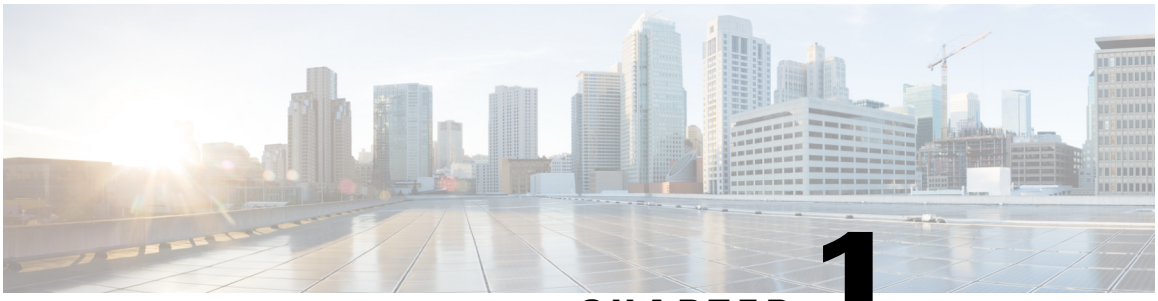
IPv6 Network Management Configuration Guide, Cisco IOS Release 15M&T

First Published: November 21, 2012

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Telnet Access over IPv6

The Telnet client and server in the Cisco software support IPv6 connections.

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.

Prerequisites for Telnet Access over IPv6

To enable Telnet access over IPv6 to a device, you must create a vty interface and password.

Information About Telnet Access over IPv6

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.

How to Enable Telnet Access over IPv6

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.

	Command or Action	Purpose
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.
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.

