Implementing IPv6 for Network Management

Last Updated: July 31, 2012

This document describes the concepts and commands used to manage Cisco applications over IPv6 and to implement IPv6 for network management.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Implementing IPv6 for Network Management

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Telnet Access over IPv6

The Telnet client and server in Cisco software support IPv6 connections. A user can establish a Telnet session directly to the device using an IPv6 Telnet client, or an IPv6 Telnet connection can be initiated from the device. A vty interface and password must be created in order to enable Telnet access to an IPv6 device.

TFTP IPv6 Support

TFTP is designed to transfer files over the network from one host to another using the most minimal set of functionality possible. TFTP uses a client/server model in which clients can request to copy files to or from a server. TFTP uses UDP over IPv4 or IPv6 as its transport, and it can work over IPv4 and IPv6 network layers.

- TFTP File Downloading for IPv6, page 2

TFTP File Downloading for IPv6

IPv6 supports TFTP file downloading and uploading using the **copy** command. The **copy** command accepts a destination IPv6 address or IPv6 hostname as an argument and saves the running configuration of the device to an IPv6 TFTP server, as follows:

```
Device# copy running-config tftp://[3ffe:xxxx:c18:1:290:27ff:fe3a:9e9a]/running-config
```

ping and traceroute Commands in IPv6

The **ping** command accepts a destination IPv6 address or IPv6 hostname as an argument and sends Internet Control Message Protocol version 6 (ICMPv6) echo request messages to the specified destination. The ICMPv6 echo reply messages are reported on the console. Extended ping functionality is also supported in IPv6.

The **traceroute** command accepts a destination IPv6 address or IPv6 hostname as an argument and will generate IPv6 traffic to report each IPv6 hop used to reach the destination address.

SSH over an IPv6 Transport

Secure shell (SSH) SSH in IPv6 functions the same and offers the same benefits as SSH in IPv4. The SSH server feature enables an SSH client to make a secure, encrypted connection to a Cisco device, and the SSH client feature enables a Cisco device to make a secure, encrypted connection to another Cisco device or to any other device running an SSH server. IPv6 enhancements to SSH consist of support for IPv6 addresses that enable a Cisco device to accept and establish secure, encrypted connections with remote IPv6 nodes over an IPv6 transport.

SNMP over an IPv6 Transport

Simple Network Management Protocol (SNMP) can be configured over IPv6 transport so that an IPv6 host can perform SNMP queries and receive SNMP notifications from a device running IPv6 software. The SNMP agent and related MIBs have been enhanced to support IPv6 addressing. This feature uses the data encryption standard (3DES) and advanced encryption standard (AES) message encryption.

- Cisco IPv6 MIBs, page 3

Cisco IPv6 MIBs

Cisco has long supported IP-MIB and IP-FORWARD-MIB in IPv4. CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB are IPv6 MIBs that are defined as being protocol-independent, but are implemented only for IPv6 objects and tables. IP-MIB and IP-FORWARD-MIB adhere to RFC 4293 and RFC 4292 standards, as follows:

- The upgrade is backward-compatible; all IP-MIB and IP-FORWARD-MIB objects and tables still appear.
- IP-MIB and IP-FORWARD-MIB include definitions of new IPv6-only, IPv4-only, and protocol-version independent (PVI) objects and tables.

CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB were removed from the Cisco releases in which CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB were applied. Information in CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB is included IP-MIB and IP-FORWARD-MIB.

MIBs Supported for IPv6

The following MIBs are supported for IPv6:

- CISCO-CONFIG-COPY-MIB
- CISCO-CONFIG-MAN-MIB
- CISCO-DATA-COLLECTION-MIB
- CISCO-FLASH-MIB
- CISCO-SNMP-TARGET-EXT-MIB
- ENTITY-MIB
- IP-FORWARD-MIB
- IP-MIB
- NOTIFICATION-LOG-MIB
- SNMP-TARGET-MIB

CISCO-CONFIG-COPY-MIB and CISCO-FLASH-MIB support IPv6 addressing when TFTP, remote copy protocol (rcp), or FTP is used.

Cisco IPv6 Embedded Management Components

Cisco embedded management components have IPv6-compliant operability in IPv6 and dual-stack IPv6 and IPv4 networks.

Syslog

The Cisco system message logging (syslog) process in IPv6 allows users to log syslog messages to external syslog servers and hosts with IPv6 addresses. This implementation allows user to specify an IPv4-based logging host (syslog server) by providing the host's IP address in IPv4 format (for example, 192.168.0.0) or IPv6 format (for example, 2001:DB8:A00:1::1/64).

CNS Agents

IPv6 addressing is supported in the Cisco Networking Services (CNS) subsystem. CNS is a foundation technology for linking users to networking services, and it provides the infrastructure for the automated configuration of large numbers of network devices. Many IPv6 networks are complex, with many devices, and each device must be configured individually. When standard configurations do not exist or have been modified, the time involved in initial installation and subsequent upgrading is considerable. ISPs need a method for sending out partial configurations to introduce new services.

To address all these issues, CNS was designed to provide "plug-and-play" network services using a central directory service and distributed agents. CNS features include CNS agents and a flow-through provisioning structure. CNS flow-through provisioning uses the CNS configuration and event agents to provide an automated workflow, eliminating the need for an onsite technician.

IPv6 addressing supports the CNS agents described in the following sections:

- CNS Configuration Agent, page 4
- CNS Event Agent, page 4
- CNS EXEC Agent, page 4
- CNS Image Agent, page 4

CNS Configuration Agent

The CNS configuration agent is involved in the initial configuration and subsequent partial configurations on a Cisco device. The configuration agent uses a CNS configuration engine to provide methods for automating initial Cisco device configurations, incremental configurations, and synchronized configuration updates, and the configuration engine reports the status of the configuration load as an event to which a network monitoring or workflow application can subscribe.

CNS Event Agent

The CNS event agent provides a transport connection to the CNS event bus for all other CNS agents. No event can be sent to the device by the configuration engine until the CNS event agent is operational and has successfully built a connection between the configuration engine and the device.

The event agent uses a CNS configuration engine to provide methods for automating initial Cisco device configurations, incremental configurations, and synchronized configuration updates.

CNS EXEC Agent

The CNS EXEC agent allows a remote application to execute a CLI command in EXEC mode on a Cisco device by sending an event message that contains the command.

CNS Image Agent

Administrators maintaining large networks of Cisco devices need an automated mechanism to load image files onto large numbers of remote devices. Network management applications are useful to determine

which images to run and how to manage images received from the Cisco online software center. Other image distribution solutions do not scale to cover thousands of devices and cannot distribute images to devices behind a firewall or using Network Address Translation (NAT). The CNS image agent enables the managed device to initiate a network connection and request an image download allowing devices using NAT, or behind firewalls, to access the image server.

The CNS image agent can be configured to use the CNS event bus. To use the CNS event bus, the CNS event agent must be enabled and connected to the CNS event gateway in the CNS Configuration Engine. The CNS image agent can also use an HTTP server that understands the CNS image agent protocol. Deployment of CNS image agent operations can use both the CNS event bus and an HTTP server.

Config Logger

Config logger tracks and reports configuration changes. Config logger supports two content types:

- Plain text--With plain-text format, the config logger reports configuration changes only.
- XML--The config logger uses XML to report the configuration change details (for example, what changed, who changed it, when changes were made, parser return code [PRC] values, and incremental NVGEN results).

HTTP(S) IPv6 Support

This feature allows the HTTP(S) client and server to support IPv6 addresses.

The HTTP server in Cisco software can service requests from both IPv6 and IPv4 HTTP clients. When the HTTP(S) server accepts a connection from a client, the server determines whether the client is an IPv4 or IPv6 host. The address family, IPv4 or IPv6, for the accept socket call is then chosen accordingly. The listening socket continues to listen for both IPv4 and IPv6 connections.

The HTTP client in Cisco software can send requests to both IPv4 and IPv6 HTTP servers.

When you use the IPv6 HTTP client, URLs with literal IPv6 addresses must be formatted using the rules listed in RFC 2732.

TCL

Tool command language (TCL) is used in Cisco software for IPv6 to support features such as embedded syslog manager (ESM), embedded event manager (EEM), interactive voice response (IVR), and tclsh parser mode. TCL supports both initiating (client) and listening (server) sockets.

