Configuring Auto IP SLAs in IP SLAs Engine 3.0

This document describes the auto IP Service Level Agreements (SLAs) function in IP SLAs Engine 3.0, including the following:

- Auto-measure groups--Each template, endpoint list, and scheduler can be configured once and then combined to create auto-measure groups for multiple operations, including operations for proactive threshold monitoring.
- Automatic registration--Cisco devices can be configured to automatically register with the source, which enables the source to discover the destination's address for building an endpoint list.
- Quality of service (QoS) performance--Support for active measurement of QoS.

Finding Feature Information

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

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

Prerequisites for Auto IP SLAs in IP SLAs Engine 3.0

- Your IP network is operational and you can access the destination device.
- If you are using a Cisco IP SLAs Responder on the destination device for any auto IP SLAs operation, the responder must be enabled before you configure the IP SLAs operation. The following operations require that an IP SLAs responder be enabled on the destination device:
  - UDP Echo
  - UPD Jitter
  - VoIP UDP

Restrictions for Auto IP SLAs in IP SLAs Engine 3.0

General Restrictions

- Auto IP SLAs is supported in IPv4 networks only.
- Only the following operation types are supported by auto IP SLAs in Cisco IP SLAs Engine 3.0:
  - Internet Control Message Protocol (ICMP) Echo
  - ICMP Jitter
  - TCP Connect
  - Internet User Datagram Protocol (UDP) Echo
  - UDP Jitter
- If you do not configure and apply a template to an auto-measure group, the default type of operation for the group is ICMP jitter.

UDP Jitter Operations Restrictions

- The 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 will be zero.
- Time synchronization is required between the source and the destination in order to provide accurate one-way delay (latency) measurements. To configure Network Time Protocol (NTP) on the source and destination, perform the tasks in the “Performing Basic System Management” chapter of the *Cisco IOS Network Management Configuration Guide* . Time synchronization is not required for one-way jitter and packet loss measurements. If the time is not synchronized between the source and destination, 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.

- 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.
- 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** command change does not increase the value beyond two hours.
- The CISCO-DATA-COLLECTION-MIB can be used to collect historical data for the operation. For information about the CISCO-DATA-COLLECTION-MIB, see http://www.cisco.com/go/mibs .

UDP Jitter Codec Operations Restrictions

- Cisco IOS IP SLAs Engine 3.0 supports only the following speech codecs:
  - G.711 a-law, 64 kbps PCM compression method
  - G.711 mu-law, 64 kbps PCM compression method
  - G.729A, 8 kbps CS-ACELP compression method
- An IP SLAs UDP jitter codec operation for analyzing VoIP SLAs will fail if control protocol is disabled.
- The **show auto template**command for UDP jitter lists the values for the "Number of statistic distribution buckets kept" and "Statistic distribution interval (milliseconds)," but these values do not apply to UDP jitter codec operations.

Proactive Threshold Measuring Restrictions

- 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 non-Jitter 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.

Auto-Measure Group Restrictions

- Only one auto IP SLAs template can be specified for each auto-measure group. Each template can be referenced by more than one group. If no template is specified for an auto-measure group, the operation type for the group is ICMP jitter by default.

- Only one auto IP SLAs endpoint list can be specified for each auto-measure group. Each endpoint list can be referenced by more than one group.
- Only one auto IP SLAs scheduler can be specified for each auto-measure group. Each scheduler can be referenced by more than one group to create a multioperations schedule.
- You cannot modify the configuration of an auto-measure group if the scheduler specified for the group is configured with a start time other than the default (pending). To avoid conflicts when you configure an auto-measure group, issue the **schedule** command last.

Information About Auto IP SLAs in IP SLAs Engine 3.0

Auto IP SLAs for Analyzing IP SLAs Operations

Auto IP SLAs in Cisco IOS IP SLAs Engine 3.0 consists of the following components:

- Endpoint list--A collection of destination endpoint addresses that can be manually configured or automatically discovered.
- Template--A set of parameters that define a single operation. You can combine any template with any endpoint list and apply the combination across many groups.
- Scheduler--Defines parameters for scheduling an operation including start time, frequency, life, age out, and probe interval. The scheduler can be associated with one or more groups.
- Auto-measure group--Created by combining one auto IP SLAs endpoint list, template, and scheduler. You can combine any template with any endpoint list and any scheduler and apply the combination across many groups.

When the group is scheduled to run, based on the scheduler, one IP SLAs operation is created for each destination address in the endpoint list. The operation type depends on the template in the group. If the group is deleted or unscheduled, the created IP SLAs operations are removed.

For configuration information, see the "How to Configure Auto IP SLAs in Cisco IOS IP SLAs Engine 3.0" section.

QoS Integration for Auto IP SLAs

QoS integration allows for active measurement of QoS performance and enables customers to test network readiness for the deployment of high priority, low latency traffic that is generated by applications such as voice and video. Auto IP SLAs traffic is marked with the appropriate Differentiated Services Code Point (DSCP) marker and then sends out the required interface.

Automatic Registration of Destinations for Auto IP SLAs Operations

Destination Cisco devices and Cisco IOS IP SLAs Responders in Cisco devices can be configured to automatically register with the source. The source will discover the destination's address and put it in an endpoint list.

How to Configure Auto IP SLAs in IP SLAs Engine 3.0

Configuring Automatic Registration of Responder on the Destination Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla responder auto-register** {*source-ipaddress*|*source-hostname*} [**client-id** *client-id*] [**endpoint-list** *template-name*] [**retry-timer** *minutes*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla responder auto-register** {*source-ipaddress*|*source-hostname*} [**client-id** *client-id*] [**endpoint-list** *template-name*] [**retry-timer** *minutes*] **Example:** `Router(config)# ip sla responder auto-register 10.1.1.23 endpoint-list autolist`	Enables destination to register with source.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring an Endpoint List on the Source Device

Perform one of the following tasks:

Using Automatic Registration to Configure an Endpoint List

Before You Begin

XThe destination device or responder must be configured for auto registration. For configuration information, see the "Configuring Automatic Registration of Responder on the Destination Device" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto discovery**
4. **ip sla auto endpoint-list type ip** *template-name*
5. **description** *description*
6. **discover** [**port** *port*]
7. **access-list** {*standard-range* | *expanded-range*}
8. **ageout** *seconds*
9. **measurement-retry** *number-of-retries*
10. **end**
11. show ip sla auto discovery

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto discovery** **Example:** `Router(config)# ip sla auto discovery`	Enables the source to discover the responder endpoints.

	Command or Action	Purpose
Step 4	**ip sla auto endpoint-list type ip** *template-name* **Example:** `Router(config)# ip sla auto endpoint-list type ip autolist`	Begins configuring an endpoint list and enters endpoint-list configuration mode.
Step 5	**description** *description* **Example:** `Router(config-epl)# description testing discovery`	(Optional) Adds descriptive text to the template being configured.
Step 6	**discover** [**port** *port*] **Example:** `Router(config-epl)# discover`	Automatically discovers the IP addresses and port numbers of endpoints on the destination and enters endpoint-list auto-discovery configuration mode.
Step 7	**access-list** {*standard-range* | *expanded-range*} **Example:** `Router(config-epl-disc)# access-list 1`	Names and adds the list of discovered IP addresses to the endpoint list being configured.
Step 8	**ageout** *seconds* **Example:** `Router(config-epl-disc)# ageout 10`	(Optional) Sets an ageout timer.
Step 9	**measurement-retry** *number-of-retries* **Example:** `Router(config-epl-disc)# measurement-retry 10`	(Optional) Specifies the number of times to retry after an operation fails.
Step 10	**end** **Example:** `Router(config-ep-disc)# end`	Returns to privileged EXEC mode.
Step 11	show ip sla auto discovery **Example:** `Router# show ip sla auto discovery`	(Optional) Displays status of auto-discovery.

