



# Stream Control Transmission Protocol (SCTP), Release 2

---

## Feature History

Release	Modification
12.2(4)T	SCTP Release 1 was integrated into Cisco IOS Release 12.2(4)T.
12.2(8)T	SCTP Release 2 was integrated into Cisco IOS 12.2(8)T. SCTP Release 2 includes updated output for these commands: <ul style="list-style-type: none"><li>• <a href="#">show ip sctp association parameters</a></li><li>• <a href="#">show ip sctp association statistics</a></li></ul>

This document describes the Stream Control Transmission Protocol (SCTP) feature in Cisco IOS Release 12.2(8)T and includes the following sections:

- [Feature Overview, page 1](#)
- [Supported Platforms, page 3](#)
- [Supported Standards, MIBs, and RFCs, page 3](#)
- [Prerequisites, page 3](#)
- [Configuration Tasks, page 4](#)
- [Troubleshooting Tips, page 4](#)
- [Configuration Examples, page 22](#)
- [Command Reference, page 23](#)
- [Glossary, page 70](#)

## Feature Overview

SCTP is a reliable datagram-oriented IP transport protocol specified by [RFC 2960](#). It provides the layer between an SCTP user application and an unreliable end-to-end datagram service such as IP. The basic service offered by SCTP is the reliable transfer of user datagrams between peer SCTP users. It performs this service within the context of an association between two SCTP hosts. SCTP is connection-oriented, but SCTP association is a broader concept than the TCP connection, for example.

SCTP provides the means for each SCTP endpoint to provide its peer with a list of transport addresses, such as address and UDP port combinations. This list is provided during association startup and shows the transport addresses through which the endpoint can be reached and from which messages originate. The SCTP association includes transfer over all the possible source and destination combinations that might be generated from the two endpoint lists (also known as multihoming).

SCTP is not explicitly configured on routers, but it underlies several Cisco applications. The commands described in this document are useful for troubleshooting when SCTP issues are suspected as the cause of problems.

## Benefits

SCTP provides the following services and features:

- Acknowledged reliable nonduplicated transfer of user data
- Application-level segmentation to conform to the maximum transmission unit (MTU) size
- Sequenced delivery of user datagrams within multiple streams
- Optional multiplexing of user datagrams into SCTP datagrams
- Enhanced reliability through support of multihoming at either end or both ends of the association
- Congestion avoidance and resistance to flooding and masquerade attacks

## Related Features and Technologies

Two upper-layer Cisco applications that make use of SCTP are described in these documents:

- [IP Transfer Point \(ITP\)](#), Cisco IOS Release 12.2(2)MB
- [PRI Backhaul Using the Stream Control Transmission Protocol and the ISDN Q.921 User Adaptation Layer](#), Cisco IOS Release 12.2(4)T

## Related Documents

- The **service timestamps** command reference entry in the “[Troubleshooting and Fault Management Commands](#)” chapter in the “[System Management Commands](#)” part of the *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.2
- [RFC 2960, Stream Control Transmission Protocol \(SCTP\)](#)
- [Cisco IOS Voice, Video, and Fax Configuration Guide](#), Release 12.2
- [Cisco IOS Voice, Video, and Fax Command Reference](#), Release 12.2

# Supported Platforms

- Cisco 3660 using the PRI Backhaul Using the Stream Control Transmission Protocol and the ISDN Q.921 User Adaptation Layer feature

## Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to [cco-locksmith@cisco.com](mailto:cco-locksmith@cisco.com). An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at <http://www.cisco.com/register>.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

<http://www.cisco.com/go/fn>

# Supported Standards, MIBs, and RFCs

## Standards

No new or modified standards are supported by this feature.

## MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

## RFCs

- *RFC2960, Stream Control Transmission Protocol (SCTP)*

# Prerequisites

- Cisco IOS Release 12.2(8)T or a later release

# Configuration Tasks

None

## Troubleshooting Tips

This section describes the Cisco IOS commands that provide troubleshooting assistance for SCTP associations and instances. There are three types of SCTP commands:

- [Show Commands](#)
- [Debug Commands](#)
- [Clear Command](#)

## Show Commands

This section explains the use of the following commands that display information about SCTP associations and instances:

- [show ip sctp association list](#)
- [show ip sctp association parameters](#)
- [show ip sctp association statistics](#)
- [show ip sctp errors](#)
- [show ip sctp instances](#)
- [show ip sctp statistics](#)



### Note

---

SCTP commands that display statistical information show only the information that is available since the last time a [clear ip sctp statistics](#) command was executed. The [clear ip sctp statistics](#) command clears all SCTP statistics, both those compiled for individual associations and those compiled overall.

---

## show ip sctp association list

The **show ip sctp association list** command provides the current SCTP association and instance identifiers, the current state of SCTP associations, and the local and remote port numbers and addresses that are used in the associations. The example below shows two current associations that are in the established state. Each association belongs to a different instance, as noted by their instance identifiers.

```
Router# show ip sctp association list

*** SCTP Association List ***

AssocID: 0, Instance ID: 0
Current state: ESTABLISHED
Local port: 8787, Addrs: 10.1.0.2 10.2.0.2
Remote port: 8787, Addrs: 10.5.0.4 10.6.0.4

AssocID: 1, Instance ID: 1
Current state: ESTABLISHED
Local port: 6790, Addrs: 10.1.0.2 10.2.0.2
Remote port: 6789, Addrs: 10.5.0.4 10.6.0.4
```

## show ip sctp association parameters

The **show ip sctp association parameters** command displays parameter values for the specified association. It requires an association identifier as an argument. Association identifiers can be obtained from the output of the **show ip sctp association list** command.

There are many parameters that are defined for each association. Some are configured parameters, and others are calculated. There are four main groupings of parameters displayed by this command:

- Association configuration parameters
- Destination address parameters
- Association boundary parameters
- Current association congestion parameters

The association configuration section displays information similar to that in the **show ip sctp association list** command, including association identifiers, state, and local and remote port and address information. The current primary destination is also displayed. The uptime for the current association establishment is included in this section. This tells you how long the current association has been established and thus gives an indication of the stability of the system.

The destination address parameters section displays information associated with each destination address. Many of the address parameters are dynamically calculated and therefore give an indication of the state of the network for each destination. The state of the destination address is either ACTIVE or FAILED, depending on whether this destination address is currently accessible. The heartbeats and timeout are configured values (configured by the application using SCTP), as is the ToS (type of service, or IP precedence) value. The ToS value shows the bits that are used in the IP header of each datagram that is sent out. The maximum transmission unit (MTU) field shows the currently configured MTU value from the corresponding interface that is used to reach the destination address. The values for congestion window (cwnd), slow-start threshold (ssthresh), retransmission timeout (RTO), round-trip time (RTT), and so forth are all calculated values, continually updated based on feedback from the network, as specified in RFC 2960. The cwnd parameter indicates how much data can be outstanding in the network for that particular destination. The RTO parameter is the retransmit timeout value. Finally, the num retrans and num times failed parameters give the current number for how many times data has been retransmitted to that address and how many times the address has been marked as failed, respectively. These parameters give an indication of how steady that particular address is and has been.

The third group of parameters are boundary parameters that usually do not change over the life of an association, although the calling application can change some of them if it wants. The verification tags (vertags) are chosen during association initialization and do not change. The number of inbound and outbound streams also do not change. The max retrans numbers show the maximum number of times that chunks are retransmitted. The association retransmit limit is the number of times that any particular chunk is retransmitted before a declaration is made that the association failed. The declaration indicates that the chunk could not be delivered on any address. Init retrans is the number of times that the chunks for initialization are retransmitted before a declaration that the attempt to establish the association failed. And the path retrans value is the maximum number of times that chunks are transmitted to any particular destination address before that address is declared failed. The cumulative SACK (cumsack) timeout specifies the maximum time that a SACK is delayed while waiting to bundle with data chunks. The bundle timeout value is the maximum time that data chunks are delayed during attempts to bundle with other data chunks. And finally, the min and max RTO values define the minimum and maximum retransmit timeout values that are allowed for the association.

The last section of parameters covers the association congestion parameters. These parameters are displayed in the last two lines of output (which are not separated from the previous section by a blank line). The receiver window (rwnd) parameters indicate the current receive window parameters for each

peer and also show the lowest value `rwnd` has been since the statistics were last cleared or since this association was established. This value can indicate if the traffic flow is being slowed down because of lack of buffer space.

The final line of parameters defines congestion information. The upper-layer protocol (ULP) can configure multiple congestion levels if desired, and when these levels are reached, SCTP informs the ULP. This gives the ULP feedback about the traffic flow and can be used to throttle or redirect traffic if desired. The parameters here indicate the number of levels defined, the current congestion level, and the highest level that has been reached. The specific thresholds for these levels are controlled by the ULP, and so interpretation of these levels depends on the specific ULP that is using this association.

The following is an example of the output for the **show ip sctp association parameters** command:

```
Router# show ip sctp association parameters 0

** SCTP Association Parameters **

AssocID: 0 Context: 0 InstanceID: 0
Assoc state: ESTABLISHED Uptime: 00:00:34.280
Local port: 8787
Local addresses: 10.1.0.2 10.2.0.2

Remote port: 8787
Primary dest addr: 10.5.0.4
Effective primary dest addr: 10.5.0.4
Destination addresses:

10.5.0.4: State: ACTIVE
Heartbeats: Enabled Timeout: 30000 ms
RTO/RTT/SRTT: 1000/0/0 ms TOS: 0 MTU: 1500
cwnd: 5000 ssthresh: 18000 outstand: 0
Num retrans: 0 Max retrans: 5 Num times failed: 0

10.6.0.4: State: ACTIVE
Heartbeats: Enabled Timeout: 30000 ms
RTO/RTT/SRTT: 1000/0/0 ms TOS: 0 MTU: 1500
cwnd: 3000 ssthresh: 18000 outstand: 0
Num retrans: 0 Max retrans: 5 Num times failed: 0

Local vertag: DA3C3BD Remote vertag: 4D95E3A
Num inbound streams: 13 outbound streams: 13
Max assoc retrans: 5 Max init retrans: 8
CumSack timeout: 200 ms Bundle timeout: 100 ms
Min RTO: 1000 ms Max RTO: 60000 ms
LocalRwnd: 9000 Low: 6400 RemoteRwnd: 16800 Low: 14900
Congest levels: 0 current level: 0 high mark: 1
```

## show ip sctp association statistics

The **show ip sctp association statistics** command requires an association identifier argument. The association identifier can be obtained from the output of the **show ip sctp association list** command.

This command displays various statistics about the specified association. The first numbers show the total number of chunks, both data and control, sent and received. The second group of statistics focuses on the data chunks sent, showing the total number sent, the number retransmitted, the number that were ordered and unordered, the average number that were bundled together, and the total bytes sent. The third group of statistics focuses on the data chunks received. It displays the total number received and the number discarded (because of duplicates), the number of ordered and unordered chunks received, the average number of chunks that were bundled, the number of bytes received, and the number of sequenced

chunks that were received out of order. The last section indicates how many datagrams have been sent, received, or are ready to be received by the calling application or ULP. The ULP statistics may be different from the chunk statistics if the datagrams are large and have been segmented by SCTP.

The following example was taken from a network with known dropped packets in one direction. The number of total chunks sent and received is larger than the number of data chunks sent and received because it also includes the control chunks sent. The number of chunks received out of sequence indicates that there are some chunks not being received in the correct order. However, the number of chunks discarded is zero, indicating that only one copy of each is arriving at this peer (some chunks are probably being dropped and the peer is retransmitting them, but there are no duplicates being received). The number of chunks being retransmitted is zero, indicating that there is no network problem in the direction of sending from this peer to the remote.

```
Router# show ip sctp association statistics 0

** Sctp Association Statistics **

AssocID/InstanceID: 0/0
Current State: ESTABLISHED
Control Chunks
  Sent: 1009  Rcvd: 988
Data Chunks Sent
  Total: 18073  Retransmitted: 0
  Ordered: 9095  Unordered: 8978
  Avg bundled: 9  Total Bytes: 1807300
Data Chunks Rcvd
  Total: 18073  Discarded: 0
  Ordered: 9095  Unordered: 8978
  Avg bundled: 9  Total Bytes: 1807300
  Out of Seq TSN: 586
ULP Dgrams
  Sent: 18073  Ready: 18073  Rcvd: 18073
```

## show ip sctp errors

The **show ip sctp errors** command indicates any errors that have been logged since the last time that the statistics were cleared. If no errors have been logged, this is indicated in the output. The following output shows one example in which no errors have been logged, and another in which there have been several different types of errors.

```
Router# show ip sctp errors

*** Sctp Error Statistics ***

No Sctp errors logged.

Router# show ip sctp errors

*** Sctp Error Statistics ***

Communication Lost:          95
Unknown INIT params rcvd:    8
Missing parameters:         18
No room for incoming data:   11
```

## show ip sctp instances

The **show ip sctp instances** command displays information for each of the currently configured instances. The instance number, local port, and address information is displayed. The instance state is either *available* or *deletion pending*. An instance enters the deletion pending state when a request is made to delete it but there are currently established associations for that instance. The instance cannot be deleted immediately and instead enters the pending state. No new associations are allowed in this instance, and when the last association is terminated or fails, the instance is deleted.

The default inbound and outbound stream numbers are used for establishing incoming associations, and the maximum number of associations allowed for this instance is shown. Finally, a snapshot of each existing association is shown, if any exist.

In this example, two current instances are active and available. The first is using local port 8787, and the second is using local port 6790. Instance identifier 0 has one current association, and instance identifier 1 has no current associations.

```
Router# show ip sctp instances

*** Sctp Instances ***

Instance ID: 0 Local port: 8787
Instance state: available
Local addrs: 10.1.0.2 10.2.0.2
Default streams inbound: 1 outbound: 1
Current associations: (max allowed: 6)
  AssocID: 0 State: ESTABLISHED Remote port: 8787
  Dest addrs: 10.5.0.4 10.6.0.4

Instance ID: 1 Local port: 6790
Instance state: available
Local addrs: 10.1.0.2 10.2.0.2
Default streams inbound: 13 outbound: 13

No current associations established for this instance.
Max allowed: 6
```

## show ip sctp statistics

The **show ip sctp statistics** command displays the overall SCTP statistics accumulated since the last **clear ip sctp statistics** command. It includes statistics for all currently established associations, as well as for any that have been terminated. The statistics indicated are similar to those shown for individual associations, with an additional section that shows overall association statistics, such as the number of active and passive associations established. An active establishment is one initiated by the local peer, and a passive establishment is one initiated by a remote peer. If both peers are attempting to establish the association, the initiation of the association is timing-dependent. This section also indicates the number of aborts and shutdowns received and the number of times the T1 and T2 timers expired. These timers are used during the initialization and shutdown procedures, respectively.

```
Router# show ip sctp statistics

** Sctp Overall Statistics **

Control Chunks
  Sent: 7872 Rcvd: 8547
Data Chunks Sent
  Total: 98681 Retransmitted: 5
  Ordered: 50241 Unordered: 48435
  Total Bytes: 9868100
```

```

Data Chunks Rcvd
  Total: 98676  Discarded: 0
  Ordered: 50241  Unordered: 48435
  Total Bytes: 9867600
  Out of Seq TSN: 2845
SCTP Dgrams
  Sent: 17504  Rcvd: 19741
ULP Dgrams
  Sent: 98676  Ready: 98676  Rcvd: 98676

Additional Stats
  Assocs Currently Estab: 0
  Active Estab: 0  Passive Estab: 2
  Aborts: 0  Shutdowns: 0
  T1 Expired: 11  T2 Expired: 0

```

## Debug Commands

This section describes the debug commands available for SCTP.



### Caution

Many SCTP debug commands should be used with extreme caution or not at all in live systems, depending on the amount of traffic, because the extra messages they generate may cause associations to fail. This caution is repeated in descriptions of the commands that may cause disruption to live systems.



### Note

SCTP debug commands display information for all current SCTP associations and cannot be limited to particular associations.

Debugs are often used with time stamps enabled to see the relevant timing of the events indicated. Use the **service timestamps** commands to add time stamps to debug output in the format MMM DD HH:MM:SS, which indicates the date and time according to the system clock. If the system clock has not been set, the date and time are preceded by an asterisk (\*) to indicate that the date and time are probably not correct. To activate millisecond time stamps for debugs, use the following commands in global configuration mode:

```

Router(config)# service timestamps debug datetime msec
Router(config)# service timestamps log datetime msec

```

For more information, see the **service timestamps** command reference entry in the “[Troubleshooting and Fault Management Commands](#)” chapter in the “[System Management Commands](#)” part of the *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.2.

