- The switch cache of the 7500 series router, called Optimum Switch Cache, is faster than the Silicon Switch Cache of the 7000 router (see Figure 4-2).

As with the 7000 router and other routers, fast switching in the 7500 router is enabled by default and it is accomplished using the Fast Switch Cache (located in the Route Switch Processor [RSP]). However, the second type of switching performed by the 7500 router, called optimum switching, is the winning card for the 7500 router, as it is even faster than the 7000 router's Silicon Switch Cache. The Optimum Switch Cache is also located on the RSP. Optimum switching must be manually enabled on each interface for each protocol, except for IP. In other words, this feature is enabled by default for IP on all supported interfaces (Ethernet, FDDI, and Serial with HDLC encapsulation). Use the following interface configuration command to enable/disable optimum switching for a protocol on an interface:

```
Router(config-if)# [no] [protocol] route-cache optimum
```

Figure 4-2 Routing and Switching in 7500 Routers

After a packet is extracted from its frame by the ingress interface, it is stored in the interface buffer and waits its turn to be sent through the CyBus to the RSP. In the RSP, the packet is stored in a buffer, and then its destination address is compared to the entries in the Optimum Switch Cache. If a match is found, the output interface is known, and an appropriate frame is swiftly created (using the information stored in the ARP cache, for example) and the packet is sent to the output interface through the CyBus. On the other hand, if a match is not found in the Optimum Switch Cache, the Fast Switch Cache is checked. If no match is found in the Fast Switch Cache either, the packet must wait in the buffer for its turn to be looked after by the appropriate routing process. The destination address (destination network, to be accurate) of this packet along with the output interface that the routing process selects for it are stored in the Fast Switch Cache and the Optimum Switch Cache (if enabled). This information shall be used for swift forwarding of the packets destined for the same network as the first packet.

The 7500 routers also feature Versatile Interface Processors (VIPs) that have a RISC processor and memory locally (on the blade). The 7500 routers can be configured to distribute routing information to be stored on the VIP. The VIP can then use the cached information to switch the packets on its own without having to send packets over to the RSP. This method of switching is called distributed switching. Distributed switching can make the processing of packets more than three times faster than silicon switching. Use the following interface configuration command to enable/disable distributed switching for a protocol on a VIP card:

```
Router(config-if)# [no] [protocol] route-cache distributed
```

Netflow Switching

Netflow switching was introduced with Cisco IOS version 11.1(2) for the Cisco 7000, 7200, and 7500 routers with an RSP (Cisco offers the RSP7000 card for the 7000 routers). Netflow identifies a flow based on the source and destination IP address, source and destination port, protocol type (number), type of service (TOS), and input interface. The other switching types keep network layer destination address and output interface pairs in the cache. If a packet's destination address matches an entry in the cache, it is sent out of the destination interface specified by the cache, regardless of which interface the packet has entered the router from and what conversation it belongs to.

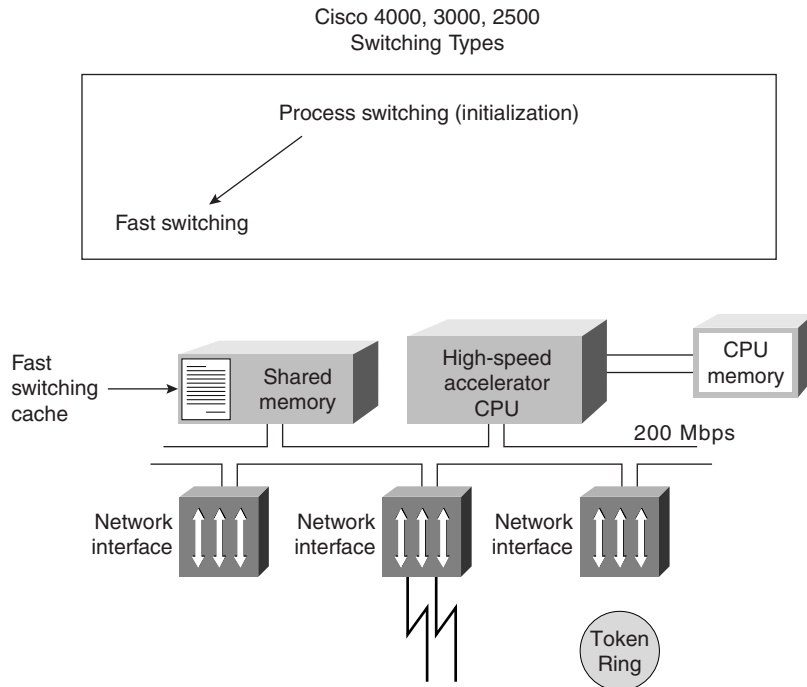
With other switching methods, even though the routing process is not performed for those packets with destination addresses matching the cache entry, other tasks (security and accounting, for example) are still performed on each packet. Netflow switching, on the other hand, caches security information and accounting information as well as routing information for each flow. As a result, once a network flow is identified and the first packet (of this flow) is processed, access list checks for subsequent packets belonging to the flow are bypassed and packet switching and statistics capture are performed in tandem.

Netflow also allows for exporting captured data to management utilities. Netflow switching, especially with the export option, can be quite resource consuming, so caution must be exercised when enabling this feature in production network routers. On 7000 and 7500 routers with RSP, Netflow switching can also be performed on a distributed basis on individual VIPs. Netflow switching can be disabled/enabled on a supported interface with the following interface configuration command:

```
router(config-if)# [no] [protocol] route-cache flow
```

Switching in 4000, 3000, and 2500 Series Routers

On the 4000, 3000, and 2500 series routers, unlike the high-end routers discussed previously, the options are process switching and fast switching only (see Figure 4-3). Fast switching, a term often used when referring to route caching in shared memory, is enabled by default on all interfaces for all supported protocols.

Figure 4-3 Routing and Switching in 4000, 3000, and 2500 Routers

Depending on the operations performed on a particular interface, fast switching might have to be disabled manually or it may get disabled automatically by the IOS while that operation or configuration is in effect (this is IOS dependent). For instance, say you enter the command that applies a priority queue to an interface; depending on the IOS version, fast switching on that interface might be disabled by the IOS automatically, or the IOS might prompt you to disable fast switching before it allows you to apply the priority queue to the interface. You may disable/enable fast switching on an interface for a particular protocol using the following command:

```
router(config-if)# [no] [protocol] route-cache
```

The output of the following command includes information about whether fast switching is enabled/disabled for a particular protocol on a particular interface:

```
router# show [protocol] interface type number
```

Example 4-1 displays a sample output of the **show interface ethernet 0** command. IP fast switching, as the output shows, is enabled on the ethernet 0 interface of the router being examined.

Example 4-1 *A Sample Output of the `show interface ethernet 0` Command*

```
A_StubR#show ip interface ethernet 0
Ethernet0 is up, line protocol is up
 Internet address is 131.1.18.14/22
 Broadcast address is 255.255.255.255
 Address determined by setup command
 MTU is 1500 bytes
 Helper address is not set
 Directed broadcast forwarding is enabled
 Outgoing access list is not set
 Inbound access list is not set
 Proxy ARP is enabled
 Security level is default
 Split horizon is enabled
 ICMP redirects are always sent
 ICMP unreachable are always sent
 ICMP mask replies are never sent
 IP fast switching is enabled
 IP fast switching on the same interface is disabled
 IP multicast fast switching is enabled
 Router Discovery is disabled
 IP output packet accounting is disabled
 IP access violation accounting is disabled
 TCP/IP header compression is disabled
 Probe proxy name replies are disabled
 Gateway Discovery is disabled
 Policy routing is disabled
 Network address translation is disabled
```

To see the statistics on the number of packets that are process switched and fast switched, issue this command:

```
router# show interface stats
```

Process-Switched Packets

With each new release of the Cisco IOS, more tasks may get added to the list of fast-switched tasks. It is important to notice that when one talks about whether a task is process switched as opposed to fast switched, one is really talking about whether the packets associated with this particular task are fast switched or process switched. To clarify this point, assume that you have enabled IPX packet debugging on a particular router's Ethernet interface. Instead of thinking that IPX debugging is a process-switched task, you must understand that only those packets that are subject to this debug—in other words, all IPX packets that exit the router through this interface—are process-switched. Here is a list of some operations (in other words, packet types) that are process switched:

- Data-link layer broadcasts
- Packets subjected to Debug

- Packets delivering error log messages to syslog
- SNMP packets
- Protocol translations—for example, 1. SR/TLB, 2. DEC and LAT to Telnet
- Tunneling—for example, 1. GRE, 2. X25 remote switching
- Custom and priority queuing
- Link compression
- Keepalives

NOTE For an accurate list of tasks (packet types) that are process switched, refer to the Cisco documentation for each device and the IOS version it is running.

Handling the Cisco IOS Debug Troubleshooting Tool

Debug is a troubleshooting command that is available from the privileged exec mode (of Cisco IOS). This command can be used to display information about various router operations and the related traffic generated or received by the router, as well as any error messages. This tool is very useful and informative, but you must be aware of the following facts regarding its use: Debug is treated as a very high priority task. It can consume a significant amount of resources, and the router is forced to process-switch the packets being debugged. Debug must not be used as a monitoring tool—it is meant to be used for a short period of time and as a troubleshooting tool. By using it you discover significant facts about the working and faulty software and/or hardware components. The following is a list of recommendations on proper usage of the **debug** command:

- If you are interested to see a timestamp with each line of the debug output, you must load the timestamp service using this command:

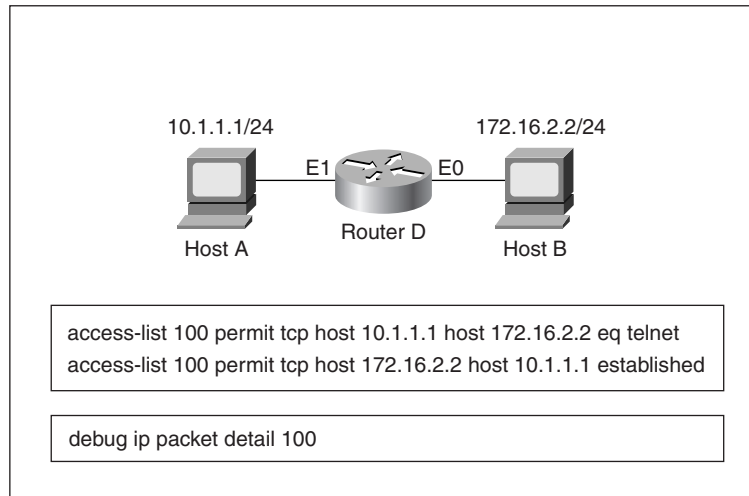
```
router(config)#service timestamps debug [datetime | uptime]
```
- If you plan to see the debug output from within a Telnet session, you need to enter the **terminal monitor** command.
- Usually, the **debug** command is used to diagnose a specific facility, task, or protocol. Sometimes a protocol suite has a specific member (e.g., TCP from among the TCP/IP protocol suite members) that you may want to focus on. When you choose the protocol you want to debug, then you usually have a choice to use the events option or the packets option of the **debug** command for that specific protocol. Event debugging is less resource intensive than packet debugging, but packet debugging produces more information.

- Turning debugging on for everything (using the **debug all** command) is seriously discouraged in production networks. You get a tremendous amount of information, very fast, but it can severely diminish the router's performance or even render it unusable. The **debug all** command is also quite useless since it presents overlapping information that is difficult if not impossible to interpret.
- Before using the **debug** command, see the CPU utilization of your router (using the **show processes cpu** command). If your router's CPU utilization is consistently at 50% or more, you are advised to debug events instead of packets.
- If possible, use the **debug** command during periods when network traffic is not at its peak and fewer critical business applications are active. Cisco routers give the **debug** command higher priority (with respect to CPU cycles) than network traffic.
- Always remember to undo debug as soon as possible. You can use the **no debug {argument}** to turn off a specific debugging type. The **no debug all** or **undebug all** commands can be used to turn off all types of debugging that may be on.
- For troubleshooting, also consider using protocol analyzers to capture and display network traffic. These have little or no impact on your network performance, yet they provide valuable information. I also recommend capturing debug info to a file for offline perusal and training.
- Using an access list with your **debug** command helps you focus the debug output on the task you are troubleshooting. See the next section for more information on this technique.

Using an Access List with Debug

With the **debug ip packet detail** command, you have the option to enter the name or number of an access list. Doing that causes the **debug** command to get focused only on those packets satisfying (permitted by) the access list's statements. Here is an example. Imagine that host A has trouble making a Telnet connection to host B, and you decide to use debug on the router connecting the segments where hosts A and B reside (see Figure 4-4).

Considering the addressing scheme used in Figure 4-4, the access list 100 permits TCP traffic from host A (10.1.1.1) to host B (172.16.2.2) with the Telnet port (23) as the destination. Access list 100 also permits established TCP traffic from host B to host A. Using access list 100 with the **debug ip packet detail** command (as shown in the figure) allows you to see only debug packets that satisfy the access list. This is an effective troubleshooting technique that requires less overhead on your router, while allowing all information on the subject you are troubleshooting to be displayed by the debug facility.

Figure 4-4 *Using Access Lists with the **debug** Command*

Error Message Logging and Limiting the Display of Error Messages

Logging messages are important sources of information for network engineers in charge of troubleshooting. This section covers the following topics:

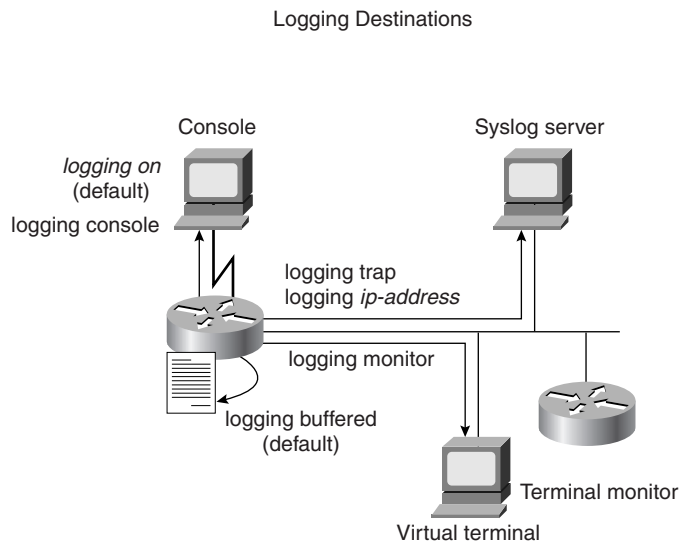
- The options for the logging messages destination
- Which destinations are the default for logging messages
- Which commands enable/disable different destinations
- How usage of different logging destinations compares with regard to the overhead they introduce on the routers
- The eight levels of logging
- Deciphering the logging messages

Message logging is enabled by default and it is directed to the console and the internal buffer. The privileged exec command **logging on** is the default setting, and if you enter **no**

logging on, all logging except console logging will be turned off. The options for logging message destinations are:

- Console
- Internal buffer
- Virtual terminal session (Telnet)
- Syslog server (see Figure 4-5)

Figure 4-5 Error Message Logging Destinations



The following relationship demonstrates how different methods of logging compare in terms of the overhead they produce:

Buffered logging < Syslog < Virtual terminal < Console logging

Table 4-1 displays the commands you must use to configure the destination of logging messages and the desired level of logging for each destination.

Table 4-1 List of Commands for Logging Message Destinations

Command	Usage Explanation
logging console [level]	This command turns console logging on and specifies the level of logging to be directed to the console. (The default setting is Enabled.)
	The no logging console command disables console logging.

Table 4-1 *List of Commands for Logging Message Destinations (Continued)*

Command	Usage Explanation
logging buffered [<i>level</i>]	Use this command to enable sending logging messages to the internal buffer (use no logging buffered to disable it) and specify the level of logging desired to be buffered. This feature is enabled by default.
logging monitor [<i>level</i>]	Use this command to enable sending logging messages to the virtual terminal sessions (use no logging monitor to disable it) and specify the level of logging desired to be directed to the virtual terminal lines. From within a virtual terminal session, typing the command terminal monitor enables the display of logging messages. The command terminal no monitor turns this feature off.
logging trap [<i>level</i>]	This command allows you to enable sending logging messages to syslog servers and specify the level of these messages. The no logging trap command disables this feature. The default level is Informational. (Also see the explanation for the logging [<i>ip-address</i>] command below.)
logging [<i>ip-address</i>]	This command identifies the IP address of the syslog server so that the router can direct its logging messages to this address. If you have a list of syslog servers to which you want to send the logging messages, you may enter this command with each server's appropriate address one by one. Use the no form of this command to take a server off the list.

There are eight levels of logging. If you specify a particular level of logging—for console logging, for example—the messages of that level and of the higher levels (numerically lower) are forwarded to the console. The levels of logging messages are explained in Table 4-2.

Table 4-2 *Logging Messages*

Level	Logging Message
0	Emergencies
1	Alerts
2	Critical
3	Errors
4	Warnings

continues

Table 4-2 *Logging Messages (Continued)*

Level	Logging Message
5	Notifications
6	Informational
7	Debugging

Please note that when you enter one of the commands for specifying the level of logging to be directed to a particular place (console or virtual terminal sessions, for example), you must enter the English phrase for the level of logging and not the numeric value for it. For instance, the command to make virtual terminal lines receive logging messages at the errors level and higher (in other words errors, critical, alerts, emergencies) would be:

```
Router(config)# logging monitor error
```

Now let us discuss the anatomy of the logging messages. Each message is associated with one of the eight levels of logging, which is referred to as the severity of the message. Table 4-3 provides the eight levels of logging messages along with their associated severity levels and short descriptions. Logging messages are composed of a % sign followed by the facility, the severity, the mnemonic, and a text message. For instance, in this message:

```
%TR-3-WIREFAULT:Unit[0],wirefault:check the lobe cable MAU connection
```

the facility is Token Ring, severity is 3 (error), mnemonic is WIREFAULT, and, of course, the text message is reporting a wire fault condition.

Table 4-3 *Logging Message Levels with Severity*

Level Name	Severity	Description	Syslog Definition
Emergencies	0	System unusable	LOG_EMERG
Alerts	1	Immediate action needed	LOG_ALERT
Critical	2	Critical conditions	LOG_CRIT
Errors	3	Error conditions	LOG_ERR
Warnings	4	Warning conditions	LOG_WARNING
Notifications	5	Normal significant conditions	LOG_NOTICE
Informational	6	Informational messages only	LOG_INFO
Debugging	7	Debugging messages	LOG_DEBUG

show logging Command

To display the state of syslog error and event logging, including host addresses, which type of logging (destination) is enabled, and other logging statistics, use the **show logging** (privileged EXEC) command. This command also displays the messages that are logged in the buffer.

Example 4-2 displays sample output of the **show logging** command.

Example 4-2 The *show logging* Command

```
Router#show logging
Syslog logging: enabled
  Console logging: disabled
  Monitor logging: level debugging, 13 messages logged.
  Trap logging: level informational, 13 messages logged.
  Logging to 171.16.20.20
SNMP logging: enabled, retransmission after 30 seconds
  69 messages logged
  Logging to 10.1.1.30, 0/10
  Logging to 10.2.1.40, 0/10
  Logging to 10.3.2.50, 0/10
```

Reachability and Step-by-Step Path Tests

Testing reachability of a node from another node in a network is one of the most basic tests to perform during support tasks. Testing the path a packet takes (identifying the nodes it goes through) is another very useful technique for troubleshooting. These tests are often used during fact gathering or when testing for results of an action taken. Ping has traditionally been known as an IP layer testing application. Ping is now available for other protocols, such as IPX. Cisco IOS provides Ping for IP, IPX, AppleTalk, and a few other protocols such as DECNET, XNS, CLNS, and VINES. Trace is an IP-layer path discovery/testing tool. Prior to IOS 12.0, Cisco IOS furnished **trace** for IP protocol only; as of IOS 12.0, however, **trace** is also available for IPX.

ping Command (IP) (User and Privileged)

The **ping** command is supported at the user and privileged exec modes. When used at the user mode, a set of default parameters such as five echoes, 100 bytes each with two-second time-outs will be used (in non-verbose form). You may enter an IP address or a name with the **ping** command (if the name can be resolved to an IP address using the local HOSTS table or using a DNS server).

Ping sends ICMP echo (echo request) to the destination, and the destination node replies to the source with an ICMP echo-reply. If you receive five echo-replies to the five echoes (echo requests) submitted, it means that five 100-byte packets could travel to the destination and back, each within a two-second time interval. There is a distinct possibility that the first

of the five echoes times out; the cause is usually attributed to the need for ARP, or, in case of a DDR connection, to the need to build a circuit.

Example 4-3 displays sample outputs of the **ping** command. If all of the five packets don't get to the destination and back, or at least not within the two-second time interval, you need to investigate further. In this situation, you should usually choose closer and closer targets; once you find one that you can communicate with, you can define the problem area and focus your troubleshooting efforts. When you encounter timeouts or administratively prohibited cases, you will have to discover (and deal with) the busy or the protected devices accordingly.

Example 4-3 *User Mode ping (IP)*

```
RouterA> ping routerB
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.2.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent, round-trip min/avg/max = 1/3/4 ms

RouterA> ping 172.16.5.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.5.5, timeout is 2 seconds:
.U.U.
Success rate is 0 percent (0/5)
```

A fundamental point of the ping tool is that it tests the round-trip path to and from a target. I often notice that when ping fails, people focus all of their effort on trying to troubleshoot the local device (source), and they forget that a ping failure is often caused by the destination device not having a path for sending the echo reply back to the source. See Table 4-4 for short descriptions of the test characters used by the ping facility. To abort a ping session, type the escape sequence (Shift, Control, and the 6 key, all at once). Another useful command that you can use when troubleshooting connectivity issues is the **debug ip icmp** command.

Table 4-4 *ping (IP) Response Characters*

Character	Description
!	Reply received
.	Time-out
U	Destination unreachable
N	Network unreachable
P	Protocol unreachable
Q	Source quench
M	Could not fragment
?	Unknown packet type

You also have the option of using the extended ping in privileged exec mode. From the privileged mode, if you just enter the **ping** command, you will be prompted for the protocol (default is IP). After selecting IP, you are prompted for the target IP address, repeat count, datagram size, and timeout in seconds; finally, you are asked if you are interested in extended commands. Table 4-5 provides explanations for the parameters you are prompted for if you choose the extended commands option.

Table 4-5 *ping (IP) Extended Commands*

Field	Explanation
Source address:	You may enter one of the router's local IP addresses or one of its interfaces.
Type of service [0]:	You may turn this bit to 1 to indicate Internet Service Quality selection.
Set DF bit in IP header? [no]:	If you answer yes, the Don't Fragment option will not allow this packet to be fragmented when it has to go through a segment with a smaller MTU and you will receive an error message from the device that wanted to fragment the packet.
Data Pattern [0xABCD]:	This prompt allows you to modify the 16-bit data pattern. All ones and all zeroes testing are commonly used to check sensitivity problems at the CSU/DSU or to detect cable problems such as crosstalk.
Loose, Strict, Record, Timestamp, Verbose [none]:	Even though it looks like this prompt is offering one (or none) of the options listed, if you select one, the prompt shows up again, in case you wanted to select more than one of the available options. If you select any option, Verbose is automatically selected also. Record is a very useful option because it displays the address(es) of the hops (up to nine) the packet goes through. Loose allows you to influence the path by specifying the address(es) of the hop(s) you want the packet to go through, perhaps as well as other hop(s). With the Strict option you specify the hop(s) that you want the packet to go through, but no other hop(s) are allowed to be visited as well. The difference between using the Record option of this command and using the traceroute command is quite interesting and worth discussing. The Record option of this command not only informs you of the hops that the echo request (of ping) went through to get to the destination, but it also informs you of the hops it visited on the return path. With the traceroute command you do not get information about the path that the echo reply takes.
Sweep range of sizes [n]:	Allows you to vary the size of the packets.

ping Command (IPX and AppleTalk)

Cisco IOS makes ping available for a number of protocols including IPX and AppleTalk. Ping for IPX and AppleTalk is available in user and privileged mode. Example 4-4 shows the syntax and the sample outputs for IPX and AppleTalk user mode pings.

Example 4-4 *User Mode ping (IPX and AppleTalk)*

```
Router> ping ipx 1000.0000.0c02.f3b4
Type escape sequence to abort.
Sending 5, 100-byte Novell Echoes to 1000.0000.0c02.f3b4, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Router>ping appletalk 100.50
Type escape sequence to abort.
Sending 5, 100-byte AppleTalk Echoes to 100.50, timeout is 2 seconds:
!!!!!
Success rate is 100 percent, round-trip min/avg/max = 3/3/7 ms
```

Cisco introduced ping for IPX as of version 8.2 of the IOS. However, since this is a Cisco proprietary tool, non-Cisco devices such as Novell servers do not respond to it. If you want your Cisco router to generate Novell-compliant pings, you can do so using the global configuration command **ipx ping-default novell**. But what if you want to be able to ping (IPX) Cisco devices as well as non-Cisco (Novell-compliant) devices? If this is the case, you should not use the **ipx ping-default novell** command. If you use the privileged mode ping (IPX), one of the questions you will be prompted with is whether you want a Novell standard echo. Hence, with the privileged mode ping (IPX) you can ping Cisco devices and have the choice to generate a Novell standard ping. Table 4-6 lists the test characters displayed in IPX ping responses along with their associated descriptions.

Table 4-6 *IPX ping Response Characters*

Character	Description
!	Reply received
.	Time-out
U	Destination unreachable
C	Congestion
I	Interrupt (user interrupted the test)
?	Unknown
&	Packet lifetime exceeded

Ping for AppleTalk sends AEP (AppleTalk Echo Protocol) packets to the destination (another AppleTalk node) and waits for replies. The response characters of AppleTalk ping along with their associated descriptions are provided in Table 4-7.

