



Raw Socket Transport Software Configuration Guide for Cisco 1000 Series Connected Grid Routers (Cisco IOS)

January 2014

OL-31251-01

This document provides details about configuring Raw Socket Transport on the Cisco 1000 Series Connected Grid Routers (hereafter referred to as the CGR 1000). Raw Socket Transport transports streams of characters from one serial interface to another over an IP network for utility applications.

This feature is also supported on the Cisco 2000 Series Connected Grid Router in Cisco IOS Release 15.2(4)M and later. For more information, see [Configuring Raw Socket Protocol on the CGR 2010 Router](#).

This document includes the following sections:

- [Information About Raw Socket Transport, page 2](#)
- [Prerequisites, page 4](#)
- [Guidelines and Limitations, page 4](#)
- [Default Settings, page 4](#)
- [Configuring Raw Socket Transport, page 5](#)
- [Verifying Configuration, page 11](#)
- [Configuration Example, page 11](#)
- [Feature History, page 14](#)
- [Raw Socket Transport Command Reference, page 15](#)



Information About Raw Socket Transport

Raw Socket is a method for transporting serial data through an IP network. The feature can be used to transport Supervisory Control and Data Acquisition (SCADA) data from Remote Terminal Units (RTUs). This method is an alternative to the Block Serial Tunnel (BSTUN) protocol.

Raw Socket Transport supports TCP or UDP as the transport protocol. An interface can be configured to use either protocol but not both at the same time. TCP transport is suitable for applications such as control applications that require acknowledged and sequenced delivery of data. For latency-sensitive applications such as line SEL relays, UDP transport provides faster transport of serial data than TCP.

In either transport mode, Raw Socket Transport supports up to 32 sessions per interface (a total of 32 x 2 sessions per CGR 1000).

Raw Socket Transport supports the following for each asynchronous serial interface:

- TCP as the transport protocol, with built-in auto TCP connection retry mechanism.
- Up to 32 TCP sessions per interface.
- Interface configuration as a server, client, or a combination of both.
- One server per interface, but multiple clients.
- VRF-awareness, which enables the router to send Raw Socket Transport traffic to a server host connected through a Virtual Private Network (VPN) Virtual Routing and Forwarding (VRF) interface.

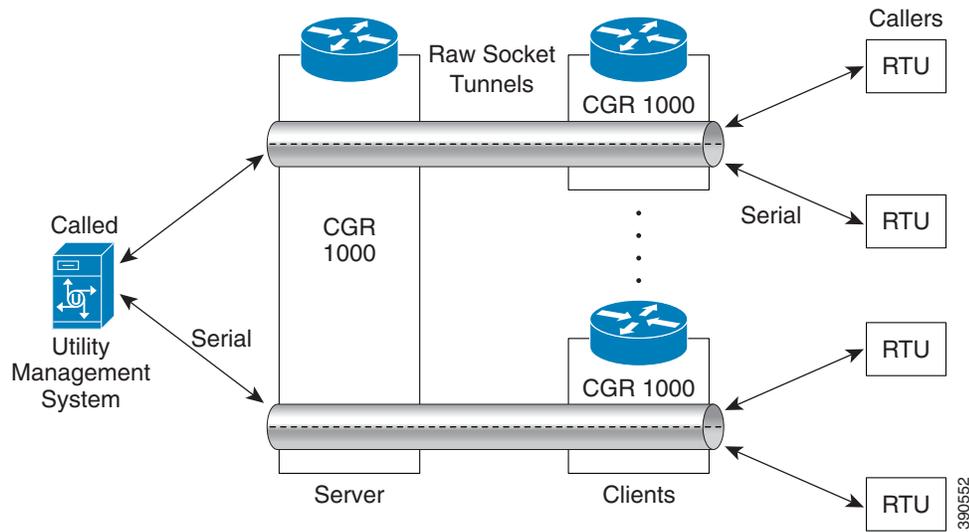
TCP Transport

TCP Raw Socket transport uses a client-server model. At most one server and multiple clients can be configured on a single asynchronous serial line. In client mode, the CGR 1000 can initiate up to 32 TCP sessions to Raw Socket servers, which can be other CGR routers or third-party devices.

[Figure 1](#) shows a sample Raw Socket TCP configuration. In this example, serial data is transferred between RTUs and a utility management system across an IP network that includes several CGR 1000 routers. One CGR 1000 router acts as a Raw Socket *server*, listening for TCP connection requests from the other CGR 1000 routers, which are configured as Raw Socket *clients*.

A Raw Socket client receives streams of serial data from the RTUs and accumulates this data in its buffer, then places the data into packets, based on user-specified packetization criteria. The Raw Socket client initiates a TCP connection with the Raw Socket server and sends the packetized data across the IP network to the Raw Socket server, which retrieves the serial data from the packets and sends it to the serial interface, and on to the utility management system.

Figure 1 Raw Socket TCP Transport



Note

When you configure the serial link interface on the router as a server, the interface's peer is the serial link interface on the client router and vice versa.

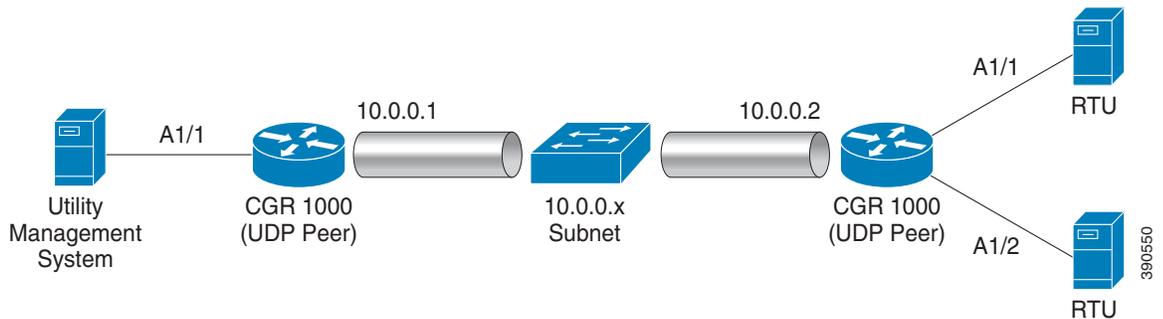
UDP Transport

UDP transport uses a peer-to-peer model. Multiple UDP connections can be configured on an asynchronous serial line.