In some of the examples shown below, time stamps have been omitted for clarity of the output.

The following debug commands are available for SCTP:

- **debug ip sctp api**
- **debug ip sctp congestion**
- **debug ip sctp init**
- **debug ip sctp multihome**
- **debug ip sctp performance**
- **debug ip sctp rcvchunks**
- **debug ip sctp rto**

- [debug ip sctp segments](#)
- [debug ip sctp segmentv](#)
- [debug ip sctp signal](#)
- [debug ip sctp sndchunks](#)
- [debug ip sctp state](#)
- [debug ip sctp timer](#)
- [debug ip sctp warnings](#)

In a live system, the debug commands for performance, state, signal, and warnings are the most useful. These debug commands show any association or destination address failures and can be used to monitor the stability of any established associations.

Debug commands other than those for performance, state, signal, and warnings can generate a great deal of output and therefore can cause associations to fail. These commands should be used only in test environments or when there are very low amounts of traffic.

Examples of each type of debug command follow.

## debug ip sctp api

The **debug ip sctp api** command shows all SCTP calls to the application programming interface (API) that are being executed and the parameters associated with these calls.



### Caution

The **debug ip sctp api** command should not be used in a live system that has any significant amount of traffic running because it can generate a lot of traffic, which can cause associations to fail.

```
Router# debug ip sctp api

*Mar 1 00:31:14.211: SCTP: sctp_send: Assoc ID: 1
*Mar 1 00:31:14.211: SCTP:                stream num: 10
*Mar 1 00:31:14.211: SCTP:                bptr: 62EE332C, dptr: 4F7B598
*Mar 1 00:31:14.211: SCTP:                datalen: 100
*Mar 1 00:31:14.211: SCTP:                context: 1
*Mar 1 00:31:14.211: SCTP:                lifetime: 0
*Mar 1 00:31:14.211: SCTP:                unordered flag: FALSE
*Mar 1 00:31:14.211: SCTP:                bundle flag: TRUE
*Mar 1 00:31:14.211: SCTP: sctp_send successful return
*Mar 1 00:31:14.211: SCTP: sctp_receive: Assoc ID: 1
*Mar 1 00:31:14.215: SCTP:                max data len: 100
*Mar 1 00:31:14.215: SCTP: sctp_receive successful return
*Mar 1 00:31:14.215: SCTP: Process Send Request
*Mar 1 00:31:14.951: SCTP: sctp_receive: Assoc ID: 0
*Mar 1 00:31:14.951: SCTP:                max data len: 100
*Mar 1 00:31:14.951: SCTP: sctp_receive successful return
*Mar 1 00:31:14.951: SCTP: sctp_send: Assoc ID: 0
*Mar 1 00:31:14.951: SCTP:                stream num: 12
*Mar 1 00:31:14.951: SCTP:                bptr: 62EE00CC, dptr: 4F65158
*Mar 1 00:31:14.951: SCTP:                datalen: 100
*Mar 1 00:31:14.951: SCTP:                context: 0
*Mar 1 00:31:14.951: SCTP:                lifetime: 0
*Mar 1 00:31:14.951: SCTP:                unordered flag: FALSE
*Mar 1 00:31:14.951: SCTP:                bundle flag: TRUE
*Mar 1 00:31:14.951: SCTP: sctp_send successful return
*Mar 1 00:31:14.951: SCTP: sctp_receive: Assoc ID: 0
*Mar 1 00:31:14.951: SCTP:                max data len: 100
*Mar 1 00:31:14.951: SCTP: sctp_receive successful return
```

## debug ip sctp congestion

The **debug ip sctp congestion** command displays various events related to calculating the current congestion parameters, including congestion window (cwnd) values per destination address and local and remote receiver window (rwnd) parameters. Information is displayed when bundling and sending data chunks, indicating the current cwnd and rwnd values and remote rwnd values, thus showing when data can or can not be sent or bundled. When chunks are acknowledged by the remote peer, the number of bytes outstanding and remote rwnd values are updated.

Information is also displayed when new chunks are received, thus decreasing the local rwnd space, and when chunks are freed because the upper-layer protocol (ULP) is receiving datagrams from SCTP and thus freeing local rwnd space.

```
Router# debug ip sctp congestion
```

```
SCTP: Assoc 0: Slow start 10.6.0.4, cwnd 3000
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7800
SCTP: Assoc 0: Free chunks, local rwnd 9000
SCTP: Assoc 0: Data chunks rcvd, local rwnd 8200
SCTP: Assoc 0: Add Sack, local a_rwnd 8200
SCTP: Assoc 0: Free chunks, local rwnd 9000
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7800
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7000
SCTP: Assoc 0: Add Sack, local a_rwnd 7000
SCTP: Assoc 0: Free chunks, local rwnd 9000
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 14000, cwnd 19500, outstand 0
SCTP: Assoc 0: Bundled 12 chunks, remote rwnd 12800, outstand 1200
SCTP: Assoc 0: Bundling data, next chunk dataLen (100) > remaining mtu size
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 12800, cwnd 19500, outstand 1200
SCTP: Assoc 0: Bundled 12 chunks, remote rwnd 11600, outstand 2400
SCTP: Assoc 0: Bundling data, next chunk dataLen (100) > remaining mtu size
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 11600, cwnd 19500, outstand 2400
SCTP: Assoc 0: Bundled 12 chunks, remote rwnd 10400, outstand 3600
SCTP: Assoc 0: Bundling data, next chunk dataLen (100) > remaining mtu size
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 10400, cwnd 19500, outstand 3600
SCTP: Assoc 0: Bundled 4 chunks, remote rwnd 10000, outstand 4000
SCTP: Assoc 0: No additional chunks waiting.
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7800
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7000
SCTP: Assoc 0: Add Sack, local a_rwnd 7000
SCTP: Assoc 0: Chunk A22F3B45 ack'd, dest 10.5.0.4, outstanding 3900
SCTP: Assoc 0: Chunk A22F3B46 ack'd, dest 10.5.0.4, outstanding 3800
SCTP: Assoc 0: Chunk A22F3B47 ack'd, dest 10.5.0.4, outstanding 3700
SCTP: Assoc 0: Chunk A22F3B48 ack'd, dest 10.5.0.4, outstanding 3600
SCTP: Assoc 0: Chunk A22F3B49 ack'd, dest 10.5.0.4, outstanding 3500
SCTP: Assoc 0: Chunk A22F3B4A ack'd, dest 10.5.0.4, outstanding 3400
SCTP: Assoc 0: Chunk A22F3B4B ack'd, dest 10.5.0.4, outstanding 3300
SCTP: Assoc 0: Chunk A22F3B4C ack'd, dest 10.5.0.4, outstanding 3200
SCTP: Assoc 0: Chunk A22F3B4D ack'd, dest 10.5.0.4, outstanding 3100
SCTP: Assoc 0: Chunk A22F3B4E ack'd, dest 10.5.0.4, outstanding 3000
SCTP: Assoc 0: Chunk A22F3B4F ack'd, dest 10.5.0.4, outstanding 2900
SCTP: Assoc 0: Chunk A22F3B50 ack'd, dest 10.5.0.4, outstanding 2800
SCTP: Assoc 0: Chunk A22F3B51 ack'd, dest 10.5.0.4, outstanding 2700
SCTP: Assoc 0: Chunk A22F3B52 ack'd, dest 10.5.0.4, outstanding 2600
SCTP: Assoc 0: Chunk A22F3B53 ack'd, dest 10.5.0.4, outstanding 2500
SCTP: Assoc 0: Chunk A22F3B54 ack'd, dest 10.5.0.4, outstanding 2400
SCTP: Assoc 0: Chunk A22F3B55 ack'd, dest 10.5.0.4, outstanding 2300
SCTP: Assoc 0: Chunk A22F3B56 ack'd, dest 10.5.0.4, outstanding 2200
```

## debug ip sctp init

The **debug ip sctp init** command shows datagrams and other information related to the initializing of new associations. All initialization chunks are shown, including the INIT, INIT\_ACK, COOKIE\_ECHO, and COOKIE\_ACK chunks. This debug command can be used to see the chunks associated with any initialization sequence, but does not display data chunks sent once the association is established. Therefore, it is safe to use in a live system that has traffic flowing when you have trouble with associations that fail and have to be reestablished.

```
Router# debug ip sctp init
```

```
*Mar 1 00:53:07.279: Sctp Test: Attempting to open assoc to remote port 8787...assoc ID is 0
*Mar 1 00:53:07.279: Sctp: Process Assoc Request
*Mar 1 00:53:07.279: Sctp: Assoc 0: dest addr list:
*Mar 1 00:53:07.279: Sctp:          addr 10.5.0.4
*Mar 1 00:53:07.279: Sctp:          addr 10.6.0.4
*Mar 1 00:53:07.279:
...
*Mar 1 00:53:13.279: Sctp: Assoc 0: Send Init
*Mar 1 00:53:13.279: Sctp:          INIT_CHUNK, len 42
*Mar 1 00:53:13.279: Sctp:          Initiate Tag: B4A10C4D, Initial TSN: B4A10C4D, rwnd 9000
*Mar 1 00:53:13.279: Sctp:          Streams Inbound: 13, Outbound: 13
*Mar 1 00:53:13.279: Sctp:          IP Addr: 10.1.0.2
*Mar 1 00:53:13.279: Sctp:          IP Addr: 10.2.0.2
*Mar 1 00:53:13.279: Sctp:          Supported addr types: 5
*Mar 1 00:53:13.307: Sctp: Process Init
*Mar 1 00:53:13.307: Sctp:          INIT_CHUNK, len 42
*Mar 1 00:53:13.307: Sctp:          Initiate Tag: 3C2D8327, Initial TSN: 3C2D8327, rwnd 18000
*Mar 1 00:53:13.307: Sctp:          Streams Inbound: 13, Outbound: 13
*Mar 1 00:53:13.307: Sctp:          IP Addr: 10.5.0.4
*Mar 1 00:53:13.307: Sctp:          IP Addr: 10.6.0.4
*Mar 1 00:53:13.307: Sctp:          Supported addr types: 5
*Mar 1 00:53:13.307: Sctp: Assoc 0: Send InitAck
*Mar 1 00:53:13.307: Sctp:          INIT_ACK_CHUNK, len 124
*Mar 1 00:53:13.307: Sctp:          Initiate Tag: B4A10C4D, Initial TSN: B4A10C4D, rwnd 9000
*Mar 1 00:53:13.307: Sctp:          Streams Inbound: 13, Outbound: 13
*Mar 1 00:53:13.307: Sctp:          Responder cookie len 88
*Mar 1 00:53:13.307: Sctp:          IP Addr: 10.1.0.2
*Mar 1 00:53:13.307: Sctp:          IP Addr: 10.2.0.2
*Mar 1 00:53:13.311: Sctp: Assoc 0: Process Cookie
*Mar 1 00:53:13.311: Sctp:          COOKIE_ECHO_CHUNK, len 88
*Mar 1 00:53:13.311: Sctp: Assoc 0: dest addr list:
*Mar 1 00:53:13.311: Sctp:          addr 10.5.0.4
*Mar 1 00:53:13.311: Sctp:          addr 10.6.0.4
*Mar 1 00:53:13.311:
*Mar 1 00:53:13.311: Sctp: Instance 0 dest addr list:
*Mar 1 00:53:13.311: Sctp:          addr 10.5.0.4
*Mar 1 00:53:13.311: Sctp:          addr 10.6.0.4
*Mar 1 00:53:13.311:
*Mar 1 00:53:13.311: Sctp: Assoc 0: Send CookieAck
*Mar 1 00:53:13.311: Sctp:          COOKIE_ACK_CHUNK
```

## debug ip sctp multihome

The **debug ip sctp multihome** command shows the source and destination of datagrams in order to monitor the use of the multihome addresses. More than one IP address parameter can be included in an INIT chunk when the INIT sender is multihomed. Datagrams should mostly be sent to the primary destination addresses unless the network is experiencing problems, in which case the datagrams can be sent to the secondary addresses.



### Caution

The **debug ip sctp multihome** command generates one debug line for each datagram sent or received. It should be used with extreme caution in a live network.

```
Router# debug ip sctp multihome

SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 476
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: Assoc 0: Send Data to dest 10.5.0.4
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 476
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 476
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: Assoc 0: Send Data to dest 10.5.0.4
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 476
SCTP: Rcvd s=10.6.0.4 8787, d=10.2.0.2 8787, len 44
SCTP: Sent: Assoc 0: s=10.2.0.2 8787, d=10.6.0.4 8787, len 44
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 476
```

## debug ip sctp performance

The **debug ip sctp performance** command reveals the average number of chunks and datagrams being sent and received per second. Once enabled, the **debug ip sctp performance** command displays this information once every 10 seconds. Note that the averages are cumulative since the last time the statistics were cleared and so may not accurately reflect the number of datagrams and chunks currently being sent and received.

In the following example, when the performance debug was first enabled, it showed a very low rate of traffic. However, it was expected that these numbers were not accurate, so a **clear ip sctp** command was executed. The average numbers adjusted quickly to reflect the accurate amount of flowing traffic.

```
Router# debug ip sctp performance

SCTP Sent: SCTP Dgrams 5, Chunks 28, Data Chunks 29, ULP Dgrams 29
SCTP Rcvd: SCTP Dgrams 7, Chunks 28, Data Chunks 29, ULP Dgrams 29
Chunks Discarded: 0, Retransmitted 0

SCTP Sent: SCTP Dgrams 6, Chunks 29, Data Chunks 30, ULP Dgrams 30
SCTP Rcvd: SCTP Dgrams 7, Chunks 29, Data Chunks 30, ULP Dgrams 30
Chunks Discarded: 0, Retransmitted 0

SCTP Sent: SCTP Dgrams 6, Chunks 29, Data Chunks 31, ULP Dgrams 31
SCTP Rcvd: SCTP Dgrams 7, Chunks 30, Data Chunks 31, ULP Dgrams 31
Chunks Discarded: 0, Retransmitted 0

SCTP Sent: SCTP Dgrams 6, Chunks 30, Data Chunks 31, ULP Dgrams 31
SCTP Rcvd: SCTP Dgrams 7, Chunks 31, Data Chunks 32, ULP Dgrams 31
Chunks Discarded: 0, Retransmitted 0

SCTP Sent: SCTP Dgrams 6, Chunks 31, Data Chunks 32, ULP Dgrams 32
SCTP Rcvd: SCTP Dgrams 7, Chunks 32, Data Chunks 32, ULP Dgrams 32
Chunks Discarded: 0, Retransmitted 0

Router# clear ip sctp statistics

SCTP Sent: SCTP Dgrams 30, Chunks 210, Data Chunks 199, ULP Dgrams 201
SCTP Rcvd: SCTP Dgrams 30, Chunks 208, Data Chunks 198, ULP Dgrams 198
Chunks Discarded: 0, Retransmitted 0

SCTP Sent: SCTP Dgrams 30, Chunks 210, Data Chunks 199, ULP Dgrams 200
SCTP Rcvd: SCTP Dgrams 30, Chunks 209, Data Chunks 199, ULP Dgrams 199
Chunks Discarded: 0, Retransmitted 0

SCTP Sent: SCTP Dgrams 30, Chunks 211, Data Chunks 200, ULP Dgrams 199
SCTP Rcvd: SCTP Dgrams 30, Chunks 209, Data Chunks 198, ULP Dgrams 198
Chunks Discarded: 0, Retransmitted 0
```

## debug ip sctp rcvchunks

The **debug ip sctp rcvchunks** command displays information about chunks that are received. It shows the stream number, sequence number, chunk length, and chunk transmission sequence number (TSN) for each chunk received, and whether the chunk is for a new datagram or is part of a datagram that is already being reassembled. The command output shows whether the datagram is complete after receiving this chunk or not and, if it is complete, whether it is in sequence within the specified stream and can be delivered to the ULP. It shows the SACKs that are sent back to the remote, indicating the cumulative TSN acknowledged, the number of fragments included, and that the datagram is received by the ULP.



### Caution

The **debug ip sctp rcvchunks** command generates multiple debug lines for each chunk received. It should be used with extreme caution in a live network.

