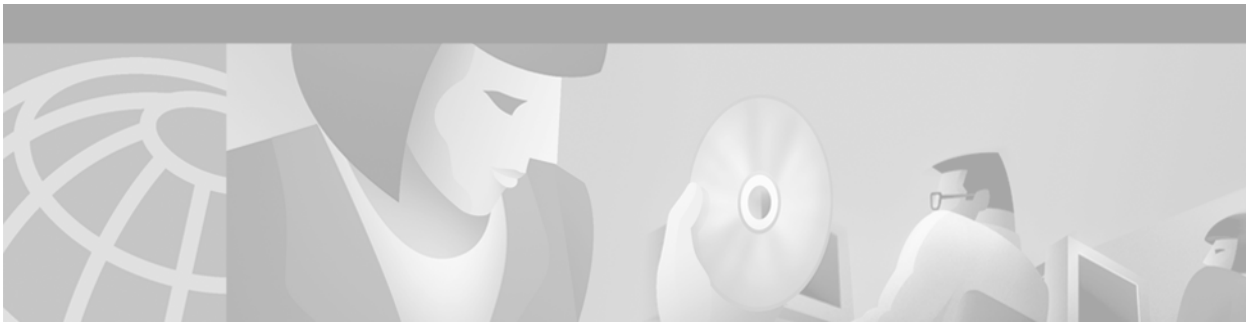

Performing Basic System Management

This chapter describes the basic tasks that you can perform to manage the general system features of the Cisco IOS software—those features that are generally not specific to a particular protocol.

This document applies to Cisco IOS Release 12.2.

For a complete description of the basic system management commands in this chapter, refer to the "Basic System Management Commands" chapter in the "Cisco IOS System Management Commands" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Basic System Management Task List

To customize the general functionality of your system, perform any of the tasks in the following sections. All tasks in this chapter are optional, though some, such as setting time and calendar services, are highly recommended.

- Configuring the System Name (Recommended)
- Customizing the CLI Prompt
- Creating and Displaying Command Aliases
- Controlling Minor Services (Recommended)
- Hiding Telnet Addresses
- Setting Time and Calendar Services (Recommended)
- Delaying EXEC Startup
- Handling an Idle Telnet Connection
- Setting the Interval for Load Data
- Limiting the Number of TCP Transactions
- Configuring Switching and Scheduling Priorities
- Modifying the System Buffer Size

See the end of this chapter for the "Basic System Management Examples" section.

Configuring the System Name

The most basic system management task is to assign a name to your system (router, access server, switch, and so on). The system name, also called the host name, is used to uniquely identify the system in your network. The system name is displayed at the CLI prompt. If no name is configured, the system default name is `Router`. To configure a name for your device, use the following command in global configuration mode:

Command	Purpose
`Router(config)# hostname name`	Sets the host name.

For an example of configuring a system name, see the section "System Configuration File Example" at the end of this chapter.

Customizing the CLI Prompt

By default, the CLI prompt consists of the system name followed by an angle bracket (>) for EXEC mode or a pound sign (#) for privileged EXEC mode. To customize the CLI prompt for your system, use either of the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# prompt string`	Customizes the CLI prompt.
`Router(config)# no service prompt config`	Disables the display of the CLI prompt.

Creating and Displaying Command Aliases

Command aliases allow you to configure alternative syntax for commands. You may want to create aliases for commonly used or complex commands. For example, you could assign the alias **save config** to the **copy running-config startup-config** command to reduce the amount of typing you have to perform, or if your users might find a **save config** command easier to remember. Use word substitutions or abbreviations to tailor command syntax for you and your user community.

To create a command alias, use the following command in global configuration mode:

Command	Purpose
`Router(config)# alias mode alias-name alias-command-line`	Configures a command alias.

To display a list of command aliases currently configured on your system, and the original command syntax for those aliases, use the following command in EXEC mode:

Command	Purpose
`Router# show aliases [mode]`	Displays all command aliases and original command syntax, or displays the aliases for only a specified command mode.

Keep in mind that any aliases you configure will only be effective on your system, and that the original command syntax will appear in the configuration file.

Controlling Minor Services

The minor services are "small servers" that run on your routing device and are useful for basic system testing and for providing basic network functions. Minor services are useful for testing connections from another host on the network.

Cisco small servers are conceptually equivalent to daemons.

Small servers provided by Cisco IOS software-based devices include TCP, UDP, HTTP, BOOTP, and Finger. For information about the HTTP server, see the"Using the Cisco Web Browser User Interface" chapter in this book.

The TCP small server provides the following minor services:

- Echo—Echoes back whatever you type. To test this service, issue the **telnet** *a.b.c.d* **echo** command from a remote host.
- Chargen—Generates a stream of ASCII data. To test this service, issue the **telnet** *a.b.c.d* **chargen** command from a remote host.
- Discard—Discards whatever you type. To test this service, issue the **telnet** *a.b.c.d* **discard** command from a remote host.
- Daytime—Returns system date and time if you have configured NTP or have set the date and time manually. To test this service, issue the **telnet** *a.b.c.d* **daytime** command from a remote host.

The User Datagram Protocol (UDP) small server provides the following minor services:

- Echo—Echoes the payload of the datagram you send.
- Chargen—Discards the datagram you send and responds with a 72 character string of ASCII characters terminated with a CR+LF (carriage return and line feed).
- Discard—Silently discards the datagram you send.

To enable TCP or UDP services, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# service tcp-small-servers`	Enables the minor TCP services echo, chargen, discard, and daytime.
`Router(config)# service udp-small-servers`	Enables the minor UDP services echo, chargen, and discard.

