Cisco Nexus 7000 Series NX-OS IP SLAs Configuration Guide

First Published: 2016-12-21

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

Preface

This preface describes the audience, organization, and conventions of the Book Title. It also provides information on how to obtain related documentation.

This chapter includes the following topics:

Preface

This preface describes the audience, organization, and conventions of the Book Title. It also provides information on how to obtain related documentation.

This chapter includes the following topics:

Audience

This publication is for experienced network administrators who configure and maintain Cisco NX-OS on Cisco Nexus 7000 Series Platform switches.

Document Conventions

Note

- As part of our constant endeavor to remodel our documents to meet our customers' requirements, we have modified the manner in which we document configuration tasks. As a result of this, you may find a deviation in the style used to describe these tasks, with the newly included sections of the document following the new format.
- The Guidelines and Limitations section contains general guidelines and limitations that are applicable to all the features, and the feature-specific guidelines and limitations that are applicable only to the corresponding feature.

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Related Documentation

Documentation for Cisco Nexus 7000 Series Switches is available at:

- Configuration Guides

  http://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/products-installation-and-configuration-guides-list.html

- Command Reference Guides

  http://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/products-command-reference-list.html

- Release Notes

  http://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/products-release-notes-list.html

- Install and Upgrade Guides

  http://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/products-installation-guides-list.html

- Licensing Guide

  http://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/products-licensing-information-listing.html

Documentation for Cisco Nexus 7000 Series Switches and Cisco Nexus 2000 Series Fabric Extenders is available at the following URL:

http://www.cisco.com/c/en/us/support/switches/nexus-2000-series-fabric-extenders/products-installation-and-configuration-guides-list.html

Documentation Feedback

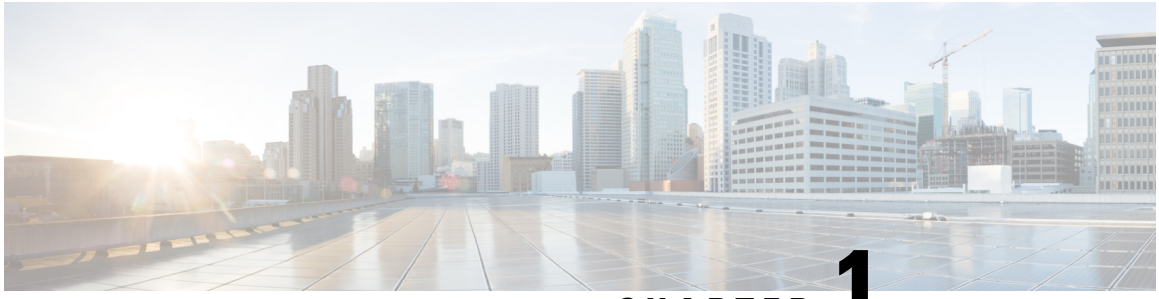
To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus7k-docfeedback@cisco.com. We appreciate your feedback.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

New and Changed Information

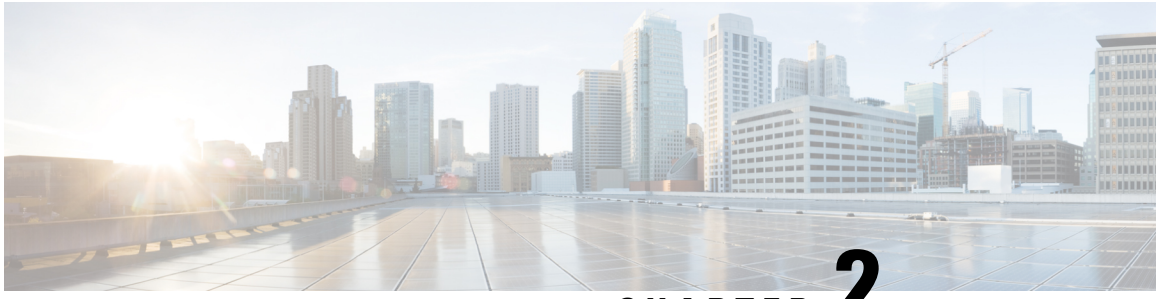
New and Changed Information

The table below summarizes the new and changed features for this document and shows the releases in which each feature is supported. Your software release might not support all the features in this document. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release.

Table 1: New and Changed Information

Feature Name	Description	Changed in Release
Configuring IP Service Level Agreements' (SLAs) Internet Control Message Protocol (ICMP) Echo Operations	Added support for operability in IPv6 networks.	8.0(1)

New and Changed Information



CHAPTER 2

Overview of IP Service Level Agreements

This chapter provides an overview of Cisco NX-OS IP Service Level Agreements (SLAs), and includes the following sections:

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Information About Cisco NX-OS IP SLAs

Many companies conduct most of their business online and any loss of service can affect the profitability of the company. Internet service providers (ISPs) and even internal IT departments now offer a defined level of service, a service level agreement (SLA), to provide their customers with a degree of predictability.

The latest performance requirements for business-critical applications, voice over IP (VoIP) networks, audio and visual conferencing, Multiprotocol Label Switching (MPLS), and Virtual Private Networks (VPNs) are creating internal pressures on converged IP networks to become optimized for performance levels. Network administrators are increasingly required to support service level agreements that support application solutions. IP Service Level Agreements (SLAs) allow you to manage IP service levels for IP applications and services.

The Cisco NX-OS IP SLAs use active traffic monitoring--the generation of traffic in a continuous, reliable, and predictable manner--for measuring network performance. Cisco NX-OS IP SLAs send data across the network to measure performance between multiple network locations or across multiple network paths. It simulates network data and IP services and collects network performance information in real time. The

information collected includes data about the response time, one-way latency, jitter (interpacket delay variance), packet loss, voice quality scoring, network resource availability, application performance, and server response time. Cisco NX-OS IP SLAs performs active monitoring by generating and analyzing traffic to measure performance either between Cisco NX-OS devices or from a Cisco NX-OS device to a remote IP device such as a network application server. Measurement statistics provided by the various Cisco NX-OS IP SLAs operations can be used for troubleshooting, problem analysis, and designing network topologies.

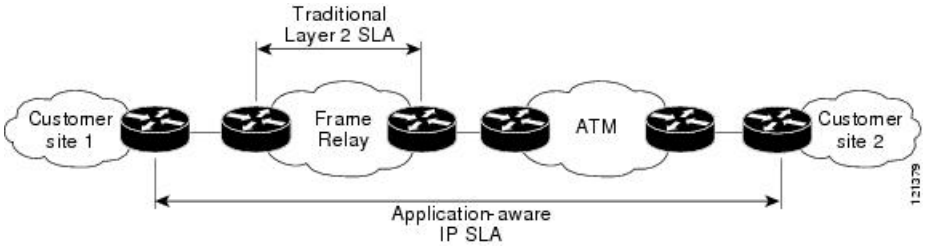
Note IPSLA do not support rollback. The rollback is related to IPSLA configuration via CLI.

Cisco NX-OS IP SLAs provides the following improvements over a traditional service level agreement:

- End-to-end measurements—The ability to measure performance from one end of the network to the other allows a broader reach and more accurate representation of the end-user experience.
- Sophistication—Statistics such as delay, jitter, packet sequence, Layer 3 connectivity, and path and download time that are broken down into bidirectional and round-trip numbers provide more data than just the bandwidth of a Layer 2 link.
- Ease of deployment—Leveraging the existing Cisco devices in a large network makes Cisco NX-OS IP SLAs easier and cheaper to implement than the physical probes often required with traditional service level agreements.
- Application-aware monitoring—Cisco NX-OS IP SLAs can simulate and measure performance statistics generated by applications running over Layer 3 through Layer 7. Traditional service level agreements can only measure Layer 2 performance.
- Pervasiveness—Cisco NX-OS IP SLAs support exists in Cisco networking devices that range from low-end to high-end switches. This wide range of deployment gives Cisco NX-OS IP SLAs more flexibility over traditional service level agreements.

The following figure shows how Cisco NX-OS IP SLAs have taken the traditional concept of Layer 2 service level agreements and applied a broader scope to support end-to-end performance measurement, including support of applications.

Figure 1: Scope of Traditional Service Level Agreement Versus Cisco NX-OS IP SLAs



Using Cisco NX-OS IP SLAs, you can measure, provide, and verify service level agreements. You can also analyze and troubleshoot network performance for IP services and applications. Depending on the specific Cisco NX-OS IP SLAs operation, statistics of delay, packet loss, jitter, packet sequence, connectivity, path, server response time, and download time can be monitored within the Cisco device and stored in both CLI and SNMP MIBs. The packets have configurable IP and application layer options such as a source and destination IP address, User Datagram Protocol (UDP)/TCP port numbers, a type of service (ToS) byte (including Differentiated Services Code Point [DSCP] and IP prefix bits), a Virtual Private Network (VPN) routing/forwarding instance (VRF), and a URL web address.

Because Cisco NX-OS IP SLAs are accessible using SNMP, it also can be used by performance monitoring applications such as CiscoWorks Internetwork Performance Monitor (IPM) and other third-party, Cisco partner performance management products.

SNMP notifications based on the data gathered by a Cisco NX-OS IP SLAs operation allow the switch to receive alerts when performance drops below a specified level and when problems are corrected. Cisco NX-OS IP SLAs use the Cisco RTTMON MIB for interaction between external Network Management System (NMS) applications and the Cisco NX-OS IP SLAs operations running on the Cisco devices. For a complete description of the object variables referenced by the Cisco NX-OS IP SLAs feature, see the text of the CISCO-RTTMON-MIB.my file, available from the Cisco MIB website.

Guidelines and Restrictions for Implementing IP Service Level Agreements

- IPv6 is available only from Cisco NX-OS Release 8.0 onwards.
- The maximum number of IP SLA-configurable operations that is supported by Cisco NX-OS software is 500.
- The current validated scale numbers for scheduling operations are as follows:
  - The number of UDP echo operations is 400 operations with default frequency.
  - The number of UDP jitter operations is 500 operations with default frequency.
  - The number of ICMP IPv4 or IPv6 echo operations is 500 operations with default frequency.
  - The number of TCP connect operations is 100 operations with default frequency.
  - We do not recommend scheduling more than 10 operations per second at the same start time because this might affect the performance of the network. We recommend the use of group scheduling configuration.

Note Setting the frequency to less than 60 seconds will increase the number of packets that will be sent. But this might negatively impact the performance of IP SLA operations when the scheduled operations have the same start time.

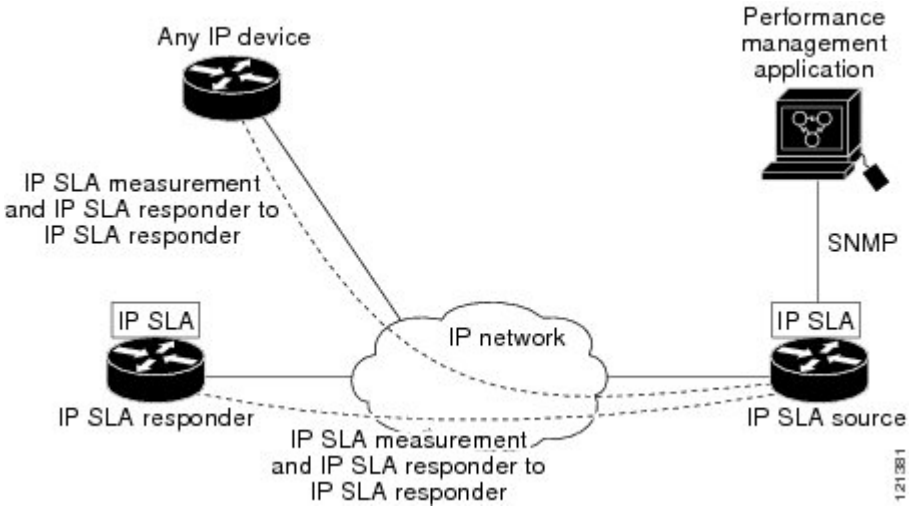
- IP SLA is not HA capable.
- Consider the following guidelines before configuring the **frequency**, **timeout**, and **threshold** commands:
  - For the UDP and ICMP jitter operations, we recommend the following guidelines:
    - frequency > timeout + 2 seconds + num_packets * packet_interval
    - timeout > rtt_threshold
    - num_packet > loss_threshold
  - For all other IP SLAs operations, we recommend the frequency > timeout > rtt_threshold guideline.

Network Performance Measurement Using Cisco NX-OS IP SLAs

Using Cisco NX-OS IP SLAs, you can monitor the performance between any area in the network: core, distribution, and edge. Monitoring can be done anytime, anywhere, without deploying a physical probe.

Cisco NX-OS IP SLAs use generated traffic to measure network performance between two networking devices such as switches. The following figure shows how Cisco NX-OS IP SLAs start when the Cisco NX-OS IP SLAs device sends a generated packet to the destination device. After the destination device receives the packet, and depending on the type of Cisco NX-OS IP SLAs operation, the device responds with time-stamp information for the source to make the calculation on performance metrics. A Cisco NX-OS IP SLAs operation performs a network measurement from the source device to a destination in the network using a specific protocol such as UDP.

Figure 2: Cisco NX-OS IP SLAs Operations



To implement a Cisco NX-OS IP SLAs network performance measurement, you must perform these tasks:

1. Enable the Cisco NX-OS IP SLAs Responder, if appropriate.
2. Configure the required Cisco NX-OS IP SLAs operation type.
3. Configure any options available for the specified Cisco NX-OS IP SLAs operation type.
4. Configure threshold conditions, if required.
5. Schedule the operation to run and then let the operation run for a period of time to gather statistics.
6. Display and interpret the results of the operation using Cisco NX-OS CLI or an network management system with SNMP.

Cisco NX-OS IP SLA Operation Types

The various types of Cisco NX-OS IP SLA operations include the following:

- Domain Name System (DNS)
- FabricPath echo
- ICMP echo (IPv4)
- ICMP echo (IPv6)
- UDP jitter
- UDP echo
- Transmission Control Protocol (TCP) connect

Cisco NX-OS IP SLA Responder and IP SLA Control Protocol

The responder is a component that is embedded in the destination Cisco routing device that allows the system to anticipate and respond to Cisco NX-OS IP SLAs request packets. The IP SLAs Responder provides accurate measurements without the need for dedicated probes and additional statistics that are not available via standard ICMP-based measurements. The Cisco NX-OS IP SLAs Control Protocol is used by the IP SLAs Responder to provide a mechanism through which the responder can be notified on which port it should listen and respond. Only a Cisco NX-OS device can be a source for a destination responder.

The IP SLAs Responder listens on a specific port for control protocol messages sent by a Cisco NX-OS IP SLAs operation. Upon receipt of the control message, the responder enables the specified UDP or TCP port for the specified duration. During this time, the responder accepts the requests and responds to them. The responder disables the port after it responds to the Cisco NX-OS IP SLAs packet or when the specified time expires.

Enabling the IP SLAs Responder on the destination device is not required for all IP SLAs operations. For example, if services that are already provided by the destination switch (such as Telnet or HTTP) are chosen, the IP SLAs Responder does not need to be enabled. For non-Cisco devices, the IP SLAs Responder cannot be configured and Cisco NX-OS IP SLAs can send operational packets only to services native to those devices.

Cisco NX-OS IP SLA Operation Scheduling

After a Cisco NX-OS IP SLAs operation has been configured, you must schedule the operation to begin capturing statistics and collecting error information. When scheduling, an operation can start immediately or start at a certain month, day, and hour. There is a pending option to set the operation to start at a later time. The pending option is also an internal state of the operation visible through SNMP. The pending state is also used when an operation is a reaction (threshold) operation waiting to be triggered. You can schedule a single Cisco NX-OS IP SLAs operation or a group of operations at one time.

Multioperations scheduling allows you to schedule multiple Cisco NX-OS IP SLAs operations using a single command through the Cisco NX-OS CLI or the CISCO RTTMON-MIB. This feature allows you to control the amount of IP SLAs monitoring traffic by scheduling the operations to run at evenly distributed times. This distribution of IP SLAs operations allows you to minimize the CPU utilization and enhance the scalability of the network.

For more details about the IP SLAs multioperations scheduling functionality, see the IP SLAs Multioperation Scheduler section.

Cisco NX-OS IP SLA Operation Threshold Monitoring

To support successful service level agreement monitoring or to proactively measure network performance, threshold functionality is essential. Consistent reliable measurements immediately identify issues and can save troubleshooting time. To roll out a service level agreement, you must have mechanisms that notify you immediately of any possible violations. Cisco NX-OS IP SLAs can send SNMP traps that are triggered by events such as the following:

- Connection loss
- Timeout
- Round-trip time threshold
- Average jitter threshold
- One-way packet loss
- One-way jitter
- One-way mean opinion score (MOS)
- One-way latency

Alternately, a Cisco NX-OS IP SLAs threshold violation can trigger another Cisco NX-OS IP SLAs operation for further analysis.

For more details on using thresholds with Cisco NX-OS IP SLAs operations, see the Proactive Threshold Monitoring for IP SLAs Operations section.

MPLS VPN Awareness

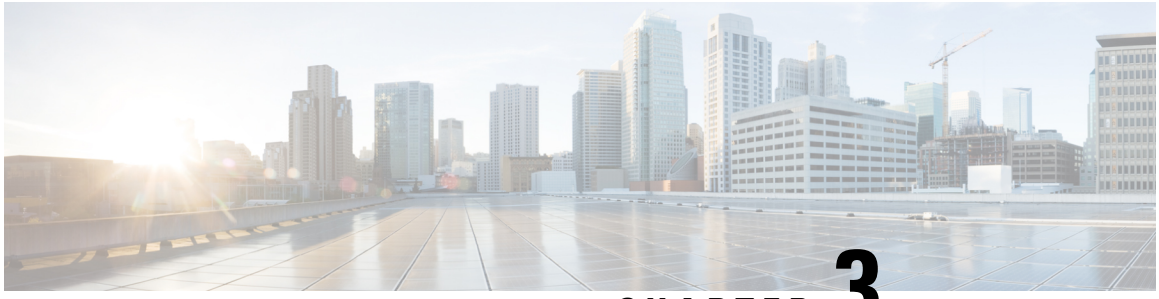
The Cisco NX-OS IP SLAs MPLS VPN Awareness feature allows you to monitor IP service levels within Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). Using IP SLAs within MPLS VPNs allows service providers to plan, provision, and manage IP VPN services according to the service level agreement for a customer. IP SLAs operations can be configured for a specific VPN by specifying a VPN routing and forwarding (VRF) name.

History Statistics

Cisco NX-OS IP SLAs maintain the following three types of history statistics:

- Aggregated statistics--By default, IP SLAs maintain two hours of aggregated statistics for each operation. The value from each operation cycle is aggregated with the previously available data within a given hour. The Enhanced History feature in IP SLAs allows for the aggregation interval to be shorter than an hour.
- Operation snapshot history--IP SLAs maintain a snapshot of data for each operation instance that matches a configurable filter, such as all, over threshold, or failures. The entire set of data is available and no aggregation takes place.
- Distribution statistics--IP SLAs maintain a frequency distribution over configurable intervals. Each time IP SLAs starts an operation, a new history bucket is created until the number of history buckets that matches the specified size or the lifetime of the operation expires. By default, the history for an IP SLAs

operation is not collected. If history is collected, each bucket contains one or more history entries from the operation. History buckets do not wrap.



CHAPTER 3

Configuring IP SLA UDP Jitter Operations

This chapter describes how to configure an IP Service Level Agreements (SLAs) UDP jitter operation 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 networks. This chapter also demonstrates how the data gathered using the UDP jitter operation can be displayed and analyzed using the Cisco software commands.

This chapter includes the following sections:

Information About the IP SLA UDP Jitter Operation

The IP SLAs UDP jitter operation can diagnose network suitability for real-time traffic applications such as voice over IP (VoIP), video over IP, or real-time conferencing.

Jitter means inter-packet delay variance. When multiple packets are sent consecutively from source to destination, for example, 10 ms apart, and if the network is behaving ideally, the destination should be receiving them 10 ms apart. But if there are delays in the network (such as queuing, arriving through alternate routes, and so on), the arrival delay between packets might be greater than or less than 10 ms. Using this example, a positive jitter value indicates that the packets arrived greater than 10 ms apart. If the packets arrive 12 ms apart, then positive jitter is 2 ms; if the packets arrive 8 ms apart, then negative jitter is 2 ms. For delay-sensitive networks such as VoIP, positive jitter values are undesirable, and a jitter value of 0 is ideal.

However, the IP SLAs UDP jitter operation does more than just monitor jitter. As the UDP jitter operation includes the data returned by the IP SLAs UDP operation, the UDP jitter operation can be used as a multipurpose data gathering operation. The packets that IP SLAs generate carry packet sending sequence, receiving sequence information, and sending and receiving time stamps from the source and the operational target. UDP jitter operations can measure the following:

- Per-direction jitter (source to destination and destination to source)
- Per-direction packet-loss
- Per-direction delay (one-way delay)

- Round-trip delay (average round-trip time)

As the paths for the sending and receiving of data may be different (asymmetric), the per-direction data allow you to more readily identify where congestion or other problems are occurring in the network.

The UDP jitter operation functions by generating synthetic (simulated) UDP traffic. The UDP jitter operation sends N UDP packets, each of size S, sent T milliseconds apart, from a source switch to a target switch, at a given frequency of F. By default, ten packet-frames (N), each with a payload size of 10 bytes (S), are generated every 10 ms (T), and the operation is repeated every 60 seconds (F). Each of these parameters are user-configurable as shown in the following table.

Table 2: UDP Jitter Operation Parameters

UDP Jitter Operation Parameter	Default	Command
Number of packets (N)	10 packets	**udp-jitter** command, **numpackets** option
Payload size per packet (S)	32 bytes	**request-data-size** command
Time between packets, in milliseconds (T)	20 ms	**udp-jitter** command, interval option
Elapsed time before the operation repeats, in seconds (F)	60 seconds	**frequency (IP SLA)** command

Prerequisites for Configuring IP SLA UDP Jitter Operations

The prerequisites for configuring IP SLAs UDP jitter operations are as follows:

- Time synchronization, such as that provided by NTP, is required between the source and the target device in order to provide accurate one-way delay (latency) measurements.

