CHAPTER 5

Managing the System

This chapter describes the basic tasks that can be completed to manage the general system (nonprotocol-specific) features. Our system management features are supported via the Simple Network Management Protocol (SNMP). This chapter describes the tasks needed to configure SNMP support on the communication server. A part of SNMP is the Management Information Base (MIB). MIBs provide variables that can be set or read to change parameters or provide information on network devices and interfaces. Cisco supports several MIBs, including the Internet standard MIB II, and also provides its own Cisco MIB. For information on the Cisco MIB, see the *Cisco Management Information Base (MIB) User Quick Reference*.

For a list of recommended books on network management, refer to the appropriate appendix in the *Communication Server Command Reference* publication.

For a complete description of the commands mentioned in this chapter, refer to of the *Communication Server Command Reference* publication.

System Management Task List

In general, you will be performing the following types of system or network management:

- Configuration Management
- Security Management
- Fault Management
- System Performance Management
- Accounting Management

The following sections describe how to perform these tasks.

Configuration Management

The configuration of network devices determines the network's behavior. To manage device configurations, you need to list and compare configuration files on running devices, store configuration files on network servers for shared access, and perform software installations and upgrades.

Note Tasks related to managing configuration files are covered in detail in the "Loading System Images, Microcode Images, and Configuration Files" chapter of this manual. See also the *Communication Server Command Reference* publication for a complete description of the commands used.

You can complete any of the following tasks to perform configuration management functions:

- Set the communication server name
- Set the communication server time services
- Configure SNMP support
- Set up security features
- Enable the Challenge Handshake Authentication Protocol (CHAP))

The following sections summarize these tasks.

Set the Communication Server Name

One of the first basic tasks is to name your communication server. The name of the communication server is considered the host name and is the name that is displayed by the system prompt. If no name is configured, the system default communication server name is *cs*. You can set the communication server to the name of your choice while in global configuration mode as follows:

Task	Command
Set the host name.	**hostname** *name*

Set the Communication Server Time Services

Communications server products provide an array of time-of-day services. These services allow the products to keep track of the current time and date to a high degree of accuracy, to synchronize multiple products to the same time, and to provide time services to other systems.

The heart of the time service is the system clock. This clock, which is always running from the moment the system starts up, keeps track of the current date and time. The system clock can be set from a number of sources, and can in turn be used to distribute the current time through various mechanisms to other systems. When the system is initialized, the system clock is set to midnight on March 1, 1993. The system clock can then be set using the following methods:

- Network Time Protocol
- Manual configuration

The system clock can provide time to the following services:

- Network Time Protocol
- User **show** commands
- Logging and debugging messages

The system clock internally keeps track of time 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 system clock keeps track of whether the time is “authoritative” or not (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.

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 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 is then used to distribute this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two machines to 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 has a radio or atomic clock directly attached, a “stratum 2” time server receives its time via NTP from a “stratum 1” time server, and so on. A machine running NTP will automatically choose as its time source the machine with the lowest stratum number that it is configured to communicate with via NTP. This strategy effectively builds a self-organizing tree of NTP speakers.

NTP is careful to avoid synchronizing to a machine whose time may not be accurate. It avoids doing so in two ways. First of all, NTP will never synchronize to a machine that is not in turn synchronized itself. Secondly, 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.

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 may 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 it is strongly recommended that the security features of NTP be used 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.

Our implementation of NTP does not support stratum 1 service; in other words, it is not possible to connect a radio or atomic clock to this communication server. It is recommended 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, our 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 will then synchronize to that machine via NTP.

NTP software for host systems is included by a number of manufacturers, and a publicly available version for systems running UNIX and its various derivatives is also available. This software allows host systems to be time-synchronized as well.

If no other source of time is available, the current time and date can be manually configured after the system is restarted. The time will remain accurate until the next system restart. It is recommended that manual configuration be used only as a last resort.

To set up time services manually, complete the following tasks. If you have an outside source to which the communication server can synchronize, you do not need to manually set the system clock.

- Configure the time zone
- Configure summer time (daylight savings time, if applicable)
- Set the system clock (if no other time source is available)

- Configure NTP authentication
- Configure NTP associations
- Configure NTP broadcast service
- Configure NTP Access restrictions
- Configure the source IP address for NTP packets
- Configure the system as an authoritative NTP server
- Monitor the time services

Configure the Time Zone

Complete the following task in global configuration mode to configure the time zone used by the communication server:

Task	Command
Set the communication server time zone.	**clock timezone** *name hours* [*minutes*]

Configure Summer 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 form of the command in global configuration mode:

Task	Command
Configure summer time using the same day of the week each year.	**clock summer-time** *name* **recurring** [*week day month hh:mm week day month hh:mm*] [*offset*]

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

Task	Command
Configure summer time using exact date and time.	**clock summer-time** *name* **date** *month day year hh:mm month day year hh:mm* [*offset*] or **clock summer-time** *name* **date** *day month year hh:mm day month year hh:mm* [*offset*]

Set the System Clock

If you have an outside source on the network that provides time services (such as an NTP server service), you do not need to manually set the system clock.

However, if you have do not have any time service source, complete one of the following tasks in EXEC mode to set the system clock:

Task	Command
Set the system clock.	**clock set** *hh:mm:ss day month year* or **clock set** *hh:mm:ss month day year*

If you want to establish very accurate time synchronization among the machines in your network, configure the Network Time Protocol (NTP). There are a number of subtasks to perform:

- Configure NTP authentication
- Configure NTP associations
- Configure NTP broadcast service
- Configure the source IP address for NTP packets
- Configure the system as an authoritative NTP server

Configure NTP Authentication

If you want to authenticate the associations with other systems for security purposes, perform the tasks that follow. The first task enables the NTP authentication feature. The second task defines each of the authentication keys. Each key has a key number, a type, and a value. Currently the only key type supported is "md5." Third, a list of "trusted" authentication keys is defined. If a key is trusted, then this system will be willing to synchronize to a system that uses this key in its NTP packets.