Because the minor services can be misused, these commands are disabled by default.

Caution Enabling minor services creates the potential for certain types of denial-of-service attacks, such as the UDP diagnostic port attack. Therefore, any network device that has UDP, TCP, BOOTP, or Finger services should be protected by a firewall or have the services disabled. For information on preventing UDP diagnostic port attacks, see the white paper titled *Defining Strategies to Protect Against UDP Diagnostic Port Denial of Service Attacks*, available on Cisco.com.

Note that the **no** form of the **service tcp-small-servers** and **service udp-small-servers** commands will appear in the configuration file to inform you when these basic services are disabled.

Controlling the BOOTP Server

You can enable or disable an async line Bootstrap Protocol (BOOTP) service on your routing device. This small server is enabled by default. Due to security considerations, this service should be disabled if you are not using it. To disable the BOOTP server on your platform, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no ip bootp server`	Disables the BOOTP server.

Because Dynamic Host Configuration Protocol (DHCP) is based on the Bootstrap Protocol, both of these service share the "well-known" UDP server port of 67 (per the internet standards and RFCs). For more information about DHCP configuration in Cisco IOS software, see the *Cisco IOS IP Configuration Guide*. For more information about BOOTP, see RFC 951. Interoperation between BOOTP and DHCP is defined in RFC 1534. DHCP is defined in RFC 2131.

Controlling the Finger Protocol

The Finger protocol allows users throughout the network to get a list of the users currently using a particular routing device. The information displayed includes the processes running on the system, the line number, connection name, idle time, and terminal location. This information is provided through the Cisco IOS software **show users** EXEC command.

To enable a Cisco device to respond to Finger (port 79) requests, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip finger`	Enables the Finger protocol service, which allows the system to respond to finger requests.

To configure the finger protocol to be compliant with RFC 1288, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip finger rfc-compliant`	Configures the device to wait for "Return" or "/W" input when processing Finger requests.

The **rfc-compliant** form of this command should not be configured for devices with more than 20 simultaneous users (see caveat CSCds92731 on Cisco.com for details). The difference between the two forms of this command is as follows: when the **ip finger** command is configured, the router will respond to a **telnet** *a.b.c.d* **finger** command from a remote host by immediately displaying the output of the **show users** command and then closing the connection. When the **ip finger rfc-compliant** command is configured, the router will wait for input before displaying anything. The remote user can then press the Return key to display the output of the **show users** command, or enter **/W** to display the output of the **show users wide** command. After this information is displayed, the connection is closed.

Hiding Telnet Addresses

You can hide addresses while attempting to establish a Telnet session. To configure the router to suppress Telnet addresses, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service hide-telnet-address`	Hides addresses while establishing a Telnet session.

The hide feature suppresses the display of the address and continues to display all other messages that normally would be displayed during a connection attempt, such as detailed error messages if the connection failed.

Use the **busy-message** line configuration command with the **service hide-telnet-address** command to customize the information displayed during Telnet connection attempts. If the connection attempt fails, the router suppresses the address and displays the message specified with the **busy-message** command.

Setting Time and Calendar Services

All Cisco routers provide an array of time-of-day services. These services allow the products to accurately keep track of the current time and date, to synchronize multiple devices to the same time, and to provide time services to other systems. The following sections describe the concepts and task associated with time and calendar services:

- Understanding Time Sources
- Configuring NTP

Understanding Time Sources

Most Cisco routers have two clocks: a battery-powered hardware clock (referenced in CLI commands as the "calendar") and a software clock (referenced in CLI commands as the "clock"). These two clocks are managed separately.

The primary source for time data on your system is the software clock. This clock runs from the moment the system starts up and keeps track of the current date and time. The software clock can be set from a number of sources and in turn can be used to distribute the current time through various mechanisms to other systems. When a router with a hardware clock is initialized or rebooted, the software clock is initially set based on the time in the hardware clock. The software clock can then be updated from the following sources:

- Network Time Protocol (NTP)
- Simple Network Time Protocol (SNTP)
- VINES Time Service
- Manual configuration (using the hardware clock)

Because the software clock can be dynamically updated it has the potential to be more accurate than the hardware clock.

The software clock can provide time to the following services:

- Access lists
- NTP
- VINES time service
- User **show** commands
- Logging and debugging messages
- The hardware clock

Note The software clock cannot provide time to the NTP or VINES Time Service if it was set using SNTP.

The software clock keeps track of time internally based on Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT). You can configure information about the local time zone and summer time (daylight savings time) so that the time is displayed correctly relative to the local time zone.

The software clock keeps track of whether the time is "authoritative" (that is, whether it has been set by a time source considered to be authoritative). If it is not authoritative, the time will be available only for display purposes and will not be redistributed.

Network Time Protocol

The Network Time Protocol (NTP) is a protocol designed to time-synchronize a network of machines. NTP runs over UDP, which in turn runs over IP. NTP Version 3 is documented in RFC 1305.

An NTP network usually gets its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server. NTP then distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two machines to the accuracy of within a millisecond of one another.

NTP uses the concept of a "stratum" to describe how many NTP "hops" away a machine is from an authoritative time source. A "stratum 1" time server typically has an authoritative time source (such as a radio or atomic clock, or a GPS time source) directly attached, a "stratum 2" time server receives its time via NTP from a "stratum 1" time server, and so on.