Configuration Examples for Telnet Access over IPv6

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
```

Additional References for IPv6 Source Guard and Prefix Guard

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv4 addressing	*IP Addressing: IPv4 Addressing Configuration Guide*
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

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

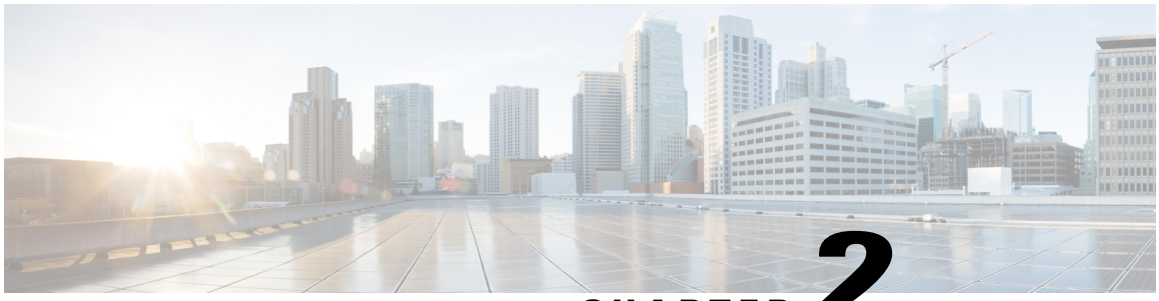
Feature Information for Telnet Access over IPv6

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 Telnet Access over IPv6

Feature Name	Releases	Feature Information
Telnet Access over IPv6	12.2(2)T 12.2(18)SXE 12.2(25)SEA 12.2(25)SG 12.2(33)SRA 15.0(2)SG Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.2SG	Telnet access over IPv6 is supported. The following commands were introduced or modified: **ipv6 access-class**, **ipv6 host**.



CHAPTER 2

IPv6 Support for TFTP

TFTP uses UDP over IPv4 or IPv6 as its transport and can work over IPv4 and IPv6 network layers.

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 IPv6 Support for TFTP

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

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
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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

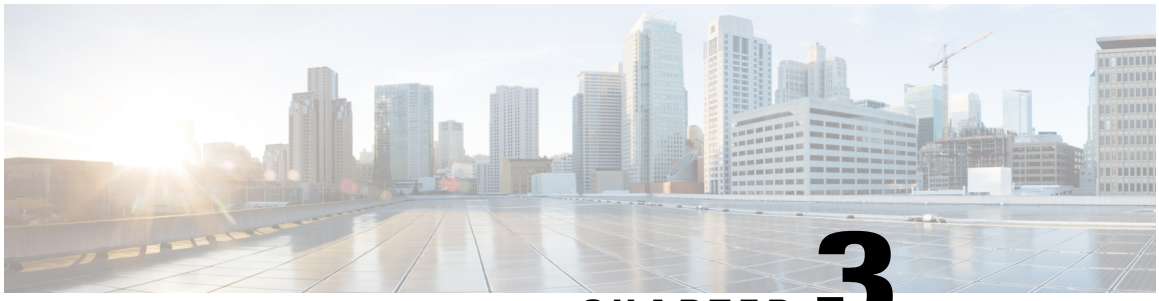
Feature Information for IPv6 Support for TFTP

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 2: Feature Information for IPv6 Support for TFTP

Feature Name	Releases	Feature Information
TFTP IPv6 Support	12.0(22)S 12.2(2)T 12.2(14)S 12.2(28)SB 15.1(1)SY 15.2(1)E	IPv6 support for TFTP is supported. No commands were introduced or modified.



CHAPTER 3

In-band OAM for IPv6

The In-band OAM for IPv6 feature supports "in-band" operation, administration, and maintenance (OAM) for IPv6 which enables you to define advanced options for recording OAM information in an IPv6 packet during its traversal through a particular network domain.

Benefits of In-band OAM for IPv6

Benefits over Active OAM (out-of-band)

In-band OAM for IPv6 enables you to use "In-band" mechanisms which do not require extra packets to be sent and hence don't change the packet traffic mix within the network. Traceroute and ping for example use ICMP messages: New packets are injected to get tracing information. Those add to the number of messages in a network, which already might be highly loaded or suffering performance issues for a particular path or traffic type.

Packet scheduling algorithms, especially for balancing traffic across equal cost paths or links, often leverage information contained within the packet, such as protocol number, IP-address or MAC-address. Probe packets would thus either need to be sent from the exact same endpoints with the exact same parameters, or probe packets would need to be artificially constructed as "fake" packets and inserted along the path. Both approaches are often not feasible from an operational perspective, be it that access to the end-system is not feasible, or that the diversity of parameters and associated probe packets to be created is too large. An in-band mechanism is an alternative in those cases.

In-band mechanisms also don't suffer from implementations, where probe traffic is handled differently (and potentially forwarded differently) by a router than regular data traffic. Traditional ping and traceroute tools

return the OAM results to the sender of the probe. It would be advantageous to separate the sending of an OAM probe from the receiving of the telemetry data. In this context, it is desired to not assume there is a bidirectional working path.

Overlay and Underlay Statistics

Different network deployments use tunneling mechanisms to create overlay or service-layer networks such as VXLAN-GPE, GRE, or LISP. Overlay networks do not offer the user of the overlay any insight into the underlay network. The path that a particular tunneled packet takes, or other operational details such as the per-hop delay/jitter in the underlay are not visible to the user of the overlay network, giving rise to diagnosis and debugging challenges in case of connectivity or performance issues. OAM tools like ping or traceroute can be used either at the overlay or at the underlay which means that the user of the overlay has typically no access to OAM in the underlay, unless specific operational procedures are put in place. With In-band OAM the operator of the underlay can offer details of the connectivity in the underlay to the user of the overlay. The operator of the egress tunnel router can choose to share the recorded information about the path with the user of the overlay.

Using coupled with mechanisms such as Segment Routing, overlay network and underlay network can be more tightly coupled. In these scenarios, the user of the overlay has detailed diagnostic information available in case of failure conditions. The user of the overlay can also use the path recording information as input to traffic steering or traffic engineering mechanisms, for example achieve path symmetry for the traffic between two endpoints.

SLA Verification

In-band OAM can help overlay-service users to verify if the negotiated SLAs for the real traffic are met by the underlay network provider. In-band OAM is different from solutions which rely on active probes to test an SLA and helps you to avoid wrong interpretations and cheating, which can happen if the probe traffic that is used to perform SLA-check is prioritized by the network provider of the underlay.

Analytics and Diagnostics

If you are a network planner or an operator, you can benefit from knowledge of the actual traffic distribution in your network. When deriving an overall network connectivity traffic matrix you typically need to correlate data gathered from each individual devices in the network. If the path of a packet is recorded while the packet is forwarded, the entire path that a packet took through the network is available to the egress system. This demands retrieving individual traffic statistics from every device in the network and correlate those statistics. You must also employ other mechanisms such as leveraging traffic engineering with null-bandwidth tunnels just to retrieve the appropriate statistics to generate the traffic matrix.

Using the individual path tracing options supported by In-band OAM for IPv6 feature, information is available at packet level granularity, rather than only at aggregate level, which is usually the case with IPFIX-style methods which employ flow-filters at the network elements. Data-center networks which use equal-cost multipath (ECMP) forwarding is an example where detailed statistics on flow distribution in the network are of high demand. If a network supports ECMP, one can create detailed statistics for the different paths packets take through the network at the egress system, without a need to correlate/aggregate statistics from every router in the system. When you use In-band OAM, transit devices are also off-loaded from the task of gathering packet statistics.

Proof of Transit

In-band OAM for IPv6 feature enables you to validate proof of transit. Several deployments use traffic engineering, policy routing, segment routing or Service Function Chaining (SFC) to steer packets through a specific set of nodes. In certain cases regulatory obligations or a compliance policy require to prove that all

packets that are supposed to follow a specific path are indeed being forwarded across the exact set of nodes specified. If a packet flow is supposed to go through a series of service functions or network nodes, it has to be proven that all packets of the flow actually went through the service chain or collection of nodes specified by the policy. In case the packets of a flow weren't appropriately processed, a verification device would be required to identify the policy violation and take corresponding actions (e.g., drop or redirect the packet, send an alert etc.) corresponding to the policy. In today's deployments, the proof that a packet traversed a particular service chain is typically delivered in an indirect way.

Information About In-band OAM for IPv6

There are different mechanisms which add tracing information to the regular data traffic, also referred to as "in-band" or "passive OAM". In-band mechanisms can complement active, probe-based mechanisms such as ping or traceroute, which are considered as "out-of-band", because the messages are transported independently from regular data traffic. The In-band OAM for IPv6 feature enables you to add tracing information to your regular data traffic which can be used in getting different telemetry data about your network with minimal impact on network performance.

In-band OAM for IPv6 feature supports the following telemetry options:

Tracing

The In-band OAM for IPv6 feature supports multiple tracing options which can be enabled to provide different statistic of your network. You can use the tracing options to indicate the following:

- Node ID in the node data.
- Ingress interface ID in the node data.
- Egress interface ID in the node data.
- Timestamp in the node data.
- Application data in the node data.

Proof of Transit

You can use In-band OAM data which is added to every packet for achieving or verifying Proof Of Transit. The OAM data is updated at every hop or every required node and is used to verify whether a packet traversed all required nodes. When the verifier receives each packet, it can validate whether the packet traversed the service chain correctly. The OAM data is used to verify whether a packet traversed the nodes it is supposed to traverse.

Sequence Numbering

The In-band OAM for IPv6 feature supports traffic sequence numbering and analysis. You can select the traffic which should be analyzed using sequence numbering. A sequence number is inserted into the packet at the ingress edge of the domain and based on the sequence of numbers received at the egress edge of the domain, traffic loss/reorder/duplicate analysis is done. The ACL entries need to be specific to the flow if the analysis needs to be done on a per flow basis. The sequence number is inserted at the ingress edge router's option in the IPv6 HbyH extension header that matches each of the ACL line entry using the 'insert' option. The analysis is enabled using the 'analyze' option which retrieves the sequence numbers from the packets and analyses the traffic flow for lost/reordered/duplicate packets.

In-band OAM in a Network

The In-band OAM for IPv6 feature can be applied for the getting the following statistic and diagnostic information about an IPv6 network in different scenarios:

- Traffic Matrix: Derive the network traffic matrix which is the traffic for a given time interval between any two edge nodes of a given domain. Can be performed for all traffic or per QoS-class.
- Flow Debugging: Discover which path(s) a particular set of traffic (identified by an n-tuple) takes in the network. Especially useful in case where the traffic is balanced across multiple paths, for example, link aggregation (LACP) and equal cost multi-pathing (ECMP).
- Loss statistics per path: Retrieve loss statistics per flow and path in the network.
- Path Heat Maps: Discover highly utilized links in the network.
- Trend analysis on traffic patterns: Analyze the forwarding path for a specific set of traffic changes over time.
- Network delay distribution: Show delay distribution across network by node or links. If enabled per application or a specific flow then the path taken with delay at each node is displayed.
- Low-Power networks: Supports application level OAM information (e.g. battery charge level) into data traffic to avoid sending extra OAM traffic which results in an extra cost on the devices. Using the battery charge level as example, avoiding sending extra OAM packets just to communicate battery health can save battery on sensors.
- Path Verification or Service Function Path Verification: Proof and verification of packets traversing check points in the network, where check points can be nodes in the network or service functions.
- Geo-location Policy: Network policy implemented based on which path packets took. Example: Only if packets originated and stayed within the trading-floor department, access to specific applications or servers is granted.

Netflow Collector

The In-band OAM for IPv6 feature provides support for Flexible Net Flow (FNF). You can use In-band OAM for IPv6 feature to enable the FNF exporter on the router and can export In-band OAM data records to netflow collector.

After de-capsulating the incoming IPv6 In-band OAM packets, the egress edge device extracts required fields from packet and exports to netflow collector. A device can export two types of packets to a netflow collector:

- Netflow data
- Netflow template

A netflow template is a set of metadata that describes the data exported. This metadata helps the collector to derive the semantics of all fields received from exporter in data packet.

In-band OAM sends different sets of semantically grouped data and its relevant templates to netflow collector. Frequency of template export can be configured on the device. It is mandatory for the collector to receive the template along with data packet to understand the data field in data packets.

Netflow Collector Templates

This section provides information about different Netflow Collector Templates that can be used with In-band OAM for IPv6 feature:

Template: Node-Id - Node Name Mapping

The Node-Id - Node Name Mappings template is used to map node id to node name for visualization application integrated with the netflow collector.

- Exporter Format: NetFlow Version 9
- Template ID: 256
- Source ID: 0
- Record Size: 36

Following is the layout for Node-Id - Node Name Mapping template:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Field                         | Type  | Offset | Size|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| iOAM my node-id               | 38001 | 0      | 4   |
| iOAM my node name             | 38002 | 4      | 32  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Field: data to be exported
- Type: ID of field used by collector to understand the meaning of field.
- Offset: Position in netflow data packet
- Size: Size of the field

Template: Sequence Numbering statistics

The Sequence Numbering statistics template is used to export statistics such as duplicate packets, reordered packets, lost packets as observed by an egress edge node in the In-band OAM domain.

- Exporter Format: NetFlow Version 9
- Template ID : 259
- Source ID : 0
- Record Size : 72

Following is the layout for Sequence Numbering statistics template:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Field |                                  Type     | Offset | Size |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++
| ipv6 source address                    | 27     | 0      | 16   |
| ipv6 destination address               | 28     | 16     | 16   |
| ipv6 source mask                       | 170    | 32     | 1    |
| ipv6 destination mask                  | 169    | 33     | 1    |
| iOAM my node-id                        | 38001  | 34     | 4    |
| IOAM packet counter                    | 38005  | 38     | 4    |
| IOAM lost packet counter               | 38023  | 42     | 4    |
| IOAM duplicate packet counter          | 38024  | 46     | 4    |
| IOAM reordered packet counter          | 38025  | 50     | 4    |
| IOAM highest PPC sequence number       | 38026  | 54     | 4    |
| IPv6 Protocol Field                    | 38028  | 58     | 2    |
| start timestamp                        | 38003  | 62     | 4    |
| end timestamp                          | 38004  | 66     | 4    |
| transport source-port                  | 7      | 70     | 2    |
| transport destination-port             | 11     | 72     | 2    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Template: Aggregated Flow Statistics/Path-record

The Aggregated Flow Statistics/Path-record (tracing) template is used by In-band OAM to export path details taken by packets between source and destination. Path Map keeps the identification of node-id, ingress and egress interface.

- Exporter Format: NetFlow Version 9
- Template ID : 260
- Source ID : 0
- Record Size : 224

Following is the layout for Aggregated Flow Statistics/Path-record (Tracing) template:

Field	Type	Offset	Size
ipv6 source address	27	0	16
ipv6 destination address	28	16	16
iOAM my node-id	38001	32	4
iOAM node1 id	38032	36	4