To configure NTP authentication, perform the following tasks in global configuration mode:

Task		Command
Step 1	Enable the NTP authentication feature.	**ntp authenticate**
Step 2	Define the authentication keys.	**ntp authentication-key** *number* **md5** *value*
Step 3	Define trusted authentication keys.	**ntp trusted-key** *number*

Configure NTP Associations

If you want to form an NTP association with another system, perform one of the tasks in global configuration mode:

Task	Command
Form an NTP association with another system: Set a peer to be synchronized by peer	**ntp peer** *ip-address* [**version** *number*] [**key** *keyid*] [**source** *interface*] [**prefer**]
or	or
Set as communication server to be synchronized by time server.	**ntp server** *ip-address* [**version** *number*] [**key** *keyid*] [**source** *interface*] [**prefer**]

The association can be a peer association (meaning that this communication server is willing to either synchronize to the other system or to allow the other system to synchronize to it), or it can be a server association (meaning that this communication server will only synchronize to the other system, and not the other way around). Note that only one end of an association needs to be configured; the other system will automatically establish the association.

Configure NTP Broadcast Service

The system can either send broadcast packets or listen to them on an interface-by-interface basis. The estimated round-trip delay for broadcast packets can also be configured. Perform one or more of these tasks in global configuration mode if you want to use NTP's broadcast feature:

Task	Command
Send NTP broadcast packets.	**ntp broadcast** [**version** *number*][**key** *nn*]
Receive NTP broadcast packets.	**ntp broadcast client**
Adjust estimated delay.	**ntp broadcastdelay** *microseconds*

Configure NTP Access Restrictions

You can control NTP access on two levels by completing the following tasks:

- Create an access group and assign a basic IP access list to it
- Disable NTP services on specific interfaces

Create an Access Group and Assign a Basic IP Access List to It

To control access to NTP services, you can create an NTP access group and apply a basic IP access list to it. To do so, perform the following task in global configuration mode:

Task	Command
Create an access group and apply a basic IP access list to it.	**ntp access-group** {**query-only** | **serve-only** | **serve** | **peer**} *number*

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. Server —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).

Disable NTP Services on a Specific Interface

NTP services are enabled on all interfaces by default. You can disable NTP packets from being received through an interface by performing the following task in interface configuration mode:

Task	Command
Disable NTP services on a specific interface.	**ntp disable**

Configure 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. Perform the following task in global configuration mode if you want to configure a specific interface from which the IP source address will be taken:

Task	Command
Configure an interface from which the IP source address will be taken.	**ntp source** *interface*

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.

Configure the System as an Authoritative NTP Server

Perform the following task 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:

Task	Command
Make the system an authoritative NTP server.	**ntp master** [*stratum*]

Monitor Time Services

You can monitor clock, calendar, and NTP EXEC services by completing the following tasks in EXEC mode:

Task	Command
Display the current system clock time.	**show clock** [**detail**]
List NTP statistics.	**show ntp associations** [**detail**] or **show ntp status**

Configure Simple Network Management Protocol (SNMP) Support

The Simple Network Management Protocol (SNMP) provides a way for network management client and server applications to communicate. It does this by providing a message format for sending information between an SNMP manager and an SNMP agent.

The SNMP agent contains Management Information Base (MIB) variables that the SNMP manager can request or change. The SNMP agent can also send traps, or messages alerting the SNMP manager to a condition on the network. Traps can indicate improper user authentication, restarts, link status (up or down), closing of a TCP connection, or loss of connection to a neighbor communication server.

Our implementation of SNMP supports all MIB II variables (as described in RFC 1213 and SNMP traps (as described in RFC 1215). Cisco also supports the definition of management information described in RFCs 1155, 1157, and 1213, and supports some or all variables in the MIBs described in the following RFCs: 1156, 1212, 1231, 1285, 1286, 1315, 1381, and 1382.

Cisco also provides its own MIB with every system. With the current software release, the Cisco MIB provides a new chassis MIB variable that enables the SNMP manager to gather data on system card descriptions, serial numbers, hardware and software revision levels, and slot locations.

See the *Cisco Management Information Base (MIB) User Quick Reference* for a detailed description of each Cisco MIB variable and SNMP trap.

You can perform the following tasks to configure SNMP support on the communication server:

- Enable SNMP and define access control
- Define SNMP trap operations
- Define the maximum SNMP packet size
- Enable the SNMP server shutdown mechanism
- Establish the contact, location, and serial number for the SNMP server
- Disable the SNMP server
- Monitor SNMP status

These tasks are described in the following sections.

Enable SNMP and Define Access Control

You can enable SNMP server operation and specify which hosts can send requests to the communication server by performing the following tasks in global configuration mode:

Task		Command
Step 1	Enable the SNMP server and define the community access string.	**snmp-server community** [*string* [**RO** | **RW**] [*list*]]
Step 2	Specify the access list that determines which hosts can send requests to the network server.	**snmp-server access-list** *list*

Define SNMP Trap Operations

The SNMP trap operations allow a system administrator to configure the communication server to send information to a network management application when a particular event occurs. You can specify the following features for SNMP server trap operations:

- Source interface
- Recipient
- Trap operation authentication
- Retransmission interval
- Message (packet) queue length for each trap host

Perform the following tasks in global configuration mode, as needed, to define traps for your system configuration:

Task	Command
Specify the source interface (and hence IP address) of the trap message.	**snmp-server trap-source** *interface-type interface-number*
Specify the recipient of the trap message.	**snmp-server host** *address community-string* [**snmp**] [**tty**]

Task	Command
Establish trap message authentication.	**snmp-server trap-authentication**
Define how often to resend trap messages on the retransmission queue.	**snmp-server trap-timeout** *seconds*
Establish the message queue length for each trap host.	**snmp-server queue-length** *length*

Define the Maximum SNMP Packet Size

You can set the maximum packet size permitted when the SNMP server is receiving a request or generating a reply. To do so, perform the following task in global configuration mode:

Task	Command
Establish the maximum packet size.	**snmp-server packetsize** *bytes*

Enable the SNMP Server Shutdown Mechanism

Using SNMP packets, a network management tool can send messages to users on virtual terminals and the console. This facility operates in a similar fashion to the EXEC **send** command; however, the SNMP request that causes the message to be issued to the users also specifies the action to be taken after the message is delivered. One possible action is a shutdown request. Because the ability to cause a reload from the network is a powerful feature, you can perform the following task in global configuraiton mode to protect it is protected by this global configuration command:

Task	Command
Requests that a shutdown occur after a message is delivered from a network management tool.	**snmp-server system-shutdown**

To understand how to use this feature with SNMP requests, read the document *mib.txt* available by anonymous ftp from *ftp.cisco.com.*

Establish the Contact, Location, and Serial Number of the SNMP Server

You can set the system contact, location, and serial number of the SNMP server so that these descriptions can be accessed via the configuration file. To do so, perform the following tasks in global configuration mode:

Task	Command
Set the system contact string.	**snmp-server contact** *text*
Set the system location string.	**snmp-server location** *text*
Set the system serial number.	**snmp-server chassis-id** *text*

Disable the SNMP Server

Once the SNMP server has been enabled with the **snmp-server community** command, you can specifically disable it by performing the following task in global configuration mode:

Task	Command
Disable SNMP server operation.	**no snmp-server**

Monitor SNMP Status

To monitor SNMP input and output statistics, including the number of illegal community string entries, errors, requested variables, and so on, complete the following task in EXEC mode:

Task	Command
Monitor SNMP status.	**show snmp**

Security Management

To manage security on the network, you need to restrict access to the system. You can do so on several different levels. You can assign passwords (and encrypt them) to restrict access to communication server terminal lines, login connections, or privileged EXEC mode. You can establish Terminal Access Controller Access Control System (TACACS) protection for network servers that have shared access, and you can create access lists to filter traffic to and from specific destinations. You can also restrict login connections to specific users with user authentication support.

To set up security features, you need to identify sensitive information, find the network access points to that information, secure these access points, and maintain the secure access points.

This section describes the following tasks used to control access to the system:

- Establish password protection
- Disable password protection
- Recover a lost password
- Create access lists and apply them to specific interfaces (or lines in the case of IP access lists) to filter access at the packet level
- Establish Terminal Access Controller Access Control System (TACACS) protection schemes
- Establish username authentication
- Enable Challenge Handshake Authentication Protocol (CHAP)
- Enable Password Authentication Protocol (PAP)

Other chapters in this guide provide information on protocol-specific security features.

Establish Password Protection

Complete the following tasks to establish password protection:

- Protect access to terminals on individual lines
- Protect access to privileged EXEC mode, and thus, to the system configuration file
- Encrypt passwords so that they cannot be read in the configuration file with the **show configuration** EXEC command or with a protocol analyzer

Protect Access to Terminal Lines

You can provide access control on a terminal line by entering the password and establishing password checking. To do so, perform the following tasks in line configuration mode:

Task		Command
Step 1	Assign a password to a terminal or other device on a line.	**password** *text*
Step 2	Enable password checking.	**login**

The **login** and **password** commands are documented further in the"Line Configuration and Terminal Setting Commands" chapter in the *Communication Server Command Reference*.

The password checker is case sensitive. The password *Secret* is different than the password *secret*, for example, and the password *two words* is an acceptable password.

Protect Access to Privileged EXEC Mode

You can control access to the system by setting a password that must be entered to gain access to the privileged EXEC mode, and therefore to the system configuration. Perform the following task in global configuration mode:

Task	Command
Establish a password for the privileged EXEC mode.	**enable password** *password*

Encrypt the Passwords

You can increase access security to your communication server by configuring the communication server to encrypt passwords, because protocol analyzers can examine packets. Encryption prevents the password from being visible in the configuration file.

Configure the communication server to encrypt passwords by performing the following task in global configuration mode:

Task	Command
Encrypt a password.	**service password-encryption**

It is not possible to recover a lost encrypted password.

Disable Password Protection

You can disable line password verification by disabling password checking. To do so, perform the following task in line configuration mode:

Task	Command
Disable password checking or allow access to a line without password verification.	**no login**

Recover a Lost Password

If your server has the nonvolatile memory option, you can accidentally lock yourself out if you enable password checking on the console terminal line and then forget the line password.

ASM-CS

To recover a lost password on the ASM-CS, force the server into factory diagnostic mode and then follow these steps:

Step 1 You will be asked if you want to set the manufacturers' addresses. Respond by typing **Yes**. You will then see the following prompt:

```
TEST-SYSTEM>
```

Step 2 Type the **enable** command to get the privileged prompt:

```
TEST-SYSTEM> enable
```

Step 3 Type the **show configuration** command to review the system configuration and find the password. Do not change anything in the factory diagnostic mode.

```
TEST-SYSTEM> show configuration
```

Step 4 To resume normal operation, restart the server and/or reset the configuration register.

Step 5 Log into the server with the password that was shown in the configuration file.

Note All debugging capabilities are turned on during diagnostic mode.

500-CS

Unplug the 500-CS. Hold down the Reset button while plugging the power cord back in., and keep holding it down for at least three seconds after the "PWR" (power) LED comes on.

Create and Apply Access Lists

This section summarizes the types of access lists that you can create to control access at the packet level. This summary provides a brief description of access lists in alphabetical order by protocol.

You can control access to lines and interfaces by completing the following tasks:

Step 1 Create an access list based upon particular restrictions.

Step 2 Assign an access list to a specific interface.

Note See the appropriate chapter in this guide for detailed task information on each protocol-specific access list. To control SNMP access, see "Enable SNMP and Define Access Control" earlier in this chapter.

Access List Ranges

Table 1-1 provides a summary of access list ranges.

Table 1-1 Summary of Numerical Access List Ranges

Protocol	Range
IP	1–99
Extended IP	100–199
Ethernet type code	200–299
Ethernet address	700–799
Novell	800–899
Extended Novell	900–999
Novell SAP	1000–1099

Dial-on-Demand Routing

You can assign access lists to DDR interfaces by performing the following tasks:

Task		Command
Step 1	Specify the number of the dialer group to which the specific interface belongs.	**dialer-group** *group-number*
Step 2	Either associate an IP access list number with the dialer group, or associate a specific protocol with the dialer group	**dialer-list** *dialer-group* **list** *list-number* **dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny**}

For more information about dial-on-demand (DDR) routing see the chapter“Configuring Dial-on-Demand Routing.”

IP Access Lists

You can create the following types of IP access lists:

- Standard
- Extended

Standard

You can create a standard IP access list based on the address mask by completing the following task:

Task	Command
Set address masks restrictions.	**access-list** *1–99* {**permit** | **deny**} *source source-mask*

You can assign a standard IP access list to a specified interface or line by performing these tasks:

Task	Command
Assign the access list restrictions to either outbound or inbound interfaces.	**ip access-group** *1–99* {**out** | **in**}
Restrict incoming and outgoing connections between a particular virtual terminal line (into a Cisco device) and the addresses in an access list.	**access-class** *list* {**in** | **out**}

Extended

You can assign extended IP access lists to a specified interface or line by performing these tasks:

Task	Command
Define an extended access list.	**access-list** *100–199* {**permit** | **deny ip** | **tcp** | **udp** | **icmp** | **igrp**} *source source-mask dest dest-mask* [**lt** | **gt** | **eq** | **neq** *dest-port*][**established**]
Apply an access list to an interface.	**ip access-group** *100–199* {**out** | **in**}

For more information about IP, see the IP routing chapter.

