Doc. No. 78-0944-06A0
June 14, 1993

CISCO SYSTEMS

Terminal Server Configuration and Reference Errata

This document supplies corrections and additional informaiton for the 9.0 version of the Cisco publication *Terminal Server Configuration and Reference* dated April 1992. Keep this document with the *Terminal Server Configuration and Reference* document for future reference.

Correction to Chapter 3, "Terminal Server User Commands"

On page 3-6 and 3-45, the syntax of the **/route** argument is incorrectly given as follows:

/route: *path*

The correct form does not use the colon character, as follows:

/route *path*

Correction to Chapter 4, "System Configuration"

The following corrections apply to Chapter 4.

Correction to "Establishing the Privileged-Level Password"

On page 4-17, replace the last paragraph of the **enable password** description with the paragraphs that follow.

When you use the **enable** command at the console terminal, the EXEC does not prompt you for a password if the privileged mode password is not set. Additionally, if the enable password is not set and the line 0 (console line) password is not set, it is only possible to enter privileged mode on the console terminal. This feature allows you to use physical security rather than passwords to protect privileged mode if that is what you prefer to do.

If the enable password is not set and the line 0 (console) password is set, it is possible to enter privileged command mode either without entering a password at the console terminal or by entering the console line password when prompted while using any other line.

Correction to "Logging Messages to a UNIX Syslog Server"

Add the note that follows to the example on page 4-35.

Note: Many UNIX systems require a tab character to be used as the "white space" separator in the /etc/syslog.conf file. Use of a space character rather than a tab may cause the entry in /etc/syslog.conf to be ignored.

On page 4-55, replace the description of the **no modem** command with the following sentence:

The **no modem** *keyword* subcommand removes modem control from a line.

Corrections to Chapter 5, "System Management"

On page 5-10 under the section "Displaying Active Processes," add the following description of the CPU utilization field:

The CPU utilization field provides a general idea of how busy the processor is. It is a ratio of the current idle time over the longest idle time. This information should be used as an estimate only.

Corrections to Chapter 6, "Interface Configuration and Support"

The following is the complete documentation regarding the collection and use of RIF information. It should be placed in the section "Token Ring Interface Support" on page 6-17.

Configuring RIFs in Source-Route Bridging Environments

This section explains how to build routing information fields (RIFs). Terminal servers on a Token Ring network in a source-route bridging environment must support the collection and use of RIF information, to provide necessary path information to the host.

A RIF is built up of ring and bridge numbers. A *ring* is a single Token Ring network segment. Each ring in the extended Token Ring network is designated by a unique 12-bit ring number. Each bridge between two Token Rings is designated by a unique 4-bit bridge number. Bridge numbers must be unique only between bridges that connect the same two Token Rings.

Figure 1 illustrates the basic format for the Routing Information Field.

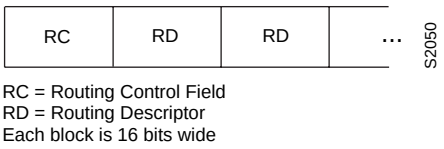

Figure 1 Basic RIF Format

Figure 2 illustrates the routing control format for the RIF. Descriptions of each field follow.

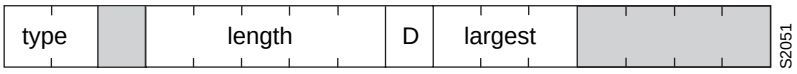

Figure 2 RIF Routing Control Format

- Shaded fields are reserved.
- type—RIF type, as follows:
  - 00: Specific route
  - 10: All rings, all routes
  - 11: All rings, spanning routes (limited broadcast)
- length—Total length in bytes of the RIF.
- D—Direction, indicated as follows:
  - 0: Interpret route left to right (forward)
  - 1: Interpret route right to left (reverse)
- largest—Largest frame that can be handled by this route, as follows:
  - 000: 516 bytes (DDN 1822)

- 001: 1500 bytes (Ethernet)
- 010: 2052 bytes
- 011: 4472 bytes (Token Ring at 4-Mb speed)
- 100: 8144 bytes (Token Bus and Token Ring at 16-Mb maximum)
- 101: 11407 bytes
- 110: 17800 bytes
- 111: 65535 bytes (initial values)

Figure 3 describes the routing descriptor format of the RIF string. Definitions of each field follow the figure.