iOAM node2 id	38035	40	4
iOAM node3 id	38038	44	4
iOAM node4 id	38041	48	4
iOAM packet counter	38005	52	4
iOAM byte count	38006	56	4
start timestamp	38003	60	4
end timestamp	38004	64	4
transport source-port	7	68	2
transport destination-port	11	70	2
ipv6 protocol filed	38028	72	2
iOAM number of nodes	38031	74	2
iOAM Path Map	38030	92	132
iOAM node1 in if id	38033	76	2
iOAM node1 eif id	38034	78	2
iOAM node2 in if id	38036	80	2
iOAM node2 eif id	38037	82	2
iOAM node3 in if id	38039	84	2
iOAM node3 eif id	38040	86	2
iOAM node4 in if id	38042	88	2
iOAM node4 eif id	38043	90	2

Template: Service Chain Details

The Service Chain Details template provides proof of transit information.

- Exporter Format: NetFlow Version 9
- Template ID : 263
- Source ID : 0
- Record Size : 24

Following is the layout for Service Chain Details template:

Field	Type	Offset	Size
iOAM my node-id	38001	0	4
start timestamp	38003	4	4
end timestamp	38004	8	4
service chain id	38045	12	4
success packet counter	38046	16	4
failure packet counter	38047	20	4

Restrictions for In-band OAM for IPv6

- The In-band OAM for IPv6 profile configurations (ipv6 ioam profile) and service profile configurations (ipv6 ioam service-profile) are applied on per-interface basis. Multiple In-band OAM for IPv6 profiles or service profiles on the same interface is not supported.
- The In-band OAM for IPv6 profiles and service profiles are linked with the ACL on a one-to-one basis. There is a one-to-one match between In-band OAM for IPv6 profile and its related service profile which is necessary for the In-band OAM for IPv6 configuration to work.
- All generic ACL options are not supported, the supported ones are Source/Destination IP match and TCP/UDP Port match.
- ECMP algorithm for the same destination, the path taken is different at any given time.
- Flow export option with line rate traffic produces memory limitation.
- The In-band OAM for IPv6 feature supports only IOS Proof of Transit with 32bit math calculation.
- Reload with In-band OAM requires a reconfiguration of interface specific configurations.

Configuring In-band OAM for IPv6

Configuring In-band OAM for IPv6 Tracing (Single Path Verification)

Configuring Encap Node

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length*
5. **exit**
6. **ipv6 ioam profile** *profile-name*
7. **path-record match-flow** *acl-name* **encap/decap**
8. **trace-type** {**N_IF_TS_APP** | **N_IFS** | **N_TS** | **N_APP** | **N_TS_APP**}
9. **trace-elts** *number*
10. **trace-tsp** {**0-sec** | **1-milisec** | **2-microsec** | **3-nanosec**}
11. **exit**
12. **ipv6 ioam node-id** *node-id*
13. **ipv6 ioam path-record**
14. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
15. **ipv6 ioam profile** *profile-name*
16. **exit**

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 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list acl_h1_h3`	Defines an IPv6 access list.

	Command or Action	Purpose
Step 4	**permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length* **Example:** `Device(config-ipv6-acl)# permit host ::A:1:1:0:6 host ::A:1:1:0:17`	Defines the source and destination IPv6 network to set permit conditions.
Step 5	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 acl configuration mode and returns to global configuration mode.
Step 6	**ipv6 ioam profile** *profile-name* **Example:** `Device(config)# ipv6 ioam profile sc1`	Defines IPv6 In-band OAM profile name.
Step 7	**path-record match-flow** *acl-name* **encap/decap** **Example:** `Device(config-ioam-profile)# path-record match-flow acl_h1_h3 encap`	Enables path record option on encap node.
Step 8	**trace-type** {**N_IF_TS_APP** | **N_IFS** | **N_TS** | **N_APP** | **N_TS_APP**} **Example:** `Device(config-ioam-profile)# trace-type N_IF_TS_APP`	Defines the trace type meta data option for the encap node. • N_IF_TS_APP - Add node-id, interfaces, timestamp and ,app data information. N_IFS - Add node-id and interfaces information. N_TS - Add node-id and timestamp information. N_APP - Add node-id and app data information. N_TS_APP - Add node-id, timestamp and app data information. **Note** This option is only available on encap node.
Step 9	**trace-elts** *number* **Example:** `Device(config-ioam-profile)# trace-elts 3`	Defines the total number of nodes in the In-band OAM for IPv6 domain. The values must be an integer. **Note** This option is only available on encap node.
Step 10	**trace-tsp** {**0-sec** | **1-milisec** | **2-microsec** | **3-nanosec**} **Example:** `Device(config-ioam-profile)# trace-tsp 2`	Trace tsp defines the value in which the time stamp is recorded in the packet. **Note** This option is only available on encap node.
Step 11	**exit** **Example:** `Device(config-ioam-profile)# exit`	Exits In-band OAM profile configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 12	**ipv6 ioam node-id** *node-id* **Example:** `Device(config)# ipv6 ioam node-id 1`	Defines the node-id which will be sent in the In-band OAM data in the new E2E header. **Note** In-band OAM node-id identifies the network node in the given network domain. You must ensure that all the nodes in the domain have unique node-id configured using this command.
Step 13	**ipv6 ioam path-record** **Example:** `Device(config)# ipv6 ioam path-record`	Enables the In-band OAM process in a network node to start adding the node-data in the IPv6 traffic passing through the network node. **Note** Path-record must be configured on all nodes in domain.
Step 14	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# int g0/2`	Configures an interface and enters interface configuration mode.
Step 15	**ipv6 ioam profile** *profile-name* **Example:** `Device(config)# ipv6 ioam profile sc1`	Defines IPv6 In-band OAM profile name.
Step 16	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Intermediate Node

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 ioam node-id** *node-id*
4. **ipv6 ioam path-record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 ioam node-id** *node-id* **Example:** `Device(config)# ipv6 ioam node-id 2`	Defines the node-id which will be sent in the In-band OAM data in the new E2E header. **Note** In-band OAM node-id identifies the network node in the given network domain. You must ensure that all the nodes in the domain have unique node-id configured using this command.
Step 4	**ipv6 ioam path-record** **Example:** `Device(config)# ipv6 ioam path-record`	Enables the In-band OAM process in a network node to start adding the node-data in the IPv6 traffic passing through the network node. **Note** Path-record must be configured on all nodes in domain.

Configuring Decap Node

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length*
5. **exit**
6. **ipv6 ioam profile** *profile-name*
7. **path-record match-flow** *acl-name***encap/decap**
8. **exit**
9. **ipv6 ioam node-id** *node-id*
10. **ipv6 ioam path-record**
11. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
12. **ipv6 ioam profile** *profile-name*
13. **exit**
14. **show ipv6 ioam trace-profile** *profile-name*

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 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list acl_h1_h3`	Defines an IPv6 access list.
Step 4	**permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length* **Example:** `Device(config-ipv6-acl)# permit host ::A:1:1:0:6 host ::A:1:1:0:17`	Defines the source and destination IPv6 network to set permit conditions.
Step 5	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 acl configuration mode and returns to global configuration mode.
Step 6	**ipv6 ioam profile** *profile-name* **Example:** `Device(config)# ipv6 ioam profile sc1`	Defines IPv6 In-band OAM profile name.
Step 7	**path-record match-flow** *acl-name***encap/decap** **Example:** `Device(config-ioam-profile)# path-record match-flow acl_h1_h3 decap`	Enables path record option on decap node.
Step 8	**exit** **Example:** `Device(config-ioam-profile)# exit`	Exits In-band OAM profile configuration mode and returns to global configuration mode.
Step 9	**ipv6 ioam node-id** *node-id* **Example:** `Device(config)# ipv6 ioam node-id 3`	Defines the node-id which will be sent in the In-band OAM data in the new E2E header. **Note** In-band OAM node-id identifies the network node in the given network domain. You must ensure that all the nodes in the domain have unique node-id configured using this command.

	Command or Action	Purpose
Step 10	**ipv6 ioam path-record** **Example:** `Device(config)# ipv6 ioam path-record`	Enables the In-band OAM process in a network node to start adding the node-data in the IPv6 traffic passing through the network node. **Note** Path-record must be configured on all nodes in domain.
Step 11	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# int g0/3`	Configures an interface and enters interface configuration mode.
Step 12	**ipv6 ioam profile** *profile-name* **Example:** `Device(config)# ipv6 ioam profile sc1`	Defines IPv6 In-band OAM profile name.
Step 13	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Step 14	**show ipv6 ioam trace-profile** *profile-name*
	
	Example:
	`Device(config)# show ipv6 ioam trace-profile sc1`
	`Trace Record Information:`
	
	`Node 1:`
	`node id : 1, ingress if : 3, egress if : 4`
	`Node 2:`
	`node id : 2, ingress if : 3, egress if : 4`
	`Node 3:`
	`node id : 3, ingress if : 3, egress if : 5`
	Note This info can be viewed only in decap node.	

Configuring In-band OAM for IPv6 Proof of Transit in a Small Network (Single Service Chain)

Configuring Encap Node

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length*
5. **exit**
6. **ipv6 ioam profile** *profile-name*
7. **service-chaining match-flow** *access-list-name* {**insert** | **analyze**}
8. **exit**
9. **ipv6 ioam service-chaining** *service-profile-name*
10. **service-chain** {**insert** | **analyze**}
11. **prime-number** *number*
12. **secret-key** *hexadecimal-value*
13. **lpc** *hexadecimal-value*
14. **polynomial2** *hexadecimal-value*
15. **bits-in-random** *integer-value*
16. **exit**
17. **ipv6 ioam node-id** *node-id*
18. **ipv6 ioam path-record**
19. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
20. **ipv6 ioam profile** *profile-name*
21. **ipv6 ioam service-profile** *service-profile-name*
22. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list acl_h1_h3`	Defines an IPv6 access list.
Step 4	**permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length* **Example:** `Device(config-ipv6-acl)# permit host ::A:1:1:0:6 host ::A:1:1:0:17`	Defines the source and destination IPv6 network to set permit conditions.
Step 5	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 acl configuration mode and returns to global configuration mode.
Step 6	**ipv6 ioam profile** *profile-name* **Example:** `Device(config-ioam-profile)# ipv6 ioam profile sc1`	Defines IPv6 In-band OAM profile name.
Step 7	**service-chaining match-flow** *access-list-name* {**insert** | **analyze**} **Example:** `Device(config-ioam-profile)# service-chaining match-flow acl_h1_h3 insert`	selects the acl and traffic for service chain validation. **Note** The insert option is enabled at the ingress edge router which inserts service chain relevant details in E2E header that matches each of the acl line entry.
Step 8	**exit** **Example:** `Device(config-ioam-profile)# exit`	Exits In-band OAM profile configuration mode and returns to global configuration mode.
Step 9	**ipv6 ioam service-chaining** *service-profile-name* **Example:** `Device(config)# ipv6 ioam service-chaining sf1`	Speicifies the IPv6 In-band OAM service chaining profile name.
Step 10	**service-chain** {**insert** | **analyze**} **Example:** `Device(config-ioam-service-chaining)# service-chain insert`	Specifies whether the given node is encap or decap node. **Note** The service-chain insert command will accept either of insert or analyze based on the node.

	Command or Action	Purpose
Step 11	**prime-number** *number* **Example:** `Device(config-ioam-service-chaining)# prime-number 61819`	Specifies the prime number required to calculate the share key. The range is 1-2147483647. **Note** Prime number must be unique across In-band OAM domain for a particular chain.
Step 12	**secret-key** *hexadecimal-value* **Example:** `Device(config-ioam-service-chaining)# secret-key 2878`	Specifies the secret key that is shared among service chain nodes. The range is 0-FFFFFFFF.
Step 13	**lpc** *hexadecimal-value* **Example:** `Device(config-ioam-service-chaining)# lpc 6ddfd460`	Specifies the value for lag range basis polynomial factor for constant. The range is 0-FFFFFFFF.
Step 14	**polynomial2** *hexadecimal-value* **Example:** `Device(config-ioam-service-chaining)# polynomial2 43259140`	Specifies the value for polynomial pre computed split share. The range is 0-FFFFFFFF.
Step 15	**bits-in-random** *integer-value* **Example:** `Device(config-ioam-service-chaining)# bits-in-random 31`	Specifies random number size. The range is 1-31.
Step 16	**exit** **Example:** `Device(config-ioam-service-chaining)# exit`	Exits In-band OAM service chaining configuration mode and returns to global configuration mode.
Step 17	**ipv6 ioam node-id** *node-id* **Example:** `Device(config)# ipv6 ioam node-id 1`	Defines the node-id which will be sent in the In-band OAM data in the new E2E header. **Note** In-band OAM node-id identifies the network node in the given network domain. You must ensure that all the nodes in the domain have unique node-id configured using this command.
Step 18	**ipv6 ioam path-record** **Example:** `Device(config)# ipv6 ioam path-record`	Enables the In-band OAM process in a network node to start adding the node-data in the IPv6 traffic passing through the network node. **Note** Path-record must be configured on all nodes in domain.

	Command or Action	Purpose
Step 19	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# int g0/2`	Configures an interface and enters interface configuration mode.
Step 20	**ipv6 ioam profile** *profile-name* **Example:** `Device(config)# ipv6 ioam profile sc1`	Defines IPv6 In-band OAM profile name.
Step 21	**ipv6 ioam service-profile** *service-profile-name* **Example:** `Device(config-ioam-profile)# ipv6 ioam service-profile sf1`	Defines IPv6 In-band OAM service profile name.
Step 22	**exit** **Example:** `Device(config)# exit`	Exits interface configuration mode and returns to global configuration mode.

Configuring Intermediate Node

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 ioam service-chaining** *service-profile-name*
4. **lpc** *hexadecimal-value*
5. **prime-number** *number*
6. **secret-key** *hexadecimal-value*
7. **polynomial2** *hexadecimal-value*
8. **bits-in-random** *integer-value*
9. **exit**
10. **ipv6 ioam node-id** *node-id*
11. **ipv6 ioam path-record**
12. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
13. **ipv6 ioam service-profile** *service-profile-name*
14. **exit**
15. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
16. **ipv6 ioam service-profile** *service-profile-name*
17. **exit**

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 ioam service-chaining** *service-profile-name* **Example:** `Device(config)# ipv6 ioam service-chaining sf1`	Speicifies the IPv6 In-band OAM service chaining profile name.
Step 4	**lpc** *hexadecimal-value* **Example:** `Device(config-ioam-service-chaining)# lpc 1`	Specifies the value for lag range basis polynomial factor for constant. The range is 0-FFFFFFFF.
Step 5	**prime-number** *number* **Example:** `Device(config-ioam-service-chaining)# prime-number 61819`	Specifies the prime number required to calculate the share key. The range is 1-2147483647. **Note** Prime number must be unique across In-band OAM domain for a particular chain.
Step 6	**secret-key** *hexadecimal-value* **Example:** `Device(config-ioam-service-chaining)# secret-key 8435`	Specifies the secret key that is shared among service chain nodes. The range is 0-FFFFFFFF.
Step 7	**polynomial2** *hexadecimal-value* **Example:** `Device(config-ioam-service-chaining)# polynomial2 64b857ac`	Specifies the value for polynomial pre computed split share. The range is 0-FFFFFFFF.
Step 8	**bits-in-random** *integer-value* **Example:** `Device(config-ioam-service-chaining)# bits-in-random 31`	Specifies random number size. The range is 1-31.
Step 9	**exit** **Example:** `Device(config-ioam-service-chaining)# exit`	Exits In-band OAM service chaining configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 10	**ipv6 ioam node-id** *node-id* **Example:** `Device(config)# ipv6 ioam node-id 2`	Defines the node-id which will be sent in the In-band OAM data in the new E2E header. **Note** In-band OAM node-id identifies the network node in the given network domain. You must ensure that all the nodes in the domain have unique node-id configured using this command.
