# Implementing Host Services and Applications


- Implementing Host Services and Applications, on page 1
- Network Connectivity Tools, on page 1
- Domain Services, on page 13
- File Transfer Services, on page 14
- Cisco inetd, on page 18
- Telnet, on page 18
- Syslog source-interface, on page 19
- HTTP Client Application, on page 19


## Implementing Host Services and Applications

Cisco IOS XR software Host Services and Applications features on the router are used primarily for checking network connectivity and the route a packet follows to reach a destination, mapping a hostname to an IP address or an IP address to a hostname, and transferring files between routers and UNIX workstations.

### Prerequisites for implementing Host Services and Applications

Ensure to install the relevant optional RPM package before using the host services or applications. For example, Install Telnet RPM before using Telnet.

## Network Connectivity Tools

***Table 1: Feature History Table***

| Feature Name | Release Information | Feature Description |
|---|---|---|
| Network Connectivity Tools: Ping and Traceroute | Release 25.4.1 | Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)<br>*This feature is now supported on:<br>• 8011-12G12X4Y-A<br>• 8011-12G12X4Y-D |

| Feature Name | Release Information | Feature Description |
|---|---|---|
| Network Connectivity Tools: Ping and Traceroute | Release 25.1.1 | Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)<br>*This feature is supported on Cisco 8011-4G24Y4H-I routers. |
| Network Connectivity Tools: Ping and Traceroute | Release 24.4.1 | Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*)<br>The **ping** command can be used to verify connectivity between devices, the **traceroute** command can be used to discover the paths packets take to a remote destination and where routing breaks down.<br>*This feature is now supported on:<br>• 8212-48FH-M<br>• 8711-32FH-M<br>• 8712-MOD-M<br>• 88-LC1-36EH<br>• 88-LC1-12TH24FH-E<br>• 88-LC1-52Y8H-EM |

Network connectivity tools enable you to check device connectivity by running traceroutes and pinging devices on the network:

# Ping

The **ping** command is a common method for troubleshooting the accessibility of devices. It uses two Internet Control Message Protocol (ICMP) query messages, ICMP echo requests, and ICMP echo replies to determine whether a remote host is active. The **ping** command also measures the amount of time it takes to receive the echo reply.

The **ping** command first sends an echo request packet to an address, and then it waits for a reply. The ping is successful only if the echo request gets to the destination, and the destination is able to get an echo reply (hostname is alive) back to the source of the ping within a predefined time interval.

The bulk option has been introduced to check reachability to multiple destinations. The destinations are directly input through the CLI. This option is supported for ipv4 destinations only.

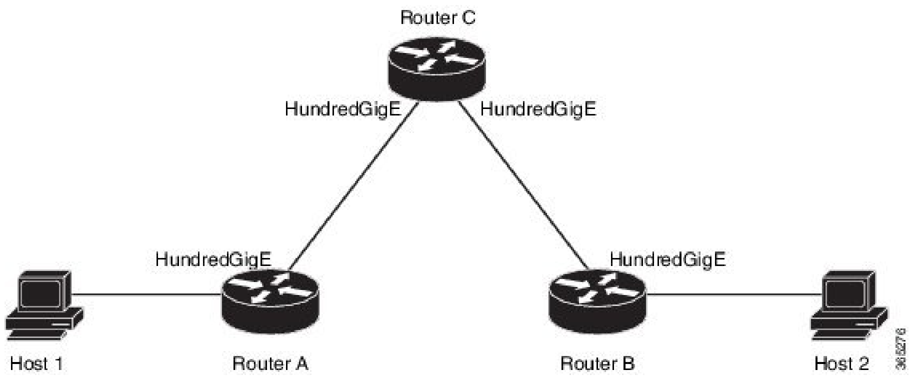
# Checking Network Connectivity

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.

### Configuration for Checking Network Connectivity

The following configuration shows an extended **ping** command sourced from the Router A HundredGigEinterface and destined for the Router B HundredGigE interface. If this ping succeeds, it is an indication that there is no routing problem. Router A knows how to get to the HundredGigEinterface of Router B, and Router B knows how to get to the HundredGigE interface of Router A. Also, both hosts have their default gateways set correctly.