NETCONF

The Network Configuration Protocol (NETCONF) defines a mechanism through which a network device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated. NETCONF uses XML-based data encoding for the configuration data and protocol messages.

SOAP Message Format

Using the Service-Oriented Access Protocol (SOAP) provides a way to format the layout of Cisco Networking Services (CNS) messages in a consistent manner. SOAP is intended for exchanging structured information in a decentralized, distributed environment. SOAP uses XML technologies to define an extensible messaging framework that provides a message format that can be exchanged over a variety of underlying protocols.

Within the SOAP message structure, there is a security header that enables CNS notification messages to authenticate user credentials.

IP SLAs for IPv6

Cisco IP Service Level Agreements (SLAs) are a portfolio of technology embedded in most devices that run Cisco software that allows Cisco customers to analyze IPv6 service levels for IPv6 applications and services, increase productivity, lower operational costs, and reduce the frequency of network outages. IP SLAs uses active traffic monitoring--the generation of traffic in a continuous, reliable, and predictable manner--for measuring network performance.

The following Cisco IP SLAs are supported for IPv6:

- Internet Control Message Protocol (ICMP) echo operation--Used to monitor end-to-end response time between a Cisco device and other devices using IPv4 or IPv6. ICMP echo is useful for troubleshooting network connectivity issues.
- TCP connect operation--Used to measure the response time taken to perform a TCP Connect operation between a Cisco device and other devices using IPv4 or IPv6.
- User Datagram Protocol (UDP) echo operation--Used to monitor end-to-end response time between a Cisco router and devices using IPv4 or IPv6 .
- UDP jitter operation--Used to analyze round-trip delay, one-way delay, one-way jitter, one-way packet loss, and connectivity in networks that carry UDP traffic in IPv4 or IPv6 networks.
- UDP jitter operation--Used to monitor VoIP quality levels in your network, allowing you to guarantee VoIP quality levels to your users in IPv4 or IPv6 networks.

How to Implement IPv6 for Network Management

Enabling Telnet Access to an IPv6 Device and Establishing a Telnet Session

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 host** *name* [*port*] *ipv6-address*
4. **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
5. **password** *password*
6. **login** [**local** | **tacacs**]
7. **ipv6 access-class** *ipv6-access-list-name* {**in** | **out**]
8. **telnet** *host* [*port*] [*keyword*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 host** *name* [*port*] *ipv6-address* **Example:** `Device(config)# ipv6 host cisco-sj 2001:DB8:20:1::12`	Defines a static hostname-to-address mapping in the hostname cache.
Step 4	**line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*] **Example:** `Device(config)# line vty 0 4`	Creates a vty interface.
Step 5	**password** *password* **Example:** `Device(config)# password hostword`	Creates a password that enables Telnet.
Step 6	**login** [**local** | **tacacs**] **Example:** `Device(config)# login tacacs`	(Optional) Enables password checking at login.
Step 7	**ipv6 access-class** *ipv6-access-list-name* {**in** | **out**] **Example:** `Device(config)# ipv6 access-list hostlist`	(Optional) Adds an IPv6 access list to the line interface. • Using this command restricts remote access to sessions that match the access list.

	Command or Action	Purpose
Step 8	**telnet** *host* [*port*] [*keyword*] **Example:** `Device(config)# telnet cisco-sj`	Establishes a Telnet session from a device to a remote host using either the hostname or the IPv6 address. • The Telnet session can be established to a device name or to an IPv6 address.