Figure 2 shows a sample Raw Socket UDP configuration. In this example, serial data is transferred between RTUs and a utility management system across an IP network that includes two CGR 1000 routers that are configured as Raw Socket UDP peers.

In this example, the Raw Socket UDP peer receives streams of serial data from the RTUs and accumulates this data in its buffer, then places the data into packets, based on user-specified packetization criteria. The Raw Socket UDP peer sends the packetized data across the IP network to the Raw Socket peer at the other end, which retrieves the serial data from the packets and sends it to the serial interface, and on to the utility management system.

Figure 2 Raw Socket UDP Transport



Serial Data Processing

The streams of serial data received by a Raw Socket peer can be packetized based on the following criteria:

- **Packet length**—You can specify a packet length that triggers the CGR 1000 to transmit the serial data to the peer. Once the CGR 1000 collects this much data in its buffer, it packetizes the accumulated data and forwards it to the Raw Socket peer.
- **Packet-timer value**—The packet timer specifies the amount of time the CGR 1000 waits to receive the next character in a stream. If a character is not received by the time the packet timer expires, the data the CGR 1000 has accumulated in its buffer is packetized and forwarded to the Raw Socket peer.
- **Special character**—You can specify a character that will trigger the CGR 1000 to packetize the data accumulated in its buffer and send it to the Raw Socket peer. When the special character (for example, a CR/LF) is received, the CGR 1000 packetizes the accumulated data and sends it to the Raw Socket peer.

In addition to the packetizing options, you can also select best-effort mode for faster processing. This mode trades reliability for speed. If the buffer is full, the CGR 1000 drops the older packets.

VRF-Aware Raw Socket

The VRF-aware Raw Socket Transport feature enables you to isolate Raw Socket traffic using a VRF for efficient management and control of serial data. After configuring a VRF, you can associate the serial interface configured for Raw Socket Transport with the VRF. See the [“Raw Socket VRF” section on page 13](#) for a configuration example.

Prerequisites

Determine how you want Raw Socket traffic transported in your network, including the network devices and interfaces to use, how the router packetizes the serial data, and whether to use VRF.

Guidelines and Limitations

Typically, UDP traffic is blocked by firewalls in the network. If the network has such firewalls, make sure to configure pinholes to allow the raw socket UDP traffic.

Default Settings

Feature	Default Setting
Raw Socket Transport	Disabled.
Packet length	No packet length is configured.
Packet timeout	10 ms.

Feature	Default Setting
Special character	No special character is configured.
Raw Socket mode	Best-effort mode is off.
TCP idle timeout	5 minutes.

Configuring Raw Socket Transport

This section includes the following topics:

- [Enabling Raw Socket Transport on the Serial Interface, page 5](#)
- [Configuring Common Raw Socket Line Options, page 6](#)
- [Configuring the Raw Socket TCP Server, page 7](#)
- [Configuring the Raw Socket TCP Client, page 8](#)
- [Configuring a Raw Socket UDP Peer-to-Peer Connection, page 9](#)

Enabling Raw Socket Transport on the Serial Interface

To enable Raw Socket Transport on the CGR 1000 router, you must first enable an asynchronous serial port and enable Raw Socket TCP or UDP encapsulation for that port.

BEFORE YOU BEGIN

Determine availability of the serial port on the CGR 1000.

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	interface async <i>slot/port</i>	Enters interface command mode for the serial slot/port. Note The slot/port configuration for the serial port can be 1/1 or 1/2.
Step 3	physical-layer async	Places the interface in asynchronous mode.
Step 4	no ip address	Disables IP processing on the interface.
Step 5	encapsulation raw-tcp or encapsulation raw-udp	Enables Raw Socket TCP encapsulation or UDP encapsulation for the serial port.
Step 6	async mode dedicated	(Optional) Places the line into dedicated asynchronous network mode.
Step 7	exit	Exits global configuration mode.

EXAMPLE

This example shows how to enable serial port 1/1 and how to enable Raw Socket TCP encapsulation on that port.

```
router# configure terminal
router(config)# interface async 1/1
router(config-if)# physical-layer async
router(config-if)# no ip address
router(config-if)# async mode dedicated
router(config-if)# encapsulation raw-tcp
router(config-if)# exit
```

This example shows how to enable serial port 1/1 and how to enable Raw Socket UDP encapsulation on that port.

```
router# configure terminal
router(config)# interface async 1/1
router(config-if)# physical-layer async
router(config-if)# no ip address
router(config-if)# async mode dedicated
router(config-if)# encapsulation raw-udp
router(config-if)# exit
```

Configuring Common Raw Socket Line Options

You can configure options common to all connections on a line. The common options apply to both TCP and UDP.

BEFORE YOU BEGIN

Enable Raw Socket Transport as described in [Enabling Raw Socket Transport on the Serial Interface, page 5](#).

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	line <i>slot/port</i>	Enters line command mode for the serial slot/port. Note The slot/port configuration for the line can be 1/1 or 1/2.
Step 3	raw-socket packet-length <i>length</i>	Specifies the packet size that triggers the CGR 1000 to transmit the data to the peer. When the CGR 1000 accumulates this much data in its buffer, it packetizes the data and forwards it to the Raw Socket peer. <i>length</i> —2 to 1400 bytes. By default, the packet-length trigger is disabled.

	Command	Purpose
Step 4	raw-socket packet-timer <i>timeout</i>	Specifies the maximum time in milliseconds the CGR 1000 waits to receive the next character in a stream. If a character is not received by the time the packet-timer expires, the accumulated data is packetized and forwarded to the Raw Socket peer. <i>timeout</i> —3 to 1000 ms. The default is 10 ms.
Step 5	raw-socket spec-char <i>ascii_char</i>	Specifies a character that will trigger the CGR 1000 to packetize the data accumulated in its buffer and send it to the Raw Socket peer. <i>ascii_char</i> —0 to 255. By default, the special character trigger is disabled.
Step 6	raw-socket mode best-effort	Enable best-effort mode for the serial line. When this mode is enabled, older packets are dropped from the head of the queue when the queue is full. By default, best-effort mode is off.

Use the **no** form of these commands to return to the default values.