If the extended **ping** command from Router A fails, it means that there is a routing problem. There could be a routing problem on any of the three routers: Router A could be missing a route to the subnet of Router B's interface, or to the subnet between Router C and Router B; Router B could be missing a route to the subnet of Router A's subnet, or to the subnet between Router C and Router A; and Router C could be missing a route to the subnet of Router A's or Router B's Ethernet segments. You should correct any routing problems, and then Host 1 should try to ping Host 2. If Host 1 still cannot ping Host 2, then both hosts' default gateways should be checked. The connectivity between the HundredGigE interface of Router A and the HundredGigE interface of Router B is checked with the extended **ping** command.




With a normal ping from Router A to Router B's HundredGigE interface, the source address of the ping packet would be the address of the outgoing interface; that is the address of the HundredGigE interface, (192.0.2.2). When Router B replies to the ping packet, it replies to the source address (that is, 192.0.2.1). This way, only the connectivity between the HundredGigE interface of Router A (192.0.2.2) and the 10gige interface of Router B (192.0.2.1) is tested.

To test the connectivity between Router A's HundredGigE interface (192.0.2.2) and Router B's interface (192.0.2.1), we use the extended **ping** command. With extended **ping**, we get the option to specify the source address of the **ping** packet.

### Configuration Example

In this use case, the extended **ping** command verifies the IP connectivity between the two IP addresses Router A (192.0.2.2) and Router B (192.0.2.1) .

```
Router# ping 192.0.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.1, timeout is 2 seconds:
```

```
!!!!!
Success rate is 100 percent (5/5)
Router#!!!!!

*/If you do not enter a hostname or an IP address on the same line as the ping command,
the system prompts you to specify the target IP address and several other command parameters.

After specifying the target IP address, you can specify alternate values for the
remaining parameters or accept the displayed default for each parameter /*

Router# ping
Tue Sep 24 02:41:45.739 UTC
Protocol [ipv4]: ipv4
Target IP address: 192.0.2.1
Repeat count [5]: 5
Datagram size [100]: 1
% A decimal number between 36 and 18024.
Datagram size [100]: 36
Timeout in seconds [2]: 1
Interval in milliseconds [10]: 1
Extended commands? [no]: y
Source address or interface: 12.12.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 36-byte ICMP Echos to 192.0.2.1, timeout is 1 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/7 ms
```

### Associated Commands

- ping

## Checking Network Connectivity for Multiple Destinations

The bulk option enables you to check reachability to multiple destinations. The destinations are directly input through the CLI. This option is supported for ipv4 destinations only.

### Configuration Example

Check reachability and network connectivity to multiple hosts on IP networks with the following IP addresses:

- 1: 192.0.2.1
- 2: 198.51.100.1
- 3: 203.0.113.1

```
Router# ping bulk ipv4 input cli batch
*/You must hit the Enter button and then specify one destination address per line*/
Please enter input via CLI with one destination per line and when done Ctrl-D/(exit) to
initiate pings:
1: 192.0.2.1
2: 198.51.100.1
3: 203.0.113.1
4:
Starting pings...
```

```
Target IP address: 192.0.2.1
Repeat count [5]: 5
Datagram size [100]: 1
% A decimal number between 36 and 18024.
Datagram size [100]: 1
% A decimal number between 36 and 18024.
Datagram size [100]: 1000
Timeout in seconds [2]: 1
Interval in milliseconds [10]: 10
Extended commands? [no]: no
Sweep range of sizes? [no]: q
% Please answer 'yes' or 'no'.
Sweep range of sizes? [no]: q
% Please answer 'yes' or 'no'.
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 1000-byte ICMP Echos to 192.0.2.1, vrf is default, timeout is 1 seconds:
!!!!!
Success rate is 100 percent (5/5),
Target IP address: 198.51.100.1
Repeat count [5]:
Datagram size [100]: q
% A decimal number between 36 and 18024.
Datagram size [100]:
Timeout in seconds [2]:
Interval in milliseconds [10]:
Extended commands? [no]:
Sweep range of sizes? [no]:
Sending 5, 100-byte ICMP Echos to 192.0.2.1, vrf is default, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5),
Target IP address:  203.0.113.1
Repeat count [5]: 4
Datagram size [100]: 100
Timeout in seconds [2]: 1
Interval in milliseconds [10]: 10
Extended commands? [no]: no
Sweep range of sizes? [no]: no
Sending 4, 100-byte ICMP Echos to 192.0.2.1, vrf is default, timeout is 1 seconds:
!!!!!
Success rate is 100 percent (4/5),
```

### Associated Commands

- ping bulk ipv4

# Traceroute

Where the **ping** command can be used to verify connectivity between devices, the **traceroute** command can be used to discover the paths packets take to a remote destination and where routing breaks down.