Enabling SSH on an IPv6 Device

This task is optional. If you do not configure SSH parameters, then the default values will be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*]
4. **exit**
5. **ssh** [**-v** { **1** | **2** } | **c** { **3des** | **aes128-cbc** | **aes192-cbc** | **aes256-cbc** } | **-l** *userid* | **-l** *userid:vrfname number ip-address ip-address* | **-l** *userid:rotary number ip-address* | **-m** { **hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96** } | **-o numberofpasswordprompts** *n* | **-p** *port-num*] { **ip-addr** | **hostname**} [**command** | **-vrf**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ssh** [**timeout** *seconds*	**authentication-retries** *integer*] **Example:** `Device(config)# IP ssh timeout 100 authentication-retries 2`	Configures SSH control variables on your device.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config)# exit`	Exits configuration mode, and returns the device to privileged EXEC mode.
Step 5	**ssh** [**-v** { **1** | **2** } | **c** { **3des** | **aes128-cbc** | **aes192-cbc** | **aes256-cbc** } | **-l** *userid* | **-l** *userid:vrfname number ip-address ip-address* | **-l** *userid:rotary number ip-address* | **-m** { **hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96** } | **-o numberofpasswordprompts** *n* | **-p** *port-num*] { **ip-addr** | **hostname**} [**command** | **-vrf**] **Example:** `Device# ssh -l userid1 2001:db8:2222:1044::72`	Starts an encrypted session with a remote networking device.

Configuring an SNMP Notification Server over IPv6

Use an SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to regulate access to the agent on the device. Optionally, you can specify one or more of the following characteristics associated with the string:

- An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent.
- A MIB view, which defines the subset of all MIB objects accessible to the given community.
- Read and write or read-only permission for the MIB objects accessible to the community.

You can configure one or more community strings. To remove a specific community string, use the **no snmp-server community** command.

The **snmp-server host** command specifies which hosts will receive SNMP notifications, and whether you want the notifications sent as traps or inform requests. The **snmp-server enable traps** command globally enables the production mechanism for the specified notification types (such as Border Gateway Protocol [BGP] traps, config traps, entity traps, and Hot Standby Router Protocol [HSRP] traps).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*]
4. **snmp-server engineID remote** {*ipv4-ip-address* | *ipv6-address*} [**udp-port** *udp-port-number*] [**vrf** *vrf-name*] *engineid-string*
5. **snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**context** *context-name*] [**read** *read-view*] [**write** *write-view*] [**notify** *notify-view*] [**access** [**ipv6** *named-access-list*] {*acl-number* | *acl-name*}]
6. **snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
7. **snmp-server user** *username group-name* [**remote** *host* [**udp-port** *port*]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access** [**ipv6** *nacl*] [**priv** {**des** | **3des** | **aes** {**128** | **192** | **256**}} *privpassword*] {*acl-number* | *acl-name*}]
8. **snmp-server enable traps** [*notification-type*] [**vrrp**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server community** *string* [**view** *view-name*] [**ro**	**rw**] [**ipv6** *nacl*] [*access-list-number*] **Example:** `Device(config)# snmp-server community mgr view restricted rw ipv6 mgr2`	Defines the community access string.
Step 4	**snmp-server engineID remote** {*ipv4-ip-address*	*ipv6-address*} [**udp-port** *udp-port-number*] [**vrf** *vrf-name*] *engineid-string* **Example:** `Device(config)# snmp-server engineID remote 3ffe:b00:c18:1::3/127 remotev6`	(Optional) Specifies the name of the remote SNMP engine (or copy of SNMP).

	Command or Action	Purpose
Step 5	**snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**context** *context-name*] [**read** *read-view*] [**write** *write-view*] [**notify** *notify-view*] [**access** [**ipv6** *named-access-list*] {*acl-number* | *acl-name*}] **Example:** `Device(config)# snmp-server group public v2c access ipv6 public2`	(Optional) Configures a new SNMP group, or a table that maps SNMP users to SNMP views.
Step 6	**snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Device(config)# snmp-server host host1.com 2c vrf trap-vrf`	Specifies the recipient of an SNMP notification operation. • Specifies whether you want the SNMP notifications sent as traps or informs, the version of SNMP to use, the security level of the notifications (for SNMPv3), and the recipient (host) of the notifications.
Step 7	**snmp-server user** *username group-name* [**remote** *host* [**udp-port** *port*]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access** [**ipv6** *nacl*] [**priv** {**des** | **3des** | **aes** {**128** | **192** | **256**}} *privpassword*] {*acl-number* | *acl-name*}] **Example:** `Device(config)# snmp-server user user1 bldg1 remote 3ffe:b00:c18:1::3/127 v2c access ipv6 public2`	(Optional) Configures a new user to an existing SNMP group. **Note** You cannot configure a remote user for an address without first configuring the engine ID for that remote host. This is a restriction imposed in the design of these commands; if you try to configure the user before the host, you will receive a warning message, and the command will not be executed.
Step 8	**snmp-server enable traps** [*notification-type*] [**vrrp**] **Example:** `Device(config)# snmp-server enable traps bgp`	Enables sending of traps or informs, and specifies the type of notifications to be sent. • If a value for the *notification-type* argument is not specified, all supported notification will be enabled on the device. • To discover which notifications are available on your device, enter the **snmp-server enable traps ?** command.