Step 11	**ipv6 ioam path-record** **Example:** `Device(config)# ipv6 ioam path-record`	Enables the In-band OAM process in a network node to start adding the node-data in the IPv6 traffic passing through the network node. **Note** Path-record must be configured on all nodes in the domain.
Step 12	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# int g0/1`	Configures an interface and enters interface configuration mode.
Step 13	**ipv6 ioam service-profile** *service-profile-name* **Example:** `Device(config-if)# ipv6 ioam service-profile sf1`	Defines IPv6 In-band OAM service profile name.
Step 14	**exit** **Example:** `Device(config)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 15	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# int g0/2`	Configures an interface and enters interface configuration mode.
Step 16	**ipv6 ioam service-profile** *service-profile-name* **Example:** `Device(config-if)# ipv6 ioam service-profile sf1`	Defines IPv6 In-band OAM service profile name.
Step 17	**exit** **Example:** `Device(config)# exit`	Exits interface configuration mode and returns to global configuration mode.

Configuring Decap Node

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length*
5. **exit**
6. **ipv6 ioam profile** *profile-name*
7. **service-chaining match-flow** *access-list-name* {**insert** | **analyze**}
8. **exit**
9. **ipv6 ioam service-chaining** *service-profile-name*
10. **service-chain** {**insert** | **analyze**}
11. **lpc** *hexadecimal-value*
12. **verifier-key** *key-value*
13. **polynomial2** *hexadecimal-value*
14. **prime-number** *number*
15. **secret-key** *hexadecimal-value*
16. **bits-in-random** *integer-value*
17. **exit**
18. **ipv6 ioam node-id** *node-id*
19. **ipv6 ioam path-record**
20. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
21. **ipv6 ioam service-profile** *service-profile-name*
22. **exit**
23. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
24. **ipv6 ioam service-profile** *service-profile-name*
25. **exit**
26. **show ipv6 ioam service-profile** *service-profile-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list acl_h1_h3`	Defines an IPv6 access list.
Step 4	**permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length* **Example:** `Device(config-ipv6-acl)# permit host ::A:1:1:0:6 host ::A:1:1:0:17`	Defines the source and destination IPv6 network to set permit conditions.
Step 5	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 acl configuration mode and returns to global configuration mode.
Step 6	**ipv6 ioam profile** *profile-name* **Example:** `Device(config-ioam-profile)# ipv6 ioam profile sc1`	Defines IPv6 In-band OAM profile name.
Step 7	**service-chaining match-flow** *access-list-name* {**insert** | **analyze**} **Example:** `Device(config-ioam-profile)# service-chaining match-flow acl_h1_h3 analyze`	Selects the ACL and traffic for service chain validation.
Step 8	**exit** **Example:** `Device(config-ioam-profile)# exit`	Exits In-band OAM profile configuration mode and returns to global configuration mode.
Step 9	**ipv6 ioam service-chaining** *service-profile-name* **Example:** `Device(config)# ipv6 ioam service-chaining sf1`	Specifies the IPv6 In-band OAM service chaining profile name.
Step 10	**service-chain** {**insert** | **analyze**} **Example:** `Device(config-ioam-service-chaining)# service-chain insert`	Specifies whether the given node is encap or decap node. **Note** The service-chain insert command will accept either of insert or analyze based on the node.
Step 11	**lpc** *hexadecimal-value* **Example:** `Device(config-ioam-service-chaining)# lpc 3`	Specifies the value for lag range basis polynomial factor for constant. The range is 0-FFFFFFFF.

	Command or Action	Purpose
Step 12	**verifier-key** *key-value* **Example:** `Device(config-ioam-service-chaining)# verifier-key 42520`	Specifies a key for service chain verifier.
Step 13	**polynomial2** *hexadecimal-value* **Example:** `Device(config-ioam-service-chaining)# polynomial2 2192c95c`	Specifies the value for polynomial pre computed split share. The range is 0-FFFFFFFF.
Step 14	**prime-number** *number* **Example:** `Device(config-ioam-service-chaining)# prime-number 61819`	Specifies the prime number required to calculate the share key. The range is 1-2147483647. **Note** Prime number must be unique across In-band OAM domain for a particular chain.
Step 15	**secret-key** *hexadecimal-value* **Example:** `Device(config-ioam-service-chaining)# secret-key BE29`	Specifies the secret key that is shared among service chain nodes. The range is 0-FFFFFFFF.
Step 16	**bits-in-random** *integer-value* **Example:** `Device(config-ioam-service-chaining)# bits-in-random 31`	Specifies random number size. The range is 1-31.
Step 17	**exit** **Example:** `Device(config)# exit`	Exits configuration mode and returns to global configuration mode.
Step 18	**ipv6 ioam node-id** *node-id* **Example:** `Device(config)# ipv6 ioam node-id 3`	Defines the node-id which will be sent in the In-band OAM data in the new E2E header. **Note** In-band OAM node-id identifies the network node in the given network domain. You must ensure that all the nodes in the domain have unique node-id configured using this command.
Step 19	**ipv6 ioam path-record** **Example:** `Device(config)# ipv6 ioam path-record`	Enables the In-band OAM process in a network node to start adding the node-data in the IPv6 traffic passing through the network node. **Note** Path-record must be configured on all nodes in the domain.

	Command or Action	Purpose
Step 20	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# int g0/1`	Configures an interface and enters interface configuration mode.
Step 21	**ipv6 ioam service-profile** *service-profile-name* **Example:** `Device(config-if)# ipv6 ioam service-profile sf1`	Assigns the IPv6 In-band OAM service profile to the interface.
Step 22	**exit** **Example:** `Device(config)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 23	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# int g0/2`	Configures an interface and enters interface configuration mode.
Step 24	**ipv6 ioam service-profile** *service-profile-name* **Example:** `Device(config-if)# ipv6 ioam service-profile sf1`	Assigns the IPv6 In-band OAM service profile to the interface.
Step 25	**exit** **Example:** `Device(config)# exit`	Exits interface configuration mode and returns to global configuration mode.

Step 26	**show ipv6 ioam service-profile** *service-profile-name*
	
	Example:
	`Device(config)# show ipv6 ioam service-profile sf1`
	
	`Service Function Chaining Statistics:`
	`SFC Validation Success count : 5`
	`SFC Validation Failure count : 0`	

Configuring In-band OAM for IPv6 Sequence Numbering

Configuring Encap Node

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length*
5. **exit**
6. **ipv6 ioam profile** *profile-name*
7. **ppc match-flow** *access-list-name* {**insert** | **analyze**}
8. **exit**
9. **ipv6 ioam node-id** *node-id*
10. **ipv6 ioam path-record**
11. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
12. **ipv6 ioam profile** *profile-name*
13. **exit**

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 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list acl_h1_h3`	Defines an IPv6 access list.
Step 4	**permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length* **Example:** `Device(config-ipv6-acl)# permit host ::A:1:1:0:6 host ::A:1:1:0:17`	Defines the source and destination IPv6 network to set permit conditions.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 ACL configuration mode and returns to global configuration mode.
Step 6	**ipv6 ioam profile** *profile-name* **Example:** `Device(config)# ipv6 ioam profile sc1`	Defines IPv6 In-band OAM profile name.
Step 7	**ppc match-flow** *access-list-name* {**insert** | **analyze**} **Example:** `Device(config-ioam-profile)# ppc match-flow acl_h1_h3 insert`	Inserts path packet counter relevant details in E2E header that matches each of the ACL line entry. **Note** The insert option is enabled at the ingress edge router.
Step 8	**exit** **Example:** `Device(config-ioam-profile)# exit`	Exits IPv6 ACL configuration mode and returns to global configuration mode.
Step 9	**ipv6 ioam node-id** *node-id* **Example:** `Device(config)# ipv6 ioam node-id 3`	Defines the node-id which will be sent in the In-band OAM data in the new E2E header. **Note** In-band OAM node-id identifies the network node in the given network domain. You must ensure that all the nodes in the domain have unique node-id configured using this command.
Step 10	**ipv6 ioam path-record** **Example:** `Device(config)# ipv6 ioam path-record`	Enables the In-band OAM process in a network node to start adding the node-data in the IPv6 traffic passing through the network node. **Note** Path-record must be configured on all nodes in the domain.
Step 11	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# int g0/2`	Configures an interface and enters interface configuration mode.
Step 12	**ipv6 ioam profile** *profile-name* **Example:** `Device(config)# ipv6 ioam profile sc1`	Defines IPv6 In-band OAM profile name.
Step 13	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Decap Node

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length*
5. **exit**
6. **ipv6 ioam profile** *profile-name*
7. **ppc match-flow** *access-list-name* {**insert** | **analyze**}
8. **collect ppc**
9. **exit**
10. **ipv6 ioam node-id** *node-id*
11. **ipv6 ioam path-record**
12. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
13. **ipv6 ioam profile** *profile-name*
14. **exit**
15. **show ipv6 ioam path-packet-counter**

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 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list acl_h1_h3`	Defines an IPv6 access list.
Step 4	**permit host** *source-ipv6-prefix / prefix-length* **host** *destination-ipv6-prefix / prefix-length* **Example:** `Device(config-ipv6-acl)# permit host ::A:1:1:0:6 host ::A:1:1:0:17`	Defines the source and destination IPv6 network to set permit conditions.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 ACL configuration mode and returns to global configuration mode.
Step 6	**ipv6 ioam profile** *profile-name* **Example:** `Device(config)# ipv6 ioam profile sc1`	Defines IPv6 In-band OAM profile name.
Step 7	**ppc match-flow** *access-list-name* {**insert** | **analyze**} **Example:** `Device(config-ioam-profile)# ppc match-flow acl_h1_h3 analyze`	Retrieves path packet counter keys and performs validation.
Step 8	**collect ppc** **Example:** `Device(config-ioam-profile)# collect ppc`	Collects path packet counter information for configured access-list in In-band OAM profile.
Step 9	**exit** **Example:** `Device(config-ioam-profile)# exit`	Exits IPv6 ACL configuration mode and returns to global configuration mode.
Step 10	**ipv6 ioam node-id** *node-id* **Example:** `Device(config)# ipv6 ioam node-id 3`	Defines the node-id which will be sent in the In-band OAM data in the new E2E header. **Note** In-band OAM node-id identifies the network node in the given network domain. You must ensure that all the nodes in the domain have unique node-id configured using this command.
Step 11	**ipv6 ioam path-record** **Example:** `Device(config)# ipv6 ioam path-record`	Enables the In-band OAM process in a network node to start adding the node-data in the IPv6 traffic passing through the network node. **Note** Path-record must be configured on all nodes in the domain.
Step 12	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# int g0/3`	Configures an interface and enters interface configuration mode.
Step 13	**ipv6 ioam profile** *profile-name* **Example:** `Device(config)# ipv6 ioam profile sc1`	Defines IPv6 In-band OAM profile name.

	Command or Action	Purpose
Step 14	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Step 15	**show ipv6 ioam path-packet-counter**
	
	Example:
	`Device(config)#  Profile sc1: Path Packet Counter Enabled:`
	`PPC analyze ACL : acl_h1_h1, refcount 1`
	`1 permit ipv6 ::a:1:1:0:17/128  ::a:1:1:0:1f/128`
	
	`Rx statistics:`
	`Total Rx Packets 5 lost packets 0`
	`Reordered packets 0 duplicate packets 0`
	`Highest Seq Recvd 5`	

Enabling Netflow Collection/Export

For using the flexible netflow (FNF) module on the router, you should configure the destination address/protocol/port of the netflow collector where we would like to send/export the collected In-band OAM data.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *flow-exporter-name*
4. **destination** {*destination IPv4 address / hostname* | *destination IPv6 address / hostname*}
5. **transport udp** *port-value*
6. **exit**
7. **ipv6 ioam fnf-exporter** *flow-exporter-name*
8. **ipv6 ioam collect traffic-statistics**
9. **ipv6 ioam collect service-chaining match-flow** *IPv6 access-list-name*
10. **ipv6 ioam collect flow-debug-summary match-flow** *IPv6 access-list-name*
11. **ipv6 ioam collect ppc**
12. **exit**

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	**flow exporter** *flow-exporter-name* **Example:** `Device(config)# flow exporter ioam-exp`	Specifies flow exporter name.
Step 4	**destination** {*destination IPv4 address / hostname* | *destination IPv6 address / hostname*} **Example:** `Device(config-flow-exporter)# destination 172.16.1.254`	Specifies the source and destination IPv6 network to set permit conditions.
Step 5	**transport udp** *port-value* **Example:** `Device(config-flow-exporter)# transport udp 2100`	Specifies the transport protocol value. The range is 1-65535.
Step 6	**exit** **Example:** `Device(config-flow-exporter)# exit`	Exits flow exporter configuration mode and returns to global configuration mode.
Step 7	**ipv6 ioam fnf-exporter** *flow-exporter-name* **Example:** `Device(config)# ipv6 ioam fnf-exporter ioam-exp`	Enable In-band OAM for IPv6 process to use the configured fnf exporter for sending all the collected data to netflow exporter.
Step 8	**ipv6 ioam collect traffic-statistics** **Example:** `Device(config)# ipv6 ioam collect traffic-statistics`	Enables the In-band OAM process to start collecting the traffic statistics and periodically export/sent the same using FNF.
Step 9	**ipv6 ioam collect service-chaining match-flow** *IPv6 access-list-name* **Example:** `Device(config)# ipv6 ioam collect service-chaining match-flow acl_h1_h3`	Specifies IPv6 access list name for match flow.

	Command or Action	Purpose
Step 10	**ipv6 ioam collect flow-debug-summary match-flow** *IPv6 access-list-name* **Example:** `Device(config)# ipv6 ioam collect flow-debug-summary match-flow acl_h1_h3`	Enables the collection of path-record between source and destination based on the configured access-list. The collected path-records can be exported to netflow collector.
Step 11	**ipv6 ioam collect ppc** **Example:** `Device(config)# ipv6 ioam collect ppc`	Collects path packet counter information for configured access-list in the In-band OAM profile.
Step 12	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC configuration mode.

Troubleshooting Tips

Use the following debug commands if you have any configuration issues:

- **debug ipv6 ioam api**- Enables the API level debugging for CLI related changes.
- **debug ipv6 ioam error**- Enables the error debugs for troubleshooting any unexpected behavior.
- **debug ipv6 ioam event**- Enables In-band OAM event debugging which can be used for enabling netflow update events debugging.
- **debug ipv6 ioam packet**- Enables packet level debugging.

  Note Enabling packet level debugging will dump part of every packet. It is recommended that this debug be used when the traffic load is less.

- **debug ipv6 ioam ppc**- Enables debugs related to sequence numbering.

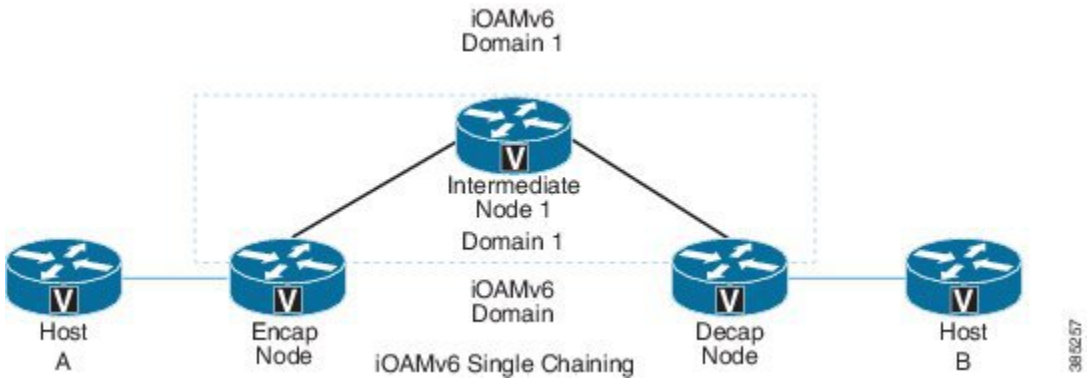
Use Cases for In-band OAM for IPv6

In-band OAM for IPv6 - Single Service Chain/Single Path Verification

You can use In-band OAM for IPv6 for service chain networks with multiple hosts and encap and decap nodes.

The figure below illustrates an In-band OAM for IPv6 Single Service Chain/Single Path Verification topology with two In-band OAM for IPv6 hosts, an encap node, a decap node and an intermediate node.

Figure 1: In-band OAM for IPv6 Single Service Chain/Single Path Verification



The following In-band OAM for IPv6 Single Service Chain/Single Path Verification configuration consists of three hosts, an encap node, a decap node, and an intermediate node:

To configure encap node:

```
enable
 configure terminal
  ipv6 access-list acl_h1_h3
  permit host ::A:1:1:0:6host ::A:1:1:0:17
  exit
 ipv6 ioam profile sc1
  service-chaining match-flow acl_h1_h3 insert
  path-record match-flow acl_h1_h3 encap
  ppc match-flow acl_h1_h3 insert
  trace-type N_IF_TS_APP
  trace-elts 3
  trace-tsp 2
  exit
 ipv6 ioam service-chaining sf1
  service-chain insert
  prime-number 61819
  secret-key 2878
  lpc 6ddfd460
  polynomial2 43259140
  bits-in-random 31
  exit
   ipv6 ioam node-id 1
   ipv6 ioam path-record
   int g0/2
   ipv6 ioam profile sc1
   ipv6 ioam service-profile sf1
 end
```

To configure intermediate node:

```
enable
 configure terminal
  ipv6 ioam service-chaining sf1
  lpc 1
  prime-number 6ddfd463
  secret-key f258fe6
  polynomial2 64b857ac
  bits-in-random 31
  exit
 ipv6 ioam node-id 2
  ipv6 ioam path-record
  int g0/1
```

```
  ipv6 ioam service-profile sf1
  exit
 int g0/2
 ipv6 ioam service-profile sf1
```

To configure decap node:

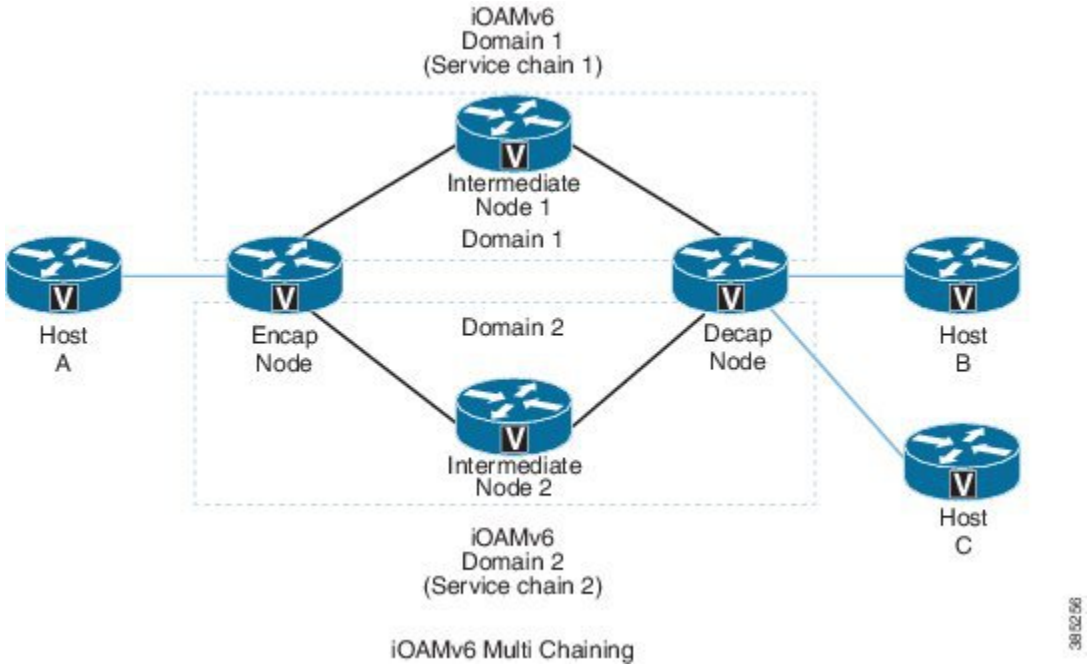
```
enable
 configure terminal
  ipv6 access-list acl_h1_h3
  permit host ::A:1:1:0:6 host ::A:1:1:0:17
  exit
 ipv6 ioam profile sc1
  service-chaining match-flow acl_h1_h3 analyze
   path-record match-flow acl_h1_h3 decap
   ppc match-flow acl_h1_h3 analyze
   collect ppc
   exit
  ipv6 ioam service-chaining sf1
   service-chain analyze
   lpc 3
   verifier-key de0598
   polynomial2 2192c95c
   prime-number 6ddfd463
   secret-key 4371091d
   bits-in-random 31
   exit
  ipv6 ioam node-id 3
   ipv6 ioam path-record
   int g0/1
    ipv6 ioam service-profile sf1
    exit
   int g0/3
    ipv6 ioam profile sc1
    ipv6 ioam service-profile sf1
    exit
```

In-band OAM for IPv6 - Multiple Service Chain/Multiple Path Verification

You can use In-band OAM for IPv6 for Multiple Service Chain/Multiple Path Verification with multiple In-band OAM for IPv6 domains and different intermediate nodes.

The figure below illustrates In-band OAM for IPv6 Multiple Service Chain/Multiple Path Verification deployment having two In-band OAM for IPv6 domains.

Figure 2: In-band OAM for IPv6 Multi Chaining/Multiple Path Verification



The following In-band OAM for IPv6 Multiple Service Chain/Multiple Path Verification configuration consists of three hosts, an encap node, a decap node, and two intermediate nodes forming the upper node (domain 1) and lower node (domain 2):

To configure encap node:

```
enable
 configure terminal
  ipv6 access-list acl_h1_h3
  permit host ::A:1:1:0:6 host ::A:1:1:0:17
  exit
 ipv6 ioam profile sc1
  service-chaining match-flow acl_h1_h3 insert
  path-record match-flow acl_h1_h3 encap
  ppc match-flow acl_h1_h3 insert
  trace-type N_IF_TS_APP
  trace-elts 3
  trace-tsp 2
  exit
 ipv6 ioam service-chaining sf1
  service-chain insert
  prime-number 6ddfd463
  secret-key 23939b16
  lpc 6ddfd460
  polynomial2 43259140
  bits-in-random 31
  exit
 ipv6 access-list acl_h1_h6
  permit host ::A:1:1:0:6 host ::A:1:1:0:1F
  exit
 ipv6 ioam profile sc2
  service-chaining match-flow acl_h1_h6 insert
  path-record match-flow acl_h1_h6 encap
```

```
 ppc match-flow acl_h1_h6 insert
 trace-type N_IF_TS_APP
 trace-elts 3
 trace-tsp 2
 exit
ipv6 ioam service-chaining sf2
 service-chain insert
 prime-number7f13d66d
 secret-key 28bacfd2
 lpc 3
 polynomial2 21a17f97
 bits-in-random 31
 exit
ipv6 ioam node-id 1
ipv6 ioam path-record
int g0/2
 ipv6 ioam profile sc1
 ipv6 ioam service-profile sf1
 exit
int g0/3
ipv6 ioam profile sc2
ipv6 ioam service-profile sf2
```

To configure the Intermediate node 1 in In-band OAM for IPv6 Domain 1:

```
enable
conf t
 ipv6 ioam service-chaining sf1
 lpc 1
 prime-number 6ddfd463
 secret-key f258fe6
 polynomial2 64b857ac
 bits-in-random 31
 exit
ipv6 ioam node-id 2
ipv6 ioam path-record
int g0/1
 ipv6 ioam service-profile sf1
 exit
int g0/2
ipv6 ioam service-profile sf1
```

To configure decap node:

```
enable
conf t
 ipv6 access-list acl_h1_h3
 permit host ::A:1:1:0:6 host ::A:1:1:0:17
 exit
ipv6 ioam profile sc1
 service-chaining match-flow acl_h1_h3 analyze
 path-record match-flow acl_h1_h3 decap
 ppc match-flow acl_h1_h3 analyze
 collect ppc
 exit
ipv6 ioam service-chaining sf1
 service-chain analyze
 lpc 3
 verifier-key de0598
 polynomial2 2192c95c
 prime-number 6ddfd463
 secret-key 4371091d
 bits-in-random 31
 exit
ipv6 access-list acl_h1_h6
 permit host ::A:1:1:0:6 host ::A:1:1:0:1F
 exit
ipv6 ioam profile sc2
 service-chaining match-flow acl_h1_h6 analyze
 path-record match-flow acl_h1_h6 decap
 ppc match-flow acl_h1_h6 analyze
 collect ppc
```

```
 exit
ipv6 ioam service-chaining sf2
 service-chain analyze
 lpc 7f13d66a
 verifier-key 1c53f3e5
 polynomial2 3af0f2c0
 prime-number 7f13d66d
 secret-key 4def63a1
 bits-in-random 31
 exit
ipv6 ioam node-id 3
ipv6 ioam path-record
int g0/1
 ipv6 ioam service-profile sf1
 exit
 int g0/2
  ipv6 ioam service-profile sf2
  exit
 int g0/3
  ipv6 ioam profile sc1
  ipv6 ioam service-profile sf1
  exitexit
 int g0/4
  ipv6 ioam profile sc2
  ipv6 ioam service-profile sf2
  end
```

To configure the Intermediate node 2 in In-band OAM for IPv6 Domain 2:

```
enable
conf t
 ipv6 ioam service-chaining sf2
  lpc 1
  prime-number 7f13d66d
  secret-key ddd8e5
  polynomial2 4bee597b
  bits-in-random 31
  exit
 ipv6 ioam node-id 4
 ipv6 ioam path-record
 int g0/1
  ipv6 ioam service-profile sf2
  exit
 int g0/2
  ipv6 ioam service-profile sf2
```

Additional References for In-band OAM for IPv6

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Standards and RFCs

Standard/RFC	Title
Requirements for In-band OAM	*draft-brockners-inband-oam-requirements-01*
Proof of Transit	*draft-brockners-proof-of-transit-01*

Standard/RFC	Title
Data Formats for In-band OAM	*draft-brockners-inband-oam-data-01*
Encapsulations for In-band OAM Data	*draft-brockners-inband-oam-transport-01*

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

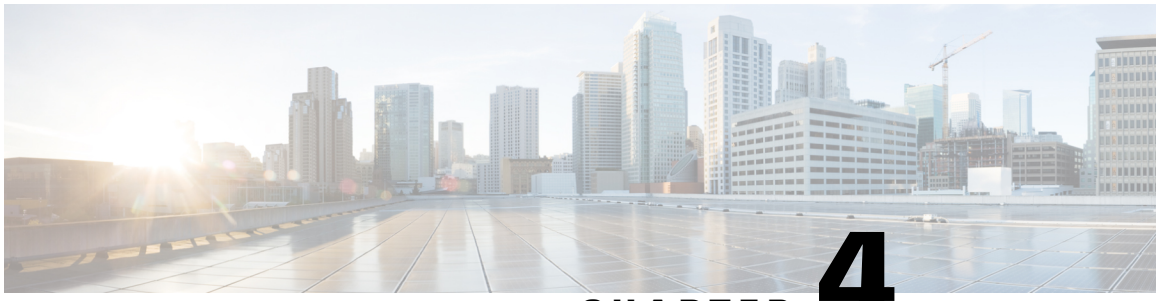
Feature Information for In-band OAM for IPv6

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 3: Feature Information for In-band OAM for IPv6

Feature Name	Releases	Feature Configuration Information
In-band OAM for IPv6	15.6(3)M	The In-band OAM for IPv6 feature supports "in-band" operation, administration, and maintenance (iOAM) for IPv6 which provides advanced options for recording OAM information in an IPv6 packet during its traversal through a particular network domain. The following commands were introduced: **ipv6 ioam profile**,**service-chaining match-flow**.



CHAPTER 4

SSH Support Over IPv6

Secure Shell (SSH) provides support for IPv6 addresses that enable a Cisco device to accept and establish secure, encrypted connections with remote IPv6 nodes over an IPv6 transport.

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.

Prerequisites for SSH Support over IPv6

- An IPsec (Data Encryption Standard [DES] or 3DES) encryption software image is loaded on your device. IPv6 transport for the SSH server and SSH client requires an IPsec encryption software image.
- A hostname and host domain are configured for your device.
- A Rivest, Shamir, and Adelman (RSA) key pair, which automatically enables SSH, is generated for your device.
- A user authentication mechanism for local or remote access is configured on your device.

- To authenticate SSH clients, configure TACACS+ or RADIUS over an IPv4 transport and then connect to an SSH server over an IPv6 transport.

The basic restrictions for SSH over an IPv4 transport apply to SSH over an IPv6 transport. The use of locally stored usernames and passwords is the only user authentication mechanism supported by SSH over an IPv6 transport. TACACS+ and RADIUS user authentication mechanisms are not supported over an IPv6 transport.

Information About SSH Support over IPv6

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.

How to Enable SSH Support over IPv6

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.

	Command or Action	Purpose
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.
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.

Configuration Examples for SSH Support over IPv6

Example: Enabling SSH on an IPv6 Device

```
Device# configure terminal
Device(config)# ip ssh
Device(config)# exit
Device(config)# ssh -l userid1 2001:db8:2222:1044::72
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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

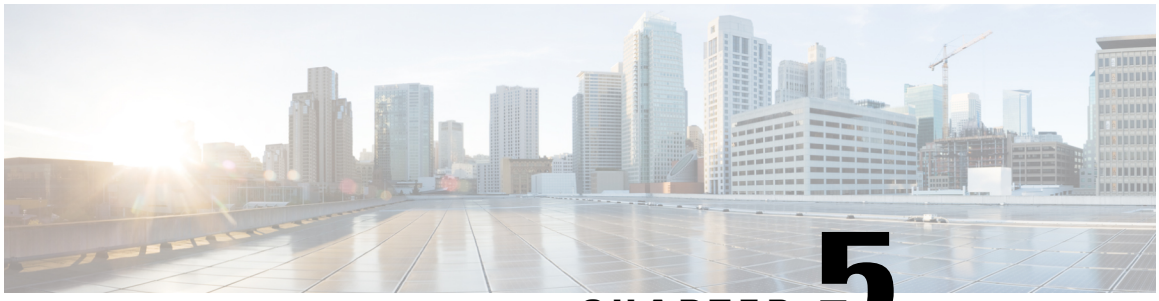
Feature Information for SSH Support over IPv6

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 4: Feature Information for SSH Support over IPv6

Feature Name	Releases	Feature Information
SSH Support over IPv6	12.2(8)T 12.2(17a)SX1 12.2(25)SEE 12.2(25)SG 12.2(33)SRA 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	SSH provides support for IPv6 addresses that enable a Cisco device to accept and establish secure, encrypted connections with remote IPv6 nodes over an IPv6 transport. The following commands were introduced or modified: **ip ssh**, **ssh**.



CHAPTER 5

SNMP over IPv6

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.

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 SNMP over IPv6

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.

How to Configure SNMP over IPv6

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**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• 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).
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**	(Optional) Configures a new user to an existing SNMP group.

	Command or Action	Purpose
	| **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`	**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.