Figure 3 Routing Descriptor Format

- Ring Number—Unique hexadecimal ring number within the bridged network.
- Bridge Number—Unique hexadecimal bridge number between any bridges connecting the same two rings.

Determining the RIF Timeout Interval

RIF information is maintained in a cache whose entries are aged. The global configuration command **rif timeout** determines the number of minutes an inactive RIF entry is kept. The full command syntax follows:

rif timeout *minutes*
no rif timeout

The default interval is 15 minutes. The minimum value is one minute. Assign a new interval value using the *minutes* argument.

The **no rif timeout** command restores the default.

The EXEC command **show rif** displays the contents of the RIF cache. The EXEC command **clear rif-cache** clears the contents of RIF cache. See the sections "Maintaining the Source-Route Bridge" and "Monitoring the Source-Route Bridge" later in this chapter for more information about these commands.

Example

This command changes the timeout period to five minutes.

```
!
rif timeout 5
!
```

Configuring a Static RIF Entry

If a Token Ring host does not support the use of IEEE 802.2 TEST or XID datagrams as explorer packets, you may need to add static information to the RIF cache of the router/bridge.

To enter static source-route information into the RIF cache, use the following variation of the **rif** global configuration command:

rif *MAC-address* [*RIF-string*] [*interface-name*]
no rif *MAC-address* [*interface-name*]

The argument *MAC-address* is a 12-digit hexadecimal string written as a dotted triple, for example 0010.0a00.20a6.

Using the command **rif** *MAC-address* without any of the optional arguments puts an entry into the RIF cache indicating that packets for this MAC address should not have RIF information.

The command **no rif** *MAC-address* removes an entry from the cache.

The optional argument *RIF-string* is a series of 4-digit hexadecimal numbers separated by a dot (.). This RIF string is inserted into the packets sent to the specified MAC address.

An interface name (for example, tokenring0) can be specified with the optional *interface-name* argument, to indicate the origin of the RIF.

Do not configure a static RIF with any of the *all rings* type codes. Doing so causes traffic for the configured host to appear on more than one ring and leads to unnecessary congestion. The format of a RIF string is illustrated in Figure 1, Figure 2, and Figure 3.

Example

In this example configuration, the path between rings 8 and 9 connected via source-route bridge 1 is described by the route descriptor 0081.0090. A full RIF, including the route control field, would be 0630.0081.0090. The static RIF entry would be submitted to the leftmost router as shown in Figure 4.

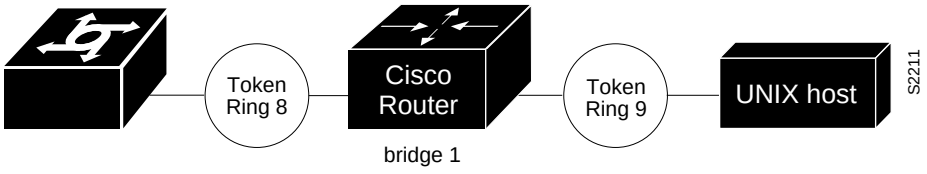

Figure 4 Assigning a RIF to a Source-Route Bridge

```
!
rif 1000.5A12.3456 0630.0081.0090
!
```

As another example, assume a datagram was sent from a Cisco router/bridge on ring 21 (15 hexadecimal), across bridge 5 to ring 256 (100 hexadecimal), and then across bridge 10 (A hexadecimal) to ring 1365 (555 hexadecimal) for delivery to a destination host on that ring. See Figure 5.