Manually Configuring an Endpoint List

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto endpoint-list type ip** *template-name*
4. **description** *description*
5. **ip-address** *address* [**-** *address* | **,** *...* **,** *address*] **port** *port*
6. **end**
7. **show ip sla auto** [**type ip** [*template-name*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto endpoint-list type ip** *template-name* **Example:** `Router(config)# ip sla auto endpoint-list type ip edgehosts-epl`	Begins configuring an endpoint list and enters endpoint-list configuration mode.
Step 4	**description** *description* **Example:** `Router(config-epl)# description manual config`	(Optional) Adds descriptive text to the template being configured.
Step 5	**ip-address** *address* [**-** *address*	**,** *...* **,** *address*] **port** *port* **Example:** `Router(config-epl)# ip-address 10.1.1.1-13 port 6500`	Adds the IP addresses of endpoints to the endpoint list being configured. • Use the **no** from of this command to modify the endpoint list by removing one or more addresses. • Repeat this command until all desired addresses are configured.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-epl)# end`	Returns to privileged EXEC mode.
Step 7	**show ip sla auto** [**type ip** [*template-name*]] **Example:** `Router# show ip sla auto type ip edgehosts-epl`	(Optional) Displays the configuration of the endpoint list.

Configuring a Scheduler on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto schedule** *schedule-id*
4. **ageout** *seconds*
5. **frequency** {*seconds* | **range** *random-frequency-range*}
6. **life** {**forever** | *seconds*}
7. **probe-interval** *milliseconds*
8. **start-time** {*hh* **:** *mm*[*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}
9. **end**
10. **show ip sla auto schedule** [*schedule-id*]

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip sla auto schedule** *schedule-id* **Example:** `Router(config)# ip sla auto schedule theschedule`	Begins configuring a scheduler and enters IP SLA auto-measure schedule configuration mode.
Step 4	**ageout** *seconds* **Example:** `Router(config-am-schedule)# ageout 43200`	(Optional) Sets an ageout timer for inactive operations.
Step 5	**frequency** {*seconds* | **range** *random-frequency-range*} **Example:** `Router(config-am-schedule)# frequency 60`	(Optional) Specifies either the frequency timer or random frequency range for multiple operations.
Step 6	**life** {**forever** | *seconds*} **Example:** `Router(config-am-schedule)# life 43200`	(Optional) Defines the lifetime of the operation.
Step 7	**probe-interval** *milliseconds* **Example:** `Router(config-am-schedule)# probe-interval 200`	(Optional) Specifies the interval for multiple operations.
Step 8	**start-time** {*hh* **:** *mm*[*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*} **Example:** `Router(config-am-schedule)# start-time now`	(Optional) Specifies the start time of an operation.
Step 9	**end** **Example:** `Router(config-am-schedule)# end`	Returns to privileged EXEC mode.
Step 10	**show ip sla auto schedule** [*schedule-id*] **Example:** `Router# show ip sla auto schedule`	(Optional) Displays the configuration of the scheduler.

Configuring a Template on the Source Device

Note If you do not configure and apply a template to an auto-measure group, the default type of operation for the group is ICMP jitter.

Configuring an ICMP Echo Operation

Perform one of the following tasks:

Configuring a Template for a Basic ICMP Echo Operation

Note We recommend using a Cisco networking device as the destination device although any networking device that supports RFC 862, *Echo protocol* , can be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip icmp-echo** *template-name*
4. **description** *description*
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip sla auto template type ip icmp-echo** *template-name* **Example:** `Router(config)# ip sla auto template type ip icmp-echo tmp-icmpecho`	Begins configuring a template and enters IP SLA template configuration mode.
Step 4	**description** *description* **Example:** `Router(config-tplt-icmp-ech)# description default oper temp for icmp echo`	(Optional) Adds descriptive text to the template being configured.
Step 5	**end** **Example:** `Router(config-tplt-icmp-ech)# end`	Returns to global configuration mode.

Example

The following output shows the configuration, including default values, of a template for an ICMP echo operation:

```
Router# show
 ip sla auto template type ip icmp-echo
IP SLAs Auto Template: tpl-icmplecho
    Measure Type: icmp-echo (control enabled)
    Description: default oper temp for icmp echo
    IP options:
        Source IP: 0.0.0.0      Source Port: 0
        VRF:    TOS: 0x0
    Operation Parameters:
        Request Data Size: 16   Verify Data: false
        Timeout: 5000           Threshold: 5000
    Statistics Aggregation option:
        Hours of statistics kept: 2
    History options:
        History filter: none
        Max number of history records kept: 15
        Lives of history kept: 0
    Statistics Distributions options:
        Distributions characteristics: RTT
        Distributions bucket size: 20
        Max number of distributions buckets: 1
    Reaction Configuration: None
```

What to Do Next

To configure this auto IP SLAs template for proactive threshold monitoring, see the “Adding Proactive Threshold Monitoring to a Template on the Source Device” section. Otherwise, see the “Configuring an Auto-Measure Group on the Source Device” section.

Configuring an ICMP Echo Operation Template with Additional Characteristics

Note We recommend using a Cisco networking device as the destination device although any networking device that supports RFC 862, *Echo Protocol* , can be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip icmp-echo** template-name
4. **description** *description*
5. **source-ip** {*ip-address* | *hostname*}
6. **tos** *number*
7. **vrf** *vrf-name*
8. **parameters**
9. **history buckets-kept** *size*
10. **history distributions-of-statistics-kept** *size*
11. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
12. **history filter** {**none** | **all** | **overThreshold** | **failures**}
13. **history hours-of-statistics-kept** *hours*
14. **history lives-kept** *lives*
15. **history statistics-distribution-interval** *milliseconds*
16. **request-data-size** *bytes*
17. **threshold** *milliseconds*
18. **timeout** *milliseconds*
19. **verify-data**
20. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip sla auto template type ip icmp-echo** template-name **Example:** `Router(config)# ip sla auto template type ip icmp-echo icmp_echo_2`	Begins configuring a template and enters IP SLA template configuration mode.
Step 4	**description** *description* **Example:** `Router(config-tplt-icmp-ech)# description custom icmp echo template`	(Optional) Adds descriptive text to the template being configured.
Step 5	**source-ip** {*ip-address* | *hostname*} **Example:** `Router(config-tplt-icmp-ech)# source-ip 172.29.139.132`	(Optional) Specifies the source for the operation.
Step 6	**tos** *number* **Example:** `Router(config-tplt-icmp-ech)# tos 160`	(Optional) Defines the Type of Service (ToS) byte in the IPv4 header of an IP SLAs operation.
Step 7	**vrf** *vrf-name* **Example:** `Router(config-tplt-icmp-ech)# vrf vpn-A`	(Optional) Allows monitoring within MultiProtocol Label Switching (MPLS) VPNs using IP SLAs operations.
Step 8	**parameters** **Example:** `Router(config-tplt-icmp-ech)# parameters`	(Optional) Enters IP SLA template parameters configuration mode.
Step 9	**history buckets-kept** *size* **Example:** `Router(config-icmp-ech-params)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 10	**history distributions-of-statistics-kept** *size* **Example:** `Router(config-icmp-ech-params)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.