Configuring Cisco IOS IPv6 Embedded Management Components

Most IPv6 embedded management components are enabled automatically when IPv6 is enabled and do not need further configuration. To configure syslog over IPv6 or disable HTTP access to a router, refer to the tasks in the following sections:

- Configuring Syslog over IPv6, page 12
- Disabling HTTP Access to an IPv6 Device, page 12

Configuring Syslog over IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **logging host** {{*ip-address* | *hostname*} | {**ipv6** *ipv6-address* | *hostname*}} [**transport** {**udp** [**port** *port-number*] | **tcp** [**port** *port-number*] [**audit**]}] [**xml** | **filtered** [**stream** *stream-id*]] [**alarm** [*severity*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**logging host** {{*ip-address* | *hostname*} | {**ipv6** *ipv6-address* | *hostname*}} [**transport** {**udp** [**port** *port-number*] | **tcp** [**port** *port-number*] [**audit**]}] [**xml** | **filtered** [**stream** *stream-id*]] [**alarm** [*severity*]] **Example:** `Device(config)# logging host ipv6 AAAA:BBBB:CCCC:DDDD::FFFF`	Logs system messages and debug output to a remote host.

Disabling HTTP Access to an IPv6 Device

HTTP access over IPv6 is automatically enabled if an HTTP server is enabled and the device has an IPv6 address. If the HTTP server is not required, it should be disabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip http server**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip http server** **Example:** `Device(config)# no ip http server`	Disables HTTP access.

Configuration Examples for Implementing IPv6 for Network Management

Examples: Enabling Telnet Access to an IPv6 Device

The following examples provide information on how to enable Telnet and start a session to or from an IPv6 device. In the following example, the IPv6 address is specified as 2001:DB8:20:1::12, and the hostname is specified as cisco-sj. The **show host** command is used to verify this information.

```
Device# configure terminal
Device(config)# ipv6 host cisco-sj 2001:DB8:20:1::12
Device(config)# end
Device# show host
Default domain is not set
Name/address lookup uses static mappings
Codes:UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined
Host                 Port  Flags      Age Type   Address(es)
cisco-sj             None  (perm, OK)  0  IPv6 2001:DB8:20:1::12
```

To enable Telnet access to a device, create a vty interface and password:

```
Device(config)# line vty 0 4
```

```
password lab
login
```

To use Telnet to access the device, you must enter the password:

```
Device# telnet cisco-sj
Trying cisco-sj (2001:DB8:20:1::12)... Open
User Access Verification
Password:
cisco-sj
.
.
.
verification
```

It is not necessary to use the **telnet** command. Specifying either the hostname or the address is sufficient, as shown in the following examples:

```
Device# cisco-sj
```

or

```
Device# 2001:DB8:20:1::12
```

To display the IPv6 connected user (line 130) on the device to which you are connected, use the **show users** command:

```
Device# show users
    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00
 130 vty 0                 idle                 00:00:22   8800::3
```

Note that the address displayed is the IPv6 address of the source of the connection. If the hostname of the source is known (either through a domain name server [DNS] or locally in the host cache), then it is displayed instead:

```
Device# show users
    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00
 130 vty 0                 idle                 00:02:47   cisco-sj
```

If the user at the connecting device suspends the session with ^6x and then enters the **show sessions** command, the IPv6 connection is displayed:

```
Device# show sessions
Conn Host               Address             Byte  Idle Conn Name
*  1 cisco-sj 2001:DB8:20:1::12      0     0 cisco-sj
```

The Conn Name field shows the hostname of the destination only if it is known. If it is not known, the output might look similar to the following:

```
Device# show sessions
Conn Host               Address             Byte  Idle Conn Name
*  1 2001:DB8:20:1::12 2001:DB8:20:1::12      0     0 2001:DB8:20:1::12
```