In the following example, a segmented datagram is received in two chunks, for stream 0 and sequence number 0. The length of the first chunk is 1452, and the second is 1 byte. The first chunk indicates that it is for a new datagram, but the second chunk indicates that it is part of an existing datagram that is already being reassembled. When the first chunk is processed, it is noted to be in sequence, but is not complete and so cannot be delivered yet. When the second chunk is received, the datagram is both in

sequence and complete. The application receives the datagram, and a SACK is shown to acknowledge that both chunks were received with no missing chunks indicated (that is, with no fragments).

```
Router# debug ip sctp rcvchunks
```

```
SCTP: Assoc 0: New chunk (0/0/1452/2C33D822) for new dgram (0)
SCTP: Assoc 0: dgram (0) is in seq
SCTP: Assoc 0: Add Sack Chunk, CumTSN=2C33D822, numFrgs=0
SCTP: Assoc 0: New chunk (0/0/1/2C33D823) for existing dgram (0)
SCTP: Assoc 0: dgram (0) is complete
SCTP: Assoc 0: ApplRecv chunk 0/0/1452/2C33D822
SCTP: Assoc 0: ApplRecv chunk 0/0/1/2C33D823
SCTP: Assoc 0: Add Sack Chunk, CumTSN=2C33D823, numFrgs=0
```

The following example is taken from a specific test in which chunks are both sent out of sequence and duplicated. The first chunk received is for stream 0, with sequence number 5. The datagram is complete, but is not in sequence because the previously received datagram was sequence number 3. A SACK chunk is sent, indicating that there is a gap after TSN 15755E58. This same chunk is received again, and the debug indicates that this chunk is a duplicate and so is not processed. The next chunk received is sequence number 7, also complete but not in sequence. The number of fragments specified is now 2, because both datagrams 4 and 6 have not been received. The duplicate chunk is discarded again. Sequence number 6 is then received, also complete, but not in sequence. The next earliest datagram received is 5, and even though that is in sequence, datagram 5 is not in sequence because datagram 4 has not been received and so neither 5 nor 6 can be delivered. Thus, there are occasions when the previous sequence number shown is in sequence, but the datagram itself is specified as not in sequence. The SACK sent at that point indicates just one fragment, because datagrams 5 through 7 are all in sequence in a block. Finally, datagram 4 is received. It is complete and in sequence, and datagrams 5 through 7 become in sequence as well, and all the datagrams can be received by the application.

```
Router# debug ip sctp rcvchunks
```

```
SCTP: Assoc 0: New chunk (0/5/50/15755E5A) for new dgram (5)
SCTP: Assoc 0: dgram (5) is complete
SCTP: Assoc 0: dgram (5) is not in seq, prev seq (3)
SCTP: Assoc 0: Add Sack Chunk, CumTSN=15755E58, numFrgs=1
SCTP: Assoc 0: Rcvd duplicate chunk: 0/5/50/15755E5A
SCTP: Assoc 0: Add Sack Chunk, CumTSN=15755E58, numFrgs=1
SCTP: Assoc 0: New chunk (0/7/50/15755E5C) for new dgram (7)
SCTP: Assoc 0: dgram (7) is complete
SCTP: Assoc 0: dgram (7) is not in seq, prev seq (5)
SCTP: Assoc 0: Add Sack Chunk, CumTSN=15755E58, numFrgs=2
SCTP: Assoc 0: Rcvd duplicate chunk: 0/7/50/15755E5C
SCTP: Assoc 0: Add Sack Chunk, CumTSN=15755E58, numFrgs=2
SCTP: Assoc 0: New chunk (0/6/50/15755E5B) for new dgram (6)
SCTP: Assoc 0: dgram (6) is complete
SCTP: Assoc 0: dgram (6) is not in seq, prev seq (5)
SCTP: Assoc 0: Add Sack Chunk, CumTSN=15755E58, numFrgs=1
SCTP: Assoc 0: Rcvd duplicate chunk: 0/6/50/15755E5B
SCTP: Assoc 0: Add Sack Chunk, CumTSN=15755E58, numFrgs=1
SCTP: Assoc 0: New chunk (0/4/50/15755E59) for new dgram (4)
SCTP: Assoc 0: dgram (4) is complete
SCTP: Assoc 0: dgram (4) is in seq
SCTP: Assoc 0: dgram (5) is now in seq
SCTP: Assoc 0: dgram (6) is now in seq
SCTP: Assoc 0: dgram (7) is now in seq
SCTP: Assoc 0: Rcvd duplicate chunk: 0/4/50/15755E59
SCTP: Assoc 0: Add Sack Chunk, CumTSN=15755E5C, numFrgs=0
SCTP: Assoc 0: ApplRecv chunk 0/4/50/15755E59
SCTP: Assoc 0: ApplRecv chunk 0/5/50/15755E5A
SCTP: Assoc 0: ApplRecv chunk 0/6/50/15755E5C
SCTP: Assoc 0: ApplRecv chunk 0/7/50/15755E5B
```

## debug ip sctp rto

The **debug ip sctp rto** command shows any adjustments that are made to the retransmission (retrans) timeout value due either to retransmission of data chunks or to unacknowledged heartbeats.



### Caution

The **debug ip sctp rto** command can generate a great deal of output. It should be used with extreme caution in a live network.

In the following example, there is only one destination address available. Each time the chunk needs to be retransmitted, the retransmission timeout (RTO) value is doubled.

```
Router# debug ip sctp rto

SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 2000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 4000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 8000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 16000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 32000 ms
```

In the next example, there is again only one destination address available. The data chunk is retransmitted several times, and the heartbeat timer also expires, causing the RTO timer to back off as well. Note that the heartbeat timer is expiring along with the data chunk retransmission timer, because SCTP is continually trying to send a chunk on which it can calculate the current round trip time (RTT). Because the data chunk is being retransmitted, an RTT calculation cannot be made on it, and the heartbeat is used instead.

```
Router# debug ip sctp rto

SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 98432842
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 2000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 98432842
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 4000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 98432842
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 8000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, heartbeat rto backoff 16000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 98432842
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 32000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, heartbeat rto backoff 60000 ms
```

## debug ip sctp segments

The **debug ip sctp segments** output shows every datagram that is sent or received and the chunks that are contained in each. The segment debug command has two forms: simple and verbose. This is the simple form of the segment output, and it shows basic information for each chunk type. See the [debug ip sctp segmentv](#) command for the verbose form of this output.



### Caution

The **debug ip sctp segments** command generates several lines of output for each datagram sent or received. It should be used with extreme caution in a live network.

The following output shows an example in which an association is established, a few heartbeats are sent, the remote endpoint fails, and the association is restarted.

Router# **debug ip sctp segments**

```

SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 56
SCTP: INIT_CHUNK, Tag: 3C72A02A, TSN: 3C72A02A
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 56
SCTP: INIT_CHUNK, Tag: 13E5AD6C, TSN: 13E5AD6C
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 136
SCTP: INIT_ACK_CHUNK, Tag: 3C72A02A, TSN: 3C72A02A
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 100
SCTP: COOKIE_ECHO_CHUNK, len 88
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 16
SCTP: COOKIE_ACK_CHUNK
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 52
SCTP: HEARTBEAT_CHUNK
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 52
SCTP: HEARTBEAT_CHUNK
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 52
SCTP: HEARTBEAT_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 56
SCTP: INIT_CHUNK, Tag: 4F2D8235, TSN: 4F2D8235
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 136
SCTP: INIT_ACK_CHUNK, Tag: 7DD7E424, TSN: 7DD7E424
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 100
SCTP: COOKIE_ECHO_CHUNK, len 88
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 16
SCTP: COOKIE_ACK_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 144
SCTP: SACK_CHUNK, TSN ack: 7DD7E423, rwnd 18000, num frags 0
SCTP: DATA_CHUNK, 4/0/100/4F2D8235
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 4F2D8235, rwnd 8900, num frags 0
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 128
SCTP: DATA_CHUNK, 4/0/100/7DD7E424
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 7DD7E424, rwnd 17900, num frags 0
SCTP: Recv: Assoc 0: s=10.6.0.4 8787, d=10.2.0.2 8787, len 44
SCTP: HEARTBEAT_CHUNK
SCTP: Sent: Assoc 0: s=10.2.0.2 8787, d=10.6.0.4 8787, len 44
SCTP: HEARTBEAT_ACK_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 128
SCTP: DATA_CHUNK, 7/0/100/4F2D8236
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 144
SCTP: SACK_CHUNK, TSN ack: 4F2D8236, rwnd 9000, num frags 0
SCTP: DATA_CHUNK, 7/0/100/7DD7E425
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 7DD7E424, rwnd 18000, num frags 0
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 7DD7E425, rwnd 17900, num frags 0
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 128
SCTP: DATA_CHUNK, 4/1/100/4F2D8237

```

## debug ip sctp segmentv

The **debug ip sctp segmentv** command output shows every datagram that is sent or received and the chunks that are contained in each. This is the verbose form of the output, and it shows detailed information for each chunk type (see the [debug ip sctp segments](#) command for the simple form output).



### Caution

The **debug ip sctp segmentv** command generates multiple lines of output for each datagram sent and received. It should be used with extreme caution in a live network.

The following output shows an example in which an association is established, a few heartbeats are sent, the remote endpoint fails, and the association is restarted.

Router# **debug ip sctp segmentv**

```
SCTP: Sent:  Assoc 0: s=10.1.0.2  8787, d=10.5.0.4  8787, len 56, ver tag 0
SCTP:      INIT_CHUNK, len 42
SCTP:      Initiate Tag: B131ED6A, Initial TSN: B131ED6A, rwnd 9000
SCTP:      Streams Inbound: 13, Outbound: 13
SCTP:      IP Addr: 10.1.0.2
SCTP:      IP Addr: 10.2.0.2
SCTP:      Supported addr types: 5
SCTP: Recv:  Assoc 0: s=10.5.0.4  8787, d=10.1.0.2  8787, len 56, ver tag 0
SCTP:      INIT_CHUNK, len 42
SCTP:      Initiate Tag: 5516B2F3, Initial TSN: 5516B2F3, rwnd 18000
SCTP:      Streams Inbound: 13, Outbound: 13
SCTP:      IP Addr: 10.5.0.4
SCTP:      IP Addr: 10.6.0.4
SCTP:      Supported addr types: 5
SCTP: Sent:  Assoc NULL: s=10.1.0.2  8787, d=10.5.0.4  8787, len 136, ver tag 5516B2F3
SCTP:      INIT_ACK_CHUNK, len 124
SCTP:      Initiate Tag: B131ED6A, Initial TSN: B131ED6A, rwnd 9000
SCTP:      Streams Inbound: 13, Outbound: 13
SCTP:      Responder cookie len 88
SCTP:      IP Addr: 10.1.0.2
SCTP:      IP Addr: 10.2.0.2
SCTP: Recv:  Assoc 0: s=10.5.0.4  8787, d=10.1.0.2  8787, len 100, ver tag B131ED6A
SCTP:      COOKIE_ECHO_CHUNK, len 88
SCTP: Sent:  Assoc NULL: s=10.1.0.2  8787, d=10.5.0.4  8787, len 16, ver tag 5516B2F3
SCTP:      COOKIE_ACK_CHUNK
SCTP: Recv:  Assoc 0: s=10.5.0.4  8787, d=10.1.0.2  8787, len 144, ver tag B131ED6A
SCTP:      SACK_CHUNK, len 16
SCTP:      TSN ack: (0xB131ED69)
SCTP:      Rcv win credit: 18000
SCTP:      Num frags: 0
SCTP:      DATA_CHUNK, flags 3, chunkLen 116
SCTP:      DATA_CHUNK, 0/0/100/5516B2F3
SCTP: Sent:  Assoc 0: s=10.1.0.2  8787, d=10.5.0.4  8787, len 28, ver tag 5516B2F3
SCTP:      SACK_CHUNK, len 16
SCTP:      TSN ack: (0x5516B2F3)
SCTP:      Rcv win credit: 8900
SCTP:      Num frags: 0
SCTP: Sent:  Assoc 0: s=10.1.0.2  8787, d=10.5.0.4  8787, len 128, ver tag 5516B2F3
SCTP:      DATA_CHUNK, flags 3, chunkLen 116
SCTP:      DATA_CHUNK, 0/0/100/B131ED6A
SCTP: Recv:  Assoc 0: s=10.6.0.4  8787, d=10.2.0.2  8787, len 44, ver tag B131ED6A
SCTP:      HEARTBEAT_CHUNK
SCTP: Sent:  Assoc 0: s=10.2.0.2  8787, d=10.6.0.4  8787, len 44, ver tag 5516B2F3
SCTP:      HEARTBEAT_ACK_CHUNK
SCTP: Recv:  Assoc 0: s=10.5.0.4  8787, d=10.1.0.2  8787, len 28, ver tag B131ED6A
SCTP:      SACK_CHUNK, len 16
```

## debug ip sctp signal

The **debug ip sctp signal** command shows signals that are sent from SCTP to the application or ULP. These signals inform the ULP of state transitions for associations or destination addresses. There is also a signal sent to the ULP when new data is available to be received, but this signal is not shown in the example output below because it occurs infrequently. This debug command can be used to see if the current associations are stable. Because it does not generate output except on state transitions, it is safe to use in a live environment. It still should be used with caution, however, depending on the number of associations being handled by the system and the stability of the network.

The **debug ip sctp state** command is often used at the same time as the **debug ip sctp signal** command. Using the two commands together gives good insight into the stability of associations.

In the following example, a new association is requested and established. The peer then restarts the association and notes that the association failed and is being reestablished. The local peer then indicates that the association has failed because it has tried to retransmit the specified chunk more than the maximum number of times without success. As a result, the association fails (because of communication loss) and is terminated. The ULP requests that the association be attempted again, and this attempt succeeds. A shutdown is then received from the remote peer, and the local peer enters the shutdown acknowledge sent state, which is followed by the association being terminated. Again, another association attempt is made and succeeds.

```
Router# debug ip sctp signal
Router# debug ip sctp state

<new assoc attempt>
00:20:08: SCTP: Assoc 0: state CLOSED -> COOKIE_WAIT
00:20:15: SCTP: Assoc 0: state COOKIE_WAIT -> ESTABLISHED
00:20:15: SCTP: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC
00:21:03: SCTP: Assoc 0: Restart rcvd from peer
00:21:03: SCTP: Assoc 0: Sent ASSOC_RESTART signal
00:21:04: SCTP: Assoc 0: chunk 62EA7F40 retransmitted more than max times, failing assoc
00:21:04: SCTP: Assoc 0: Sent ASSOC_FAILED signal, reason: SCTP_COMM_LOST
00:21:04: SCTP: Assoc 0: Sent ASSOC_TERMINATE signal
00:21:04: SCTP: Assoc 0: state ESTABLISHED -> CLOSED
<new assoc attempt>
00:21:04: SCTP: Assoc 0: state CLOSED -> COOKIE_WAIT
00:21:04: SCTP: Assoc 0: state COOKIE_WAIT -> COOKIE_ECHOED
00:21:04: SCTP: Assoc 0: state COOKIE_ECHOED -> ESTABLISHED
00:21:04: SCTP: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC
00:21:04: SCTP: Assoc 0: Sent TERMINATE_PENDING signal
00:21:04: SCTP: Assoc 0: state ESTABLISHED -> SHUTDOWN_ACKSENT
00:21:04: SCTP: Assoc 0: Sent ASSOC_TERMINATE signal
00:21:04: SCTP: Assoc 0: state SHUTDOWN_ACKSENT -> CLOSED
<new assoc attempt>
00:21:04: SCTP: Assoc 0: state CLOSED -> COOKIE_WAIT
00:21:04: SCTP: Assoc 0: state COOKIE_WAIT -> COOKIE_ECHOED
00:21:04: SCTP: Assoc 0: state COOKIE_ECHOED -> ESTABLISHED
00:21:04: SCTP: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC
```

In the following example, the associations themselves are stable, but a particular destination address fails. Because both currently established associations are using the same destination addresses (with different ports), both of the associations indicate the destination address failure. When the destination address again becomes active, the ULPs are informed.

```
Router#
00:26:27: SCTP: Assoc 1: Sent DESTADDR_FAILED signal for destaddr 10.6.0.4
00:26:28: SCTP: Assoc 0: Sent DESTADDR_FAILED signal for destaddr 10.6.0.4
Router#
00:30:41: SCTP: Assoc 1: Sent DESTADDR_ACTIVE signal for destaddr 10.6.0.4
00:30:41: SCTP: Assoc 0: Sent DESTADDR_ACTIVE signal for destaddr 10.6.0.4
```

## debug ip sctp sndchunks

The **debug ip sctp sndchunks** command shows the following types of information about all chunks that are being sent to remote SCTP peers:

- Application send requests from the local SCTP peer
- Chunks being bundled and sent to the remote peer
- Processing of the SACKs from the remote peer, indicating which chunks were successfully received
- Chunks that are marked for retransmission