NTP avoids synchronizing to a machine whose time may not be accurate in two ways. First, NTP will never synchronize to a machine that is not in turn synchronized itself. Second, NTP will compare the time reported by several machines, and will not synchronize to a machine whose time is significantly different than the others, even if its stratum is lower. This strategy effectively builds a self-organizing tree of NTP servers.

The Cisco implementation of NTP does not support stratum 1 service; in other words, it is not possible to connect to a radio or atomic clock (for some specific platforms, however, you can connect a GPS time-source device). We recommend that time service for your network be derived from the public NTP servers available in the IP internet.

If the network is isolated from the internet, the Cisco implementation of NTP allows a machine to be configured so that it acts as though it is synchronized via NTP, when in fact it has determined the time using other means. Other machines can then synchronize to that machine via NTP.

A number of manufacturers include NTP software for their host systems, and a publicly available version for systems running UNIX and its various derivatives is also available. This software also allows UNIX-derivative servers to acquire the time directly from an atomic clock which would subsequently propagate time information along to Cisco routers.

The communications between machines running NTP (known as "associations") are usually statically configured; each machine is given the IP address of all machines with which it should form associations. Accurate timekeeping is made possible by exchanging NTP messages between each pair of machines with an association.

However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each machine can simply be configured to send or receive broadcast messages. However, the accuracy of timekeeping is marginally reduced because the information flow is one-way only.

The time kept on a machine is a critical resource, so we strongly recommend that you use the security features of NTP to avoid the accidental or malicious setting of incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism.

When multiple sources of time (VINES, hardware clock, manual configuration) are available, NTP is always considered to be more authoritative. NTP time overrides the time set by any other method.

Simple Network Time Protocol

Simple Network Time Protocol (SNTP) is a simplified, client-only version of NTP for use on Cisco 1003, Cisco 1004, Cisco 1005, Cisco 1600, Cisco 1720, and Cisco 1750 routers. SNTP can receive only the time from NTP servers; it cannot be used to provide time services to other systems.

SNTP typically provides time within 100 milliseconds of the accurate time, but it does not provide the complex filtering and statistical mechanisms of NTP. In addition, SNTP does not authenticate traffic, although you can configure extended access lists to provide some protection. An SNTP client is more vulnerable to misbehaving servers than an NTP client and should be used only in situations where strong authentication is not required.

You can configure SNTP to request and accept packets from configured servers or to accept NTP broadcast packets from any source. When multiple sources are sending NTP packets, the server with the best stratum is selected. (See the "Network Time Protocol" section for a description of strata.) If multiple servers are at the same stratum, a configured server is preferred over a broadcast server. If multiple servers pass both tests, the first one to send a time packet is selected. SNTP will choose a new server only if it stops receiving packets from the currently selected server, or if a better server (according to the above criteria) is discovered.

VINES Time Service

Time service is available when Banyan VINES is configured. This protocol is a standard part of VINES. The Cisco implementation allows the VINES time service to be used in two ways. First, if the system has learned the time from some other source, it can act as a VINES time server and provide time to other machines running VINES. Second, it can use the VINES time service to set the software clock if no other form of time service is available.

Note Support for Banyan VINES and XNS is removed from Cisco IOS software in Cisco IOS Release 12.2(13)T and later.

Hardware Clock

Some routers contain a battery-powered hardware clock that tracks the date and time across system restarts and power outages. The hardware clock is always used to initialize the software clock when the system is restarted.

Note Within the CLI command syntax, the hardware clock is referred to as the "system calendar."

If no other source is available, the hardware clock can be considered to be an authoritative source of time and be redistributed via NTP or VINES time service. If NTP is running, the hardware clock can be updated periodically from NTP, compensating for the inherent drift in the hardware clock.

Configuring NTP

NTP services are disabled on all interfaces by default. The following sections contain optional tasks that you can perform on your networking device:

- Configuring Poll-Based NTP Associations
- Configuring Broadcast-Based NTP Associations
- Configuring an NTP Access Group
- Configuring NTP Authentication
- Disabling NTP Services on a Specific Interface

- Configuring the Source IP Address for NTP Packets
- Configuring the System as an Authoritative NTP Server
- Updating the Hardware Clock
- Configuring an External Reference Clock

Configuring Poll-Based NTP Associations

Networking devices running NTP can be configured to operate in variety of association modes when synchronizing time with reference time sources. There are two ways that a networking device can obtain time information on a network: by polling host servers and by listening to NTP broadcasts. In this section, we will focus on the poll-based association modes. Broadcast-based NTP associations will be discussed in the next section.

The following are two most commonly used, poll-based association modes:

- Client mode
- Symmetric active mode

The *client* and the *symmetric active* modes should be used when NTP is required to provide a high level of time accuracy and reliability.

When a networking device is operating in the *client mode*, it polls its assigned time serving hosts for the current time. The networking device will then pick a host from all the polled time servers to synchronize with. Since the relationship that is established in this case is a client-host relationship, the host will not capture or use any time information sent by the local client device. This mode is most suited for file-server and workstation clients that are not required to provide any form of time synchronization to other local clients. Use the **ntp server** command to individually specify the time serving hosts that you want your networking device to consider synchronizing with and to set your networking device to operate in the *client mode*.

When a networking device is operating in the *symmetric active mode*, it polls its assigned time serving hosts for the current time and it responds to polls by its hosts. Since this is a peer-to-peer relationship, the host will also retain time-related information about the local networking device that it is communicating with. This mode should be used when there is a number of mutually redundant servers that are interconnected via diverse network paths. Most Stratum 1 and stratum 2 servers on the Internet today adopt this form of network setup. Use the **ntp peer** command to individually specify the time serving hosts that you want your networking device to consider synchronizing with and to set your networking device to operate in the *symmetric active mode*.