Example: Disabling HTTP Access to the Device

In the following example, the **show running-config** command is used to show that HTTP access is disabled on the device:

```
Device# show running-config

Building configuration...
```

```
!
Current configuration : 1490 bytes
!
version 12.2
!
hostname Device
!
no ip http server
!
line con 0
line aux 0
line vty 0 4
```

Examples: Configuring an SNMP Notification Server over IPv6

The following example permits any SNMP to access all objects with read-only permission using the community string named public. The device also will send Border Gateway Protocol (BGP) traps to the IPv4 host 172.16.1.111 and IPv6 host 3ffe:b00:c18:1::3/127 using SNMPv1 and to the host 172.16.1.27 using SNMPv2c. The community string named public will be sent with the traps.

```
Device(config)# snmp-server community public
Device(config)# snmp-server enable traps bgp
Device(config)# snmp-server host 172.16.1.27 version 2c public
Device(config)# snmp-server host 172.16.1.111 version 1 public
Device(config)# snmp-server host 3ffe:b00:c18:1::3/127 public
```

Example: Associate an SNMP Server Group with Specified Views

In the following example, the SNMP context A is associated with the views in SNMPv2c group GROUP1 and the IPv6 named access list public2:

```
Device(config)# snmp-server context A
Device(config)# snmp mib community-map commA context A target-list commAVpn
Device(config)# snmp mib target list commAVpn vrf CustomerA
Device(config)# snmp-server view viewA ciscoPingMIB included
Device(config)# snmp-server view viewA ipForward included
Device(config)# snmp-server group GROUP1 v2c context A read viewA write viewA notify
access ipv6 public2
```

Example: Create an SNMP Notification Server

The following example configures the IPv6 host as the notification server:

```
Device> enable
Device# configure terminal
Device(config)# snmp-server community mgr view restricted rw ipv6 mgr2
Device(config)# snmp-server engineID remote 3ffe:b00:c18:1::3/127 remotev6
Device(config)# snmp-server group public v2c access ipv6 public2
Device(config)# snmp-server host host1.com 2c vrf trap-vrf
Device(config)# snmp-server user user1 bldg1 remote 3ffe:b00:c18:1::3/127 v2c access ipv6
public2
Device(config)# snmp-server enable traps bgp
Device(config)# exit
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 supported features	"Start Here: Cisco IOS Software Release Specifics for IPv6 Features ," *Cisco IOS IPv6 Configuration Guide*
Basic IPv6 configuration tasks	"Implementing IPv6 Addressing and Basic Connectivity ," *Cisco IOS IPv6 Configuration Guide*
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS IPv6 Command Reference*
SSH configuration information	*Cisco IOS Security Command Reference*
IPv4 CNS, SOAP	"Cisco Networking Services," *Cisco IOS Network Management Configuration Guide*
NETCONF	"Network Configuration Protocol," *Cisco IOS Network Management Configuration Guide*
IP SLAs for IPv6	• IP SLAs--Analyzing IP Service Levels Using the ICMP Echo Operation • IP SLAs--Analyzing IP Service Levels Using the TCP Connect Operation • IP SLAs--Analyzing IP Service Levels Using the UDP Echo Operation • IP SLAs--Analyzing IP Service Levels Using the UDP Jitter Operation • IP SLAs--Analyzing VoIP Service Levels Using the UDP Jitter Operation

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
• CISCO-CONFIG-COPY-MIB • CISCO-CONFIG-MAN-MIB • CISCO-DATA-COLLECTION-MIB • CISCO-FLASH-MIB • CISCO-RTTMON-IPv6-MIB • CISCO-SNMP-TARGET-EXT-MIB • ENTITY-MIB • IP-FORWARD-MIB • IP-MIB • NOTIFICATION-LOG-MIB • SNMP-TARGET-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2732	*Format for Literal IPv6 Addresses in URLs*
RFC 3414	*User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*
RFC 3484	*Default Address Selection for Internet Protocol version 6 (IPv6)*
RFC 4292	IP Forwarding Table MIB
RFC 4293	*Management Information Base for the Internet Protocol (IP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Implementing IPv6 for Network Management

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for Managing Cisco IOS Applications over IPv6

Feature Name	Releases	Feature Information
CNS Agents for IPv6	12.2(33)SB 12.2(33)SRC 12.2(50)SY 12.4(20)T	CNS configuration and event agents use a CNS configuration engine to provide methods for automating initial Cisco IOS device configurations, incremental configurations, and synchronized configuration updates, and the configuration engine reports the status of the configuration load as an event to which a network monitoring or workflow application can subscribe.
HTTP(S) IPv6 Support	12.2(33)SB 12.2(33)SRC 12.2(50)SY 12.4(20)T	This feature enhances the HTTP(S) client and server to support IPv6 addresses.
IP SLAs for IPv6	12.2(33)SB 12.2(33)SRC 12.2(50)SY 12.4(20)T 15.0(1)S	IP SLAs are supported for IPv6.
IPv6 for Config Logger	12.2(33)SB 12.2(33)SRC 12.2(50)SY 12.4(20)T	Config logger tracks and reports configuration changes.
IPv6 NETCONF Support	12.2(33)SB 12.2(33)SRC 12.2(50)SY 12.4(20)T	The Network Configuration Protocol (NETCONF) defines a simple mechanism through which a network device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated.

Feature Name	Releases	Feature Information
IPv6--syslog over IPv6	12.2(33)SB 12.2(33)SRC 12.2(33)SXI 12.4(4)T	The Cisco IOS syslog process in IPv6 allows users to log syslog messages to external syslog servers and hosts with IPv6 addresses.
IPv6 Services--IP-FORWARD-MIB Support	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T	A MIB is a database of objects that can be managed on a device. The managed objects, or variables, can be set or read to provide information on the network devices and interfaces.
IPv6 Services--IP-MIB Support	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T	A MIB is a database of objects that can be managed on a device. The managed objects, or variables, can be set or read to provide information on the network devices and interfaces.
IPv6 Services--RFC 4293 IP-MIB (IPv6 only) and RFC 4292 IP-FORWARD-MIB (IPv6 only)	12.2(33)SB 12.2(58)SE 12.2(54)SG 12.2(33)SRC 12.2(50)SY 15.1(3)T	IP-FORWARD-MIB and IP-MIB were updated to RFC 4292 and RFC 4293 standards, respectively.
IPv6 Support for TCL	12.2(33)SRC 12.2(50)SY 12.4(20)T	IPv6 supports TCL.
IPv6 Support in SOAP	12.2(33)SB 12.2(33)SRC 12.2(50)SY 12.4(20)T	SOAP is a protocol intended for exchanging structured information in a decentralized, distributed environment.
SNMP over IPv6	12.0(27)S 12.2(33)SRB 12.2(33)SXI 12.3(14)T 12.4 12.4(2)T	SNMP can be configured over IPv6 transport so that an IPv6 host can perform SNMP queries and receive SNMP notifications from a device running Cisco IOS IPv6.
SNMPv3 - 3DES and AES Encryption Support	12.2(33)SB 12.2(33)SRB 12.2(33)SXI 12.4(2)T	SNMP for IPv6 supports 3DES and AES encryption.

Feature Name	Releases	Feature Information
SSH over an IPv6 Transport	12.0(22)S 12.2(8)T 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.3 12.3(2)T 12.4 12.4(2)T 15.0(1)S	SSH in IPv6 functions the same and offers the same benefits as SSH in IPv4--the SSH Server feature enables an SSH client to make a secure, encrypted connection to a Cisco router and the SSH Client feature enables a Cisco router to make a secure, encrypted connection to another Cisco router or to any other device running an SSH server.
Telnet Access over IPv6	12.0(22)S 12.2(2)T 12.2(14)S 12.2(28)SB 12.3 12.3(2)T 12.4 12.4(2)T	The Telnet client and server in the Cisco IOS software support IPv6 connections. A user can establish a Telnet session directly to the router using an IPv6 Telnet client, or an IPv6 Telnet connection can be initiated from the router.
TFTP File Downloading for IPv6	12.0(22)S 12.2(2)T 12.2(14)S 12.2(28)SB 12.3 12.3(2)T 12.4 12.4(2)T	IPv6 supports TFTP file downloading and uploading.