### Caution

The **debug ip sctp sndchunks** command generates large amounts of data if there is any significant amount of traffic flowing. It should be used with extreme caution in live networks.

```
Router# debug ip sctp sndchunks

SCTP: Assoc 0: ApplSend, chunk: 0/10412/100/A23134F8 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10443/100/A23134F9 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10448/100/A231355C to 10.5.0.4
SCTP: Assoc 0: Set oldest chunk for dest 10.5.0.4 to TSN A23134F8
SCTP: Assoc 0: Bundling data, added 0/10412/100/A23134F8, outstanding 100
SCTP: Assoc 0: Bundling data, added 5/10443/100/A23134F9, outstanding 200
SCTP: Assoc 0: Bundling data, added 4/10545/100/A23134FA, outstanding 300
SCTP: Assoc 0: Bundling data, added 10/10371/100/A23134FB, outstanding 400
SCTP: Assoc 0: Bundling data, added 11/10382/100/A23134FC, outstanding 500
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A231350F, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313510
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A2313527, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313528
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A231353F, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313540
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A2313557, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313558
SCTP: Assoc 0: ApplSend, chunk: 10/10385/100/A23135BE to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 8/10230/100/A23135BF to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10459/100/A23135C0 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 4/10558/100/A23135C1 to 10.5.0.4
SCTP: Assoc 0: Set oldest chunk for dest 10.5.0.4 to TSN A231355D
SCTP: Assoc 0: Bundling data, added 5/10449/100/A231355D, outstanding 100
SCTP: Assoc 0: Bundling data, added 3/10490/100/A231355E, outstanding 200
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A23135A4, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A23135A5
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A23135BC, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A23135BD
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A23135C1, numFrgs=0
SCTP: Assoc 0: ApplSend, chunk: 5/10460/100/A23135C2 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10461/100/A23135C3 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 11/10403/100/A2313626 to 10.5.0.4
SCTP: Assoc 0: Set oldest chunk for dest 10.5.0.4 to TSN A23135C2
SCTP: Assoc 0: Bundling data, added 5/10460/100/A23135C2, outstanding 100
SCTP: Assoc 0: Bundling data, added 5/10461/100/A23135C3, outstanding 200
SCTP: Assoc 0: Bundling data, added 5/10462/100/A23135C4, outstanding 300
SCTP: Assoc 0: Bundling data, added 4/10559/100/A23135C5, outstanding 400
SCTP: Assoc 0: Bundling data, added 4/10560/100/A23135C6, outstanding 500
SCTP: Assoc 0: Bundled 12 chunk(s) in next dgram to 10.5.0.4
SCTP: Assoc 0: Bundling data, added 1/10418/100/A2313622, outstanding 9700
SCTP: Assoc 0: Bundling data, added 3/10502/100/A2313623, outstanding 9800
SCTP: Assoc 0: Bundling data, added 7/10482/100/A2313624, outstanding 9900
SCTP: Assoc 0: Bundling data, added 3/10503/100/A2313625, outstanding 10000
SCTP: Assoc 0: Bundling data, added 11/10403/100/A2313626, outstanding 10100
```

```

SCTP: Assoc 0: Bundled 5 chunk(s) in next dgram to 10.5.0.4
SCTP: Assoc 0: Mark chunk A23135C2 for retrans
SCTP: Assoc 0: Mark chunk A23135C3 for retrans
SCTP: Assoc 0: Mark chunk A23135C4 for retrans
SCTP: Assoc 0: Mark chunk A23135C5 for retrans
SCTP: Assoc 0: Mark chunk A23135C6 for retrans
SCTP: Assoc 0: Mark chunk A23135C7 for retrans
SCTP: Assoc 0: Mark chunk A23135C8 for retrans
SCTP: Assoc 0: Mark chunk A23135C9 for retrans
SCTP: Assoc 0: Mark chunk A23135CA for retrans
SCTP: Assoc 0: Bundled 6 chunk(s) in next dgram to 10.6.0.4
SCTP: Assoc 0: Mark chunk A23135C2 for retrans
SCTP: Assoc 0: Mark chunk A23135C3 for retrans
SCTP: Assoc 0: Mark chunk A23135C4 for retrans

```

## debug ip sctp state

For a description of this command, see the “[debug ip sctp signal](#)” section on page 19.

## debug ip sctp timer

The **debug ip sctp timer** command displays information about all started, stopped, and triggering SCTP timers. After they have been started, many SCTP timers are not restarted until they expire or are stopped. For these timers, the first call succeeds in starting the timer, and subsequent calls do nothing until the timer either expires or is stopped. For example, the retransmission timer is started when the first chunk is sent, but then is not started again for subsequent chunks when there is outstanding data.



### Caution

The **debug ip sctp timer** command generates a significant amount of output. It should be used with extreme caution in a live network.

The following example shows output from the **debug ip sctp timer** command:

```

Router# debug ip sctp timer

SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Timer BUNDLE triggered
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Stopping RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Stopping RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer

```

```
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Stopping CUMSACK timer
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
```

## debug ip sctp warnings

The **debug ip sctp warnings** command displays information on any unusual situation that is encountered. These situations may or may not indicate problems, depending on the particulars of the situation. Below are some examples of events or conditions that are flagged as warnings.

```
Router# debug ip sctp warnings

SCTP: Assoc 0: No cookie in InitAck, discarding
SCTP: Assoc 0: Incoming INIT_ACK: inbound streams reqd 15, allowed 13
SCTP: Assoc 0: Incoming INIT_ACK request: outbound streams req'd 13, allowed 1
SCTP: Assoc 0: Remote verification tag in init ack is zero, discarding
SCTP: Remote verification tag in init is zero, discarding
SCTP: Assoc 0: Rwnd less than min allowed (1500) in incoming INITACK, rcvd 0
SCTP: Assoc 0: Rwnd less than min allowed (1500) in incoming INITACK, rcvd 1499
SCTP: Rwnd in INIT too small (0), discarding
SCTP: Rwnd in INIT too small (1499), discarding
SCTP: Unknown INIT param 16537 (0x4099), length 8
SCTP: Assoc 0: Unknown INITACK param 153 (0x99), length 8
SCTP: Assoc 0: No cookie in InitAck, discarding
SCTP: Assoc 0: No cookie in InitAck, discarding
SCTP: Processing INIT, invalid param len 0, discarding...
SCTP: Assoc 0: Processing INITACK, invalid param len 0, discarding...
```

## Clear Command

The **clear ip sctp statistics** command clears all the SCTP statistics, both those compiled for individual associations and those compiled overall.

## Configuration Examples

None

# Command Reference

This section documents new and modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.2 command reference publications.

- [clear ip sctp statistics](#)
- [debug ip sctp api](#)
- [debug ip sctp congestion](#)
- [debug ip sctp init](#)
- [debug ip sctp multihome](#)
- [debug ip sctp performance](#)
- [debug ip sctp rcvchunks](#)
- [debug ip sctp rto](#)
- [debug ip sctp segments](#)
- [debug ip sctp segmentv](#)
- [debug ip sctp signal](#)
- [debug ip sctp sndchunks](#)
- [debug ip sctp state](#)
- [debug ip sctp timer](#)
- [debug ip sctp warnings](#)
- [show ip sctp association list](#)
- [show ip sctp association parameters](#)
- [show ip sctp association statistics](#)
- [show ip sctp errors](#)
- [show ip sctp instances](#)
- [show ip sctp statistics](#)

# clear ip sctp statistics

To clear statistics counts for Stream Control Transmission Protocol (SCTP) activity, use the **clear ip sctp statistics** command in privileged EXEC mode.

**clear ip sctp statistics**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(2)T	This command was introduced.

**Usage Guidelines** This command clears both individual and overall statistics.

**Examples** The following command shows how to empty the buffer that holds SCTP statistics. No output is generated from this command.

```
Router# clear ip sctp statistics
```

Related Commands	Command	Description
	<a href="#">debug ip sctp api</a>	Reports SCTP diagnostic information and messages.
	<a href="#">debug ip sctp congestion</a>	Shows a list of all current SCTP associations.
	<a href="#">show ip sctp association parameters</a>	Shows the parameters configured for the association defined by the association identifier.
	<a href="#">show ip sctp association statistics</a>	Shows the current statistics for the association defined by the association identifier.
	<a href="#">show ip sctp errors</a>	Shows error counts logged by SCTP.
	<a href="#">show ip sctp instances</a>	Shows all currently defined SCTP instances.
	<a href="#">show ip sctp statistics</a>	Shows overall statistics counts for SCTP.
	<a href="#">show iua as</a>	Shows information about the current condition of an application server.
	<a href="#">show iua asp</a>	Shows information about the current condition of an application server process.

# debug ip sctp api

To provide diagnostic information about Stream Control Transmission Protocol (SCTP) application programming interfaces (APIs), use the **debug ip sctp api** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp api**

**no debug ip sctp api**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** In a live system, the debugs for performance, state, signal, and warnings are the most useful. These show any association or destination address failures and can be used to monitor the stability of any established associations.



**Caution**

The **debug ip sctp api** command should not be used in a live system that has any significant amount of traffic running because it can generate a lot of traffic, which can cause associations to fail.

**Examples** The following example shows SCTP calls to the API that are being executed and the parameters associated with these calls:

```
Router# debug ip sctp api

*Mar 1 00:31:14.211: SCTP: sctp_send: Assoc ID: 1
*Mar 1 00:31:14.211: SCTP:          stream num: 10
*Mar 1 00:31:14.211: SCTP:          bptr: 62EE332C, dptr: 4F7B598
*Mar 1 00:31:14.211: SCTP:          datalen: 100
*Mar 1 00:31:14.211: SCTP:          context: 1
*Mar 1 00:31:14.211: SCTP:          lifetime: 0
*Mar 1 00:31:14.211: SCTP:          unorder flag: FALSE
*Mar 1 00:31:14.211: SCTP:          bundle flag: TRUE
*Mar 1 00:31:14.211: SCTP: sctp_send successful return
*Mar 1 00:31:14.211: SCTP: sctp_receive: Assoc ID: 1
*Mar 1 00:31:14.215: SCTP:          max data len: 100
*Mar 1 00:31:14.215: SCTP: sctp_receive successful return
*Mar 1 00:31:14.215: SCTP: Process Send Request
*Mar 1 00:31:14.951: SCTP: sctp_receive: Assoc ID: 0
*Mar 1 00:31:14.951: SCTP:          max data len: 100
*Mar 1 00:31:14.951: SCTP: sctp_receive successful return
.
.
.
```

Table 1 describes the significant fields shown in the display.

**Table 1** *debug ip sctp api Field Descriptions*

Field	Description
Assoc ID	Association identifier.
stream num	SCTP stream number.
bptr, dptr	Address of the buffer that contains the data, and address of the start of the data.
datalen	Length of the data that the application is sending (the datagram).
context	A value that is meaningful to the application. Returned with the datagram if the datagram ever needs to be retrieved.
lifetime	Not used.
unordered flag	Specifies that the datagram should be sent as unordered data.
bundle flag	Indicates whether the application wants the datagram to be delayed slightly, trying to bundle it with other data being sent.
max data len	Maximum length of data that can be received—the size of the receive buffer.

#### Related Commands

Command	Description
<a href="#">clear ip sctp statistics</a>	Empties the buffer that holds SCTP statistics.
<a href="#">debug ip sctp congestion</a>	Shows a list of all current SCTP associations.
<a href="#">show ip sctp association parameters</a>	Shows the parameters configured for the association defined by the association identifier.
<a href="#">show ip sctp association statistics</a>	Shows the current statistics for the association defined by the association identifier.
<a href="#">show ip sctp errors</a>	Shows error counts logged by SCTP.
<a href="#">show ip sctp instances</a>	Shows all currently defined SCTP instances.
<a href="#">show ip sctp statistics</a>	Shows overall statistics counts for SCTP.
<a href="#">show iua as</a>	Shows information about the current condition of an application server.
<a href="#">show iua asp</a>	Shows information about the current condition of an application server process.

# debug ip sctp congestion

To provide diagnostic information about Stream Control Transmission Protocol (SCTP) congestion parameters, use the **debug ip sctp congestion** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp congestion**

**no debug ip sctp congestion**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** In a live system, the debugs for performance, state, signal, and warnings are the most useful. These show any association or destination address failures and can be used to monitor the stability of any established associations.

Debug commands other than those for performance, state, signal, and warnings can generate a great deal of output and therefore can cause associations to fail. These commands should be used only in test environments or when there are very low amounts of traffic.

**Examples** The following example shows parameters used to calculate SCTP congestion:

```
Router# debug ip sctp congestion

SCTP: Assoc 0: Slow start 10.6.0.4, cwnd 3000
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7800
SCTP: Assoc 0: Free chunks, local rwnd 9000
SCTP: Assoc 0: Data chunks rcvd, local rwnd 8200
SCTP: Assoc 0: Add Sack, local a_rwnd 8200
SCTP: Assoc 0: Free chunks, local rwnd 9000
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7800
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7000
SCTP: Assoc 0: Add Sack, local a_rwnd 7000
SCTP: Assoc 0: Free chunks, local rwnd 9000
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 14000, cwnd 19500, outstand 0
SCTP: Assoc 0: Bundled 12 chunks, remote rwnd 12800, outstand 1200
SCTP: Assoc 0: Bundling data, next chunk dataLen (100) > remaining mtu size
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 12800, cwnd 19500, outstand 1200
SCTP: Assoc 0: Bundled 12 chunks, remote rwnd 11600, outstand 2400
SCTP: Assoc 0: Bundling data, next chunk dataLen (100) > remaining mtu size
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 11600, cwnd 19500, outstand 2400
SCTP: Assoc 0: Bundled 12 chunks, remote rwnd 10400, outstand 3600
SCTP: Assoc 0: Bundling data, next chunk dataLen (100) > remaining mtu size
SCTP: Assoc 0: Bundle for 10.5.0.4, rem rwnd 10400, cwnd 19500, outstand 3600
SCTP: Assoc 0: Bundled 4 chunks, remote rwnd 10000, outstand 4000
```

```

SCTP: Assoc 0: No additional chunks waiting.
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7800
SCTP: Assoc 0: Data chunks rcvd, local rwnd 7000
SCTP: Assoc 0: Add Sack, local a_rwnd 7000
SCTP: Assoc 0: Chunk A22F3B45 ack'd, dest 10.5.0.4, outstanding 3900
SCTP: Assoc 0: Chunk A22F3B46 ack'd, dest 10.5.0.4, outstanding 3800
SCTP: Assoc 0: Chunk A22F3B47 ack'd, dest 10.5.0.4, outstanding 3700
SCTP: Assoc 0: Chunk A22F3B48 ack'd, dest 10.5.0.4, outstanding 3600
SCTP: Assoc 0: Chunk A22F3B49 ack'd, dest 10.5.0.4, outstanding 3500
SCTP: Assoc 0: Chunk A22F3B4A ack'd, dest 10.5.0.4, outstanding 3400
SCTP: Assoc 0: Chunk A22F3B4B ack'd, dest 10.5.0.4, outstanding 3300
SCTP: Assoc 0: Chunk A22F3B4C ack'd, dest 10.5.0.4, outstanding 3200
SCTP: Assoc 0: Chunk A22F3B4D ack'd, dest 10.5.0.4, outstanding 3100
SCTP: Assoc 0: Chunk A22F3B4E ack'd, dest 10.5.0.4, outstanding 3000
SCTP: Assoc 0: Chunk A22F3B4F ack'd, dest 10.5.0.4, outstanding 2900
SCTP: Assoc 0: Chunk A22F3B50 ack'd, dest 10.5.0.4, outstanding 2800
SCTP: Assoc 0: Chunk A22F3B51 ack'd, dest 10.5.0.4, outstanding 2700
SCTP: Assoc 0: Chunk A22F3B52 ack'd, dest 10.5.0.4, outstanding 2600
SCTP: Assoc 0: Chunk A22F3B53 ack'd, dest 10.5.0.4, outstanding 2500
SCTP: Assoc 0: Chunk A22F3B54 ack'd, dest 10.5.0.4, outstanding 2400
SCTP: Assoc 0: Chunk A22F3B55 ack'd, dest 10.5.0.4, outstanding 2300
SCTP: Assoc 0: Chunk A22F3B56 ack'd, dest 10.5.0.4, outstanding 2200

```

Table 2 describes the significant fields shown in the display.

**Table 2** *debug ip sctp congestion Field Descriptions*

Field	Description
cwnd	Congestion window values for destination address.
rwnd, a_rwnd	Receiver window values as defined in RFC 2960.
outstanding	Number of bytes outstanding.

#### Related Commands

Command	Description
<a href="#">clear ip sctp statistics</a>	Empties the buffer that holds SCTP statistics.
<a href="#">debug ip sctp congestion</a>	Shows a list of all current SCTP associations.
<a href="#">show ip sctp association parameters</a>	Shows the parameters configured for the association defined by the association identifier.
<a href="#">show ip sctp association statistics</a>	Shows the current statistics for the association defined by the association identifier.
<a href="#">show ip sctp errors</a>	Shows error counts logged by SCTP.
<a href="#">show ip sctp instances</a>	Shows all currently defined SCTP instances.
<a href="#">show ip sctp statistics</a>	Shows overall statistics counts for SCTP.
<a href="#">show iua as</a>	Shows information about the current condition of an application server.
<a href="#">show iua asp</a>	Shows information about the current condition of an application server process.

# debug ip sctp init

To show datagrams and other information related to the initializing of new Stream Control Transmission Protocol (SCTP) associations, use the **debug ip sctp init** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp init**

**no debug ip sctp init**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** All initialization chunks are shown, including the INIT, INIT\_ACK, COOKIE\_ECHO, and COOKIE\_ACK chunks. This debug command can be used to see the chunks associated with any initialization sequence, but does not display data chunks sent once the association is established. Therefore, it is safe to use in a live system that has traffic flowing when you have trouble with associations failing and being reestablished.

**Examples** The following example shows initialization chunks for SCTP associations:

```
Router# debug ip sctp init

*Mar 1 00:53:07.279: SCTP Test: Attempting to open assoc to remote port 8787...assoc ID is 0
*Mar 1 00:53:07.279: SCTP: Process Assoc Request
*Mar 1 00:53:07.279: SCTP: Assoc 0: dest addr list:
*Mar 1 00:53:07.279: SCTP:                addr 10.5.0.4
*Mar 1 00:53:07.279: SCTP:                addr 10.6.0.4
*Mar 1 00:53:07.279:
...
*Mar 1 00:53:13.279: SCTP: Assoc 0: Send Init
*Mar 1 00:53:13.279: SCTP:                INIT_CHUNK, len 42
*Mar 1 00:53:13.279: SCTP:                Initiate Tag: B4A10C4D, Initial TSN: B4A10C4D, rwnd 9000
*Mar 1 00:53:13.279: SCTP:                Streams Inbound: 13, Outbound: 13
*Mar 1 00:53:13.279: SCTP:                IP Addr: 10.1.0.2
*Mar 1 00:53:13.279: SCTP:                IP Addr: 10.2.0.2
*Mar 1 00:53:13.279: SCTP:                Supported addr types: 5
*Mar 1 00:53:13.307: SCTP: Process Init
*Mar 1 00:53:13.307: SCTP:                INIT_CHUNK, len 42
*Mar 1 00:53:13.307: SCTP:                Initiate Tag: 3C2D8327, Initial TSN: 3C2D8327, rwnd 18000
*Mar 1 00:53:13.307: SCTP:                Streams Inbound: 13, Outbound: 13
*Mar 1 00:53:13.307: SCTP:                IP Addr: 10.5.0.4
*Mar 1 00:53:13.307: SCTP:                IP Addr: 10.6.0.4
*Mar 1 00:53:13.307: SCTP:                Supported addr types: 5
*Mar 1 00:53:13.307: SCTP: Assoc 0: Send InitAck
```

## debug ip sctp init

```

*Mar 1 00:53:13.307: SCTP:          INIT_ACK_CHUNK, len 124
*Mar 1 00:53:13.307: SCTP:          Initiate Tag: B4A10C4D, Initial TSN: B4A10C4D, rwnd 9000
*Mar 1 00:53:13.307: SCTP:          Streams Inbound: 13, Outbound: 13
*Mar 1 00:53:13.307: SCTP:          Responder cookie len 88
*Mar 1 00:53:13.307: SCTP:          IP Addr: 10.1.0.2
*Mar 1 00:53:13.307: SCTP:          IP Addr: 10.2.0.2
*Mar 1 00:53:13.311: SCTP: Assoc 0: Process Cookie
*Mar 1 00:53:13.311: SCTP:          COOKIE_ECHO_CHUNK, len 88
*Mar 1 00:53:13.311: SCTP: Assoc 0: dest addr list:
*Mar 1 00:53:13.311: SCTP:                      addr 10.5.0.4
*Mar 1 00:53:13.311: SCTP:                      addr 10.6.0.4
*Mar 1 00:53:13.311:
*Mar 1 00:53:13.311: SCTP: Instance 0 dest addr list:
*Mar 1 00:53:13.311: SCTP:                      addr 10.5.0.4
*Mar 1 00:53:13.311: SCTP:                      addr 10.6.0.4
*Mar 1 00:53:13.311:
*Mar 1 00:53:13.311: SCTP: Assoc 0: Send CookieAck
*Mar 1 00:53:13.311: SCTP:          COOKIE_ACK_CHUNK

```

Table 3 describes the significant fields shown in the display.

**Table 3** *debug ip sctp init Field Descriptions*

Field	Description
Initiate Tag	Initiation chunk identifier.
Initial TSN	Initial transmission sequence number.
rwnd	Receiver window values.

## Related Commands

Command	Description
<b>clear ip sctp statistics</b>	Empties the buffer that holds SCTP statistics.
<b>debug ip sctp congestion</b>	Shows a list of all current SCTP associations.
<b>show ip sctp association parameters</b>	Shows the parameters configured for the association defined by the association identifier.
<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
<b>show ip sctp instances</b>	Shows all currently defined SCTP instances.
<b>show ip sctp statistics</b>	Shows overall statistics counts for SCTP.
<b>show iua as</b>	Shows information about the current condition of an application server.
<b>show iua asp</b>	Shows information about the current condition of an application server process.

# debug ip sctp multihome

To show the source and destination of datagrams in order to monitor the use of the multihome addresses for Stream Control Transmission Protocol (SCTP), use the **debug ip sctp multihome** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp multihome**

**no debug ip sctp multihome**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** More than one IP address parameter can be included in an initialization (INIT) chunk when the INIT sender is multihomed. Datagrams should be sent to the primary destination addresses unless the network is experiencing problems, in which case the datagrams should be sent to secondary addresses.



**Caution**

The **debug ip sctp multihome** command generates one debug line for each datagram sent or received. It should be used with extreme caution in a live network.

**Examples** The following example shows source and destination for multihomed addresses:

```
Router# debug ip sctp multihome

SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 476
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: Assoc 0: Send Data to dest 10.5.0.4
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 476
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 1404
SCTP: Rcvd s=10.5.0.4 8787, d=10.1.0.2 8787, len 476
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: Assoc 0: Send Data to dest 10.5.0.4
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 1404
```

## ■ debug ip sctp multihome

```

SCTP: Sent:  Assoc 0:  s=10.1.0.2  8787, d=10.5.0.4  8787, len 476
SCTP: Rcvd s=10.6.0.4  8787, d=10.2.0.2  8787, len 44
SCTP: Sent:  Assoc 0:  s=10.2.0.2  8787, d=10.6.0.4  8787, len 44
SCTP: Rcvd s=10.5.0.4  8787, d=10.1.0.2  8787, len 28
SCTP: Rcvd s=10.5.0.4  8787, d=10.1.0.2  8787, len 28
SCTP: Rcvd s=10.5.0.4  8787, d=10.1.0.2  8787, len 1404
SCTP: Rcvd s=10.5.0.4  8787, d=10.1.0.2  8787, len 1404
SCTP: Sent:  Assoc 0:  s=10.1.0.2  8787, d=10.5.0.4  8787, len 28
SCTP: Rcvd s=10.5.0.4  8787, d=10.1.0.2  8787, len 1404
SCTP: Rcvd s=10.5.0.4  8787, d=10.1.0.2  8787, len 476

```

Table 4 describes the significant fields shown in the display.

**Table 4** *debug ip sctp multihome Field Descriptions*

Field	Description
s	Source address and port.
d	Destination address and port.

## Related Commands

Command	Description
<a href="#">clear ip sctp statistics</a>	Empties the buffer that holds SCTP statistics.
<a href="#">debug ip sctp congestion</a>	Shows a list of all current SCTP associations.
<a href="#">show ip sctp association parameters</a>	Shows the parameters configured for the association defined by the association identifier.
<a href="#">show ip sctp association statistics</a>	Shows the current statistics for the association defined by the association identifier.
<a href="#">show ip sctp errors</a>	Shows error counts logged by SCTP.
<a href="#">show ip sctp instances</a>	Shows all currently defined SCTP instances.
<a href="#">show ip sctp statistics</a>	Shows overall statistics counts for SCTP.
<a href="#">show iua as</a>	Shows information about the current condition of an application server.
<a href="#">show iua asp</a>	Shows information about the current condition of an application server process.

# debug ip sctp performance

To display the average number of Stream Control Transmission Protocol (SCTP) chunks and datagrams being sent and received per second, use the **debug ip sctp performance** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp performance**

**no debug ip sctp performance**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** In a live system, the debugs for performance, state, signal, and warnings are the most useful. These show any association or destination address failures and can be used to monitor the stability of any established associations.

Once enabled, the **debug ip sctp performance** command displays the average number of chunks and datagrams being sent and received per second once every 10 seconds. Note that the averages are cumulative since the last time the statistics were cleared using the **clear ip sctp statistics** command and may not accurately reflect the number of datagrams and chunks currently being sent and received at that particular moment.

**Examples** The following example shows a low rate of traffic:

```
Router# debug ip sctp performance

SCTP Sent: SCTP Dgrams 5, Chunks 28, Data Chunks 29, ULP Dgrams 29
SCTP Rcvd: SCTP Dgrams 7, Chunks 28, Data Chunks 29, ULP Dgrams 29
Chunks Discarded: 0, Retransmitted 0

SCTP Sent: SCTP Dgrams 6, Chunks 29, Data Chunks 30, ULP Dgrams 30
SCTP Rcvd: SCTP Dgrams 7, Chunks 29, Data Chunks 30, ULP Dgrams 30
Chunks Discarded: 0, Retransmitted 0
```

[Table 5](#) describes the significant fields shown in the display.

**Table 5** *debug ip sctp performance Field Descriptions*

Field	Description
SCTP Dgrams	Datagram sent to or received from the network.
Chunks	Includes data chunks and control chunks sent or received.

**Table 5** *debug ip sctp performance Field Descriptions (continued)*

<b>Field</b>	<b>Description</b>
Data Chunks	Data chunks sent or received.
ULP Dgrams	Upper-layer protocol (ULP) datagrams, which are datagrams sent to or received from the ULP or application.

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>clear ip sctp statistics</b>	Empties the buffer that holds SCTP statistics.
<b>debug ip sctp congestion</b>	Shows a list of all current SCTP associations.
<b>show ip sctp association parameters</b>	Shows the parameters configured for the association defined by the association identifier.
<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
<b>show ip sctp instances</b>	Shows all currently defined SCTP instances.
<b>show ip sctp statistics</b>	Shows overall statistics counts for SCTP.
<b>show iua as</b>	Shows information about the current condition of an application server.
<b>show iua asp</b>	Shows information about the current condition of an application server process.

# debug ip sctp rcvchunks

To provide diagnostic information about chunks received with Stream Control Transmission Protocol (SCTP), use the **debug ip sctp rcvchunks** command in privileged EXEC mode. To disable diagnostic reporting, use the **no** form of this command.

**debug ip sctp rcvchunks**

**no debug ip sctp rcvchunks**

## Syntax Description

This command has no arguments or keywords.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.2(4)T	This command was introduced.

## Usage Guidelines

The **debug ip sctp rcvchunks** command shows the following information about received chunks:

- Whether the chunk is for a new datagram or is part of a datagram that is being reassembled.
- Whether the datagram is complete after receiving this chunk.
- If the datagram is complete, whether the datagram is in sequence within the specified stream and can be delivered to the upper-layer protocol (ULP).
- The selective acknowledgements (SACKs) that are returned to the remote SCTP peer.
- The cumulative transmission sequence number (Cum TSN) that was acknowledged and the number of fragments included.
- Whether the datagram is received by the ULP.



### Caution

The **debug ip sctp rcvchunks** command generates multiple debug lines for each chunk received. It should be used with extreme caution in a live network.

## Examples

In the following example, a segmented datagram is received in two chunks for stream 0 and sequence number 0. The length of the first chunk is 1452 bytes, and the second is 1 byte. The first chunk indicates that it is for a new datagram, but the second chunk indicates that it is part of an existing datagram that is already being reassembled. When the first chunk is processed, it is noted to be in sequence, but is not complete and so cannot be delivered yet. When the second chunk is received, the datagram is both in sequence and complete. The application receives the datagram, and a SACK is shown to acknowledge that both chunks were received with no missing chunks indicated (that is, with no fragments).

```
Router# debug ip sctp rcvchunks
```

```
SCTP: Assoc 0: New chunk (0/0/1452/2C33D822) for new dgram (0)
SCTP: Assoc 0: dgram (0) is in seq
```

```

SCTP: Assoc 0: Add Sack Chunk, CumTSN=2C33D822, numFrag=0
SCTP: Assoc 0: New chunk (0/0/1/2C33D823) for existing dgram (0)
SCTP: Assoc 0: dgram (0) is complete
SCTP: Assoc 0: ApplRecv chunk 0/0/1452/2C33D822
SCTP: Assoc 0: ApplRecv chunk 0/0/1/2C33D823
SCTP: Assoc 0: Add Sack Chunk, CumTSN=2C33D823, numFrag=0

```

Table 6 describes the significant fields shown in the display.

**Table 6** *debug ip sctp rcvchunks Field Descriptions*

Field	Description
0 / 0 / 1452 / 2C33D822	Stream number / datagram sequence number / chunk length, in bytes / chunk transmission sequence number.
Sack Chunk	Selective acknowledgement chunk.
CumTSN	Cumulative transmission sequence number that is being acknowledged.
numFrag	Number of fragments, or missing chunks.
ApplRecv	Application has received the chunk.

#### Related Commands

Command	Description
<a href="#">clear ip sctp statistics</a>	Empties the buffer that holds SCTP statistics.
<a href="#">debug ip sctp congestion</a>	Shows a list of all current SCTP associations.
<a href="#">show ip sctp association parameters</a>	Shows the parameters configured for the association defined by the association identifier.
<a href="#">show ip sctp association statistics</a>	Shows the current statistics for the association defined by the association identifier.
<a href="#">show ip sctp errors</a>	Shows error counts logged by SCTP.
<a href="#">show ip sctp instances</a>	Shows all currently defined SCTP instances.
<a href="#">show ip sctp statistics</a>	Shows overall statistics counts for SCTP.
<a href="#">show iua as</a>	Shows information about the current condition of an application server.
<a href="#">show iua asp</a>	Shows information about the current condition of an application server process.

# debug ip sctp rto

To show adjustments that are made to the retransmission timeout (RTO) value when using Stream Control Transmission Protocol (SCTP), use the **debug ip sctp rto** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp rto**

**no debug ip sctp rto**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** The **debug ip sctp rto** command shows adjustments that are made to the retransmission (retrans) timeout value because of either retransmission of data chunks or unacknowledged heartbeats.



**Caution**

The **debug ip sctp rto** command can generate a great deal of output. It should be used with extreme caution in a live network.

**Examples** In the following example, there is only one destination address available. Each time the chunk needs to be retransmitted, the RTO value is doubled.

```
Router# debug ip sctp rto

SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 2000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 4000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 8000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 16000 ms
SCTP: Assoc 0: destaddr 10.5.0.4, retrans timeout on chunk 942BAC55
SCTP: Assoc 0: destaddr 10.5.0.4, rto backoff 32000 ms
```

Related Commands	Command	Description
	<a href="#">clear ip sctp statistics</a>	Empties the buffer that holds SCTP statistics.
	<a href="#">debug ip sctp congestion</a>	Shows a list of all current SCTP associations.

Command	Description
<a href="#">show ip sctp association parameters</a>	Shows the parameters configured for the association defined by the association identifier.
<a href="#">show ip sctp association statistics</a>	Shows the current statistics for the association defined by the association identifier.
<a href="#">show ip sctp errors</a>	Shows error counts logged by SCTP.
<a href="#">show ip sctp instances</a>	Shows all currently defined SCTP instances.
<a href="#">show ip sctp statistics</a>	Shows overall statistics counts for SCTP.
<a href="#">show iua as</a>	Shows information about the current condition of an application server.
<a href="#">show iua asp</a>	Shows information about the current condition of an application server process.

# debug ip sctp segments

To show short diagnostics for every datagram that is sent or received with Stream Control Transmission Protocol (SCTP), use the **debug ip sctp segments** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp segments**

**no debug ip sctp segments**

## Syntax Description

This command has no arguments or keywords.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.2(4)T	This command was introduced.

## Usage Guidelines

The **debug ip sctp segments** command provides the short form of the output about datagrams. For the verbose form, use the **debug ip sctp segmentv** command.



### Caution

The **debug ip sctp segments** command generates several lines of output for each datagram sent or received. It should be used with extreme caution in a live network.

## Examples

The following output shows an example in which an association is established, a few heartbeats are sent, the remote endpoint fails, and the association is restarted.

```
Router# debug ip sctp segments

SCTP: Sent:  Assoc 0: s=10.1.0.2  8787, d=10.5.0.4  8787, len 56
SCTP:      INIT_CHUNK, Tag: 3C72A02A, TSN: 3C72A02A
SCTP: Recv:  Assoc 0: s=10.5.0.4  8787, d=10.1.0.2  8787, len 56
SCTP:      INIT_CHUNK, Tag: 13E5AD6C, TSN: 13E5AD6C
SCTP: Sent:  Assoc NULL: s=10.1.0.2  8787, d=10.5.0.4  8787, len 136
SCTP:      INIT_ACK_CHUNK, Tag: 3C72A02A, TSN: 3C72A02A
SCTP: Recv:  Assoc 0: s=10.5.0.4  8787, d=10.1.0.2  8787, len 100
SCTP:      COOKIE_ECHO_CHUNK, len 88
SCTP: Sent:  Assoc NULL: s=10.1.0.2  8787, d=10.5.0.4  8787, len 16
SCTP:      COOKIE_ACK_CHUNK
SCTP: Sent:  Assoc 0: s=10.1.0.2  8787, d=10.5.0.4  8787, len 52
SCTP:      HEARTBEAT_CHUNK
SCTP: Sent:  Assoc 0: s=10.1.0.2  8787, d=10.5.0.4  8787, len 52
SCTP:      HEARTBEAT_CHUNK
SCTP: Sent:  Assoc 0: s=10.1.0.2  8787, d=10.5.0.4  8787, len 52
SCTP:      HEARTBEAT_CHUNK
SCTP: Recv:  Assoc 0: s=10.5.0.4  8787, d=10.1.0.2  8787, len 56
SCTP:      INIT_CHUNK, Tag: 4F2D8235, TSN: 4F2D8235
SCTP: Sent:  Assoc NULL: s=10.1.0.2  8787, d=10.5.0.4  8787, len 136
SCTP:      INIT_ACK_CHUNK, Tag: 7DD7E424, TSN: 7DD7E424
```

## debug ip sctp segments

```

SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 100
SCTP: COOKIE_ECHO_CHUNK, len 88
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 16
SCTP: COOKIE_ACK_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 144
SCTP: SACK_CHUNK, TSN ack: 7DD7E423, rwnd 18000, num frags 0
SCTP: DATA_CHUNK, 4/0/100/4F2D8235
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 4F2D8235, rwnd 8900, num frags 0
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 128
SCTP: DATA_CHUNK, 4/0/100/7DD7E424
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 7DD7E424, rwnd 17900, num frags 0
SCTP: Recv: Assoc 0: s=10.6.0.4 8787, d=10.2.0.2 8787, len 44
SCTP: HEARTBEAT_CHUNK
SCTP: Sent: Assoc 0: s=10.2.0.2 8787, d=10.6.0.4 8787, len 44
SCTP: HEARTBEAT_ACK_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 128
SCTP: DATA_CHUNK, 7/0/100/4F2D8236
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 144
SCTP: SACK_CHUNK, TSN ack: 4F2D8236, rwnd 9000, num frags 0
SCTP: DATA_CHUNK, 7/0/100/7DD7E425
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 7DD7E424, rwnd 18000, num frags 0
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 28
SCTP: SACK_CHUNK, TSN ack: 7DD7E425, rwnd 17900, num frags 0
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 128
SCTP: DATA_CHUNK, 4/1/100/4F2D8237

```

Table 7 describes the significant fields shown in the display.

**Table 7** debug ip sctp segments Field Descriptions

Field	Description
s	Source address and port.
d	Destination address and port.
len	Length of chunk, in bytes.
Tag	The identifier for an initialization chunk.
TSN	Transmission sequence number.
rwnd	Receiver window value.
num frags	Number of fragments received.
7 / 0 / 100 / 4F2D8236	(Data chunks) Stream number / datagram sequence number / chunk length, in bytes / chunk transmission sequence number.

## Related Commands

Command	Description
<b>clear ip sctp statistics</b>	Empties the buffer that holds SCTP statistics.
<b>debug ip sctp congestion</b>	Shows a list of all current SCTP associations.
<b>debug ip sctp segmentv</b>	Shows every datagram that is sent or received and the chunks that are contained in each. This is the verbose form of the output, and it shows detailed information for each chunk type.

Command	Description
<b>show ip sctp association parameters</b>	Shows the parameters configured for the association defined by the association identifier.
<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
<b>show ip sctp instances</b>	Shows all currently defined SCTP instances.
<b>show ip sctp statistics</b>	Shows overall statistics counts for SCTP.
<b>show iua as</b>	Shows information about the current condition of an application server.
<b>show iua asp</b>	Shows information about the current condition of an application server process.

# debug ip sctp segmentv

To show verbose diagnostics for every datagram that is sent or received with Stream Control Transmission Protocol (SCTP), use the **debug ip sctp segmentv** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp segmentv**

**no debug ip sctp segmentv**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** The **debug ip sctp segmentv** command provides the verbose form of the output for datagrams. For the simple form, use the **debug ip sctp segments** command.



**Caution**

The **debug ip sctp segmentv** command generates multiple lines of output for each datagram sent and received. It should be used with extreme caution in a live network.

**Examples** The following output shows an example in which an association is established, a few heartbeats are sent, the remote endpoint fails, and the association is restarted.

```
Router# debug ip sctp segmentv

SCTP: Sent:  Assoc 0: s=10.1.0.2  8787, d=10.5.0.4  8787, len 56, ver tag 0
SCTP:      INIT_CHUNK, len 42
SCTP:      Initiate Tag: B131ED6A, Initial TSN: B131ED6A, rwnd 9000
SCTP:      Streams Inbound: 13, Outbound: 13
SCTP:      IP Addr: 10.1.0.2
SCTP:      IP Addr: 10.2.0.2
SCTP:      Supported addr types: 5
SCTP: Recv:  Assoc 0: s=10.5.0.4  8787, d=10.1.0.2  8787, len 56, ver tag 0
SCTP:      INIT_CHUNK, len 42
SCTP:      Initiate Tag: 5516B2F3, Initial TSN: 5516B2F3, rwnd 18000
SCTP:      Streams Inbound: 13, Outbound: 13
SCTP:      IP Addr: 10.5.0.4
SCTP:      IP Addr: 10.6.0.4
SCTP:      Supported addr types: 5
SCTP: Sent:  Assoc NULL: s=10.1.0.2  8787, d=10.5.0.4  8787, len 136, ver tag 5516B2F3
SCTP:      INIT_ACK_CHUNK, len 124
SCTP:      Initiate Tag: B131ED6A, Initial TSN: B131ED6A, rwnd 9000
SCTP:      Streams Inbound: 13, Outbound: 13
SCTP:      Responder cookie len 88
SCTP:      IP Addr: 10.1.0.2
```

```

SCTP:      IP Addr: 10.2.0.2
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 100, ver tag B131ED6A
SCTP:      COOKIE_ECHO_CHUNK, len 88
SCTP: Sent: Assoc NULL: s=10.1.0.2 8787, d=10.5.0.4 8787, len 16, ver tag 5516B2F3
SCTP:      COOKIE_ACK_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 144, ver tag B131ED6A
SCTP:      SACK_CHUNK, len 16
SCTP:      TSN ack: (0xB131ED69)
SCTP:      Rcv win credit: 18000
SCTP:      Num frags: 0
SCTP:      DATA_CHUNK, flags 3, chunkLen 116
SCTP:      DATA_CHUNK, 0/0/100/5516B2F3
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 28, ver tag 5516B2F3
SCTP:      SACK_CHUNK, len 16
SCTP:      TSN ack: (0x5516B2F3)
SCTP:      Rcv win credit: 8900
SCTP:      Num frags: 0
SCTP: Sent: Assoc 0: s=10.1.0.2 8787, d=10.5.0.4 8787, len 128, ver tag 5516B2F3
SCTP:      DATA_CHUNK, flags 3, chunkLen 116
SCTP:      DATA_CHUNK, 0/0/100/B131ED6A
SCTP: Recv: Assoc 0: s=10.6.0.4 8787, d=10.2.0.2 8787, len 44, ver tag B131ED6A
SCTP:      HEARTBEAT_CHUNK
SCTP: Sent: Assoc 0: s=10.2.0.2 8787, d=10.6.0.4 8787, len 44, ver tag 5516B2F3
SCTP:      HEARTBEAT_ACK_CHUNK
SCTP: Recv: Assoc 0: s=10.5.0.4 8787, d=10.1.0.2 8787, len 28, ver tag B131ED6A
SCTP:      SACK_CHUNK, len 16

```

Table 8 describes the significant fields shown in the display.

**Table 8** *debug ip sctp segmentv Field Descriptions*

Field	Description
s	Source address and port.
d	Destination address and port.
len	Length of chunk, in bytes.
ver tag	Verification identifier.
Tag	The identifier for an initialization chunk.
TSN	Transmission sequence number.
rwnd	Receive window value.
num frags	Number of fragments received.
Rcv win credit	Receive window value. Same as rwnd.
7 / 0 / 100 / 4F2D8236	(Data chunks) Stream number / datagram sequence number / chunk length, in bytes / chunk transmission sequence number.

#### Related Commands

Command	Description
<a href="#">clear ip sctp statistics</a>	Empties the buffer that holds SCTP statistics.
<a href="#">debug ip sctp congestion</a>	Shows a list of all current SCTP associations.
<a href="#">debug ip sctp segments</a>	Shows short diagnostics for every datagram that is sent or received with SCTP.

<b>Command</b>	<b>Description</b>
<b>show ip sctp association parameters</b>	Shows the parameters configured for the association defined by the association identifier.
<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
<b>show ip sctp instances</b>	Shows all currently defined SCTP instances.
<b>show ip sctp statistics</b>	Shows overall statistics counts for SCTP.
<b>show iua as</b>	Shows information about the current condition of an application server.
<b>show iua asp</b>	Shows information about the current condition of an application server process.

# debug ip sctp signal

To show signals that are sent from Stream Control Transmission Protocol (SCTP) to the application or upper-layer protocol (ULP), use the **debug ip sctp signal** command in privileged EXEC mode. To disable diagnostic reporting, use the **no** form of this command.

**debug ip sctp signal**

**no debug ip sctp signal**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** The **debug ip sctp signal** command can be used to see if the current associations are stable or not. Because it generates output only on state transitions, it is safe to use in a live environment. It still should be used with caution, however, depending on the number of associations being handled by the system and the stability of the network.

The **debug ip sctp state** command is often used at the same time as the **debug ip sctp signal** command. Using the two commands together gives good insight into the stability of associations.

**Examples** In the following example, a new association is requested and established. The peer then restarts the association and notes that the association failed and is being reestablished. The local peer then indicates that the association has failed because it has tried to retransmit the specified chunk more than the maximum number of times without success. As a result, the association fails (because of communication loss) and is terminated. The upper-layer protocol (ULP) requests that the association be attempted again, and this attempt succeeds. A shutdown is then received from the remote peer, and the local peer enters the shutdown acknowledge sent state, which is followed by the association being terminated. Again, another association attempt is made and succeeds.

```
Router# debug ip sctp signal
Router# debug ip sctp state

<new assoc attempt>
00:20:08: SCTP: Assoc 0: state CLOSED -> COOKIE_WAIT
00:20:15: SCTP: Assoc 0: state COOKIE_WAIT -> ESTABLISHED
00:20:15: SCTP: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC
00:21:03: SCTP: Assoc 0: Restart rcvd from peer
00:21:03: SCTP: Assoc 0: Sent ASSOC_RESTART signal
00:21:04: SCTP: Assoc 0: chunk 62EA7F40 retransmitted more than max times, failing assoc
00:21:04: SCTP: Assoc 0: Sent ASSOC_FAILED signal, reason: SCTP_COMM_LOST
00:21:04: SCTP: Assoc 0: Sent ASSOC_TERMINATE signal
00:21:04: SCTP: Assoc 0: state ESTABLISHED -> CLOSED
<new assoc attempt>
```

## debug ip sctp signal

```

00:21:04: SCTP: Assoc 0: state CLOSED -> COOKIE_WAIT
00:21:04: SCTP: Assoc 0: state COOKIE_WAIT -> COOKIE_ECHOED
00:21:04: SCTP: Assoc 0: state COOKIE_ECHOED -> ESTABLISHED
00:21:04: SCTP: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC
00:21:04: SCTP: Assoc 0: Sent TERMINATE_PENDING signal
00:21:04: SCTP: Assoc 0: state ESTABLISHED -> SHUTDOWN_ACKSENT
00:21:04: SCTP: Assoc 0: Sent ASSOC_TERMINATE signal
00:21:04: SCTP: Assoc 0: state SHUTDOWN_ACKSENT -> CLOSED
<new assoc attempt>
00:21:04: SCTP: Assoc 0: state CLOSED -> COOKIE_WAIT
00:21:04: SCTP: Assoc 0: state COOKIE_WAIT -> COOKIE_ECHOED
00:21:04: SCTP: Assoc 0: state COOKIE_ECHOED -> ESTABLISHED
00:21:04: SCTP: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC

```

## Related Commands

Command	Description
<a href="#">clear ip sctp statistics</a>	Empties the buffer that holds SCTP statistics.
<a href="#">debug ip sctp congestion</a>	Shows a list of all current SCTP associations.
<a href="#">debug ip sctp state</a>	Shows SCTP state transitions.
<a href="#">show ip sctp association parameters</a>	Shows the parameters configured for the association defined by the association identifier.
<a href="#">show ip sctp association statistics</a>	Shows the current statistics for the association defined by the association identifier.
<a href="#">show ip sctp errors</a>	Shows error counts logged by SCTP.
<a href="#">show ip sctp instances</a>	Shows all currently defined SCTP instances.
<a href="#">show ip sctp statistics</a>	Shows overall statistics counts for SCTP.
<a href="#">show iua as</a>	Shows information about the current condition of an application server.
<a href="#">show iua asp</a>	Shows information about the current condition of an application server process.

# debug ip sctp sndchunks

To show information about chunks that are being sent to remote Stream Control Transmission Protocol (SCTP) peers, use the **debug ip sctp sndchunks** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp sndchunks**

**no debug ip sctp sndchunks**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** The **debug ip sctp sndchunks** command provides the following information:

- Application send requests from the local SCTP peer
- Chunks being bundled and sent to the remote peer
- Processing of the SACKs from the remote peer, indicating which chunks were successfully received
- Chunks that are marked for retransmission



**Caution**

The **debug ip sctp sndchunks** command generates large amounts of data if there is any significant amount of traffic flowing. It should be used with extreme caution in live networks.