The specific mode that you should set each of your networking devices to depends primarily on the role that you want it to assume as a timekeeping device (server or client) and its proximity to a stratum 1 timekeeping server.

A networking device engages in polling when it is operating as a client or a host in the *client mode* or when it is acting as a peer in the *symmetric active mode*. Although polling does not usually exact a toll on memory and CPU resources such as bandwidth, an exceedingly large number of ongoing and simultaneous polls on a system can seriously impact the performance of a system or slow the performance of a given network. To avoid having an excessive number of ongoing polls on a network, you should limit the number of direct, peer-to-peer or client-to-server associations. Instead, you should consider using NTP broadcasts to propagate time information within a localized network.

Command	Purpose
`Router(config)# ntp peer ip-address [normal-sync] [version number] [key keyid] [source interface] [prefer]`	Forms a peer association with another system.
`Router(config)# ntp server ip-address [version number] [key keyid] [source interface] [prefer]`	Forms a server association with another system.

Note that only one end of an association needs to be configured; the other system will automatically establish the association.

Caution The **ntp clock-period** command is automatically generated to reflect the constantly changing *correction factor* when the **copy running-configuration startup-configuration** command is entered to save the configuration to NVRAM. Do not attempt to manually use the **ntp clock-period** command. Ensure that you remove this command line when copying configuration files to other devices.

For an example of configuring an NTP server-peer relationship, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Configuring Broadcast-Based NTP Associations

Broadcast-based NTP associations should be used when time accuracy and reliability requirements are modest and if your network is localized and has a large number of clients (more than 20). Broadcast-based NTP associations is also recommended for use on networks that have limited bandwidth, system memory, or CPU resources.

When a networking device is operating in the *broadcastclient mode*, it does not engage in any polling. Instead, it listens for NTP broadcast packets transmitted by broadcast time servers. Consequently, time accuracy can be marginally reduced since time information flows only one way.

Use the **ntp broadcast client** command to set your networking device to listen for NTP broadcast packets propagated through a network. In order for *broadcastclient mode* to work, the broadcast server and its clients must be located on the same subnet. The time server that is transmitting NTP broadcast packets will also have to be enabled on the interface of the given device using the **ntp broadcast** command.

To configure an interface to send NTP broadcasts, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ntp broadcast [version number]`	Configures the specified interface to send NTP broadcast packets.

To configure an interface to receive NTP broadcasts, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ntp broadcast client`	Configures the specified interface to receive NTP broadcast packets.

To manually set the estimated round-trip delay between the device and the NTP broadcast server, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ntp broadcastdelay microseconds`	Adjusts the estimated round-trip delay for NTP broadcasts.

Caution The **ntp clock-period** command is automatically generated to reflect the constantly changing *correction factor* when the **copy running-configuration startup-configuration** command is entered to save the configuration to NVRAM. Do not attempt to manually use the **ntp clock-period** command. Ensure that you remove this command line when copying configuration files to other devices.

For an example of configuring broadcast-based NTP associations, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Configuring an NTP Access Group

The access list-based restriction scheme allows you to grant or deny certain access privileges to an entire network, a subnet within a network, or a host within a subnet. To define an NTP access group, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ntp access-group {query-only | serve-only | serve | peer} access-list-number`	Creates an access group and applies a basic IP access list to it.

The access group options are scanned in the following order, from least restrictive to most restrictive:

1. **peer**—Allows time requests and NTP control queries and allows the system to synchronize itself to a system whose address passes the access list criteria.
2. **serve**—Allows time requests and NTP control queries, but does not allow the system to synchronize itself to a system whose address passes the access list criteria.
3. **serve-only**—Allows only time requests from a system whose address passes the access list criteria.
4. **query-only**—Allows only NTP control queries from a system whose address passes the access list criteria.

If the source IP address matches the access lists for more than one access type, the first type is granted. If no access groups are specified, all access types are granted to all systems. If any access groups are specified, only the specified access types will be granted.

For details on NTP control queries, see RFC 1305 (NTP version 3).

Configuring NTP Authentication

The encrypted NTP authentication scheme should be used when a reliable form of access control is required. Unlike the access list-based restriction scheme which is based on IP addresses, the encrypted authentication scheme uses authentication keys and an authentication process to determine if NTP synchronization packets sent by designated peers or servers on a local network are deemed as trusted before the time information that it carries along with it, is accepted.

The authentication process begins from the moment an NTP packet is created. Cryptographic checksum keys are generated using the MD5 Message Digest Algorithm and are embedded into the NTP synchronization packet that is sent to a receiving client. Once a packet is received by a client, its cryptographic checksum key is decrypted and checked against a list of trusted keys. If the packet contains a matching authenticator key, the timestamp information that is contained within it is accepted by the receiving client. NTP synchronization packets that do not contain a matching authenticator key will be ignored.

It is important to note that the encryption and decryption processes used in NTP authentication can be very CPU-intensive and can seriously degrade the accuracy of the time that is propagated within a network. If your network setup permits a more comprehensive model of access control, you should consider the use of the access list-based form of control instead.

After NTP authentication is properly configured, your networking device will only synchronize with and provide synchronization to trusted time sources. To enable your networking device to send and receive encrypted synchronization packets, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ntp authenticate`	Enables the NTP authentication feature.
Step 2	`Router(config)# ntp authentication-key` *number* `md5` *value*	Defines the authentication keys. Each key has a key number, a type, and a value. Currently the only key type supported is **md5**.
Step 3	`Router(config)# ntp trusted-key` *key-number*	Defines trusted authentication keys. If a key is trusted, this system will be ready to synchronize to a system that uses this key in its NTP packets.

Note In Cisco IOS software versions previous to release 12.0, the cryptotype value is displayed along with the ntp authentication key md5 value when the **show running-configuration** command is entered. Avoid copying and pasting the string cryptotype value that is displayed with the authentication-key as it will result in authentication failure.

Disabling NTP Services on a Specific Interface

NTP services are disabled on all interfaces by default.

NTP is enabled globally when any NTP commands are entered. you can selectively prevent NTP packets from being received through a specific interface by using the following command in interface configuration mode to turn off NTP on a given interface:

Command	Purpose
`Router(config-if)# ntp disable`	Disables NTP services on a specific interface.

Configuring the Source IP Address for NTP Packets

When the system sends an NTP packet, the source IP address is normally set to the address of the interface through which the NTP packet is sent. Use the following command in global configuration mode if you want to configure a specific interface from which the IP source address will be taken:

Command	Purpose
`Router(config)# ntp source` *interface*	Configures an interface from which the IP source address will be taken.

This interface will be used for the source address for all packets sent to all destinations. If a source address is to be used for a specific association, use the **source** parameter on the **ntp peer** or **ntp server** command shown earlier in this chapter.

Configuring the System as an Authoritative NTP Server

Use the following command in global configuration mode if you want the system to be an authoritative NTP server, even if the system is not synchronized to an outside time source:

Command	Purpose
`Router(config)# ntp master` [*stratum*]	Makes the system an authoritative NTP server.

Note Use the **ntp master** command with caution. It is very easy to override valid time sources using this command, especially if a low stratum number is configured. Configuring multiple machines in the same network with the **ntp master** command can cause instability in timekeeping if the machines do not agree on the time.

For an example of configuring an authoritative NTP server, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Updating the Hardware Clock

On devices that have hardware clocks (system calendars), you can configure the hardware clock to be periodically updated from the software clock. This is advisable for any device using NTP, because the time and date on the software clock (set using NTP) will be more accurate than the hardware clock, because the time setting on the hardware clock has the potential to drift slightly over time.

Use the following command in global configuration mode if a routing device is synchronized to an outside time source via NTP and you want the hardware clock to be synchronized to NTP time:

Command	Purpose
`Router(config)# ntp update-calendar`	Configures the system to update its hardware clock from the software clock at periodic intervals.

For an example of configuring NTP to update the calendar, see the section "Clock, Calendar, and NTP Configuration Examples" at the end of this chapter.

Configuring an External Reference Clock

Because Cisco's implementation of NTP does not support stratum 1 service, it is not possible to connect to a radio or atomic clock (for some specific platforms however, you can connect a GPS timesource device). However, certain Cisco devices allow you to connect a external GPS-based time-source device for the purposes of distributing a time signal to your network using NTP.

For example, the Trimble Palisade NTP Synchronization Kit can be connected to the auxiliary port of a Cisco 7200 Series router. Also, selected platforms support the use of GPS clocks from Symmetricom (formerly Telecom-Solutions). The refclock (reference clock) drivers provided on these platforms provides the ability to receive an RTS time-stamp signal on the auxiliary port of your routing device.

To configure a Trimble Palisade GPS product connected to the auxiliary port of a Cisco 7200 series router as the NTP reference clock, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line aux 0`	Enters line configuration mode for the auxiliary port 0.
Step 2	`Router(config-line)# ntp refclock trimble pps none stratum 1`	Enables the driver that allows the Trimble Palisade NTP Synchronization Kit to be used as the NTP reference clock source (Cisco 7200 series routers only).

To configure a Symmetricom GPS product connected to the auxiliary port of a supported router or switch as the NTP reference clock, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line aux 0`	Enters line configuration mode for the auxiliary port zero.
Step 2	`Router(config-line)# ntp refclock telecom-solutions pps cts stratum 1`	Enables the driver that allows the Symmetricom GPS product to be used as the NTP reference clock source.

To configure a PPS signal as the source for NTP synchronization, use the following form of the **ntp refclock** command in line configuration mode:

Command	Purpose
`Router(config-line)# ntp refclock pps {cts | ri} [inverted] [pps-offset number] [stratum number] [timestamp-offset number]`	Configures a PPS signal as the source for NTP synchronization.

Verifying the Status of the External Reference Clock

To verify the status of NTP components, use the following commands in privileged EXEC mode:

Command	Purpose
`Router#` **`show ntp associations`**	Displays the status of NTP associations, including the status of the GPS reference clock.
`Router#` **`show ntp status`**	Displays the status of NTP.
`Router#` **`debug ntp refclock`**	Allows advanced monitoring of reference clock activities for the purposes of debugging.

Configuring SNTP

SNTP generally is supported on those platforms that do not provide support for NTP, such as the Cisco 1000 series, 1600 series, and 1700 series platforms. SNTP is disabled by default. In order to enable SNTP, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)#` **`sntp server`** {*address* | *hostname*} [**`version`** *number*]	Configures SNTP to request NTP packets from an NTP server.
`Router(config)#` **`sntp broadcast client`**	Configures SNTP to accept NTP packets from any NTP broadcast server.

Enter the **sntp server** command once for each NTP server. The NTP servers must be configured to respond to the SNTP messages from the router.

If you enter both the **sntp server** command and the **sntp broadcast client** command, the router will accept time from a broadcast server but prefer time from a configured server, assuming that the strata are equal. To display information about SNTP, use the **show sntp** EXEC command.

Configuring VINES Time Service

Note Support for Banyan VINES and XNS has been removed from Cisco IOS software, beginning in Cisco IOS Release 12.2(13)T. The following VINES commands are not available in releases derived from 12.2(13)T, such as the 12.3 mainline release.

To distribute the system time and date to other devices on the network using VINES time services, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`vines time use-system`**	Distributes the system software clock time to other VINES systems.

To set the system time and date from received VINES time services, use the following command in global configuration mode:

Command	Purpose
`Router(config)# vines time set-system`	Sets the software clock system time from received VINES time services.

Configuring Time and Date Manually

If no other source of time is available, you can manually configure the current time and date after the system is restarted. The time will remain accurate until the next system restart. We recommend that you use manual configuration only as a last resort.

To set up time services, complete the tasks in the following sections as needed. If you have an outside source to which the router can synchronize, you do not need to manually set the software clock.

- Configuring the Time Zone
- Configuring Summer Time (Daylight Savings Time)
- Manually Setting the Software Clock
- Using the Hardware Clock

Configuring the Time Zone

To manually configure the time zone used by the Cisco IOS software, use the following command in global configuration mode :

Command	Purpose
`Router(config)# clock timezone` *zone hours-offset* [*minutes-offset*]	Sets the time zone. The *zone* argument is the name of the time zone (typically a standard acronym). The *hours-offset* argument is the number of hours the time zone is different from UTC. The *minutes-offset* argument is the number of minutes the time zone is different from UTC.

Tips The *minutes-offset* argument of the **clock timezone** command is available for those cases where a local time zone is a percentage of an hour different from UTC/GMT. For example, the time zone for some sections of Atlantic Canada (AST) is UTC -3.5. In this case, the necessary command would be **clock timezone AST -3 30**.

For an example of configuring the time zone, see the section "Clock, Calendar, and NTP Configuration Examples" at the end of this chapter.

Configuring Summer Time (Daylight Savings Time)

To configure summer time (daylight savings time) in areas where it starts and ends on a particular day of the week each year, use the following command in global configuration mode:

Command	Purpose
`Router(config)# clock summer-time zone recurring [week day month hh:mm week day month hh:mm [offset]]`	Configures a recurring summer time start and end date. The *offset* argument is used to indicate the number of minutes to add to the clock during summer time.

If summer time in your area does not follow this pattern, you can configure the exact date and time of the next summer time event by using one of the following commands in global configuration mode:

Command	Purpose

`Router(config)# clock summer-time zone date month date year hh:mm month date year hh:mm [offset]`
or
`Router(config)# clock summer-time zone date date month year hh:mm date month year hh:mm [offset]`
	Configures a specific summer time start and end date. The *offset* argument is used to indicate the number of minutes to add to the clock during summer time.

For an example of configuring summer time, see the section "Clock, Calendar, and NTP Configuration Examples" at the end of this chapter.

Manually Setting the Software Clock

Generally, if the system is synchronized by a valid outside timing mechanism, such as an NTP or VINES clock source, or if you have a router with a hardware clock, you need not set the software clock. Use this command if no other time sources are available. The time specified in this command is relative to the configured time zone. To set the software clock manually, use the following command in privileged EXEC mode:

Command	Purpose

`Router# clock set hh:mm:ss date month year`
or
`Router# clock set hh:mm:ss month date year`
	Sets the software clock.

Using the Hardware Clock

Most Cisco devices have a separate hardware-based clock in addition to the software-based clock. The hardware clock is a chip with a rechargeable backup battery that can retain the time and date information across reboots of the device.

To maintain the most accurate time update from an authoritative time source on the network, the software clock should receive time updates from an authoritative time on the network. The hardware clock should in turn be updated at regular intervals from the software clock while the system is running.

To customize the use of the hardware clock on your system, perform any of the following optional tasks:

- Setting the Hardware Clock
- Configuring the Router as a Network Time Source

Setting the Hardware Clock

The hardware clock (system calendar) maintains time separately from the software clock. The hardware clock continues to run when the system is restarted or when the power is turned off. Typically, the hardware clock needs to be manually set only once, when the system is first installed.

You should avoid setting the hardware clock manually if you have access to a reliable external time source. Time synchronization should instead be established using NTP.

If you do not have access to an external time source, use one of the forms of the following command in EXEC mode to set the hardware clock:

Command	Purpose

`Router> calendar set hh:mm:ss day month year`
or
`Router> calendar set hh:mm:ss month day year`
	Sets the hardware clock manually.

Configuring the Router as a Network Time Source

By default, the time maintained on the software clock is not considered to be authoritative and will not be redistributed with NTP or VINES Time Service. To classify the hardware clock as authoritative, use the following command in global configuration mode:

Command	Purpose
`Router(config)# clock calendar-valid`	Enables the router to act as a valid time source to which network peers can synchronize.

For an example of making the hardware clock authoritative, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Setting the Software Clock from the Hardware Clock

To set the software clock to the new hardware clock setting, use the following command in EXEC mode:

Command	Purpose
`Router# clock read-calendar`	Sets the software clock from the hardware clock.

Setting the Hardware Clock from the Software Clock

To update the hardware clock with a new software clock setting, use the following command in EXEC mode:

Command	Purpose
`Router# clock update-calendar`	Sets the hardware clock from the software clock.

Monitoring Time and Calendar Services

To monitor clock, calendar, and NTP EXEC services, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show calendar`	Displays the current hardware clock time.
`Router# show clock [detail]`	Displays the current software clock time.
`Router# show ntp associations [detail]`	Displays the status of NTP associations.
`Router# show ntp status`	Displays the status of NTP.
`Router# show sntp`	Displays information about SNTP (Cisco 1003, Cisco 1004, Cisco 1005, Cisco 1600, Cisco 1720, or Cisco 1750 routers only).

Configuring Time Ranges

Cisco IOS allows implementation of features based on the time of day. The **time-range** global configuration command defines specific times of the day and week, which then can be referenced by a function, so that those time restrictions are imposed on the function itself.

In Cisco IOS Release 12.2, IP and IPX extended access lists are the only functions that can use time ranges. The time range allows the network administrator to define when the **permit** or **deny** statements in the access list are in effect. Prior to the introduction of this feature, access list statements were always in effect once they were applied. Both named or numbered access lists can reference a time range.

Benefits of time ranges include the following:

- The network administrator has more control over permitting or denying a user access to resources. These resources could be an application (identified by an IP address/mask pair and a port number), policy routing, or an on-demand link (identified as interesting traffic to the dialer).
- Network administrators can set a time-based security policy, including the following:
  - Perimeter security using the Cisco IOS Firewall feature set or access lists
  - Data confidentiality with Cisco Encryption Technology or IPSec
- Policy-based routing and queueing functions are enhanced.
- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost effectively.
- Service providers can dynamically change a committed access rate (CAR) configuration to support the quality of service (QoS) service level agreements (SLAs) that are negotiated for certain times of day.
- Network administrators can control logging messages. Access list entries can log traffic at certain times of the day, but not constantly. Therefore, administrators can simply deny access without needing to analyze many logs generated during peak hours.

Defining a Time Range

Note The time range relies on the system's software clock. For the time range feature to work the way you intend, you need a reliable clock source. We recommend that you use NTP to synchronize the system's software clock.

To define a time range, use the following commands beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# time-range time-range-name`	Assigns a name to the time range to be configured and enters time-range configuration mode.

Step 2	`Router(config-time-range)# absolute [start time date] [end time date]`
	or
	`Router(config-time-range)# periodic days-of-the-week hh:mm to [days-of-the-week] hh:mm`
		Specifies when the time range will be in effect. Use some combination of these commands; multiple **periodic** statements are allowed; only one **absolute** statement is allowed.

Repeat these tasks if you have multiple items you want in effect at different times. For example, repeat the steps to include multiple **permit** or **deny** statements in an access list in effect at different times. For more information about these commands, refer to the "Basic System Management Commands" chapter in the "Cisco IOS System Management Commands" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Referencing the Time Range

In order for a time range to be applied, you must reference it by name in a feature that can implement time ranges. You can reference the time range in the following Cisco IOS software features:

- IP Extended Access Lists
  - Refer to the "Configuring IP Services" chapter of the Release 12.2 *Cisco IOS IP Configuration Guide* for instructions on creating an IP Extended Access List and referencing a time range.
- IPX Extended Access Lists
  - Refer to the "Configuring Novell IPX" chapter of the Release 12.2 *Cisco IOS AppleTalk and Novell IPX Configuration Guide* for instructions on creating an IPX Extended Access List and referencing a time range.

Delaying EXEC Startup

To delay the startup of the EXEC process on noisy lines until the line has been idle for 3 seconds, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service exec-wait`	Delays startup of the EXEC.

This command is useful on noisy modem lines or when a modem attached to the line is configured to ignore MNP or V.42 negotiations, and MNP or V.42 modems may be dialing in. In these cases, noise or MNP/V.42 packets might be interpreted as usernames and passwords, causing authentication failure before the user can type a username or password. The command is not useful on nonmodem lines or lines without some kind of login configured.

Handling an Idle Telnet Connection

To configure the Cisco IOS software to set the TCP window to zero (0) when the Telnet connection is idle, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service telnet-zero-idle`	Sets the TCP window to zero when the Telnet connection is idle.

Normally, data sent to noncurrent Telnet connections is accepted and discarded. When **service telnet-zero-idle** is enabled, if a session is suspended (that is, some other connection is made active), the TCP window is set to zero. This action prevents the remote host from sending any more data until the connection is resumed. Use this command when it is important that all messages sent by the host be seen by the users and the users are likely to use multiple sessions. Do not use this command if your host will eventually time out and log out a TCP user whose window is zero.

Setting the Interval for Load Data

You can change the period of time over which a set of data is used for computing load statistics. Decisions, such as for dial backup, depend on these statistics. If you decrease the load interval, the average statistics are computed over a shorter period of time and are more responsive to bursts of traffic.

To change the length of time for which a set of data is used to compute load statistics, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# load-interval seconds`	Sets the length of time for which data is used for load calculations.

Limiting the Number of TCP Transactions

When using a standard TCP implementation to send keystrokes between machines, TCP tends to send one packet for each keystroke typed, which can use up bandwidth and contribute to congestion on larger networks.

John Nagle's algorithm (RFC 896) helps alleviate the small-packet problem in TCP. The first character typed after connection establishment is sent in a single packet, but TCP holds any additional characters typed until the receiver acknowledges the previous packet. Then the second, larger packet is sent, and

additional typed characters are saved until the acknowledgment comes back. The effect is to accumulate characters into larger chunks, and pace their transmission to the network at a rate matching the round-trip time of the given connection. This method is usually preferable for all TCP-based traffic.

By default, the Nagle algorithm is not enabled. To enable the Nagle algorithm and thereby reduce the number of TCP transactions, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service nagle`	Enables the Nagle slow packet avoidance algorithm.

Configuring Switching and Scheduling Priorities

The normal operation of the network server allows the switching operations to use as much of the central processor as is required. If the network is running unusually heavy loads that do not allow the processor the time to handle the routing protocols, you may need to give priority to the system process scheduler. To do so, use the following command in global configuration mode:

Command	Purpose
`Router(config)# scheduler interval` *milliseconds*	Defines the maximum amount of time that can elapse without running the lowest-priority system processes.

To change the amount of time that the CPU spends on fast-switching and process-level operations on the Cisco 7200 series and Cisco 7500 series routers, use the following command in global configuration mode:

Command	Purpose
`Router(config)# scheduler allocate` *network-microseconds process-microseconds*	For the Cisco 7200 series and Cisco 7500 series routers, changes the default time the CPU spends on process tasks and fast switching.

Caution We recommend that you do not change the default values of the **scheduler allocate** command.

To configure the characteristics for a looping process, use the following command in global configuration mode:

Command	Purpose
`Router(config)# scheduler process-watchdog {hang | normal | reload | terminate}`	Configures an action for a looping process.

Modifying the System Buffer Size

You can adjust initial buffer pool settings and the limits at which temporary buffers are created and destroyed. To do so, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# buffers {small | middle | big | verybig | large | huge | type number} {permanent | max-free | min-free | initial} number`	Adjusts the system buffer sizes.
`Router(config)# buffers huge size number`	Dynamically resizes all huge buffers to the value that you supply.

Caution Normally you need not adjust these parameters; do so only after consulting with technical support personnel. Improper settings can adversely impact system performance.

During normal system operation, there are two sets of buffer pools: public and interface. They behave as follows:

- The buffers in the public pools grow and shrink based upon demand. Some public pools are temporary and are created and destroyed as needed. Other public pools are permanently allocated and cannot be destroyed. Public buffer pools are labeled as small, middle, big, large, very big, and huge.
- Interface pools are static—that is, they are all permanent. One interface pool exists for each interface. For example, a Cisco 4000 1E 4T configuration has one Ethernet buffer pool and four serial buffer pools. In the **buffers** EXEC command, the *type* and *number* arguments allow the user to tune the interface pools.

See the section "Buffer Modification Examples" at the end of this chapter for more information.

The server has one pool of queueing elements and six public pools of packet buffers of different sizes. For each pool, the server keeps count of the number of buffers outstanding, the number of buffers in the free list, and the maximum number of buffers allowed in the free list. To display statistics about the buffer pool on the system, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> show buffers`	Displays all public pool information.
`Router> show buffers address hex-addr`	Displays buffer information for an address.
`Router> show buffers all [dump | header | packet]`	Displays all public and interface pool information.
`Router> show buffers assigned [dump | header | packet]`	Displays a listing of all buffers in use.
`Router> show buffers failures [dump | header | packet]`	Displays buffer allocation failures.
`Router> show buffers free [dump | header | packet]`	Displays buffers available for use.
`Router> show buffers old [dump | header | packet]`	Displays buffers older than one minute.
`Router> show buffers input-interface interface-type identifier`	Displays buffer information for an input interface.
`Router> show buffers pool pool name`	Displays all interface pool information.

Basic System Management Examples

This section provides the following system management examples:

- System Configuration File Example
- Clock, Calendar, and NTP Configuration Examples
- Buffer Modification Examples

System Configuration File Example

The following is an example of a typical system configuration file:

```
! Define line password
line 0 4
 password secret
 login
!
! Define privileged-level password
enable-password Secret Word
!
! Define a system hostname
hostname TIP
! Specify a configuration file to load at system startup
boot host host1-confg 192.168.1.111
boot host host2-confg 192.168.1.111
! Specify the system image to boot at startup
boot system sys1-system 192.168.13.111
boot system sys2-system 192.168.1.111
boot system rom
!
! Enable SNMP
snmp-server community red
snmp-server enable traps snmp authentication
snmp-server host 192.168.1.27 public
snmp-server host 192.168.1.111 public
snmp-server host 192.168.2.63 public
!
! Define TACACS server hosts
tacacs-server host 192.168.1.27
tacacs-server host 192.168.13.33
tacacs-server host 192.168.1.33
!
! Define a message-of-the-day banner
banner motd ^C
The Information Place welcomes you

Please call 1-800-555-2222 for a login account, or enter
your password at the prompt.
^C
```

Clock, Calendar, and NTP Configuration Examples

In the following example, a router with a hardware clock has server associations with two other systems, sends broadcast NTP packets, periodically updates the hardware clock, and redistributes time into VINES:

```
clock timezone PST -8
```

```
clock summer-time PDT recurring
ntp update-calendar
ntp server 192.168.13.57
ntp server 192.168.11.58
interface Ethernet 0/0
 ntp broadcast
vines time use-system
```

In the following example, a router with a hardware clock has no outside time source, so it uses the hardware clock as an authoritative time source and distributes the time via NTP broadcast packets:

```
clock timezone MET 2
clock calendar-valid
ntp master
interface fddi 0/0
 ntp broadcast
```

Buffer Modification Examples

The following example instructs the system to keep at least 50 small buffers free:

```
Router> buffers small min-free 50
```

The following example instructs the system to keep no more than 200 middle buffers free:

```
Router> buffers middle max-free 200
```

The following example instructs the system to create one large temporary extra buffer, just after a reload:

```
Router> buffers large initial 1
```

The following example instructs the system to create one permanent huge buffer:

```
Router> buffers huge permanent 1
```