	Command or Action	Purpose
Step 11	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Router(config-icmp-ech-params)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 12	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `Router(config-icmp-ech-params)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 13	**history hours-of-statistics-kept** *hours* **Example:** `Router(config-icmp-ech-params)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 14	**history lives-kept** *lives* **Example:** `Router(config-icmp-ech-params)# history lives-kept 5`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 15	**history statistics-distribution-interval** *milliseconds* **Example:** `Router(config-icmp-ech-params)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 16	**request-data-size** *bytes* **Example:** `Router(config-icmp-ech-params)# request-data-size 64`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.
Step 17	**threshold** *milliseconds* **Example:** `Router(config-icmp-ech-params)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 18	**timeout** *milliseconds* **Example:** `Router(config-icmp-ech-params)# 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 19	**verify-data** **Example:** `Router(config-icmp-ech-params)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption. **Note** Use this command with caution during normal operations.
Step 20	**end** **Example:** `Router(config-icmp-ech-params)# end`	Returns to privileged EXEC mode.

What to Do Next

To configure this auto IP SLAs template for proactive threshold monitoring, see the "Adding Proactive Threshold Monitoring to a Template on the Source Device" section. Otherwise, see the "Configuring an Auto-Measure Group on the Source Device" section.

Configuring an ICMP Jitter Operation

Perform one of the following tasks:

Configuring a Template for a Basic ICMP Jitter Operation

Note

- When compared to the IP SLAs UDP jitter operation, the IP SLAs ICMP jitter operation may provide less accurate measurements because the accuracy of the measurements provided by a non-Cisco destination device cannot be determined.
- Because ICMP packets do not support voice technology, the IP SLAs ICMP jitter operation does not support Mean Operation Sore (MOS), Calculated Planning Impairment Factor (ICPIF), or estimated transmission rating factor (R-factor) reaction configuration capabilities.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip icmp-jitter** *template-name*
4. **description** *description*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto template type ip icmp-jitter** *template-name* **Example:** `Router(config)# ip sla auto template type ip icmp-jitter basic_icmp_jitter`	Begins configuring the template and enters IP SLA template configuration mode.
Step 4	**description** *description* **Example:** `Router(config-tplt-icmp-jtr)# description default oper temp for icmp-jitter`	(Optional) Adds descriptive text to the template being configured.
Step 5	**end** **Example:** `Router(config-tplt-icmp-jtr)# end`	Returns to privileged EXEC mode.

Example

The following output shows the configuration, including default values, of a template for a basic ICMP jitter operation:

```
Router# show ip sla auto template type ip icmp-jitter
IP SLAs Auto Template: basic_icmp_jitter
    Measure Type: icmp-jitter
    Description: default oper temp for icmp jitter
    IP options:
        Source IP: 0.0.0.0
        VRF:    TOS: 0x0
    Operation Parameters:
        Number of Packets: 10   Inter packet interval: 20
        Timeout: 5000           Threshold: 5000
    Statistics Aggregation option:
        Hours of statistics kept: 2
    Statistics Distributions options:
        Distributions characteristics: RTT
        Distributions bucket size: 20
```

```
    Max number of distributions buckets: 1
Reaction Configuration: None
```

What to Do Next

To configure this auto IP SLAs template for proactive threshold monitoring, see the "Adding Proactive Threshold Monitoring to a Template on the Source Device" section. Otherwise, see the "Configuring an Auto-Measure Group on the Source Device" section.

Configuring an ICMP Jitter Operation Template with Additional Characteristics

Note

- When compared to the IP SLAs UDP jitter operation, the IP SLAs ICMP jitter operation may provide less accurate measurements because the accuracy of the measurements provided by a non-Cisco destination device cannot be determined.
- Because ICMP packets do not support voice technology, the IP SLAs ICMP jitter operation does not support MOS, ICPIF, or estimated R-factor reaction configuration capabilities.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip icmp-jitter** *template-name*
4. **description** *description*
5. **source-ip** {*ip-address* | *hostname*}
6. **tos** *number*
7. **vrf** *vrf-name*
8. **parameters**
9. **history distributions-of-statistics-kept** *size*
10. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
11. **history filter** {**none** | **all** | **overThreshold** | **failures**}
12. **history hours-of-statistics-kept** *hours*
13. **history statistics-distribution-interval** *milliseconds*
14. **interval** *milliseconds*
15. **num-packet** *packet-number*
16. **threshold** *milliseconds*
17. **timeout** *milliseconds*
18. **verify-data**
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto template type ip icmp-jitter** *template-name* **Example:** `Router(config)# ip sla auto template type ip icmp-jitter custom_imcp_jitter`	Begins configuring a template and enters IP SLA template configuration mode.
Step 4	**description** *description* **Example:** `Router(config-tplt-icmp-jtr)# description custom icmp jitter template`	(Optional) Adds descriptive text to the template being configured.
Step 5	**source-ip** {*ip-address* | *hostname*} **Example:** `Router(config-tplt-icmp-jtr)# source-ip 10.1.2.34`	(Optional) Specifies the source for the operation.
Step 6	**tos** *number* **Example:** `Router(config-tplt-icmp-jtr)# tos 160`	(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.
Step 7	**vrf** *vrf-name* **Example:** `Router(config-tplt-icmp-jtr)# vrf vpn-A`	(Optional) Allows monitoring within MPLS VPNs using IP SLAs operations.
Step 8	**parameters** **Example:** `Router(config-tplt-icmp-jtr)# parameters`	(Optional) Enters IP SLA template parameters configuration mode.

	Command or Action	Purpose
Step 9	**history distributions-of-statistics-kept** *size* **Example:** `Router(config-icmp-jtr-params)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 10	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Router(config-icmp-jtr-params)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 11	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `Router(config-icmp-jtr-params)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 12	**history hours-of-statistics-kept** *hours* **Example:** `Router(config-icmp-jtr-params)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 13	**history statistics-distribution-interval** *milliseconds* **Example:** `Router(config-icmp-jtr-params)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 14	**interval** *milliseconds* **Example:** `Router(config-icmp-jtr-params)# interval 40`	(Optional) Sets number of milliseconds between sent packets.
Step 15	**num-packet** *packet-number* **Example:** `Router(config-icmp-jtr-params)# num-packets 100`	(Optional) Sets number of packets to be sent.
Step 16	**threshold** *milliseconds* **Example:** `Router(config-icmp-jtr-params)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.

	Command or Action	Purpose
Step 17	**timeout** *milliseconds* **Example:** `Router(config-icmp-jtr-params)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 18	**verify-data** **Example:** `Router(config-icmp-jtr-params)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption. **Note** Use this command with caution during normal operations because it generates unnecessary overhead.
Step 19	**end** **Example:** `Router(config-icmp-jtr-params)# end`	Returns to global configuration mode.

What to Do Next

To configure this auto IP SLAs template for proactive threshold monitoring, see the "Adding Proactive Threshold Monitoring to a Template on the Source Device" section. Otherwise, see the "Configuring an Auto-Measure Group on the Source Device" section.