NovelI IPX

You can create several types of IPX access lists to restrict access based on source and destination address and mask, IPX protocol, name, and byte pattern. Perform the appropriate tasks from the following list to do so:

Task	Command
Create a standard IPX access list.	**access-list** *number* {**deny** | **permit**} *source-network*[.*source-node*[*source-node-mask*]] [*destination-network*[.*destination-node* [*destination-node-mask*]]]
Create an extended IPX access list.	**access-list** *access-list-number* {**deny** | **permit**} *protocol* [*source-network*[.*source-node* [[*source-network-mask.*]*source-node-mask*]] *source-socket* [*destination-network* [.*destination-node* [[*destination-network-mask.*]*destination-node-mask*]] *destination-socket*]]
Create an IPX access list for SAP filters.	**access-list** *access-list-number* {**deny** | **permit**} *network*[.*node*] [*network.node-mask*] [*service-type* [*server-name*]]
Create an access list for filtering IPX NetBIOS packets by node name.	**netbios access-list host** *name* {**deny** | **permit**} *string*
Create an access list for filtering IPX NetBIOS packets by arbitrary byte pattern.	**netbios access-list bytes** *name* {**deny** | **permit**} *offset-byte-pattern*

You can create several types of IPX filters by applying access lists to individual interfaces by performing the following tasks:

Task	Command
Apply a generic filter to an interface.	**ipx access-group** *800–999*
Control which networks are added to the routing table when IPX routing updates are received.	**ipx input-network-filter** *800–999*
Control which networks are advertised in routing updates sent out by the communication server.	**ipx output-network-filter** *800–999*
Control the communication servers from which routing updates are accepted.	**ipx router filter** *800–999*
Filter incoming service advertisements.	**ipx input-sap-filter** *1000–1099*
Filter outgoing service advertisements.	**ipx output-sap-filter** *1000–1099*

Task	Command
Filter service advertisements received from a particular communication server.	**ipx communication server-sap-filter** *1000–1099*
Filter the list of servers in GNS response messages.	**ipx output-gns-filter** *1000–1099*
Filter incoming packets by node name.	**ipx netbios input-access-filter host** *name*
Filter incoming packets by byte pattern.	**ipx netbios input-access-filter bytes** *name*
Filter outgoing packets by node name.	**ipx netbios output-access-filter host** *name*
Filter outgoing packets by byte pattern.	**ipx netbios output-access-filter bytes** *name*
Apply a broadcast message filter to an interface.	**ipx helper-list** *800–999*

For more information about configuring Novell IPX, see the chapter“Configuring Novell IPX.”

Establish Terminal Access Control

You can configure the communication server to use a special TCP/IP protocol called Terminal Access Controller Access Control System (TACACS). TACACS provides an additional level of control over servers running on a timesharing system. The Defense Data Network (DDN) developed TACACS to control access to its TAC servers; Cisco patterned its TACACS support after the DDN application.

You can establish TACACS password protection on both user and privileged levels of the system EXEC. The TACACS security program allows you to set these features:

- Set TACACS user ID and password checking at the user level
- Disable password checking at the user level and guarantee “last resort” login
- Set optional password verification
- Set TACACS user ID and password checking at the privileged level
- Disable checking at the privileged level and guarantee “last resort” login
- Set notification messages when the user makes a connection, accesses the privileged EXEC level, or logs out
- Set authentication when the user makes a connection or accesses the privileged EXEC level
- Set separate TACACS server host name with custom retransmit and timeout intervals
- Establish a limit on login attempts
- Enable an extended TACACS mode that supports system accounting and logging applications

The following sections describe these tasks.

Note Additional protection using TCP/IP access lists might also be required. Refer to the TCP/IP chapter for more information.

Set TACACS Password Protection at the User Level

You can enable password checking at login by performing the following task in line configuration mode:

Task	Command
Set the TACACS-style user ID and password checking mechanism.	**login tacacs**

The **login tacacs** command is documented further in the chapter"Line Configuration and Terminal Setting Commands," in the *Communication Server Command Reference*.

Disable Password Checking at the User Level

You can disable password checking at the user level and guarantee forced login by performing the following task in global configuration mode:

Task	Command
Set "last resort" options for logins.	**tacacs-server last-resort** {**password** | **succeed**}

Set Optional Password Verification

You can specify that the first TACACS request to a TACACS server be made without password verification, perform the following task in global configuration mode:

Task	Command
Set TACACS password as optional.	t**acacs-server optional-passwords**

Set TACACS Password Protection at the Privileged Level

You can enable the use of TACACS to determine whether a user can access the privileged command level by performing the following task in global configuration mode:

Task	Command
Set the TACACS-style user ID and password checking mechanism at the privileged command level.	**enable use-tacacs**

Disable Password Checking at the Privileged Level

You can specify what happens if the TACACS server does not respond by performing the following task in global configuration mode:

Task	Command
Set "last resort" options for logins at the privileged prompt.	**enable last-resort** {**succeed** | **password**}

Set Notification of User Actions

You can cause a message to be transmitted to the TACACS server with retransmission being performed by a background process by performing the following task in global configuration mode:

Task	Command
Set server notification of user actions.	**tacacs-server notify** {**connection** | **enable** | **logout**}

Set Authentication of User Actions

For a TCP connection, you can specify that if a user tries to make a connection, the communication server requires a response from the network or communication server indicating whether the user can make the connection. You can also specify that the communication server should perform authentication even when a user is not logged in.

For a SLIP or PPP session, you can specify that if a user tries to start a session, the communication server requires a response from the network or communication server indicating whether the user can start the session. You can specify that the communication server should perform authentication even when a user is not logged in. You can also request that the communication server install access lists.

For use of the **enable** command, you can specify that if a user issues the **enable** command, the communication server must respond indicating whether the user can give the command.

To configure any of these scenarios, perform the following task in global configuration mode:

Task	Command
Set server authentication of user actions.	**tacacs-server authenticate** {**connection** [**always**] | **enable** | **slip** [**always**] [**access-lists**]}

The **tacacs-server authenticate** command is only available when you have set up an extended TACACS server using the latest Cisco extended TACACS server software, available using FTP (see the README file in the *ftp.cisco.com* directory).

Establish the TACACS Server Host and Response Times

You can specify the names of the IP host or hosts maintaining a TACACS server. The software searches for the hosts in the order specified, so this feature can be useful for setting up a list of preferred servers.

You can also modify the number of times the system software searches the list of TACACS servers and the interval it waits for a reply.

Perform the following tasks in global configuration mode, as needed, for your system configuration:

Task	Command
Specify a TACACS host.	**tacacs-server host** *name*
Specify the number of times the server will search the list of TACACS server hosts before giving up.	**tacacs-server retransmit** *retries*
Set the interval the server waits for a TACACS server host to reply.	**tacacs-server timeout** *seconds*

Set Limits on Login Attempts

You can set controls on the number of login attempts that can be made on a line set up for TACACS by performing the following task in global configuration mode:

Task	Command
Control the number of login attempts that can be made on a line set for TACACS verification.	**tacacs-server attempts** *count*

Enable the Extended TACACS Mode

Extended TACACS mode provides information about the terminal requests to help set up UNIX auditing trails and accounting files for tracking use of protocol translators, communication servers, and communication servers. The information includes responses from these network devices and validation of user requests.

An unsupported, extended TACACS server is available from Cisco Systems using **ftp** for UNIX users who want to create the auditing programs (see the README file in the *ftp.cisco.com* directory).

Extended TACACS differs from standard TACACS in that standard TACACS provides only username and password information.

To enable extended TACACS mode, perform the following task in global configuration mode:

Task	Command
Enable an extended TACACS mode.	**tacacs-server extended**

Establish Username Authentication

You can create a username-based authentication system. This can be useful for the following reasons:

- To provide a TACACS-like username and encrypted password authentication system for those networks that cannot support TACACS.
- To provide special case logins, for example, access list verification, no password verification, autocommand execution at login, and "no escape" situation.

Perform the following tasks in global configuration mode, as needed for your system configuration:

Task	Command
Establish username authentication with encrypted passwords or by access list.	**username** *name* [**no password** | **password** *encryptiontype password*] **username** *name* [**access-class** *number*]
Specify a command to automatically execute.	**username** *name* [**autocommand** *command*]
Set a "no escape" login environment.	**username** *name* [**noescape**] [**nohangup**]

The keyword **noescape** prevents users from using escape characters on the hosts to which they are connected.

Enable Challenge Handshake Authentication Protocol (CHAP)

Access control using Challenge Handshake Authentication Protocol (CHAP) is available on all serial interfaces configured for PPP encapsulation. The authentication feature reduces the risk of security violations on your communication server.

Note CHAP is supported only on lines using PPP encapsulation.

When CHAP is enabled, a remote device (a PC, workstation, or communication server) attempting to connect to the local communication server is requested, or "challenged," to respond. The challenge consists of a random number and the host name of the local communication server. This challenge is transmitted to the remote device. The required response is an encrypted version of a secret password, or "secret," plus the host name of the remote device. The remote device verifies the secret by looking up the host name that was received in the challenge. When the local communication server receives the challenge response, it verifies the secret by looking up the name of the remote device given in the response. The secret passwords must be identical on the remote device and the local communication server.

By transmitting this encrypted response, the secret is never transmitted, thus preventing other devices from stealing it and gaining illegal access to the system. Without the proper response, the remote device cannot connect to the local communication server.

CHAP transactions occur only when a link is established. The local communication server does not request a password during the rest of the call. (The local communication server can, however, respond to such requests from other devices during a call.)

To use CHAP, perform the following task in interface configuration mode:

Task	Command
Enable CHAP on the interface.	**ppp authentication chap**

CHAP is specified in RFC 1334. It is an additional authentication phase of the PPP Link Control Protocol.

Once you have enabled CHAP, the local communication server requires a password from remote devices. If the remote device does not support CHAP, no traffic is passed to that device.

For more information about CHAP and PPP see the"Configuring Interfaces" chapter and the chapter"Configuring SLIP and PPP."

Enable Password Authentication Protocol (PAP)

Access control using the Password Authentication Protocol (PAP) is available on all serial interfaces. The authentication feature reduces the risk of security violations on your communication server.

Note PAP is supported only on lines that use PPP encapsulation.

To use PAP, perform the following task in interface configuration mode:

Task	Command
Enable PAP on the interface.	**ppp authentication pap**

Fault Management

To manage network faults, you need to discover, isolate, and fix the problems. You can discover problems with the system's monitoring commands, isolate problems with the system's test commands, and resolve problems with other commands, including the **debug** command.

This section introduces basic fault management commands. For detailed troubleshooting procedures and a variety of scenarios, see the *Troubleshooting Internetworking Systems* publication. For complete detail on all **debug** commands, see the *Debug Command Reference* publication.

To perform general fault management, complete the following tasks:

- Display system information
- Test network connectivity
- Limit TCP transactions
- Test interfaces and memory
- Log system error messages
- Enable debug operations

Most chapters in this guide include fault management tasks listed under a monitoring and maintaining section.

Display System Information

To provide information about system processes, the software includes an extensive list of EXEC commands that begin with the word **show** and, when executed, display detailed tables of system information. Following is a list of the more common system management **show** commands. Perform these tasks in EXEC mode to display the information described:

Task	Command
Display information about all active processes.	**show processes** [**cpu** | **memory**]
Display the configured protocols.	**show protocols**

Look for specific **show** commands in the tables of configuration tasks found throughout the chapters in this guide. See the *Communication Server Command Reference* publication for detailed descriptions of the commands.

The following sections describe the EXEC commands you can use to monitor and troubleshoot the voltage and temperature of your system environment.

Test Network Connectivity

Complete the following tasks to test basic network connectivity:

- Set up TCP keepalive packet service
- Test connections with the **ping** command
- Trace packet routes

Set up TCP Keepalive Packet Service

The TCP keepalive capability allows a communication server to detect when the host with which it is communicating experiences a system failure, even if data stops being transmitted (in either direction). This is most useful on incoming connections. For example, if a host failure occurs while talking to a printer, the communication server might never notice, since the printer does not generate any traffic in the opposite direction. If keepalives are enabled, they are sent once every minute on

otherwise idle connections. If five minutes pass and no keepalives are detected, the connection is closed. The connection will also be closed if the host replies to a keepalive packet with a reset packet. This will happen if the host crashes and comes back up again.