EXAMPLE

```
router# configure terminal
router(config)# line 1/1
router(config-line)# raw-socket packet-length 32
router(config-line)# raw-socket packet-timer 500
router(config-line)# raw-socket special-char 3
```

Configuring Raw Socket TCP

After enabling Raw Socket TCP encapsulation, you configure the TCP server and/or clients.

Configuring the Raw Socket TCP Server

BEFORE YOU BEGIN

Enable a serial port and Raw Socket TCP encapsulation for that port, as described in [“Enabling Raw Socket Transport on the Serial Interface”](#) section on page 5.

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enters configuration mode.
Step 2	line <i>slot/port</i>	Enters line command mode for the serial slot/port. Note The slot/port configuration for the line can be 1/1 or 1/2.

	Command	Purpose
Step 3	<code>raw-socket tcp server port [ip_address]</code>	Starts the Raw Socket Transport TCP server for an asynchronous line interface. In Raw Socket server mode, the CGR 1000 listens for incoming connection requests from Raw Socket clients. <i>port</i> —Port number the server listens on. <i>ip_address</i> —(Optional) Local IP address on which the server listens for connection requests.
Step 4	<code>raw-socket tcp idle-timeout session_timeout</code>	Sets the Raw Socket Transport TCP session timeout for the asynchronous line interface. If no data is transferred between the client and server over this interval, then the TCP session closes. The client then automatically attempts to reestablish the TCP session with the server. This timeout setting applies to all Raw Socket Transport TCP sessions under this particular line. <i>session_timeout</i> —Currently configured session idle timeout in minutes. The default is 5 minutes.

To remove a Raw Socket TCP server, use the **no raw-socket tcp server** command.

EXAMPLE

This example shows how to configure a Raw Socket TCP server for an asynchronous serial line. The TCP server listens for TCP client connection requests on local port 4000 and local IP address 10.0.0.1. If no data is exchanged between the Raw Socket TCP server and one of the TCP clients for 10 minutes, then the TCP session closes, and the Raw Socket client attempts to reestablish the session with the Raw Socket server.

```
router# configure terminal
router(config)# line 1/1
router(config-line)# raw-socket tcp server 4000 10.0.0.1
router(config-line)# raw-socket tcp idle-timeout 10
router(config-line)# exit
router(config)#
```

Configuring the Raw Socket TCP Client

BEFORE YOU BEGIN

Enable a serial port and Raw Socket TCP encapsulation for that port, as described in [“Enabling Raw Socket Transport on the Serial Interface”](#) section on page 5.

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enters configuration mode.
Step 2	line slot/port	Enters line command mode for the serial slot/port. Note The slot/port configuration for the line can be 1/1 or 1/2.
Step 3	raw-socket tcp client <i>dest_ip_address</i> <i>dest_port</i> [<i>local_ip_address</i>] [<i>local_port</i>]	Specifies settings for Raw Socket Transport TCP client sessions. <i>dest_ip_address</i> —Destination IP address of the remote Raw Socket server. <i>dest_port</i> —Destination port number to use for the TCP connection to the remote server. <i>local_ip_address</i> —(Optional) Local IP address that the client can also bind to. <i>local_port</i> —(Optional) Local port number that the client can also bind to.

To remove a Raw Socket TCP client, use the **no raw-socket tcp client** command.

EXAMPLE

This example shows how to configure a Raw Socket TCP client for an asynchronous serial line. The CGR 1000 (router), serving as a Raw Socket client, initiates TCP sessions with a Raw Socket server and forwards packetized serial data to it. The router collects streams of serial data in its buffer; when it accumulates 827 bytes in its buffer, the router packetizes the data and forwards it to the Raw Socket server. If the router and the Raw Socket server do not exchange any data for 10 minutes, then the TCP session with the Raw Socket server closes, and the router attempts to reestablish the session with the Raw Socket server.

```
router# configure terminal
router(config)# line 1/1
router(config-line)# raw-socket tcp client 10.0.0.1 4000
router(config-line)# raw-socket packet-length 827
router(config-line)# raw-socket tcp idle-timeout 10
router(config-line)# exit
router(config)#
```

Configuring a Raw Socket UDP Peer-to-Peer Connection

After enabling Raw Socket UDP encapsulation and the common line options, you configure the Raw Socket UDP peer-to-peer connection. The local port on one end of the connection should be the destination port on the other end.

BEFORE YOU BEGIN

Enable a serial port and Raw Socket UDP encapsulation for that port, as described in [“Enabling Raw Socket Transport on the Serial Interface”](#) section on page 5.

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enters configuration mode.
Step 2	line slot/port	Enters line command mode for the serial slot/port. Note The slot/port configuration for the line can be 1/1 or 1/2.
Step 3	raw-socket udp connection <i>dest_ip_address</i> <i>dest_port local_port [local_ip_address]</i>	Specifies settings for Raw Socket Transport UDP connections. <i>dest_ip_address</i> —Destination IP address to use for the UDP connection. <i>dest_port</i> —Destination port number to use for the UDP connection. <i>local_port</i> —Local port number for the UDP connection. <i>local_ip_address</i> —(Optional) Local IP address for the UDP connection.

To remove a Raw Socket UDP connection, use the **no raw-socket udp connection** command.

EXAMPLE

This example shows how to configure a Raw Socket UDP connection between router A (local IP address 192.168.0.8) and router B (local IP address 192.168.0.2).

Router A

```
router# configure terminal
router(config)# line 1/1
router(config-line)# raw-socket udp connection 192.168.0.2 5000 7000
router(config-line)# raw-socket mode best-effort
router(config-line)# exit
router(config)#
```

Router B

```
router# configure terminal
router(config)# line 1/1
router(config-line)# raw-socket udp connection 192.168.0.8 7000 5000
router(config-line)# raw-socket mode best-effort
router(config-line)# exit
router(config)#
```

Verifying Configuration

Command	Purpose
show running-config	Shows the configuration of the CGR 1000, including those features that are active and their settings.
show raw-socket tcp detail	Displays information about Raw Socket Transport TCP activity.
show raw-socket tcp sessions	Displays information about Raw Socket Transport TCP sessions.
show raw-socket tcp statistics	Displays Raw Socket Transport TCP statistics for each asynchronous serial line.
show raw-socket udp detail	Displays information about Raw Socket Transport UDP activity.
show raw-socket udp sessions	Displays information about Raw Socket Transport UDP sessions.
show raw-socket udp statistics	Displays Raw Socket Transport UDP statistics for each asynchronous serial line.
clear raw-socket statistics	Clears Raw Socket Transport statistics for a specific TTY interface or for all asynchronous serial lines.