Configuring a TCP Connect Operation

Perform one of the following tasks:

Configuring a Template for a Basic TCP Connect Operation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip tcp-connect** *template-name*
4. **description** *description*
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto template type ip tcp-connect** *template-name* **Example:** `Router(config)# ip sla auto template type ip tcp-connect basic_tcp_connect`	Begins configuring a template and enters IP SLA IP SLA template configuration mode.
Step 4	**description** *description* **Example:** `Router(config-tplt-tcp-conn)# description default oper temp for tcp-connect`	(Optional) Adds descriptive text to the template being configured.
Step 5	**end** **Example:** `Router(config-tplt-tcp-conn)# end`	Returns to privileged EXEC mode.

Example
The following output shows the configuration, including default values, of a template for a basic TCP connect operation:

```
Router# show ip sla auto template type ip tcp-connect
IP SLAs Auto Template: basic_tcp_connect
    Measure Type: tcp-connect (control enabled)
    Description: default oper temp for tcp-connect
    IP options:
        Source IP: 0.0.0.0      Source Port: 0
        VRF:    TOS: 0x0
    Operation Parameters:
        Timeout: 5000           Threshold: 5000
    Statistics Aggregation option:
        Hours of statistics kept: 2
    History options:
        History filter: none
        Max number of history records kept: 15
        Lives of history kept: 0
    Statistics Distributions options:
        Distributions characteristics: RTT
        Distributions bucket size: 20
        Max number of distributions buckets: 1
    Reaction Configuration: None
```

What to Do Next

To configure this auto IP SLAs template for proactive threshold monitoring, see the "Adding Proactive Threshold Monitoring to a Template on the Source Device" section. Otherwise, see the "Configuring an Auto-Measure Group on the Source Device" section.

Configuring a TCP Connect Operation Template with Additional Characteristics

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip tcp-connect** *template-name*
4. **control** {**enable** | **disable**}
5. **description** *description*
6. **source-ip** {*ip-address* | *hostname*}
7. **source-port** *port-number*
8. **tos** *number*
9. **vrf** *vrf-name*
10. **parameters**
11. **history buckets-kept** *size*
12. **history distributions-of-statistics-kept** *size*
13. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
14. **history filter** {**none** | **all** | **overThreshold** | **failures**}
15. **history hours-of-statistics-kept** *hours*
16. **history lives-kept** *lives*
17. **history statistics-distribution-interval** *milliseconds*
18. **threshold** *milliseconds*
19. **timeout** *milliseconds*
20. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip sla auto template type ip tcp-connect** *template-name* **Example:** `Router(config)# ip sla auto template type ip tcp-connect custom_tcp_connect`	Begins configuring a template and enters IP SLA template configuration mode.
Step 4	**control** {**enable** | **disable**} **Example:** `Router(config-tplt-tcp-conn)# control enable`	(Optional) Specifies whether control messages are sent to responder before beginning the IP SLAs operation. • Default--Control is enabled.
Step 5	**description** *description* **Example:** `Router(config-tplt-tcp-conn)# description custom tcp-connect template`	(Optional) Adds descriptive text to the template being configured.
Step 6	**source-ip** {*ip-address* | *hostname*} **Example:** `Router(config-tplt-tcp-conn)# source-ip 10.1.2.34`	(Optional) Specifies the source for the operation.
Step 7	**source-port** *port-number* **Example:** `Router(config-tplt-tcp-conn)# source-port 23`	(Optional) Specifies the source port for the operation.
Step 8	**tos** *number* **Example:** `Router(config-tplt-tcp-conn)# tos 160`	(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.
Step 9	**vrf** *vrf-name* **Example:** `Router(config-tplt-tcp-conn)# vrf vpn-A`	(Optional) Allows monitoring within MPLS VPNs using IP SLAs operations.
Step 10	**parameters** **Example:** `Router(config-tplt-tcp-conn)# parameters`	Enters IP SLA template parameters configuration mode.

	Command or Action	Purpose
Step 11	**history buckets-kept** *size* **Example:** `Router(config-tcp-conn-params)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 12	**history distributions-of-statistics-kept** *size* **Example:** `Router(config-tcp-conn-params)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 13	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Router(config-tcp-conn-params)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 14	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `Router(config-tcp-conn-params)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 15	**history hours-of-statistics-kept** *hours* **Example:** `Router(config-tcp-conn-params)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 16	**history lives-kept** *lives* **Example:** `Router(config-tcp-conn-params)# history lives-kept 5`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 17	**history statistics-distribution-interval** *milliseconds* **Example:** `Router(config-tcp-conn-params)# 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 18	**threshold** *milliseconds* **Example:** `Router(config-tcp-conn-params)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 19	**timeout** *milliseconds* **Example:** `Router(config-tcp-conn-params)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 20	**end** **Example:** `Router(config-tcp-conn-params)# end`	Returns to privileged EXEC mode.

What to Do Next

To configure this auto IP SLAs template for proactive threshold monitoring, see the "Adding Proactive Threshold Monitoring to a Template on the Source Device" section. Otherwise, see the "Configuring an Auto-Measure Group on the Source Device" section.

Configuring a UDP Echo Operation

Perform one of the following tasks:

Configuring a Template for a Basic UDP Echo Operation

Note We recommend using a Cisco networking device as the destination device although any networking device that supports RFC 862, *Echo Protocol* , can be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip udp-echo** *template-name*
4. **description** *description*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto template type ip udp-echo** *template-name* **Example:** `Router(config)# ip sla auto template type ip udp-echo basic_udp_echo`	Begins configuring a template and enters IP SLA template configuration mode.
Step 4	**description** *description* **Example:** `Router(config-tplt-udp-ech)# description default oper temp for udp echo`	(Optional) Adds descriptive text to the template being configured.
Step 5	**end** **Example:** `Router(config-tplt-udp-ech)# end`	Returns to privileged EXEC mode.

Example
The following output shows the configuration, including default values, of a template for a basic UDP echo operation:

```
Router# show ip sla auto template type ip udp-echo
IP SLAs Auto Template: basic_udp_echo
    Measure Type: udp-echo (control enabled)
    Description: default oper temp for udp echo
    IP options:
        Source IP: 0.0.0.0      Source Port: 0
        VRF:    TOS: 0x0
    Operation Parameters:
        Request Data Size: 16   Verify Data: false
        Timeout: 5000           Threshold: 5000
    Statistics Aggregation option:
        Hours of statistics kept: 2
    History options:
        History filter: none
        Max number of history records kept: 15
        Lives of history kept: 0
```

```
Statistics Distributions options:
    Distributions characteristics: RTT
    Distributions bucket size: 20
    Max number of distributions buckets: 1
Reaction Configuration: None
```

What to Do Next

To configure this auto IP SLAs template for proactive threshold monitoring, see the "Adding Proactive Threshold Monitoring to a Template on the Source Device" section. Otherwise, see the "Configuring an Auto-Measure Group on the Source Device" section.