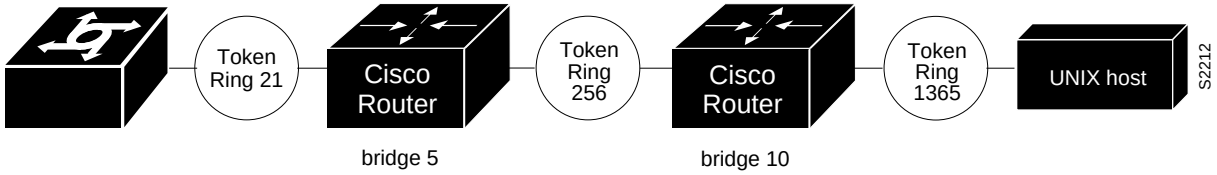

Figure 5 Assigning a RIF to a Two-Hop Path

The RIF in the leftmost router describing this two-hop path is 0830.0155.100a.5550 and is entered as follows:

```
!
rif 1000.5A01.0203 0830.0155.100a.5550
!
```

Maintaining the Source-Route Bridge

Use this EXEC command to maintain the source-route bridge cache.

clear rif-cache

The **clear rif-cache** command clears the entire RIF cache.

Displaying the RIF Cache

The **show rif** EXEC command displays the current contents of the RIF cache. Enter this command at the EXEC prompt:

show rif

The following is a sample display of **show rif**:

```
Codes: * interface, - static, + remote
Hardware Addr  How   Idle (min)  Routing Information Field
5C02.0001.4322 rg5           -   0630.0053.00B0
5A00.0000.2333 TR0           3   08B0.0101.2201.0FF0
5B01.0000.4444 -             -   -
0000.1403.4800 TR1           0   -
0000.2805.4C00 TR0           *   -
0000.2807.4C00 TR1           *   -
0000.28A8.4800 TR0           0   -
0077.2201.0001 rg5          10   0830.0052.2201.0FF0
```

Collecting and Using Routing Information Field (RIF) Information

ASM-CSs on a Token Ring network in a source-route bridging environment must support the collection and use of RIF information, to provide necessary path information to the host.

Level 3 routers that use protocol-specific information rather than MAC information to route datagrams must be able to collect and use RIF information to ensure that the Level 3 routers can transmit datagrams across a source-route bridge. The Cisco software default is to not collect and use RIF information for routed protocols. This allows operation with software that does not understand or properly use RIF information.

To enable collection and use of RIF information, use the **multiring** interface subcommand. The full command syntax follows:

multiring ip
no multiring ip

When it is enabled, the router will source packets that include information used by source-route bridges. This allows a terminal server with Token Ring interfaces to connect to a source-bridged Token Ring network.

The **no multiring ip** subcommand with the appropriate keyword disables the use of RIF information for the protocol specified.

Example

These commands enable a Token Ring interface for the IP protocol. RIFs will be generated for IP frames.

```
!
interface tokenring 0
multiring ip
ip address 131.108.183.37  255.255.255.0
!
```

Corrections to Chapter 8, "X.25 Configuration and Management"

Add the following text to the section "Configuring DDN X.25" on page 8-13:

For situations requiring a high degree of security, the Defense Data Network Blacker Front- End Encryption (BFE) device is supported. If the router is attached to such a device, the **bfex25** keyword must be used with the **encapsulation** subcommand:

encapsulation bfex25

This encapsulation provides a mapping from Class A IP addresses to the type of X.121 addresses expected by the BFE encryption device.

Corrections to Chapter 12, "LAT Configuration and Management"

Add the text that follows to the section "Specifying Access Conditions" on page 12-15.

When both IP and LAT connections are allowed from a terminal line, and an IP access list is applied to that line with the **access-class** line subcommand, you also must create a LAT access list numbered the same if you want to allow any LAT connections from that terminal. This is because you can specify only one incoming and one outgoing access list number for each terminal line, and when checking LAT access lists, if the list specified does not exist, the system denies all LAT connections.

Corrections to Chapter 13, "TCP/IP Configuration and Management"

Extended access lists are now supported in the terminal server on terminal lines. The following information applies to Chapter 13, "TCP/IP Configuration and Management," in the section titled "Restricting Terminal Connections."

Configuring Extended Access Lists

Extended access lists allow finer granularity in control of connections allowed to or from a specific terminal server port. For example, users may be restricted to only making connections to the telnet port, or incoming access to a port may be restricted to "privileged" ports on the original host.