Configuration Example

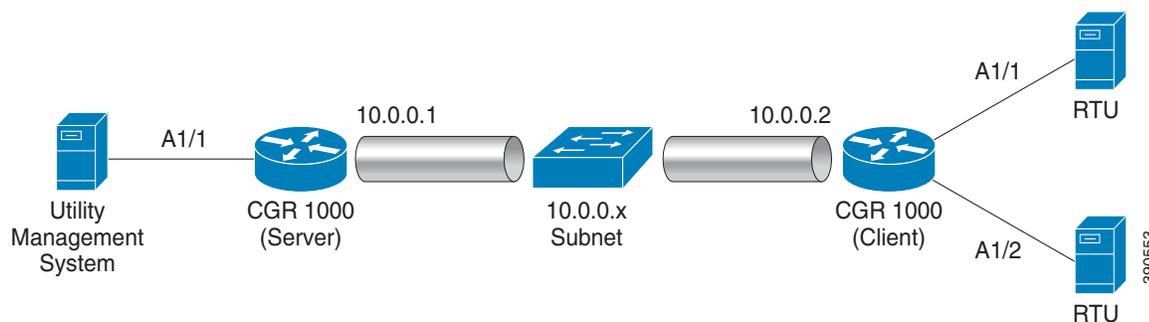
The following sections include Raw Socket Transport configuration examples:

- [Raw Socket TCP, page 11](#)
- [Raw Socket VRF, page 13](#)
- [Raw Socket UDP, page 12](#)

Raw Socket TCP

The following example shows a Raw Socket Transport configuration in which a CGR 1000 router acts as the server, and another CGR 1000 router acts as the client.

Figure 3 *Raw Socket Transport Configuration Example*



The following table displays the configuration of the server and client CGR 1000s highlighted in [Figure 3](#):

CGR 1000 Server Configuration	CGR 1000 Client Configuration
<pre> ... interface async 1/1 physical-layer async no ip address encapsulation raw-tcp ! ... line 1/1 raw-socket tcp server 5000 10.0.0.1 raw-socket packet-timer 3 raw-socket tcp idle-timeout 5 ... </pre>	<pre> ... interface async 1/1 physical-layer async no ip address encapsulation raw-tcp ! interface async 1/2 physical-layer async no ip address encapsulation raw-tcp ! ... line 1/1 raw-socket tcp client 10.0.0.1 5000 10.0.0.2 9000 raw-socket packet-length 32 raw-socket tcp idle-timeout 5 line 1/2 raw-socket tcp client 10.0.0.1 5000 10.0.0.2 9001 raw-socket packet-length 32 raw-socket tcp idle-timeout 5 </pre>

Raw Socket UDP

This example shows the configuration for a Raw Socket UDP connection between two CGR 1000 routers:

From Router1

```

interface GigabitEthernet0/1
ip address 192.168.0.8 255.255.255.0
duplex auto
speed auto

interface Async1/2
physical-layer async
no ip address
encapsulation raw-udp

line 1/2
raw-socket udp connection 192.168.0.2 2 2
raw-socket mode best-effort

```

From Router2

```

interface GigabitEthernet0/1
ip address 192.168.0.2 255.255.255.0
load-interval 60
duplex auto
speed auto
no keepalive

interface Async1/2
physical-layer async
no ip address

```

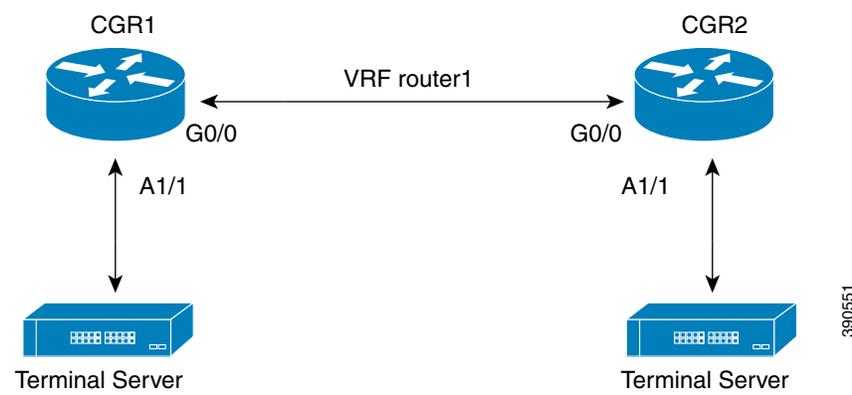
```
encapsulation raw-udp

line 1/2
raw-socket udp connection 192.168.0.8 2 2
raw-socket mode best-effort
```

Raw Socket VRF

The following example shows a Raw Socket VRF configuration in which two CGR 1000 routers, configured for Raw Socket Transport, connect through a VRF. CGR1 serves as the Raw Socket TCP server, and CGR2 serves as the Raw Socket TCP client.

Figure 4 Raw Socket Transport VRF Configuration Example



Following are the configurations of CGR1 and CGR2 as shown in [Figure 4](#):

CGR1 Configuration

Defining VRF on the router:

```
vrf definition router1
 rd 100:1
 route-target export 100:3
 route-target import 100:3
 !
 address-family ipv4
 exit-address-family
```

Applying VRF configuration on the interface:

```
interface GigabitEthernet0/0
 vrf forwarding router1
 ip address 100.100.100.2 255.255.255.0
 duplex auto
 speed auto
```

Applying raw-tcp on the serial interface:

```
interface async 1/1
 physical-layer async
 vrf forwarding router1
 no ip address
 encapsulation raw-tcp
```

Applying raw-tcp on the line:

```
line 1/1
  raw-socket tcp server 5000 4.4.4.4
```

CGR2 Configuration

Defining VRF on the router:

```
vrf definition router1
  rd 100:1
  route-target export 100:3
  route-target import 100:3
  !
  address-family ipv4
  exit-address-family
```

Applying VRF configuration on the interface:

```
interface GigabitEthernet0/0
  vrf forwarding router1
  ip address 100.100.100.1 255.255.255.0
  duplex auto
  speed auto
```

Applying raw-tcp on the serial interface:

```
interface async 1/1
  physical-layer async
  vrf forwarding router1
  no ip address
  encapsulation raw-tcp
```

Applying raw-tcp on line:

```
line 1/1
  raw-socket tcp client 4.4.4.4 5000
```

Feature History

Feature Name	Release	Feature Information
Raw Socket Transport	15.2(4)M	Initial support of the feature on the CGR 2010.
	15.3(3)M	Addition of VRF Awareness for TCP Raw Socket.
	15.4(1)CG	Initial support on the Cisco 1000 Series Connected Grid Routers (Cisco IOS).