**Examples** The following example shows output for the **debug ip sctp sndchunks** command for a case in which data chunks are being sent, with some of them marked for retransmission:

```
Router# debug ip sctp sndchunks

SCTP: Assoc 0: ApplSend, chunk: 0/10412/100/A23134F8 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10443/100/A23134F9 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10448/100/A231355C to 10.5.0.4
SCTP: Assoc 0: Set oldest chunk for dest 10.5.0.4 to TSN A23134F8
SCTP: Assoc 0: Bundling data, added 0/10412/100/A23134F8, outstanding 100
SCTP: Assoc 0: Bundling data, added 5/10443/100/A23134F9, outstanding 200
SCTP: Assoc 0: Bundling data, added 4/10545/100/A23134FA, outstanding 300
SCTP: Assoc 0: Bundling data, added 10/10371/100/A23134FB, outstanding 400
SCTP: Assoc 0: Bundling data, added 11/10382/100/A23134FC, outstanding 500
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A231350F, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313510
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A2313527, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313528
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A231353F, numFrgs=0
```

```

SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313540
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A2313557, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A2313558
SCTP: Assoc 0: ApplSend, chunk: 10/10385/100/A23135BE to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 8/10230/100/A23135BF to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10459/100/A23135C0 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 4/10558/100/A23135C1 to 10.5.0.4
SCTP: Assoc 0: Set oldest chunk for dest 10.5.0.4 to TSN A231355D
SCTP: Assoc 0: Bundling data, added 5/10449/100/A231355D, outstanding 100
SCTP: Assoc 0: Bundling data, added 3/10490/100/A231355E, outstanding 200
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A23135A4, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A23135A5
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A23135BC, numFrgs=0
SCTP: Assoc 0: Reset oldest chunk on addr 10.5.0.4 to A23135BD
SCTP: Assoc 0: Process Sack Chunk, CumTSN=A23135C1, numFrgs=0
SCTP: Assoc 0: ApplSend, chunk: 5/10460/100/A23135C2 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 5/10461/100/A23135C3 to 10.5.0.4
SCTP: Assoc 0: ApplSend, chunk: 11/10403/100/A2313626 to 10.5.0.4
SCTP: Assoc 0: Set oldest chunk for dest 10.5.0.4 to TSN A23135C2
SCTP: Assoc 0: Bundling data, added 5/10460/100/A23135C2, outstanding 100
SCTP: Assoc 0: Bundling data, added 5/10461/100/A23135C3, outstanding 200
SCTP: Assoc 0: Bundling data, added 5/10462/100/A23135C4, outstanding 300
SCTP: Assoc 0: Bundling data, added 4/10559/100/A23135C5, outstanding 400
SCTP: Assoc 0: Bundling data, added 4/10560/100/A23135C6, outstanding 500
SCTP: Assoc 0: Bundled 12 chunk(s) in next dgram to 10.5.0.4
SCTP: Assoc 0: Bundling data, added 1/10418/100/A2313622, outstanding 9700
SCTP: Assoc 0: Bundling data, added 3/10502/100/A2313623, outstanding 9800
SCTP: Assoc 0: Bundling data, added 7/10482/100/A2313624, outstanding 9900
SCTP: Assoc 0: Bundling data, added 3/10503/100/A2313625, outstanding 10000
SCTP: Assoc 0: Bundling data, added 11/10403/100/A2313626, outstanding 10100
SCTP: Assoc 0: Bundled 5 chunk(s) in next dgram to 10.5.0.4
SCTP: Assoc 0: Mark chunk A23135C2 for retrans
SCTP: Assoc 0: Mark chunk A23135C3 for retrans
SCTP: Assoc 0: Mark chunk A23135C4 for retrans
SCTP: Assoc 0: Mark chunk A23135C5 for retrans
SCTP: Assoc 0: Mark chunk A23135C6 for retrans
SCTP: Assoc 0: Mark chunk A23135C7 for retrans
SCTP: Assoc 0: Mark chunk A23135C8 for retrans
SCTP: Assoc 0: Mark chunk A23135C9 for retrans
SCTP: Assoc 0: Mark chunk A23135CA for retrans
SCTP: Assoc 0: Bundled 6 chunk(s) in next dgram to 10.6.0.4
SCTP: Assoc 0: Mark chunk A23135C2 for retrans
SCTP: Assoc 0: Mark chunk A23135C3 for retrans
SCTP: Assoc 0: Mark chunk A23135C4 for retrans

```

Table 9 describes the significant fields shown in the display.

**Table 9** *debug ip sctp sndchunks Field Descriptions*

Field	Description
0 / 10412 / 100 / A23134F8	Stream number / datagram sequence number / chunk length, in bytes / chunk transmission sequence number.
outstanding	Number of bytes outstanding to the specified destination address.
CumTSN	Cumulative transmission sequence number (TSN).
numFrgs	Number of fragments sent.

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>clear ip sctp statistics</b>	Empties the buffer that holds SCTP statistics.
<b>debug ip sctp congestion</b>	Shows a list of all current SCTP associations.
<b>show ip sctp association parameters</b>	Shows the parameters configured for the association defined by the association identifier.
<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
<b>show ip sctp instances</b>	Shows all currently defined SCTP instances.
<b>show ip sctp statistics</b>	Shows overall statistics counts for SCTP.
<b>show iua as</b>	Shows information about the current condition of an application server.
<b>show iua asp</b>	Shows information about the current condition of an application server process.

# debug ip sctp state

To show state transitions in the Stream Control Transmission Protocol (SCTP), use the **debug ip sctp state** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp state**

**no debug ip sctp state**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** The **debug ip sctp state** command can be used to see if the current associations are stable or not. Because it generates output only on state transitions, it is safe to use in a live environment. It still should be used with caution, however, depending on the number of associations being handled by the system and the stability of the network.

The **debug ip sctp state** command is often used at the same time as the **debug ip sctp signal** command. Using the two commands together gives good insight into the stability of associations.

**Examples** In the following example, a new association is requested and established. The peer then restarts the association and notes that the association failed and is being reestablished. The local peer then indicates that the association has failed because it has tried to retransmit the specified chunk more than the maximum number of times without success. As a result, the association fails (because of communication loss) and is terminated. The upper-layer protocol (ULP) requests that the association be attempted again, and this attempt succeeds. A shutdown is then received from the remote peer, and the local peer enters the shutdown acknowledge sent state, which is followed by the association being terminated. Again, another association attempt is made and succeeds.

```
Router# debug ip sctp signal
Router# debug ip sctp state
```

```
<new assoc attempt>
00:20:08: SCTP: Assoc 0: state CLOSED -> COOKIE_WAIT
00:20:15: SCTP: Assoc 0: state COOKIE_WAIT -> ESTABLISHED
00:20:15: SCTP: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC
00:21:03: SCTP: Assoc 0: Restart rcvd from peer
00:21:03: SCTP: Assoc 0: Sent ASSOC_RESTART signal
00:21:04: SCTP: Assoc 0: chunk 62EA7F40 retransmitted more than max times, failing assoc
00:21:04: SCTP: Assoc 0: Sent ASSOC_FAILED signal, reason: SCTP_COMM_LOST
00:21:04: SCTP: Assoc 0: Sent ASSOC_TERMINATE signal
00:21:04: SCTP: Assoc 0: state ESTABLISHED -> CLOSED
<new assoc attempt>
```

```

00:21:04: Sctp: Assoc 0: state CLOSED -> COOKIE_WAIT
00:21:04: Sctp: Assoc 0: state COOKIE_WAIT -> COOKIE_ECHOED
00:21:04: Sctp: Assoc 0: state COOKIE_ECHOED -> ESTABLISHED
00:21:04: Sctp: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC
00:21:04: Sctp: Assoc 0: Sent TERMINATE_PENDING signal
00:21:04: Sctp: Assoc 0: state ESTABLISHED -> SHUTDOWN_ACKSENT
00:21:04: Sctp: Assoc 0: Sent ASSOC_TERMINATE signal
00:21:04: Sctp: Assoc 0: state SHUTDOWN_ACKSENT -> CLOSED
<new assoc attempt>
00:21:04: Sctp: Assoc 0: state CLOSED -> COOKIE_WAIT
00:21:04: Sctp: Assoc 0: state COOKIE_WAIT -> COOKIE_ECHOED
00:21:04: Sctp: Assoc 0: state COOKIE_ECHOED -> ESTABLISHED
00:21:04: Sctp: Assoc 0: Sent ASSOC_UP signal for CONFIGD_ASSOC

```

Table 10 describes the significant fields shown in the display.

**Table 10** debug ip sctp state Field Descriptions

Field	Description
CLOSED -> COOKIE_WAIT	SCTP endpoint sends initialization chunk and moves to the COOKIE-WAIT state to wait for acknowledgement and a state cookie from the remote endpoint.
COOKIE_WAIT -> COOKIE_ECHOED	SCTP endpoint returns the state cookie to the remote endpoint and enters COOKIE-ECHOED state.
COOKIE_ECHOED -> ESTABLISHED	SCTP endpoint enters ESTABLISHED state after receiving acknowledgement that the state cookie has been received by the remote endpoint.
ESTABLISHED -> SHUTDOWN_ACKSENT	SCTP endpoint enters SHUTDOWN-ACKSENT state after receiving a shutdown message and sending a shutdown acknowledgement to the remote endpoint.
SHUTDOWN_ACKSENT -> CLOSED	SCTP endpoint enters CLOSED state.

#### Related Commands

Command	Description
<b>clear ip sctp statistics</b>	Empties the buffer that holds SCTP statistics.
<b>debug ip sctp congestion</b>	Shows a list of all current SCTP associations.
<b>debug ip sctp signal</b>	Shows signals that are sent from SCTP to the application or upper-layer protocol (ULP).
<b>show ip sctp association parameters</b>	Shows the parameters configured for the association defined by the association identifier.
<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
<b>show ip sctp instances</b>	Shows all currently defined SCTP instances.
<b>show ip sctp statistics</b>	Shows overall statistics counts for SCTP.

■ debug ip sctp state

<b>Command</b>	<b>Description</b>
<b>show iua as</b>	Shows information about the current condition of an application server.
<b>show iua asp</b>	Shows information about the current condition of an application server process.

# debug ip sctp timer

To provide information about Stream Control Transmission Protocol (SCTP) timers that are started, stopped, and triggering, use the **debug ip sctp timer** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp timer**

**no debug ip sctp timer**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** Many SCTP timers should not be restarted after they have been started once. For these timers, the first call succeeds in starting the timer, and subsequent calls do nothing until the timer either expires or is stopped. For example, the retransmission timer is started when the first chunk is sent, but then is not started again for subsequent chunks when there is outstanding data.



**Caution**

The **debug ip sctp timer** command generates a significant amount of output. It should be used with extreme caution in a live network.

**Examples** The following example shows the starting and stopping of various SCTP timers:

```
Router# debug ip sctp timer

SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Timer BUNDLE triggered
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Stopping RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Starting RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Stopping RETRANS timer for destaddr 10.5.0.4
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
```

```

SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting
SCTP: Assoc 0: Stopping CUMSACK timer
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Assoc 0: Starting CUMSACK timer
SCTP: Timer already started, not restarting

```

Table 11 describes the significant fields shown in the display.

**Table 11** *debug ip sctp timer Field Descriptions*

Field	Description
CUMSACK	Cumulative selective acknowledgement.
RETRANS	Retransmission.

#### Related Commands

Command	Description
<b>clear ip sctp statistics</b>	Empties the buffer that holds SCTP statistics.
<b>debug ip sctp congestion</b>	Shows a list of all current SCTP associations.
<b>show ip sctp association parameters</b>	Shows the parameters configured for the association defined by the association identifier.
<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
<b>show ip sctp instances</b>	Shows all currently defined SCTP instances.
<b>show ip sctp statistics</b>	Shows overall statistics counts for SCTP.
<b>show iua as</b>	Shows information about the current condition of an application server.
<b>show iua asp</b>	Shows information about the current condition of an application server process.

# debug ip sctp warnings

To display diagnostic information about unusual situations in Stream Control Transmission Protocol (SCTP), use the **debug ip sctp warnings** command in privileged EXEC mode. To disable this diagnostic reporting, use the **no** form of this command.

**debug ip sctp warnings**

**no debug ip sctp warnings**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

**Usage Guidelines** In a live system, the debugs for performance, state, signal, and warnings are the most useful. They show any association or destination address failures and can be used to monitor the stability of established associations.

The **debug ip sctp warnings** command displays information on any unusual situation that is encountered. These situations may or may not indicate problems, depending on the particulars of the situation.

**Examples** The following example shows some events and conditions that are flagged as warnings:

```
Router# debug ip sctp warnings

SCTP: Assoc 0: No cookie in InitAck, discarding
SCTP: Assoc 0: Incoming INIT_ACK: inbound streams req'd 15, allowed 13
SCTP: Assoc 0: Incoming INIT_ACK request: outbound streams req'd 13, allowed 1
SCTP: Assoc 0: Remote verification tag in init ack is zero, discarding
SCTP: Remote verification tag in init is zero, discarding
SCTP: Assoc 0: Rwnd less than min allowed (1500) in incoming INITACK, rcvd 0
SCTP: Assoc 0: Rwnd less than min allowed (1500) in incoming INITACK, rcvd 1499
SCTP: Rwnd in INIT too small (0), discarding
SCTP: Rwnd in INIT too small (1499), discarding
SCTP: Unknown INIT param 16537 (0x4099), length 8
SCTP: Assoc 0: Unknown INITACK param 153 (0x99), length 8
SCTP: Assoc 0: No cookie in InitAck, discarding
SCTP: Assoc 0: No cookie in InitAck, discarding
SCTP: Processing INIT, invalid param len 0, discarding...
SCTP: Assoc 0: Processing INITACK, invalid param len 0, discarding...
```

Related Commands	Command	Description
	<b>clear ip sctp statistics</b>	Empties the buffer that holds SCTP statistics.
	<b>debug ip sctp congestion</b>	Shows a list of all current SCTP associations.
	<b>show ip sctp association parameters</b>	Shows the parameters configured for the association defined by the association identifier.
	<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
	<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
	<b>show ip sctp instances</b>	Shows all currently defined SCTP instances.
	<b>show ip sctp statistics</b>	Shows overall statistics counts for SCTP.
	<b>show iua as</b>	Shows information about the current condition of an application server.
	<b>show iua asp</b>	Shows information about the current condition of an application server process.

# show ip sctp association list

To display identifiers and information for current Stream Control Transmission Protocol (SCTP) associations and instances, use the **show ip sctp association list** command in privileged EXEC mode.

## show ip sctp association list

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(2)MB	This command was introduced as part of the <b>show ip sctp</b> command.
	12.2(2)T	This command was introduced as the <b>show ip sctp association list</b> command.

**Examples** The following is sample output from the **show ip sctp association list** command for three association identifiers:

```
Router# show ip sctp association list

*** Sctp Association List ***

AssocID:0, Instance ID:0
Current state:ESTABLISHED
Local port:8989, Addrs:10.1.0.2 10.2.0.2
Remote port:8989, Addrs:10.6.0.4 10.5.0.4

AssocID:1, Instance ID:0
Current state:ESTABLISHED
Local port:8989, Addrs:10.1.0.2 10.2.0.2
Remote port:8990, Addrs:10.6.0.4 10.5.0.4

AssocID:2, Instance ID:0
Current state:ESTABLISHED
Local port:8989, Addrs:10.1.0.2 10.2.0.2
Remote port:8991, Addrs:10.6.0.4 10.5.0.4
```

[Table 12](#) describes the significant fields shown in the display.

**Table 12** *show ip sctp association list* Field Descriptions

Field	Description
AssocID	SCTP association identifier.
Instance ID	SCTP association instance identifier.
Current state	SCTP association state, which can be ESTABLISHED, CLOSED, COOKIE-WAIT, or COOKIE-ECHOED.

**Table 12** *show ip sctp association list Field Descriptions (continued)*

<b>Field</b>	<b>Description</b>
Local port, Addr	Port and IP address for the local SCTP endpoint.
Remote port, Addr	Port and IP address for the remote SCTP endpoint.

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>clear ip sctp statistics</b>	Clears statistics counts for SCTP.
<b>debug ip sctp api</b>	Reports SCTP diagnostic information and messages.
<b>show ip sctp association parameters</b>	Shows the parameters configured for the association defined by the association identifier.
<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
<b>show ip sctp instances</b>	Shows the currently defined SCTP instances.
<b>show ip sctp statistics</b>	Shows the overall statistics counts for SCTP.
<b>show iua as</b>	Shows information about the current condition of an application server.
<b>show iua asp</b>	Shows information about the current condition of an application server process.

# show ip sctp association parameters

To show configured and calculated parameters for the specified Stream Control Transmission Protocol (SCTP) association, use the **show ip sctp association parameters** command in privileged EXEC mode.

**show ip sctp association parameters** *assoc-id*

<b>Syntax Description</b>	<i>assoc-id</i>	Association identifier, which can be obtained from the output of the <a href="#">show ip sctp association list</a> command.
---------------------------	-----------------	---

<b>Command Modes</b>	Privileged EXEC
----------------------	-----------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.2(2)MB	This command was introduced as part of the <b>show ip sctp</b> command.
	12.2(2)T	This command was introduced as the <b>show ip sctp association parameters</b> command.
	12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.
	12.2(8)T	Three new output fields were added to this command: Outstanding bytes, per destination address; Round trip time (RTT), per destination address; and Smoothed round trip time (SRTT), per destination address.