Configuration Examples for SNMP over IPv6

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 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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 SNMP over IPv6

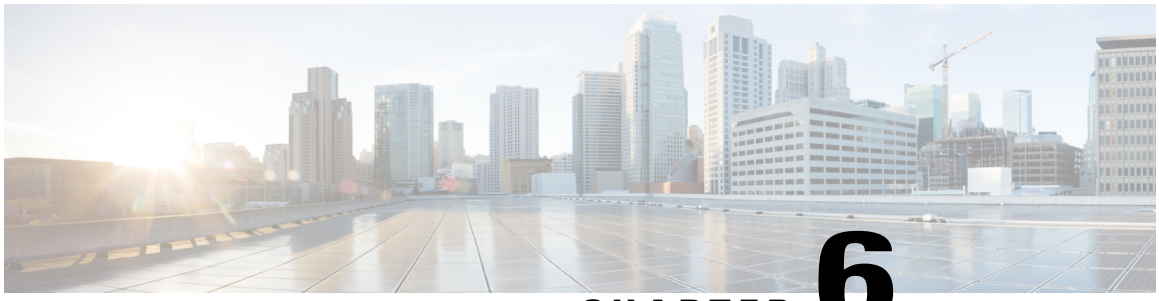
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 5: Feature Information for SNMP over IPv6

Feature Name	Releases	Feature Information
SNMP over IPv6	12.2(33)SRB 12.2(33)SXI 12.2(44)SE 12.2(44)SG 12.3(14)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	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. The following commands were introduced or modified: **snmp-server community**, **snmp-server enable traps**, **snmp-server engineID remote**, **snmp-server group**, **snmp-server host**, **snmp-server user**.

Feature Name	Releases	Feature Information
SNMPv3--3DES and AES Encryption Support	12.2(33)SRB 12.2(33)SXI 12.2(50)SG 12.2(52)SE 12.4(2)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports the SNMPv3 - 3DES and AES Encryption Support feature. No commands were introduced or modified.



CHAPTER 6

IPv6 MIBs

This document is about MIBs that are implemented for IPv6. 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 they are implemented only for IPv6 objects and tables.

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 IPv6 MIBs

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.

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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

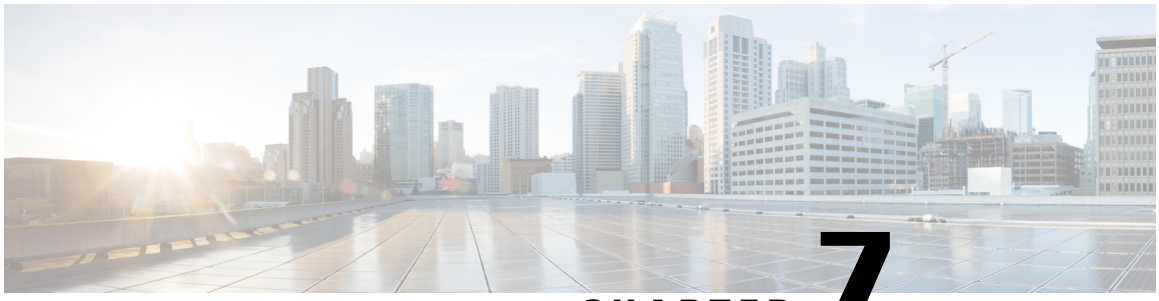
Feature Information for IPv6 MIBs

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 6: Feature Information for IPv6 MIBs

Feature Name	Releases	Feature Information
IPv6 MIBs	12.0(22)S 12.2(14)S 12.2(15)T 12.2(28)SB 12.2(33)SRA 12.2(50)SY 15.0(1)SY Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.9S	This feature is supported in IPv6. No commands were introduced or modified. In Cisco IOS XE Release 3.9S, support was added for the Cisco ISR 4400 Series Routers.
IPv6 Services: RFC 4293 IP-MIB (IPv6 Only) and RFC 4292 IP-FORWARD-MIB (IPv6 Only)	12.2(33)SRC 12.2(50)SY 12.2(54)SG 12.2(58)SE 15.0(2)SG 15.0(1)SY 15.1(3)T Cisco IOS XE Release 2.1 3.2SG	IP-FORWARD-MIB and IP-MIB were updated to RFC 4292 and RFC 4293 standards, respectively. No commands were introduced or modified.



CHAPTER 7

IPv6 Embedded Management Components

Cisco IPv6 embedded management components have IPv6-compliant operability in IPv6 and hybrid IPv6 and IPv4 networks. This document describes the following embedded management components: syslog, config logger, TCL, NETCONF, and the SOAP message format.

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 IPv6 Embedded Management Components

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).

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).

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.

How to Configure IPv6 Embedded Management Components

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.

Configuration Examples for IPv6 Embedded Management Components

Example: Configuring Syslog over IPv6

```
Device(config)# logging host ipv6 AAAA:BBBB:CCCC:DDDD::FFFF transport tcp port 1470
```

Additional References for IPv6 Embedded Management Components

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco_IOS_IPv6_Feature_ Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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 IPv6 Embedded Management Components

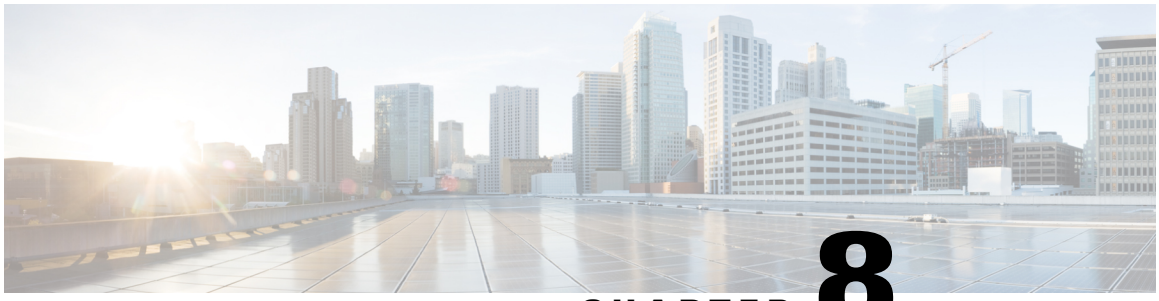
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 7: Feature Information for IPv6 Embedded Management Components

Feature Name	Releases	Feature Information
IPv6: Config Logger	12.2(33)SB 12.2(33)SRC 12.2(50)SG 12.2(50)SY 12.4(20)T 15.0(1)SY 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports this feature. No commands were introduced or modified.
IPv6: NETCONF	12.2(33)SB 12.2(33)SRC 12.2(50)SG 12.2(50)SY 12.4(20)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports this feature. No commands were introduced or modified.

Feature Name	Releases	Feature Information
IPv6 Support in SOAP	12.2(33)SB 12.2(33)SRC 12.2(50)SG 12.2(50)SY 12.4(20)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports this feature. No commands were introduced or modified.
IPv6: TCL	12.2(33)SB 12.2(33)SRC 12.2(50)SG 12.2(50)SY 12.4(20)T 15.0(1)SY 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports this feature. No commands were introduced or modified.
Syslog over IPv6	12.2(33)SB 12.2(33)SRC 12.2(33)SXI 12.2(44)SE 12.2(44)SG 12.4(4)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	The Cisco syslog process in IPv6 allows users to log syslog messages to external syslog servers and hosts with IPv6 addresses. The following command was introduced: **logging host**.



CHAPTER 8

IPv6 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 provides the infrastructure for the automated configuration of large numbers of network devices. The document describes CNS agents supported in IPv6.

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 IPv6 CNS Agents

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

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.

Additional References for IPv6 IOS Firewall

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
IPv6 commands	Cisco IOS IPv6 Command Reference
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

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

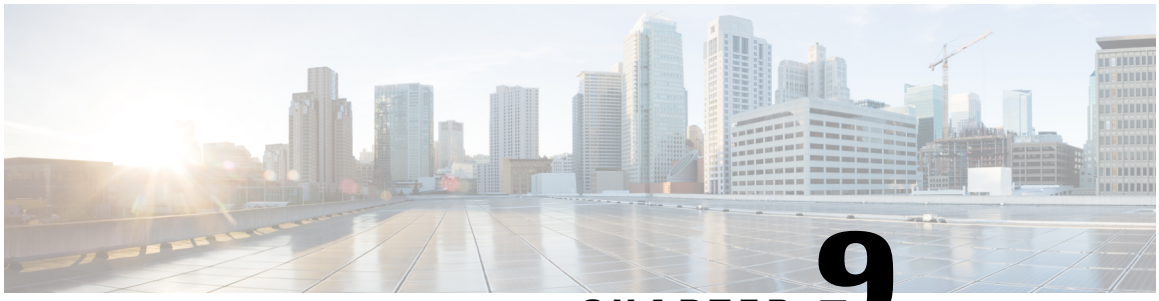
Feature Information for IPv6 CNS Agents

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 8: Feature Information for IPv6 CNS Agents

Feature Name	Releases	Feature Information
IPv6 CNS Agents	12.2(33)SB 12.2(33)SRC 12.2(50)SY 12.4(20)T Cisco IOS XE Release 3.9S	CNS configuration and event agents use a CNS configuration engine to provide methods for automating initial 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. No commands were introduced or modified. In Cisco IOS XE Release 3.9S, support was added for the Cisco CSR 1000V.



CHAPTER 9

IPv6 HTTP(S)

Hypertext Transfer Protocol server HTTP(S) is a Cisco IPv6 embedded management component. Cisco IPv6 embedded management components have IPv6-compliant operability in IPv6 and hybrid IPv6 and IPv4 networks.

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 IPv6 HTTP(S)

Cisco IPv6 Embedded Management Components

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

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.

How to Configure IPv6 HTTP(S)

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 IPv6 HTTP(S)

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
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IP access list commands	*Cisco IOS Security Command Reference*
Configuring IP access lists	"Creating an IP Access List and Applying It to an Interface"

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

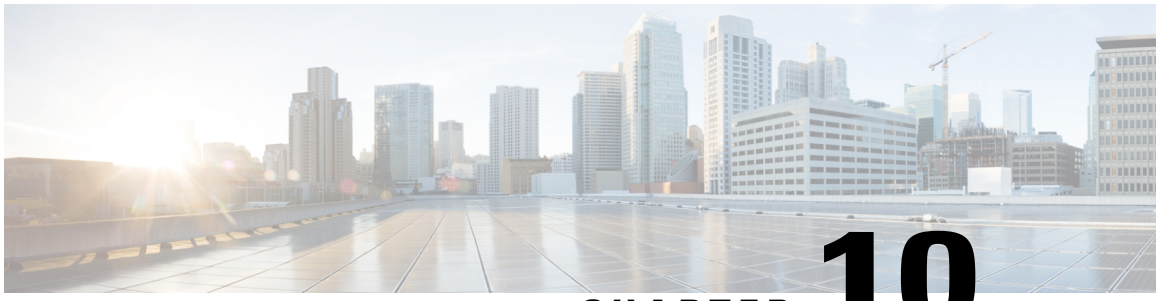
Feature Information for IPv6 HTTP(S)

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 9: Feature Information for IPv6 HTTP(S)

Feature Name	Releases	Feature Information
IPv6 HTTP(S)	12.2(33)SB 12.2(33)SRC 12.2(50)SY 12.4(20)T 15.0(1)SY Cisco IOS XE Release 3.8S	This feature enables the HTTP(S) client and server to support IPv6 addresses. The following command was modified: **ip http server**.