Raw Socket Transport Command Reference

- [clear raw-socket statistics, page 16](#)
- [debug raw-socket driver event, page 18](#)
- [debug raw-socket driver packet, page 19](#)
- [debug raw-socket tcp event, page 20](#)
- [debug raw-socket tcp packet, page 22](#)
- [encapsulation raw-tcp, page 23](#)
- [encapsulation raw-udp, page 24](#)
- [raw-socket mode best-effort, page 25](#)
- [raw-socket packet-length, page 26](#)
- [raw-socket packet-timer, page 27](#)
- [raw-socket special-char, page 28](#)
- [raw-socket tcp client, page 29](#)
- [raw-socket tcp idle-timeout, page 30](#)
- [raw-socket tcp server, page 31](#)
- [raw-socket udp connection, page 32](#)
- [show raw-socket tcp detail, page 33](#)
- [show raw-socket tcp sessions, page 35](#)
- [show raw-socket tcp statistics, page 36](#)
- [show raw-socket udp detail, page 37](#)
- [show raw-socket udp sessions, page 39](#)
- [show raw-socket udp statistics, page 40](#)

clear raw-socket statistics

To clear Raw Socket Transport statistics for a specific TTY interface or for all asynchronous interfaces, use this command:

```
clear raw-socket statistics { tty_line_num | all }
```

Syntax Description

<i>tty_line_num</i>	The asynchronous TTY line number, a value between 3 and 136.
all	All asynchronous TTY line numbers.

Usage Guidelines

After running this command, run the **show raw-socket tcp statistics** or **show raw-socket udp statistics** command to verify that the raw statistics have been cleared.

Examples

This is an example of the **clear raw-socket statistics** command:

```
router# show raw-socket tcp statistics
----- Network-Serial
Statistics-----
Interface tty  vrf_name  sessions  network_in_bytes  network_out_bytes  network_to_tty_frames
tty_to_network_frames
  0/0/0  3  -----  1  168264  5153864  6232
160707
  0/0/1  4  -----  1  168264  5153864  6232
159407
  0/0/6  9  -----  1  0  0  0
0
router# show raw-socket udp statistics
-----Network-Serial Statistics
-----
Interface tty  vrf_name  sessions  network_in_bytes  network_out_bytes  network_to_tty_frames
tty_to_network_frames
  0/2/6  41  -----  1  238135496  238125328  59533874
59531332

router# clear raw-socket statistics all
router# show raw-socket tcp statistics
-----Network-Serial Statistics
-----
Interface tty  vrf_name  sessions  network_in_bytes  network_out_bytes  network_to_tty_frames
tty_to_network_frames
  0/0/0  3  -----  1  0  0  0
0
  0/0/1  4  -----  1  0  0  0
0
  0/0/6  9  -----  1  0  0  0
0
router# show raw-socket udp statistics
-----Network-Serial Statistics
-----
Interface tty  vrf_name  sessions  network_in_bytes  network_out_bytes  network_to_tty_frames
tty_to_network_frames
  0/2/6  41  -----  1  0  0  0
0
```


debug raw-socket driver event

To turn Raw Socket Transport driver event debugging on, use this command:

debug raw-socket driver event

Usage Guidelines

Issues you can debug with this command include actions related to:

- Packet timer
- Packet length



Caution

Unless for debugging purposes, we do not recommend turning this command on because it slows down console access and impacts CPU performance, especially in cases of heavy traffic.

To turn this debug command off, run one of these commands:

- **no debug raw-socket** driver event
- **no debug all**

Examples

This example shows the output of the **debug raw-socket driver event** command:

```
router# debug raw-socket driver event
*Jun 15 16:12:28.791: rawsocket_get_pak[tty 3]: get pak buffer from pool (0x3168CE24)
*Jun 15 16:12:28.791: rawsocket_get_pak[tty 3]: successfully get pak (0x2A842C70)
*Jun 15 16:12:28.791: rawsocket_tcp_to_async[tty 3]: put pak(0x2A842C70) to queue
*Jun 15 16:12:28.791: rawsocket_async_output[tty 3]: putpkt(0x2A842C70) to physical async
line
*Jun 15 16:12:29.083: rawsocket_rx_service[tty 3]: recieved 32 bytes from serial interface
*Jun 15 16:12:29.083: rawsocket_rx_service[tty 3]: Accumulated length(32) met configured
pak-length(32)
*Jun 15 16:12:29.083: rawsocket_get_pak[tty 3]: get pak buffer from pool (0x3168CE24)
*Jun 15 16:12:29.083: rawsocket_get_pak[tty 3]: successfully get pak (0x316A0DA8)
*Jun 15 16:12:29.083: rawsocket_async_to_tcp[tty 3]: send pak(0x316A0DA8) to rawsocket
```

debug raw-socket driver packet

To turn on debugging for issues related to Raw Socket Transport driver packets, use this command:

```
debug raw-socket driver packet
```

Usage Guidelines

Issues related to Raw Socket Transport driver packets include packetizing actions based on some or all of the following:

- Packet length
- Packet timer
- Special character



Caution

Unless for debugging purposes, we do not recommend turning this command on because it slows down console access and impacts CPU performance, especially in cases of heavy traffic.

To turn this debug command off, run one of these commands:

- **no debug raw-socket driver packet**
- **no debug all**

Examples

This example shows the output of the **debug raw-socket driver packet** command:

```
router# debug raw-socket driver packet
*Jun 15 16:09:48.775: rawsocket_async_output[tty 3]: Received 27 byte from socket...
*Jun 15 16:09:48.775: [Socket --> Async] 05 64 14 C4 04 00 03 00 C7 17 C9 CA 01 3C 02 06
3C 03 06 3C 04 06 3C 01 06 C0 5F
*Jun 15 16:09:48.959: [Async --> Socket] tty(3) Received 32 byte from serial...
*Jun 15 16:09:48.959: [Async --> Socket] 05 64 FF 44 03 00 04 00 48 60 57 CA 81 00 00 01
01 00 00 27 00 00 00 00 00 03 3F A1 01 00 00 27
*Jun 15 16:09:48.991: [Async --> Socket] tty(3) Received 32 byte from serial...
*Jun 15 16:09:48.991: [Async --> Socket] 55 55 55 55 55 55 55 55 55 55 0A 02 6A AF 00 00
22 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
*Jun 15 16:09:49.027: [Async --> Socket] tty(3) Received 32 byte from serial...
*Jun 15 16:09:49.027: [Async --> Socket] 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
BB C3 01 01 01 01 01 01 01 14 05 00 00 1D 00 00 00
*Jun 15 16:09:49.059: [Async --> Socket] tty(3) Received 32 byte from serial...
```

debug raw-socket tcp event

To debug issues related to Raw Socket Transport TCP and UDP sessions, use this command to turn TCP and UDP event debugging on:

debug raw-socket tcp event

Usage Guidelines

Issues that you can debug with this command include:

- Interface and connection registration
- TCP socket establish/release information
- Configuration event
- Retry information



Caution

Unless for debugging purposes, we do not recommend turning this command on because it slows down console access and impacts CPU performance, especially in cases of heavy traffic.

To turn this debug command off, run one of these commands:

- **no debug raw-socket tcp event**
- **no debug all**

Examples

This example shows the output of the **debug raw-socket tcp event** command:

```
router# debug raw-socket tcp event
Raw TCP Event Trace debugging is on
router#
router#
router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
router(config)#line 0/0/0
router(config-line)#no raw-socket tcp client 1.1.1.1 1
router(config-line)#
*May 17 04:12:04.831: Raw Socket: stopping TCP client on interface 0/0/0 tty 3
*May 17 04:12:04.831: Raw Socket: --> server addr: 1.1.1.1 1, local addr: any_ip any_port
*May 17 04:12:04.835: Raw Socket: released connection socket 0 on virtual line 0
*May 17 04:12:04.835: Raw Socket: unregistered interface 0/0/0 tty 3 from virtual line 0
*May 17 04:12:04.835: Raw Socket: unregistered config event at 0x30E28C2C
router(config-line)#
router(config-line)#
router(config-line)#raw-socket tcp client 1.1.1.1 1
router(config-line)#
*May 17 04:13:10.919: Raw Socket: starting TCP client on interface 0/0/0 tty 3
*May 17 04:13:10.919: Raw Socket: --> server addr: 1.1.1.1 1, local addr: any_ip 1
*May 17 04:13:10.919: Raw Socket: registered config event at 0x30E2874C
*May 17 04:13:11.023: Raw Socket: executing config event at 0x30E2874C retry count (0)
*May 17 04:13:11.023: Raw Socket: start client: interface 0/0/0 tty 3 server addr 1.1.1.1
1
*May 17 04:13:11.023: Raw Socket: connecting to server 1.1.1.1 1
*May 17 04:13:11.023: Raw Socket: created connection socket 0 successfully
*May 17 04:13:11.023: Raw Socket: registered interface 0/0/0 tty 3 to virtual line 0
*May 17 04:13:11.023: Raw Socket: registered client connection 0 on virtual line 0
*May 17 04:13:11.023: Raw Socket: established socket 0 on virtual line 0
router(config-line)#
```

```
router(config-line)#  
router(config-line)#end  
router#no debug raw-socket tcp event  
Raw TCP Event Trace debugging is off
```


encapsulation raw-tcp

To configure an asynchronous interface to run in raw TCP encapsulation, use this command:

```
encapsulation raw-tcp
```

Usage Guidelines

To use this command, switch to the Interface Configuration mode (config-if).

Examples

This example shows how to use the **encapsulation raw-tcp** command:

```
router# configure terminal  
router(config)# interface async 1/1  
router(config-if)# encapsulation raw-tcp
```

encapsulation raw-udp

To configure an asynchronous interface to run in raw UDP encapsulation, use this command:

```
encapsulation raw-udp
```

Usage Guidelines

To use this command, switch to the Interface Configuration mode (config-if).

Examples

This example shows how to use the **encapsulation raw-udp** command:

```
router# configure terminal  
router(config)# interface async 1/1  
router(config-if)# encapsulation raw-udp
```

raw-socket mode best-effort

To enable best-effort mode for the serial line, use this command:

```
raw-socket mode best-effort
```

Usage Guidelines

Use best-effort mode for faster transmission of serial data to the Raw Socket peer. Older packets are dropped from the head of the queue when the queue is full.

To use this command, switch to the Line Configuration mode (config-line).

Examples

```
router# configure terminal  
router(config)# line 1/1  
router(config-line)# raw-socket mode best-effort  
router(config-line)# exit  
router(config)#
```

raw-socket packet-length

To set the packet length that the asynchronous serial driver uses to packetize the serial bytes into TCP or UDP frames, use this command:

```
raw-socket packet-length length
```

Syntax Description

<i>length</i>	Packet length in bytes (2–1400).
---------------	----------------------------------

Usage Guidelines

To use this command, switch to the Line Configuration mode (config-line).

Examples

This example shows how to use the **raw-socket packet-length** command:

```
router# configure terminal  
router# line 1/1  
router(config-line)# raw-socket packet-length 827
```

raw-socket packet-timer

To set the packet timer that the asynchronous serial driver uses when packetizing serial bytes into TCP or UDP frames, use this command:

```
raw-socket packet-timer packet_timer
```

Syntax Description

<i>packet_timer</i>	The timer value in ms to trigger packetization.
---------------------	-------------------------------------------------

Usage Guidelines

By default, the serial interface packetizes the serial received bytes in its buffer every 10 ms. However, you can change this value using this command. After creating the packet, the interface sends the packet to the peer serial interface on the server side through the Raw Socket Transport tunnel.

On the utility RTU server side, you can set this value to 3 ms to flush polling requests out as soon as possible. The shorter the timer, the sooner the polling data is sent to the RTU client.