<b>Usage Guidelines</b>	This command provides information to determine the stability of SCTP associations, dynamically calculated statistics about destinations, and values to assess network congestion.
-------------------------	---

**Examples** The following example shows the IP SCTP association parameters for association 0:

```
Router# show ip sctp association parameters 0

** SCTP Association Parameters **

AssocID: 0 Context: 0 InstanceID: 1
Assoc state: ESTABLISHED Uptime: 19:05:57.425
Local port: 8181
Local addresses: 10.1.0.3 10.2.0.3

Remote port: 8181
Primary dest addr: 10.5.0.4
Effective primary dest addr: 10.5.0.4
Destination addresses:

10.5.0.4: State: ACTIVE
Heartbeats: Enabled Timeout: 30000 ms
RTO/RTT/SRTT: 1000/16/38 ms TOS: 0 MTU: 1500
cwnd: 5364 ssthresh: 3000 outstand: 768
Num retrans: 0 Max retrans: 5 Num times failed: 0

10.6.0.4: State: ACTIVE
Heartbeats: Enabled Timeout: 30000 ms
```

```
show ip sctp association parameters
```

```

RTO/RTT/SRTT: 1000/4/7 ms  TOS: 0  MTU: 1500
cwnd: 3960  ssthresh: 3000  outstand: 0
Num retrans: 0  Max retrans: 5  Num times failed: 0

Local vertag: 9A245CD4  Remote vertag: 2A08D122
Num inbound streams: 10  outbound streams: 10
Max assoc retrans: 5  Max init retrans: 8
CumSack timeout: 200 ms  Bundle timeout: 100 ms
Min RTO: 1000 ms  Max RTO: 60000 ms
LocalRwnd: 18000  Low: 13455  RemoteRwnd: 15252  Low: 13161
Congest levels: 0  current level: 0  high mark: 325

```

Table 13 describes the significant fields shown in the display.

**Table 13** *show ip sctp association parameters Field Descriptions*

Field	Description
AssocID	SCTP association identifier.
Context	Internal upper-layer handle.
InstanceID	SCTP association instance identifier.
Assoc state	SCTP association state, which can be ESTABLISHED, CLOSED, COOKIE-WAIT, or COOKIE-ECHOED.
Uptime	Duration of time that the association has been active.
Local port	Port number for the local SCTP endpoint.
Local addresses	IP addresses for the local SCTP endpoint.
Remote port	Port number for the remote SCTP endpoint.
Primary dest addr	Primary destination address.
Effective primary dest addr	Current primary destination address.
Heartbeats	Status of heartbeats.
Timeout	Heartbeat timeout.
RTO/RTT/SRTT	Retransmission timeout, round trip time, and smoothed round trip time, calculated from network feedback.
TOS	IP precedence setting.
MTU	Maximum transmission unit.
cwnd	Congestion window value calculated from network feedback. The amount of data that can be outstanding in the network for that particular destination.
ssthresh	Slow-start threshold value calculated from network feedback.
outstand	Number of outstanding bytes.
Num retrans	Current number of times data has been retransmitted to that address.
Max retrans	Maximum number of times data has been retransmitted to that address.
Num times failed	Number of times the address has been marked as failed.
Local vertag, Remote vertag	The verification tags (vertags) are chosen during association initialization and do not change.
Num inbound streams, outbound streams	Maximum inbound and outbound streams. This number does not change.

**Table 13** *show ip sctp association parameters Field Descriptions (continued)*

<b>Field</b>	<b>Description</b>
Max assoc retrans	Maximum association retransmit limit is the number of times any particular chunk may be retransmitted before a declaration that the association failed, which indicates that the chunk could not be delivered on any address.
Max init retrans	Maximum initial retransmit limit is the number of times the chunks for initialization may be retransmitted before declaring that the attempt to establish the association failed.
CumSack timeout	Cumulative selective acknowledge (SACK) timeout. The maximum time that a SACK may be delayed while attempting to bundle together with data chunks.
Bundle timeout	Maximum time that data chunks may be delayed while attempting to bundle with other data chunks.
Min RTO, Max RTO	Minimum and maximum retransmit timeout values allowed for the association.
LocalRwnd, RemoteRwnd	Local and remote receive windows.
Congest levels: current level, high mark	Current congestion level, highest number of packets queued.

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>clear ip sctp statistics</b>	Clears statistics counts for SCTP.
<b>debug ip sctp api</b>	Reports SCTP diagnostic information and messages.
<b>debug ip sctp congestion</b>	Shows a list of all current SCTP associations.
<b>show ip sctp association list</b>	Displays identifiers and information for current SCTP associations and instances.
<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
<b>show ip sctp instances</b>	Shows all currently defined SCTP instances.
<b>show ip sctp statistics</b>	Shows overall statistics counts for SCTP.
<b>show iua as</b>	Shows information about the current condition of an application server.
<b>show iua asp</b>	Shows information about the current condition of an application server process.

# show ip sctp association statistics

To show statistics that have accumulated for the specified Stream Control Transmission Protocol (SCTP) association, use the **show ip sctp association statistics** command in privileged EXEC mode.

**show ip sctp association statistics** *assoc-id*

## Syntax Description

*assoc-id* Association identifier, which can be obtained from the output of the [show ip sctp association list](#) command.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.2(2)MB	This command was introduced as part of the <b>show ip sctp</b> command.
12.2(2)T	This command was introduced as the <b>show ip sctp association statistics</b> command.
12.2(8)T	Two new output fields were added to this command: <ul style="list-style-type: none"> <li>• Number of unordered data chunks sent</li> <li>• Number of unordered data chunks received</li> </ul>

## Usage Guidelines

This command shows only the information that has become available since the last time a [clear ip sctp statistics](#) command was executed.

## Examples

The following example shows the statistics accumulated for SCTP association 0:

```
Router# show ip sctp association statistics 0

** Sctp Association Statistics **

AssocID/InstanceID: 0/1
Current State: ESTABLISHED
Control Chunks
  Sent: 623874  Rcvd: 660227
Data Chunks Sent
  Total: 14235644  Retransmitted: 60487
  Ordered: 6369678  Unordered: 6371263
  Avg bundled: 18  Total Bytes: 640603980
Data Chunks Rcvd
  Total: 14496585  Discarded: 1755575
  Ordered: 6369741  Unordered: 6371269
  Avg bundled: 18  Total Bytes: 652346325
  Out of Seq TSN: 3069353
ULP Dgrams
  Sent: 12740941  Ready: 12740961  Rcvd: 12740941
```

Table 14 describes the significant fields shown in the display.

**Table 14** *show ip sctp association statistics Field Descriptions*

Field	Description
AssocID/InstanceID	SCTP association identifier and instance identifier.
Current State	State of SCTP association.
Control Chunks	SCTP control chunks sent and received.
Data Chunks Sent	SCTP data chunks sent.
Data Chunks Rcvd	SCTP data chunks received.
ULP Dgrams	Number of datagrams sent and received by the Upper Layer Protocol (ULP).

#### Related Commands

Command	Description
<a href="#">clear ip sctp statistics</a>	Clears statistics counts for SCTP.
<a href="#">debug ip sctp api</a>	Reports SCTP diagnostic information and messages.
<a href="#">debug ip sctp congestion</a>	Shows a list of all current SCTP associations.
<a href="#">show ip sctp association list</a>	Displays identifiers and information for current SCTP associations and instances.
<a href="#">show ip sctp association parameters</a>	Shows the parameters configured for the association defined by the association identifier.
<a href="#">show ip sctp errors</a>	Shows error counts logged by SCTP.
<a href="#">show ip sctp instances</a>	Shows all currently defined SCTP instances.
<a href="#">show ip sctp statistics</a>	Shows overall statistics counts for SCTP.
<a href="#">show iua as</a>	Shows information about the current condition of an application server.
<a href="#">show iua asp</a>	Shows information about the current condition of an application server process.

# show ip sctp errors

To show the error counts logged by the Stream Control Transmission Protocol (SCTP), use the **show ip sctp errors** command in privileged EXEC mode.

**show ip sctp errors**

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(2)MB	This command was introduced as part of the <b>show ip sctp</b> command.
	12.2(2)T	This command was introduced as the <b>show ip sctp errors</b> command.

**Usage Guidelines** This command displays all errors across all associations that have been logged since the last time that the SCTP statistics were cleared using the **clear ip sctp statistics** command. If no errors have been logged, this is indicated in the output.

**Examples** The following example shows a session with no errors:

```
Router# show ip sctp errors

*** Sctp Error Statistics ***
No Sctp errors logged.
```

The following example shows a session with SCTP errors:

```
Router# show ip sctp errors

** Sctp Error Statistics **
Invalid verification tag:      5
Communication Lost:           64
Destination Address Failed:   3
Unknown INIT params rcvd:    16
Invalid cookie signature:     5
Expired cookie:               1
Peer restarted:               1
No Listening instance:         2
Missing parameters:           36
```

Related Commands	Command	Description
	<b>clear ip sctp statistics</b>	Clears statistics counts for SCTP.
	<b>debug ip sctp api</b>	Reports SCTP diagnostic information and messages.
	<b>debug ip sctp congestion</b>	Shows a list of all current SCTP associations.

Command	Description
<b>show ip sctp association parameters</b>	Shows the parameters configured for the association defined by the association ID.
<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association ID.
<b>show ip sctp instances</b>	Shows the currently defined SCTP instances.
<b>show ip sctp statistics</b>	Shows overall statistics counts for SCTP.
<b>show iua as</b>	Shows information about the current condition of an AS.
<b>show iua asp</b>	Shows information about the current condition of an ASP.

# show ip sctp instances

To display information for each of the currently configured Stream Control Transmission Protocol (SCTP) instances, use the **show ip sctp instances** command in privileged EXEC mode.

## show ip sctp instances

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(2)MB	This command was introduced as part of the <b>show ip sctp</b> command.
	12.2(2)T	This command was introduced as the <b>show ip sctp instances</b> command.

**Examples** The following example shows available IP SCTP instances:

```
Router# show ip sctp instances

*** SCTP Instances ***

Instance ID:0 Local port:8989
Instance state:available
Local addrs:10.1.0.2 10.2.0.2
Default streams inbound:1 outbound:1
Current associations: (max allowed:6)
  AssocID:0 State:ESTABLISHED Remote port:8989
    Dest addrs:10.6.0.4 10.5.0.4
  AssocID:1 State:ESTABLISHED Remote port:8990
    Dest addrs:10.6.0.4 10.5.0.4
  AssocID:2 State:ESTABLISHED Remote port:8991
    Dest addrs:10.6.0.4 10.5.0.4

Instance ID:1 Local port:9191
Instance state:available
Local addrs:10.1.0.2 10.2.0.2
Default streams inbound:1 outbound:1

No current associations established for this instance.
Max allowed:6
```

Related Commands	Command	Description
	<a href="#">clear ip sctp statistics</a>	Clears statistics counts for SCTP.
	<a href="#">debug ip sctp api</a>	Reports SCTP diagnostic information and messages.
	<a href="#">debug ip sctp congestion</a>	Shows a list of all current SCTP associations.

Command	Description
<b>show ip sctp association parameters</b>	Shows the parameters configured for the association defined by the association identifier.
<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
<b>show ip sctp statistics</b>	Shows the overall statistics counts for SCTP.
<b>show iua as</b>	Shows information about the current condition of an AS.
<b>show iua asp</b>	Shows information about the current condition of an ASP.

# show ip sctp statistics

To show the overall statistics counts for Stream Control Transmission Protocol (SCTP) activity, use the **show ip sctp statistics** command in privileged EXEC mode.

## show ip sctp statistics

**Syntax Description** This command has no arguments or keywords.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(2)MB	This command was introduced as part of the <b>show ip sctp</b> command.
	12.2(2)T	This command was introduced as the <b>show ip sctp statistics</b> command.

**Usage Guidelines** The **show ip sctp statistics** command displays the overall SCTP statistics accumulated since the last **clear ip sctp statistics** command. It includes numbers for all currently established associations, as well as for any that have been terminated. The statistics indicated are similar to those shown for individual associations, with an additional section showing overall association statistics.

**Examples** The following example shows IP SCTP statistics:

```
Router# show ip sctp statistics

*** SCTP Overall Statistics ***

Total Chunks Sent:          36380
Total Chunks Rcvd:         36599

Data Chunks Rcvd In Seq:   34042
Data Chunks Rcvd Out of Seq:192
Total Data Chunks Sent:    34213
Total Data Chunks Rcvd:    34234
Total Data Bytes Sent:     3412962
Total Data Bytes Rcvd:     3415025
Total Data Chunks Discarded:0
Total Data Chunks Retrans: 0

Total SCTP Dgrams Sent:    5544
Total SCTP Dgrams Rcvd:    5935
Total ULP Dgrams Sent:     34234
Total ULP Dgrams Ready:    34234
Total ULP Dgrams Rcvd:     34234
```

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>clear ip sctp statistics</b>	Clears statistics counts for SCTP.
	<b>debug ip sctp api</b>	Reports SCTP diagnostic information and messages.
	<b>debug ip sctp congestion</b>	Shows a list of all current SCTP associations.
	<b>show ip sctp association parameters</b>	Shows the parameters configured and calculated for the association defined by the association identifier.
	<b>show ip sctp association statistics</b>	Shows the current statistics for the association defined by the association identifier.
	<b>show ip sctp errors</b>	Shows error counts logged by SCTP.
	<b>show ip sctp instances</b>	Shows all currently defined SCTP instances.
	<b>show iua as</b>	Shows information about the current condition of an AS.
	<b>show iua asp</b>	Shows information about the current condition of an ASP.

# Glossary

**association**—A protocol relationship between SCTP endpoints, that comprises the two SCTP endpoints and all protocol state information.

**chunk**—A block of information contained in an SCTP datagram. It may be a control chunk that contains some type of information to control the association or a data chunk that contains ULP data. If a ULP datagram is too large to send in one chunk as defined by the current MTU, it is segmented into multiple smaller chunks.

**cwnd**—congestion window. An SCTP variable, in number of bytes, that limits the data transmissions that a user can send to a particular destination transport address before receiving an acknowledgement.

**MTU**—maximum transmission unit. Maximum packet size, in bytes, that a particular interface can handle.

**multihomed**—The ability to have more than one IP address. An SCTP endpoint is considered multihomed if there is more than one transport address that can be used as a destination address to reach that endpoint. All transport addresses used by an SCTP endpoint must use the same port number, but can use multiple IP addresses.

**RTO**—retransmission timeout. The duration of the SCTP endpoint retransmission timer (T3-rtx), which ensures data delivery in the absence of any feedback from the endpoint peer.

**rwnd**—receiver window. An SCTP variable that a data sender uses to store the most recently calculated receiver window of its peer, in number of bytes. The receiver window gives the data sender an indication of the space available in the receiver's inbound buffer.

**SACK**—selective acknowledgement. The control chunk type that is used to acknowledge all data chunks within an association.

**SCTP**—Stream Control Transmission Protocol. Protocol designed by the Signaling Transport working group of the Internet Engineering Task Force (IETF) to transport messages reliably over IP networks. Though specifically designed with PSTN signaling in mind, SCTP is meant to be a general IP transport protocol.

**SCTP datagram**—A datagram sent to or received from the network. An SCTP datagram has an SCTP common header followed by one or more control chunks or data chunks, or both. An SCTP datagram is distinguished from a ULP datagram, which is received from or sent to the ULP or application. A datagram can be segmented into multiple data chunks if it is larger than the smallest MTU size.

**SCTP endpoint**—The logical sender or receiver of SCTP packets. On a multihomed host, an SCTP endpoint is represented to its peers as a combination of two sets of addresses: the set of eligible destination transport addresses to which SCTP packets can be sent and the set of eligible source transport addresses from which SCTP packets can be received. All transport addresses used by an SCTP endpoint must use the same port number, but they can use multiple IP addresses (see *multihoming*). A transport address used by an SCTP endpoint must not be used by another SCTP endpoint. In other words, a transport address is unique to an SCTP endpoint.

**SS7**—Signaling System 7. Standard CCS system used with BISDN and ISDN. Developed by Bellcore.

**ssthresh**—slow-start threshold. An SCTP variable that contains the threshold that the endpoint uses to determine whether to perform slow-start or congestion avoidance on a particular destination transport address. Ssthresh is in number of bytes.

**stream**—In SCTP, a stream refers to a sequence of user messages that are to be delivered to the upper-layer protocol. All the user messages in a stream must be delivered in the order they entered the stream. This definition of *stream* is in contrast to its usage in TCP, where *stream* refers to a sequence of bytes.

**TSN**—transmission sequence number. A unique, sequential 32-bit number assigned to each individual data chunk to be handled in an association. The TSN numbers are assigned over the association as a whole and do not account for individual streams.

**ULP**—upper-layer protocol. The logical higher-layer application that uses the services of SCTP.

**ULP datagram**—A datagram received from or sent to the ULP or application. This is distinguished from an SCTP datagram, which is received from or sent to the IP network.