Table 4-7 *AppleTalk ping Response Characters*

Character	Description
!	Reply received
.	Time-out
B	Bad echo reply received
C	Echo with bad DDP checksum received
E	Error encountered during sending of the echo packet
R	No route available to send the echo packet

traceroute Command (IP) (User and Privileged)

Use the **traceroute** command to find the path between IP devices. Trace, introduced with the release 10.0 of Cisco IOS, is currently available only for the IP protocol. The **traceroute** command can be executed from the user and the privileged exec modes, but from the privileged exec mode, you have the ability to use the extended trace, which is more flexible and informative.

NOTE The command **traceroute** is commonly used in its short form, **trace**.

With the release 12.0 of Cisco IOS, **traceroute** is also available for IPX.

The **traceroute** application starts by sending probes (UDP) with TTL value of 1, and keeps on incrementing the TTL value and sending the probes until the destination is reached. When the TTL value equals 1, the probe goes as far as the first hop (router), which responds with a time-exceeded message (ICMP TTL exceeded). Note that when a packet reaches a router its TTL is reduced by one. Next, the TTL is incremented to 2, and the probe reaches the second hop in the path to the destination, and so on, until the destination is reached. The destination node sends a port-unreachable message (ICMP port unreachable) back to the source because it cannot deliver the packet to an application (the default destination port of the probe is UDP port 33434). Cisco IOS generates three probes for each TTL value, and if a response is not received within a time interval (times out), it prints an asterisk (*) on its output. Trace terminates when the destination is reached, the maximum TTL is exceeded, or the user interrupts it with the escape sequence.

The basic **traceroute** command (available in the user and privileged exec modes) uses the IP address of the egress interface as its source IP address, uses three seconds for its timeout value, sends three probes for each TTL value, and has 30 for its maximum TTL value. On the other hand, the extended trace, available from the privileged exec mode only, allows you to modify its operational parameters. Furthermore, with the extended trace, similarly to extended ping, you can also specify the source address of your probes and, if needed, choose the Loose, Strict, Record, Timestamp, and Verbose options. Example 4-5 displays a screen capture that shows the behavior of the extended **trace** command.

It is worthwhile mentioning that Cisco documentation warns that you might get a lot of timeouts with **traceroute**. The explanation given indicates that some devices do not generate port unreachable messages, and some attempt to use the TTL value of the received probe for the response packet. Both of these cases may cause the originating device to experience a lot of timeouts (asterisks).

Example 4-5 *Extended trace*

```
Router> trace
Protocol [IP]:
Target IP address: A_BackR.ciscocit.com
Source address:
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port number[33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to A_BackR.ciscocit.com (10.11.100.200)
  1 A_StubR.ciscocit.com (172.16.15.100) 70 msec 70 msec 79 msec
  2 A_BackR.ciscocit.com (10.11.100.200) 80 msec 84 msec 82 msec
```

Information Needed by Technical Support

In some troubleshooting cases you have to seek assistance from Cisco Technical Support. When customer support engineers (CSE) at Cisco Systems open a case, they need a set of information from a caller. The following paragraphs list and describe different types of information to furnish to the CSEs.

You must identify your company and your service arrangement. Next, you have to provide a statement of your problem, a brief history of the problem, a list of reported symptoms, an indication of how often the symptoms are observed, and any actions you have taken so far. Network diagram(s) and the output of the **show version** and **show running-config** commands are among the most general information you will have to furnish.

If you are dealing with hang or crash scenarios, the output of **show stacks** and core dump (and exception dump) are usually asked for.

If your case has mostly to do with performance degradation, the following commands' outputs will provide a wealth of related facts, and are therefore necessary:

- **show interfaces**
- **show buffers**
- **show memory**
- **show processes [cpu]**

When you face loss of functionality scenarios, for instance, if a protocol or a connection is faulty, the outputs of the following commands are likely to be requested:

- **show protocol**
- **show [protocol] protocol**
- **show [protocol] route**
- **show [protocol] traffic** (e.g., **show ipx traffic**)
- **show [protocol] interfaces**
- **show [protocol] access-lists**

Regardless of the nature of your problem, the outputs of **trace**, **debug**, protocol analyzer captures, and so on may also be asked for.

Also, be aware of the command **show tech-support**, which displays output equivalent to entering many troubleshooting **show** commands at once. The **show tech-support** command output comprises the following sections:

- **show version**
- **show running-config** (in privileged exec mode)
- **show controllers**
- **show stack**
- **show interfaces**
- **show processes mem**
- **show processes cpu**
- **show buffers**

show version Command

This command is one of the most popular fact-gathering commands. Example 4-6 displays a sample of the **show version** command executed at a Cisco 2514 router.

Example 4-6 *show version Command Output*

```
A_StubR#show version
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-JS-L), Version 11.3(6), RELEASE SOFTWARE (fc1)
Copyright © 1986-1998 by cisco Systems, Inc.
Compiled Tue 06-Oct-98 22:17 by kpma
Image text-base: 0x03048CF4, data-base: 0x00001000

ROM: System Bootstrap, Version 5.2(8a), RELEASE SOFTWARE
BOOTFLASH: 3000 Bootstrap Software (IGS-RXBOOT), Version 10.2(8a),
      RELEASE SOFTWARE (fc1)

A_StubR uptime is 25 minutes
System restarted by power-on
System image file is "flash:c2500-js-l_113-6.bin", booted via flash

cisco 2500 (68030) processor (revision D) with 4096K/2048K bytes of memory.
Processor board ID 04203139, with hardware revision 00000000
Bridging software.
X.25 software, Version 3.0.0.
SuperLAT software copyright 1990 by Meridian Technology Corp).
TN3270 Emulation software.
2 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read ONLY)

Configuration register is 0x2102
```

The output of the **show version** command provides a valuable set of information. Depending on your type of router, different hardware configuration and non-standard software options are displayed by the **show version** command. The following paragraph focuses on the general output of this command:

On the first few lines of output, the **show version** command displays the IOS version number and its internal name. The IOS internal name tells you about its capabilities and options. In Example 4-6 the IOS version is 11.3(6) and its name is C2500-JS-L. For a description of the IOS naming convention for different routers, refer to Cisco Connection Online (CCO).

In the second section of the output, the Bootstrap software and the RXBOOT image versions are displayed.

Next, you can see the system uptime, how the system last restarted, and the image filename and where it loaded from (the image filename is modifiable and may not be the name it was originally given by Cisco Systems). Please note that if the router encounters errors (such as software crashes) that force the router to reload, that information (reason for reload) will be displayed here and it can be quite useful to the Cisco TAC engineer.

The section near the bottom provides hardware information (processor type, memory size, existing controllers) and non-standard software options.

The very last line of the **show version** command's output displays the value of the config-register in hex format.

Buffers and Queues

System buffers are memory allocated from main system memory (also referred to as shared memory) to hold packets while they are process-switched. There are parameters regarding these buffers that can be tuned, but that is not often recommended, and tuning them has no effect on any route caching methods you may have (fast, autonomous, silicon, or optimum switching, for instance). To enhance the (packet) processing power of your router, you are usually encouraged to take advantage of your router's route caching capabilities. Because the buffers that hold packets while they are being process switched are memory borrowed from the main system (RAM) memory, the memory available is not too limited. One can conjecture that this is an advantage of process switching. There are six buffer sizes, each of which is appropriate for a specific range of packet sizes:

Small Buffers: 104 Bytes
Middle Buffers: 600 Bytes
Big Buffers: 1524 Bytes
Very Big Buffers: 4520 Bytes
Large Buffers: 5024 Bytes
Huge Buffers: 18024 Bytes

You must keep in mind that a buffer must be allocated and free at the time a packet arrives, or the packet will be dropped and the number of misses (shown in the output of the **show buffers** command, discussed later) is incremented. Furthermore, the router cannot afford to have too many buffers allocated and free (to avoid potential drops and misses), as that will reduce the available memory (from the shared pool) needed for other purposes. This challenge is addressed by having a minimum number of buffers (for each size) allocated at all times, and dynamically allocated and de-allocated buffers based on the traffic rate (process-switched packets sent and received). The parameters that are used for managing buffers follow and apply to each buffer size:

- **Permanent**—The minimum number of buffers allocated. Buffers are de-allocated (trimmed) at times, but the number of allocated buffers will not go below this.
- **Max-Free**—When the number of buffers that are allocated but not used (free) exceeds this value, a trim (de-allocation) is triggered. The memory is returned to the shared pool, and can be used for other purposes.

- **Min-Free**—As the allocated (free) buffers are used up, the number of free buffers is reduced. When the number of free buffers reduces to be equal to the Min-Free parameter, buffer allocation (create) is triggered. This attempts to always have a minimum number of allocated and unused buffers available for each packet size.
- **Initial**—This parameter indicates how many buffers should be allocated (for a particular packet size) at the router initialization time. This value is usually larger than Permanent.