Configuring a UDP Echo Operation Template with Additional Characteristics

Note We recommend using a Cisco networking device as the destination device although any networking device that supports RFC 862, *Echo Protocol* , can be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip udp-echo** *template-name*
4. **description** *description*
5. **source-ip** {*ip-address* | *hostname*}
6. **source-port** port-number
7. **tos** *number*
8. **vrf** *vrf-name*
9. **parameters**
10. **history buckets-kept** *size*
11. **history distributions-of-statistics-kept** *size*
12. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
13. **history filter** {**none** | **all** | **overThreshold** | **failures**}
14. **history hours-of-statistics-kept** *hours*
15. **history lives-kept** *lives*
16. **history statistics-distribution-interval** *milliseconds*
17. **request-data-size** *bytes*
18. **threshold** *milliseconds*
19. **timeout** *milliseconds*
20. **verify-data**
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto template type ip udp-echo** *template-name* **Example:** `Router(config)# ip sla auto template type ip udp-echo custom_udp_echo`	Begins configuring a template and enters IP SLA template configuration mode.
Step 4	**description** *description* **Example:** `Router(config-tplt-udp-ech)# description custom udp echo template`	(Optional) Adds descriptive text to the template being configured.
Step 5	**source-ip** {*ip-address* | *hostname*} **Example:** `Router(config-tplt-udp-ech)# source-ip 10.1.2.34`	(Optional) Specifies the source for the operation.
Step 6	**source-port** port-number **Example:** `Router(config-tplt-udp-ech)# source-port 23`	(Optional) Specifies the source port for the operation.
Step 7	**tos** *number* **Example:** `Router(config-tplt-udp-ech)# tos 160`	(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.
Step 8	**vrf** *vrf-name* **Example:** `Router(config-tplt-udp-ech)# vrf vpn-A`	(Optional) Allows monitoring within MPLS VPNs using IP SLAs operations.

	Command or Action	Purpose
Step 9	**parameters** **Example:** `Router(config-tplt-udp-ech)# parameters`	Enters IP SLA template parameters configuration mode.
Step 10	**history buckets-kept** *size* **Example:** `Router(config-udp-ech-params)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 11	**history distributions-of-statistics-kept** *size* **Example:** `Router(config-udp-ech-params)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 12	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Router(config-udp-ech-params)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 13	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `Router(config-udp-ech-params)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 14	**history hours-of-statistics-kept** *hours* **Example:** `Router(config-udp-ech-params)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 15	**history lives-kept** *lives* **Example:** `Router(config-udp-ech-params)# history lives-kept 5`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.

	Command or Action	Purpose
Step 16	**history statistics-distribution-interval** *milliseconds* **Example:** `Router(config-udp-ech-params)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 17	**request-data-size** *bytes* **Example:** `Router(config-udp-ech-params)# request-data-size 64`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.
Step 18	**threshold** *milliseconds* **Example:** `Router(config-udp-ech-params)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 19	**timeout** *milliseconds* **Example:** `Router(config-udp-ech-params)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 20	**verify-data** **Example:** `Router(config-udp-ech-params)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption. **Note** Use this command with caution during normal operations.
Step 21	**end** **Example:** `Router(config-udp-ech-params)# end`	Returns to privileged EXEC mode.

What to Do Next

To configure this auto IP SLAs template for proactive threshold monitoring, see the "Adding Proactive Threshold Monitoring to a Template on the Source Device" section. Otherwise, see the "Configuring an Auto-Measure Group on the Source Device" section.

Configuring a UDP Jitter Operation on the Source Device

Perform one of the following tasks:

Configuring a Template for a Basic UDP Jitter Operation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip udp-jitter** *template-name*
4. **description** *description*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto template type ip udp-jitter** *template-name* **Example:** `Router(config)# ip sla auto template type ip basic_udp_jitter`	Begins configuring a template and enters IP SLA template configuration mode.
Step 4	**description** *description* **Example:** `Router(config-tplt-udp-jtr)# description default oper temp for udp jitter`	(Optional) Adds descriptive text to the template being configured.
Step 5	**end** **Example:** `Router(config-tplt-udp-jtr)# end`	Returns to privileged EXEC mode.

Example

The following output shows the configuration, including default values, of a template for a basic UDP jitter operation:

```
Router# show ip sla auto template type ip udp-jitter
IP SLAs Auto Template: basic_udp_jitter
    Measure Type: udp-jitter (control enabled)
    Description: default oper temp for udp jitter
    IP options:
        Source IP: 0.0.0.0      Source Port: 0
        VRF:    TOS: 0x0
    Operation Parameters:
        Request Data Size: 32   Verify Data: false
        Number of Packets: 10   Inter packet interval: 20
        Timeout: 5000           Threshold: 5000
        Granularity: msec       Operation packet priority: normal
    Statistics Aggregation option:
        Hours of statistics kept: 2
    Statistics Distributions options:
        Distributions characteristics: RTT
        Distributions bucket size: 20
        Max number of distributions buckets: 1
    Reaction Configuration: None
```

What to Do Next

To configure this auto IP SLAs template for proactive threshold monitoring, see the "Adding Proactive Threshold Monitoring to a Template on the Source Device" section. Otherwise, see the "Configuring an Auto-Measure Group on the Source Device" section.

Configuring a Template for a Basic VoIP SLAs UDP Operation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip udp-jitter** *template-name*
4. **codec** *codec-type* [**advantage-factor** *value*][**codec-numpackets** *number-of-packets*][**codec-interval** *milliseconds*][**codec-size** *number-of-bytes*]
5. **description** *description*
6. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto template type ip udp-jitter** *template-name* **Example:** `Router(config)# ip sla auto template type ip voip_g711alaw`	Begins configuring a template and enters IP SLA template configuration mode.
Step 4	**codec** *codec-type* [**advantage-factor** *value*][**codec-numpackets** *number-of-packets*][**codec-interval** *milliseconds*][**codec-size** *number-of-bytes*] **Example:** `Router(config-tplt-udp-jtr)# codec g711alaw`	Sets the codec for the operation being configured.
Step 5	**description** *description* **Example:** `Router(config-tplt-udp-jtr)# description oper template for voip udp`	(Optional) Adds descriptive text to the template being configured.
Step 6	**end** **Example:** `Router(config-tplt-udp-jtr)# end`	Returns to privileged EXEC mode.

Example

The following output shows the configuration, including default values, of a template for a basic VoIP SLAs UDP jitter operation:

```
Router# show ip sla auto template type ip udp-jitter voip_g711alaw
IP SLAs Auto Template: voip_g711alaw
    Measure Type: udp-jitter (control enabled)
    Description: oper template for voip udp
    IP options:
        Source IP: 0.0.0.0      Source Port: 0
        VRF:    TOS: 0x0
    Operation Parameters:
        Verify Data: false
        Timeout: 5000           Threshold: 5000
        Codec: g711alaw Number of packets: 1000
        Interval: 20    Payload size: 16        Advantage factor: 0
        Granularity: msec       Operation packet priority: normal
    Statistics Aggregation option:
```

```
    Hours of statistics kept: 2
Statistics Distributions options:
    Distributions characteristics: RTT
    Distributions bucket size: 20
    Max number of distributions buckets: 1
Reaction Configuration: None
```

What to Do Next

To configure this auto IP SLAs template for proactive threshold monitoring, see the "Adding Proactive Threshold Monitoring to a Template on the Source Device" section. Otherwise, see the "Configuring an Auto-Measure Group on the Source Device" section.