The **traceroute** command records the source of each ICMP "time-exceeded" message to provide a trace of the path that the packet took to reach the destination. You can use the IP **traceroute** command to identify the path that packets take through the network on a hop-by-hop basis. The command output displays all network layer (Layer 3) devices, such as routers, that the traffic passes through on the way to the destination.

The **traceroute** command uses the Time To Live (TTL) field in the IP header to cause routers and servers to generate specific return messages. The **traceroute** command sends a User Datagram Protocol (UDP) datagram to the destination host with the TTL field set to 1. If a router finds a TTL value of 1 or 0, it drops the datagram and sends back an ICMP time-exceeded message to the sender. The traceroute facility determines the address of the first hop by examining the source address field of the ICMP time-exceeded message.

To identify the next hop, the **traceroute** command sends a UDP packet with a TTL value of 2. The first router decrements the TTL field by 1 and sends the datagram to the next router. The second router sees a TTL value of 1, discards the datagram, and returns the time-exceeded message to the source. This process continues until the TTL increments to a value large enough for the datagram to reach the destination host (or until the maximum TTL is reached).

To determine when a datagram reaches its destination, the **traceroute** command sets the UDP destination port in the datagram to a very large value that the destination host is unlikely to be using. When a host receives a datagram with an unrecognized port number, it sends an ICMP port unreachable error to the source. This message indicates to the traceroute facility that it has reached the destination.

## Checking Packet Routes

The **traceroute** command allows you to trace the routes that packets actually take when traveling to their destinations.

### Configuration Example

Trace the route from 192.0.2.1 to 198.51.100.1:

```
Router# traceroute 198.51.100.1
Type escape sequence to abort.
Tracing the route to 198.51.100.1
 1  192.0.2.1 39 msec  *  3 msec

*/If you do not enter a hostname or an IP address on the same line as the traceroute
command, the system prompts you to specify the target IP address and several other
command parameters. After specifying the target IP address, you can specify alternate
values for the remaining parameters or accept the displayed default for each
parameter/*

Router #traceroute
Protocol [ipv4]:
Target IP address: 198.51.100.1
Source address: 192.0.2.1
Numeric display? [no]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:

Type escape sequence to abort.
Tracing the route to 198.51.100.1
 1  192.0.2.1.1 3 msec  *  3 msec
```

### Associated Commands

- traceroute

# Enhanced ICMP error messages for traceroute

Enhanced ICMP error messages for traceroute is a network diagnostic feature that

- adds detailed information to ICMP error messages generated during traceroute operations
- provides specifics about the incoming and outgoing interfaces such as interface name, IP address, and MTU and the next-hop IP address at each hop, and
- enables deeper network path visibility and more effective troubleshooting for administrators.

*Table 2: Feature History Table*

| Feature Name | Release Information | Feature Description |
|---|---|---|
| Enhanced ICMP error messages for traceroute | Release 25.4.1 | Introduced in this release on: Fixed Systems (8200 [ASIC: Q100, Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC:Q200]) ; Modular Systems (8800 [LC ASIC: Q100, Q200, P100])<br><br>You now get detailed insights into each step of the network path. Building on the existing traceroute utility, this feature adds more diagnostic information to ICMP error messages. At each hop, the router shows not only the IP address, but also the incoming and outgoing interfaces and the next hop IP address. This makes it easier to find and fix network issues.<br><br>This feature introduces these changes:<br><br>CLIs:<br>• **icmp ipv4 mpls extended-diagnostics**<br>• **icmp ipv6 mpls extended-diagnostics**<br>• **icmp ipv4 vrf extended-diagnostics permitted-remote-address**<br>• **icmp ipv6 vrf extended-diagnostics permitted-remote-address**<br>• The **extended** keyword is added to the **traceroute** command.<br><br>YANG Data Model:<br>• `Cisco-IOS-XR-um-ip-icmp-cfg`<br>• `Cisco-IOS-XR-traceroute-act`<br><br>(see GitHub, YANG Data Models Navigator) |

## Restriction for enhanced ICMP messages for traceroute

Enhanced ICMP messages for traceroute are not supported on SRv6.

## Benefits of enhanced ICMP error messages for traceroute

Enhanced ICMP error messages for traceroute provide several benefits.

- These messages improve visibility into network paths, which allows network operators to diagnose issues more effectively.
- Enhanced ICMP error messages enable identification of specific interfaces and their characteristics at each hop along the network path.
- They help to solve limitations encountered in complex environments, such as those involving unnumbered interfaces or networks that use only loopback IPv4 addresses.
- These messages offer a clearer understanding of how packets traverse the network, supporting more accurate network analysis and troubleshooting.

## Enable enhanced ICMP error messages for traceroute

Use this procedure to provide extended diagnostic information in ICMP error messages, improving the effectiveness of traceroute for network troubleshooting.

By default, extended ICMP diagnostics are disabled. You must configure which remote addresses or prefixes are permitted to receive enhanced diagnostics. Optionally, enable diagnostics for MPLS traffic or use a wildcard to cover all destinations. This configuration is shared with the ICMP PROBE Utility (RFC 8335). For information on how to configure ICMP PROBE utility refer to Configure ICMP PROBE utility. The implementation of enhanced ICMP error messages for traceroute complies with RFC 5837.

**Before you begin**

- Identify the IP address prefixes (IPv4 or IPv6) for which you want to enable extended diagnostics (up to 1024 prefixes are supported).
- If your network uses MPLS, determine whether extended diagnostics for MPLS traffic should also be enabled.

### Procedure

**Step 1** Enable extended ICMP diagnostics for the desired VRF and IP address or prefix.

**Example:**

```
Router# configure
Router(config)# icmp ipv4 vrf default extended-diagnostics permitted-remote-address 192.168.0.0/16
Router(config-icmp-vrf-diag)# commit
```

**Step 2** (Optional) Enable extended diagnostics for MPLS traffic.

**Example:**

```
Router# configure
Router(config)# icmp ipv4 mpls
```

```
Router(config-icmp-mpls)# extended-diagnostics
Router(config-icmp-mpls-diag)# commit
```

When MPLS support is enabled, incoming MPLS packets are checked against the list of permitted remote prefixes for the default VRF.

The same configuration can also be repeated in IPv6.

**Step 3** Repeat step 1 and 2 on each intermediate router along the traceroute path.

This configuration should be applied to the routers that are expected to generate the enhanced ICMP error messages.

**Step 4** Verify the configuration.

**Example:**

```
Router# show running-config icmp ipv4 vrf extended-diagnostics
```

**Step 5** Display enhanced diagnostic information, if configured and available.

**Example:**

```
Router# traceroute ipv4 192.168.1.10 source Loopback0 extended
ype escape sequence to abort.
Tracing the route to 192.168.1.10

 1  192.168.1.1  16 msec  15 msec  14 msec
     incoming GigabitEthernet0/0/0/1:
       Address: 192.168.1.1
       MTU: 1500
     outgoing GigabitEthernet0/0/0/2:
       Address: 192.168.1.2
       MTU: 1500
     next hop:
       Address: 192.168.1.10

 2  192.168.1.10  21 msec
     incoming GigabitEthernet0/0/0/2:
       Address: 192.168.1.10
       MTU: 1500
```

---

The routers include extended diagnostic information in ICMP error messages for specified remote address prefixes. Enhanced diagnostics are displayed in traceroute output for supported destinations.

### What to do next

Review traceroute output to confirm that diagnostic information is present. Adjust permitted prefixes or enable MPLS diagnostics on additional routers as needed.

# ICMP probe utility

An ICMP probe utility is a network diagnostic tool that

- enables you to query the operational status of local interfaces or directly connected remote neighbors
- provides detailed reports whether the interface is active, and whether IPv4 and IPv6 are running and
- operates via a proxy device, making it useful for troubleshooting when direct, bidirectional connectivity to the probed interface is unavailable.

*Table 3: Feature History Table*

| Feature Name | Release Information | Feature Description |
| --- | --- | --- |
| ICMP probe utility | Release 25.4.1 | Introduced in this release on: Fixed Systems (8200 [ASIC: Q100, Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC:Q200]) ; Modular Systems (8800 [LC ASIC: Q100, Q200, P100])<br><br>This feature introduces a new network diagnostic tool called `probe`, similar to `ping` but with enhanced capabilities. It allows users to query the operational status of local interfaces or directly connected remote neighbors, providing details like probe can report whether the interface is active, and whether IPv4 and IPv6 are running. probe operates through a proxy device, making it suitable for troubleshooting scenarios where direct bidirectional connectivity to the probed interface is not available.<br><br>This feature introduces these changes:<br><br>CLI:<br><br>• **probe** |

ICMP probe utilities expand upon the traditional ping diagnostic tool by delivering enhanced information about network interfaces and neighbors. First introduced in Release 25.4.1, these utilities allow network administrators to diagnose connectivity and status even in complex environments where direct access may not be possible.

## Benefits of using ICMP probe utility

The ICMP probe utility offers significant benefits for enhanced network diagnostics.

- The utility supports both IPv4 and IPv6 probe operations, including cross-address family tests, such as sending an IPv4 probe to assess IPv6 destinations.
- It enables network administrators to gain granular visibility into interface and neighbor health by reporting the state and reachability of each interface and neighbor for every probe.

- The ICMP probe utility helps identify asymmetric routing issues and troubleshooting scenarios that traditional ping commands may not detect, particularly in cases of partial-path or interface-specific problems.
- It provides detailed results that summarize the success and failure of each probe attempt, which assists administrators in rapidly isolating connectivity issues.
- The utility allows users to specify both the source and destination addresses, making it especially well-suited for complex or multi-homed network environments.

## Configure ICMP probe utility

Use this procedure to setup the ICMP probe feature to diagnose interface or neighbor reachability on your device.

The RFC 8335 probe feature, disabled by default, directly re-uses the configuration mechanism established for RFC 5837 ICMP Error Message Extensions, often referred to as extended diagnostics. To enable probe functionality, users must explicitly define a per-VRF inclusion list of IPv4 or IPv6 prefixes. This shared configuration ensures that the permitted remote addresses specified for extended diagnostics will simultaneously govern the destination IP addresses to which probe replies may be sent, thereby restricting both enhanced ICMP error messages and probe responses to authorized targets.

For information on icmp enhanced messages for traceroute refer to Enhanced ICMP error messages traceroute.

### Procedure

**Step 1** Enable the ICMP probe feature and define permitted remote addresses.

**Example:**

```
Router# configure
Router(config)# icmp ipv4 vrf default extended-diagnostics permitted-remote-address 192.168.1.0/24
Router(config-icmp-vrf-diag)# commit
```

**Note**
You can specify up to 1024 IPv4 and IPv6 prefixes.

**Step 2** Initiate a local or remote probe to test reachability.

For a local probe (local to the proxy device):

**Example:**

```
Router# probe 192.168.1.10 source Loopback0 by-name BVI1
probe 192.168.1.9 -> 192.168.1.10
8 bytes from 192.168.1.10: icmp_seq=0 ttl=64 time=1.202 ms. Interface=BVI1
8 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=1.089 ms. Interface=BVI1
8 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=1.115 ms. Interface=BVI1
```

For a remote probe testing a neighbor's reachability:

**Example:**

```
Router# probe 2001:db8::1 source GigabitEthernet0/0/0/0 by-remote-address fe80::1
probe 2001:db8::2 -> 2001:db8::1
8 bytes from 2001:db8::1: icmp_seq=0 hlim=254 time=25.213 ms. Neighbor-state=Reachable
8 bytes from 2001:db8::1: icmp_seq=1 hlim=254 time=22.986 ms. Neighbor-state=Reachable
8 bytes from 2001:db8::1: icmp_seq=2 hlim=254 time=22.576 ms. Neighbor-state=Reachable
```

**Note**
probe replies are sent only if the source address matches an entry in the permitted remote address list.

The device responds to probe commands, reporting status details such as interface activity or neighbor reachability.

# Domain Services

Cisco IOS XR software domain services acts as a Berkeley Standard Distribution (BSD) domain resolver. The domain services maintains a local cache of hostname-to-address mappings for use by applications, such as Telnet, and commands, such as **ping** and **traceroute**. The local cache speeds the conversion of host names to addresses. Two types of entries exist in the local cache: static and dynamic. Entries configured using the **domain ipv4 host** or **domain ipv6 host** command are added as static entries, while entries received from the name server are added as dynamic entries.

The name server is used by the World Wide Web (WWW) for translating names of network nodes into addresses. The name server maintains a distributed database that maps hostnames to IP addresses through the DNS protocol from a DNS server. One or more name servers can be specified using the **domain name-server** command.

When an application needs the IP address of a host or the hostname of an IP address, a remote-procedure call (RPC) is made to the domain services. The domain service looks up the IP address or hostname in the cache, and if the entry is not found, the domain service sends a DNS query to the name server.

You can specify a default domain name that Cisco IOS XR software uses to complete domain name requests. You can also specify either a single domain or a list of domain names. Any IP hostname that does not contain a domain name has the domain name you specify appended to it before being added to the host table. To specify a domain name or names, use either the **domain name** or **domain list** command.

## Configuring Domain Services

DNS-based hostname-to-address translation is enabled by default. If hostname-to-address translation has been disabled using the **domain lookup disable** command, re-enable the translation using the **no domain lookup disable** command.

### Configuration Example

Define a static hostname-to-address mapping. Associate (or map) the IPv4 addresses (192.0.2.1 and 10.2.0.2 198.51.100.1) with two hosts. The host names are host1 and host2.

```
Defining the Domain Host
==================================
Router# configure
Router(config)#domain ipv4 host host1 192.168.7.18
Router(config)#domain ipv4 host host2 10.2.0.2 192.168.7.33
Router(config)#commit

Defining the Domain Name
==========================
*/Define cisco.com as the default domain name/*
Router#configure
Router(config)#domain name cisco.com
```

```
Router(config)#commit

Specifying the Addresses of the Name Servers
============================================
*/Specify host 192.168.1.111 as the primary name server
and host 192.168.1.2 as the secondary server/*
Router#configure
Router(config)#domain name-server 192.168.1.111
Router(config)#domain name-server 192.168.1.2
Router(config)#commit
```

### Verification

```
Router#show hosts
Default domain is cisco.com
Name/address lookup uses domain service
Name servers: 192.168.1.111, 192.168.1.2

Host                       Flags    Age(hr) Type Address(es)
host2                    (perm, OK)   0      IP  10.2.0.2
                                                 192.168.7.33
host1                    (perm, OK)   0      IP  192.168.7.18
```

### Associated Commands

- domain name
- domain list
- domain name-server
- domain ipv4 host
- domain ipv6 host

# File Transfer Services

File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP), remote copy protocol (rcp) rcp clients, and Secure Copy Protocol (SCP) are implemented as file systems or resource managers. For example, path names beginning with tftp:// are handled by the TFTP resource manager.

The file system interface uses URLs to specify the location of a file. URLs commonly specify files or locations on the WWW. However, on Cisco routers, URLs also specify the location of files on the router or remote file servers.

When a router crashes, it can be useful to obtain a copy of the entire memory contents of the router (called a core dump) for your technical support representative to use to identify the cause of the crash. SCP, FTP, TFTP, rcp can be used to save the core dump to a remote server.

## FTP

File Transfer Protocol (FTP) is part of the TCP/IP protocol stack, which is used for transferring files between network nodes. FTP is defined in RFC 959.

## Configuring a Router to Use FTP Connections

You can configure the router to use FTP connections for transferring files between systems on the network. You can set the following FTP characteristics:

- Passive-mode FTP
- Password
- IP address

### Configuration Example

Enable the router to use FTP connections. Configure the software to use passive FTP connections, a password for anonymous users, and also specify the source IP address for FTP connections.

```
Router#configure
Router(config)#ftp client passive

Router(config)#ftp client anonymous-password xxxx
Router(config)#ftp client source-interface HundredGigE 0/0/0/0
Router(config)#commit
```

### Running Configuration

```
Router#show running-config ftp client passive
ftp client passive

Router#show running-config ftp client anonymous-password xxxx
ftp client anonymous-password xxxx
Router#show running-config ftp client source-interface HundredGigE 0/0/0/0
```

### Associated Commands

- ftp client passive
- ftp client anonymous-password
- ftp client source-interface

# TFTP

Trivial File Transfer Protocol (TFTP) is a simplified version of FTP that allows files to be transferred from one computer to another over a network, usually without the use of client authentication (for example, username and password).

## Configuring a Router to Use TFTP Connections

### Configuration Example

Configure the router to use TFTP connections and set the IP address of the HundredGigE 0/0/0/0 as the source address for TFTP connections:

```
Router#configure
Router(config)#tftp client source-interface HundredGigE 0/0/0/0
Router(config)#commit
```

#### Running Configuration

```
Router#show running-config tftp client source-interface HundredGigE 0/0/0/0
tftp client source-interface HundredGigE 0/0/0/0
```

#### Verification

```
Router#show cinetd services
Vrf Name Family  Service  Proto Port ACL max_cnt  curr_cnt  wait   Program Client Option

default  v4     tftp udp    69      unlimited  0       wait   tftpd    sysdb disk0:

default  v4      telnet   tcp    23      10         0       nowait telnetd  sysdb
```

#### Associated Commands

- tftp client source-interface type
- show cinetd services

## TFTP Server

*Table 4: Feature History Table*

| Feature Name | Release Information | Feature Description |
|---|---|---|
| TFTP Server support in Router | Release 7.5.4 | You can now configure the Cisco 8000 Series Router as a TFTP server to serve requests from client routers. This capability reduces costs and time delays in your network by eliminating the redundancy of having a machine that acts only as a server on every network segment.<br>This feature introduces the following commands:<br>• **tftp server**<br>• **show cinetd services** |

It’s expensive and inefficient to have a machine that acts only as a server on every network segment. However, when you don’t have a server on every segment, your network operations can incur substantial time delays across network segments. You can configure a router to serve as a TFTP server to avoid such time delays while you use your router for its regular functions.

A router that you configure as a TFTP server enables the router to serve requests from client routers. It includes services such as providing client routers with system image or router configuration files from its flash memory. You can also configure the router to respond to other types of service requests.

### Configuring a Router as a TFTP Server

The server and client router must be able to reach each other before the TFTP function can be implemented. Verify this connection by testing the connection between the server and client router using the **ping** command.

This task allows you to configure the router as a TFTP server so other devices acting as TFTP clients are able to read and write files from and to the router under a specific directory, such as slot0:, /tmp, and so on (TFTP home directory).

**Note** For security reasons, the TFTP server requires that a file must already exist for a write request to succeed.

### Configuration Example

Configure the router (home directory disk0:) as the TFTP server.

```
Router#configure
Router(config)#tftp ipv4 server homedir disk0
Router(config)#commit
```

### Running Configuration

```
Router#show running-config tftp ipv4 server homedir disk0:
tftp vrf default ipv4 server homedir disk0:
```

### Verification

```
Router#show cinetd services
Vrf Name Family Service     Proto Port ACL  max_cnt  curr_cnt wait  Program Client Option

default  v4     tftp    udp   69      unlimited 0       wait   tftpd   sysdb disk0:

default  v4       telnet      tcp   23      10        0       nowait telnetd sysdb
```

# SCP

Secure Copy Protocol (SCP) is a file transfer protocol which provides a secure and authenticated method for transferring files. SCP relies on SSHv2 to transfer files from a remote location to a local location or from local location to a remote location.

Cisco IOS XR software supports SCP server and client operations. If a device receives an SCP request, the SSH server process spawns the SCP server process which interacts with the client. For each incoming SCP subsystem request, a new SCP server instance is spawned. If a device sends a file transfer request to a destination device, it acts as the client.

When a device starts an SSH connection to a remote host for file transfer, the remote device can either respond to the request in Source Mode or Sink Mode. In Source Mode, the device is the file source. It reads the file from its local directory and transfers the file to the intended destination. In Sink Mode, the device is the destination for the file to be transferred.

Using SCP, you can copy a file from the local device to a destination device or from a destination device to the local device.

Using SCP, you can only transfer individual files. You cannot transfer a file from a destination device to another destination device.

## Transferring Files Using SCP

Secure Copy Protocol (SCP) allows you to transfer files between source and destination devices. You can transfer one file at a time. If the destination is a server, SSH server process must be running.

### Configuration Example

Transfers the file "test123.txt" from the local directory to the remote directory.

```
Router#scp /harddisk:/test123.txt xyz@1.75.55.1:/auto/remote/test123.txt
Connecting to 1.75.55.1...
Password:
Router#commit
```

### Verification

Verify if the file "test123.txt" is copied:

```
xyz-lnx-v1:/auto/remote> ls -altr test123.txt
-rw-r--r-- 1 xyz eng 0 Nov 23 09:46 test123.txt
```

### Associated Commands

- scp

# Cisco inetd

Cisco Internet services process daemon (Cinetd) is a multithreaded server process that is started by the system manager after the system has booted. Cinetd listens for Internet services such as Telnet service, TFTP service, and so on. Whether Cinetd listens for a specific service depends on the router configuration. For example, when the **tftp server** command is entered, Cinetd starts listening for the TFTP service. When a request arrives, Cinetd runs the server program associated with the service.

**Note** You must install Telnet RPM before using the Telnet service and the **show cinetd services** command.

```
RP/0/RP0/CPU0:ios#show cinetd services
Wed Aug 21 16:49:42.609 UTC

Vrf Name   Family  Service  Proto  Port  ACL   max_cnt   curr_cnt  wait   Program Client     Option

default    v4      telnet   tcp    23     10       0     nowait  telnetd   sysdb
```

# Telnet

Enabling Telnet allows inbound Telnet connections into a networking device.

### Prerequisites

Ensure to install Telnet RPM before using the Telnet service and **show cinetd services** command.

### Configuration Example

Enable telnet and limit the number of simultaneous users that can access the router to 10.

```
Router# configure
Router(config)# telnet ipv4 server max-servers 10
Router(config)# commit
```

### Verification

```
Router# show cinetd services
Vrf Name  Family    Service   Proto Port ACL max_cnt  curr_cnt  wait   Program Client Option

default   v4         tftp      udp  69       unlimited  0        wait   tftpd   sysdb
disk0:
default   v4     telnet  tcp  23        10  0        nowait telnetd sysdb
```

# Syslog source-interface

You can configure the logging source interface to identify the syslog traffic, originating in a VRF from a particular router, as coming from a single device.

### Configuration Example

Enable a source interface for the remote syslog server. Configure interface loopback 2 to be the logging source interface for the default vrf.

```
Router#configure
Router(config)#logging source-interface Loopback2

Router(config)#commit
```

### Running Configuration

```
Router#show running-config logging
/*Logging configuration after changing the source into loopback2 interface.
logging console debugging
logging monitor debugging
logging facility local4
logging 123.100.100.189 vrf default severity info port default
logging source-interface Loopback2
```

### Associated Commands

- logging source-interface
- show running-configuration logging

# HTTP Client Application

HTTP Client allows files to be transferred from http server to another device over a network using HTTP protocol. You can configure http client and various parameters associated with it by using the **http client** command.

### Configure HTTP Client

HTTP Client application is available by default. You can configure http client settings or view and modify the existing settings. To configure the settings, use the **http client** command in XR Config mode.

```
Router#configure
Router(config)#http client ?
connection          Configure HTTP Client connection
response            How long HTTP Client waits for a response from the server
                    for a request message before giving up
secure-verify-host  Verify that if server certificate is for the serverit is known as
secure-verify-peer  Verify authenticity of the peer's certificate
source-interface    Specify interface for source address
ssl                 SSL configuration to be used for HTTPS requests
tcp-window-scale    Set tcp window-scale factor for High Latency links
version             HTTP Version to be used in HTTP requests
vrf                 Name of vrf
```

*Table 5: Commands used to configure HTTP Client settings*

| Features | Description |
|---|---|
| **connection** | Configure HTTP Client connection by using either retry or timeout options. |
| **response** | How long HTTP Client waits for a response from the server for a request message before giving up. |
| **secure-verify-host** | Verify host in peer's certificate. To disable verifying this, you can use the command **http client secure-verify-host disable** |
| **secure-verify-peer** | Verify authenticity of the peer's certificate. |
| **source-interface** | Specifies the interface for source address for all outgoing HTTP connections. You can enter either an ipv4 or ipv6 address or both. |
| **ssl** version | SSL version (configuration) to be used for HTTPS requests. |
| **tcp-window-scale** scale | Set tcp window-scale factor for high latency links. |
| **version** version | HTTP version to be used in HTTP requests.<br>• 1.0 - HTTP1.0 will be used for all HTTP requests.<br>• 1.1 - HTTP1.1 will be used for all HTTP requests.<br>• default libcurl - will use HTTP version automatically. |
| **vrf** name | Name of vrf. |

### Configuration Examples

**Example 1:** This example shows how to set the tcp window-scale to 8.

```
Router(config)#http client tcp-window-scale 8
```

**Example 2:** This example shows how to set the HTTP version to 1.0.

```
Router(config)#http client version 1.0
```