To use this command, switch to the Line Configuration mode (config-line).

Examples

This example shows how to use the **raw-socket packet-timer** command:

```
router# configure terminal  
router(config)# line 1/1  
router(config-line)# raw-socket packet-timer 1000
```

raw-socket special-char

To set the special character that triggers the packetization of incoming bytes, use this command:

```
raw-socket special-char special_character
```

Syntax Description

special_character The ASCII character (0–255).

Usage Guidelines

When the serial interface detects this special character, the interface packetizes the received bytes into a TCP or UDP frame and sends it to the peer through the Raw Socket Transport tunnel.

To use this command, switch to the Line Configuration mode (config-line).

Examples

This example shows how to use the **raw-socket special-char** command:

```
router# configure terminal  
router(config)# line 1/1  
router(config-line)# raw-socket special-char 13
```

raw-socket tcp client

To initiate a Raw Socket Transport TCP client session, use this command:

```
raw-socket tcp client dest_ip_address dest_port [local_ip_address] [local_port]
```

Syntax Description	<i>dest_ip_address</i>	Destination IP address and port number of the remote TCP server.
	<i>dest_port</i>	
	<i>local_ip_address</i>	(Optional) Local IP address and port number that the client can also bind to.
	<i>local_port</i>	

Usage Guidelines To use this command, switch to the Line Configuration mode (config-line).

Examples This example shows how to use the **raw-socket tcp client** command:

```
cgr2010-master-2066#configure terminal  
cgr2010-master-2066(config)#line 1/1  
cgr2010-master-2066(config-line)#raw-socket tcp client 10.0.0.1 4000 10.0.20.30 6000
```

raw-socket tcp idle-timeout

To set the Raw Socket Transport TCP session timeout for an asynchronous line interface, use this command:

```
raw-socket tcp idle-timeout session_timeout
```

Syntax Description	<i>session_timeout</i>	Currently configured session idle timeout in minutes.
---------------------------	------------------------	-------------------------------------------------------

Usage Guidelines This timeout setting applies to all Raw Socket Transport TCP sessions under this particular asynchronous line. By default, the session idle timeout is 5 minutes.

To use this command, switch to the Line Configuration mode (config-line).

Examples This example shows how to use the **raw-socket tcp idle-timeout** command:

```
cgr2010-master-2066#configure terminal  
cgr2010-master-2066(config)#line 1/1  
cgr2010-master-2066(config-line)#raw-socket tcp idle-timeout 120
```

raw-socket tcp server

To start the Raw Socket Transport TCP server for an async asynchronous line interface, use this command:

```
raw-socket tcp server port [ip_address]
```

Syntax Description

<i>port</i>	Server port number the server listens on.
<i>ip_address</i>	(Optional) Server IP address.

Usage Guidelines

To use this command, switch to the Line Configuration mode (config-line).

Examples

This example shows how to use the **raw-socket tcp server** command:

```
cgr2010-master-2066#configure terminal  
cgr2010-master-2066(config)#line 1/1  
cgr2010-master-2066(config-line)#raw-socket tcp server 5000 10.0.0.30
```

raw-socket udp connection

To initiate a Raw Socket Transport UDP session, use this command:

```
raw-socket udp connection dest_ip_address dest_port local_port [local_ip_address]
```

Syntax Description

<i>dest_ip_address</i>	Destination IP address of the remote UDP peer.
<i>dest_port</i>	Destination port number from 1 to 65535.
<i>local_port</i>	Local port number from 1 to 65535.
<i>local_ip_address</i>	(Optional) Local IP address.

Usage Guidelines

To use this command, switch to the Line Configuration mode (config-line).

Examples

This example shows how to use the **raw-socket udp connection** command:

```
router# configure terminal  
router(config)# line 1/1  
router(config-line)# raw-socket udp connection 192.168.0.2 5000 7000  
router(config-line)# exit  
router(config)#
```

show raw-socket tcp detail

To display the details of Raw Socket Transport TCP activity, mainly for debugging purposes, use this command:

show raw-socket tcp detail

Usage Guidelines

This command displays:

- Line Registration and Connections
 - Current status for each virtual line (up to 32 lines), including the asynchronous interface and VRF currently registered with this line.
 - Server status. If the status is on, this line accepts call-in sessions.
 - Server master socket (listening socket).
 - Server listening port.
 - Total number of Raw Socket Transport TCP sessions currently running for this asynchronous interface.

Under each line there can be up to 32 sessions. For each session, this command displays the following:

- [out]—Indicates that the session was initiated by this line interface (referred to as *caller*).
 - [in]—Indicates that the session was initiated by the remote peer (referred to as *called*).
 - l_ip—Local IP address for this session.
 - l_port—Local port number.
 - d_ip—Destination IP address.
 - d_port—Destination port number.
 - w_err—Number of socket write errors.
- Socket Mapping

Displays information about each socket and its connection details.
 - Configuration Event List

Displays all the Raw Socket Transport TCP configurations and retry information.

Examples

This example shows the output of the **show raw-socket tcp detail** command:

```
#show raw-socket tcp detail
----- Line Registration and Connections -----
Line 00, if: 0/2/0 vrfname: -----tty: 35 , server status: on, socket: 0, listen port: 5008 , total
sessions: 0

----- Socket Mapping -----
Socket  status  interface  tty   vrf_name  hwidb   local_ip_addr  local_port  dest_ip_addr  dest_port
protocol
[ 0] listening  0/2/0    35                2B33EAF0   -----    5008          -----    ----
TCP

----- Configuration Event List -----
```

■ show raw-socket tcp detail

Event_addr	initiator	event_action	local_ip_addr	local_port	dest_ip_addr	dest_port	retry_count
protocol							
0x2AE56C7C	0/2/0	start server	-----	5008	-----	-----	0
TCP							

show raw-socket tcp sessions

To display Raw Socket Transport TCP session details, use this command:

```
show raw-socket tcp sessions
```

Usage Guidelines

The returned raw TCP session details include this information

Parameter	Description
Interface	Asynchronous serial interface (and its TTY number) associated with this TCP session.
tty	
vrf_name	Name of the VRF associated with this TCP session.
socket	TCP socket number.
mode	The mode: <ul style="list-style-type: none"> client—Indicates that this interface is the caller. server—Indicates that this interface is being called.
local_ip_addr	Local IP address.
local_port	Local port number.
dest_ip_addr	Destination IP address.
dest_port	Destination port number.
up_time	Session up time.
idle_time	Session idle time.
timeout	Currently configured session idle timeout.

