

Raw Socket Transport

This section contains the following topics:

- Raw Socket Transport Overview, on page 1
- Information About Raw Socket Transport, on page 1
- Prerequisites, on page 4
- Guidelines and Limitations, on page 4
- Default Settings, on page 4
- Configuring Raw Socket Transport, on page 4
- Rawsocket Keepalive Configuration CLI, on page 10
- Verifying Configuration, on page 11
- Raw Socket Transport Configuration Examples, on page 11

Raw Socket Transport Overview

Raw Socket Transport transports streams of characters from one serial interface to another over an IP network for utility applications.

This document describes Raw Socket Transport for the IR1800 and provides a reference section describing the Raw Socket Transport commands.

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.

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

- TCP as the transport protocol, with built-in auto TCP connection retry mechanism.
- Up to 32 TCP sessions.

- Interface configuration as a server, client, or a combination of both.
- One server 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.

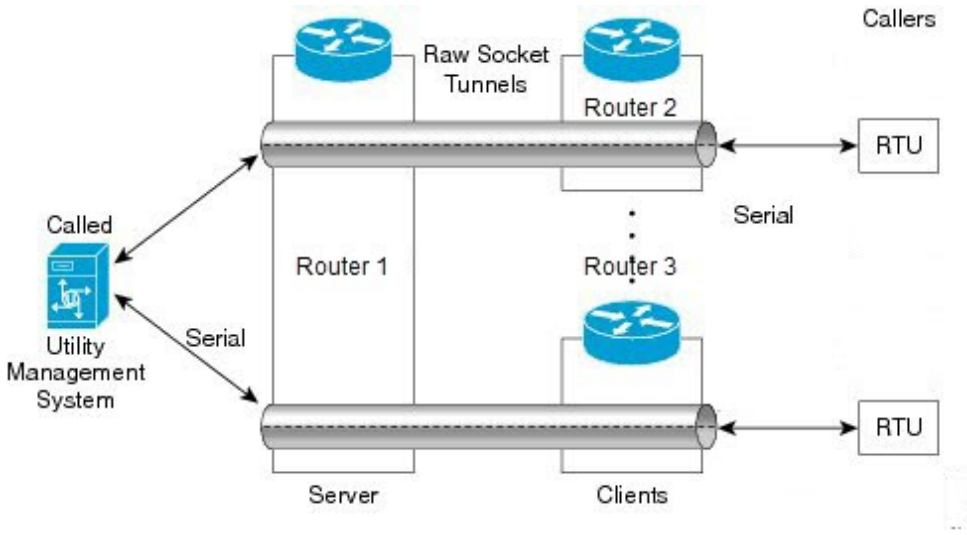
This section includes the following topics:

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 IR1800 can initiate up to 32 TCP sessions to Raw Socket servers, which can be other IR1800 routers or third-party devices.

The following figure 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 IR1800 routers. One IR1800 router (Router 1) acts as a Raw Socket server, listening for TCP connection requests from the other IR1800 routers (Router 2 and Router 3), 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.



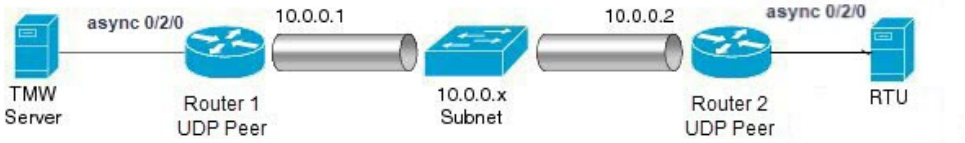
Note When you configure the serial link interface on the router as a server, the interface's peer is the serial link interface on the client router and vice versa.

UDP Transport

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

The following figure 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 routers (Router 1 which is an IR1800 and Router 2 which is an IR807) 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.



Serial Data Processing

When the default serial protocol, Asynchronous Communication Protocol, is used, 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 IR1800 to transmit the serial data to the peer. Once the IR1800 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 IR1800 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 IR1800 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 IR1800 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 IR1800 packetizes the accumulated data and sends it to the Raw Socket peer.

See the "Configuring Common Raw Socket Line Options" procedure on page 6 for information about configuring the processing options.

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, 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

The guidelines and limitations are as given:

- 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.
- The `access-class <acl> in` command is unsupported for serial raw-socket line configurations on IR platforms using IOS-XE.
- When an access list is applied to a raw-socket-enabled serial interface, traffic is processed according to the configured permit and deny entries.

Default Settings

Feature	Default Setting
Raw Socket Transport	Disabled.
Packet length	No packet length is configured.
Serial Protocol	Asynchronous Communication Protocol
Packet timeout	15 ms.
Special character	No special character is configured.
Raw Socket mode	Best-effort mode is off, not supported on the IR1801.
TCP idle timeout	5 minutes.

Configuring Raw Socket Transport

This section includes the following topics:

Enabling Raw Socket Transport on the Serial Interface

To enable Raw Socket Transport on the IR1800 router, you must first enable an asynchronous serial port and enable Raw Socket TCP or UDP encapsulation for that port.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface async0** */slot /port*	Enters the interface command mode for the async slot/port.
Step 3	**no ip address**	Disables IP processing on the interface.
Step 4	Do one of the following: • **encapsulation raw-tcp** • **encapsulation raw-udp**	Enables Raw Socket TCP encapsulation or UDP encapsulation for the serial port.

Enable Serial Port Example

This example shows how to enable serial port 0/2/0 and how to enable Raw Socket TCP encapsulation on that port.

```
router# configure terminal
router(config)# interface async0/2/0
router(config-if)# no ip address
router(config-if)# encapsulation raw-tcp
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, on page 4.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**line 0** */slot /port*	Enters line command mode for the serial slot/port.
Step 3	**raw-socket packet-length** *length*	Specifies the packet size that triggers the IR1800 to transmit the data to the peer. When the IR1800 accumulates this much data in its buffer, it packetizes the data and forwards it to the Raw Socket peer. *length*— 2 to 1400 bytes.

	Command or Action	Purpose
		By default, the packet-length trigger is disabled.
Step 4	**raw-socket packet-timer** *timeout*	Specifies the maximum time in milliseconds the IR1800 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. *timeout* —3 to 1000 ms. The default is 15 ms.
Step 5	**raw-socket spec-char** *ascii_char*	Specifies a character that will trigger the IR1800 to packetize the data accumulated in its buffer and send it to the Raw Socket peer. *ascii_char*— 0 to 255. By default, the special character trigger is disabled.

What to do next

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

Configuring Common Raw Socket Line Options Example

```
router# configure terminal
router(config)# line 0/2/0
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, on page 4.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.

	Command or Action	Purpose
Step 2	**line 0** */slot /port*	Enters line command mode for the serial slot/port.
Step 3	**raw-socket tcp server** *port* [*ip_address*]	Starts the Raw Socket Transport TCP server for an asynchronous line interface. In Raw Socket server mode, the IR1800 listens for incoming connection requests from Raw Socket clients. *port* – Port number the server listens on. *ip_address* – (Optional) Local IP address on which the server listens for connection requests.
Step 4	**raw-socket tcp idle-timeout** *session_timeout*	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. *session_timeout* –Currently configured session idle timeout in minutes. The default is 5 minutes.

What to do next

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

Configuring Common Raw Socket TCP Server 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 0/2/0
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, on page 4.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**line 0** */slot /port*	Enters line command mode for the serial slot/port.
Step 3	**raw-socket tcp client** *dest_ip_address dest_port [local_ip_address] [local_port]*	Specifies settings for Raw Socket Transport TCP client sessions. *dest_ip_address* – Destination IP address of the remote Raw Socket server. *dest_port* – Destination port number to use for the TCP connection to the remote server. *local_ip_address* – (Optional) Local IP address that the client can also bind to. *local_port* – (Optional) Local port number that the client can also bind to.
Step 4	**raw-socket tcp idle-timeout** *session_timeout*	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 is closed. 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. *session_timeout* –Currently configured session idle timeout in minutes. The default is 5 minutes.
Step 5	**tcp keepalive** *interval*	Sets the Raw Socket Transport TCP session keepalive interval for the asynchronous line interface. The router sends keepalive messages based on the configured interval. You may need to configure this interval, for example, when sending raw TCP traffic over a cellular interface. *interval* – Currently configured keepalive interval in seconds. Range is 1-864000 seconds. The default is 1 second.

What to do next

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

Raw Socket TCP Client Example

This example shows how to configure a Raw Socket TCP client for an asynchronous serial line. The IR1800 (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 0/2/0
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)#
```

Raw socket feature enhancement

A raw socket feature enhancement is a network configuration capability that:

- allows the user to input the maximum number of retries available to the write socket,
- provides a range from 1 to 1000 retries with a default of 10 retries, and
- includes a new CLI command **raw-socket tcp max-retries <1-1000>** where <1-1000> is the maximum number of retries.

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, on page 4.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**line 0** */slot /port*	Enters line command mode for the serial slot/port.
Step 3	**raw-socket udp connection** *dest_ip_address dest_port local_port [local_ip_address]*	Specifies settings for Raw Socket Transport UDP connections. *dest_ip_address* – Destination IP address to use for the UDP connection.

	Command or Action	Purpose
		dest_port – Destination port number to use for the UDP connection. *local_port* – Local port number for the UDP connection. *local_ip_address* – (Optional) Local IP address for the UDP connection.

What to do next

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

Raw Socket UDP Connection 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 0/2/0
router(config-line)# raw-socket udp connection 192.168.0.2 5000 7000
router(config-line)# exit
router(config)#
```

Router B

```
router# configure terminal
router(config)# line 0/2/0
router(config-line)# raw-socket udp connection 192.168.0.8 7000 5000
router(config-line)# exit
router(config)#
```

Rawsocket Keepalive Configuration CLI

Rawsocket keepalive for async interfaces is a feature that existed in classic IOS platforms. As part of 17.10.1a, the feature will be extended to IOS-XE based platforms. A new CLI with the following syntax will be added under rawsocket.

```
Router(config-line)#raw-socket tcp keepalive interval
```

CLI Changes

On IOS-XE platforms starting from 17.10.1a, there is a CLI correction and an additional CLI was added as part of raw-socket.

The correction is for the **raw-socket idle timeout** command. There is now an option to configure the timeout based on minutes and seconds, whereas the previous configuration used only minutes.

```
Router(config-line)# raw-socket tcp idle-timeout [0-1440] [<0-59> | cr]
```

The additional CLI is for clearing the raw-socket TCP clients. The command syntax is **clear raw-socket line** *[1-145|tty|x/y/z]* for example:

```
Router# clear raw-socket line 0/2/0
```

Note When initiating clear raw-socket line, raw-socket sessions will be cleared for raw-socket clients from the **show raw-socket tcp sessions** command. Connections will be re-established after a TCP hand-shake, which can be done by doing shut/no shut on TCP connection interface.

Verifying Configuration

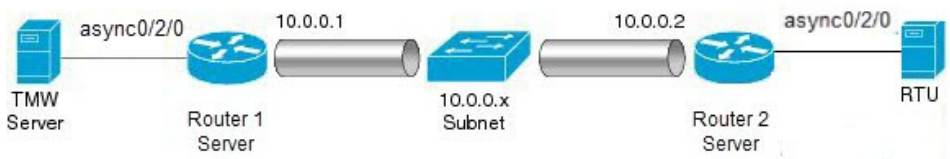
Command	Purpose
show running-config	Shows the configuration of the IR1800, including those features that are active and their settings.
show raw-socket tcp detail	Displays information about Raw Socket Transport TCP activity.
show raw-socket tcp sessions	Displays information about Raw Socket Transport 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.

Raw Socket Transport Configuration Examples

The following sections include Raw Socket Transport configuration examples:

Raw Socket TCP

The following example shows a Raw Socket Transport configuration in which an IR1800 router (Router 1) acts as the server, and another IR809 (Router 2) acts as the client.



The following table displays the configuration of the server and client IR1800s highlighted in the above figure:

IR1800 Server Configuration	IR807 Client Configuration
... interface async0/2/0 no ip address **encapsulation raw-tcp** ! ... line 0/2/0 **raw-socket tcp server 5000 10.0.0.1** **raw-socket packet-timer 3** **raw-socket tcp idle-timeout 5** ...	... interface async0 no ip address **encapsulation raw-tcp** ! interface async1 no ip address **encapsulation raw-tcp** ! ... line 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 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**

Raw Socket UDP Example

This example shows the configuration for a Raw Socket UDP connection between two IR1800 routers:

From Router1

```
interface GigabitEthernet0/1
ip address 192.168.0.8 255.255.255.0
duplex auto
speed auto
interface async0/2/0
no ip address
encapsulation raw-udp
line 0/2/0
raw-socket udp connection 192.168.0.2 2 2
```

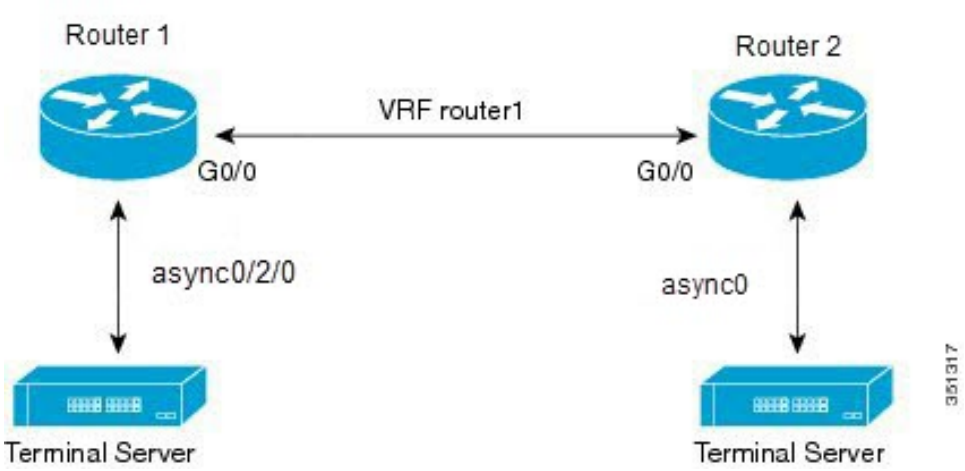
From Router2

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

```
interface async0/2/0
no ip address
encapsulation raw-udp
line 0/2/0
raw-socket udp connection 192.168.0.8 2 2
```

Raw Socket VRF

The following example shows a Raw Socket VRF configuration in which two routers, configured for Raw Socket Transport, connect through a VRF. Router1 is an IR1800, serves as the Raw Socket TCP server, and Router2 is an IR807 serves as the Raw Socket TCP client.



Following are the configurations of Router1 and Router2 as shown in the above figure:

Router1 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 async0/2/0
 vrf forwarding router1
 no ip address
 encapsulation raw-tcp
```

Applying raw-tcp on the line:

```
line 0/2/0
 raw-socket tcp server 5000 4.4.4.4
```

Router2 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 async0
 vrf forwarding router1
 no ip address
 encapsulation raw-tcp
```

Applying raw-tcp on line:

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