Configuring a UDP Jitter Operation Template with or without Codec with Additional Characteristics

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip udp-jitter** *template-name*
4. **codec** *codec-type* [**advantage-factor** *value*] [**codec-numpackets** *number-of-packets*] [**codec-interval** *milliseconds*] [**codec-size** *number-of-bytes*]
5. **control** {**enable** | **disable**}
6. **description** *description*
7. **source-ip** {*ip-address* | *hostname*}
8. **source-port** *port-number*
9. **tos** *number*
10. **vrf** *vrf-name*
11. **parameters**
12. **precision** {**milliseconds** | **microseconds**}
13. **clock-tolerance ntp oneway** {**absolute** *value* | **percent** *value*}
14. **history distributions-of-statistics-kept** *size*
15. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
16. **history hours-of-statistics-kept** *hours*
17. **history statistics-distribution-interval** *milliseconds*
18. **interval** *milliseconds*
19. **num-packets** *packet-number*
20. **operation-packet-priority** {**normal** | **high**}
21. **request-data-size** *bytes*
22. **threshold** *milliseconds*
23. **timeout** *milliseconds*
24. **verify-data**
25. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto template type ip udp-jitter** *template-name* **Example:** `Router(config)# ip sla auto template type ip udp-jitter custom_udp_jitter`	Begins configuring a template and enters IP SLA template configuration mode.
Step 4	**codec** *codec-type* [**advantage-factor** *value*] [**codec-numpackets** *number-of-packets*] [**codec-interval** *milliseconds*] [**codec-size** *number-of-bytes*] **Example:** `Router(config-tplt-udp-jtr)# codec g711ulaw advantage-factor 2`	(Optional) Required for analyzing VoIP SLAs. Sets the codec for the operation being configured. **Note** You should not specify values for the interval, size, and number of packet options unless you have a specific reason to override the defaults.
Step 5	**control** {**enable** | **disable**} **Example:** `Router(config-tplt-udp-jtr)# control enable`	(Optional) Specifies whether control messages are sent to responder before beginning the IP SLAs operation. • For UDP jitter codec operations--Do not disable control protocol. • Default--Control is enabled.
Step 6	**description** *description* **Example:** `Router(config-tplt-udp-jtr)# description custom udp jitter template`	(Optional) Adds descriptive text to the template being configured.
Step 7	**source-ip** {*ip-address* | *hostname*} **Example:** `Router(config-tplt-udp-jtr)# source-ip 10.1.2.34`	(Optional) Specifies the source for the operation.

	Command or Action	Purpose
Step 8	**source-port** *port-number* **Example:** `Router(config-tplt-udp-jtr)# source-port 23`	(Optional) Specifies the source port for the operation.
Step 9	**tos** *number* **Example:** `Router(config-tplt-udp-jtr)# tos 160`	(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.
Step 10	**vrf** *vrf-name* **Example:** `Router(config-tplt-udp-jtr)# vrf vpn-A`	(Optional) Allows monitoring within MPLS VPNs using IP SLAs operations.
Step 11	**parameters** **Example:** `Router(config-tplt-udp-jtr)# parameters`	(Optional) Enters IP SLA template parameters configuration mode.
Step 12	**precision** {**milliseconds** | **microseconds**} **Example:** `Router(config-udp-jtr-params)# precision microseconds`	(Optional) Sets the level of precision at which the statistics for an IP SLAs operation are measured • This command must be configured before the **clock-tolerance ntp oneway**command.
Step 13	**clock-tolerance ntp oneway** {**absolute** *value* | **percent** *value*} **Example:** `Router(config-udp-jtr-params)# clock-tolerance ntp oneway percent 100`	(Optional) Sets the acceptable NTP clock synchronization tolerance for a one-way IP SLAs operation measurement.
Step 14	**history distributions-of-statistics-kept** *size* **Example:** `Router(config-udp-jtr-params)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 15	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Router(config-udp-jtr-params)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.

	Command or Action	Purpose
Step 16	**history hours-of-statistics-kept** *hours* **Example:** `Router(config-udp-jtr-params)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 17	**history statistics-distribution-interval** *milliseconds* **Example:** `Router(config-udp-jtr-params)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 18	**interval** *milliseconds* **Example:** `Router(config-udp-jtr-params)# interval 40`	(Optional) Specifies the interval between packets for an IP SLA jitter operation.
Step 19	**num-packets** *packet-number* **Example:** `Router(config-udp-jtr-params)# num-packets 100`	(Optional) Specifies the number of packets for an IP SLAs jitter operation.
Step 20	**operation-packet-priority** {**normal** | **high**} **Example:** `Router(config-udp-jtr-params)# operation-packet-priority high`	(Optional) Specifies the packet priority for an IP SLAs jitter operation.
Step 21	**request-data-size** *bytes* **Example:** `Router(config-udp-jtr-params)# request-data-size 64`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.
Step 22	**threshold** *milliseconds* **Example:** `Router(config-udp-jtr-params)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 23	**timeout** *milliseconds* **Example:** `Router(config-udp-jtr-params)# 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 24	**verify-data** **Example:** `Router(config-udp-jtr-params)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption. **Note** Use this command with caution during normal operations.
Step 25	**end** **Example:** `Router(config-udp-jtr-params)# end`	Returns to privileged EXEC mode.

What to Do Next

To configure this auto IP SLAs template for proactive threshold monitoring, see the "Adding Proactive Threshold Monitoring to a Template on the Source Device" section. Otherwise, see the "Configuring an Auto-Measure Group on the Source Device" section.

Adding Proactive Threshold Monitoring to a Template on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto template type ip** *operation template-name*
4. **description** *description*
5. **react** [*monitored-element*[[**action-type**{*type-of-action*}] [**threshold-type**{**average** [*number-of-measurements*] | **consecutive** [*occurrences*] | **immediate** | **never** | **xofy**[*x-value y-value*]}] [**threshold-value** *upper-threshold lower-threshold*]]]
6. **exit**
7. **ip sla logging traps**
8. Do one of the following:
   - **snmp-server enable traps rtr**
   - **snmp-server enable traps syslog**
9. **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*]
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto template type ip** *operation template-name* **Example:** `Router(config)# ip sla auto template type ip icmp-echo icmp_echo_2`	Begins configuring a template and enters the appropriate submode, based on the *operation* argument, of IP SLA template configuration mode.
Step 4	**description** *description* **Example:** `Router(config-tplt-icmp-ech)# description custom icmp echo template`	(Optional) Adds descriptive text to the template being configured.
Step 5	**react** [*monitored-element*[[**action-type**{*type-of-action*}] [**threshold-type**{**average** [*number-of-measurements*] | **consecutive** [*occurrences*] | **immediate** | **never** | **xofy**[*x-value y-value*]}] [**threshold-value** *upper-threshold lower-threshold*]]] **Example:** `Router(config-tplt-icmp-ech)# react rtt action-type trapOnly threshold-type consecutive 3`	(Optional) Defines proactive threshold monitoring and reaction parameters.
Step 6	**exit** **Example:** `Router(config-tplt-icmp-ech)# exit`	Exits this configuration mode.
Step 7	**ip sla logging traps** **Example:** `Router(config)# ip sla logging traps`	(Optional) Enables IP SLAs syslog notifications from CISCO-RTTMON-MIB. • Required if the **react** command is configured with the **trapOnly** keyword.
Step 8	Do one of the following: • **snmp-server enable traps rtr**	(Optional) Enables system to generate CISCO-RTTMON-MIB traps.

	Command or Action	Purpose

	• **snmp-server enable traps syslog**
	
	Example:
	`Router(config)# snmp-server enable traps rtr`
	
	Example:
	`Router(config)# snmp-server enable traps syslog`
		or
		(Optional) Enables system to generate CISCO-SYSLOG-MIB traps.

Step 9	**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:** `Router(config)# snmp-server host 10.1.1.1 public syslog`	(Optional) Sends traps to a remote host. • Required if the**snmp-server enable traps** command is configured.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits this configuration mode and goes to privileged EXEC mode.

Configuring an Auto-Measure Group on the Source Device

Before You Begin

Before configuring an auto-measure group, you must configure the auto IP SLAs template, endpoint list, and scheduler to be specified for the auto-measure group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla auto group type ip** *group-name*
4. **description** *description*
5. **template** {**icmp-echo** | **icmp-jitter** | **tcp-connect** | **udp-echo** | **udp-jitter**} *template-name*
6. **destination** *template-name*
7. **schedule** *schedule-id*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla auto group type ip** *group-name* **Example:** `Router(config)# ip sla auto group type ip customergroup`	Begins configuring an auto-measure group and enters IP SLA auto-measure group configuration mode.
Step 4	**description** *description* **Example:** `Router(config-am-group)# description customergroup`	(Optional) Adds descriptive text to the auto-measure group being configured.
Step 5	**template** {**icmp-echo** | **icmp-jitter** | **tcp-connect** | **udp-echo** | **udp-jitter**} *template-name* **Example:** `Router(config-am-group)# template icmp-echo tmp-icmpecho`	(Optional) Specifies a template for the group being configured. • The default operation type for group is ICMP jitter.
Step 6	**destination** *template-name* **Example:** `Router(config-am-group)# destination edgehosts-epl`	Specifies an endpoint list for the group being configured.
Step 7	**schedule** *schedule-id* **Example:** `Router(config-am-group)# schedule theschedule`	Specifies a scheduler for the group being configured.
Step 8	**end** **Example:** `Router(config-am-group)# end`	Returns to privileged EXEC mode.

Configuration Examples for Auto IP SLAs in IP SLAs Engine 3.0

Example IP SLAs Auto-Measure Group

The following sample output shows the configuration of an auto-measure group (customergroup) for an ICMP echo operation as well as information about each operation that was created for each destination in the endpoint list (edgehosts-epl):

```
Router# show running-config
.
.
.
ip sla auto template type ip icmp-echo tmp-icmpecho
ip sla auto endpoint-list type ip edgehosts-epl
ip-address 10.1.1.1 port 6500
ip sla auto group type ip customergroup
schedule theschedule
template icmp-echo tmp-icmpecho
destination edgehosts-epl
ip sla auto schedule theschedule
start-time now
!
Router# show ip sla auto group

Group Name: customergroup
    Description:
    Activation Trigger: Immediate
    Destination: edgehosts-epl
    Schedule: theschedule
    Measure Template: tmp-icmpecho(icmp-echo)
IP SLAs auto-generated operations of group customergroup
  sno     oper-id         type             dest-ip-addr/port
    1   299389922      icmp-echo             10.1.1.1/NA
```

Example Configuring Auto IP SLAs Using SNMP

Example MIB Strings for Auto IP SLAs Template

This example shows the MIB string for configuring a template (tmp-icmpecho) for an ICMP echo operation.

The Cisco IOS command for creating this template is **ip sla auto template type ip** *operation template-name*.

The syntax for the MIB string to create an template is as follows:

MIBobject **.** *length-of-name* **.** *ASCII-value-for-each-character-in-name*

The period (.) between each variable is required and a period (.) is also required between each character in the *ASCII-value-for-each-character-in-the-name* argument.

The following MIB strings are for configuring the template (tmp-icmpecho) in this example. See the table below for the appropriate values for each variable in the string.

- To create the template, the MIB string is:

```
cipslaIcmpEchoTmplRowStatus.
12.
116.109.108.45.105.99.109.112.101.99.104.111 -i 4
```

- To display this configuration in the running configuration, the MIB string is:

```
cipslaIcmpEchoTmplStorageType.
12.
116.109.108.45.105.99.109.112.101.99.104.111 -i 3
```

- To configure a value of 255 for ToS in this same template, the MIB string is:

```
cipslaIcmpEchoTmplTOS.
12.
116.109.108.45.105.99.109.112.101.99.104.111 -g 255
```

Table 1: Values for MIB Strings to Create, Display, and Modify Configuration of Template

Variable	Value	Comments
MIBobject	cipslaIcmpEchoTmplRowStatus	Creates a template.
	cipslaIcmpEchoTmplStorageType	Displays MIB string in running configuration.
	cipslaIcmpEchoTmplTOS	Adds value for ToS to template.
length-of-nam e	12	To determine this value, count the number of characters in the name. The name of this template in this example is: tmp-icmpecho.
ASCII-value-for-each-character-in-name	116.109.108.45.105.99.109.112.101.99.104.111	To determine this value, replace each letter of the template name with its ASCII equivalent. Each of the ASCII characters must be separated by a period (.). The name of this template in this example is: tmp-icmpecho.

Example MIB Strings for Auto IP SLAs Endpoint List

The following example shows the MIB string for configuring an endpoint list (edgehosts-epl).

The Cisco IOS commands for configuring an endpoint list are:

- **ip sla auto endpoint-list type ip** *template-name*
- **ip-address** *ip-address* **port** *port*

The syntax for the MIB string to configure an endpoint list is as follows:

MIBobject **.** *length-of-name* **.** *ASCII-value-for-each-character-in-name* **.** *type* **.** *number-of-octets-in-address* **.** *destination-address.destination-port*

The period (.) between each variable is required and a period (.) is also required between each character in the *ASCII-value-for-each-character-in-the-name argument*.

The following MIB string is for configuring an endpoint list (edgehosts-echo) with a destination IP address (10.1.1.1) and port (6500). See the table below for the appropriate values for each variable in the string

```
cipslaIPEndPointRowStatus.13.101.100.103.101.104.111.115.116.115.45.101.112.108.1.4.10.1.11.1.6500
 -i 4
```

Table 2: Values for MIB Strings to Configure an IP Address and Port in an Auto IP SLAs Endpoint List

Variable	Value	Comments
MIBobject	cipslaIPEndPointRowStatus	Creates an endpoint list.
length-of-template-nam e	13	To determine this value, count the number of characters in the name The name of this endpoint list in this example is: edgehosts-echo.
ASCII-value-for-each-character-in-name	101.100.103.101.104.111.115.116.115.45.101.112.108	To determine this value, replace each letter of the name with its ASCII equivalent. Each of the ASCII characters must be separated by a period (.). The name of this endpoint list in this example is: edgehosts-echo.
type	1	Type is: IP.
number- of-octets-in-address	4	IPv4 address (10.101.101.1) has four octets.
destination-address	10.1.1.1	Standard URL.
destination-port	6500	Port number.

Example MIB Strings for Auto IP SLAs Scheduler

This example shows the MIB strings for configuring a scheduler (theschedule) with an immediate start time.

The Cisco IOS commands for configuring a scheduler are:

- **ip sla auto schedule** *schedule-id*
- **start-time now**

The syntax for the MIB string to create a scheduler is as follows:

MIBobject **.** *length-of-name* **.** *ASCII-value-for-each-character-in-name*

The period (**.**) between each variable is required and a period (.) is also required between each character in the *ASCII-value-for-each-character-in-the-name* argument.

The following MIB strings are for configuring the scheduler (theschedule) in this example. See the table below for the appropriate values for each variable in the string.

- To create the scheduler, the MIB string is:

```
cipslaAutoGroupSchedRowStatus.11.116.104.101.115.99.104.101.100.117.108.101 -i 4
```

- To configure the start time in the schedule, the MIB string is

```
cipslaAutoGroupSchedStartTime.11.116.104.101.115.99.104.101.100.117.108.101 -g 0
```

Table 3: Values for MIB Strings to Configure an Auto IP SLAs Scheduler

Variable	Value	Comments
MIBobject	cipslaAutoGroupSchedRowStatus	Creates a scheduler.
	cipslaAutoGroupSchedStartTime	Configures start time.
length-of-nam e	11	To determine this value, count the number of characters in the name. The name of this scheduler in this example is: theschedule.
ASCII-value-for-each-character-in-name	116.104.101.115.99.104.101.100.117.108.101	To determine this value, replace each letter of the template name with its ASCII equivalent. Each of the ASCII characters must be separated by a period (.). The name of this template in this example is: theschedule.

Example MIB Strings for an Auto-Measure Group

This example shows the MIB strings for configuring an auto-measure group (customergroup).

The Cisco IOS commands for configuring a group are:

- **ip sla auto group type ip** *group-name*
- **template icmp-echo** *template-name*
- **destination** *template-name*
- **schedule** *schedule-id*

The syntax for the MIB string to configure a group is as follows:

MIBobject **.** *type* **.** *length-of-name* **.** *ASCII-value-for-each-character-in-name*

The period (**.**) between each variable is required and a period (.) is also required between each character in the *ASCII-value-for-each-character-in-the-name* argument.

Use the following MIB strings to configure the group (customergroup) in this example via SNMP. See the table below for the appropriate values for each variable in the string.

- To create the group, the MIB string is:

```
cipslaAutoGroupRowStatus.1.13.99.117.115.116.111.109.101.114.71.114.111.117.112 -i 4 \
```

- To specify the template for this group, the MIB string is

```
cipslaAutoGroupOperType.1.13.99.117.115.116.111.109.101.114.71.114.111.117.112 -i 1
cipslaAutoGroupOperTemplateName.1.13.99.117.115.116.111.109.101.114.71.114.111.117.112 -D
"tml-icmpecho"
```

- To specify the endpoint list for this group, the MIB string is:

```
cipslaAutoGroupDestEndPointName.1.13.99.117.115.116.111.109.101.114.71.114.111.117.112 -D
"edgehosts-epl"
```

- To specify a scheduler for this group, the MIB string is:

```
cipslaAutoGroupSchedulerId.1.13.99.117.115.116.111.109.101.114.71.114.111.117.112 -D
"theschedule"
```

Table 4: Values for MIB Strings to Configure an IP SLAs Auto-Measure Group

Variable	Value	Comments
MIBobject	cipslaAutoGroupSchedRowStatus	Creates a group
	cipslaAutoGroupOperType	Defines operation type for group.
	cipslaAutoGroupOperTemplateName	Specifies a template.
	cipslaAutoGroupDestEndPointName	Specifies an endpoint list
	cipslaAutoGroupSchedulerId	Specifies a scheduler.
type	1	Type is: IP.
length-of-nam e	13	To determine this value, count the number of characters in the name. The name of this group in this example is: customergroup.

Variable	Value	Comments
ASCII-value-for-each-character-in-name	99.117.115.116.111.109.101.114.71.114.111.117.112	To determine this value, replace each letter of the template name with its ASCII equivalent. Each of the ASCII characters must be separated by a period (.). The name of this group in this example is: customergroup.

Example Using SNMP Commands to Create an Auto IP SLAs Operation

The following example shows how to use SNMP commands to create the configuration in the preceding examples using the setany application:

```
setany -v2c <ipaddress> public
cipslaIcmpEchoTmplStorageType.12.116.109.108.45.105.99.109.112.101.99.104.111 -i 3 \


               cipslaIcmpEchoTmplRowStatus.12.116.109.108.45.105.99.109.112.101.99.104.111
 -i 4



setany -v2c <ipaddress> public
cipslaIcmpEchoTmplTOS.12.116.109.108.45.105.99.109.112.101.99.104.111 -g 255



setany -v2c <ipaddress> public
cipslaIPEndPointRowStatus.13.101.100.103.101.104.111.115.116.115.45.101.112.108.1.4.10.1.1.1.6500
 -i 4 \



cipslaIPEndPointStorageType.13.101.100.103.101.104.111.115.116.115.45.101.112.108.1.4.10.1.1.1.6500
 -i 3



setany -v2c <ipaddress> public
cipslaAutoGroupSchedRowStatus.11.116.104.101.115.99.104.101.100.117.108.101 -i 4 \


cipslaAutoGroupSchedStorageType.11.116.104.101.115.99.104.101.100.117.108.101 -i 3



setany -v2c <ipaddress> public
cipslaAutoGroupSchedStartTime.11.116.104.101.115.99.104.101.100.117.108.101 -g 0



setany -v2c <ipaddress> public
cipslaAutoGroupRowStatus.1.13.99.117.115.116.111.109.101.114.71.114.111.117.112 -i 4 \



cipslaAutoGroupStorageType.1.13.99.117.115.116.111.109.101.114.71.114.111.117.112 -i 3



setany -v2c <ipaddress> public
```

```
cipslaAutoGroupSchedulerId.1.13.99.117.115.116.111.109.101.114.71.114.111.117.112 -D
"theschedule"


setany -v2c <ipaddress> public
cipslaAutoGroupOperType.1.13.99.117.115.116.111.109.101.114.71.114.111.117.112 -i 1 \


cipslaAutoGroupOperTemplateName.1.13.99.117.115.116.111.109.101.114.71.114.111.117.112 -D
"tml-icmpecho"


setany -v2c <ipaddress> public
cipslaAutoGroupDestEndPointName.1.13.99.117.115.116.111.109.101.114.71.114.111.117.112 -D
"edgehosts-epl"
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP SLAs commands	Cisco IOS IP SLAs Command Reference

Standards and RFCs

Standard/RFC	Title
No new or modified standards or RFCs are supported by this feature, and support for existing standards has not been modified by features in this document.	--

MIBs

MIB	MIBs Link
• CISCO-IPSLA-VIDEO-MIB • CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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 Auto IP SLAs in IP SLAs Engine 3.0

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

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

Table 5: Feature Information for Auto IP SLAs in IP SLAs Engine 3.0

Feature Name	Releases	Feature Information
Cisco IOS IP SLAs Engine 3.0	15.1(1)T	Auto IP SLAs in Cisco IOS IP Service Level Agreements (SLAs) Engine 3.0 enables you to combine a template, endpoint list, and scheduler to easily configure multiple operations, including operations for proactive threshold monitoring, and provides auto registration which enables the source to discover the destination's address for building an endpoint list. The following commands were introduced or modified: **access-list**, **ageout**, **codec**(tplt), **clock-tolerance ntp oneway**, **description**(IP SLA), **control**, **destination**(am-group),**discover**(epl), **frequency**(am-schedule), **history buckets-kept**, **history distributions-of-statistics-kept**, **history enhanced**, **history filter**, **history hours-of-statistics-kept**, **history lives-kept**, **history statistics-distribution-interval**, **interval**(params), **ip-address** (epl), **ip sla auto discovery**, **ip sla auto endpoint-list**, **ip sla auto group**, **ip sla auto schedule**, **ip sla auto template**, **ip sla responder auto-register**,**life**, **measurement-retry**, **num-packets**, **operation-packet priority**, **parameters**, **precision**, **probe-interval**, **react**, **request-data-size**, **schedule**, **show ip sla auto discovery**, **show ip sla auto group**, **show ip sla auto summary-statistics**, **show ip sla auto schedule**, **show ip sla auto endpoint-list**, **show ip sla auto template**, **source-ip** (tplt), **source-port**, **start-time**, **template**(am-group), **threshold** (IP SLA), **tos**(IP SLA),**timeout**(IP SLA), **verify-data**(IP SLA), **vrf**(IP SLA).