After the session idle time reaches the specified idle timeout, the session is removed. The client reconnects to the server based on the retry mechanism.

Examples

This example shows the output of the **show raw-socket tcp sessions** command:

```
router# show raw-socket tcp sessions
```

```
----- TCP Sessions
-----
Interface tty  vrf_name  socket  mode  local_ip_addr  local_port  dest_ip_addr  dest_port  up_time
idle_time/timeout
0/0/0      3         -----  0      client      100.0.0.2    49574       3.3.3.3       2012      17:22:54
00:00:06/120min
0/0/1      4         -----  1      client      100.0.0.2    47584       3.3.3.3       2012      17:22:54
00:00:06/120min
0/2/0      35        -----  2      server      -----      5008        listening     ----      -----
-----
```

show raw-socket tcp statistics

To display Raw Socket Transport TCP statistics for each asynchronous interface, use this command:

```
show raw-socket tcp statistics
```

Usage Guidelines

The displayed statistics include:

- Interface TTY number.
- VRF name.
- Total Raw Socket TCP sessions currently running for this interface.
- Total bytes received from remote peers to this interface.
- Total output bytes from this interface to remote peers.
- Total packetized frames from remote peers to this interface.
- Total packetized frames from this interface to remote peers.

Examples

This example shows the output of the **show raw-socket tcp statistics** command:

```
#show raw-socket tcp statistics
```

```
router# show raw-socket tcp statistics
```

```
----- Network-Serial
Statistics-----
Interface tty  vrf_name  sessions  network_in_bytes  network_out_bytes  network_to_tty_frames
tty_to_network_frames
 0/0/0  3  -----  1          168264          5153864          6232
160707
 0/0/1  4  -----  1          168264          5153864          6232
159407
 0/0/6  9  -----  1           0              0              0
0
```

show raw-socket udp detail

To display the details of Raw Socket Transport UDP activity, mainly for debugging purposes, use this command:

show raw-socket udp detail

Usage Guidelines

This command displays:

- Line Registration and Connections
 - Current status for each virtual line (up to 32 lines), including the asynchronous interface currently registered with this line.
 - Server status. The status is always off for UDP.
 - Server master socket (listening socket).
 - Server listening port.
 - Total number of Raw Socket Transport UDP sessions currently running for this asynchronous interface.

Under each line there can be up to 32 sessions. For each session, this command displays the following:

- [out]—Indicates that the session was initiated by this line interface (referred to as *caller*).
 - [in]—Indicates that the session was initiated by the remote peer (referred to as *called*).
 - l_ip—Local IP address for this session.
 - l_port—Local port number.
 - d_ip—Destination IP address.
 - d_port—Destination port number.
 - w_err—Number of socket write errors.
- Socket Mapping

Displays information about each socket and its connection details.
 - Configuration Event List

Displays all the Raw Socket Transport UDP configurations and retry information.

Examples

This example shows the output of the **show raw-socket udp detail** command:

```
#show raw-socket udp detail
----- Line Registration and Connections -----
Line 00, if: 0/0/0 vrfname: -----tty: 3 , server status: off, socket: ---, listen port: -----, total
sessions: 1
  <--[out] Session 0 , socket: 0, l_ip: 0.0.0.0, l_port: 6000 , d_ip: 2.2.2.2, d_port: 7000 w_err: 0

----- Socket Mapping -----
Socket status interface tty vrf_name hwidb local_ip_addr local_port dest_ip_addr dest_port
protocol
[ 0] connected 0/0/0 3 4070DB8C 0.0.0.0 6000 2.2.2.2 7000
UDP
----- Configuration Event List -----
```

Event_addr	initiator	event_action	local_ip_addr	local_port	dest_ip_addr	dest_port	retry_count
protocol							
0x2A5E0F90	0/0/0	start client	-----	6000	172.27.164.30	7000	0
UDP							

show raw-socket udp sessions

To display Raw Socket Transport UDP session details, use this command:

```
show raw-socket udp sessions
```

Usage Guidelines

The returned raw UDP session details include this information:

Parameter	Description
Interface tty	Asynchronous serial interface (and its TTY number) associated with this UDP session.
vrf_name	Name of the VRF associated with this UDP session.
socket	UDP socket number.
mode	The mode is always client for UDP sessions.
local_ip_addr	Local IP address.
local_port	Local port number.
dest_ip_addr	Destination IP address.
dest_port	Destination port number.
up_time	Session up time.
idle_time	Session idle time.
timeout	Currently configured session idle timeout.

After the session idle time reaches the specified idle timeout, the session is removed. The client reconnects to the server based on the retry mechanism.

Examples

This example shows the output of the **show raw-socket udp sessions** command:

```
router# show raw-socket udp sessions

----- UDP Sessions
-----
Interface tty  vrf_name  socket  mode  local_ip_addr  local_port  dest_ip_addr  dest_port  up_time
idle_time/timeout
 0/2/6   41   -----   14   client   0.0.0.0     7000     3.3.3.3     6000     5d21h
4d04h/5 min
```

show raw-socket udp statistics

To display Raw Socket Transport UDP statistics for each asynchronous interface, use this command:

```
show raw-socket udp statistics
```

Usage Guidelines

The displayed statistics include:

- Interface TTY number.
- VRF name.
- Total Raw Socket UDP sessions currently running for this interface.
- Total bytes received from remote peers to this interface.
- Total output bytes from this interface to remote peers.
- Total packetized frames from remote peers to this interface.
- Total packetized frames from this interface to remote peers.

Examples

This example shows the output of the **show raw-socket udp statistics** command:

```
#show raw-socket udp statistics
```

```
----- Network-Serial
Statistics-----
Interface tty  vrf_name  sessions  network_in_bytes  network_out_bytes  network_to_tty_frames
tty_to_network_frames
  0/2/6  41  -----  1  238135496  238125328  59533874
59531332
```

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at: <http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

No combinations are authorized or intended under this document.

© 2014 Cisco Systems, Inc. All rights reserved.