  Time synchronization is not required for the one-way jitter and packet loss measurements. If the time is not synchronized between the source and target devices, one-way jitter and packet loss data are returned, but values of “0” are returned for the one-way delay measurements provided by the UDP jitter operation.
- Before configuring any IP SLAs application, you can use the **show ip sla application** command to verify that the operation type is supported on your software image.

Guidelines and Limitations for UDP Jitter Operations

Configuring CoPP for IP SLA Packets

When using IP SLA operations on a large scale, a specific CoPP configuration to allow the IP SLA packets to pass through might be needed. Since IP SLA uses user defined UDP ports, there is no way to allow all IP

SLA packets to the control plane. However, you can specify each destination/source port that IP SLA can use.

For more information about the verified scalability of the number of IP SLA probes, see the *Cisco Nexus 7000 Series NX-OS Verified Scalability Guide*.

The following shows an example of a CoPP configuration that allows IP SLA packets to pass through. It assumes destination ports and source ports in the range of 6500-7000.

```
ip access-list copp-system-sla-allow
  10 remark ### ALLOW SLA control packets from 1.1.1.0/24
  20 permit udp 1.1.1.0/24 any eq 1967
  30 remark ### ALLOW SLA data packets from 1.1.1.0/24 using ports 6500-7000
  40 permit udp 1.1.1.0/24 any range 6500 7000
  statistics per-entry
ip access-list copp-system-sla-deny
  10 remark ### this is a catch-all to match any other traffic
  20 permit ip any any
  statistics  per-entry
class-map type control-plane match-any copp-system-class-management-allow
  match access-group name copp-system-sla-allow
class-map type control-plane match-any copp-system-class-management-deny
  match access-group name copp-system-sla-deny
policy-map type control-plane copp-system-policy
    class copp-system-class-management-allow
    set cos 7
    police cir 4500 kbps bc 250 ms conform transmit violate drop
    class copp-system-class-management-deny
    police cir 4500 kbps bc 250 ms conform drop violate drop
control-plane
  service-policy input copp-system-policy
```

Configuring and Scheduling a UDP Jitter Operation on the Source Device

This section describes how to configure and schedule a UDP jitter operation.

Configuring the IP SLA Responder on the Destination Device

This section describes how to configure the responder on the destination device.

Note A responder should not configure a permanent port for the same sender. If the responder configures the permanent port for the same sender, even if the packets are successfully sent (no timeout or packet loss issues), the jitter values are zero.

Procedure

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

	Command or Action	Purpose
	`switch> enable`	
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: • **ip sla responder** *Example:* `switch(config)# ip sla responder` • **ip sla responder udp-echo ipaddress** *ip-address* **port** *port* *Example:* `switch(config)# ip sla responder udp-echo ipaddress 172.29.139.132 port 5000`	- • (Optional) Temporarily enables the responder functionality on a Cisco device in response to control messages from a source. • (Optional) Required only if protocol control is disabled on a source. Permanently enables the responder functionality on the specified IP addresses and port. Control is enabled by default.
Step 4	**exit** **Example:** `switch(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring and Scheduling a Basic UDP Jitter Operation on the Source Device

This section describes how to configure and schedule a basic UDP jitter operation on the source device.

Tip

- If the IP SLAs operation is not running and generating statistics, add the **verify-data** command to the configuration of the operation (while configuring in IP SLA configuration mode) to enable data verification. When enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla sender trace** and **debug ip sla sender error** commands to help troubleshoot issues with an IP SLAs operation.

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `switch(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**sourceport** *port-number*] [**control** { **enable**| **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] **Example:** `switch(config-ip-sla)# udp-jitter 172.29.139.134 5000`	Configures the IP SLAs operation as a UDP jitter operation and enters UDP jitter configuration submode. Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target switches.
Step 5	**frequency** *seconds* **Example:** `switch(config-ip-sla-jitter)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**exit** **Example:** `switch(config-ip-sla-jitter)# exit`	Exits UDP jitter configuration submode and returns to global configuration mode.
Step 7	**ip sla schedule** *operation-number* [**life** {*forever*| *seconds*}] [**start-time** {*hh:mm*[*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}] [**ageout** *seconds*] [**recurring**] **Example:** `switch(config)# ip sla schedule 5 start-time now life forever`	Configures the scheduling parameters for an individual IP SLAs operation.
Step 8	**exit** **Example:** `switch(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 9	**show ip sla configuration** [*operation-number*] **Example:** `switch# show ip sla configuration 10`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps or for starting another operation, see the Configuring Proactive Threshold Monitoring, on page 81 section.

To display statistics of an IP SLA operation over the last one hour and interpret the results, use the **show ip sla statistics** command. Checking the output for fields that correspond to criteria in your service level agreement helps you to determine whether the service metrics are acceptable. To display the aggregated IP SLA history, use the **show ip sla statistics aggregated** command.

Configuring and Scheduling a UDP Jitter Operation with Additional Characteristics

This section describes how to configure and schedule a UDP jitter operation with additional characteristics.

- The IP SLAs UDP jitter operation does not support the IP SLAs History feature (statistics history buckets) because of the large data volume involved with UDP jitter operations, which means that the following commands are not supported for UDP jitter operations: **history buckets-kept**, **history filter**, **historylives-kept**, **samples-of-history-kept**, and **show ip sla history**.
- The MIB used by IP SLAs (CISCO-RTTMON-MIB) limits the hours-of-statistics kept for the UDP jitter operation to two hours. Configuring a larger value using the **history hours-of-statistics***hours* global configuration change does not increase the value beyond two hours. However, the Data Collection MIB can be used to collect historical data for the operation. For information, see the CISCO-DATA-COLLECTION-MIB at http://www.cisco.com/go/mibs.

Tip

- If the IP SLAs operation is not running and generating statistics, add the **verify-data** command to the configuration of the operation (while configuring in IP SLA configuration mode) to enable data verification. When enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla sender trace** and **debug ip sla sender error** commands to help troubleshoot issues with an IP SLAs operation.

Before you begin

Before configuring a UDP jitter operation on the source device, the IP SLAs Responder must be enabled on the target device (the operational target). The IP SLAs Responder is available only on Cisco NX-OS software based devices. To enable the responder, perform the task in the “Configuring the IP SLAs Responder on the Destination Device” section.

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Switch(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets***number-of-packets*] [**interval** *interpacket-interval*] **Example:** `Switch(config-ip-sla)# udp-jitter 172.29.139.134 5000`	Configures the IP SLAs operation as a UDP jitter operation and enters UDP jitter configuration submode. • Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target switches.
Step 5	**history distributions-of-statistics-kept** *size* **Example:** `Switch(config-ip-sla-jitter)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 6	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Switch(config-ip-sla-jitter)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 7	**frequency** *seconds* **Example:** `Switch(config-ip-sla-jitter)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 8	**history hours-of-statistics-kept** *hours* **Example:** `Switch(config-ip-sla-jitter)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.

	Command or Action	Purpose
Step 9	**owner** *owner-id* **Example:** `Switch(config-ip-sla-jitter)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 10	**request-data-size** *bytes* **Example:** `Switch(config-ip-sla-jitter)# request-data-size 64`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.
Step 11	**history statistics-distribution-interval** *milliseconds* **Example:** `Switch(config-ip-sla-jitter)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 12	**tag** *text* **Example:** `Switch(config-ip-sla-jitter)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 13	**threshold** *milliseconds* **Example:** `Switch(config-ip-sla-jitter)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 14	**timeout** *milliseconds* **Example:** `Switch(config-ip-sla-jitter)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 15	**tos** *number* **Example:** `Switch(config-ip-sla-jitter)# tos 160`	(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.
Step 16	**verify-data** **Example:** `Switch(config-ip-sla-jitter)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.

	Command or Action	Purpose
Step 17	**vrf** *vrf-name* **Example:** `Switch(config-ip-sla-jitter)# vrf vpn-A`	(Optional) Allows monitoring within Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) using IP SLAs operations.
Step 18	**exit** **Example:** `Switch(config-ip-sla-jitter)# exit`	Exits UDP jitter configuration submode and returns to global configuration mode.
Step 19	**ip sla schedule** *operation-number* [**life** {**forever**| *seconds*}] [**start-time** {*hh*:*mm*[:*ss*] [*monthday* | *daymonth*] | **pending** | **now** | **after***hh*:*mm*:*ss*}] [**ageout***seconds*] [**recurring**] **Example:** `Switch(config)# ip sla schedule 5 start-time now life forever`	Configures the scheduling parameters for an individual IP SLAs operation.
Step 20	**exit** **Example:** `Switch(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 21	**show ip sla configuration** [*operation-number*] **Example:** `Switch# show ip sla configuration 10`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

Configuration Example for a UDP Jitter Operation

This example shows two operations that are configured as UDP jitter operations, with operation 2 starting five seconds after the first operation. Both operations will run indefinitely.

```
ip sla 1
 udp-jitter 20.0.10.3 65051 num-packets 20
 request-data-size 160
 tos 128
 frequency 30
ip sla schedule 1 start-time after 00:05:00
ip sla 2
 udp-jitter 20.0.10.3 65052 num-packets 20 interval 10
 request-data-size 20
 tos 64
 frequency 30
ip sla schedule 2 start-time after 00:05:05
```

On the target (destination) device:

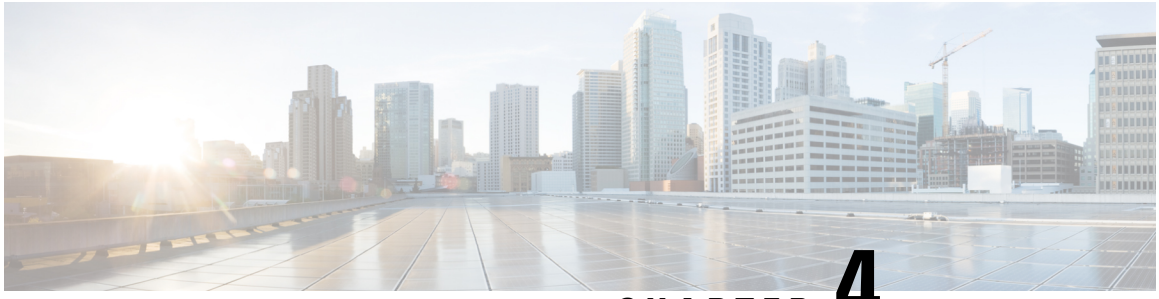
```
ip sla responder
```

Feature History for UDP Jitter

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 3: Feature History for UDP Jitter

Feature Name	Release	Feature Information
UDP Jitter	6.1(1)	This feature was introduced.



CHAPTER 4

Configuring IP SLA UDP Jitter Operations for VoIP

This chapter describes how to configure an IP Service Level Agreements (SLAs) User Datagram Protocol (UDP) jitter operation to proactively monitor Voice over IP (VoIP) quality levels in your network, allowing you to guarantee VoIP quality levels to your users in IPv4 networks. The IP SLAs VoIP UDP jitter operation accurately simulates VoIP traffic using common codecs and calculates consistent voice quality scores, such as Mean Opinion Score (MOS) and Calculated Planning and Improvement Factor (ICPIF), between Cisco devices in the network.

Note In this document, the term Voice refers to Internet telephony applications. The term Voice over IP can include the transmission of multimedia (both voice and video) over IP networks.

This chapter includes the following sections:

Guidelines and Limitations for IP SLAs UDP Jitter Operations for VoIP

- This feature uses UDP traffic to generate approximate Voice over IP scores. It does not provide support for the Real-Time Transport Protocol (RTP).
- The Calculated Planning Impairment Factor (ICPIF) and MOS values provided by this feature, while consistent within IP SLAs, are estimates only and are intended only for relative comparisons. The values may not match values that are determined using other methods.

- Predictions of customer opinion (such as those listed for the E-Model transmission rating factor R and derived Mean Opinion Scores) that are determined by any method are intended only for transmission planning and analysis purposes and should not be interpreted as reflecting actual customer opinions.

Configuring CoPP for IP SLA Packets

When using IP SLA operations on a large scale, a specific CoPP configuration to allow the IP SLA packets to pass through might be needed. Since IP SLA uses user defined UDP ports, there is no way to allow all IP SLA packets to the control plane. However, you can specify each destination/source port that IP SLA can use.

For more information about the verified scalability of the number of IP SLA probes, see the *Cisco Nexus 7000 Series NX-OS Verified Scalability Guide*.

The following shows an example of a CoPP configuration that allows IP SLA packets to pass through. It assumes destination ports and source ports in the range of 6500-7000.

```
ip access-list copp-system-sla-allow
  10 remark ### ALLOW SLA control packets from 1.1.1.0/24
  20 permit udp 1.1.1.0/24 any eq 1967
  30 remark ### ALLOW SLA data packets from 1.1.1.0/24 using ports 6500-7000
  40 permit udp 1.1.1.0/24 any range 6500 7000
  statistics per-entry
ip access-list copp-system-sla-deny
  10 remark ### this is a catch-all to match any other traffic
  20 permit ip any any
  statistics  per-entry
class-map type control-plane match-any copp-system-class-management-allow
  match access-group name copp-system-sla-allow
class-map type control-plane match-any copp-system-class-management-deny
  match access-group name copp-system-sla-deny
policy-map type control-plane copp-system-policy
    class copp-system-class-management-allow
    set cos 7
    police cir 4500 kbps bc 250 ms conform transmit violate drop
    class copp-system-class-management-deny
    police cir 4500 kbps bc 250 ms conform drop violate drop
control-plane
  service-policy input copp-system-policy
```

Calculated Planning Impairment Factor

The ICPIF originated in the 1996 version of ITU-T recommendation G.113, “Transmission impairments,” as part of the formula Icpif = Itot - A. An ICPIF refers to the “calculated planning impairment factor.” The ICPIF attempts to quantify, for comparison and planning purposes, the key impairments to voice quality that are encountered in the network.

The ICPIF is the sum of measured impairment factors (total impairments or Itot) minus a user-defined access Advantage Factor (A) that is intended to represent the user’s expectations, based on how the call was placed (for example, a mobile call versus a land-line call). In its expanded form, the full formula is expressed as follows:

$Icpif = Io + Iq + Idte + Idd + Ie - A$

where

- *Io* represents impairments caused by nonoptimal loudness rating.

- *Iq* represents impairments caused by PCM quantizing distortion.
- *Idte* represents impairments caused by talker echo.
- *Idd* represents impairments caused by one-way transmission times (one-way delay).
- *Ie* represents impairments caused by equipment effects, such as the type of codec used for the call and packet loss.
- *A* represents an access Advantage Factor (also called the user Expectation Factor) that compensates for users who might accept some degradation in quality in return for ease of access.

ICPIF values are expressed in a typical range of 5 (very low impairment) to 55 (very high impairment). ICPIF values numerically less than 20 are generally considered "adequate." While intended to be an objective measure of voice quality, the ICPIF value is also used to predict the subjective effect of combinations of impairments.

The following table, taken from G.113 (02/96), shows how sample ICPIF values are expected to correspond to subjective quality judgement.

Upper Limit for ICPIF	**Speech Communication Quality**
5	Very good
10	Good
20	Adequate
30	Limiting case
45	Exceptional limiting case
55	Customers likely to react strongly (complaints, change of network operator)

For more details on the ICPIF, see the 1996 version of the G.113 specification.

Note The latest version of the ITU-T G.113 Recommendation (2001), no longer includes the ICPIF model. Instead, G.107 states "The Impairment Factor method, used by the E-model of ITU-T G.107, is now recommended. The earlier method that used Quantization Distortion Units is no longer recommended." The full E-Model (also called the ITU-T Transmission Rating Model), expressed as $R = Ro - Is - Id - Ie + A$, provides the potential for more accurate measurements of call quality by refining the definitions of impairment factors (see the 2003 version of the G.107 for details). Though the ICPIF shares terms for impairments with the E-Model, the two models are different. The IP SLAs VoIP UDP Operation feature takes advantage of observed correspondences between the ICPIF, transmission rating factor R, and MOS values, but does not support the E-Model.

Mean Opinion Scores

The quality of transmitted speech is a subjective response of the listener. Each codec used for transmission of VoIP provides a certain level of quality. A common benchmark used to determine the quality of sound produced by specific codecs is the mean opinion score (MOS). With MOS, a wide range of listeners have

judged the quality of voice samples sent using particular codecs, on a scale of 1 (poor quality) to 5 (excellent quality). The opinion scores are averaged to provide the mean for each sample.

The following table shows MOS ratings and the corresponding description of quality for each value.

Table 4: MOS Ratings

Score	Quality	Description of Quality Impairment
5	Excellent	Imperceptible
4	Good	Just perceptible, but not annoying
3	Fair	Perceptible and slightly annoying
2	Poor	Annoying but not objectionable
1	Bad	Very annoying and objectionable

As the MOS ratings for codecs and other transmission impairments are known, an estimated MOS can be computed and displayed based on measured impairments. This estimated value is designated as MOS-CQE (Mean Opinion Score; Conversational Quality, Estimated) by the ITU in order to distinguish it from objective or subjective MOS values (see P.800.1 for details).

Voice Performance Monitoring Using IP SLAs

One of the key metrics in measuring voice and video quality over an IP network is jitter. Jitter indicates the variation in delay between arriving packets (inter-packet delay variance). Jitter affects voice quality by causing uneven gaps in the speech pattern of the person talking. Other key performance parameters for voice and video transmission over IP networks include latency (delay) and packet loss. IP SLAs allow you to simulate and measure these parameters in order to ensure your network is meeting or exceeding service-level agreements with your users.

IP SLAs provide a UDP jitter operation, which consists of UDP probe packets sent across the network from an origin device to a specific destination (called the operational target). This synthetic traffic is used to record the amount of jitter for the connection, as well as the round-trip time, per-direction packet loss, and one-way delay time (one-way latency). (Synthetic traffic indicates that the network traffic is simulated; that is, the traffic is generated by IP SLAs.) Data, in the form of collected statistics, can be displayed for multiple tests over a user-defined period of time, allowing you to see, for example, how the network performs at different times of the day or over the course of a week. The jitter probe can use the IP SLAs Responder to provide minimal latency at the receiving end.

The IP SLAs VoIP UDP jitter operation modifies the standard UDP jitter operation by adding the capability to return MOS and ICPIF scores in the data collected by the operation, in addition to the metrics already gathered by the UDP jitter operation. This VoIP-specific implementation allows you to determine the performance of your VoIP network.

Codec Simulation Within IP SLAs

The IP SLAs VoIP UDP jitter operation computes statistics by sending n UDP packets, each of size s, sent t milliseconds apart, from a given source switch to a given target switch, at a given frequency f. The target switch must be running the IP SLAs Responder in order to process the probe operations.

To generate MOS and ICPIF scores, you specify the codec type used for the connection when configuring the VoIP UDP jitter operation. Based on the type of codec that you configure for the operation, the number of packets (n), the size of each payload (s), the inter-packet time interval (t), and the operational frequency (f) are automatically configured with default values. However, you are given the option, if needed, to manually configure these parameters in the syntax of the **udp-jitter** command.

The following table shows the default parameters that are configured for the operation by codec.

Table 5: Default VoIP UDP Jitter Operation Parameters by Codec

Codec	Default Request Size (Packet Payload) (s)	Default Interval Between Packets (t)	Default Number of Packets (n)	Frequency of Probe Operations (f)
G.711 mu-Law (g711ulaw)	160 + 12 RTP bytes	20 ms	1000	Once every 1 minute
G.711 A-Law (g711alaw)	160 + 12 RTP bytes	20 ms	1000	Once every 1 minute
G.729A (g729a)	20 + 12 RTP bytes	20 ms	1000	Once every 1 minute

For example, if you configure the VoIP UDP jitter operation to use the characteristics for the g711ulaw codec, by default a probe operation is sent once a minute (**f**). Each probe operation consists of 1000 packets (**n**), each packet containing 180 bytes of synthetic data (**s**), sent 20 milliseconds apart (**t**).

IP SLAs ICPIF Value

ICPIF value computation with the Cisco NX-OS software is based primarily on the two main factors that can impair voice quality: delayed packets and lost packets. Because packet delay and packet loss can be measured by IP SLAs, the ICPIF formula, *Icpif=Io+Iq+Idte+Idd+Ie-A*, is simplified by assuming that the values of *Io*, *Iq*, and *Idte* are zero, as follows:

Total Impairment Factor (Icpif) = Delay Impairment Factor (Idd) + Equipment Impairment Factor (Ie) — Expectation/Advantage Factor (A)

The ICPIF value is computed by adding a Delay Impairment Factor, which is based on a measurement of delayed packets, and an Equipment Impairment Factor, which is based on a measurement of lost packets. From this sum of the total impairments measured in the network, an impairment variable (the Expectation Factor) is subtracted to yield the ICPIF.

Cisco gateways use this formula to calculate the ICPIF for received VoIP data streams.

Delay Impairment Factor

The Delay Impairment Factor (*Idd*) is a number based on two values. One value is fixed and is derived using the static values (as defined in the ITU standards) for Codec Delay, Look Ahead Delay, and Digital Signal Processing (DSP) Delay. The second value is a variable and is based on the measured one-way delay (round-trip time measurement divided by 2). The one-way delay value is mapped to a number using a mapping table that is based on a G.107 (2002 version) analytic expression.

The following table shows sample correspondences between the one-way delay measured by IP SLAs and Delay Impairment Factor values.

Table 6: Sample Correspondence of One-Way Delay to ICPIF Delay Impairment

One-Way Delay (ms)	Delay Impairment Factor
50	1
100	2
150	4
200	7

Equipment Impairment Factor

The Equipment Impairment Factor (*Ie*) is a number based on the amount of measured packet loss. The amount of measured packet loss, expressed as a percentage of total number of packets sent, corresponds with an Equipment Impairment Factor that is defined by the codec.