To set up the TCP keepalive packet service, perform the following task in global configuration mode:

Task	Command
Generate TCP keepalive packets on idle network connections, either incoming connections initiated by a remote host, or outgoing connections initiated by a user.	s**ervice tcp-keepalives** {**in** | **out**}

Test Connections with the Ping Command

As an aid to diagnosing basic network connectivity, many network protocols support an echo protocol. The protocol involves sending a special datagram to the destination host, then waiting for a reply datagram from that host. Results from this echo protocol can help in evaluating the path-to-host reliability, delays over the path, and whether the host can be reached or is functioning.

To use the echo protocol, perform the following task in either user EXEC or privileged EXEC mode:

Task	Command
Invoke a diagnostic tool for testing connectivity.	**ping** [*protocol*] {*host* | *address*}

Look for specific **ping** commands in the tables of configuration tasks found throughout the chapters in this guide. See the *Communication Server Command Reference* publication for detailed descriptions of the command.

Trace Packet Routes

You can discover the routes IP packets will actually take when traveling to their destination. To do so, perform one of the following tasks:

Trace packet routes through the network (privileged level).	**trace** [*destination*]
Trace packet routes through the network (user level).	**trace ip** *destination*

Limit TCP Transactions

You can limit TCP transactions by performing the following task in global configuration mode:

Task	Command
Enable the Nagle slow packet avoidance algorithm. RFC 896.	**service nagle**

Convert Line Numbers from Octal to Decimal

You can convert the line numbers on the ASM-CS communication server from their default octal to a decimal number.

Task	Command
Specify that line numbers be displayed and interpreted as decimal numbers rather than octal numbers.	**service decimal-tty**

Test Memory and Interfaces

You can test the status of the following items:

- Flash memory
- System memory
- Interfaces

Note Performing these tasks is not recommended; they are intended to aid manufacturing personnel in checking system functionality.

Log System Error Messages

By default, the network servers send the output from the EXEC command **debug** and system error messages to the console terminal. You can redirect these messages, as well as output from asynchronous events such as interface transition, to other destinations. These destinations include virtual terminals, internal buffers, and UNIX hosts running a syslog server; the syslog format is compatible with 4.3 BSD UNIX.

Additionally, you can set the severity level of the messages to control the type of messages displayed. You can also have log messages timestamped to enhance real-time debugging and management.

With the current software release, there are three new syslog messages at LOG_NOTICE syslog level that make it easier to check the status of how the system provides address resolution. An example follows:

```
%LINK-5-BOOTP: Ethernet0 address 131.108.160.24, resolved by 131.108.1.111
%LINK-5-RARP: Ethernet0 address 131.108.160.24, resolved by 131.108.1.111
%LINK-5-SLARP: Ethernet0 address 131.108.160.24, resolved by 131.108.1.111
```

There are also new startup messages that help you identify NVRAM problems:

```
Warning: NVRAM device not found
Warning: NVRAM invalid, possibly due to write erase
```

The following level 4 LOG_WARNING message has been added for FDDI status information:

```
%FDDISTAT-4-STATUS: FDDI state indication detected on interface variable
```

The possible values for *indication* are listed in the next paragraph. The variable will be replaced with something like fddi0, for example.

Changes in status reflect interface connectivity or cabling problems (or fixes). The possible status reports include the following indications:

```
isolated
wrap A
wrap B
wrap a-B
thru A
thru B
thru A-B
```

Log Errors to a UNIX Syslog Daemon

To set up the syslog daemon on a 4.3 BSD UNIX system, include a line such as the following in the file */etc/syslog.conf*:

```
local7.debugging                    /usr/adm/logs/cisco.log
```

The **local7** keyword specifies the logging facility to be used; this can be changed.

The **debugging** argument specifies the syslog level. Refer to your syslog manual page.

The syslog daemon sends messages at or above this level to the file specified in the next field. The file must already exist, and the syslog daemon must have permission to write to it.

Set the Error Message Display Device

By default, error messages are directed to the system console. To direct messages to other devices, perform one of the following tasks in global configuration mode:

Task	Command
Log messages to an internal buffer.	**logging buffered**
Log messages to a UNIX syslog server host.	**logging** *host*
Redirect messages to the system console.	**no logging on**

To display system error messages to a nonconsole terminal, perform the following task in privileged EXEC mode

Task	Command
Display system error message on a nonconsole terminal.	**terminal monitor l**

The **terminal monitor** command is described further in the chapter“Line Configuration and Terminal Setting Commands,” in the *Communication Server Command Reference*.

Define the Error Message Severity Level and Facilities

You can limit messages displayed to the selected device by specifying the severity level of the error message. To do so, perform one of the following tasks in global configuration mode:

Task	Command
Limit messages logged to the console.	**logging console** *level*
Limit messages logged to the terminal lines.	**logging monitor** *level*
Limit messages logged to the syslog servers.	**logging trap** *level*

Table 1-2 lists the error message levels and corresponding UNIX syslog definitions.

Table 1-2 Error Message Logging Keywords

Level Keyword	Level	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 but significant condition	LOG_NOTICE
informational	6	Informational messages only	LOG_INFO
debugging	7	Debugging messages	LOG_DEBUG

Current software generates four categories of error messages:

- Error messages about software or hardware malfunctions, displayed at levels **warnings** through **emergencies**
- Output from the **debug** commands, displayed at the **debugging** level
- Interface up/down transitions and system restart messages, displayed at the **notification** level
- Reload requests and low-process stack messages, displayed at the **informational** level

The default is to log to the console messages at the **debugging** level and higher.

Define the Syslog Facility

You can also configure the system log (syslog) facility in which error messages are sent by performing the following task in global configuration mode:

Task	Command
Configure system log facilities.	**logging facility** *facility-type*

Table 1-3 lists the logging facility types and their descriptions.

Table 1-3 Logging Facility Types

Facility Type	Description
auth	Indicates the authorization system.
cron	Indicates the cron facility.
daemon	Indicates the system daemon.
kern	Indicates the Kernel.
local0–7	Reserved for locally defined messages
lpr	Indicates line printer system.
mail	Indicates mail system.
news	Indicates USENET news.
sys9	Indicates system use.
sys10	Indicates system use.

sys11	Indicates system use.
sys12	Indicates system use.
sys13	Indicates system use.
sys14	Indicates system use.
syslog	Indicates the system log.
user	Indicates user process.
uucp	Indicates UNIX-to-UNIX copy system.

To display the addresses and levels associated with the current logging setup, as well as any other logging statistics, perform the following task in EXEC mode:

Task	Command
Display the state of syslog error and event logging, including host addresses and whether console logging is enabled.	**show logging**

Enable Timestamps on Log Messages

By default, log messages are not timestamped. You can enable timestamping of log messages by performing the following task in global configuration mode:

Task	Command
Enable log timestamps with time since thesystem was rebooted.	**service timestamps log uptime**
	or
Enable timestamps, setting a specific time.	**service timestamps log datetime** [**msec**] [**localtime**] [**show-timezone**]

Enable Debug Operations

Your communication server includes hardware and software to aid in tracking down problems with the communication server or with other hosts on the network. The privileged **debug** EXEC commands start the console display of several classes of network events. The following tasks describe, in general, the system debug message feature. Refer to the *Debug Command Reference* publication for all information regarding **debug** commands. Also refer to the *Troubleshooting Internetworking Systems* publication.

Task	Command
Display the state of each debugging option.	**show debugging**
Display a list and brief description of all the **debug** command options.	**debug ?**
Begin message logging for the specified **debug** command.	**debug** *command*
Turn message logging off for the specified **debug** command.	**undebug** *command*

You can configure timestamping of system debug messages. Timestamping enhances real-time debugging by providing the relative timing of logged events. This information is especially useful when customers send debugging output to your technical support personnel for assistance. To enable timestamping of system debug messages, perform the following task in global configuration mode:

Task	Command
Enable debug timestamps with time since thesystem was rebooted.	**service timestamps debug uptime**
	or
Enable timestamps, setting a specific time.	**service timestamps debug datetime** [**msec**] [**localtime**] [**show-timezone**]

Normally, the messages are displayed only on the console terminal. See the section "Set the Error Message Display Device" earlier in this chapter to change the output device.

Note The system gives high priority to debugging output. For this reason, debugging commands should be turned on only for troubleshooting specific problems or during troubleshooting sessions with technical support personnel. Excessive debugging output can render the system inoperable.

System Performance Management

To manage system performance, you need to monitor and determine response time, error rates, and availability. Once these factors are determined, you can perform load balancing and modify system parameters to enhance performance. For example, priority queuing allows you to prioritize traffic order.

This section describes how to manage general system performance by completing the following tasks:

- Establish queuing strategies
- Adjust the system buffer size

Establish Queuing Strategies

Cisco provides two types of queuing strategies for prioritizing network traffic:

- Priority queueing
- Custom queuing

You can configure both priority queuing and custom queueing, but you can only assign *either* a priority group or a custom queue to an interface.

Priority output queuing is a mechanism that allows the administrator to set priorities on the type of traffic passing through the network. Packets are classified according to various criteria, including protocol and subprotocol type, and then queued on one of four output queues (high, medium, normal, and low).

When the server is ready to transmit a packet, it scans the priority queues in order, from highest to lowest, to find the highest priority packet. After that packet is completely transmitted, the server scans the priority queues again. If a priority output queue fills up, packets will be dropped and, for IP, quench indications will be sent to the original transmitter.

Although you can enable priority output queuing for any interface, the intended application was for low bandwidth, congested serial interfaces. The priority output queuing mechanism allows traffic control based on protocol or interface type. You can also set the size of the queue and defaults for what happens to packets that are not defined by priority output queue rules.

The priority output queuing mechanism can be used to manage traffic from all networking protocols. Additional fine-tuning is available for IP and for setting boundaries on the packet size.

Note Priority queuing introduces extra overhead that is acceptable for slow interfaces, but might not be acceptable for higher-speed interfaces such as Ethernet.

Note Priority queuing does not operate over X.25.

The four priority queues—high, medium, normal, and low—are listed in order from highest to lowest priority. Keepalives sourced by the network server are always assigned to the high priority queue; all other management traffic (such as IGRP updates) must be configured. Packets that are not classified by the priority list mechanism are assigned to the normal queue.

A priority list is a set of rules that describes how packets should be assigned to priority queues. A priority list might also describe a default priority or the queue size limits of the various priority queues.

Priority queuing introduces a fairness problem in that packets classified to lower-priority queues may not get serviced in a timely manner, or at all, depending upon the bandwidth used by packets sent from the higher-priority output queues.

With custom output queuing, a “weighted fair” queuing strategy is implemented for the processing of interface output queues. You can control the percentage of an interface’s available bandwidth that is used by a particular kind of traffic. When custom queuing is enabled on an interface, the system maintains 11 output queues for that interface that can be used to modify queueing behavior.

For queue numbers 1 through 10, the system cycles through the queues sequentially, delivering packets in the current queue before moving on to the next. Associated with each output queue is a configurable byte count, which specifies how many bytes of data the system should deliver from the current queue before it moves on to the next queue. When a particular queue is being processed, packets are sent until the number of bytes sent exceed the queue byte count or the queue is empty. Bandwidth used by a particular queue can only be indirectly specified in terms of byte count and queue length.

Queue number 0 is a system queue; it is emptied before any of the queues numbered 1 through 10 are processed. The system enqueues high-priority packets, such as keepalive packets, to this queue. Other traffic cannot be configured to use this queue.

You can set up both priority queuing and custom queuing on your network, but you can assign only one or the other to an interface.

Following is a list of the priority-setting tasks that you can choose from, depending upon the needs of your network:

- Set priority by protocol type
- Assign a default priority

- Set priority by interface type
- Specify the maximum packets and bytes in the priority queues
- Assign a priority list or a custom queue to an interface
- Monitor the priority and custom queue lists

See the following sections for more information about these tasks.

Set Priority by Protocol Type

You can establish queuing priorities based upon the protocol type. All Cisco-supported protocols are allowed. Perform one of the following tasks in global configuration mode:

Task	Command
Establish an output priority queuing list.	**priority-list** *list* **protocol** *protocol-name* {**high** | **medium** | **normal** | **low**} *queue-keyword keyword-value*
Establish a custom queuing list.	**queue-list** *list* **protocol** *protocol-name queue-number queue-keyword keyword-value*

Queue keywords provide additional options including byte-count, TCP services and port number assignments, and IP and Novell access list assignments.

Assign a Default Priority

You can assign a queue for those packets that did not match any other rule in the list. To do so, perform one of the following tasks in global configuration mode:

Task	Command
Assign a priority queue for those packets that do not match any other rule in the priority list.	**priority-list** *list* **default** {**high** | **medium** | **normal** | **low**}
Assign a queue number for those packets that do not match any other rule in the custom queue list.	**queue-list** *list* **default** *queue-number*

Set Priority by Interface Type

You can establish queuing priorities on packets entering from a specific interface. Perform the followint tasks in global configuration mode:

Task	Command
Establish queuing priorities on packets entering from a given interface.	**priority-list** *list* **interface** *interface-type interface-number* {**high** | **medium** | **normal** | **low**}
Establish custom queuing based on packets entering from a given interface.	**queue-list** *list* **interface** *interface-type interface-number queue-number*

Specify the Maximum Packets and Bytes in the Priority Queues

You can specify the maximum number of packets that might be waiting in each of the priority queues. Perform the following tasks:

Task	Command
Specify the maximum number of packets that can be waiting in each of the priority queues	**priority-list** *list* **queue-limit** *high-limit medium-limit normal-limit low-limit*
Specify the maximum number of packets that can be waiting in each of the priority queues	**queue-list** *list* **queue** *queue-number* **limit** *limit-number*
Designate the byte size allowed per queue.	**queue-list** *list* **queue** *queue-number* **byte-count** *byte-count number*

Both the **limit** and **byte-count** keywords might appear as arguments to the **queue-list** *list* **queue** command.

Assign a Priority List or a Custom Queue to an Interface

You can assign a priority list number to an interface. Only one list can be assigned per interface. Perform the following tasks:

Task	Command
Assign a priority list number to the interface.	**priority-group** *list*
Assign a custom queue list number to the interface.	**custom-queue-list** *list*

On an interface, you can either apply priority queueing or custom queueng. With priority queuing, some lower priority queues might not get serviced at all, depending on the bandwidth used by packets in higher-priority queues. With the queue-list command, you can configure the amount of bandwidth used by a particular kind of traffic.

Monitor the Priority and Custom Queuing Lists

You can display information about the input and output queues when priority queuing is enabled on an interface.Perform either the of the following tasks in EXEC mode:

Task	Command
Show the status of the priority queuing lists.	**show queuing priority**
Show the status of the custom queue lists.	**show queuing custom**

If you enter the show queuing command without any keywords, status on both custom and priority queue lists are displayed.

Modify 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, perform one of the following tasks in global configuration mode:

Task	Command
Adjust the system buffer sizes.	**buffers** {**small** | **middle** | **big** | **large** | **huge**} {**permanent** | | **max-free** | **min-free** | **initial**} *number*
Dynamically resize all huge buffers to the value that you supply.	**buffers huge size** *number*

During normal system operation, there are several pools of different sized buffers. These pools grow and shrink based upon demand. Some buffers are temporary and are created and destroyed as needed. Other buffers are permanently allocated and cannot be destroyed.

Note It is normally not necessary to adjust these parameters; *do so only after consulting with technical support personnel*. Improper settings could adversely impact system performance.

For examples of modifying the system buffer size, see the examples at the end of this chapter.

The server has one pool of queuing elements and five 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, perform the following task in EXEC mode:

Task	Command
List statistics for the buffer pools.	**show buffers** [*interface*]

Accounting Management

Accounting management allows you to track both individual and group usage of network resources. You can then reallocate resources as needed. For example, you can change the system timers and configure TCP keepalives. See also the IP accounting feature, described in the chapter“Configuring IP,” in this manual.

Additional tasks for measuring system resources are covered in other protocol-specific chapters.

Display Stack Utilization

You can display stack utilization of processes and interrupt routines, including the reason for the last system reboot. This feature is useful for anaylzing system crashes. To display stack utilization, perform the following task in EXEC mode:

Task	Command
Display stack utilization of processes and interrupt routines.	**show stacks**

Display Memory Utilization

You can display memory pool statistics, including summary information about the activites of the system memory allocator and memory utilization. Perform the following tasks:

Task	Command
Display memory pool statistics including summary information about the activities of the system memory allocator and a block-by-block listing of memory use.	**show memory** [**free**]
Display information about memory utilization.	**show processes memory**

System Management Examples

The following is a list of the examples in this section:

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
! Define host filenames
boot host host1-confg 131.108.1.111
boot host host2-confg 131.108.1.111
! Define system filenames
boot system sys1-system 131.108.13.111
boot system sys2-system 131.108.1.111
!
! Enable SNMP
snmp-server community
snmp-server trap-authentication
snmp-server host 131.108.1.27 public
snmp-server host 131.108.1.111 public
snmp-server host 131.108.2.63 public
!
! Define TACACS server hosts
tacacs-server host 131.108.1.27
tacacs-server host 131.108.13.33
tacacs-server host 131.108.1.33
!
! 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
```

Examples of Modifying Buffers

In the following example, the system will try to keep at least 50 small buffers free:

```
buffers small min-free 50
```

In the following example, the system will try to keep no more than 200 medium buffers free:

```
buffers middle max-free 200
```

In the following example, the system will try to create one large temporary extra buffer, just after a reload:

```
buffers large initial 1
```

In the following example, the system will try to create one permanent huge buffer:

```
buffers huge permanent 1
```

Username Examples

The following sample configuration sets up secret passwords on communication servers A, B, and C, thus enabling the three communication servers to connect to each other.

To authenticate connections between communication servers A and B, enter the following commands:

- On communication server A:

```
username B password a-b_secret
```

- On communication server B:

```
username A password a-b_secret
```

To authenticate connections between communication servers A and C, enter the following commands:

- On communication server A:

```
username C password a-c_secret
```

- On communication server C:

```
username A password a-c_secret
```

To authenticate connections between communication servers B and C, enter the following commands:

- On communication server B:

```
username C password b-c_secret
```

- On communication server C:

```
username B password b-c_secret
```

When you specify an encryption type of 0 to enter an unencrypted password, the system displays the encrypted version of the password. For example, suppose you enter the following command:

```
username bill password westward
```

The system would display this command like this:

```
username bill password 7 21398211
```

The encrypted version of the password is 21398211. The password was encrypted by the Cisco-defined encryption algorithm, as indicated by the "7."

If you were to enter the following command, the system would assume that the password is already encrypted and would do no encryption. It would display the command exactly as you typed it:

```
username bill password 7 21398211
username bill password 7 21398211
```