When faced with performance degradation support scenarios, the **show buffers** command is very useful. If you see a large number of misses reported for a particular buffer size, you may have to adjust the Permanent or Min-Free parameters for that particular buffer size.

The number of failures indicates how many times the allocation of more buffers has been unsuccessful. Please consult your technical support representative before adjusting any buffer parameters. Example 4-7 shows the syntax for adjusting buffer parameters.

Example 4-7 *Adjusting Buffer Parameters*

```
A_StubR(config)#buffers ?
Ethernet  IEEE 802.3
Serial    Serial
big       Big buffers
huge      Huge buffers
large     Large buffers
middle    Middle buffers
small     Small buffers
verybig   Very Big buffers

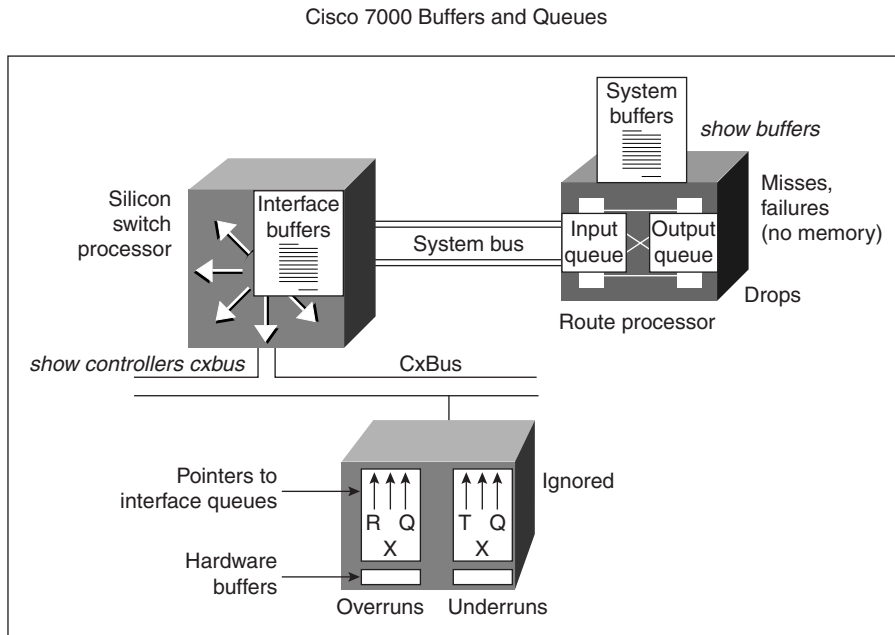
A_StubR(config)#buffers big ?
initial   Temporary buffers allocated at system reload
max-free  Maximum number of free buffers
min-free  Minimum number of free buffers
permanent Number of permanent buffers

A_StubR(config)#buffers big min-free ?
<0-20480> Number of buffers
```

Buffers and Queues (Cisco 7000/7010)

In the Cisco 7000, routers' packets are held in interface hardware buffers, SP or SSP buffers, and RP buffers. Figure 4-6 displays those Cisco 7000 router components along with the error conditions associated with each of them.

Figure 4-6 Cisco 7000 Buffers and Error Conditions



Interface input and output queues on the RP are linked lists of processor buffers used for each interface. An interface queue can hence be composed of different-sized buffers. The queue associated with an interface can grow (to a configured limit) and shrink (down to zero). When a packet must be held in the input queue or output queue of an interface (on the RP), a buffer is taken away from the appropriate allocated and free list (based on size). However, if the input or output queue of an interface reaches its maximum size, the queue cannot grow larger. In other words, after a queue reaches its maximum size, it will likely start dropping subsequent packets. When you look at the **show interface** command output, pay attention to the number of drops reported for input and output queues. The default size of the input queue (hold queue) and output queue (hold queue) are 75 (packets) and 40 (packets) correspondingly.

The SP or SSP has 512 KB of memory. This memory is partially used for route caching (autonomous switching) and partially used for buffering the packets copied in from different interface processors. Use the **show controllers cxbus** command to find the allocation of interface buffers on the SP.

There are hardware buffers that are used to hold packets on the interface hardware itself. If an interface buffer gets full, an ignore is registered. In other words, every time an interface cannot accept a frame due to an input buffer being full, the ignore counter is incremented by one. Ignore is the result of either a high-speed interface accepting frames close to wire speed and SP not being able to put the data out of the interface hardware buffer fast enough, or a CxBus that is so busy that the interface hardware does not get enough chances to unload data from its buffers to the SP.

Buffers and Queues (Cisco 4000/3000/2500)

In Cisco 4000/3000/2500 routers, because there is no SP, SSP, or RSP, buffers reside on the shared memory. Packets enter into an interface (hardware buffer), and they are sent directly to the shared memory.

In shared memory, a packet is kept in the input queue corresponding to the ingress interface. Next, an output interface for the packet is possibly selected (perhaps after the process switching or fast switching task completes). Then the packet is held in the output queue of the egress interface, until it is finally sent to the output interface hardware.

show buffers Command

The **show buffers** command displays information (statistics) on buffer elements, public buffer pools, and interface buffer pools. Buffer elements are small data structures that are used for internal operating system queues or when a buffer must be associated with more than one queue. The public buffer pools are presented in the second section of the **show buffers** output and each buffer size (small, middle, and so on) is presented with its own statistics. The last section of the **show buffers** command output displays the buffer statistics for each of the router's interfaces. Example 4-8 presents a sample output (partial) from the **show buffers** exec command.

Example 4-8 A Sample Output of the *show buffers* Exec Command

```
A_StubR#show buffers
Buffer elements:
    500 in free list (500 max allowed)
    3846 hits, 0 misses, 0 created

Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50):
    49 in free list (20 min, 150 max allowed)
    1440 hits, 0 misses, 0 trims, 0 created
    0 failures (0 no memory)
Middle buffers, 600 bytes (total 25, permanent 25):
    25 in free list (10 min, 150 max allowed)
    703 hits, 0 misses, 0 trims, 0 created
    0 failures (0 no memory)
Big buffers, 1524 bytes (total 50, permanent 50):
```

Example 4-8 A Sample Output of the *show buffers Exec Command (Continued)*

```

50 in free list (5 min, 150 max allowed)
150 hits, 0 misses, 0 trims, 0 created
0 failures (0 no memory)
VeryBig buffers, 4520 bytes (total 10, permanent 10):
  10 in free list (0 min, 100 max allowed)
  0 hits, 0 misses, 0 trims, 0 created
  0 failures (0 no memory)
Large buffers, 5024 bytes (total 0, permanent 0):
  0 in free list (0 min, 10 max allowed)
  0 hits, 0 misses, 0 trims, 0 created
  0 failures (0 no memory)
Huge buffers, 18024 bytes (total 0, permanent 0):
  0 in free list (0 min, 4 max allowed)
  0 hits, 0 misses, 0 trims, 0 created
  0 failures (0 no memory)

Interface buffer pools:
Ethernet0 buffers, 1524 bytes (total 32, permanent 32):
  8 in free list (0 min, 32 max allowed)
  24 hits, 0 fallbacks
  8 max cache size, 8 in cache

```

show memory Command

The **show memory** exec command is often used to check the amount of a router's free memory. In troubleshooting cases where router performance is the focus, this is a major command used to see the statistics about the router's memory. Example 4-9 displays a sample output (partial) of this command executed on a Cisco 2514 router.

Example 4-9 A Sample Output of the *show memory Exec Command*

```

A_BackR#show memory
Head                Total(b)   Used(b)   Free(b)   Lowest(b)  Largest(b)
Processor 90C3C     3597252   983900    2613352   2604696    2611612
I/O        400000     2097152   391980    1705172   1705172    1704752

Processor memory

Address  Bytes  Prev.  Next  Ref  PrevF  NextF  Alloc PC What
90C3C   1064   0      91090  1    .      .      1A1630 List Elements
91090   2864   90C3C  91BEC  1    .      .      31A1630 List Headers
91BEC   2668   91090  92684  1    .      .      3150160 TTY data
92684   2000   91BEC  92E80  1    .      .      3152534 TTY Input Buf
92E80   512    92684  930AC  1    .      .      3152564 TTY Output Buf
930AC   3000   92E80  93C90  1    .      .      31B2252 Interrupt Stack
.
.
.

I/O memory

```

continues

Example 4-9 A Sample Output of the *show memory Exec Command (Continued)*

Address	Bytes	Prev.	Next	Ref	PrevF	NextF	Alloc PC	What
400000	260	0	400130	1			3183DD8	*Packet Data*
400130	260	400000	400260	1			3183DD8	*Packet Data*
400260	260	400130	400390	1			3183DD8	*Packet Data*
400390	260	400260	4004C0	1			3183DD8	*Packet Data*
.								
.								
.								

The **show memory** exec command's output is organized in separate sections. In the first section you can see the summary statistics about processor memory and I/O memory (see Example 4-9). Then you can see the more detailed (block-by-block) display of memory information first for the processor memory, and then for the I/O memory. The output is not uniform across different router platforms. For example, if you execute this command on a Cisco 7000 router, the output will include processor memory and multibus memory statistics. If you execute this command on Cisco 4000 series routers, you will receive information about SRAM and I/O memory as well as processor memory. In all cases, the processor memory statistics are shown. You must pay attention to the total amount of memory, amount used, and the total amount of free memory. The Cisco TAC engineer helping you might ask questions about your router's memory utilization or simply request the output. Ask your technical support representative about the amount of free memory he/she recommends (on average) to be available.

show processes Command

The **show processes** exec command displays your router's CPU utilization and a list of active processes along with their corresponding process ID, priority, scheduler test (status), CPU time used, number of times invoked, and so on. This command is also very useful when you are evaluating your router's performance and CPU utilization.

A sample output of the **show processes** command is shown in Example 4-10. As you can see, the first line of the output shows the CPU utilization for the last five seconds, one minute, and five minutes. The output provides 4%/4% in front of the CPU utilization for the last five seconds: the first number is the total utilization and the second number is the utilization due to interrupt routines.

Example 4-10 A Sample Output of the *show processes Exec Command*

```
A_BackR#show processes

CPU utilization for five seconds: 4%/4%; one minute: 6%; fiveminutes: 5%

PID Q Ty PC Runtime(ms) Invoked uSecs Stacks TTY Process
1 C sp 31B6178 28 353 79 736/1000 0 Load Meter
```

Example 4-10 A Sample Output of the *show processes Exec* Command (Continued)

2	M *	0	324	154	2103	2588/4000	0	Exec
3	L st	31A7112	5584	167	33437	1768/2000	0	Check heaps
4	C we	31ACF5A	0	1	0	1732/2000	0	Pool Manager
5	M st	3147E02	4	2	2000	1700/2000	0	Timers
6	M we	30E6690	0	2	0	1700/2000	0	SerialBackgroun
7	L we	31D4BD0	4	52	76	1612/2000	0	ARP Input
8	M we	32D6848	4	2	2000	1624/2000	0	DDR Timers
9	M we	30EA414	0	1	0	1736/2000	0	SERIAL A'detect
10	M we	31F9270	604	398	1517	3024/4000	0	IP Input
11	M we	3264668	84	206	407	1556/2000	0	CDP Protocol
.
.
.

When you decide to use the **show processes** command, try to execute it a few times, with a one-minute lapse in between, to get a more reliable idea about which processes are invoked most often and how much CPU time they consumed. You may also execute the **show processes cpu** command to get the five-second, one-minute, and five-minute display of CPU utilization for each process. Table 4-8 shows some of the column headings used in the output of the **show processes** command with a brief description for each of them.

Table 4-8 Some of the Column Headings of the *show processes* Command Output

Column Heading	Description
PID	Process ID
Q	Priority (C: Critical, H: High, M: Medium, L: Low)
Ty	Scheduler Test (status) *: currently running E: waiting We: waiting for an event Sa: sleeping until an absolute time Si: sleeping for a time interval St: sleeping until a timer expires Hg: hung Xx: dead

continues

Table 4-8 *Some of the Column Headings of the show processes Command Output (Continued)*

Column Heading	Description
PC	Program Counter
Runtime	CPU time the process has used (in milliseconds)
Invoked	Number of times the process has been invoked
uSecs	Number of microseconds of CPU time used at each invoke

show controllers cxbus Command

The **show controllers cxbus** exec command is used on the Cisco 7x00 series routers to display information about the SP, the CxBus controller, and the contents and microcode of cards attached to the bus. This command's output for the most part is useful for diagnostic tasks performed by Cisco support engineers only. Example 4-11 is a sample output of the **show controllers cxbus** command executed on a Cisco 7000 router.

Example 4-11 *A Sample Output of the show controllers cxbus Command*

```

Router# show controllers cxbus
Switch Processor 5, hardware version 11.1, microcode version 172.6
  Microcode loaded from system
  512 Kbytes of main memory, 128 Kbytes cache memory
  75 1520 byte buffers, 86 4484 byte buffers
  Restarts: 0 line down, 0 hung output, 0 controller error
CIP 3, hardware version 1.1, microcode version 170.1
  Microcode loaded from system
  CPU utilization 7%, sram 145600/512K, dram 86688/2M
  Interface 24 - Channel 3/0
    43 buffer RX queue threshold, 61 buffer TX queue limit, buffer size 4484
    ift 0007, rql 32, tq 0000 0468, tq1 61
    Transmitter delay is 0 microseconds
  Interface 25 - Channel 3/1
    43 buffer RX queue threshold, 61 buffer TX queue limit, buffer size 4484
    ift 0007, rql 34, tq 0000 0000, tq1 61
    Transmitter delay is 0 microseconds
.
.
.

```

The top portion of the output displayed by Example 4-11 tells you the SP's hardware and microcode version, main memory and cache memory size, number of different buffer sizes, and the number of restarts due to line down (communication line), hung output, or controller error. The second part of the output displayed by Example 4-11 includes information about the Channel Interface Processor (CIP). In this section, you can see the CIP's hardware and microcode version, CPU utilization, free and total SRAM memory (Static RAM is a high speed memory used for operational code), free and total DRAM

memory (Dynamic RAM is normal memory used for packets, data, and so on), and information about each of the CIP interfaces.

show stacks Command

show stacks is an exec command that is commonly used to diagnose system crash situations. The first section of this command's output displays stack utilization of processes and interrupt routines, and the reason for the last system reboot. When a system crash happens, failure type, failure program counter (PC), address (operand address), and a stack trace are saved by the ROM Monitor. The **show stacks** command displays the data saved by the ROM Monitor. The stack trace is displayed in the second section of the **show stacks** command output (if there has been a system failure).

In the past, support engineers would submit the stack trace of their router to Cisco System's technical support representatives, who had access to symbol tables, object files, source code, and the stack decoder software. Today, the stack decoder is available online (from the CCO) and you can cut your router's stack trace from the output of the **show stacks** command and paste it in the input field of the stack decoder software. Stack decoder decodes the stack trace and creates a symbol file. The symbol file (perhaps along with other information in the trace) usually provides enough information to isolate the cause of any problems that were experienced.

Example 4-12 shows an example of the show stacks output from a bus error. The message "System was restarted by bus error" indicates that the processor tried to use a device or a memory location that either did not exist or did not respond properly (this could be due to a software bug or a hardware problem). Operand address, the address the processor was trying to access when the system crashed, is used as the clue to tell if the failure was due to software or hardware. If the operand address (reported on the output of the **show stacks** command) is valid, the problem is probably in the hardware. In other words, the operand address, not the program counter, provides the memory map location of the error, which can be used to infer the general area of the router where the error occurred.

Example 4-12 A Sample Output of the *show stacks* Command

```
Router# show stacks

Minimum process stacks:
Free/Size  Name
 652/1000  Router Init
 726/1000  Init
 744/1000  BGP Open
 686/1200  Virtual Exec
Interrupt Level stacks:
Level  Called Free/Size  Name
 1      0      1000/1000  env-flash
 3      738    900/1000  Multiport Communications Interfaces
 5      178    970/1000  Console UART
```

continues

Example 4-12 A Sample Output of the *show stacks* Command (Continued)

```

System was restarted by bus error at PC 0xAD1F4, address 0xD0D0D1A
GS Software (GS3), Version 10.2
Compiled Tue 11-Aug-94 13:27 by jthomas
Stack trace from system failure:
FP: 0x29C158, RA: 0xACFD4
FP: 0x29C184, RA: 0xAD20C
FP: 0x29C1B0, RA: 0xACFD4
FP: 0x29C1DC, RA: 0xAD304
FP: 0x29C1F8, RA: 0xAF774
FP: 0x29C214, RA: 0xAF83E
FP: 0x29C228, RA: 0x3E0CA
FP: 0x29C244, RA: 0x3BD3C

```

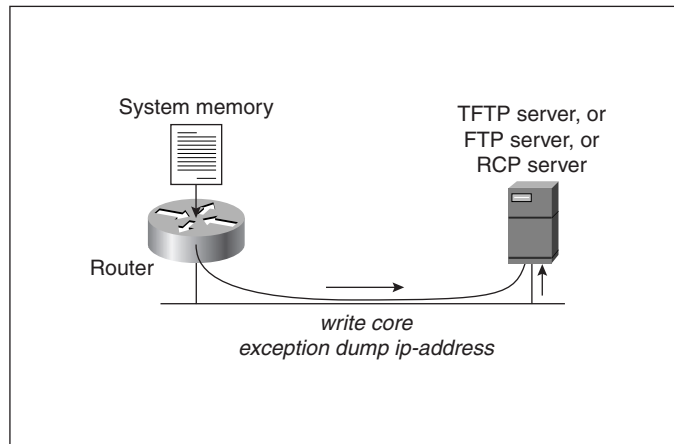
Failure types are usually one of the following: bus error, address error, watchdog timeout, parity error, or emulator trap. Table 4-9 displays common failure types with a brief description for each of them.

Table 4-9 Common Failure Types Reported by the *show stacks* Command

Failure Type	Description
Bus error	The processor tried to use a device or a memory location that either did not exist or did not respond properly (could be due to software bug or hardware error).
Address error (software forced crash)	The software tried to access data on incorrectly aligned boundaries (usually indicates a software bug).
Watchdog timeout	Watchdog timer was not reset and caused a trap. Watchdog timers are used by Cisco processors to prevent certain system hangs (indicates a hardware or software bug).
Parity error	Internal hardware checks have failed (this is due to hardware problems).
Emulator trap	Processor executed an illegal instruction (illegal branching). A hardware problem, such as ROM failure, can also cause an emulator trap error.

Core Dumps

The full copy of memory image is called a core dump. This image can be useful for determining the cause of a crash. Core dumps are usually submitted to Cisco support engineers, who are specialized in analyzing the memory image (using source codes, memory maps, etc). The core dump can transfer the binary image file using TFTP, FTP, or RCP protocols (see Figure 4-7). You must remember that performing a core dump disrupts regular network operation.

Figure 4-7 *Creating Core Dump*

If your router is malfunctioning, but has not crashed, use the **write core** command to generate a core dump without reloading. You must make sure that your server (TFTP, FTP, or RCP server) is reachable and has enough storage space. You must also learn the file-naming convention that the server's operating system supports. Finally, find out whether you need to create an empty file (with the desired name) on the server in advance.

The **exception dump ip-address** global configuration command (*ip-address* is the address of your TFTP, FTP, or RCP server) causes the router to attempt to produce a core dump when it crashes. By default, the core dump is written to a file named *hostname-core* on your server (*hostname* is the name of the router). If you want to change the name of the core file, the **exception core-file filename** command allows you to do that.

Keep in mind that depending on the type of crash, this procedure does not always succeed. Finally, you need to be aware that using TFTP has a limitation. If you use TFTP to dump the core file to a server, the router will only dump the first 16 MB of its memory image (and if the router's memory is larger than 16 MB, part of the image will be missing). Hence, RCP or FTP is recommended to dump the core file for routers with larger than 16 MB of memory.

Foundation Summary

The Foundation Summary is a collection of quick reference information that provides a convenient review of many key concepts in this chapter. For those of you who already feel comfortable with the topics in this chapter, this summary helps you recall a few details. For those of you who just read this chapter, this review should help solidify some key facts. For any of you doing your final prep before the exam, these tables and figures are a convenient way to review the day before the exam.

Handling Cisco IOS Troubleshooting Tools

These tools and commands provide a wealth of information that can be very useful for troubleshooting purposes, but due to their impact on router and network performance they need to be handled and used properly. These powerful tools use up a router's CPU cycles and memory, may be given higher priority than network traffic, and may disable some features such as fast switching.

Routing and Switching Tasks and Route Caching

Routing and switching are two of the important tasks that routers perform. Routing is basically the process of selecting one or more output interfaces for a packet (if possible), whereas switching is basically the process of moving the packet within the anatomy of the router from one location or component of the router to another. Switching is simpler than routing. Routing requires the main processor's attention (it interrupts the main processor) and takes CPU cycles and therefore it is responsible for most of the delay (latency) introduced by a router.

Route caching is a technique that reduces this latency and frees up the main processor from having to handle too many interrupts. Once a packet is processed by the routing process, and an output interface is selected based on the packet's destination (layer 3) address, this address/output-interface pair can be saved in a cache and be used for quick processing of the subsequent packets with the same destination network address. Because both routing and switching tasks were performed on the first packet, it is considered to have been process-switched. The subsequent packets (with the same destination network address) need not be process-switched. Since the routing information built from the processing of the first packet is available in the routing cache, the subsequent packets are fast switched. The place where the route caching information is held varies from router to router, and it also depends on the option enabled.

Route Caching Methods and Commands

Route caching methods available in different Cisco router series and the commands to enable them are displayed in Table 4-10 (to disable any of these switching modes, use the **no** form of the command):

Table 4-10 *Route Caching Methods and Commands*

Interface Configuration Command (IP is shown as the protocol example)	Route Caching (Switching) Method Enabled	Cisco Router Series Support
ip route-cache	Fast switching	All
ip route-cache cbus	Fast switching and autonomous switching	7000 series with SP
ip route-cache sse	Silicon switching	7000 series with SSP
ip route-cache optimum	Optimum switching	7x00 series with RSP
ip route-cache distributed	Distributed	7x00 series with VIP
ip route-cache flow	Netflow	7x00 series with RSP
ip route-cache flow	Distributed Netflow	7x00 series with RSP and VIP
ip route-cache distributed		

Debug Notes

debug is a troubleshooting command used to display information about various router operations and the related traffic generated or received by the router, as well as any error messages. This tool lets you discover significant facts about the working and faulty software and/or hardware components.

- **debug** is available from the privileged exec mode (of Cisco IOS).
- **debug** is treated as a very high priority task.
- **debug** can consume a significant amount of resources.
- The router is forced to process-switch the packets being debugged.
- **debug** must not be used as a monitoring tool.
- Use it for a short period of time and as a troubleshooting tool.

If you want to see a timestamp with each line of the **debug** output, you must load the timestamp service using this command:

```
router(config)#service timestamps debug [datetime | uptime]
```

If you plan to see the debug output from within a Telnet session, you need to enter the **terminal monitor** command.

Usually, the **debug** command is used to diagnose a specific facility, task, or protocol. Sometimes the protocol has a specific member that you may want to focus on. Once you decide what you want to debug, then you usually have a choice to use the events option or the packets option of the **debug** command for that specific protocol. Event debugging is less resource intensive than packet debugging, but packet debugging produces more information.

Turning debugging on for everything (using the **debug all** command) is seriously discouraged in production networks. You will get a tremendous amount of information, very fast, but it can severely diminish the router's performance or even render it unusable.

Before starting to use the **debug** command, see the CPU utilization of your router (using the **show processes cpu** command). If your router's CPU utilization is consistently at 50% or more, you are advised to debug events instead of packets.

If possible, use the **debug** command during periods when network traffic is not at its peak and fewer critical business applications are active. Cisco routers give the **debug** command higher priority (with respect to CPU cycles) than network traffic.

Always remember to undo **debug** as soon as possible. You can use the **no debug {argument}** to turn off a specific debugging type. The **no debug all** or **undebug all** commands can be used to turn off all types of debugging that may be on.

For troubleshooting, also consider using protocol analyzers to capture and display network traffic. These have little or no impact on your network performance, yet they provide valuable information.

Using an access list with your **debug** command helps you focus the debug output on the task you are troubleshooting. The syntax for using an access list with the **debug** command is:

```
Router# debug ip packet detail access-list-number
```

Logging Options

Table 4-11 shows logging options and their corresponding commands.

Table 4-11 *Logging Options and Their Corresponding Commands*

Command	Usage Explanation
logging console [<i>level</i>]	This command turns console logging on and specifies the level of logging to be directed to the console. (The default setting is Enabled.)
	The no logging console command disables console logging.

Table 4-11 *Logging Options and Their Corresponding Commands (Continued)*

Command	Usage Explanation
logging buffered [level]	Use this command to enable sending logging messages to the internal buffer (use no logging buffered to disable it) and specify the level of logging desired to be buffered. This feature is enabled by default.
logging monitor [level]	Use this command to enable sending logging messages to the virtual terminal sessions (use no logging monitor to disable it) and specify the level of logging desired to be directed to the virtual terminal lines. From within a virtual terminal session, typing the command terminal monitor enables displaying of logging messages. The command terminal no monitor turns this feature off.
logging trap [level]	This command allows you to enable sending logging messages to syslog servers and specify the level of these messages. The no logging trap command disables this feature. The default level is Informational. (See also the explanation for the logging [ip-address] command below)
logging [ip-address]	This command identifies the IP address of the syslog server so that the router can direct its logging messages to this address. If you have a list of syslog servers to which you want to send the logging messages, you may enter this command one time with each server's appropriate address. Use the no form of this command to take a server off the list.