The following table shows sample correspondences between the packet loss measured by IP SLAs and Equipment Impairment Factor values corresponding with each other.

Table 7: Sample Correspondence of Measured Packet Loss to ICPIF Equipment Impairment

Packet Loss (as a percentage of total number of packets sent)	Equipment Impairment Value for PCM (G.711) Codecs	Equipment Impairment Value for the CS-ACELP (G.729A) Codec
2%	12	20
4%	22	30
6%	28	38
8%	32	42

Expectation Factor

The Expectation Factor, also called the Advantage Factor (*A*), represents the expectation that users might accept some degradation in quality in return for ease of access. For example, a mobile phone user in a hard-to-reach location might expect that the connection quality will not be as good as a traditional land-line connection. This variable is also called the Advantage Factor (short for Access Advantage Factor) because it attempts to balance an increased access advantage against a decline in voice quality.

The table below, adapted from ITU-T Rec. G.113, defines a set of provisional maximum values for *A* in terms of the service provided.

Table 8: Advantage Factor Recommended Maximum Values

Communication Service	Advantage/Expectation Factor: Maximum value of A
Conventional wire-line (land-line)	0
Mobility (cellular connections) within a building	5
Mobility within a geographical area or moving in a vehicle	10
Access to hard-to-reach location; (for example, via multi-hop satellite connections)	20

These values are only suggestions. To be meaningful, you should use the factor *A* and its selected value in a specific application consistently in any planning model that you adopt. However, the values in the table should be considered as the absolute upper limits for *A*.

The default Advantage Factor for IP SLAs VoIP UDP jitter operations is always zero.

IP SLAs MOS Value

IP SLAs use an observed correspondence between ICPIF and MOS values to estimate an MOS value.

Note The abbreviation MOS represents MOSCQE (Mean Opinion Score; Conversational Quality, Estimated).

The E model, as defined in G.107 (03/2003), predicts the subjective quality that is experienced by an average listener by combining the impairment caused by transmission parameters (such as loss and delay) into a single rating, the transmission rating factor R (the R Factor). This rating, expressed in a scale of 0 (worst) to 100 (best), can be used to predict subjective user reactions, such as the MOS. Specifically, the MOS can be obtained from the R Factor with a converting formula. Conversely, a modified inverted form can be used to calculate R Factors from MOS values.

There is also a relationship between the ICPIF value and the R Factor. IP SLAs takes advantage of this correspondence by deriving the approximate MOS score from an estimated R Factor, which, in turn, is derived from the ICPIF score.

The following table shows the MOS values that are generated for corresponding ICPIF values.

Table 9: Correspondence of ICPIF Values to MOS Values

ICPIF Range	MOS	Quality Category
0 - 3	5	Best
4 - 13	4	High
14 - 23	3	Medium
24 - 33	2	Low

ICPIF Range	MOS	Quality Category
34 - 43	1	Poor

IP SLAs always express the estimated MOS value as a number in the range of 1 to 5, with 5 being the best quality. A MOS value of 0 (zero) indicates that MOS data could not be generated for the operation.

Configuring and Scheduling an IP SLAs VoIP UDP Jitter Operation

Note

- Currently, IP SLAs supports only the following speech codecs (compression methods):
  - G.711 A Law (g711alaw: 64 kbps PCM compression method)
  - G.711 mu Law (g711ulaw: 64 kbps PCM compression method)
  - G.729A (g729a: 8 kbps CS-ACELP compression method)
- The following commands, available in UDP jitter configuration mode, are not valid for UDP jitter (codec) operations:
  - **history distributions-of-statistics-kept**
  - **history statistics-distribution-interval**
  - **request-data-size**
- Specifying the codec-type will configure the appropriate default values for the **codec-interval**, **codec-size**, and **codec-numpacket** options. You should not specify values for the interval, size, and number of packet options unless you have a specific reason to override the defaults (for example, approximating a different codec).
- The **show ip sla configuration** command will list the values for the number of statistic distribution buckets kept and statistic distribution interval (microseconds), but these values do not apply to jitter (codec) operations.

Tip

- If the IP SLAs operation is not running and generating statistics, add the **verify-data** command to the configuration of the operation (while configuring in IP SLA configuration mode) to enable data verification. When enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

Procedure

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

	Command or Action	Purpose
	`switch> enable`	
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `switch(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* **codec** *codec-type* [**codec-numpackets** *number-of-packets*] [**codec-size** *number-of-bytes*] [**codec-interval** *milliseconds*] [**advantage-factor** *value*] [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] **Example:** `switch(config-ip-sla)# udp-jitter 209.165.200.225 16384 codec g711alaw advantage-factor 10`	Configures the operation as a jitter (codec) operation that will generate VoIP scores in addition to latency, jitter, and packet loss statistics.
Step 5	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `switch(config-ip-sla-jitter)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 6	**frequency** *seconds* **Example:** `switch(config-ip-sla-jitter)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 7	**history hours-of-statistics-kept** *hours* **Example:** `switch(config-ip-sla-jitter)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.

	Command or Action	Purpose
Step 8	**owner** *owner-id* **Example:** `switch(config-ip-sla-jitter)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 9	**tag** *text* **Example:** `switch(config-ip-sla-jitter)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 10	**threshold** *microseconds* **Example:** `switch(config-ip-sla-jitter)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 11	**timeout** *microseconds* **Example:** `switch(config-ip-sla-jitter)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 12	**tos** *number* **Example:** `switch(config-ip-sla-jitter)# tos 160`	(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.
Step 13	**verify-data** **Example:** `switch(config-ip-sla-jitter)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.
Step 14	**vrf** *vrf-name* **Example:** `switch(config-ip-sla-jitter)# vrf vpn-A`	(Optional) Allows monitoring within Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) using IP SLAs operations.
Step 15	**exit** **Example:** `switch(config-ip-sla-jitter)# exit`	Exits UDP jitter configuration submode and returns to global configuration mode.

	Command or Action	Purpose
Step 16	**ip sla schedule** *operation-number* [**life** {**forever**| *seconds*}] [**start-time** {*hh*:*mm*[:*ss*] [*monthday* | *daymonth*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**] **Example:** `switch(config)# ip sla schedule 5 start-time now life forever`	Configures the scheduling parameters for an individual IP SLAs operation.
Step 17	**exit** **Example:** `switch(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 18	**show ip sla configuration** [*operation-number*] **Example:** `switch# show ip sla configuration 10`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps or for starting another operation, see the Configuring Proactive Threshold Monitoring, on page 81 section.

To display statistics of an IP SLA operation over the last one hour and interpret the results, use the **show ip sla statistics** command. Checking the output for fields that correspond to criteria in your service level agreement helps you to determine whether the service metrics are acceptable. To display the aggregated IP SLA history, use the **show ip sla statistics aggregated** command.

Configuration Examples for IP SLAs VoIP UDP Operation

This example assumes that the IP SLAs Responder is enabled on the device at 101.101.101.1:

```
switch# conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip sla 10
switch(config-ip-sla)# udp-jitter 101.101.101.1 16384 codec g711alaw advantage-factor 2
switch(config-ip-sla-jitter)# owner admin_bofh
switch(config-ip-sla-jitter)# precision microseconds
switch(config-ip-sla-jitter)# exit
switch(config)# ip sla schedule 10 start-time now
switch(config)# exit
switch# show ip sla config 10
IP SLAs Infrastructure Engine-III
Entry number: 10
Owner: admin_bofh
Tag:
Operation timeout (milliseconds): 5000
Type of operation to perform: udp-jitter
Target address/Source address: 101.101.101.1/0.0.0.0
Target port/Source port: 16384/0
```

```
Type Of Service parameter: 0x0
Codec type: g711alaw
Codec Number Of Packets: 1000
Codec Packet Size: 172
Codec Interval (milliseconds): 20
Advantage Factor: 2
Verify data: No
Operation Stats Precision : microseconds
Operation Packet Priority : normal
NTP Sync Tolerance : 0 percent
Vrf Name: default
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (microseconds): 20

switch#

switch# show running-config | begin "ip sla 10"
ip sla 10
  udp-jitter 101.101.101.1 16384 codec g711alaw advantage-factor 2
    precision microseconds
    owner admin_bofh
ip sla schedule 10 start-time now
no logging console
.
.
.
switch# show ip sla configuration 10
Entry number: 10
Owner: admin_bofh
Tag:
Type of operation to perform: jitter
Target address: 101.101.101.1
Source address: 0.0.0.0
Target port: 16384
Source port: 0
Operation timeout (milliseconds): 5000
Codec Type: g711alaw
Codec Number Of Packets: 1000
Codec Packet Size: 172
Codec Interval (milliseconds): 20
Advantage Factor: 2
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Control Packets: enabled
Operation frequency (seconds): 60
Next Scheduled Start Time: Start Time already passed
Life (seconds): 3600
Entry Ageout (seconds): never
Status of entry (SNMP RowStatus): Active
Connection loss reaction enabled: No
```

```
Timeout reaction enabled: No
Verify error enabled: No
Threshold reaction type: Never
Threshold (milliseconds): 5000
Threshold Falling (milliseconds): 3000
Threshold Count: 5
Threshold Count2: 5
Reaction Type: None
Number of statistic hours kept: 2
Number of statistic distribution buckets kept: 1
Statistic distribution interval (microseconds): 20
Enhanced History:
```

When a codec type is configured for a jitter operation, the standard jitter "Request size (ARR data portion)," "Number of packets," and "Interval (microseconds)" parameters do not be appear in the **show ip sla** configuration command output. Instead, values for "Codec Packet Size," "Codec Number of Packets," and "Codec Interval (microseconds)" appear.

Configuration Examples for IP SLAs VoIP UDP Operation Statistics Output

This example shows how to display voice scores (ICPIF and MOS values) for the jitter (codec) operation:

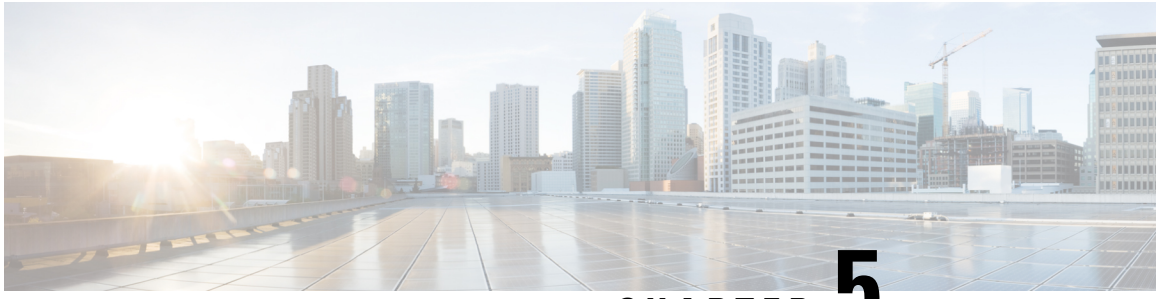
```
switch# show ip sla st
IPSLAs Latest Operation Statistics
IPSLA operation id: 1
Type of operation: udp-jitter
        Latest RTT: 11999 microseconds
Latest operation start time: 02:39:33 UTC Sat May 05 2012
Latest operation return code: OK
Latest operation NTP sync state: NO_SYNC
RTT Values:
        Number Of RTT: 10
RTT Min/Avg/Max: 9000/11999/17000 microseconds
Latency one-way time:
        Number of Latency one-way Samples: 0
        Source to Destination Latency one way Min/Avg/Max: 0/0/0 microseconds
        Destination to Source Latency one way Min/Avg/Max: 0/0/0 microseconds
Jitter Time:
        Number of SD Jitter Samples: 9
        Number of DS Jitter Samples: 9
        Source to Destination Jitter Min/Avg/Max: 0/223/2001 microseconds
        Destination to Source Jitter Min/Avg/Max: 0/2001/6001 microseconds
Packet Loss Values:
        Loss Source to Destination: 0
        Source to Destination Loss Periods Number: 0
```

Feature History for UDP Jitter

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 10: Feature History for UDP Jitter

Feature Name	Release	Feature Information
UDP Jitter	6.1(1)	This feature was introduced.



CHAPTER 5

Configuring IP SLAs UDP Echo Operations

This chapter describes how to configure an IP Service Level Agreements (SLAs) User Datagram Protocol (UDP) Echo operation to monitor end-to-end response time between a Cisco switch and devices using IPv4. UDP echo accuracy is enhanced by using the IP SLAs Responder at the destination Cisco switch. This module also demonstrates how the results of the UDP echo operation can be displayed and analyzed to determine how a UDP application is performing.

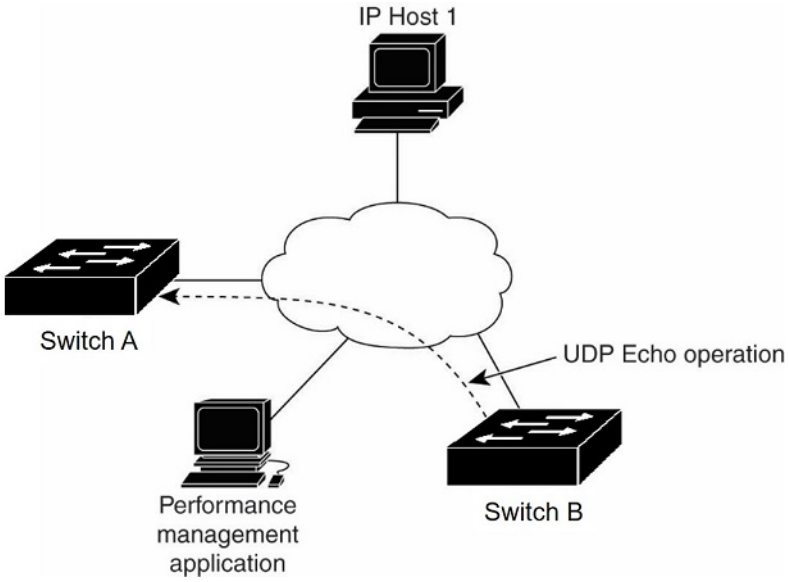
This chapter includes the following sections:

UDP Echo Operation

The UDP echo operation measures end-to-end response time between a Cisco switch and devices using IP. UDP is a transport layer (Layer 4) Internet protocol that is used for many IP services. UDP echo is used to measure response times and test end-to-end connectivity.

In the following figure, Switch A is configured as an IP SLAs Responder and Switch B is configured as the source IP SLAs device.

Figure 3: UDP Echo Operation



The response time (round-trip time) is computed by measuring the time taken between sending a UDP echo request message from Switch B to the destination switch--Switch A--and receiving a UDP echo reply from Switch A. UDP echo accuracy is enhanced by using the responder at Switch A, the destination Cisco device. If the destination switch is a Cisco switch, the IP SLAs Responder sends a UDP datagram to any port number that you specified. Using the IP SLAs Responder is optional for a UDP echo operation when using Cisco devices. The IP SLAs Responder cannot be configured on non-Cisco devices.

The results of a UDP echo operation can be useful in troubleshooting issues with business-critical applications by determining the round-trip delay times and testing connectivity to both Cisco and non-Cisco devices.

Guidelines and Limitations for UDP Echo Operations

Configuring CoPP for IP SLA Packets

When using IP SLA operations on a large scale, a specific CoPP configuration to allow the IP SLA packets to pass through might be needed. Since IP SLA uses user defined UDP ports, there is no way to allow all IP SLA packets to the control plane. However, you can specify each destination/source port that IP SLA can use.

For more information about the verified scalability of the number of IP SLA probes, see the *Cisco Nexus 7000 Series NX-OS Verified Scalability Guide*.

The following shows an example of a CoPP configuration that allows IP SLA packets to pass through. It assumes destination ports and source ports in the range of 6500-7000.

```
ip access-list copp-system-sla-allow
  10 remark ### ALLOW SLA control packets from 1.1.1.0/24
  20 permit udp 1.1.1.0/24 any eq 1967
  30 remark ### ALLOW SLA data packets from 1.1.1.0/24 using ports 6500-7000
  40 permit udp 1.1.1.0/24 any range 6500 7000
  statistics per-entry
ip access-list copp-system-sla-deny
```

```
  10 remark ### this is a catch-all to match any other traffic
  20 permit ip any any
  statistics  per-entry
class-map type control-plane match-any copp-system-class-management-allow
  match access-group name copp-system-sla-allow
class-map type control-plane match-any copp-system-class-management-deny
  match access-group name copp-system-sla-deny
policy-map type control-plane copp-system-policy
    class copp-system-class-management-allow
    set cos 7
    police cir 4500 kbps bc 250 ms conform transmit violate drop
    class copp-system-class-management-deny
    police cir 4500 kbps bc 250 ms conform drop violate drop
control-plane
  service-policy input copp-system-policy
```

Configuring the IP SLAs Responder on the Destination Device

Before you begin

If you are using the IP SLAs Responder, ensure that the networking device to be used as the responder is a Cisco device and that you have connectivity to that device through the network.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `switch> enable`	Enables privileged EXEC mode Enter your password if prompted.
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **ip sla responder**
	Example:
	`switch(config)# ip sla responder`
	• **ip sla responder udp-echo ipaddress** *ip-address* **port** *port*
	Example:
	`switch(config)# ip sla responder udp-echo ipaddress 172.29.139.132 port 5000`
		• Temporarily enables the IP SLAs Responder functionality on a Cisco device in response to control messages from the source.
		• Required only if the protocol control is disabled on the source. This command permanently enables the IP SLAs Responder functionality on a specified IP address and port.
		Control is enabled by default.

Step 4	**exit** **Example:** `switch(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Basic UDP Echo Operation on the Source Device

This section describes how to configure a basic UDP echo operation on the source.

Note To add proactive threshold conditions and reactive triggering for generating traps, or for starting another operation, to an IP SLAs operation, see the "Configuring Proactive Threshold Monitoring" section.

Before you begin

If you are using the IP SLAs Responder, ensure that you have completed the "Configuring the IP SLAs Responder on the Destination Device" section before you start this task.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `switch(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-echo** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*} **sourceport** *port-number*] [**control** {**enable** | **disable**}] **Example:** `switch(config-ip-sla)# udp-echo 172.29.139.134 5000`	Defines a UDP echo operation and enters IP SLA UDP configuration mode. Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target switches.
Step 5	**frequency** *seconds* **Example:** `switch(config-ip-sla-udp)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**end** **Example:** `switch(config-ip-sla-udp)# end`	Returns to privileged EXEC mode.

Configuring a UDP Echo Operation with Optional Parameters on the Source Device

This section describes how to configure a UDP echo operation with optional parameters on the source device.

Note To add proactive threshold conditions and reactive triggering for generating traps, or for starting another operation, to an IP SLAs operation, see the "Configuring Proactive Threshold Monitoring" section.

Before you begin

If you are using an IP SLAs Responder in this operation, the responder must be configured on the destination device. See the "Configuring the IP SLAs Responder on the Destination Device" section.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `switch(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-echo** {destination-ip-address | destination-hostname} destination-port [**source-ip** {ip-address | hostname} **source-port** *port-number*] [**control** {**enable** | **disable**}] **Example:** `switch(config-ip-sla)# udp-echo 172.29.139.134 5000`	Defines a UDP echo operation and enters IP SLA UDP configuration mode. Use the control disable keyword combination only if you disable the IP SLAs control protocol on both the source and target switches.
Step 5	**history buckets-kept** *size* **Example:** `switch(config-ip-sla-udp)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 6	**data-pattern** *hex-pattern* **Example:**	(Optional) Specifies the data pattern in an IP SLAs operation to test for data corruption.

	Command or Action	Purpose
	`switch(config-ip-sla-udp)# data-pattern`	
Step 7	**history distributions-of-statistics-kept** *size* **Example:** `switch(config-ip-sla-udp)# history distributionsof- statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 8	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `switch(config-ip-sla-udp)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 9	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `switch(config-ip-sla-udp)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 10	**frequency** *seconds* **Example:** `switch(config-ip-sla-udp)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 11	**history hours-of-statistics-kept** *hours* **Example:** `switch(config-ip-sla-udp)# history hours-ofstatistics- kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 12	**history lives-kept** *lives* **Example:** `switch(config-ip-sla-udp)# history lives-kept 5`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 13	**owner** *owner-id* **Example:** `switch(config-ip-sla-udp)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 14	**request-data-size** *bytes* **Example:** `switch(config-ip-sla-udp)# request-data-size 64`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.
Step 15	**history statistics-distribution-interval** *milliseconds* **Example:** `switch(config-ip-sla-udp)# history statistics distribution- interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.

	Command or Action	Purpose
Step 16	**tag** *text* **Example:** `switch(config-ip-sla-udp)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 17	**threshold** *milliseconds* **Example:** `switch(config-ip-sla-udp)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 18	**timeout** *milliseconds* **Example:** `switch(config-ip-sla-udp)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 19	**tos** *number* **Example:** `switch(config-ip-sla-jitter)# tos 160`	(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.
Step 20	**verify-data** **Example:** `switch(config-ip-sla-udp)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.
Step 21	**exit** **Example:** `switch(config-ip-sla-udp)# exit`	Exits UDP configuration submode and returns to global configuration mode.

Scheduling IP SLAs Operations

This section describes how to schedule IP SLAs operations.

Before you begin

Note

- All IP SLAs operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group is limited to a maximum of 125 characters, including commas (,).