To define an extended access list, use the extended version of the **access-list** subcommand, as follows:

access-list *list* {**permit**|**deny**} *protocol source source-mask destination destination-mask* [*operator operand*] [**established**]

The argument *list* is an integer from 100 through 199 that you assign to identify one or more extended permit/deny conditions as an extended access list. A list number in the range 100 to 199 distinguishes an extended access list from a standard access list. The condition keywords **permit** and **deny** determine whether the router allows or disallows a connection when a packet matches an access condition. The router stops checking the extended access list after a match occurs. All conditions must be met to make a match.

The argument *protocol* is one of the following keywords:

- **ip**
- **tcp**
- **udp**
- **icmp**

Use the keyword **ip** to match any Internet protocol, including TCP, UDP, and ICMP.

The argument *source* is an Internet source address in dotted-decimal format. The argument *source-mask* is a mask, also in dotted-decimal format, of source address bits to be ignored. The router uses the *source* and *source-mask* arguments to match the source address of a packet. For example, to match any address on a Class C network 192.31.7.0, the argument *source-mask* would be 0.0.0.255. The arguments *destination* and *destination-mask* are dotted-decimal values for matching the destination address of a packet.

To differentiate further among packets, you can specify the optional arguments *operator* and *operand* to compare destination ports, service access points, or contact names. Note that the **ip** and **icmp** protocol keywords do not allow port distinctions.

For the **tcp** and **udp** protocol keywords, the argument *operator* can be one of these keywords:

- **lt**—less than
- **gt**—greater than
- **eq**—equal
- **neq**—not equal

The argument *operand* is the decimal destination port for the specified protocol.

For the TCP protocol there is an additional keyword, **established**, that does not take an argument. A match occurs if the TCP datagram has the ACK or RST bits set, indicating an established connection. The nonmatching case is that of the initial TCP datagram to form a connection; the software goes on to other rules in the access list to determine whether a connection is allowed in the first place.

Note: After an access list is initially created, any subsequent additions (possibly entered from the terminal) are placed at the *end* of the list. In other words, you cannot selectively add or remove access lists command lines from an access list.

Controlling Line Access

To restrict incoming and outgoing connections between a particular terminal line or group of lines (into a Cisco device) and the addresses in an access list, use the **access-class** line configuration subcommand. Full command syntax for this command follows:

access-class *list* {**in**|**out**}
no access-class *list* {**in**|**out**}

This command restricts connections on a line or group of lines to certain Internet addresses.

The argument *list* is an integer from 1 through 199 that identifies a specific access list of Internet addresses.

The keyword **in** applies to incoming connections, such as virtual terminals. The keyword **out** applies to outgoing Telnet connections.

The **no access-class** line configuration subcommand removes access restrictions on the line for the specified connections.

Example 1:

The following example defines an access list that permits only hosts on network *192.89.55.0* to connect to the virtual terminal ports on the router.

```
access-list 12 permit 192.89.55.0  0.0.0.255
line 1 5
access-class 12 in
```

Use the **access-class** keyword **out** to define the access checks made on outgoing connections. (A user who types a host name at the router prompt to initiate a Telnet connection is making an outgoing connection.)

Note: Set identical restrictions on all the virtual terminal lines, because a user can connect to any of them.

Example 2:

The following example defines an extended access list that permits only telnet and rlogin.

```
access-list 101 permit tcp 0.0.0.0 0.0.0.0 0.0.0.0 255.255.255.255
eq 23
access-list 101 permit tcp 0.0.0.0 0.0.0.0 0.0.0.0 255.255.255.255
eq 513
!(implicit deny of everything else)
! public terminals can only telnet and rlogin line 1 20 access-class
101 out
!
```

Extended access-lists also can be used with **slip access-class** *list* [**in**|**out**].

Corrections to Chapter 15, "XRemote Configuration and Management"

On page 15-1, in the first bulleted list, the third bullet incorrectly lists "EXEC commands for troubleshooting TN3270 operation" and should say "EXEC commands for troubleshooting XRemote operation."

This document is to be used in conjunction with the *Terminal Server Configuration and Reference* publication.