The following is a comparison of the overhead of different logging methods:

Buffered logging < Syslog < Virtual terminal < Console logging

Information Needed by Technical Support

General Information

- Your company's name and service arrangement number
- A statement of the problem
- A brief history of the problem

- A list of reported symptoms, how often the symptoms are observed, and the actions taken so far
- Network diagram(s)
- A list of protocols in use and policies in place
- Outputs of the **show version** and **show running-config** commands

Crash Situations

- **show stacks**
- Core dump

Performance Degradation Situations

- **show interfaces**
- **show buffers**
- **show memory**
- **show processes [cpu]**

Loss of Functionality Situations

- **show protocol**
- **show [protocol] route**
- **show [protocol] traffic**
- **show [protocol] interfaces**
- **show [protocol] access-lists**

Output of the show tech-support Command

- **show version**
- **show running-config** (in privileged exec mode)
- **show controllers**
- **show stack**
- **show interfaces**

- **show processes mem**
- **show processes cpu**
- **show buffers**

Terms and Concepts Related to Buffer and Queues

Buffer Sizes:

- Small buffers: 104 Bytes
- Middle buffers: 600 Bytes
- Big buffers: 1524 Bytes
- Very big buffers: 4520 Bytes
- Large buffers: 5024 Bytes
- Huge buffers: 18024 Bytes

Configuration Parameters:

- **Permanent**—The minimum number of buffers allocated. Buffers are de-allocated (trimmed) at times, but the number of allocated buffers will not go below this.
- **Max-Free**—When the number of buffers that are allocated but not used (free) reaches this value, a trim (de-allocation) is triggered. The memory is returned to the shared pool and can be used for other purposes.
- **Min-Free**—As the allocated (free) buffers are used up, the number of free buffers is reduced. When the number of free buffers reduces to be equal to the Min-Free parameter, buffer allocation (create) is triggered. This attempts to always have a minimum number of allocated and unused buffers available for each packet size.
- **Initial**—This parameter indicates how many buffers should be allocated (for a particular packet size) at the router initialization time. This value is usually larger than Permanent.

Reported Conditions

- **Ignored**—The number of packets ignored is shown in the output of the **show interfaces** command. If a buffer (on the interface hardware) gets full, an ignore is registered. In other words, every time an interface cannot accept a frame due to the input buffer being full, the ignore counter is incremented by one.

- **Dropped**—The number of dropped packets is shown in the output of the **show interfaces** command. If the input or output queue of an interface reaches its maximum size, the queue cannot grow larger and will start dropping the subsequent packets.
- **Misses**—The number of misses is shown in the output of the **show buffers** command. The number of misses is incremented for each occurrence of a buffer not being allocated and free at the time a packet arrives.
- **Failures (no memory)**—The number of failures is shown in the output of the **show buffers** command, and it indicates how many times the allocation of more buffers has been unsuccessful.

Q&A

The answers to the following questions can be found in Appendix A. Some of the questions in this section are repeated from the “Do I Know This Already” Quiz so that you can gauge the advancement of your knowledge of this subject matter.

- 1 Briefly explain why Cisco IOS troubleshooting commands and tools need proper handling.

- 2 What does proper handling of troubleshooting tools entail?

- 3 Define switching and specify whether it is considered a complex task.

- 4 Define routing and compare its complexity to switching.

- 5 List the sources of information used by a routing process for building its routing table.

- 6 When a packet is process-switched, what major tasks are performed?

7 Provide a short and generic explanation for route caching (or fast switching) and the purpose behind it.

8 Which of the route caching methods are not enabled by default? And from which configuration mode (prompt level) can they be enabled?

9 On which component of the Cisco 7000 router is the Fast Switch Cache located?

10 Name the two major components that participate in the routing and switching tasks within a Cisco 7000 router.

11 What is the difference between the Silicon Switch Processor (SSP) and the Switch Processor (SP) with respect to the switching cache options?

12 What is the command for enabling IP fast switching on an interface?

- 13** What is the command for enabling IP autonomous switching on a Cisco 7000 series router interface?

- 14** What is the command for enabling IP silicon switching on a Cisco 7000 series router (with SSP) interface?

- 15** With regard to speed and switching optimization, how did Cisco Systems improve the Cisco 7500 routers in comparison to the 7000 series?

- 16** What switching method (route caching) can be enabled on the 7000/7500 series routers' VIP cards?

- 17** On which Cisco router models is Netflow switching supported? (Specify the IOS version.)

- 18** What information does Netflow use as the basis of identifying a flow?

- 19** Briefly describe the advantages of Netflow switching. Also specify whether there should be any precautions with respect to enabling Netflow switching.

- 20** What is the command for enabling IP Netflow switching on a supported router interface?

- 21** What are the only switching (route caching) options on the 4000, 3000, and 2500 series routers?

- 22** The output of which command includes information about whether fast switching is enabled/disabled for a particular protocol on a particular interface?

- 23** Which command can be used to see the statistics on the number of packets that are process switched and fast switched?

- 24** Provide at least three examples of operations or packet types that are process switched.

25 What are some facts about the **debug** privileged exec mode command that one must keep in mind before using it?

26 Which service must be loaded if you need to see a timestamp with each of the **debug** output lines? (Also provide the command syntax.)

27 What command enables you to see the debug output from within a Telnet session?

28 Compare debugging with the packet option to debugging with the events option.

29 Which command enables debugging for all protocols and activities? Are there any concerns regarding usage of this command?

30 Before you enable debugging on a router, you are encouraged to check the router's CPU utilization. What is the command that allows you to do that? If the utilization is above 50%, are you encouraged to debug packets or to debug events?

- 31** Specify the command syntax for enabling debugging for those IP packets that satisfy (are permitted by) an access list 100.

- 32** What is the default setting (for example, enabled/disabled, default destination) for message logging?

- 33** How do the logging message destination options compare in terms of the overhead they introduce to a router?

- 34** Which Cisco IOS router command turns console logging on and specifies the level of logging to be directed to the console?

- 35** By using which Cisco IOS router command can you enable sending logging messages to the internal buffer and specify the level of logging desired to be buffered?

- 36** Specify the Cisco IOS router command that enables sending logging messages to the virtual terminal sessions and specifies the level of logging desired to be directed to the virtual terminal lines.

- 37 What Cisco IOS router command would you use to make a router's logging messages be sent to a syslog server at IP address 10.1.2.3?

- 38 What Cisco IOS router command would you use to make virtual terminal lines receive logging messages at the errors level and higher (i.e., errors, critical, alerts, emergencies)?

- 39 What is the severity of the following logging message?

```
%TR-3-WIREFAULT:Unit[0],wirefault:check the lobe cable MAU connection
```

- 40 What information does the output of the **show logging** Cisco IOS exec command display?

- 41 What are your choices in order to make your Cisco IOS router generate Novell-compliant (ipx) pings?

- 42 In which situation (loss of functionality, crash, or performance degradation) will the output of the **show stack** command most likely be asked for?

43 The outputs of the **show memory** and the **show processes [cpu]** commands will most likely be asked for in which situation (loss of functionality, crash, or performance degradation)?

44 In which situation (loss of functionality, crash, or performance degradation) will you most likely be asked to produce and provide a core dump for the technical support representative?

45 Which **show** command conveniently produces output equivalent to the output of **show version**, **show running-config**, **show controllers**, and a few other **show** commands?

46 Which Cisco IOS router command's output displays the current setting (value) of the config-register?

47 What is the outcome of not having an allocated and free buffer available for a packet?

48 Explain the role of the parameter called Permanent in buffer management.

49 Explain the role of the parameter called Max-Free in buffer management.

50 Explain the role of the parameter called Min-Free in buffer management.

51 Explain the role of the parameter called Initial in buffer management.

52 If the output of **show buffer** command displays a large number of misses, increasing the value of which one of the buffer management parameters (Permanent, Min-Free, Max-Free, Initial) will most likely remedy the situation?

53 What does the number of failures displayed on the output of **show buffer** command indicate?

54 Using what command can you find out the allocation of interface buffers on the Switch Processor of the Cisco 7x00 series routers?

55 What does the **show buffers** command display?

56 Which Cisco IOS command's output displays statistics about router memory (for example, amount of free processor memory)?

57 What information does the output of the Cisco IOS **show processes exec** command display?

58 The **show processes** command's output provides two numbers separated by a slash (for example, 4%/4%) for the CPU utilization over the last five seconds. How are those numbers interpreted?

59 Specify the command (with appropriate parameters) that displays the five-seconds, one-minute, and five-minute CPU utilization for each of the active processes.

60 What information can be gathered from the output of the Cisco IOS **show stacks exec** command?

61 Which command allows you to generate a core dump without reloading?

62 To ensure that a core dump can be sent to a server and saved successfully, what are some of the preliminary tasks and tests you must perform?

63 Which command causes the router to attempt to produce a core dump when it crashes?

64 By default, what is the name of the file that the core dump is written to?

65 Which command allows you to change the name of the core file?