Tip

- If the IP SLAs operation is not running and generating statistics, add the **verify-data** command to the configuration of the operation (while configuring in IP SLA configuration mode) to enable data verification. When enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error**commands to help troubleshoot issues with an IP SLAs operation.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **ip sla schedule** *operation-number* [**life forever** { | *seconds*}] [**starttime** {*hh : mm*[: *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh : mm : ss*}] [**ageout** *seconds*] [**recurring**]
	Example:
	`ip sla schedule operation-number [life {forever | seconds}] [starttime {hh : mm[: ss] [month day | day month] | pending | now | after hh : mm : ss}] [ageout seconds] [recurring]`
	• **ip sla group schedule** *group-operation-number operation-id-numbers* **schedule-period** *schedule-period-range* [**ageout** *seconds*] [**frequency** *group-operation-frequency*] [**life**{**forever** | *seconds*}] [**starttime**{ *hh:mm*[:*ss*] [*month day* | *day month]* | **pending** | **now** | **after** *hh:mm:ss*}]
	Example:
	`switch(config)# ip sla group schedule 1 3,4,6-9`
		-
		• For individual IP SLAs operations only:
		Configures the scheduling parameters for an individual IP SLAs operation.
		• For the multioperations scheduler only:
		Specifies an IP SLAs operation group number and the range of operation numbers to be scheduled in global configuration mode.

	Command or Action	Purpose
Step 4	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `switch# show ip sla group schedule`	(Optional) Displays the IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `switch# show ip sla configuration`	(Optional) Displays the IP SLAs configuration details.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps or for starting another operation, see the Configuring Proactive Threshold Monitoring, on page 81 section.

To display statistics of an IP SLA operation over the last one hour and interpret the results, use the **show ip sla statistics** command. Checking the output for fields that correspond to criteria in your service level agreement helps you to determine whether the service metrics are acceptable. To display the aggregated IP SLA history, use the **show ip sla statistics aggregated** command.

Configuration Example for a UDP Echo Operation

This example shows how to configure an IP SLAs operation type of UDP echo that starts immediately and runs indefinitely:

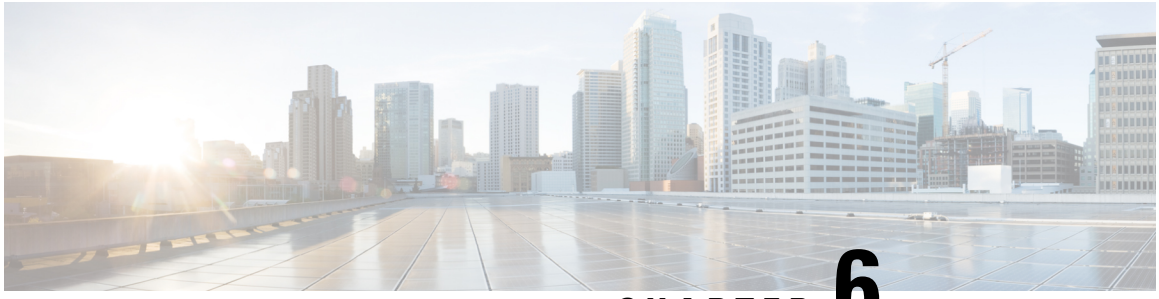
```
ip sla 5
udp-echo 172.29.139.134 5000
frequency 30
request-data-size 160
tos 128
timeout 1000
tag FLL-RO
ip sla schedule 5 life forever start-time now
```

Feature History for UDP Echo

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 11: Feature History for UDP Echo

Feature Name	Release	Feature Information
UDP Echo	6.1(1)	This feature was introduced.



CHAPTER 6

Configuring IP SLAs TCP Connect Operations

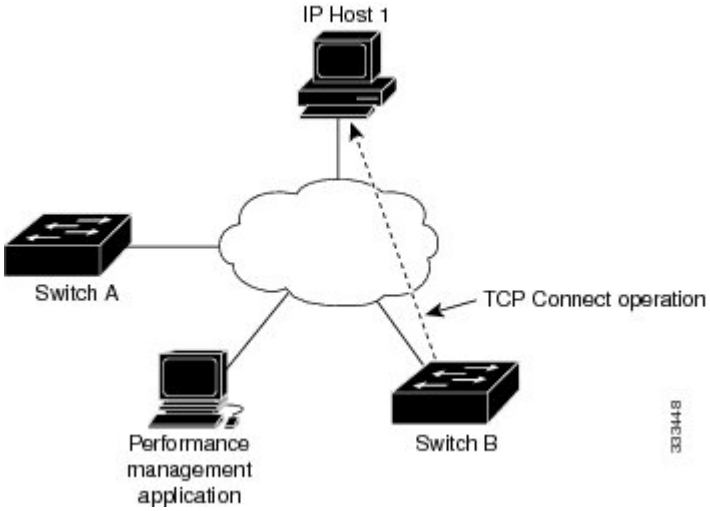
This chapter describes how to configure an IP Service Level Agreements (SLAs) TCP Connect operation to measure the response time taken to perform a TCP Connect operation between a Cisco switch and devices using IPv4. TCP Connect accuracy is enhanced by using the IP SLAs Responder at the destination Cisco switch. This chapter also describes how the results of the TCP Connect operation can be displayed and analyzed to determine how the connection times to servers and hosts within your network can affect IP service levels. The TCP Connect operation is useful for measuring response times for a server used for a particular application or connectivity testing for server availability.

This chapter includes these sections.

Information About the TCP Connect Operation

The IP SLAs TCP Connect operation measures the response time taken to perform a TCP Connect operation between a Cisco switch and devices using IP. TCP is a transport layer (Layer 4) Internet protocol that provides reliable full-duplex data transmission. The destination device can be any device using IP or an IP SLAs Responder.

In the following figure, Switch B is configured as the source IP SLAs device and a TCP Connect operation is configured with the destination device as IP Host 1.



The connection response time is computed by measuring the time taken between sending a TCP request message from Switch B to IP Host 1 and receiving a reply from IP Host 1.

TCP Connect accuracy is enhanced by using the IP SLAs Responder at the destination Cisco device. If the destination switch is a Cisco switch, the IP SLAs Responder makes a TCP connection to any port number that you specified. If the destination is not a Cisco IP host, then you must specify a known destination port number such as 21 for FTP, 23 for Telnet, or 80 for an HTTP server.

Using the IP SLAs Responder is optional for a TCP Connect operation when using Cisco devices. The IP SLAs Responder cannot be configured on non-Cisco devices.

TCP Connect is used to test virtual circuit availability or application availability. Server and application connection performance can be tested by simulating Telnet, SQL, and other types of connections to help you verify your IP service levels.

Guidelines and Limitations for Configuring IP SLAs TCP Connect Operations

Configuring CoPP for IP SLA Packets

When using IP SLA operations on a large scale, a specific CoPP configuration to allow the IP SLA packets to pass through might be needed. Since IP SLA uses user defined UDP ports, there is no way to allow all IP SLA packets to the control plane. However, you can specify each destination/source port that IP SLA can use.

For more information about the verified scalability of the number of IP SLA probes, see the *Cisco Nexus 7000 Series NX-OS Verified Scalability Guide*.

The following shows an example of a CoPP configuration that allows IP SLA packets to pass through. It assumes destination ports and source ports in the range of 6500-7000.

```
ip access-list copp-system-sla-allow
  10 remark ### ALLOW SLA control packets from 1.1.1.0/24
  20 permit udp 1.1.1.0/24 any eq 1967
  30 remark ### ALLOW SLA data packets from 1.1.1.0/24 using ports 6500-7000
```

```
  40 permit udp 1.1.1.0/24 any range 6500 7000
  statistics per-entry
ip access-list copp-system-sla-deny
  10 remark ### this is a catch-all to match any other traffic
  20 permit ip any any
  statistics  per-entry
class-map type control-plane match-any copp-system-class-management-allow
  match access-group name copp-system-sla-allow
class-map type control-plane match-any copp-system-class-management-deny
  match access-group name copp-system-sla-deny
policy-map type control-plane copp-system-policy
    class copp-system-class-management-allow
    set cos 7
    police cir 4500 kbps bc 250 ms conform transmit violate drop
    class copp-system-class-management-deny
    police cir 4500 kbps bc 250 ms conform drop violate drop
control-plane
  service-policy input copp-system-policy
```

Configuring the IP SLAs Responder on the Destination Device

This section describes how to configure the IP SLAs Responder on the destination device.

Before you begin

If you are using the IP SLAs Responder, ensure that the networking device to be used as the responder is a Cisco device and that you have connectivity to that device through the network.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **ip sla responder**
	Example:
	`switch(config)# ip sla responder`
	• **ip sla responder tcp-connect ipaddress** *ip-address* **port** *port*
	Example:
	`switch(config)# ip sla responder tcp-connect ipaddress 172.29.139.132 port 5000`
		-
		• (Optional) Temporarily enables IP SLAs Responder functionality on a Cisco device in response to control messages from source.
		• (Optional) Required only if protocol control is disabled on the source. The command permanently enables the IP SLAs Responder functionality on a specified IP address and port.
		Control is enabled by default.

	Command or Action	Purpose
Step 4	**exit** **Example:** `switch(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring and Scheduling a TCP Connect Operation on the Source Device

This section describes how to configure and schedule a TCP connect operation on the source device.

Perform only one of the following tasks to configure and schedule a TCP connect operation on the source device:

- Configuring and scheduling a basic TCP connect operation on the source device
- Configuring and scheduling a TCP connect operation with optional parameters on the source device

Configuring and Scheduling a Basic TCP Connect Operation on the Source Device

This section describes how to configure and schedule a basic TCP connect operation on a source device.

Note If an IP SLAs Responder is permanently enabled on the destination IP address and port, use the **control disable** keywords with the **tcp-connect** command to disable control messages.

Tip

- If the IP SLAs operation is not running and generating statistics, add the **verify-data** command to the configuration of the operation (while configuring in IP SLA configuration mode) to enable data verification. When enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla sender trace** and **debug ip sla sender error** commands to help troubleshoot issues with an IP SLAs operation.

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `switch(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**tcp-connect** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*} **source-port***port-number*] [**control** {**enable** | **disable**}] **Example:** `switch(config-ip-sla)# tcp-connect 172.29.139.132 5000`	Defines a TCP Connect operation and enters IP SLA TCP configuration mode. Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target switches.
Step 5	**frequency** *seconds* **Example:** `switch(config-ip-sla-tcp)# frequency 60`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**exit** **Example:** `switch(config-ip-sla-tcp)# exit`	Exits IP SLA TCP configuration mode and returns to global configuration mode.
Step 7	**ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm*[:*ss*] [*monthday* | *daymonth*] | **pending** | **now** | **after** *hh*:*mm*:*ss*] [**ageout** *seconds*] [**recurring**] **Example:** `switch(config)# ip sla schedule 10 start-time now life forever`	Configures the scheduling parameters for an individual IP SLAs operation.
Step 8	**exit** **Example:** `switch(config)# exit`	(Optional) Exits the global configuration mode and returns to privileged EXEC mode.

Example

This example shows how to configure an IP SLAs operation type of TCP Connect that will start immediately and run indefinitely:

```
ip sla 9
 tcp-connect 172.29.139.132 5000
 frequency 10
!
ip sla schedule 9 life forever start-time now
```

What to do next

To add proactive threshold conditions and reactive triggering for generating traps or for starting another operation, see the Configuring Proactive Threshold Monitoring, on page 81 section.

To display statistics of an IP SLA operation over the last one hour and interpret the results, use the **show ip sla statistics** command. Checking the output for fields that correspond to criteria in your service level agreement helps you to determine whether the service metrics are acceptable. To display the aggregated IP SLA history, use the **show ip sla statistics aggregated** command.

Configuring and Scheduling a TCP Connect Operation with Optional Parameters on the Source Device

This section describes how to configure and schedule a TCP connect operation with optional parameters on a source device.

Note If an IP SLAs Responder is permanently enabled on the destination IP address and port, use the **control disable** keywords with the **tcp-connect** command to disable control messages.

Tip

- If the IP SLAs operation is not running and generating statistics, add the **verify-data** command to the configuration of the operation (while configuring in IP SLA configuration mode) to enable data verification. When enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

Procedure

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

	Command or Action	Purpose
	`switch> enable`	
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `switch(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**tcp-connect** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] [**control** {**enable** | **disable**}] **Example:** `switch(config-ip-sla)# tcp-connect 172.29.139.132 5000`	Defines a TCP Connect operation and enters IP SLA TCP configuration mode. Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target switches.
Step 5	**history buckets-kept** *size* **Example:** `switch(config-ip-sla-tcp)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 6	**history distributions-of-statistics-kept** *size* **Example:** `switch(config-ip-sla-tcp)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 7	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `switch(config-ip-sla-tcp)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 8	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `switch(config-ip-sla-tcp)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.

	Command or Action	Purpose
Step 9	**frequency** *seconds* **Example:** `switch(config-ip-sla-tcp)# frequency 60`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 10	**history hours-of-statistics-kept** *hours* **Example:** `switch(config-ip-sla-tcp)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 11	**history lives-kept** *lives* **Example:** `switch(config-ip-sla-tcp)# history lives-kept 5`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 12	**owner** *owner-id* **Example:** `switch(config-ip-sla-tcp)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 13	**history statistics-distribution-interval** *milliseconds* **Example:** `switch(config-ip-sla-tcp)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 14	**tag** *text* **Example:** `switch(config-ip-sla-tcp)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 15	**threshold** *milliseconds* **Example:** `switch(config-ip-sla-tcp)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 16	**timeout** *milliseconds* **Example:** `switch(config-ip-sla-tcp)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

	Command or Action	Purpose
Step 17	**tos** *number* **Example:** `switch(config-ip-sla-jitter)# tos 160` **Example:**	(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.
Step 18	**exit** **Example:** `switch(config-ip-sla-tcp)# exit`	Exits TCP configuration submode and returns to global configuration mode.
Step 19	**ip sla schedule** *operation-number* [**life** {**forever**| *seconds*}] [**start-time** {*hh*:*mm*[:*ss*] [*monthday* | *daymonth*] | **pending** | **now** | **after** *hh*:*mm*:*ss*] [**ageout** *seconds*] [**recurring**] **Example:** `switch(config)# ip sla schedule 10 start-time now life forever`	Configures the scheduling parameters for an individual IP SLAs operation.
Step 20	**exit** **Example:** `switch(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 21	**show ip sla configuration** [*operation-number*] **Example:** `switch# show ip sla configuration 10`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

Example

This example shows how to configure all the IP SLAs parameters (including defaults) for the TCP Connect operation number 10:

```
switch# show ip sla configuration 10
IP SLAs Infrastructure Engine-III
Entry number: 10
Owner: admin
Tag: TelnetPollServer1
Operation timeout (milliseconds): 10000
Type of operation to perform: tcp-connect
Target address/Source address: 101.101.101.1/0.0.0.0
Target port/Source port: 5000/0
Type Of Service parameter: 0xa0
Vrf Name: default
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
```

```
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 10000
Distribution Statistics:
   Number of statistic hours kept: 4
   Number of statistic distribution buckets kept: 5
   Statistic distribution interval (milliseconds): 10
Enhanced History:
   Aggregation Interval:900 Buckets: 100
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 25
   History Filter Type: Failures
```

What to do next

To add proactive threshold conditions and reactive triggering for generating traps or for starting another operation, see the Configuring Proactive Threshold Monitoring, on page 81 section.

To display statistics of an IP SLA operation over the last one hour and interpret the results, use the **show ip sla statistics** command. Checking the output for fields that correspond to criteria in your service level agreement helps you to determine whether the service metrics are acceptable. To display the aggregated IP SLA history, use the **show ip sla statistics aggregated** command.

Configuration Example for a TCP Connect Operation

This example shows how to configure a TCP Connect operation from Switch B to the Telnet port (TCP port 23) of IP Host 1 (IP address 10.0.0.1), as shown in the "TCP Connect Operation" figure in the "Information About the IP SLAs TCP Connect Operation" section. The operation is scheduled to start immediately. In this example, the control protocol is disabled on the source (Switch B). IP SLAs use the control protocol to notify the IP SLAs Responder to enable the target port temporarily. This action allows the Responder to reply to the TCP Connect operation. In this example, because the target is not a switch and a well-known TCP port is used, there is no need to send the control message.

Switch A Configuration

```
configure terminal
 ip sla responder tcp-connect ipaddress 10.0.0.1 port 23
```

Switch B Configuration

```
ip sla 9
 tcp-connect 10.0.0.1 23 control disable
 frequency 30
 tos 128
 timeout 1000
 tag FLL-RO
ip sla schedule 9 start-time now
```

This example shows how to configure a TCP Connect operation with a specific port, port 21, and without an IP SLAs Responder. The operation is scheduled to start immediately and run indefinitely.

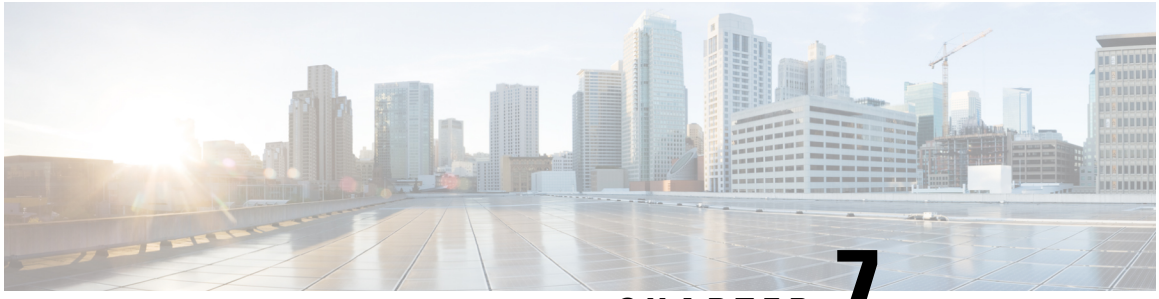
```
ip sla 9
 tcp-connect 173.29.139.132 21 control disable
 frequency 30
ip sla schedule 9 life forever start-time now
```

Feature History for TCP Connect

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 12: Feature History for TCP Connect

Feature Name	Release	Feature Information
TCP Connect	6.1(1)	This feature was introduced.



CHAPTER 7

Configuring a Multioperations Scheduler

This chapter describes how to schedule multiple operations using the IP Service Level Agreements (IP SLAs) Multioperations Scheduler.

This chapter includes the following sections:

Information About the IP SLAs Multioperations Scheduler

Normal scheduling of IP SLAs operations allows you to schedule one operation at a time. If you have large networks with thousands of IP SLAs operations to monitor network performance, normal scheduling (scheduling each operation individually) is inefficient and time-consuming.

Multiple operations scheduling allows you to schedule multiple IP SLAs operations using a single command through the command-line interface (CLI) or the CISCO-RTTMON-MIB. This feature allows you to control the amount of IP SLAs monitoring traffic by scheduling the operations to run at evenly distributed times. You must specify the operation ID numbers to be scheduled and the time range over which all the IP SLAs operations should start. This feature automatically distributes the IP SLAs operations at equal intervals over a specified time frame. The spacing between the operations (start interval) is calculated and the operations are started. This distribution of IP SLAs operations helps to minimize the CPU utilization and enhances the scalability of the network.

The IP SLAs multiple operations scheduling functionality allows you to schedule multiple IP SLAs operations as a group, using the following configuration parameters:

- Group operation number—Group configuration or group schedule number of the IP SLAs operation to be scheduled.
- Operation ID numbers—A list of IP SLAs operation ID numbers in the scheduled operation group.
- Schedule period—Amount of time for which the IP SLAs operation group is scheduled.
- Ageout—Amount of time to keep the operation in memory when it is not actively collecting information. By default, the operation remains in memory indefinitely.
- Frequency—Amount of time after which each IP SLAs operation is restarted. When the frequency option is specified, it overwrites the operation frequency of all operations that belong to the group. When the frequency option is not specified, the frequency for each operation is set to the value of the schedule period.
- Life—Amount of time in which the operation actively collects information. You can configure the operation to run indefinitely. By default, the lifetime of an operation is one hour.
- Start time—Time when the operation starts collecting information. You can specify an operation to start immediately or at an absolute start time using hours, minutes, seconds, day, and month.

The IP SLAs multiple operations scheduling functionality schedules the maximum number of operations possible without aborting. However, this functionality skips those IP SLAs operations that are already running or those operations that are not configured and therefore do not exist. The total number of operations are calculated based on the number of operations specified in the command, irrespective of the number of operations that are missing or already running. The IP SLAs multiple operations scheduling functionality displays a message that shows the number of active and missing operations. However, these messages are displayed only if you schedule operations that are not configured or are already running.

A main benefit for scheduling multiple IP SLAs operations is that the load on the network is reduced by distributing the operations equally over a scheduled period. This distribution helps you to achieve more consistent monitoring coverage. Consider configuring 60 operations to start during the same 1-second interval over a 60-second schedule period. If a network failure occurs 30 seconds after all 60 operations have started and the network is restored before the operations are due to start again (in another 30 seconds), this failure would never be detected by any of the 60 operations. However, if the 60 operations are distributed equally at 1-second intervals over a 60-second schedule period, then some of the operations would detect the network failure. Conversely, if a network failure occurs when all 60 operations are active, all 60 operations would fail, indicating that the failure is possibly more severe than it really is.

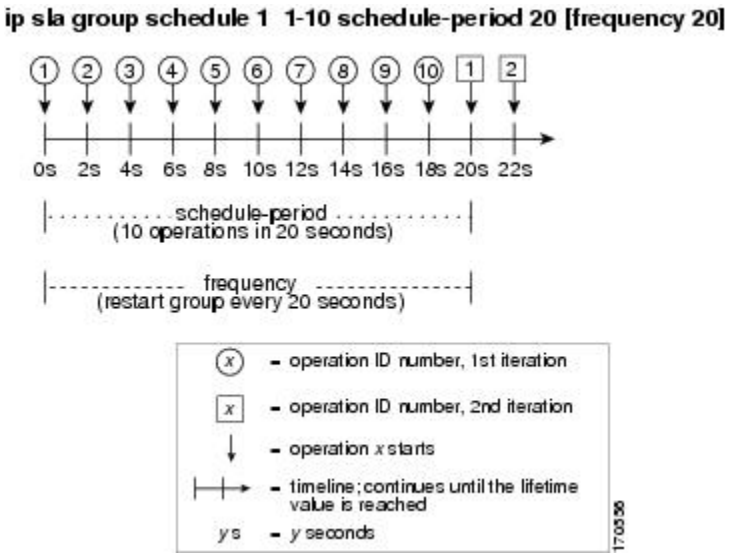
Operations of the same type and same frequency should be used for IP SLAs multiple operations scheduling. If you do not specify a frequency, the default frequency is the same as that of the schedule period. The schedule period is the period of time in which all the specified operations should run.

Default Behavior of IP SLAs Multiple Operations Scheduling

The IP SLAs Multiple Operations Scheduling feature allows you to schedule multiple IP SLAs operations as a group.

The following figure shows the scheduling of operation group 1 that includes operation 1 to operation 10. Operation group 1 has a schedule period of 20 seconds, which means that all operations in the group are started at equal intervals within a 20-second period. By default, the frequency is set to the same value as the configured schedule period. As shown in the figure, configuring the frequency is optional because 20 is the default.

Figure 4: Schedule Period Equals Frequency--Default Behavior



In this example, the first operation (operation 1) in operation group 1 starts at 0 seconds. All 10 operations in operation group 1 (operation 1 to operation 10) must be started in the schedule period of 20 seconds. The start time of each IP SLAs operation is evenly distributed over the schedule period by dividing the schedule period by the number of operations (20 seconds divided by 10 operations). Therefore, each operation starts 2 seconds after the previous operation.

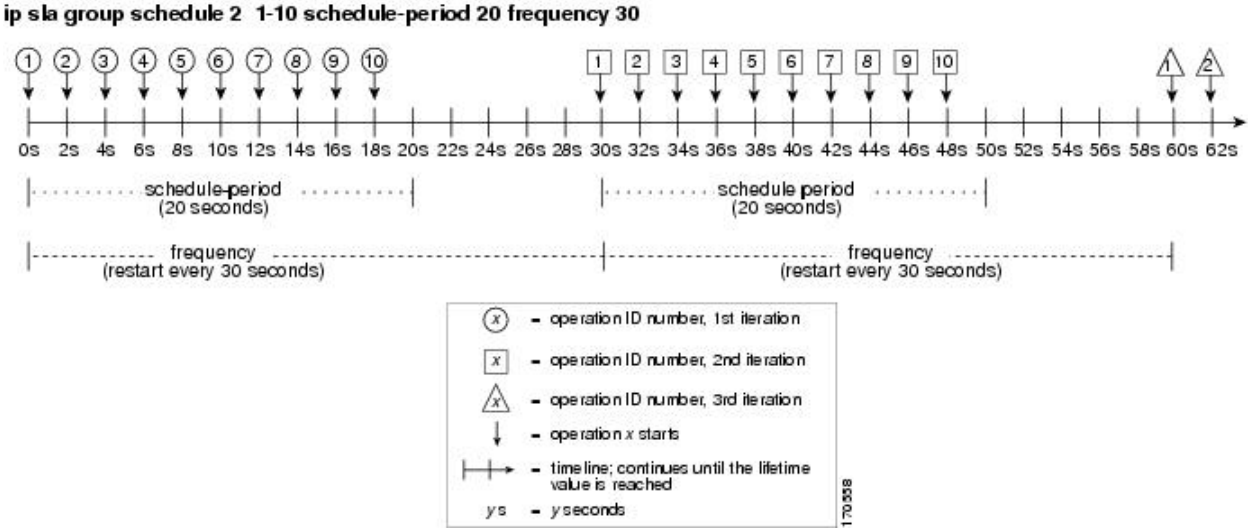
The frequency is the period of time that passes before the operation group is started again (repeated). If the frequency is not specified, the frequency is set to the value of the schedule period. In the example shown in the figure, operation group 1 starts again every 20 seconds. This configuration provides optimal division (spacing) of operations over the specified schedule period.

Multiple Operations Scheduling with Scheduling Period Less Than Frequency

The frequency value is the amount of time that passes before the schedule group is restarted. If the schedule period is less than the frequency, there is a period of time in which no operations are started.

The following figure shows the scheduling of operation 1 to operation 10 within operation group 2. Operation group 2 has a schedule period of 20 seconds and a frequency of 30 seconds.

Figure 5: Schedule Period Is Less Than Frequency



In this example, the first operation (operation 1) in operation group 2 starts at 0 seconds. All 10 operations in operation group 2 (operation 1 to operation 10) must be started in the schedule period of 20 seconds. The start time of each IP SLAs operation is evenly distributed over the schedule period by dividing the schedule period by the number of operations (20 seconds divided by 10 operations). Therefore, each operation starts 2 seconds after the previous operation.

In the first iteration of operation group 2, operation 1 starts at 0 seconds, and the last operation (operation 10) starts at 18 seconds. However, because the group frequency has been configured to 30 seconds, each operation in the operation group is restarted every 30 seconds. So, after 18 seconds, there is a gap of 10 seconds as no operations are started in the time from 19 seconds to 29 seconds. At 30 seconds, the second iteration of operation group 2 starts. As all ten operations in the operation group 2 must start at an evenly distributed interval in the configured schedule period of 20 seconds, the last operation (operation 10) in the operation group 2 always starts 18 seconds after the first operation (operation 1).

As shown in the figure, the following events occur:

- At 0 seconds, the first operation (operation 1) in operation group 2 is started.
- At 18 seconds, the last operation (operation 10) in operation group 2 is started, which means that the first iteration (schedule period) of operation group 1 ends here.
- From 19 to 29 seconds, no operations are started.
- At 30 seconds, the first operation (operation 1) in operation group 2 is started again. The second iteration of operation group 2 starts here.
- At 48 seconds (18 seconds after the second iteration started), the last operation (operation 10) in operation group 2 is started, and the second iteration of operation group 2 ends.
- At 60 seconds, the third iteration of operation group 2 starts.

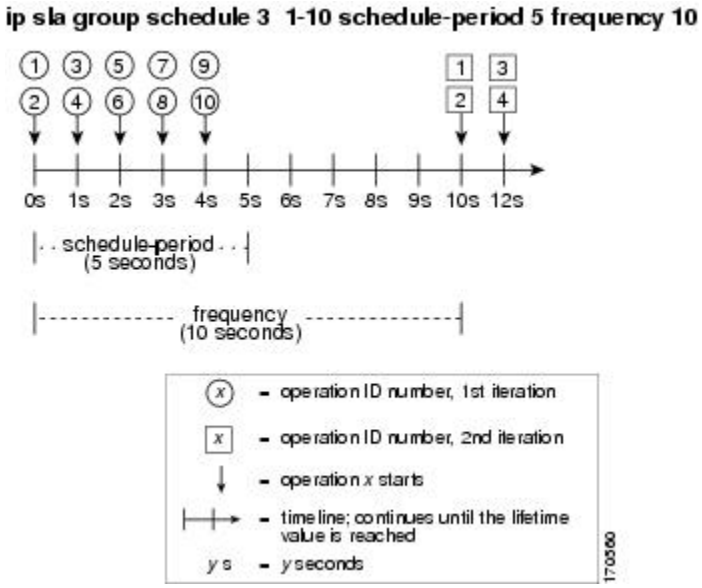
This process continues until the lifetime of operation group 2 ends. The lifetime value is configurable. The default lifetime for an operation group is forever.

Multiple Operations Scheduling When the Number of IP SLAs Operations are Greater than the Schedule Period

The minimum time interval between the start of IP SLAs operations in a group operation is 1 second. Therefore, if the number of operations to be scheduled is greater than the schedule period, the IP SLAs multiple operations scheduling functionality schedules more than one operation to start within the same 1-second interval. If the number of operations getting scheduled does not equally divide into 1-second intervals, the operations are equally divided at the start of the schedule period with the remaining operations to start at the last 1-second interval.

The following figure shows the scheduling of operation 1 to operation 10 within operation group 3. Operation group 3 has a schedule period of 5 seconds and a frequency of 10 seconds.

Figure 6: Number of IP SLAs Operations Is Greater Than the Schedule Period—Even Distribution



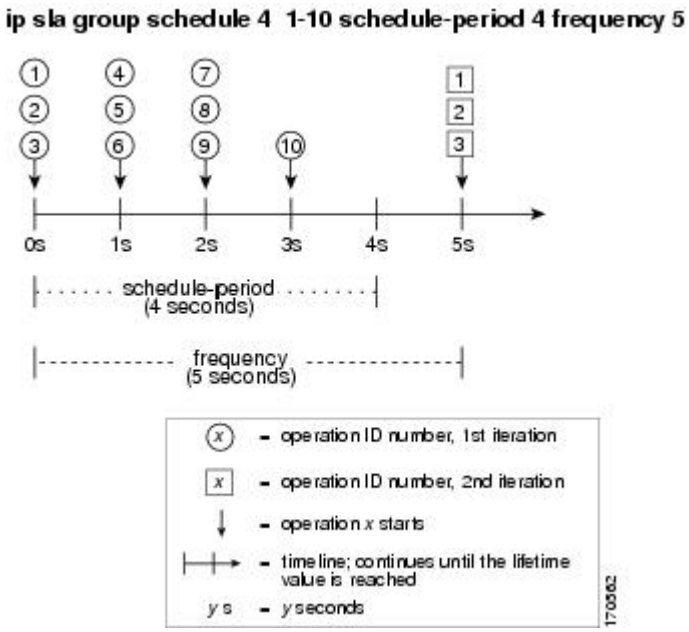
In this example, when dividing the schedule period by the number of operations (5 seconds divided by 10 operations, which equals one operation every 0.5 seconds), the start time of each IP SLAs operation is less than 1 second. Because the minimum time interval between the start of IP SLAs operations in a group operation is 1 second, the IP SLAs multiple operations scheduling functionality instead calculates how many operations it should start in each 1-second interval by dividing the number of operations by the schedule period (10 operations divided by 5 seconds). Therefore, as shown in the previous figure, two operations are started every 1 second.

As the frequency is set to 10 in this example, each iteration of operation group 3 will start 10 seconds after the start of the previous iteration. However, this distribution is not optimal as there is a gap of 5 seconds (frequency minus schedule period) between the cycles.

If the number of operations getting scheduled does not equally divide into 1-second intervals, then the operations are equally divided at the start of the schedule period with the remaining operations to start at the last 1-second interval.

The following figure shows the scheduling of operation 1 to operation 10 within operation group 4. Operation group 4 has a schedule period of 4 seconds and a frequency of 5 seconds.

Figure 7: Number of IP SLAs Operations Is Greater Than the Schedule Period—Uneven Distribution



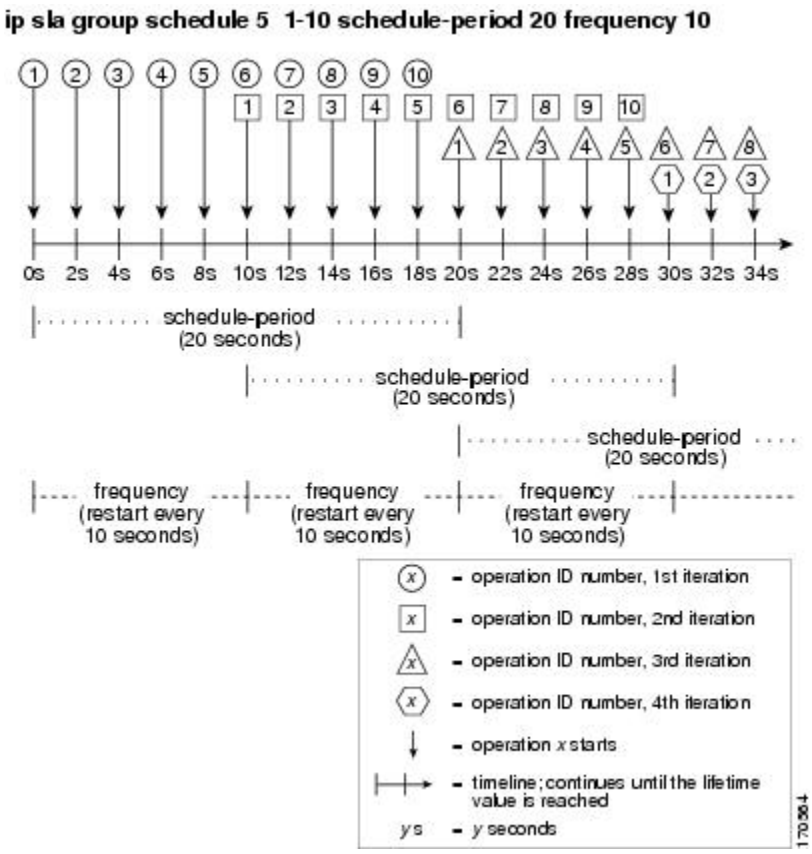
In this example, the IP SLAs multiple operations scheduling functionality calculates how many operations it should start in each 1-second interval by dividing the number of operations by the schedule period (10 operations divided by 4 seconds, which equals 2.5 operations every 1 second). Because the number of operations does not equally divide into 1-second intervals, this number will be rounded off to the next whole number (see the figure) with the remaining operations to start at the last 1-second interval.

Multiple Operations Scheduling with Scheduling Period Greater Than Frequency

The value of the frequency is the amount of time that passes before the schedule group is restarted. If the schedule period is greater than the frequency, there is a period of time in which the operations in one iteration of an operation group overlaps with the operations of the following iteration.

The following figure shows the scheduling of operation 1 to operation 10 within operation group 5. Operation group 5 has a schedule period of 20 seconds and a frequency of 10 seconds.

Figure 8: IP SLAs Group Scheduling with Schedule Period Greater Than Frequency



In this example, the first operation (operation 1) in operation group 5 starts at 0 seconds. All 10 operations in operation group 5 (operation 1 to operation 10) must be started in the schedule period of 20 seconds. The start time of each IP SLAs operation is evenly distributed over the schedule period by dividing the schedule period by the number of operations (20 seconds divided by 10 operations). Therefore, each operation starts 2 seconds after the previous operation.

In the first iteration of operation group 5, operation 1 starts at 0 seconds, and operation 10, the last operation in the operation group, starts at 18 seconds. Because the operation group is configured to restart every 10 seconds (**frequency 10**), the second iteration of operation group 5 starts again at 10 seconds, before the first iteration is completed. Therefore, an overlap of operations 6 to 10 of the first iteration occurs with operations 1 to 5 of the second iteration during the time period of 10 to 18 seconds (see the previous figure). Similarly, there is an overlap of operations 6 to 10 of the second iteration with operations 1 to 5 of the third iteration during the time period of 20 to 28 seconds.

In this example, the start time of operation 1 and operation 6 does not need to be at exactly the same time, but will be within the same 2-second interval.

The configuration described in this section is not recommended because you can configure multiple operations to start within the same 1-second interval by configuring the number of operations greater than the schedule period.

IP SLAs Random Scheduler

The IP SLAs Multioperation Scheduler feature provides the capability to schedule multiple IP SLAs operations to begin at intervals equally distributed over a specified duration of time and to restart at a specified frequency. With the IP SLAs Random Scheduler feature, you can now schedule multiple IP SLAs operations to begin at random intervals uniformly distributed over a specified duration of time and to restart at uniformly distributed random frequencies within a specified frequency range. Random scheduling improves the statistical metrics for assessing network performance.

Note The IP SLAs Random Scheduler feature is not in compliance with RFC 2330 because it does not account for inter-packet randomness.

The random scheduler option is disabled by default. To enable the random scheduler option, you must set a frequency range when configuring a group schedule in global configuration mode. The group of operations restarts at uniformly distributed random frequencies within the specified frequency range. The following guidelines apply for setting the frequency range:

- The starting value of the frequency range should be greater than the timeout values of all the operations in the group operation.
- The starting value of the frequency range should be greater than the schedule period (amount of time for which the group operation is scheduled). This guideline ensures that the same operation does not get scheduled more than once within the schedule period.

The following guidelines apply if the random scheduler option is enabled:

- The individual operations in a group operation will be uniformly distributed to begin at random intervals over the schedule period.
- The group of operations restarts at uniformly distributed random frequencies within the specified frequency range.
- The minimum time interval between the start of each operation in a group operation is 100 milliseconds (0.1 seconds). If the random scheduler option is disabled, the minimum time interval is 1 second.
- Only one operation can be scheduled to begin at any given time. If the random scheduler option is disabled, multiple operations can begin at the same time.
- The first operation will always begins at 0 milliseconds of the schedule period.
- The order in which each operation in a group operation begins is random.

Prerequisites for an IP SLAs Multioperation Scheduler

- Configure the IP SLAs operations to be included in a group before scheduling the group.
- Determine the IP SLAs operations you want to schedule as a single group.
- Identify the network traffic type and the location of your network management station.
- Identify the topology and the types of devices in your network.
- Decide on the frequency of testing for each operation.

Scheduling Multiple IP SLAs Operations

This section describes how to schedule multiple IP SLAs operations.

Before you begin

Note

- The frequency of all operations scheduled in a multioperation group should be the same.
- The operation ID numbers are limited to a maximum of 125 characters. Do not give large integer values as operation ID numbers.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 3	**ip sla group schedule** *group-operation-number operation-id-numbers* **schedule-period** *schedule-period-range* [**ageout** *seconds*] [**frequency** *group-operation-frequency*] [**life**{**forever** | *seconds*}] [**start-time** {*hh:mm*[*:ss*] [*monthday* | *daymonth*] | **pending** | **now** | **after** *hh:mm:ss*}] **Example:** `switch(config)# ip sla group schedule 1 3,4,6-9`	Specifies an IP SLAs operation group number and the range of operation numbers to be scheduled in global configuration mode.
Step 4	**exit** **Example:** `switch(config)# exit`	Returns to the privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `switch# show ip sla group schedule`	(Optional) Displays the IP SLAs group schedule details.

	Command or Action	Purpose
Step 6	**show ip sla configuration** **Example:** `switch# show ip sla configuration`	(Optional) Displays the IP SLAs configuration details.

Enabling the IP SLAs Random Scheduler

This section describes how to enable the IP SLAs Random Scheduler.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 3	**ip sla group schedule** *group-operation-number operation-id-numbers* **schedule-period** *seconds* [**ageout** *seconds*] [**frequency** [*seconds*| **range** *random-frequency-range*]] [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm*[*:ss*] [*monthday* | *daymonth*] | **pending** | **now** | **after***hh:mm:ss*}] **Example:** `switch(config)# ip sla group schedule 2 1-3 schedule-period 50 frequency range 80-100`	Specifies the scheduling parameters of a group of IP SLAs operations. To enable the random scheduler option, you must configure the **frequency range** *random-frequency-range* keywords and argument.
Step 4	**exit** **Example:** `switch(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying IP SLAs Multiple Operations Scheduling

This section describes how to verify IP SLAs multiple operations scheduling.

Procedure

	Command or Action	Purpose
Step 1	**show ip sla statistics** **Example:** `switch# show ip sla statistics`	(Optional) Displays the IP SLAs operation details.
Step 2	**show ip sla group schedule** **Example:** `switch# show ip sla group schedule`	(Optional) Displays the IP SLAs group schedule details.
Step 3	**show ip sla configuration** **Example:** `switch# show ip sla configuration`	(Optional) Displays the IP SLAs configuration details.

Examples

After you schedule the multiple IP SLAs operations, you can verify the latest operation details using the appropriate **show** commands.

This example shows how to schedule IP SLAs operations 1 through 20 in the operation group 1 with a schedule period of 60 seconds and a life value of 1200 seconds. By default, the frequency is equivalent to the schedule period. In this example, the start interval is 3 seconds (schedule period divided by number of operations).

```
switch (config)# ip sla group schedule 1 1-20 schedule-period 60 life 1200
```

This example shows how to display the details of the scheduled multiple IP SLAs operation:

```
switch# show ip sla group schedule
Group Entry Number: 1
Probes to be scheduled: 1-20
Total number of probes: 20
Schedule period: 60
Group operation frequency: Equals schedule period
Status of entry (SNMP RowStatus): Active
Next Scheduled Start Time: Start Time already passed
Life (seconds): 1200
Entry Ageout (seconds): never
```

This example shows how to display the details of the scheduled multiple IP SLAs operation. The example shows that the IP SLAs operations are multiple scheduled (TRUE).

```
switch# show ip sla config 1
IP SLAs Infrastructure Engine-III
Entry number: 1
Owner:
Tag:
Operation timeout (milliseconds): 5000
```

```
Type of operation to perform: udp-jitter
Target address/Source address: 101.101.101.1/0.0.0.0
Target port/Source port: 5000/0
Type Of Service parameter: 0x0
Request size (ARR data portion): 32
Packet Interval (milliseconds)/Number of packets: 20/10
Verify data: No
Vrf Name: default
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : TRUE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
```

This example shows how to display the latest operation start time of the scheduled multiple IP SLAs operation, when the operations are scheduled at equal intervals:

```
switch# show ip sla statistics | include Latest operation start time
Latest operation start time: *03:06:21.760 UTC Tue Oct 21 2003
Latest operation start time: *03:06:24.754 UTC Tue Oct 21 2003
Latest operation start time: *03:06:27.751 UTC Tue Oct 21 2003
Latest operation start time: *03:06:30.752 UTC Tue Oct 21 2003
Latest operation start time: *03:06:33.754 UTC Tue Oct 21 2003
Latest operation start time: *03:06:36.755 UTC Tue Oct 21 2003
Latest operation start time: *03:06:39.752 UTC Tue Oct 21 2003
Latest operation start time: *03:06:42.753 UTC Tue Oct 21 2003
Latest operation start time: *03:06:45.755 UTC Tue Oct 21 2003
Latest operation start time: *03:06:48.752 UTC Tue Oct 21 2003
Latest operation start time: *03:06:51.753 UTC Tue Oct 21 2003
Latest operation start time: *03:06:54.755 UTC Tue Oct 21 2003
Latest operation start time: *03:06:57.752 UTC Tue Oct 21 2003
Latest operation start time: *03:07:00.753 UTC Tue Oct 21 2003
Latest operation start time: *03:07:03.754 UTC Tue Oct 21 2003
Latest operation start time: *03:07:06.752 UTC Tue Oct 21 2003
Latest operation start time: *03:07:09.752 UTC Tue Oct 21 2003
Latest operation start time: *03:07:12.753 UTC Tue Oct 21 2003
Latest operation start time: *03:07:15.755 UTC Tue Oct 21 2003
Latest operation start time: *03:07:18.752 UTC Tue Oct 21 2003
```

Configuration Example for Scheduling Multiple IP SLAs Operations

This example shows how to schedule IP SLAs operations 1 to 10 in the operation group 1 with a schedule period of 20 seconds. By default, the frequency is equivalent to the schedule period.

```
switch# ip sla group schedule 1 1-10 schedule-period 20
```

This example shows how to display the scheduled multiple IP SLAs operation. The last line in the example indicates that the IP SLAs operations are multiple scheduled (TRUE).

```
switch# show ip sla group schedule
Multi-Scheduling Configuration:
Group Entry Number: 1
Probes to be scheduled: 1-10
Schedule period :20
Group operation frequency: 20
Multi-scheduled: TRUE
```

Configuration Example for Enabling the IP SLAs Random Scheduler

This example shows how to schedule IP SLAs operations 1 to 3 as a group (identified as group 2). In this example, the operations are scheduled to begin at uniformly distributed random intervals over a schedule period of 50 seconds. The first operation is scheduled to start immediately. The interval is chosen from the specified range upon every invocation of the probe. The random scheduler option is enabled and the uniformly distributed random frequencies at which the group of operations will restart is chosen within the range of 80-100 seconds.

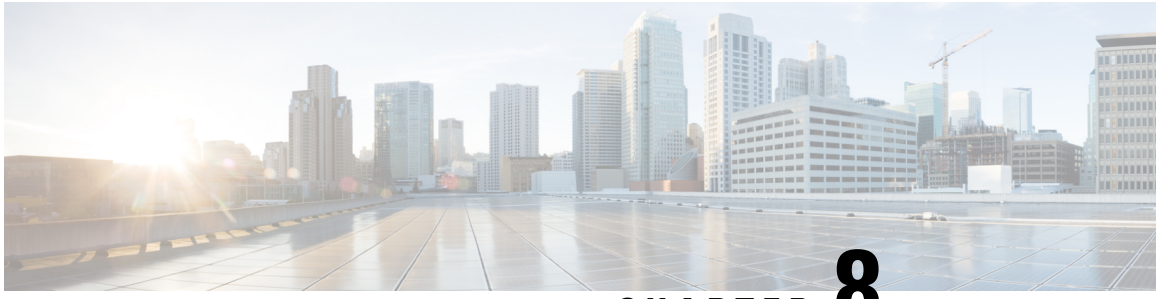
```
ip sla group schedule 2 1-3 schedule-period 50 frequency range 80-100 start-time now
```

Feature History for Multioperation Scheduler

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 13: Feature History for Multioperation Scheduler

Feature Name	Release	Feature Information
Multioperation Scheduler	6.1(1)	This feature was introduced.



CHAPTER 8

IP SLAs TWAMP Responder

The Two-Way Active Measurement Protocol (TWAMP) defines a flexible method for measuring round-trip IP performance between any two devices.

TWAMP enables complete IP performance measurement. TWAMP also provides a flexible choice of solutions because it supports all devices deployed in the network.

This chapter describes how to configure the Two-Way Active Measurement Protocol (TWAMP) responder on a Cisco device to measure IP performance between the Cisco device and a non-Cisco TWAMP control device on your network.

Note IPv6 is supported for IP SLA TWAMP Responder on the RSP3 module.

Prerequisites for TWAMP Responder

For the TWAMP responder to function, a TWAMP control-client and the session-sender must be configured in your network.

Restrictions for TWAMP Responder

For the TWAMP Responder to function, the TWAMP server and the session-reflector must be configured on the same Cisco device.

Information About TWAMP Responder

TWAMP

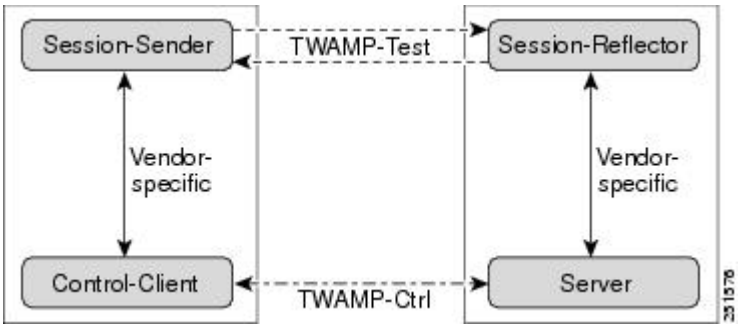
The IETF Two-Way Active Measurement Protocol (TWAMP) defines a standard for measuring round-trip network performance between any two devices that support the TWAMP protocols. The TWAMP-Control protocol is used to set up performance measurement sessions. The TWAMP-Test protocol is used to send and receive performance-measurement probes.

The TWAMP architecture is composed of the following four logical entities that are responsible for starting a monitoring session and exchanging packets:

- The control-client sets up, starts, and stops TWAMP-Test sessions.
- The session-sender instantiates TWAMP-Test packets that are sent to the session-reflector.
- The session-reflector reflects a measurement packet upon receiving a TWAMP-Test packet. The session reflector does not collect packet statistics in TWAMP.
- The TWAMP server is an end system that manages one or more TWAMP sessions and is also capable of configuring per-session ports in the end points. The server listens on the TCP port. The session-refector and server make up the TWAMP responder in an IP SLAs operation.

Although TWAMP defines the different entities for flexibility, it also allows for logical merging of the roles on a single device for ease of implementation. The figure below shows the four entities that make up the TWAMP architecture.

Figure 9: TWAMP Architecture

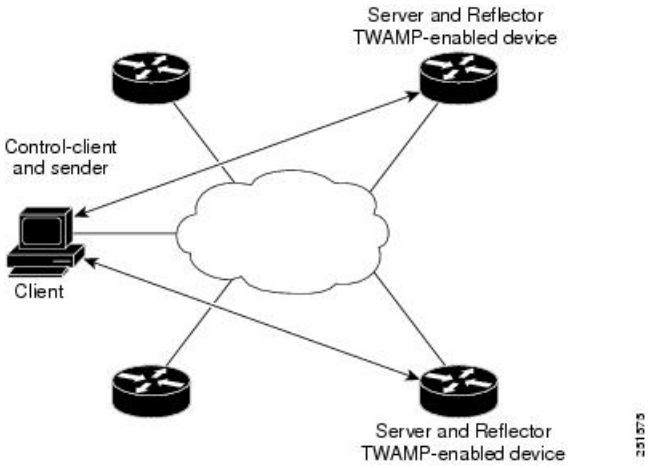


TWAMP Responder

A TWAMP responder interoperates with the control-client and session-sender on another device that supports TWAMP. In the TWAMP Responder feature, the session-reflector and TWAMP server that make up the responder must be co-located on the same device. TWAMP for IPv6 is also supported.

In the figure below, one device is the control-client and session-sender (TWAMP control device), and the other two devices are Cisco devices that are configured as IP SLAs TWAMP responders. Each IP SLAs TWAMP responder is both a TWAMP server and a session-reflector.

Figure 10: TWAMP Responders in a Basic TWAMP Deployment



How to Configure a TWAMP Responder

Configuring the TWAMP Server

Note For IP SLAs TWAMP Responder, the TWAMP server and the session-reflector are configured on the same device.

Procedure

Step 1 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 2 **feature sla twamp-server**

Example:

```
Device(config)#  feature sla twamp-server
```

Enables the TWAMP server part of the SLA.

Step 3 **ip sla server twamp**

Example:

```
Device(config)#  ip sla server twamp
```

Configures the device as a TWAMP server and enters TWAMP server configuration mode.

Step 4 **port** *port-number*

Example:

```
Device(config-twamp-srvr)# port 9000
```

(Optional) Configures the port to be used by the TWAMP server to listen for connection and control requests.

Step 5 **timer inactivity** *seconds*

Example:

```
Device(config-twamp-srvr)# timer inactivity 900
```

(Optional) Configures the inactivity timer for a TWAMP control session. Default inactivity timer is 900 seconds; minimum timer is 1 second; and maximum timer is 6000 seconds.

Step 6 **end**

Example:

```
Device(config-twamp-srvr)# end
```

Returns to privileged EXEC mode.

Configuring the Session-Reflector

Note For TWAMP Responder, the TWAMP server and the session-reflector are configured on the same device.

Procedure

Step 1 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 2 **ip sla responder**

Example:

```
Device(config)# ip sla responder
```

Enables the IP SLA responder for general IP SLAs operations—sending and receiving of IP SLAs control packets.

Step 3 **ip sla responder twamp**

Example:

```
Device(config)# ip sla responder twamp
```

Configures the device as a TWAMP responder and enters TWAMP reflector configuration mode. Enabling the responder allows the generation of packet loss statistics on the device sending IP SLAs operations.

Step 4 **timeout** *seconds*

Example:

```
Device(config-twamp-ref)#  timeout 900
```

(Optional) Configures an inactivity timer for a TWAMP test session. Default inactivity timer is 900 seconds; minimum timer is 1 second; and maximum timer is 604800 seconds.

Step 5 **end**

Example:

```
Device(config-twamp-ref)# end
```

Exits to privileged EXEC mode.

Configuration Examples for TWAMP Responder

TWAMP Responder Example

The following example shows how to configure the TWAMP server and the session-reflector for TWAMP Responder on the same Cisco device. In this configuration, port 862 is the (default) port to be used by the TWAMP server to listen for connection and control requests. The default port for the server listener is the RFC-specified port and can be reconfigured, if required.

Note In order for the TWAMP responder to function, a control-client and the session-sender must be configured in your network.

```
Device# configure terminal
Device(config)#  feature sla twamp-server
Device(config)#  ip sla server twamp
Device(config-twamp-srvr)#  exit
Device(config)#  ip sla responder
Device(config)#  ip sla responder twamp
Device(config-twamp-ref)# end
Device> show running-config
.
.
.
ip sla responder
ip sla responder twamp
ip sla server twamp
```

TWAMP Responder Show Commands Example

```
Device# show ip sla twamp ?
connection Display TWAMP connections
session Display TWAMP Sessions
standards Display TWAMP standards implemented

Device# show ip sla twamp standards
Feature Organization Standard
TWAMP Server IETF RFC5357
TWAMP Reflector IETF RFC5357

Device# show ip sla twamp session
IP SLAs Responder TWAMP is: Enabled
Recvr Addr: 30.30.30.1
Recvr Port: 7147
Sender Addr: 30.30.30.2
Sender Port: 50790
Sender VRF: default
Session Id: 30.30.30.1:15918249420668138422:DF55BEE9
Connection Id: 21

Device# show ip sla twamp connection ?
detail Current Connection Details
requests Current Connection Requests

Device# show ip sla twamp connection detail
Connection Id: 21
Client IP Address: 30.30.30.2
Client Port: 58316
Client VRF: default
Mode: Unauthenticated
Connection State: Connected
Control State: Active
Number of Test Requests - 0:1
```

Additional References

Related Documents

Related Topic	Document Title
IP SLAs commands	Cisco Nexus 7000 Series NX-OS IP SLAs Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 5357.	*Two-Way Active Measurement Protocol (TWAMP)*
RFC 4656	*One-way Active Measurement Protocol (OWAMP)*

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

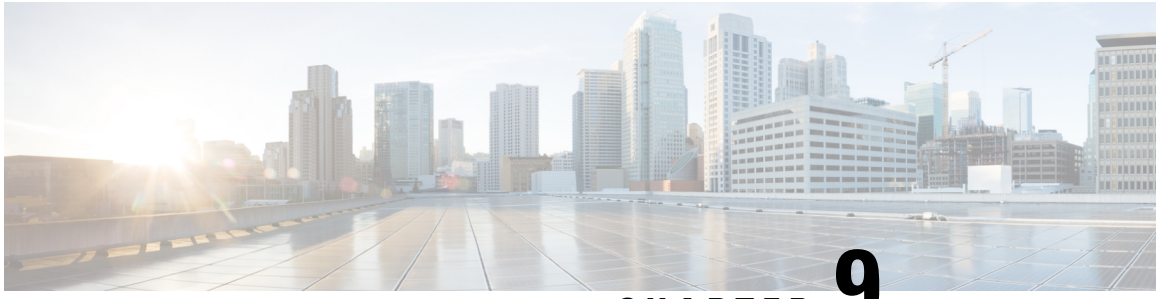
Feature Information for TWAMP Responder

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 14: Feature Information for TWAMP Responder

Feature Name	Releases	Feature Information
TWAMP Responder	Cisco NX-OS Release 8.3(1)	This feature enables you to configure the TWAMP server and the session-reflector on a Cisco device for measuring the round-trip performance between an IP SLA TWAMP responder and a non-Cisco TWAMP control device in your network.



CHAPTER 9

Configuring Proactive Threshold Monitoring for IP SLAs Operations

This chapter describes the proactive monitoring capabilities of IP Service Level Agreements (SLAs) using thresholds and reaction triggering.

This chapter includes the following sections:

Information About IP SLAs Reaction Configuration

IP SLAs reactions are configured to trigger when a monitored value exceeds or falls below a specified level or when a monitored event, such as a timeout or connection loss, occurs. If IP SLAs measure too high or too low of any configured reaction, IP SLAs can generate a notification to a network management application or trigger another IP SLA operation to gather more data.

IP SLAs Threshold Monitoring and Notifications

IP SLAs support proactive threshold monitoring and notifications for performance parameters such as average jitter, unidirectional latency, bidirectional round-trip time (RTT), and connectivity for most IP SLAs operations. The proactive monitoring capability also provides options for configuring reaction thresholds for important VoIP related parameters including unidirectional jitter, unidirectional packet loss, and unidirectional VoIP voice quality scoring.

Notifications for IP SLAs are configured as a triggered reaction. Packet loss, jitter, and Mean Operation Score (MOS) statistics are specific to IP SLAs jitter operations. Notifications can be generated for violations in either direction (source-to-destination and destination-to-source) or for out-of-range RTT values for packet loss and jitter. Events, such as traps, are triggered when the RTT value rises above or falls below a specified threshold.

IP SLAs can generate system logging (syslog) messages when a reaction condition occurs. System logging messages can be sent as Simple Network Management Protocol (SNMP) traps (notifications) using the CISCO-RTTMON-MIB. SNMP traps for IP SLAs are supported by the CISCO-RTTMON-MIB and CISCO-SYSLOG-MIB.

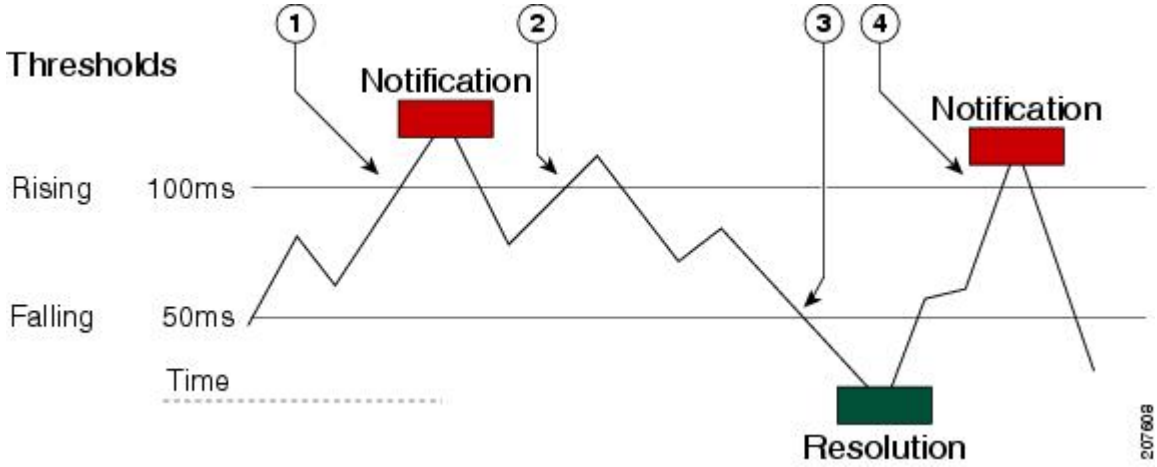
Severity levels in the CISCO-SYSLOG-MIB are SyslogSeverity INTEGER {emergency(1), alert(2), critical(3), error(4), warning(5), notice(6), info(7), debug(8)}.

The values for severity levels are defined differently for the system logging process in the Cisco NX-OS software. Severity levels for the system logging process in the Cisco NX-OS software are: {emergency (0), alert (1), critical (2), error (3), warning (4), notice (5), informational (6), debugging (7)}.

IP SLAs threshold violations are logged as level 6 (informational) within the Cisco NX-OS system logging process but are sent as level 7 (info) traps from the CISCO-SYSLOG-MIB.

Notifications are not issued for every occurrence of a threshold violation. The following figure shows the sequence for a triggered reaction that occurs when the monitored element exceeds the upper threshold. An event is sent and a notification is issued when the rising threshold is exceeded for the first time. Subsequent threshold-exceeded notifications are issued only after the monitored value falls below the falling threshold before exceeding the rising threshold again.

Figure 11: IP SLAs Triggered Reaction Condition and Notifications for Threshold Exceeded



1	An event is sent and a threshold-exceeded notification is issued when the rising threshold is exceeded for the first time.
2	Consecutive over-rising threshold violations occur without issuing additional notifications.
3	The monitored value goes below the falling threshold.
4	Another threshold-exceeded notification is issued when the rising threshold is exceeded only after the monitored value first fell below the falling threshold.

Note A lower-threshold notification is also issued the first time that the monitored element falls below the falling threshold (3). Subsequent notifications for lower-threshold violations are issued only after the rising threshold is exceeded before the monitored value falls below the falling threshold again.

RTT Reactions for Jitter Operations

RTT reactions for jitter operations are triggered only at the end of the operation and use the latest value for the return-trip time (LatestRTT), which matches the value of the average return-trip time (RTTAvg).

SNMP traps for RTT for jitter operations are based on the value of the average return-trip time (RTTAvg) for the whole operation and do not include RTT values for each individual packet sent during the operation. For example, if the average is below the threshold, up to half of the packets can actually be above the threshold, but this detail is not included in the notification because the value is for the whole operation only.

Only syslog messages are supported for RTTAvg threshold violations. Syslog nmessages are sent from the CISCO-RTTMON-MIB.

Configuring Proactive Threshold Monitoring

This section describes how to configure thresholds and reactive triggering for generating traps or starting another operation.

Before you begin

- Configure IP SLAs operations to be started when violation conditions are met.

Note

- RTT reactions for jitter operations are triggered only at the end of the operation and use the latest value for the return-trip time (LatestRTT).
- SNMP traps for RTT for jitter operations are based on the average value for the return-trip time (RTTAvg) for the whole operation only and do not include return-trip time values for individual packets sent during the operation. Only syslog messages are supported for RTTAvg threshold violations.
- Only syslog messages are supported for RTT violations during jitter operations.
- Only SNMP traps are supported for RTT violations during nonjitter operations.
- Only syslog messages are supported for non-RTT violations other than timeout, connectionLoss, or verifyError.
- Both SNMP traps and syslog messages are supported for timeout, connectionLoss, or verifyError violations only.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip sla reaction-configuration** *operation-number* **react** *monitored-element* [**action-type** *option*] [**threshold-type** {**average** [*number-of-measurements*] | **consecutive** [*occurrences*] | **immediate** | **never** | **xofy** [*x-valuey-value*]}] [**threshold-value** *upper-thresholdlower-threshold*] **Example:** `switch(config)# ip sla reaction-configuration 10 react jitterAvg threshold-type immediate threshold-value 5000 3000 action-type trapAndTrigger`	Configures the action (SNMP trap or IP SLAs trigger) that is to occur based on violations of specified thresholds.
Step 4	**ip sla reaction-trigger** *operation-number target-operation* **Example:** `switch(config)# ip sla reaction-trigger 10 2`	(Optional) Starts another IP SLAs operation when the violation conditions are met. Required only if the **ip sla reaction-configuration** command is configured with either the **trapAndTrigger** or **triggerOnly** keyword.
Step 5	**ip sla logging traps** **Example:** `switch(config)# ip sla logging traps`	(Optional) Enables IP SLAs syslog messages from CISCO-RTTMON-MIB.
Step 6	**snmp-server enable traps ip sla** **Example:** `switch(config)# snmp-server enable traps ip sla`	(Optional) Enables system to generate CISCO-RTTMON-MIB traps.
Step 7	**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:** `switch(config)# snmp-server host 10.1.1.1 public`	(Optional) Sends traps to a remote host. Required if the **snmp-server enable traps** command is configured.
Step 8	**exit** **Example:** `switch(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 9	**show ip sla reaction configuration** [*operation-number*] **Example:** `switch# show ip sla reaction configuration 10`	(Optional) Displays the configuration of proactive threshold monitoring.

	Command or Action	Purpose
Step 10	**show ip sla reaction trigger** [*operation-number*] **Example:** `switch# show ip sla reaction trigger 2`	(Optional) Displays the configuration status and operational state of target operations to be triggered.

Configuration Example for an IP SLAs Reaction Configuration

This example shows how to configure IP SLAs operation 10 to send an SNMP logging trap when the MOS value either exceeds 4.9 (best quality) or falls below 2.5 (poor quality):

```
switch(config)# ip sla reaction-configuration 10 react mos threshold-type immediate
threshold-value 490 250 action-type trapOnly
```

This example shows how to display the default configuration:

```
switch# show ip sla reaction-configuration 1
Entry number: 1
Index: 1
Reaction: mos
Threshold Type: Immediate
Rising: 490
Falling: 250
Action Type: Trap only
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip sla reaction-configuration 10 react mos threshold-type immediate
threshold-value 490 250 action-type trapOnly
switch(config)# show ip sla reaction-configuration 1
Entry number: 1
Reaction: rtt
Threshold Type: Never
Rising (milliseconds): 5000
Falling (milliseconds): 3000
Threshold Count: 5
Threshold Count2: 5
Action Type: None
```

Verification Example for an IP SLAs Reaction Configuration

This example shows that multiple monitored elements are configured for the IP SLAs operation (1), as indicated by the values of Reaction: in the output:

```
switch# show ip sla reaction-configuration

Entry Number: 1
Reaction: RTT
Threshold type: Never
Rising (milliseconds): 5000
Falling (milliseconds): 3000
Threshold Count: 5
Threshold Count2: 5
```

```
Action Type: None
Reaction: jitterDSAvg
Threshold type: average
Rising (milliseconds): 5
Falling (milliseconds): 3
Threshold Count: 5
Threshold Count2: 5
Action Type: triggerOnly
Reaction: jitterDSAvg
Threshold type:  immediate
Rising (milliseconds): 5
Falling (milliseconds): 3
Threshold Count: 5
Threshold Count2: 5
Action Type: trapOnly
Reaction: PacketLossSD
Threshold type:  immediate
Rising (milliseconds): 5
Threshold Falling (milliseconds): 3
Threshold Count: 5
Threshold Count2: 5
Action Type: trapOnly
```

Configuration Example for Triggering SNMP Notifications

This example shows how to configure proactive threshold monitoring so that CISCO-SYSLOG-MIB traps are sent to the remote host at 10.1.1.1 if the threshold values for RTT or VoIP MOS are violated:

```
! Configure the operation on source.
switch(config)# ip sla 1

switch(config-ip-sla)# udp-jitter 10.1.1.1 3000 codec g711alaw
switch(config-ip-sla-jitter)# exit

switch(config)# ip sla schedule 1 start now life forever

! Configure thresholds and reactions.
switch(config)# ip sla reaction-configuration 1 react rtt threshold-type immediate
threshold-value 3000 2000 action-type trapOnly

switch(config)# ip sla reaction-configuration 1 react MOS threshold-type consecutive 4
threshold-value 390 220 action-type trapOnly

switch(config)# ip sla logging traps

! The following command sends traps to the specified remote host.
switch(config)# snmp-server host 10.1.1.1 version 2c public

! The following command is needed for the system to generate CISCO-SYSLOG-MIB traps.
switch(config)# snmp-server enable traps
```

This example shows that IP SLAs threshold violation notifications are generated as level 6 (informational) in the Cisco NX-OS system logging process:

```
3d18h:%RTT-6-SAATHRESHOLD:RTR(11):Threshold exceeded for MOS
```

This example shows an SNMP notification from the CISCO-SYSLOG-MIB for the same violation is a level 7 (info) notification:

```
3d18h:SNMP:V2 Trap, reqid 2, errstat 0, erridx 0
 sysUpTime.0 = 32613038
 snmpTrapOID.0 = ciscoSyslogMIB.2.0.1
 clogHistoryEntry.2.71 = RTT
 clogHistoryEntry.3.71 = 7
 clogHistoryEntry.4.71 = SAATHRESHOLD
 clogHistoryEntry.5.71 = RTR(11):Threshold exceeded for MOS
 clogHistoryEntry.6.71 = 32613037
```

Feature History for Proactive Threshold Monitoring

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 15: Feature History for Proactive Threshold Monitoring

Feature Name	Release	Feature Information
Proactive Threshold Monitoring	6.1(1)	This feature was introduced.

CHAPTER 10

Configuring IP SLA PBR Object Tracking

This chapter describes the PBR object tracking capabilities of IP Service Level Agreements (SLAs).

This chapter includes the following sections:

IP SLA PBR Object Tracking

This feature allows you to make sure that the next hop is reachable before that route is used. If the next hop is not reachable, another route is used as defined in the policy-based routing (PBR) configuration. If no other route is present in the route map, the routing table is used.

Object Tracking

Object tracking monitors objects such as the following:

- State of the line protocol of an interface
- Existence of an entry in the routing table

Clients, such as PBR, can register their interest in specific, tracked objects and then take action when the state of the objects changes.

IP SLA PBR Object Tracking Overview

The PBR Object Tracking feature gives policy-based routing (PBR) access to all the objects that are available through the tracking process. The tracking process enables you to track individual objects—such as ICMP ping reachability, routing adjacency, an application running on a remote device, a route in the Routing Information Base (RIB)—or to track the state of an interface line protocol.

Object tracking functions in the following manner: PBR informs the tracking process that a certain object should be tracked, and the tracking process then notifies PBR when the state of that object changes.

Configuring IP SLA PBR Object Tracking

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**ip sla** *operation-number* **Example:** `switch(config)# ip sla 1`	Starts a Cisco IOS IP Service Level Agreement (SLA) operation configuration and enters IP SLA configuration mode.
Step 3	**icmp-echo** *destination-ip-address* **Example:** `switch(config-ip-sla)# icmp-echo 10.3.3.2`	Configures an IP SLA Internet Control Message Protocol (ICMP) echo probe operation.
Step 4	**exit** **Example:** `switch(config-ip-sla)# exit`	Exits IP SLA configuration mode and returns the router to global configuration mode.
Step 5	**ip sla schedule** *operation-number* **life forever start-time now** **Example:** `switch(config)# ip sla schedule 1 life forever start-time now`	Configures the scheduling parameters for a single Cisco IOS IP SLA operation. • In this example, the time parameters for the IP SLA operation are configured. **Note** Repeat Steps 2 to 5 to configure and schedule other IP SLA operations.
Step 6	**track** *object-number* **ip sla** *entry-number* **reachability** **Example:** `switch(config)# track 1 ip sla 1 reachability`	Tracks the reachability of an object and enters tracking configuration mode. **Note** Repeat this step to track other operations.
Step 7	**exit** **Example:** `switch(config-track)# exit`	Exits tracking configuration mode and returns the router to global configuration mode.

	Command or Action	Purpose
Step 8	**ip access-list standard** *access-list-name* **Example:** `switch(config)# ip access-list standard ACL`	Defines an IP access list access control list (ACL) in order to enable filtering for packets.
Step 9	**permit ip** *source destination* **Example:** `switch(config-acl)# permit ip 192.0.2.0/24 198.51.100.0/24`	Creates an access control list (ACL) rule that permits traffic matching its conditions.
Step 10	**ipv6 access-list** *access-list-name* **Example:** `switch(config)# ipv6 access-list IPv6ACL`	Defines an IPv6 access list ACL in order to enable filtering for packets.
Step 11	**permit ipv6** *source destination* **Example:** `switch(config-ipv6-acl)# permit ipv6 2001:DB8::/32 2001:DB8::/48`	Creates an access control list (ACL) rule that permits traffic matching its conditions.
Step 12	**exit** **Example:** `switch(config-ipv6-acl)# exit`	Exits ACL configuration mode and returns the router to global configuration mode.
Step 13	**route-map** *map-tag* **Example:** `switch(config)# route-map PBR`	Specifies a route map and enters route-map configuration mode.
Step 14	**match ip address** *access-list-name* **Example:** `switch(config-route-map)# match ip address ACL`	Distributes any routes that have a destination IPv4 network number address that is permitted by a standard access list.
Step 15	**match ipv6 address** *access-list-name* **Example:** `switch(config-route-map)# match ipv6 address IPv6ACL`	Distributes any routes that have a destination IPv6 network number address that is permitted by a standard access list.
Step 16	**set ip next-hop verify-availability** *next-hop-address* **track** *object*	Configures the route map to verify the reachability of the tracked object.

	Command or Action	Purpose
	Example: `switch(config-route-map)# set ip next-hop verify-availability 198.51.100.2 track 1`	**Note** Repeat this step to configure the route map to verify the reachability of other tracked objects.
Step 17	**set ipv6 next-hop verify-availability** *next-hop-address* **track** *object* **Example:** `switch(config-route-map)# set ipv6 next-hop verify-availability 2001:DB8:1::1 track 1`	Configures the route map to verify the reachability of the tracked object. **Note** Repeat this step to configure the route map to verify the reachability of other tracked objects.
Step 18	**set ip default next-hop verify-availability** *next-hop-address* **track** *object* **Example:** `switch(config-route-map)# set ip default next-hop verify-availability 192.0.2.2 track 1`	Configures the route map to verify the reachability of the default next hop.
Step 19	**set ipv6 default next-hop verify-availability** *next-hop-address* **track** *object* **Example:** `switch(config-route-map)# set ipv6 default next-hop verify-availability 2001:DB8:0:ABCD::1 track 1`	Configures the route map to verify the reachability of the default next hop.
Step 20	**exit** **Example:** `switch(config-route-map)# exit`	Exits route-map configuration mode and returns the router to global configuration mode.
Step 21	**interface** *type number* **Example:** `switch(config)# interface ethernet 0/0`	Specifies an interface type and number and enters interface configuration mode.
Step 22	**ip address** *ip-address mask* **Example:** `switch(config-if)# ip address 10.2.2.1 255.255.255.0`	Specifies a primary IP address for an interface.
Step 23	**ipv6 address** *ip-address mask* **Example:**	Specifies a primary IPv6 address for an interface.

	Command or Action	Purpose
	`switch(config-if)# ipv6 address 2001:DB8::/48`	
Step 24	**ip policy route-map** *map-tag* **Example:** `switch(config-if)# ip policy route-map PBR`	Enables policy routing and identifies a route map to be used for policy routing.
Step 25	**ipv6 policy route-map** *map-tag* **Example:** `switch(config-if)# ipv6 policy route-map PBR`	Enables IPv6 policy routing and identifies a route map to be used for policy routing.
Step 26	**end** **Example:** `switch(config-if)# end`	Exits interface configuration mode and returns the router to privileged EXEC mode.
Step 27	**show track** *object-number* **Example:** `switch# show track 1`	(Optional) Displays tracking information. Use this command to verify the configuration.
Step 28	**show route-map** *map-name* **Example:** `switch# show route-map PBR`	(Optional) Displays route map information.

Example: Configuring IP SLA PBR Object Tracking

This example shows that object tracking is configured for PBR:

```
! Configure and schedule IP SLA operations
ip sla 1
 icmp-echo 10.3.3.2
ip sla schedule 1 life forever start-time now
!
ip sla 2
 udp-echo 10.4.4.2
ip sla schedule 2 life forever start-time now
!
ip sla 3
 icmp-echo 10.5.5.2
ip sla schedule 3 life forever start-time now
!
ip sla 4
 icmp-echo 10.6.6.2
```

```
ip sla schedule 4 life forever start-time now
!
ip sla 5
 icmp-echo 10.7.7.2
ip sla schedule 5 life forever start-time now
!
! Configure Object Tracking to track the operations
!
track 1 ip sla 1 reachability
track 2 ip sla 2 reachability
track 3 ip sla 3 reachability
track 4 ip sla 4 reachability
track 5 ip sla 5 reachability
!
! Configure ACL
ip access-list standard ACL
 permit ip 10.2.2.0/24 10.1.1.1/32
!
! Configure PBR policing on the router
route-map PBR
 match ip address ACL
 set ip next-hop verify-availability 10.3.3.2 track 1
 set ip next-hop verify-availability 10.4.4.2 track 2
 set ip next-hop verify-availability 10.5.5.2 track 3
!
! Apply PBR policy on the incoming interface of the router.
interface ethernet 0/0
 ip address 10.2.2.1 255.255.255.0
  ip policy route-map PBR
!
! Display PBR related information
show route-map
show track brief
show ip sla stat
show ip sla application
!
```

Feature History for IP SLA PBR Object Tracking

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 16: Feature History for IP SLA PBR Object Tracking

Feature Name	Release	Feature Information
IP SLA PBR Object Tracking	6.2(2)	This feature was introduced.

CHAPTER 11

Configuring IP SLAs DNS Operations

This chapter describes the DNS operations capabilities of IP Service Level Agreements (SLAs).

This chapter includes the following sections:

IP SLAs DNS Operations

This section describes how to configure the IP SLAs DNS operations to measure the difference between the time taken to send a DNS request and receive a reply.

Guidelines and Limitations for IP SLA DNS Operations

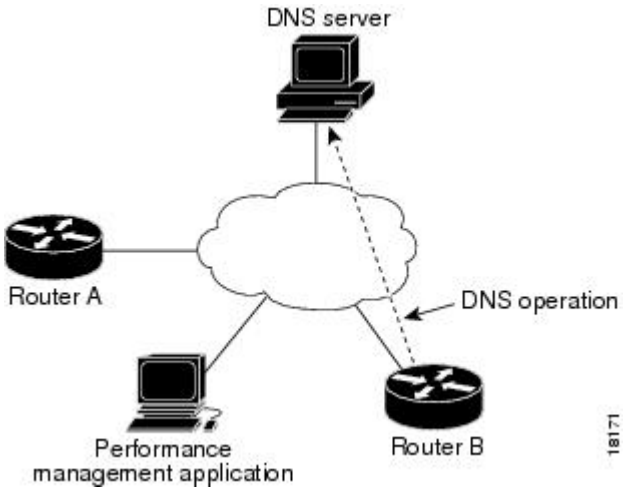
- IPv6 is not supported for IP SLA DNS operations.

DNS Operation

The DNS operation measures the difference between the time taken to send a DNS request and receive a reply. DNS is used in the Internet for translating names of network nodes into addresses. The IP SLAs DNS operation queries for an IP address if you specify a hostname or queries for a hostname if you specify an IP address.

In the following figure, Device B is configured as the source IP SLAs device and a DNS operation is configured with the DNS server as the destination device.

Figure 12: DNS Operation



The connection response time is computed by measuring the difference between the time taken to send a request to the DNS server and the time a reply is received by Device B. The resulting DNS lookup time can help you analyze your DNS performance. Faster DNS lookup times translate to a faster web server access experience.

Configuring a Basic DNS Operation on the Source Device

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**ip sla** *operation-number* **Example:** `switch(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 3	**dns** {*destination-ip-address* | *destination-hostname*} **name-server** *ip-address* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] **Example:** `switch(config-ip-sla)# dns host1 name-server 172.20.2.132`	Defines a DNS operation and enters IP SLA DNS configuration mode.

	Command or Action	Purpose
Step 4	**frequency** *seconds* **Example:** `switch(config-ip-sla-dns)# frequency 60`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 5	**end** **Example:** `switch(config-ip-sla-dns)# end`	Exits to privileged EXEC mode.

Configuring a DNS Operation with Optional Parameters on the Source Device

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**ip sla** *operation-number* **Example:** `switch(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 3	**dns** {*destination-ip-address* | *destination-hostname*} **name-server** *ip-address* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] **Example:** `switch(config-ip-sla)# dns host1 name-server 172.20.2.132`	Defines a DNS operation and enters IP SLA DNS configuration mode.
Step 4	**history buckets-kept** *size* **Example:** `switch(config-ip-sla-dns)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.

	Command or Action	Purpose
Step 5	**history distributions-of-statistics-kept** *size* **Example:** `switch(config-ip-sla-dns)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 6	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `switch(config-ip-sla-dns)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 7	**frequency** *seconds* **Example:** `switch(config-ip-sla-dns)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 8	**history hours-of-statistics-kept** *hours* **Example:** `switch(config-ip-sla-dns)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 9	**history lives-kept** *lives* **Example:** `switch(config-ip-sla-dns)# history lives-kept 2`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 10	**owner** *owner-id* **Example:** `switch(config-ip-sla-dns)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 11	**history statistics-distribution-interval** *milliseconds* **Example:** `switch(config-ip-sla-dns)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 12	**tag** *text* **Example:** `switch(config-ip-sla-dns)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.

	Command or Action	Purpose
Step 13	**threshold** *milliseconds* **Example:** `switch(config-ip-sla-dns)# threshold 9000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 14	**timeout** *milliseconds* **Example:** `switch(config-ip-sla-dns)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 15	**end** **Example:** `switch(config-ip-sla-dns)# end`	Exits to privileged EXEC mode.

Scheduling IP SLAs Operations

Note

- All IP SLAs operations that you want to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group is limited to a maximum of 125 characters in length, including commas (,).

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	Use one of the following. • **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**] • **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* |	Configures the scheduling parameters for an individual IP SLAs operation. Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

	Command or Action	Purpose

	schedule-together} [**ageout** *seconds*] [**frequency** *group-operation-frequency*] [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm*[:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*[:*ss*]}]
	Example:
	`switch(config)# ip sla schedule 10 life forever start-time now`
	Example:
	`switch(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`	

Step 3	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 4	**show ip sla group schedule** **Example:** `switch# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 5	**show ip sla configuration** **Example:** `switch# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Configuration Example for a DNS Operation

This example shows how to configure a DNS operation from Device B to the DNS server (IP address 172.20.2.132) as shown in the “DNS Operation” figure in the “DNS Operation” section. The operation is scheduled to start immediately. In this example, the target address is a hostname and the DNS operation will query the DNS server for the IP address associated with the hostname host1. No configuration is required at the DNS server.

```
ip sla 11
 dns host1 name-server 172.20.2.132
 frequency 50
 timeout 8000
 tag DNS-Test
ip sla schedule 11 start-time now
```

Configuration Example for a Basic DNS Operation on the Source Device

This example shows how to configure a basic DNS operation on the source device:

```
switch# configure terminal
switch(config)# ip sla 10
switch(config-ip-sla)# dns host1 name-server 172.20.2.132
switch(config-ip-sla-dns)# frequency 60
switch(config-ip-sla-dns)# end
```

Configuration Example for a DNS Operation with Optional Parameters on the Source Device

This example shows how to configure a DNS operation with optimal parameters on the source device:

```
switch# configure terminal
switch(config)# ip sla 10
switch(config-ip-sla)# dns host1 name-server 172.20.2.132
switch(config-ip-sla-dns)# history buckets-kept 25
switch(config-ip-sla-dns)# history distributions-of-statistics-kept 5
switch(config-ip-sla-dns)# history filter failures
switch(config-ip-sla-dns)# frequency 30
switch(config-ip-sla-dns)# history hours-of-statistics-kept 4
switch(config-ip-sla-dns)# history lives-kept 2
switch(config-ip-sla-dns)# owner admin
switch(config-ip-sla-dns)# history statistics-distribution-interval 10
switch(config-ip-sla-dns)# tag TelnetPollServer1
switch(config-ip-sla-dns)# threshold 9000
switch(config-ip-sla-dns)# timeout 10000
switch(config-ip-sla-dns)# end
```

Configuration Example for Scheduling IP SLAs Operations

This example shows how to schedule IP SLAs operations:

```
switch# configure terminal
switch(config)# ip sla schedule 10 life forever start-time now
switch(config)# exit
switch# show ip sla group schedule
switch# show ip sla configuration
```

Feature History for IP SLAs DNS Operations

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 17: Feature History for IP SLAs DNS Operations

Feature Name	Release	Feature Information
IP SLAs DNS Operations	6.2(2)	This feature was introduced.

CHAPTER 12

Configuring IP SLAs ICMP Echo Operations

This module describes how to configure an IP Service Level Agreements (SLAs) Internet Control Message Protocol (ICMP) Echo operation to monitor end-to-end response time between two devices using IPv4 or IPv6. ICMP Echo is useful for troubleshooting network connectivity issues. This module also demonstrates how the results of the ICMP Echo operation can be displayed and analyzed to determine how the network IP connections are performing.

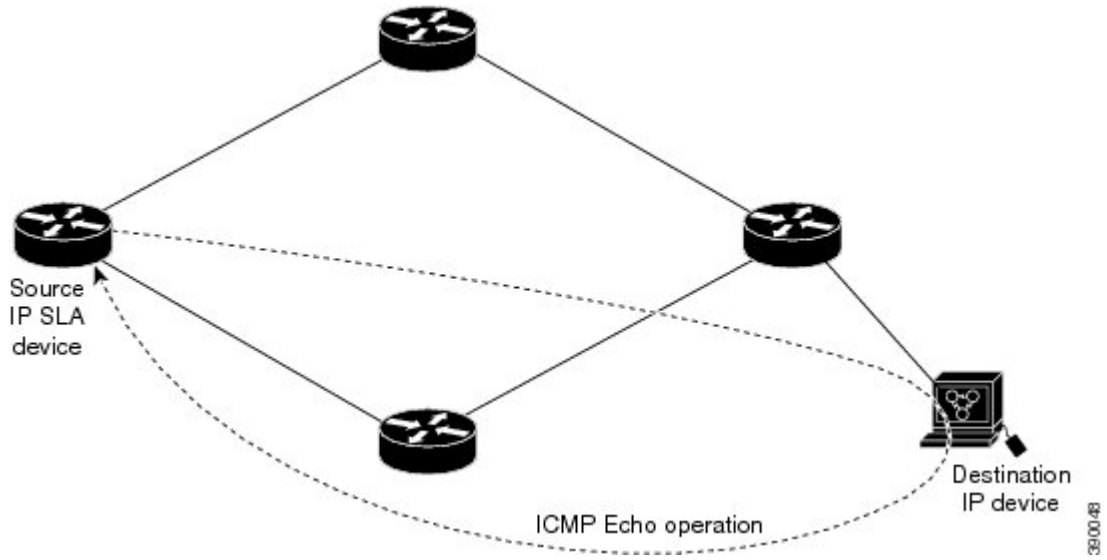
This chapter includes the following sections:

ICMP Echo Operation

The Internet Control Message Protocol (ICMP) Echo operation measures the end-to-end response time between two devices that use IP. The response time is computed by measuring the time taken between sending an ICMP Echo request message to the destination and receiving an ICMP Echo reply. An ICMP Echo is useful for troubleshooting network connectivity issues. The results of the ICMP Echo operation can be displayed and analyzed to determine how the network IP connections are performing.

In the following figure, the ICMP Echo operation uses the ping test to measure the response time between the source IP SLAs device and the destination IP device. Many customers use IP SLAs ICMP-based operations, in-house ping testing, or ping-based dedicated probes for response time measurements.

Figure 13: ICMP Echo Operation



The IP SLAs ICMP Echo operation conforms to the same IETF specifications for ICMP ping testing and the two methods result in the same response times.

Guidelines and Limitations for IP SLAs ICMP Echo Operations

We recommend that you use a Cisco networking device as the destination device although you can use any networking device that supports RFC 862, the Echo protocol.

Configuring an ICMP Echo Operation

Note You do not need to configure an IP SLAs Responder on the destination device.

Perform one of the following tasks:

- Configuring a basic ICMP Echo operation on the source device
- Configuring an ICMP Echo operation with optional parameters

Configuring a Basic ICMP Echo Operation on a Source Device

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal`	
Step 2	**feature sla sender** **Example:** `switch(config)# feature sla sender`	Enables the IP SLAs operation feature.
Step 3	**ip sla** *operation-number* **Example:** `switch(config)# ip sla 6`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.

Step 4	**icmp-echo** {*destination-ipv4-address* | *destination-ipv6-address* | *destination-hostname*} [**source-ip** {*ipv4-address* | *ipv6-address* | *hostname*} | **source-interface** *interface-name*]
	Example:
	`switch(config-ip-sla)# icmp-echo 192.0.2.134`
	Example:
	`switch(config-ip-sla)# icmp-echo 2016:1:1:1::2`
		Defines an ICMP Echo operation and enters IP SLA ICMP Echo configuration mode.
		Note IPv6 is available only from Cisco NX-OS Release 8.0 onwards.

Step 5	**end** **Example:** `switch(config-ip-sla-echo)# end`	Exits IP SLA ICMP Echo configuration mode and returns to privileged EXEC mode.

Configuring an ICMP Echo Operation with Optional Parameters

Before you begin

Perform this task on the source device.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature sla sender** **Example:** `switch(config)# feature sla sender`	Enables the IP SLAs operation feature.
Step 3	**ip sla** *operation-number* **Example:**	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.

	Command or Action	Purpose
	`switch(config)# ip sla 6`	

Step 4	**icmp-echo** {*destination-ipv4-address* | *destination-ipv6-address* | *destination-hostname*} [**source-ip** {*ipv4-address* | *ipv6-address* | *hostname*} | **source-interface** *interface-name*]
	Example:
	`switch(config-ip-sla)# icmp-echo 192.0.2.134 source-ip 192.0.2.132`
	Example:
	`switch(config-ip-sla)# icmp-echo 2016:1:1:1::2 source-ip 2016:1:1:1::2`
		Defines an Echo operation and enters IP SLA Echo configuration mode.
		Note IPv6 is available only from Cisco NX-OS Release 8.0 onwards.

Step 5	(Optional) **history buckets-kept** *size* **Example:** `switch(config-ip-sla-echo)# history buckets-kept 25`	Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 6	(Optional) **history distributions-of-statistics-kept** *size* **Example:** `switch(config-ip-sla-echo)# history distributions-of-statistics-kept 5`	Sets the number of statistics distributions that are kept per hop during an IP SLAs operation.
Step 7	(Optional) **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets* **Example:** `switch(config-ip-sla-echo)# history enhanced interval 900 buckets 100`	Enables enhanced history gathering for an IP SLAs operation.
Step 8	(Optional) **history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `switch(config-ip-sla-echo)# history filter failures`	Defines the type of information kept in the history table for an IP SLAs operation.
Step 9	(Optional) **frequency** *seconds* **Example:** `switch(config-ip-sla-echo)# frequency 30`	Sets the rate at which a specified IP SLAs operation repeats.
Step 10	(Optional) **history hours-of-statistics-kept** *hours* **Example:** `switch(config-ip-sla-echo)# history hours-of-statistics-kept 4`	Sets the number of hours for which statistics are maintained for an IP SLAs operation.

	Command or Action	Purpose
Step 11	(Optional) **history lives-kept** *lives* **Example:** `switch(config-ip-sla-echo)# history lives-kept 5`	Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 12	(Optional) **owner** *owner-id* **Example:** `switch(config-ip-sla-echo)# owner admin`	Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 13	(Optional) **request-data-size** *bytes* **Example:** `switch(config-ip-sla-echo)# request-data-size 64`	Sets the protocol data size in the payload of an IP SLAs operation's request packet.
Step 14	(Optional) **history statistics-distribution-interval** *milliseconds* **Example:** `switch(config-ip-sla-echo)# history statistics-distribution-interval 10`	Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 15	(Optional) **tag** *text* **Example:** `switch(config-ip-sla-echo)# tag TelnetPollServer1`	Creates a user-specified identifier for an IP SLAs operation.
Step 16	(Optional) **threshold** *milliseconds* **Example:** `switch(config-ip-sla-echo)# threshold 10000`	Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 17	(Optional) **timeout** *milliseconds* **Example:** `switch(config-ip-sla-echo)# timeout 10000`	Sets the amount of time an IP SLAs operation waits for a response from its request packet.

Step 18	(Optional) Depending on the IP version you are using, use one of the following commands:
	• **tos** *number*
	• **traffic-class** *number*
	Example:
	`switch(config-ip-sla-echo)# tos 160`
	Example:
	`switch(config-ip-sla-echo)# traffic-class 160`
		In an IPv4 network, defines the ToS byte in the IPv4 header of an IP SLA operation.
		or
		In an IPv6 network, defines the traffic class byte in the IPv6 header for a supported IP SLA operation.
		Note IPv6 is available only from Cisco NX-OS Release 8.0 onwards.

	Command or Action	Purpose
Step 19	(Optional) **verify-data** **Example:** `switch(config-ip-sla-echo)# verify-data`	Causes an IP SLAs operation to check each reply packet for data corruption.
Step 20	(Optional) **vrf** {*vrf-name* | **default** | **management**} **Example:** `switch(config-ip-sla-echo)# vrf vpn-A`	Allows monitoring within Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) using IP SLAs operations.
Step 21	**end** **Example:** `switch(config-ip-sla-echo)# end`	Exits IP SLA Echo configuration mode and returns to privileged EXEC mode.

Scheduling IP SLAs Operations

Note

- All IP SLAs operations that you want to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group is limited to a maximum of 125 characters in length, including commas (,).

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	Perform one of the following tasks: • **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}] [**ageout** *seconds*] [*recurring*] • **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] [**frequency** *group-operation-frequency*] [**life** {**forever** | *seconds*}] [**start-time**	Configures the scheduling parameters for an individual IP SLAs operation. Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

	Command or Action	Purpose

	{*hh:mm*[*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm*[*:ss*]}]
	Example:
	`switch(config)# ip sla schedule 10 life forever start-time now`
	Example:
	`switch(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`	

Step 3	**exit** **Example:** `switch(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 4	**show ip sla group schedule** **Example:** `switch# show ip sla group schedule`	Displays IP SLAs group schedule details.
Step 5	**show ip sla configuration** **Example:** `switch# show ip sla configuration`	Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP SLAs operation is not running and not generating statistics, add the **verify-data** command to the configuration of the operation (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering to an IP Service Level Agreements (SLAs) operation for the purpose of generating traps or for starting another operation, see the “Configuring Proactive Threshold Monitoring” section in the “Configuring Proactive Threshold Monitoring for IP SLAs Operations” chapter.

Configuration Examples for IP SLA ICMP Echo Operations

Note IPv6 is available only from Cisco NX-OS Release 8.0 onwards.

Example: Configuring a Basic ICMP Echo Operation on a Source Device

This example shows how to configure a basic ICMP Echo operation on a source device using IPv4:

```
switch# configure terminal
switch(config)# feature sla sender
switch(config)# ip sla 6
switch(config-ip-sla)# icmp-echo 192.0.2.134 source-ip 192.0.2.132
switch(config-ip-sla-echo)# end
```

This example shows how to configure a basic ICMP Echo Operation on a source device using IPv6:

```
switch# configure terminal
switch(config)# feature sla sender
switch(config)# ip sla 6
switch(config-ip-sla)# icmp-echo 2016:1:1:1::2 source-ip 2016:1:1:1::2
switch(config-ip-sla-echo)# end
```

Example: Configuring an ICMP Echo Operation with Optional Parameters

This example shows how to configure an IP SLAs operation type of ICMP Echo using IPv4 that will start immediately and run indefinitely:

```
switch# configure terminal
switch(config)# feature sla sender
switch(config)# ip sla 6
switch(config-ip-sla)# icmp-echo 192.0.2.134 source-ip 192.0.2.132
switch(config-ip-sla-echo)# frequency 300
switch(config-ip-sla-echo)# request-data-size 38
switch(config-ip-sla-echo)# tos 160
switch(config-ip-sla-echo)# timeout 6000
switch(config-ip-sla-echo)# tag SFO-RO
switch(config-ip-sla-echo)# end
```

This example shows how to configure an IP SLA operation type of ICMP Echo using IPv6 that will start immediately and run indefinitely:

```
switch# configure terminal
switch(config)# feature sla sender
switch(config)# ip sla 6
switch(config-ip-sla)# icmp-echo 2016:1:1:1::2 source-ip 2016:1:1:1::2
switch(config-ip-sla-echo)# frequency 300
switch(config-ip-sla-echo)# request-data-size 38
switch(config-ip-sla-echo)# traffic-class 160
switch(config-ip-sla-echo)# timeout 6000
switch(config-ip-sla-echo)# tag SFO-RO
switch(config-ip-sla-echo)# end
```

Example: Scheduling IP SLAs Operations

This example shows how to schedule an IP SLAs operation that is already configured:

```
switch# configure terminal
switch(config)# ip sla schedule 6 life forever start-time now
switch(config)# exit
```

Additional References for IP SLAs ICMP Echo Operations

Standards and RFCs

Standard/RFC	Title
RFC 862	Echo Protocol

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History for IP SLAs ICMP Echo Operations

Table 18: Feature History for IP SLAs ICMP Echo Operations

Feature Name	Releases	Feature Information
IP SLAs ICMP Echo Operation	8.0(1)	Added support for operability in IPv6 networks.
IP SLAs ICMP Echo Operation	6.2(2)	The Cisco IP SLAs ICMP echo operation allows you to measure the end-to-end network response time between two devices using IPv4.

CHAPTER 13

Configuring IP SLAs for FabricPath Echo Operation

This module describes how to configure an IP Service Level Agreement (SLA) for FabricPath Echo operation to monitor end-to-end response time between two devices in the FabricPath network.

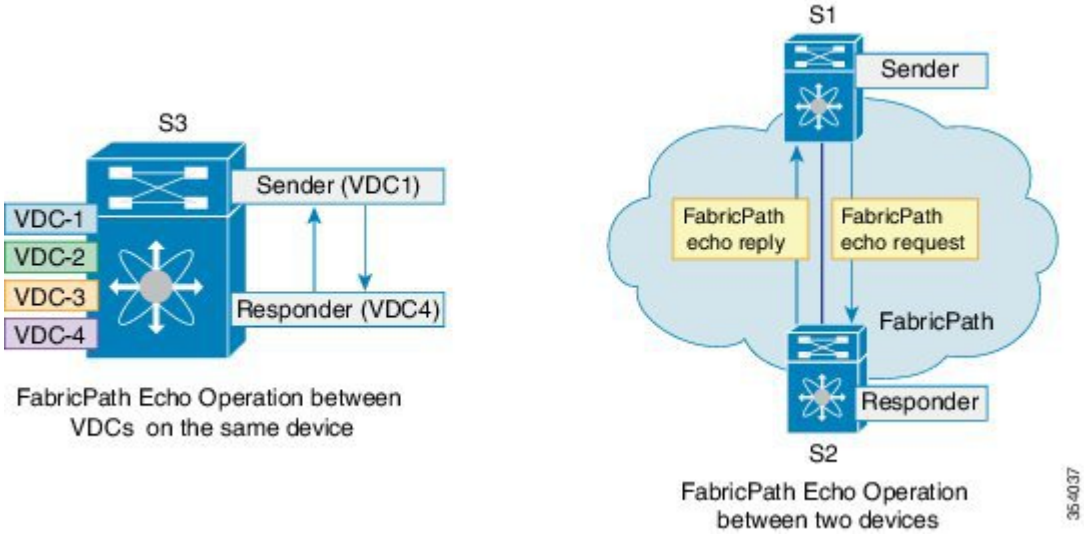
This chapter includes the following sections:

FabricPath Echo Operation Overview

FabricPath Echo operation measures end-to-end response time between two devices in the FabricPath network. The response time is computed by measuring the time taken between sending one FabricPath echo request message (packet) to the destination switch and receiving an echo reply. This provides the round-trip time (RTT) for the packet.

The illustrations in the following figures show the IP SLA FabricPath Echo operation between Virtual Device Contexts (VDCs) on the same device and between two devices, one configured as Sender and the other as Responder.

Figure 14: FabricPath Echo Operation



FabricPath Echo operations help network operators and administrators diagnose data plane failures in Transparent Interconnection of Lots of Links (TRILL) or FabricPath networks by measuring the active performance of the corresponding network and verifying the connectivity of the flow. Users can regularly send echo packets and monitor network performance using system logs and Simple Network Management Protocol (SNMP) traps.

Guidelines and Limitations for Configuring IP SLAs for a FabricPath Echo Operation

Currently, FabricPath Echo operations do not support the **history enhanced** command.

Configuring IP SLAs for FabricPath Echo Operation

Before you begin

Enable the FabricPath feature set in the VDC.

Install the FabricPath feature set before you enable FabricPath on a device. Refer to the chapter "Configuring FabricPath Switching" in the Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide for information about installing and enabling the FabricPath feature set.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**feature sla sender** **Example:** `switch(config)# feature sla sender`	Enables the IP SLA Operation feature.
Step 3	**ip sla** *operation-number* **Example:** `switch(config)# ip sla 1`	Begins configuration for an IP SLA operation and enters IP SLA configuration mode. *Operation-number* is the identification numeral for the IP SLA operation.
Step 4	**fabric-path echo** *switch-id* [**profile** *profile-id* [**interface** *type number*]] **Example:** `switch(config-ip-sla)#fabric-path echo 1 profile 2 interface ethernet 1/0`	Sends a FabricPath echo request to the destination switch with the given profile and interface and enters FabricPath Echo configuration mode. **profile** and **interface** are optional parameters. If they are not specified, the operation takes the default profile and interface selected by the Fabric Operation, Administration, and Maintenance (OAM) process. **Note** If no interface is specified within the profile, the default interface is selected. When a profile has multiple interfaces listed, only the first interface in the list is chosen.
Step 5	**timeout** *milliseconds* **Example:** `switch(config-ip-sla-fabric)#timeout 5000`	(Optional) Configures the time period for which the IP SLA operation waits for a response from its destination device.
Step 6	**frequency** *seconds* **Example:** `switch(config-ip-sla-fabric)#frequency 60`	(Optional) Sets the rate at which the specified SLA operations are repeated. **Note** For an error-free operation, specify a frequency value that is greater than the timeout value.
Step 7	**threshold** *milliseconds* **Example:** `switch(config-ip-sla-fabric)#threshold 5000`	(Optional) Sets the upper threshold value for the RTT measurement. If the RTT exceeds the threshold value, but is less than the timeout value, it generates a reaction event. Default value is 5000 ms.
Step 8	**owner** *owner-id* **Example:**	(Optional) Configures the SNMP owner of an IP SLA operation.

	Command or Action	Purpose
	`switch(config-ip-sla-fabric)#owner admin`	
Step 9	**tag** *text* **Example:** `switch(config-ip-sla-fabric)#tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLA operation.
Step 10	**history buckets-kept** *size* **Example:** `switch(config-ip-sla-fabric)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are maintained during the lifetime of an IP SLA operation.
Step 11	**history distributions-of-statistics-kept** *size* **Example:** `switch(config-ip-sla-fabric)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions maintained per hop during an IP SLA operation.
Step 12	**history lives-kept** *lives* **Example:** `switch(config-ip-sla-fabric)# history lives-kept 5`	(Optional) Sets the number of lives maintained in the history table for an IP SLA operation.
Step 13	**history hours-of-statistics-kept** *hours* **Example:** `switch(config-ip-sla-fabric)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics is maintained for an IP SLA operation.
Step 14	**ip sla schedule** *operation-number* [**life** { **forever** | *seconds*}] [**start-time** {[*hh:mm:ss*] [*month day* | *day month* | **pending** | **now** | **after** *hh:mm:ss* }] [**ageout** *seconds*] [*recurring*] **Example:** `switch(config-ip-sla-fabric)# ip sla schedule 1 life 60 start-time now`	Configures the scheduling parameters for an individual IP SLA operation.
Step 15	**end** **Example:** `switch(config-ip-sla-fabric)# end`	Exits IP SLA FabricPath echo configuration mode and returns to EXEC mode.
Step 16	**show ip sla configuration** **Example:** `switch(config)# show ip sla configuration`	Displays IP SLA configuration status.
Step 17	**show ip sla statistics** **Example:**	Displays IP SLA operation statistics over the last one hour.

	Command or Action	Purpose
	`switch(config)# show ip sla statistics`	

Configuring IP SLA Reaction Configuration for Performance Metrics

IP SLA reactions are configured to trigger when a monitored value exceeds or falls below a specified level, or when a monitored event, such as a timeout or connection loss, occurs. If IP SLAs measure too high or too low in a configured reaction, IP SLAs can generate a notification to a network management application or trigger another IP SLA operation to gather more data.

You can configure IP SLA reaction for FabricPath Echo operation to monitor the *timeout* and *rtt* values.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**ip sla reaction-configuration** *operation-number* **react** *monitored-element* [**threshold-type** {**average** | **consecutive** | **immediate** | **never** | **xofy**}] [**action-type** {**trapOnly** | **triggerOnly** | **trapAndTrigger** | **none**}] [**threshold-value** *upper-threshold lower-threshold*] **Example:** `switch(config)# ip sla reaction-configuration 1 react rtt action-type trapAndTrigger threshold-type immediate threshold-value 100 50`	Configures the action (SNMP trap or IP SLA trigger) that should occur based on violations of specified thresholds. This command enables you to monitor two elements, timeout and RTT.

IP SLA FabricPath Echo Operation Return Codes

A FabricPath Echo operation generates responses (return codes) depending on certain conditions of the operation. The following table lists the various responses from a FabriPath echo operation and the conditions under which they are generated.

Use the **show ip sla statistics** command to view the return codes of an echo operation.

IP SLA FabricPath Echo Operation Response	Explanation
OK	This response is generated under the following conditions: • Loopback operation is successful • RTT is less than timeout value • RTT is less than threshold value
Over Threshold	This response is generated under the following conditions: • loopback operation is successful • RTT is less than timeout value • RTT is greater than threshold value
Timeout	This response is generated when the replies to an echo request do not come at the expected time. This occurs under any of the following conditions: • Request is not sent • Request times out • Destination is unreachable • An unknown code is returned • VLAN does not exist • VLAN is in suspended state • Request is malformed • There is an unsupported TLVS • Cross-connect error occurs • RBridge nickname is unknown • There is no AF • There is an MTU mismatch • Interface is in forwarding state • Service tag is nonexistent • Service tag is in suspended state • Trace route is in progress to get hop count

IP SLA FabricPath Echo Operation Response	Explanation
Internal Error	This response is generated under the following conditions: • Memory allocation failure • Configured profile does not exist • Configured interface does not exist or is shut down

Configuration Examples for IP SLA FabricPath Echo

Example: Configuring an IP SLA FabricPath Echo Operation

The following example shows how to configure operation 6 as an IP SLA FabricPath Echo operation with basic parameters:

```
switch# configure terminal
switch(config)# feature sla sender
switch(config)# ip sla 6
switch(config-ip-sla)# fabric-path echo 1 profile 10 interface ethernet 1/0
switch(config-ip-sla-fabric)# exit
```

Example: Configuring IP SLA FabricPath Echo Operation with Optional Parameters

The following example shows how to configure operation 6 as an IP SLA FabricPath Echo operation with optional parameters:

```
switch# configure terminal
switch(config)# feature sla sender
switch(config)# ip sla 6
switch(config-ip-sla)# fabric-path echo 1 profile 10 interface ethernet 1/0
switch(config-ip-sla-fabric)# timeout 5000
switch(config-ip-sla-fabric)# frequency 60
switch(config-ip-sla-fabric)# owner my_owner
switch(config-ip-sla-fabric)# tag my_tag
switch(config-ip-sla-fabric)# threshold 5000
switch(config-ip-sla-fabric)# history buckets-kept 25
switch(config-ip-sla-fabric)# history distributions-of-statistics-kept 5
switch(config-ip-sla-fabric)# history lives-kept 5
switch(config-ip-sla-fabric)# history hours-of-statistics kept 4
```

Example: Scheduling an IP SLA FabricPath Echo Operation

The following example shows how to schedule an IP SLA FabricPath Echo operation that is already configured.

```
switch# configure terminal
switch(config)# ip sla schedule 6 start-time now life 60
switch(config)# exit
```

Example: Verifying IP SLA FabricPath Echo Operation

The following examples show how to verify a FabricPath Echo configuration using **show** commands:

```
switch# show ip sla statistics
IP SLA Latest Operation Statistics

IPSLA operation id: 6
 Latest RTT: 5 ms
Latest operation start time: 09:54:52 UTC Mon Aug 27 2012
Latest operation return code: OK
Number of successes: 1
Number of failures: 0
Operation time to live: 0 sec

switch# show ip sla statistics aggregated
IPSLAs Latest Operation Statistics

IPSLA operation id: 6
        Min/Max/Avg RTT: 5/5/5 ms

switch# show ip sla configuration
IP SLAs Infrastructure Engine-III
Entry number: 1
Owner:
Tag:
Operation timeout (milliseconds): 5000
Type of operation to perform: fabric-path-echo
Switch ID: 1
Profile ID: 10
Interface: Ethernet 1/0
Schedule:
   Operation frequency (seconds): 60
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 60
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
```

Feature Information for Configuring IP SLAs for FabricPath Echo Operation

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 19: Feature Information for Configuring IP SLAs for FabricPath Echo

Feature Name	Release	Feature Information
IP SLA FabricPath Echo Operation	7.2(0)D1(1)	The Cisco IP SLA FabricPath Echo operation enables you to measure the end-to-end response time between two devices in a FabricPath network.

INDEX

S

T

U

V