CHAPTER 10

IP SLAs for IPv6

Cisco IP Service Level Agreements (SLAs) are a portfolio of technology embedded in most devices that run Cisco software. SLAs allow Cisco customers to analyze IPv6 service levels for IPv6 applications and services, increase productivity, lower operational costs, and reduce the frequency of network outages.

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 IP SLAs for IPv6

Cisco IPv6 Embedded Management Components

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

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.

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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

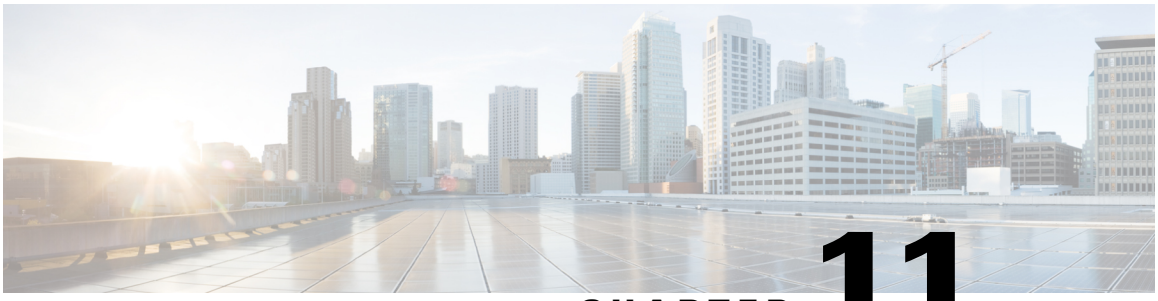
Feature Information for IP SLAs for IPv6

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 10: Feature Information for IP SLAs for IPv6

Feature Name	Releases	Feature Information
IP SLAs for IPv6	12.2(33)SRC 12.2(50)SG 12.2(50)SY 12.4(20)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports this feature. No commands were introduced or modified.



CHAPTER 11

IPv6 RFCs

Standards and RFCs

RFCs	Title
RFC 1195	*Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*
RFC 1267	*A Border Gateway Protocol 3 (BGP-3)*
RFC 1305	*Network Time Protocol (Version 3) Specification, Implementation and Analysis*
RFC 1583	*OSPF version 2*
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1886	*DNS Extensions to Support IP version 6*
RFC 1918	*Address Allocation for Private Internets*
RFC 1981	*Path MTU Discovery for IP version 6*
RFC 2080	*RIPng for IPv6*
RFC 2281	*Cisco Hot Standby Router Protocol (HSRP)*
RFC 2332	*NBMA Next Hop Resolution Protocol (NHRP)*
RFC 2373	*IP Version 6 Addressing Architecture*
RFC 2374	*An Aggregatable Global Unicast Address Format*
RFC 2375	*IPv6 Multicast Address Assignments*
RFC 2401	*Security Architecture for the Internet Protocol*

RFCs	Title
RFC 2402	*IP Authentication Header*
RFC 2404	*The Use of Hash Message Authentication Code Federal Information Processing Standard 180-1 within Encapsulating Security Payload and Authentication Header*
RFC 2406	*IP Encapsulating Security Payload (ESP)*
RFC 2407	*The Internet Security Domain of Interpretation for ISAKMP*
RFC 2408	*Internet Security Association and Key Management Protocol*
RFC 2409	*Internet Key Exchange (IKE)*
RFC 2427	*Multiprotocol Interconnect over Frame Relay*
RFC 2428	*FTP Extensions for IPv6 and NATs*
RFC 2460	*Internet Protocol, Version 6 (IPv6) Specification*
RFC 2461	*Neighbor Discovery for IP Version 6 (IPv6)*
RFC 2462	*IPv6 Stateless Address Autoconfiguration*
RFC 2463	*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
RFC 2464	*Transmission of IPv6 Packets over Ethernet*
RFC 2467	*Transmission of IPv6 Packets over FDDI*
RFC 2472	*IP Version 6 over PPP*
RFC 2473	*Generic Packet Tunneling in IPv6 Specification*
RFC 2474	*Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
RFC 2475	*An Architecture for Differentiated Services Framework*
RFC 2492	*IPv6 over ATM*
RFC 2545	*Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*

RFCs	Title
RFC 2590	*Transmission of IPv6 Packets over Frame Relay Specification*
RFC 2597	*Assured Forwarding PHB*
RFC 2598	*An Expedited Forwarding PHB*
RFC 2640	*Internet Protocol, Version 6 Specification*
RFC 2684	*Multiprotocol Encapsulation over ATM Adaptation Layer 5*
RFC 2697	*A Single Rate Three Color Marker*
RFC 2698	*A Two Rate Three Color Marker*
RFC 2710	*Multicast Listener Discovery (MLD) for IPv6*
RFC 2711	*IPv6 Router Alert Option*
RFC 2732	*Format for Literal IPv6 Addresses in URLs*
RFC 2765	*Stateless IP/ICMP Translation Algorithm (SIIT)*
RFC 2766	*Network Address Translation-Protocol Translation (NAT-PT)*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2893	*Transition Mechanisms for IPv6 Hosts and Routers*
RFC 3056	*Connection of IPv6 Domains via IPv4 Clouds*
RFC 3068	*An Anycast Prefix for 6to4 Relay Routers*
RFC 3095	*RObust Header Compression (ROHC): Framework and Four Profiles: RTP, UDP, ESP, and Uncompressed*
RFC 3107	*Carrying Label Information in BGP-4*
RFC 3137	*OSPF Stub Router Advertisement*
RFC 3147	*Generic Routing Encapsulation over CLNS*
RFC 3152	*Delegation of IP6.ARPA*
RFC 3162	*RADIUS and IPv6*

RFCs	Title
RFC 3315	*Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3319	*Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiated Protocol (SIP) Servers*
RFC 3392	*Capabilities Advertisement with BGP-4*
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 3513	*Internet Protocol Version 6 (IPv6) Addressing Architecture*
RFC 3576	*Change of Authorization*
RFC 3587	*IPv6 Global Unicast Address Format*
RFC 3590	*Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*
RFC 3596	*DNS Extensions to Support IP Version 6*
RFC 3633	*DHCP IPv6 Prefix Delegation*
RFC 3646	*DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3697	*IPv6 Flow Label Specification*
RFC 3736	*Stateless DHCP Service for IPv6*
RFC 3756	*IPv6 Neighbor Discovery (ND) Trust Models and Threats*
RFC 3759	*RObust Header Compression (ROHC): Terminology and Channel Mapping Examples*
RFC 3775	*Mobility Support in IPv6*
RFC 3810	*Multicast Listener Discovery Version 2 (MLDv2) for IPv6*
RFC 3846	*Mobile IPv4 Extension for Carrying Network Access Identifiers*

RFCs	Title
RFC 3879	*Deprecating Site Local Addresses*
RFC 3898	*Network Information Service (NIS) Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3954	*Cisco Systems NetFlow Services Export Version 9*
RFC 3956	*Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*
RFC 3963	*Network Mobility (NEMO) Basic Support Protocol*
RFC 3971	*SEcure Neighbor Discovery (SEND)*
RFC 3972	*Cryptographically Generated Addresses (CGA)*
RFC 4007	*IPv6 Scoped Address Architecture*
RFC 4075	*Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6*
RFC 4087	*IP Tunnel MIB*
RFC 4091	*The Alternative Network Address Types (ANAT) Semantics for the Session Description Protocol (SDP) Grouping Framework*
RFC 4092	*Usage of the Session Description Protocol (SDP) Alternative Network Address Types (ANAT) Semantics in the Session Initiation Protocol (SIP)*
RFC 4109	*Algorithms for Internet Key Exchange version 1 (IKEv1)*
RFC 4191	*Default Router Preferences and More-Specific Routes*
RFC 4193	*Unique Local IPv6 Unicast Addresses*
RFC 4214	*Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*
RFC 4242	*Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 4282	*The Network Access Identifier*
RFC 4283	*Mobile Node Identifier Option for Mobile IPv6*

RFCs	Title
RFC 4285	*Authentication Protocol for Mobile IPv6*
RFC 4291	*IP Version 6 Addressing Architecture*
RFC 4292	*IP Forwarding Table MIB*
RFC 4293	*Management Information Base for the Internet Protocol (IP)*
RFC 4302	*IP Authentication Header*
RFC 4306	*Internet Key Exchange (IKEv2) Protocol*
RFC 4308	*Cryptographic Suites for IPsec*
RFC 4364	*BGP MPLS/IP Virtual Private Networks (VPNs)*
RFC 4382	*MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base*
RFC 4443	*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
RFC 4552	*Authentication/Confidentiality for OSPFv3*
RFC 4594	*Configuration Guidelines for DiffServ Service Classes*
RFC 4601	*Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification*
RFC 4610	*Anycast-RP Using Protocol Independent Multicast (PIM)*
RFC 4649	*Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option*
RFC 4659	*BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*
RFC 4724	*Graceful Restart Mechanism for BGP*
RFC 4798	*Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*
RFC 4818	*RADIUS Delegated-IPv6-Prefix Attribute*
RFC 4861	*Neighbor Discovery for IP version 6 (IPv6)*

RFCs	Title
RFC 4862	*IPv6 Stateless Address Autoconfiguration*
RFC 4884	*Extended ICMP to Support Multi-Part Messages*
RFC 4885	*Network Mobility Support Terminology*
RFC 4887	*Network Mobility Home Network Models*
RFC 5015	*Bidirectional Protocol Independent Multicast (BIDIR-PIM)*
RFC 5059	*Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*
RFC 5072	*IPv6 over PPP*
RFC 5095	*Deprecation of Type 0 Routing Headers in IPv6*
RFC 5120	*M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)*
RFC 5130	*A Policy Control Mechanism in IS-IS Using Administrative Tags*
RFC 5187	*OSPFv3 Graceful Restart*
RFC 5213	*Proxy Mobile IPv6*
RFC 5308	*Routing IPv6 with IS-IS*
RFC 5340	*OSPF for IPv6*
RFC 5460	*DHCPv6 Bulk Leasequery*
RFC 5643	*Management Information Base for OSPFv3*
RFC 5838	*Support of Address Families in OSPFv3*
RFC 5844	*IPv4 Support for Proxy Mobile IPv6*
RFC 5845	*Generic Routing Encapsulation (GRE) Key Option for Proxy Mobile IPv6*
RFC 5846	*Binding Revocation for IPv6 Mobility*
RFC 5881	*Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)*

RFCs	Title
RFC 5905	*Network Time Protocol Version 4: Protocol and Algorithms Specification*
RFC 5969	*IPv6 Rapid Deployment on IPv4 Infrastructures (6RD) -- Protocol Specification*
RFC 6105	*IPv6 Router Advertisement Guard*
RFC 6620	*FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses*