



PRI Backhaul and IUA Support Using SCTP

This chapter describes how to implement Stream Control Transmission Protocol (SCTP) features. SCTP is not explicitly configured on routers, but it underlies several Cisco applications. This chapter describes how to configure several features that use SCTP and how to troubleshoot SCTP problems.

SCTP is used with the following Cisco IOS software features:

- PRI Backhaul Using SCTP and the ISDN Q.921 User Adaptation Layer
- Support for IUA with SCTP for Cisco Access Servers

Feature History for PRI Backhaul Using SCTP and the ISDN Q.921 User Adaptation Layer

| Release | Modification |
|------------|---|
| 12.1(1)T | This feature was introduced on the Cisco AS5300. |
| 12.2(4)T | This feature was introduced on the Cisco 2600 series, Cisco 3600 series, and Cisco MC3810 series. |
| 12.2(2)XB1 | This feature was implemented on the Cisco AS5850. |

Feature History for Support for IUA with SCTP for Cisco Access Servers

| Release | Modification |
|-----------|------------------------------|
| 12.2(15)T | This feature was introduced. |

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.



Note

For more information about related Cisco IOS voice features, see the following:

- “[Overview of ISDN Voice Interfaces](#)” on page 3
- Entire Cisco IOS Voice Configuration Library—including library preface and glossary, other feature documents, and troubleshooting documentation—at http://www.cisco.com/en/US/products/ps6441/prod_configuration_guide09186a0080565f8a.html.

For a list of references cited in this chapter, see the “[Additional References](#)” section on page 274.

Contents

- [Prerequisites for Implementing SCTP Features](#), page 220
- [Restrictions for Implementing SCTP Features](#), page 220
- [Information About SCTP and SCTP Features](#), page 221
- [How to Configure SCTP Features](#)
- [Configuration Examples for SCTP Options](#), page 260
- [Additional References](#), page 274

Prerequisites for Implementing SCTP Features

- Perform the prerequisites that are listed in the [“Prerequisites for Configuring an ISDN Voice Interface”](#) section on page 15.

PRI Backhaul Using SCTP and the ISDN Q.921 User Adaptation Layer Feature

- Configure ISDN to backhaul Q.921 signaling to the media gateway controller
- For Cisco AS5850, install or implement the following:
 - MGCP 1.0
 - IUA 0.4
 - ISDN network-side support to terminate multiple voice PRIs

Restrictions for Implementing SCTP Features

Restrictions are described in the [“Restrictions for Configuring ISDN Voice Interfaces”](#) section on page 4. In addition, the following apply.

PRI Backhaul Using SCTP and the ISDN Q.921 User Adaptation Layer Feature

- Backhaul: Does not support backhauling for Basic Rate Interface (BRI).
- Capacity: Supports only two application-server processes (ASPs) per application server. Supports only three explicit IP addresses per SCTP association endpoint.
- IUA messages: Does not support new-traffic failover.



Note

The IUA specification describes an optional feature known as New Traffic Failover. In this process, all messages for calls pending completion during failover are sent to the inactive media-gateway controller, and messages for new calls are sent to the newly active controller. These IUA messages for new calls are not supported.

- Load balancing: Does not support load balancing between ASPs on a per-call basis.
- Platforms: Is not supported on the Cisco 2600XM series, Cisco 2691, Cisco 2800 series, Cisco 3700 series, and Cisco 3800 series.

- Signaling: Supports Facility Associated Signaling (FAS) and Non-Facility Associated Signaling (NFAS) PRI D-channel signaling only; does not support any other signaling protocols, including NFAS with backup D-channel signaling.

Support for IUA with SCTP for Cisco Access Servers Feature

- Backhaul: Does not support Q.931 PRI backhaul on the Cisco PGW 2200.
- Platforms: Is not supported on the Cisco 2600XM series or Cisco 2691.
- Transport: Does not support concurrent Redundant Link Manager (RLM) and SCTP transport on the access-server gateway. You can configure one or the other but not both at the same time.



Note

- For more information about the Cisco PGW 2200, see [Support for IUA with SCTP](#).
- For more information about IUA with SCTP, see [Support for IUA with SCTP for Cisco Access Servers](#).

Information About SCTP and SCTP Features



Note

General information about ISDN voice interfaces is presented in the “[Information About ISDN Voice Interfaces](#)” section on page 4.

To configure SCTP, you should understand the following concepts:

- [SCTP Topology, page 221](#)
- [IUA, page 223](#)
- [Multiple NFAS Groups, page 223](#)
- [Features That Use SCTP, page 225](#)

SCTP Topology

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, within the context of an association between two SCTP hosts. SCTP is connection-oriented, but SCTP association is a broader concept than, for example, TCP connection.

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

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
- Interoperability with third-party call agents

SCTP allows you to terminate multiple switches and trunk groups on a gateway to add scalability. Adding trunk groups does not require more memory or processing resources because SCTP supports multiple streams in a single SCTP association. SCTP is a reliable transport protocol for message-oriented communications; SCTP is specifically designed to support PSTN signaling messages over IP networks.

SCTP allows you to configure at least one trunk group per T1 or E1 interface available on a given platform. A gateway platform with four T1 or E1 interfaces, for example, can control four unique trunk groups per device. Certain platforms, such as the Cisco AS5800 and Cisco AS5850, can deliver the individual T1 or E1 trunk groups over a high-speed interface, such as T3, which operates at 45 Mbps.

Table 48 shows the number of trunk groups supported per gateway platform.

Table 48 SS7 Interconnect for Voice-Gateway Trunk Groups per Gateway

| Platform | Supported Trunk Groups | Comments |
|------------------|------------------------|--|
| Cisco AS5300 | 4 | Verify both T1 and E1 cards. |
| Cisco AS5350 | 8 | Verify both T1 and E1 cards. Verify with Integrated SLT option. Note For more information, see <i>Integrated Signaling Link Terminal</i> , Cisco IOS Release 12.2(11)T. |
| Cisco AS5350 CT3 | 28 | Verify CT3 DS-3 card. Verify with Integrated SLT option. |
| Cisco AS5400 | 16 | Verify both T1 and E1 cards. Verify with Integrated SLT option. |
| Cisco AS5400 CT3 | 28 | Verify CT3 DS-3 card. Verify with Integrated SLT option. |
| Cisco AS5850 | 112 | Verify E1 cards and CT3 DS-3 cards. Note T1 ports and the 112 supported trunk groups are available only with CT3 cards. |

In a typical network topology, only one SCTP association is configured between a signaling controller and a gateway. Multiple IP addresses on either side can be designated to the same association to achieve link redundancy. On a gateway, signaling messages for all trunk groups are carried over on the same SCTP association to the same signaling controller. Trunk groups on a gateway can also be controlled through different signaling controllers. In such cases, you can configure multiple associations on a gateway and direct them to different signaling controllers.

IUA

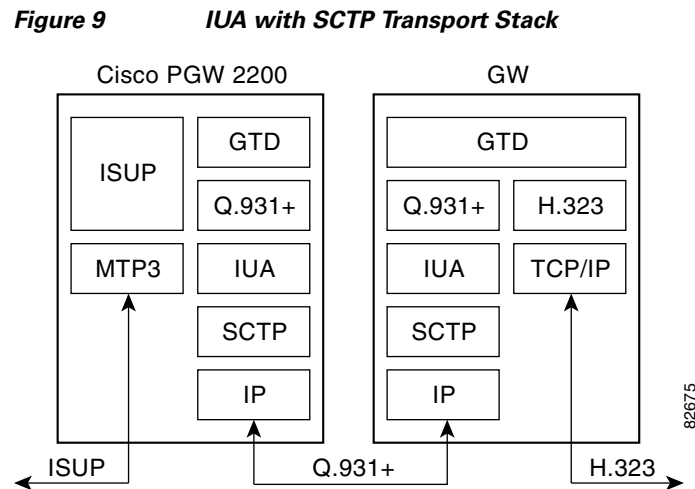
IUA is the adaptation layer that makes SCTP services available to Q.921 services users, such as Q.931, Q Signaling (QSIG), and National ISDN-2 with Cisco extensions (Cisco NI2+). IUA supports the standard interlayer primitives provided by Q.921. As a result, an upper-layer protocol (ULP) that typically used Q.921 services can easily migrate to IUA.

IUA service points are represented to the upper-layer protocol as application servers. Each application server is bound to an SCTP local endpoint managed by an SCTP instance. A remote signaling controller is known as an ASP. An ASP is connected to the local endpoint through a single SCTP association.

The IUA module creates associations between the signaling gateway and the MGC based on configuration requests. It also manages multiple ASPs as defined in the IETF IUA specification. IUA performs the following functions:

- Requests SCTP associations based on configuration information.
- Manages the destination address list and requests a new primary destination in the event of a failure.
- Manages the ASP state machine for each association.
- Manages the application-server state machine across all ASPs associated with a single application.
- Provides service for multiple applications simultaneously to handle different Layer 3 signaling protocols (Q.931 and Q.SIG, for example), or to communicate with different sets of call agents.

Figure 9 shows IUA with SCTP transport stack.



To use IUA services, you must make the application server and ASP available and bind a trunk group to an application server for its Layer 2 server. For configuration information, see the [“Configure IUA” section on page 230](#).

Multiple NFAS Groups

On a gateway, trunk groups are defined as Non-Facility Associated Signaling (NFAS) groups. An NFAS group is a group of ISDN PRI trunks with a single dedicated D channel. In a voice-gateway solution, the D channel in a trunk group is symbolic because SS7 is used as the signaling mechanism. The D channels defined for each NFAS group are actually DS0 bearer channels for voice or modem calls. Therefore, each NFAS has a corresponding D channel for which it is allocated.

A symbolic D-channel interface is dedicated to a trunk group. Each D-channel interface is bound to an application server and a dedicated stream is associated with this interface. Thus, the NFAS group identification can be recovered on each side of the SCTP association through this two-stage mapping as long as both sides share the same configuration information. Multiplexing of multiple trunk groups through a single association is accomplished this way, for example. If all interfaces on a gateway are controlled through a single SC, all interfaces are bound to the same application server.

The SCTP stream is a logical identification of the grouping of messages and consumes little additional memory and processing power. Each association can support as many as 65,355 streams.

Figure 10 shows the mapping between the trunk group, D-channel interface, and SCTP stream.

Figure 10 Mapping Between Trunk Group, Interface, and Stream

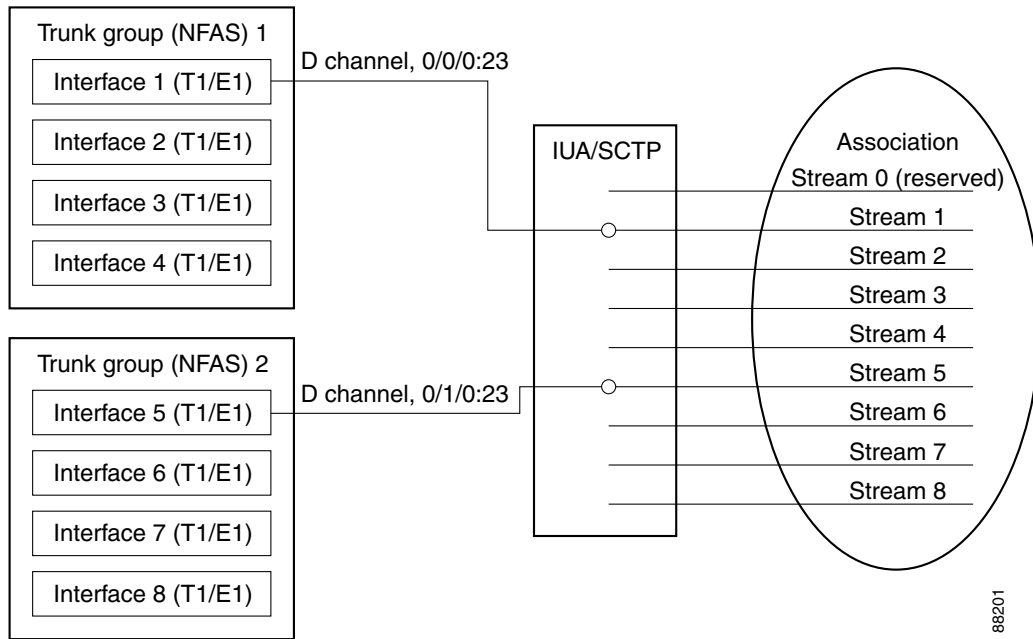
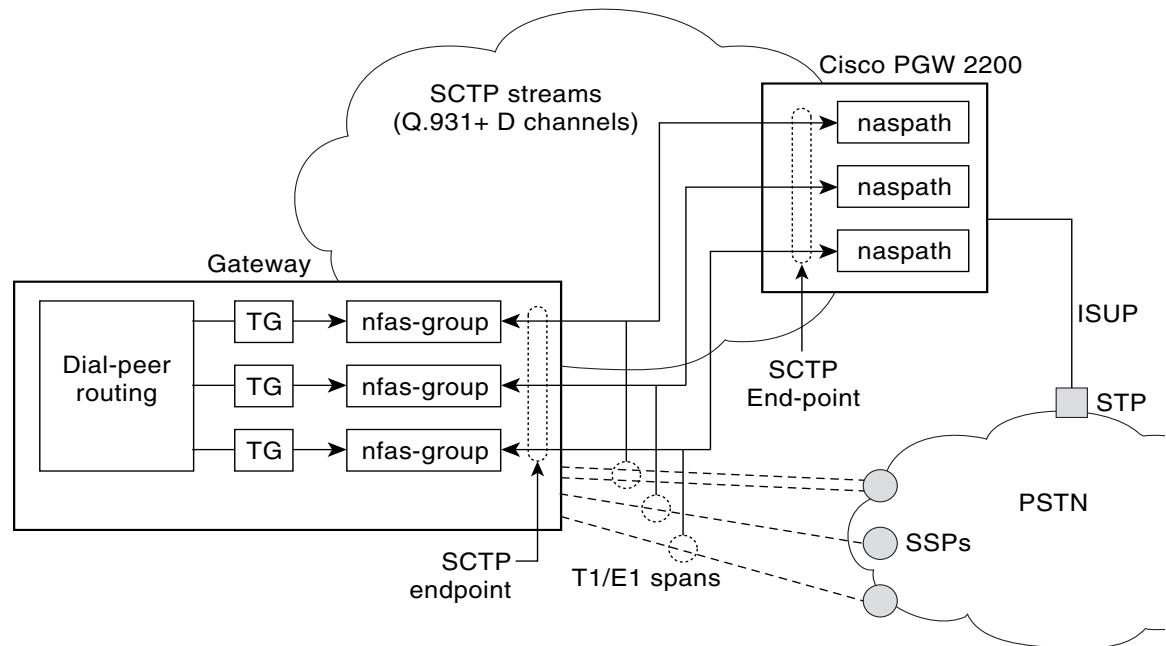


Figure 11 shows the NFAS group and SCTP association.

Figure 11 NFAS Group and SCTP Association



The IUA transport protocol using SCTP is supported on the Cisco PGW 2200; the Cisco PGW 2200 now uses IUA to communicate with Cisco access servers.

IUA with SCTP on the Cisco PGW 2200 provides the following services:

- Eliminates the scaling limitations in previous releases of Cisco MGC software for the number of NFAS-groups allowed per RLM.
- Supports upgrading from RLM-based communication to IUA-based communication without losing stable active calls.
- RLM-based communication is still supported. However, since this is a new functionality, the backward compatibility of the SCTP-based transports is not applicable.
- IUA interface can be used with Cisco access servers that support NAS and Digital Private Network Signaling System (DPNSS) signaling.
- Introduces IUA and SCTP operational measurements.



Note

For more information about IUA and SCTP on the Cisco PGW 2200, see [Support for IUA with SCTP](#).

Features That Use SCTP

The following features use SCTP:

- [PRI Backhaul Using SCTP and the ISDN Q.921 User Adaptation Layer](#), page 226
- [Support for IUA with SCTP for Cisco Access Servers](#), page 228

PRI Backhaul Using SCTP and the ISDN Q.921 User Adaptation Layer

This feature (sometimes called PRI Q.921 Signaling Backhaul) provides standards-based ISDN signaling backhaul capability on Cisco IOS gateways. It fills the need for PRI Q.921 signaling backhaul that works with third-party call agents or media-gateway controllers (MGCs) where call processing for voice calls is carried out by call-control servers, and packet-network connections are made using protocols such as Media Gateway Control Protocol (MGCP) and Simple Gateway Control Protocol (SGCP). It enables solutions such as Integrated Access, IP PBX, and Telecommuter on the Cisco 3600 series, Cisco AS5300, and Cisco AS5850. It provides a configuration interface for Cisco IOS software implementation and implements protocol message flows for SCTP and IUA.

This feature provides the following:

- PRI backhaul—Specific implementation for backhauling PRI



Note For more information about PRI backhaul using SCTP, see [PRI Backhaul Using the Stream Control Transmission Protocol and the ISDN Q.921 User Adaptation Layer](#).

- SCTP—New general-transport protocol that can be used for backhauling signaling messages
- ISDN User Adaptation Layer (IUA)—Mechanism for backhauling any Layer 3 protocol that normally uses Q.921

This feature supports interoperability with third-party call agents. It also supports the following solutions that require signaling backhaul:

- IP PBX
- IP Centrex
- Enterprise toll bypass
- IXC/tandem bypass

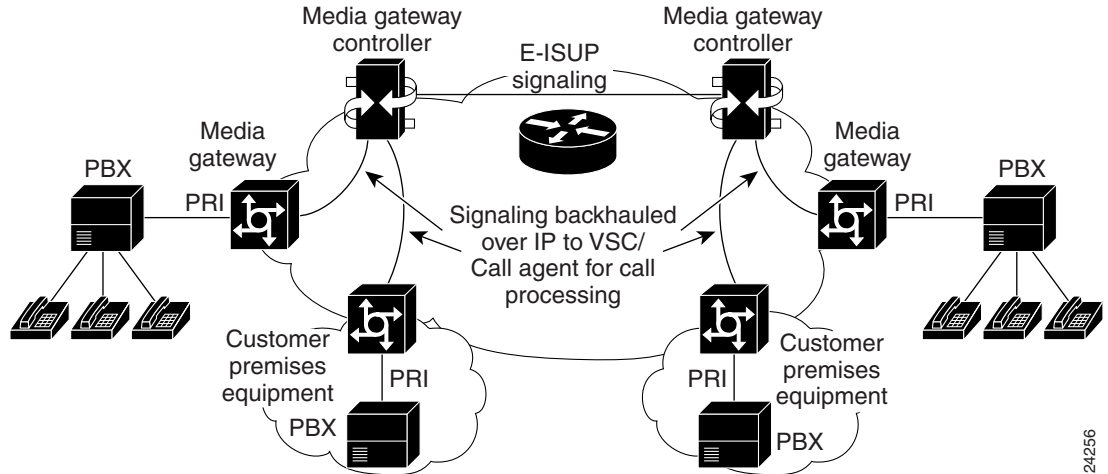
Signaling backhaul facilitates the handling of voice traffic coming from the signaling endpoints that communicate using facility-associated signaling. Facility-associated signaling requires the signaling channel (channel that carries call-signaling information) to share a digital facility with the bearer channels. ISDN PRI is one example of facility-associated signaling. ISDN signaling backhaul is required in the MGCP-based call-control architecture to enable end-to-end voice solutions.

This feature implements the IETF standards-based signaling backhaul protocols. This standards-based signaling transport support enables any third-party call agent that supports the standards to work with Cisco gateways. ISDN signaling backhaul is required in the MGCP-based call-control architecture to enable end-to-end voice solutions.

This feature migrates the proprietary PRI backhaul infrastructure to open standards. Backhaul is carried out using industry-standard SCTPs and ISDN IUA protocols as defined by the SIGTRAN working group of the IETF. It supports backhauling for ISDN-based signaling protocols only.

Figure 12 shows an example of PRI signaling backhaul. The MGC provides call processing and gateway control.

Figure 12 PRI Signaling Backhaul

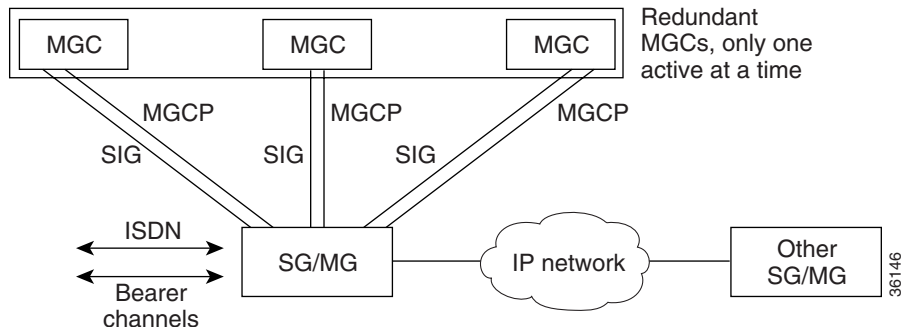


Ordinarily, signaling backhaul occurs at a common boundary for all protocols. For ISDN, signaling backhaul occurs at the Layer 2 (Q.921) and Layer 3 (Q.931) boundaries. The lower layers of the protocol (Q.921) are terminated and processed on the gateway, while the upper layers (Q.931) are backhauled to the MGC using SCTP. Signaling backhaul provides the advantage of distributed protocol processing. This permits greater expandability and scalability while offloading lower-layer protocol processing from the MGC.

Signaling transport between entities is applied to ensure that signaling information is transported with the required functionality and performance. The signaling gateway or MGC receives both ISDN signaling and bearer-channel data. ISDN signaling is backhauled up to an MGC or call agent using the SIG protocol stack. You can configure each signaling gateway to use up to three MGCs within an application server for redundancy. Multiple application servers can also be supported on a signaling gateway. MGCP is then used to control the bearer channels.

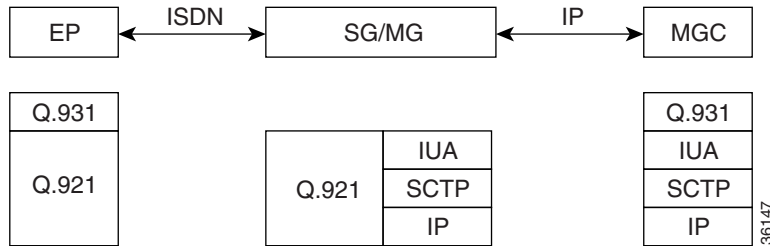
Figure 13 shows the functional model for PRI signaling transport.

Figure 13 Signaling Transport Model



SCTP is a peer-to-peer protocol; IUA is a client-server protocol. Figure 14 shows the protocol flow from an ISDN endpoint, through the signaling gateway, and then to a call agent or media gateway controller.

Figure 14 Protocol Flow



PRI Backhaul Using the Stream Control Transmission Protocol and the ISDN Q.921 User Adaptation Layer on the Cisco 3660 supports the following on a Cisco 3660:

- 20 calls per hour per DS-0 bearer circuit (3-minute average call duration)
- 460 calls per hour per PRI circuit (23 bearer channels): $20 \times 23 = 460$
- 5520 calls per hour per Cisco 3660 (12 PRI trunks): $460 \times 12 = 5520$
- 1.5333 calls per Cisco 3660 per second. 5520 divided by $(60 \times 60) = 1.5333$
- 7 signaling messages per call (both setup and tear down)
- 10.8 signaling messages per second per Cisco 3660: $7 \times 1.5333 = 10.8$

Support for IUA with SCTP for Cisco Access Servers

This feature supports IUA with SCTP for the Cisco AS5x00, Cisco 2420, Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series. It is to be used as an alternative to the existing IP-based User Datagram Protocol (UDP)-to-Reliable Link Manager (RLM) transport between the Cisco PGW 2200 and Cisco gateways.

IUA with SCTP acts as the call signaling IP transport mechanism in a voice-gateway solution. These combined protocols are also used for Signaling System 7 (SS7) Interconnect solutions, which allow required flexibility in connecting Intermachine trunks from more than one PSTN switch (multiple trunk groups) to the Cisco gateways. This feature also allows you to interconnect with multiple carriers on high-capacity Cisco AS5x00 gateways for load balancing and redundancy.

IUA and SCTP protocols provide the following services:

- Trunk groups are defined on a T1/E1 interface basis.
- All DS0 bearer channels in a specific T1/E1 interface are included in the same trunk group and cannot be split into different trunk groups.
- Multiple T1/E1 interfaces on the same gateway can be provisioned in a single trunk group or split into multiple trunk groups. The maximum number of trunk groups that a platform can support is equal to the maximum number of T1/E1 interfaces that the platform can configure.

This feature supports SCTP, multiple non-facility associated signaling, and IUA.

How to Configure SCTP Features

This section contains the following procedures:

- [Configuring PRI Backhaul Using the SCTP and the ISDN Q.921 User Adaptation Layer, page 229](#)
- [Configuring Support for IUA with SCTP for Cisco Access Servers Feature, page 236](#)
- [Troubleshooting Tips, page 247](#)

Configuring PRI Backhaul Using the SCTP and the ISDN Q.921 User Adaptation Layer

Configuration Options

The following is an example of an application-server configuration on a gateway:

```
AS as1 10.4.8.69 10.4.9.69 2577
```

Application server as1 is configured to use two local IP addresses and port 2577. IP address values that are set apply to all IP addresses of the application-server process.

An application-server process can be viewed as a local representation of an SCTP association since it specifies a remote endpoint that communicates with an application-server local endpoint. An application-server process is defined for a given application server. For example, the following configuration defines remote signaling controller asp1 at two IP addresses for application server as1. The remote SCTP port number is 2577:

```
AS as1 10.4.8.69 10.4.9.69 2477
ASP asp1 AS as1 10.4.8.68 10.4.9.68 2577
```

Multiple application-server processes can be defined for a single application server for the purpose of redundancy, but only one process can be active. The other process is inactive and becomes active at failover.

In the Cisco media-gateway-controller solution, a signaling controller is always the client that initiates association with a gateway. During initiation, you can request outbound and inbound stream numbers, but the gateway allows only a number that is at least one digit higher than the number of interfaces (T1/E1) allowed for the platform.

The number of streams to assign to a given association is implementation dependent. During initialization of the IUA association, you need to specify the total number of streams that can be used. Each D channel is associated with a specific stream within the association. With multiple-trunk-group support, every interface can potentially be a separate D channel.

At startup, the IUA code checks for all the possible T1, E1, or T3 interfaces and sets the total number of inbound and outbound streams supported accordingly. In most cases, there is only a need for one association between the GW and the media gateway controller. For the rare case that you are configuring multiple application server associations to various media gateway controllers, the overhead from the unused streams would have minimal impact. The NFAS D channels are configured for one or more interfaces, where each interface is assigned a unique stream ID.

The total number of streams for the association needs to include an additional stream for the SCTP management messages. So during startup the IUA code adds one to the total number of interfaces (streams) found.

You can manually configure the number of streams per association. In the backhaul scenario, if the number of D-channel links is limited to one, allowing the number of streams to be configurable avoids the unnecessary allocation of streams in an association that will never be used. For multiple associations between a GW and multiple media gateway controllers, the configuration utility is useful in providing only the necessary number of streams per association. Overhead from the streams allocated but not used in the association is negligible.

If you manually configure the number of streams through the CLI, the IUA code cannot distinguish between a startup event, which automatically sets the streams to the number of interfaces, or if the value is set manually during runtime. If you configure the number of SCTP streams manually, you must add one plus the number of interfaces using the **sctp-streams** keyword. Otherwise, IUA needs always to add one for the management stream, and the total number of streams increments by one after every reload.

When you set the SCTP stream with the command-line interface, you cannot change the inbound and outbound stream support once the association is established with SCTP. The value takes effect when you first remove the IUA application server configuration and then configure it back as the same application server or a new one. The other option is to reload the router.

To configure the PRI Backhaul Using SCTP and the ISDN Q.921 User Adaptation Layer feature, perform the following tasks:

- [Configure IUA, page 230](#)
- [Configure ISDN Signaling \(PRI\) Backhaul, page 232](#)
- [Verify PRI Backhaul, page 234](#)



Caution

When the Fast Ethernet interface is configured for auto negotiation, it can take up to two seconds to initialize. Two examples of the interface initializing is when the **no shutdown** command is entered, or if the cable is removed and then plugged back in. To avoid any problems, the Fast Ethernet interface should not be configured for auto negotiation. The duplex and speed parameters should be set according to the requirements of the network, and should not be set to auto.

Configure IUA



Note

To configure IUA, perform the following steps.

The steps below direct you to configure an application server and the ASP first to allow an NI2+ to be bound to the IUA transport layer protocol. The application server is a logical representation of the SCTP local endpoint. The local endpoint can have more than one IP address but must use the same port number.

Prerequisites

- Ensure that Cisco IOS Release 12.2(15)T or later is installed and running on your system.
- Configure ISDN to backhaul Q.921 signaling to the third-party call agent (MGC).
- Ensure that your Cisco AS5850 has the following:
 - MGCP 1.0
 - IUA 0.4
 - ISDN network side support to terminate multiple voice PRIs

SUMMARY STEPS

1. enable
2. configure terminal
3. iua
4. as
5. asp as
6. asp sctp-keepalives
7. asp ip-precedence
8. as fail-over-timer
9. exit

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---|--|
| Step 1 | enable Example: Router> enable | Enters privileged EXEC mode. Enter your password if prompted. |
| Step 2 | configure terminal Example: Router# configure terminal | Enters global configuration mode. |
| Step 3 | iua Example: Router(config)# iua | Enters IUA configuration mode and specifies backhaul using SCTP. |
| Step 4 | as as-name {local-ip1 [local-ip2]} [local-sctp-port] Example: Router(config-iua)# as as5400-3 10.1.2.34 10.1.2.35 2577 | Defines an application server on a gateway. You can specify up to three local IP addresses (note that SCTP has built-in support for multihomed machines). |
| Step 5 | asp asp-name as as-name {remote-ip1 [remote-ip2]} [remote-sctp-port] Example: Router(config-iua)# asp asp1 as as5400-3 10.4.8.68 10.4.9.68 2577 | Defines an ASP. Use this command to establish SCTP associations. Note A maximum of three ASPs can be configured per application server. |
| Step 6 | asp asp-name sctp-keepalives remote-ip keepalive-value Example: Router(config-iua)# asp asp1 sctp-keepalives 10.1.2.234 600 | (Optional) Sets SCTP keepalive behavior, in ms, for the specified ASP and IP address. Range: 1000 to 60000. Default: 500. Note Find the current value by examining the show ip sctp association parameters command output under heartbeats. |

| | Command or Action | Purpose |
|--------|--|---|
| Step 7 | <pre>asp <i>asp-name</i> ip-precedence <i>remote-ip</i> <i>ip-precedence-level</i></pre> <p>Example: Router(config-iaa)# asp asp1 ip-precedence 10.1.2.345 7</p> | <p>(Optional) Sets the IP precedence level for protocol data units (PDUs) for the specified IP address.</p> <p>Range for a given address is 0 to 7. Default for normal IP precedence handling is 0.</p> |
| Step 8 | <pre>as <i>as-name</i> fail-over-timer <i>time</i></pre> <p>Example: Router(config-iaa)# as as5400-3 fail-over-timer 10000</p> | <p>(Optional) Sets the failover timer value, in ms. IUA waits for this amount of time for one ASP to take over from another ASP during failover.</p> <p>Note Find the current failover timer value by examining the show iua as all command output.</p> |
| Step 9 | <pre>exit</pre> <p>Example: Router(config-iaa)# exit</p> | <p>Exits the current mode.</p> |

Configure ISDN Signaling (PRI) Backhaul

To configure ISDN signaling (PRI) backhaul, perform the following steps.

Prerequisites

- Ensure that Cisco IOS Release 12.2(4)T or later is installed and running on your system.

SUMMARY STEPS

- enable
- configuration terminal
- controller
- pri-group timeslots service
- exit
- interface serial
- isdn switch-type
- isdn bind-13 IUA-backhaul as
- Repeat as needed.
- exit

| | Command | Purpose |
|----------------|--|---|
| Step 1 | enable Example: Router> enable | Enters privileged EXEC mode. Enter your password if prompted. |
| Step 2 | configure terminal Example: Router# configure terminal | Enters global configuration mode. |
| Step 3 | controller t1 0 Example: Router(config)# controller t1 0 | Enters controller configuration mode for slot 0. |
| Step 4 | pri-group timeslots 1-24 service mgcp Example: Router(config-control)# pri-group timeslots 1-24 service mgcp | Sets the control protocol used for backhaul to MGCP. You cannot share controller timeslots between backhaul and other Layer 3 protocols. |
| Step 5 | exit Example: Router(config-control)# exit | Exits the current mode. |
| Step 6 | interface serial 0:23 Example: Router(config)# interface serial 0:23 | Enters serial-interface configuration mode for the specified controller and timeslot. The D-channel timeslot is (channelized T1): 23 or (channelized E1):15. |
| Step 7 | isdn switch-type switch-type Example: Router(config-if)# isdn switch-type primary-4ess | Specifies the ISDN switch type (can be done in either global configuration mode or interface mode). |
| Step 8 | isdn bind-13 IUA-backhaul as as-name Example: Router(config-if)# isdn bind-13 IUA-backhaul as server1 | Configures ISDN to backhaul Q.931 to the media gateway controller. |
| Step 9 | Repeat the preceding steps for each T1 interface that uses backhaul. | — |
| Step 10 | exit Example: Router(config-if)# exit | Exits the current mode. |

Verify PRI Backhaul

To verify PRI backhaul, perform the following steps (listed alphabetically).

SUMMARY STEPS

1. **show iua as**
2. **show iua asp**
3. **show isdn status**
4. **show running-config**

DETAILED STEPS

Step 1 **show iua as {all | name as-name}**

Use this command to display the current state of the active application server and show the PRI interfaces configured on the application server.

The following output shows that the current state of the application server (as1) is active and that there are four PRI interfaces configured to use this application server:

```
Router# show iua as all

Name of AS :as1
  Total num of ASPs configured :2
  Current state : ACTIVE
  Active ASP :asp1
  Number of ASPs up :1
  Fail-Over time : 4000 milli seconds
  Local address list : 10.21.0.2
  Local port 9900
  Interface IDs registered with this AS
    Interface ID      stream #
    256 (serial1/0:23)  1
    257 (serial1/1:23)  2
    512 (serial2/0:23)  3
    513 (serial2/1:23)  4
```

Step 2 **show iua asp {all | name asp-name}**

Use this command to display the current state of the active ASP and show information about the SCTP association being used by this ASP.

The following output shows that the current state of the ASP (asp1) is active. It also shows information about the SCTP association being used by this ASP.

```
Router# show iua asp all

Name of ASP :asp1
Current State of ASP:ASP-Active
Current state of underlying SCTP Association IUA_ASSOC_ESTAB , assoc id
0
SCTP Association information :
  Local Receive window :9000
  Remote Receive window :9000
  Primary Dest address requested by IUA 10.23.0.16
  Effective Primary Dest address 10.23.0.16
Remote address list : 10.23.0.16
Remote Port :9900
Statistics :
```

```

Invalid SCTP signals Total :0 Since last 0
SCTP Send failures :0

Name of ASP :asp2
Current State of ASP:ASP-Down
Current state of underlying SCTP Association IUA_ASSOC_INIT , assoc id
0
Remote address list : 10.23.0.16
Remote Port :9911
Statistics :
Invalid SCTP signals Total :0 Since last 0
SCTP Send failures :0

```

Step 3 show isdn status

Use this command to display the status of all ISDN interfaces, including active layers, timer information, and switch-type settings.

Use it also to display the status of ISDN backhaul. If connection to the media gateway controller is lost, the router shuts down Layer 2 so that it cannot receive calls. When the connection is back up, you can use this command to verify that Layer 2 was also brought back up correctly.

The following sample output shows Layer 2 status, as defined by the `MULTIPLE_FRAME_ESTABLISHED` message, to be up. The L3 protocol and state status are highlighted:

```

Router# show isdn status

Global ISDN Switchtype = primary-5ess
ISDN Serial1/0:23 interface
  dsl 0, interface ISDN Switchtype = primary-5ess
  L2 Protocol = Q.921 L3 Protocol(s) = IUA BACKHAUL
  Layer 1 Status:
    ACTIVE
  Layer 2 Status:
    TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
  Layer 3 Status:
    0 Active Layer 3 Call(s)
  Active dsl 0 CCBs = 0
  The Free Channel Mask: 0x807FFFFF
ISDN Serial1/1:23 interface
  dsl 1, interface ISDN Switchtype = primary-5ess
  L2 Protocol = Q.921 L3 Protocol(s) = IUA BACKHAUL
  Layer 1 Status:
    ACTIVE
  Layer 2 Status:
    TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
  Layer 3 Status:
    0 Active Layer 3 Call(s)
  Active dsl 1 CCBs = 0
  The Free Channel Mask: 0x807FFFFF
ISDN Serial2/0:23 interface
  dsl 2, interface ISDN Switchtype = primary-5ess
  L2 Protocol = Q.921 L3 Protocol(s) = IUA BACKHAUL
  Layer 1 Status:
    ACTIVE
  Layer 2 Status:
    TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
  Layer 3 Status:
    0 Active Layer 3 Call(s)
  Active dsl 2 CCBs = 0
  The Free Channel Mask: 0x807FFFFF
ISDN Serial2/1:23 interface
  dsl 3, interface ISDN Switchtype = primary-5ess

```

```

L2 Protocol = Q.921 L3 Protocol(s) = IUA BACKHAUL
Layer 1 Status:
    ACTIVE
Layer 2 Status:
    TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
Layer 3 Status:
    0 Active Layer 3 Call(s)
Active dsl 3 CCBs = 0
The Free Channel Mask: 0x807FFFFFFF
Total Allocated ISDN CCBs = 0

```

Step 4 show running-config

Use this command to display basic router configuration.

**Note**

For troubleshooting tips, see the [“Troubleshooting Tips”](#) section on page 247.

Configuring Support for IUA with SCTP for Cisco Access Servers Feature

This section contains the following procedures:

- [Configure IUA for Cisco Access Servers, page 236](#)
- [Configure the SCTP T1 Initiation Timer, page 236](#)
- [Create NFAS Groups and Bind Them to the Application Server, page 239](#)
- [Migrate from RLM to IUA with SCTP, page 241](#)
- [Modify a PRI Group on an MGC, page 242](#)
- [Verify Support for IUA with SCTP, page 243](#)

Configure IUA for Cisco Access Servers

To configure IUA for Cisco access servers, follow the steps for configuring IUA for PRI Q.921 backhaul, as described in the [“Configure IUA”](#) section on page 230.

Configure the SCTP T1 Initiation Timer

To configure the SCTP T1 initiation timer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **iua**
4. **as**
5. **as fail-over-timer**
6. **as sctp-startup-rtx**

7. `as sctp-streams`
8. `as sctp-t1init`
9. `asp as`
10. `asp ip-precedence`
11. `asp as`
12. `asp sctp-keepalive`
13. `asp sctp-max-association`
14. `asp sctp-path-retransmission`
15. `asp sctp-t3-timeout`
16. `exit`

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|--|
| Step 1 | <p><code>enable</code></p> <p>Example: Router> enable</p> | Enters privileged EXEC mode. Enter your password if prompted. |
| Step 2 | <p><code>configure terminal</code></p> <p>Example: Router# configure terminal</p> | Enters global configuration mode. |
| Step 3 | <p><code>iua</code></p> <p>Example: Router(config)# iua</p> | Enters IUA configuration mode and specifies backhaul using SCTP. |
| Step 4 | <p><code>as as-name {localip1 [localip2]} [local-sctp-port]</code></p> <p>Example: Router(config-iua)# as as5400-3 10.1.2.34 10.1.2.35 2577</p> | Defines an application server on a gateway. |
| Step 5 | <p><code>as as-name fail-over-timer time</code></p> <p>Example: Router(config-iua)# as as5400-3 fail-over 10000</p> | <p>(Optional) Sets the failover timer value, in ms.</p> <p>Note Find the failover timer value by examining the <code>show iua as all</code> command output.</p> |
| Step 6 | <p><code>as as-name sctp-startup-rtx number</code></p> <p>Example: Router(config-iua)# as as5400-3 sctp-startup-rtx 8</p> | Configures the SCTP startup retransmission interval. |

| | Command or Action | Purpose |
|---------|--|---|
| Step 7 | <p>as <i>as-name</i> sctp-streams <i>number</i></p> <p>Example: Router(config-iaa)# as as5400-3 sctp-streams 56</p> | <p>Configures the number of SCTP streams for this application server.</p> <p>Although the gateway help function displays a range of 2 to 57, the upper end of the range (also the default) is determined by your hardware, and is equal to the number of controllers on that gateway and NAS one plus. If you enter a number higher than that, the system assumes the default.</p> <p>Note If you want to set this value to something other than the default, add one to the number of D channel interfaces that you want to use concurrently.</p> |
| Step 8 | <p>as <i>as-name</i> sctp-tlinit <i>number</i></p> <p>Example: Router(config-iaa)# as as1 sctp-tlinit 1000</p> | <p>Sets the SCTP T1 initiation timer, in ms.</p> |
| Step 9 | <p>asp <i>asp-name</i> as <i>as-name</i> <i>ip-address</i></p> <p>Example: Router(config-iaa)# asp asp1 as as1 10.4.8.68 10.4.9.68</p> | <p>Creates an ASP and specifies to which application server it belongs.</p> |
| Step 10 | <p>asp <i>asp-name</i> ip-precedence <i>remote-ip-address</i> <i>number</i></p> <p>Example: Router(config-iaa)# asp asp1 ip-precedence 10.1.2.345 7</p> | <p>Specifies the IP precedence level for protocol data units (PDUs) for a given IP address.</p> <p>Default for normal IP precedence handling is 0.</p> |
| Step 11 | <p>asp <i>asp-name</i> as <i>as-name</i> (<i>remote-ip</i> [<i>remote-ip2</i>]) [remote-sctp-port]</p> <p>Example: Router(config-iaa)# asp asp1 as as5400-3 10.4.8.68 10.4.9.68 2577</p> | <p>Defines an ASP. Use this command to establish SCTP associations.</p> |
| Step 12 | <p>asp <i>asp-name</i> sctp-keepalive <i>remote-ip-address</i> <i>number</i></p> <p>Example: Router(config-iaa)# asp asp1 sctp-keepalive 10.1.2.234 1000</p> | <p>(Optional) Specifies the IP address to enable and disable keepalives and control SCTP keepalives on destination IP addresses.</p> |
| Step 13 | <p>asp <i>asp-name</i> sctp-max-association <i>ip-address</i> <i>number</i></p> <p>Example: Router(config-iaa)# asp asp1 sctp-max-association 10.10.10.10 20</p> | <p>Sets the maximum association retransmissions for this ASP.</p> |

| | Command or Action | Purpose |
|---------|--|--|
| Step 14 | <pre>asp asp-name sctp-path-retransmission ip-address number</pre> <p>Example: Router(config-iaa)# asp aspl sctp-path-retransmission 10.10.10.10 2 </p> | Sets the SCTP path retransmissions for this ASP. |
| Step 15 | <pre>asp asp-name sctp-t3-timeout ip-address number</pre> <p>Example: Router(config-iaa)# asp aspl sctp-t3-timeout 10.10.10.10 60000 </p> | Enters IUA-SCTP configuration mode and sets the SCTP T3 retransmission timeout for this ASP. |
| Step 16 | <pre>exit</pre> <p>Example: Router(config-iaa-sctp)# exit </p> | Exits the current mode. |

Create NFAS Groups and Bind Them to the Application Server

To create NFAS groups and bind them to the application server, perform the following steps.



Note

- This procedure configures two T1 interfaces into two NFAS groups or trunk groups that are served by the same application server with two different SCTP streams (ASPs). It allows you to configure the NFAS primary D channel and bind the channel to an IUA application server.
- The steps for configuring the T1/E1 interface are the same as the steps using RLM, but multiple NFAS groups can now be defined to support multiple trunk groups. All interfaces in an NFAS are treated as one trunk group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **controller t1 1/0/0**
4. **pri-group timeslots 1-23 nfas-d primary nfas-int 0 nfas-group iua**
5. **exit**
6. **controller t1 1/0/1**
7. **pri-group timeslots 1-23 nfas-d primary nfas-int 0 nfas-group iua**
8. **exit**

| | Command or Action | Purpose |
|--------|---|---|
| Step 1 | enable Example: Router> enable | Enters privileged EXEC mode. Enter your password if prompted. |
| Step 2 | configure terminal Example: Router# configure terminal | Enters global configuration mode. |
| Step 3 | controller t1 1/0/0 Example: Router(config)# controller t1 1/0/0 | Enters controller configuration mode on the first T1 controller. |
| Step 4 | pri-group timeslots 1-23 nfas-d primary nfas-int 0 nfas-group number iua as-name Example: Router(config-controller)# pri-group timeslots 1-23 nfas-d primary nfas-int 0 nfas-group 1 iua as-1 | Configures the NFAS primary D channel on one channelized T1 controller and binds the D channel to an IUA application server. You can choose any timeslot other than 24 to be the virtual container for the D channel parameters for ISDN. Keywords and arguments are as follows: <ul style="list-style-type: none"> • nfas-group number—NFAS group • iua as-name—Must match the name of an application server that was set up during IUA configuration. <p>Note For more information, see the “Configure IUA” section on page 230.</p> |
| Step 5 | exit Example: Router(config-controller)# exit | Exits the current mode on the first controller. |
| Step 6 | controller t1 1/0/1 Example: Router# controller t1 1/0/1 | Enters controller configuration mode on the second T1 controller. |
| Step 7 | pri-group timeslots 1-23 nfas-d primary nfas-int 0 nfas-group number iua as-name Example: Router(config-controller)# pri-group timeslots 1-23 nfas-d primary nfas-int 0 nfas-group 1 iua as-1 | Configures the NFAS primary D channel on another channelized T1 controller and binds the D channel to an IUA application server. Keywords and arguments are as above. The argument <i>as-name</i> must match the name of an application server that was set up during IUA configuration. |
| Step 8 | exit Example: Router(config-if)# exit | Exits the current mode. |

Migrate from RLM to IUA with SCTP

To migrate from RLM to IUA with SCTP, perform the following steps.



Note

The following changes have been made between RLM and IUA with SCTP:

- Application server and ASP configuration lines must precede the controller configuration lines in the configuration text file.
- RLM group configuration must be removed from the D channel configuration.
- For the D channel, the **interface serial** commands are now replaced by **interface D channel** commands.
- Any **isdn bind** commands must be removed from the D channel. Binding of the NFAS groups now takes place when you use the **pri-group** commands for IUA with SCTP.

For more information, see the [“SCTP Migration from RLM to IUA: Example” section on page 273](#).

SUMMARY STEPS

1. **enable**
2. **copy run tftp**
3. Remove the “isdn rlm-group 1” line
4. **copy tftp start**
5. **reload**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|--|
| Step 1 | enable Example: Router> enable | Enters privileged EXEC mode. Enter your password if prompted. |
| Step 2 | copy run tftp Example: Router# copy run tftp | Copies the running configuration to a TFTP server. Make a backup copy of the running configuration. Enter the IP address and destination filename when prompted. Note Make all edits to the configuration text file that you have copied over to your TFTP server. Some TFTP servers might require that the name of the file that you intend to copy over is already existing and has write permissions on the TFTP server onto which you are copying. |

| | Command or Action | Purpose |
|--------|---|--|
| Step 3 | <p>For RLM, remove the “isdn rlm-group 1” line shown in bold.</p> <p>Example:</p> <pre>interface Serial3/0:1:23 no ip address isdn switch-type primary-ni isdn incoming-voice modem isdn T321 30000 isdn T303 20000 isdn T200 2000 isdn rlm-group 1 isdn negotiate-bchan resend-setup isdn bchan-number-order ascending no cdp enable</pre> | Links IUA instead of RLM by removing the “isdn rlm-group 1” line from the interface serial output. |
| Step 4 | <p>copy tftp start</p> <p>Example:</p> <pre>Router# copy tftp start</pre> | Copies the new configuration to the startup configuration. |
| Step 5 | <p>reload</p> <p>Example:</p> <pre>Router# reload</pre> | Reloads the router. |

Modify a PRI Group on an MGC

To modify a PRI group on an MGC, perform the following steps.

Prerequisites

- Remove **isdn bind** commands from the D channel. Binding of the NFAS groups takes place when you use the **pri-group** commands for IUA with SCTP.



Note For more information, see the [“Trunk Group Bound to an Application Server: Example” section on page 274.](#)

SUMMARY STEPS

- enable
- configure terminal
- interface Dchannel3/0:1
- shutdown
- exit

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|---|
| Step 1 | enable Example: Router> enable | Enters privileged EXEC mode. Enter your password if prompted. |
| Step 2 | configure terminal Example: Router# configure terminal | Enters global configuration mode. |
| Step 3 | interface Dchannel3/0:1 Example: Router(config)# interface Dchannel3/0:1 | Enters interface configuration mode for the specified D channel that is to be shut down. This is the format used for IUA. |
| Step 4 | shutdown Example: Router(config-if)# shutdown | Shuts down the D channel. |
| Step 5 | exit Example: Router(config-if)# exit | Exits the current mode. |

Verify Support for IUA with SCTP

To verify support for IUA with SCTP, perform the following steps (listed alphabetically).

SUMMARY STEPS

1. **show ip sctp association list**
2. **show ip sctp association parameters**
3. **show ip sctp association statistics**
4. **show ip sctp errors**
5. **show ip sctp instances**
6. **show ip sctp statistics**
7. **show isdn service**
8. **show isdn status**
9. **show running-config**

DETAILED STEPS

Step 1 **show ip sctp association list**

Use this command to display current SCTP association and instance identifiers, current state of SCTP associations, and 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
```

Step 2 show ip sctp association parameters

Use this command to display parameter values for the specified association. This command requires an association identifier as an argument. Association identifiers can be obtained from the output of the **show ip sctp association list** command.

Many parameters are defined for each association, some of them configured and some of them calculated. They fall into the following main groupings:

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

```
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

```

Step 3 **show ip sctp association statistics**

Use this command to display statistics about the specified association, including the following: 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.



Note This command requires an association identifier argument, which you can obtain from output of the **show ip sctp association list** command.

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

```

Step 4 **show ip sctp errors**

Use this command to display errors logged since last time that the statistics were cleared.

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

```

Step 5 show ip sctp instances

Use this command to display 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

```

Step 6 show ip sctp statistics

Use this command to display the overall Sctp statistics accumulated since the last **clear ip sctp statistics** command for currently established associations and those that have terminated. The command also displays the number of aborts and shutdowns received and the number of times the T1 (initialization) and T2 (shutdown) timers expired.

```

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

```

Step 7 show isdn service

Use this command to display information about ISDN channels and the service states.

Step 8 show isdn status

Use this command to display the status of all ISDN interfaces, including active layers, timer information, and switch-type settings.

Step 9 show running-config

Use this command to display the basic router configuration.

Troubleshooting Tips

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

**Caution**

Use debug commands with extreme caution or not at all in live systems, depending on the amount of traffic. Debug commands other than those for performance, state, signal, and warnings can generate a great deal of output and therefore cause associations to fail. Use these commands only in test environments or during times of very low traffic volume.

**Note**

- SCTP debug commands display information for all current SCTP associations and cannot be limited to particular associations.
- SCTP debug 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.
- Sample outputs for the debug commands are shown in the [“Examples” section on page 249](#).

- You can use debugs with timestamps enabled to see the relevant timing of the events indicated. To add timestamps to debug output, use the **service timestamps debug** and **service timestamps log**, optionally with the **msec** keyword. Output is in the format MMM DD HH:MM:SS, which indicates the date and time according to the system clock. If the system clock is not set, the date and time are preceded by an asterisk (*) to indicate that the date and time are probably not correct.
 - For more information on SCTP debug commands, see [Stream Control Transmission Protocol \(SCTP\)](#).
-
- Use the **debug ip sctp api** command to show all SCTP calls to the application programming interface (API) that are being executed and the parameters associated with these calls.
 - Use the **debug ip sctp congestion** command to display 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 ULP is receiving datagrams from SCTP and thus freeing local rwnd space.
 - Use the **debug ip sctp init** command to display 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. You can use this command to see the chunks associated with any initialization sequence, but it 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.
 - Use the **debug ip sctp multihome** command to display the source and destination of datagrams in order to monitor 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 they can be sent to the secondary addresses.
 - Use the **debug ip sctp performance** command to display the average number of chunks and datagrams being sent and received per second once every 10 seconds. 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.
 - Use the **debug ip sctp rcvchunks** command to display information about chunks that are received, including the following: 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 a datagram that is already being reassembled. Command output shows whether the datagram is complete after receiving this chunk or not and, if 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.
 - Use the **debug ip sctp rto** command to display adjustments to the retransmission (retrans) timeout value due to retransmission of data chunks or unacknowledged heartbeats.
 - Use the **debug ip sctp segments** command to display every datagram that is sent or received and the chunks that are contained in each. The command has two forms: simple and verbose. This simple form of the command shows basic information for each chunk type.

- Use the **debug ip sctp segmentv** command to show every datagram that is sent or received and the chunks that are contained in each. The command has two forms: simple and verbose. This verbose form of the output shows detailed information for each chunk type.
- Use the **debug ip sctp signal** command to display signals that are sent from SCTP to the application or ULP. These signals inform the ULP of state transitions for associations or destination addresses. Signal s sent to the ULP when new data is available to be received may not be shown because they occur infrequently. You can use this command to determine whether or not 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.



Note The **debug ip sctp state** and **debug ip sctp signal** commands are often used together to provide insight into the stability of associations.

- Use the **debug ip sctp sndchunks** command to display 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
- Use the **debug ip sctp state** command with the **debug ip sctp signal** command to provide insight into the stability of associations.
- Use the **debug ip sctp timer** command to display information about all started, stopped, and triggering SCTP timers. Many SCTP timers, after they are started, are not restarted until they expire or are stopped; the first call starts the timer, and subsequent calls do nothing until the timer either expires or is stopped.
- Use the **debug ip sctp warnings** command to display information on any unusual situation that is encountered. These situations may or may not indicate problems, depending on the particulars of the situation.
- Use the **debug iua as** command to display debug messages for the IUA application server when an ISDN backhaul connection is initially established.
- Use the **debug iua asp** command to display debug messages for the IUA ASP when an ISDN backhaul connection is initially established.

Examples

This section contains the following output examples (commands are listed alphabetically):

- [Sample Output for the debug ip sctp api Command, page 250](#)
- [Sample Output for the debug ip sctp congestion Command, page 250](#)
- [Sample Output for the debug ip sctp init Command, page 251](#)
- [Sample Output for the debug ip sctp multihome Command, page 252](#)
- [Sample Output for the debug ip sctp performance Command, page 253](#)
- [Sample Output for the debug ip sctp rcvchunks Command, page 253](#)

- [Sample Output for the debug ip sctp rto Command, page 254](#)
- [Sample Output for the debug ip sctp segments Command, page 255](#)
- [Sample Output for the debug ip sctp segmentv Command, page 256](#)
- [Sample Output for the debug ip sctp signal Command and the debug ip sctp state Command, page 257](#)
- [Sample Output for the debug ip sctp sndchunks Command, page 258](#)
- [Sample Output for the debug ip sctp timer Command, page 259](#)
- [Sample Output for the debug ip sctp warnings Command, page 260](#)
- [Sample Output for the debug iua Command, page 260](#)

Sample Output for the debug ip sctp api Command



Caution

Do not use this command in a live system that has any significant amount of traffic running. It can generate significant traffic, and 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:          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
*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:          unorder 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
```

Sample Output for the debug ip sctp congestion Command

```
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

```

Sample Output for the debug ip sctp init Command

```
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

```

Sample Output for the debug ip sctp multihome Command



Caution

This command generates one debug line for each datagram sent or received. Use 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

```

Sample Output for the debug ip sctp performance Command

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
```

Sample Output for the debug ip sctp rcvchunks Command



Caution

This command generates multiple debug lines for each chunk received. Use 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, 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
```

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, numFrag=1
SCTP: Assoc 0: Rcvd duplicate chunk: 0/5/50/15755E5A
SCTP: Assoc 0: Add Sack Chunk, CumTSN=15755E58, numFrag=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, numFrag=2
SCTP: Assoc 0: Rcvd duplicate chunk: 0/7/50/15755E5C
SCTP: Assoc 0: Add Sack Chunk, CumTSN=15755E58, numFrag=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, numFrag=1
SCTP: Assoc 0: Rcvd duplicate chunk: 0/6/50/15755E5B
SCTP: Assoc 0: Add Sack Chunk, CumTSN=15755E58, numFrag=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, numFrag=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
```

Sample Output for the debug ip sctp rto Command



Caution

This command can generate a great deal of output. Use 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
```

Sample Output for the debug ip sctp segments Command



Caution

This command generates several lines of output for each datagram sent or received. Use 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

```

Sample Output for the debug ip sctp segmentv Command



Caution

This command generates multiple lines of output for each datagram sent and received. Use 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

```

Sample Output for the debug ip sctp signal Command and the debug ip sctp state Command

This example shows signals that are sent from SCTP to the application or ULP. A signal is also sent to the ULP when new data is available to be received, but this signal is not shown in the output below because it occurs infrequently.

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 upper-layer protocols 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

```

Sample Output for the debug ip sctp sndchunks Command



Caution

This command generates significant data if there is any significant amount of traffic flowing. Use 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

```

Sample Output for the debug ip sctp timer Command



Caution

This command generates a significant amount of output. Use with extreme caution in a live network.

```
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

```

Sample Output for the debug ip sctp warnings Command

```
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...
```

Sample Output for the debug iua Command

The following example shows that state debugging is turned on for all application servers and that the application server is active:

```
Router# debug iua as state all

IUA :state debug turned ON for ALL AS

00:11:52:IUA:AS as1 number of ASPs up is 1
00:11:57:IUA:AS as1 xsition AS-Up --> AS-Active, cause - ASP aspl
```

The following example shows that peer message debugging is turned on for all digital signal processors (DSPs) and that the ASP is active:

```
Router# debug iua asp peer-msg all

IUA :peer message debug turned ON for ALL ASPs

Router#
00:04:58:IUA :recieved ASP_UP message on ASP aspl
00:04:58:IUA:ASP aspl xsition ASP-Down --> ASP-Up , cause - rcv peer
msg
ASP-UP
00:04:58:IUA:sending ACK of type 0x304 to asp aspl
00:05:03:IUA:rcv ASP_ACTIVE message for ASP aspl
00:05:03:IUA:ASP aspl xsition ASP-Up --> ASP-Active, cause - rcv peer
msg
ASP-Active
```

Configuration Examples for SCTP Options

- [Application-Server and Application-Server-Process: Example, page 261](#)
- [Application-Server and Application-Server-Process with IUA: Example, page 262](#)
- [ISDN Signaling Backhaul: Example, page 265](#)
- [IUA Configuration: Example, page 265](#)
- [PRI Group on an MGC: Example, page 272](#)
- [SCTP Configuration: Example, page 273](#)

- [SCTP Migration from RLM to IUA: Example, page 273](#)
- [Trunk Group Bound to an Application Server: Example, page 274](#)

Application-Server and Application-Server-Process: Example

The following shows sample SCTP configuration options using the help menu for the **as** and **asp** commands:

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# iua

Router(config-iua)# as as1 ?

A.B.C.D          Specify (up to two) Local IP address
Fail-Over-Timer  Configure the Fail-Over timer for this AS
sctp-startup-rtx Configure the SCTP max startup retransmission timer
sctp-streams     Configure the number of SCTP streams for this AS
sctp-tlinit      Configure the SCTP T1 init timer

Router(config-iua)# as as1 sctp-startup-rtx ?

<2-20> Set SCTP Maximum Startup Retransmission Interval

Router(config-iua)# as as1 sctp-streams ?

<1-56> Specify number of SCTP streams for association

Router(config-iua)# as as1 sctp-tlinit ?

<1000-60000> Set SCTP T1 init timer (in milliseconds)

Router(config-iua)# asp asp1 as as1 ?

A.B.C.D Specify (up to two) IP addresses of the call-agent

Router(config-iua)# asp asp1 ?

AS              Specify which AS this ASP belongs to
IP-Precedence   Set IP precedence bits for a IP address in this ASP
sctp-keepalives Modify the keep-alive behaviour of an IP address in this
                ASP
sctp-max-assoc  Set SCTP max association retransmissions for this ASP
sctp-path-retran Set SCTP path retransmissions for this ASP
sctp-t3-timeout Set SCTP T3 retransmission timeout for this ASP

Router(config-iua)# asp asp1 sctp-keep ?

A.B.C.D specify the IP address to enable/disable keep alives

Router(config-iua)# asp asp1 sctp-keepalive 10.10.10.10 ?

<1000-60000> specify keep alive interval (in milliseconds)

Router(config-iua)# asp asp1 sctp-max-assoc ?

A.B.C.D specify the IP address

Router(config-iua)# asp asp1 sctp-max-assoc 10.10.10.10 ?
```

```

<2-20>  specify maximum associations
default use default value of max associations for this address

Router(config-iaa)# asp asp1 sctp-path-retran ?

A.B.C.D  specify the IP address

Router(config-iaa)# asp asp1 sctp-path-retran 10.10.10.10 ?

<2-10>  specify maximum path retransmissions
default use default value of max path retrans for this address

Router(config-iaa)# asp asp1 sctp-t3-timeout ?

A.B.C.D  specify the IP address

Router(config-iaa)# asp asp1 sctp-t3-timeout 10.10.10.10 ?

<300-60000>  specify T3 retransmission timeout (in milliseconds)
default      use default value of T3 for this address

```

Application-Server and Application-Server-Process with IUA: Example

The following example shows a running application-server configuration with IUA configured with one application server (as1) and two application-server processes (asp1 and asp2). Four T1s (T1 1/0, 1/1, 2/0, 2/1) are configured to use IUA backhaul.

```

Router# show running-config

Building configuration...

Current configuration :2868 bytes
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname iua_3660_b
!
logging rate-limit console 10 except errors
!
memory-size iomem 30
voice-card 1
!
voice-card 2
!
voice-card 3
!
voice-card 4
!
voice-card 5
!
voice-card 6
!
ip subnet-zero
!
no ip domain-lookup
!

```

```
no ip dhcp-client network-discovery
iua
  AS as1 10.21.0.2 9900
  ASP asp1 AS as1 10.23.0.16 9900
  ASP asp2 AS as1 10.23.0.16 9911
isdn switch-type primary-5ess
!
fax interface-type modem
mta receive maximum-recipients 0
!
controller T1 1/0
  framing esf
  clock source line primary
  linecode b8zs
  pri-group timeslots 1-24 service mgcp
!
controller T1 1/1
  framing esf
  linecode b8zs
  pri-group timeslots 1-24 service mgcp
!
controller T1 2/0
  framing esf
  linecode b8zs
  pri-group timeslots 1-24 service mgcp
!
controller T1 2/1
  framing esf
  linecode b8zs
  pri-group timeslots 1-24 service mgcp
!
controller T1 3/0
  framing sf
  linecode ami
!
controller T1 3/1
  framing sf
  linecode ami
!
controller T1 4/0
  framing sf
  linecode ami
!
controller T1 4/1
  framing sf
  linecode ami
!
controller T1 5/0
  framing sf
  linecode ami
!
controller T1 5/1
  framing sf
  linecode ami
!
controller T1 6/0
  framing sf
  linecode ami
!
controller T1 6/1
  framing sf
  linecode ami
!
interface FastEthernet0/0
```

```

ip address 10.21.0.3 255.255.0.0 secondary
ip address 10.21.0.2 255.255.0.0
speed 10
half-duplex
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface Serial1/0:23
no ip address
ip mroute-cache
no logging event link-status
isdn switch-type primary-5ess
isdn incoming-voice voice
isdn bind-13 iua-backhaul as1
no cdp enable
!
interface Serial1/1:23
no ip address
ip mroute-cache
no logging event link-status
isdn switch-type primary-5ess
isdn incoming-voice voice
isdn guard-timer 3000
isdn T203 10000
isdn bind-13 iua-backhaul as1
no cdp enable
!
interface Serial2/0:23
no ip address
ip mroute-cache
no logging event link-status
isdn switch-type primary-5ess
isdn incoming-voice voice
isdn guard-timer 3000
isdn T203 10000
isdn bind-13 iua-backhaul as1
no cdp enable
!
interface Serial2/1:23
no ip address
ip mroute-cache
no logging event link-status
isdn switch-type primary-5ess
isdn incoming-voice voice
isdn T203 10000
isdn bind-13 iua-backhaul as1
no cdp enable
!
ip classless
ip route 10.0.0.0 255.0.0.0 10.21.0.17
ip route 11.0.0.10 255.255.255.255 FastEthernet0/0
ip route 172.0.0.0 255.0.0.0 172.18.194.1
ip http server
!
snmp-server manager
!
call rsvp-sync
!
voice-port 1/0:23
!

```

```

voice-port 1/1:23
!
voice-port 2/0:23
!
voice-port 2/1:23
!
no mgcp timer receive-rtcp
!
mgcp profile default
!
dial-peer cor custom
!
line con 0
  transport input none

line aux 0
line vty 0 4
  login
!
end

```

ISDN Signaling Backhaul: Example

The following sample output shows that Layers 1, 2, and 3 are enabled and active. Layer 3 shows the number of active ISDN calls.

Notice that the Layer 2 protocol is Q.921 and the Layer 3 protocol is BACKHAUL. This verifies that the system is configured to backhaul ISDN.

If you are connected to a live line, you should see that Layer 1 is active and Layer 2 is MULTIPLE_FRAME_ESTABLISHED, meaning that the ISDN line is up and active.

Router# **show isdn status**

```

*00:03:34.423 UTC Sat Jan 1 2000
Global ISDN Switchtype = primary-net5
ISDN Serial1:23 interface
  dsl 0, interface ISDN Switchtype = primary-net5
  L2 Protocol = Q.921 L3 Protocol(s) = BACKHAUL
Layer 1 Status:
  ACTIVE
Layer 2 Status:
  TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
Layer 3 Status:
  NLCB:callid=0x0, callref=0x0, state=31, ces=0 event=0x0
  NLCB:callid=0x0, callref=0x0, state=0, ces=1 event=0x0
  0 Active Layer 3 Call(s)
Activated dsl 0 CCBs = 0
Number of active calls = 0
Number of available B-channels = 23
Total Allocated ISDN CCBs = 0

```

IUA Configuration: Example

The following is an example of an application-server configuration on a gateway:

```
as as5400-3 10.4.8.69 10.4.9.69 2577
```

In the configuration above, an application server named as-named as5400-3 is configured to use two local IP addresses and a port number of 2577. IP address values that are set apply to all IP addresses of the ASP.

The following configuration example defines a remote signaling controller *asp1* at two IP addresses for the application server named as5400-3. The remote SCTP port number is 2577:

```
Router(config-iaa)# as as5400-3 10.4.8.69 10.4.9.69 2477
```

```
Router(config-iaa)# asp asp1 as as5400-3 10.4.8.68 10.4.9.68 2577
```

Multiple ASPs can be defined for a single application server for the purpose of redundancy, but only one ASP can be active. The other ASP is inactive and only becomes active after fail-over.

In the Cisco MGC solution, a signaling controller is always the client that initiates the association with a gateway. During the initiation phase, you can request outbound and inbound stream numbers, but the gateway only allows a number that is at least one digit higher than the number of interfaces (T1/E1) allowed for the platform.

The number of streams to assign to a given association is implementation dependent. During the initialization of the IUA association, you need to specify the total number of streams that can be used. Each D channel is associated with a specific stream within the association. With multiple trunk group support, every interface can potentially be a separate D channel.

At startup, the IUA code checks for all the possible T1, E1, or T3 interfaces and sets the total number of inbound and outbound streams supported accordingly. In most cases, there is only a need for one association between the GW and the MGC. For the rare case that you are configuring multiple application-server associations to various MGCs, the overhead from the unused streams would have minimal impact. The NFAS D channels are configured for one or more interfaces, where each interface is assigned a unique stream ID.

The total number of streams for the association needs to include an additional stream for the SCTP management messages. So during startup the IUA code adds one to the total number of interfaces (streams) found.

You have the option to manually configure the number of streams per association. In the backhaul scenario, if the number of D channel links is limited to one, allowing the number of streams to be configurable avoids the unnecessary allocation of streams in an association that will never be used. For multiple associations between a GW and multiple MGCs, the configuration utility is useful in providing only the necessary number of streams per association. The overhead from the streams allocated but not used in the association is negligible.

If the number of streams is manually configured through the CLI, the IUA code cannot distinguish between a startup event, which automatically sets the streams to the number of interfaces, or if the value is set manually during runtime. If you are configuring the number of SCTP streams manually, you must add one plus the number of interfaces using the **sctp-streams** keyword with the **as** command. Otherwise, IUA needs to always add one for the management stream, and the total number of streams increments by one after every reload.

When you set the SCTP stream with the CLI, you cannot change the inbound and outbound stream support once the association is established with SCTP. The value takes effect when you first remove the IUA application-server configuration and then configure it back as the same application server or a new one. The other option is to reload the router.

The following is an example of an application-server configuration on a gateway. The configuration shows that an application server named as5400-3 is configured to use two local IP addresses and a port number of 2577:

```
Router(config-iaa)# as as5400-3 10.1.2.34 10.1.2.35 2577
```

The following example sets the failover time (in milliseconds) between 1 and 10 seconds. Entering a value of 1000 would equal one second. Entering a value of 10000 would equal 10 seconds. In this example, the failover timer has been set to 10 seconds:

```
Router(config-iaa)# as as5400-3 fail-over 10000
```

The following example specifies the number of SCTP streams for this association. In this example, 57 is the maximum number of SCTP streams allowed:

```
Router(config-iaa)# as as5400-3 sctp-streams 57
```

The following example sets the SCTP maximum startup retransmission interval. In this example, 20 is the maximum interval allowed:

```
Router(config-iaa)# as as5400-3 sctp-startup 20
```

The following example sets the SCTP T1 initiation timer in milliseconds. In this example, 60000 is the maximum time allowed:

```
Router(config-iaa)# as as5400-3 sctp-t1init 60000
```

The following example specifies the IP address to enable and disable keepalives:

```
Router(config-iaa)# asp asp1 sctp-keepalive 10.1.2.34
```

The following example specifies the keepalive interval in milliseconds. Valid values range from 1000 to 60000. In this example, the maximum value of 60000 ms is used:

```
Router(config-iaa)# asp asp1 sctp-keepalive 10.10.10.10 60000
```

The following example specifies the IP address for the SCTP maximum association and the maximum association value. Valid values are from 2 to 20. The default is 20, which is the maximum value allowed:

```
Router(config-iaa)# asp asp1 sctp-max-association 10.10.10.10 20
```

The following example specifies the IP address for the SCTP path retransmission and the maximum path retransmission value. Valid values are from 2 to 10. The default is 10, which is the maximum value allowed:

```
Router(config-iaa)# asp asp1 sctp-path-retransmissions 10.10.10.10 10
```

The following examples specifies the IP address for SCTP T3 timeout and specifies the T3 timeout value in milliseconds. Valid timeout values are from 300 to 60000. The default is 60000, which is the maximum timeout value allowed:

```
Router(config-iaa)# asp asp1 sctp-t3-timeout 10.10.10.10 60000
```

The following example configures the following:

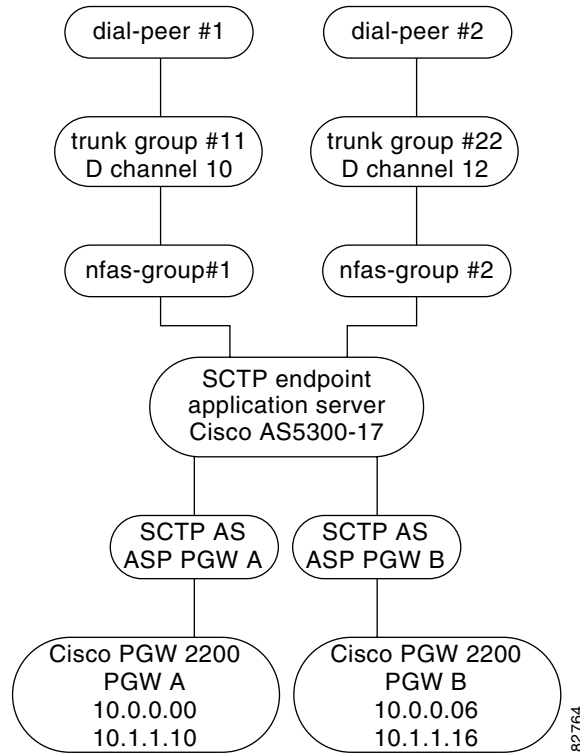
1. Creates an IUA application server (Cisco AS5300-17) that has two local IP addresses (10.0.0.07 and 10.1.1.17) and local port 2097.
2. IUA application server Cisco AS5300-17 is connected by two SCTP associations (ASP PGW A and ASP PGW B) to two hot-standby Cisco PGW 2200s (Cisco PGW 2200 PGW A and Cisco PGW 2200 PGW B). Cisco PGW 2200 PGW A has remote IP addresses 10.0.0.00 and 10.1.1.10, and Cisco PGW 2200 PGW B has remote IP addresses 10.0.0.06 and 10.1.1.16.
3. Two NFAS groups (nfas-group 1 and nfas-group 2), which are both bound to IUA application server as5300-17.
4. Two trunk groups (trunk-group 11 and trunk-group 22)—Trunk-group 11 is bound to interface Dchannel0 and trunk-group 22 is bound to interface Dchannel2.

```
Router(config-iaa)# as as5300-17 10.0.0.07 10.1.1.17 2097
```

```
Router(config-ia)# asp pgwa AS as5300-17 10.0.0.00 10.1.1.10 2097
Router(config-ia)# asp pgwb AS as5300-17 10.0.0.06 10.1.1.16 2097
```

Figure 15 shows the configuration above in diagram form with two outgoing POTS dial-peers (dial-peer 1 and dial-peer 2)—dial-peer 1 points to trunk-group 11, and dial-peer 2 points to trunk-group 22.

Figure 15 Specific ASP Example Configuration



The following is example output from the above configuration:

```
iua
  AS as5300-17 10.0.0.07 10.1.1.17 2097
  ASP pgwa AS as5300-17 10.0.0.00 10.1.1.10 2097
  ASP pgwb AS as5300-17 10.0.0.06 10.1.1.16 2097
!
!
controller E1 0
  framing NO-CRC4
  clock source line primary
  pri-group timeslots 1-31 nfas-d primary nfas-int 0 nfas-group 1 iua as5300-17
!
controller E1 1
  framing NO-CRC4
  clock source line secondary 1
  pri-group timeslots 1-31 nfas-d none nfas-int 1 nfas-group 1
!
controller E1 2
  framing NO-CRC4
  pri-group timeslots 1-31 nfas-d primary nfas-int 0 nfas-group 2 iua as5300-17
!
controller E1 3
  framing NO-CRC4
  pri-group timeslots 1-31 nfas-d none nfas-int 1 nfas-group 2
!
```

```

!
interface Ethernet0
  description the ip is 10.0.0.06 for interface e0
  ip address 10.0.0.06 255.255.255.0
  no ip route-cache
  no ip mroute-cache
!
interface FastEthernet0
  description the primary ip is 10.1.1.16 for interface f0
  ip address 10.1.1.10 255.255.255.0
  no ip route-cache
  no ip mroute-cache
  duplex auto
  speed auto
!
interface Dchannel0
  no ip address
  trunk-group 11
  isdn timer t309 100
  isdn timer t321 30000
  isdn incoming-voice modem
  isdn T303 20000
  isdn negotiate-bchan resend-setup
  no cdp enable
!
interface Dchannel2
  no ip address
  trunk-group 22
  isdn timer t309 100
  isdn timer t321 30000
  isdn incoming-voice modem
  isdn T303 20000
  isdn negotiate-bchan resend-setup
  no cdp enable
!
trunk group 11
!
trunk group 22
!
dial-peer voice 1 pots
  incoming called-number
  destination-pattern 997001
  direct-inward-dial
  trunk-group 11
  forward-digits all
!
dial-peer voice 2 pots
  incoming called-number
  destination-pattern 997002
  direct-inward-dial
  trunk-group 22
  forward-digits all
!

```

The following example shows a running application-server configuration with IUA configured with one application server (as1) and two ASPs (asp1 and asp2). Four T1s (T1 1/0, 1/1, 2/0, 2/1) are configured to use IUA backhaul.

```

Router# show running config

Building configuration...

Current configuration :2868 bytes
!

```

```

version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname iua_3660_b
!
logging rate-limit console 10 except errors
!
memory-size iomem 30
voice-card 1
!
voice-card 2
!
voice-card 3
!
voice-card 4
!
voice-card 5
!
voice-card 6
!
ip subnet-zero
!
no ip domain-lookup
!
no ip dhcp-client network-discovery
iua
    AS as1 10.21.0.2 9900
    ASP asp1 AS as1 10.23.0.16 9900
    ASP asp2 AS as1 10.23.0.16 9911
isdn switch-type primary-5ess
!
fax interface-type modem
mta receive maximum-recipients 0
!
controller T1 1/0
    framing esf
    clock source line primary
    linecode b8zs
    pri-group timeslots 1-24 service mgcp
!
controller T1 1/1
    framing esf
    linecode b8zs
    pri-group timeslots 1-24 service mgcp
!
controller T1 2/0
    framing esf
    linecode b8zs
    pri-group timeslots 1-24 service mgcp
!
controller T1 2/1
    framing esf
    linecode b8zs
    pri-group timeslots 1-24 service mgcp
!
controller T1 3/0
    framing sf
    linecode ami
!
controller T1 3/1
    framing sf

```

```
    linecode ami
!
controller T1 4/0
    framing sf
    linecode ami
!
controller T1 4/1
    framing sf
    linecode ami
!
controller T1 5/0
    framing sf
    linecode ami
!
controller T1 5/1
    framing sf
    linecode ami
!
controller T1 6/0
    framing sf
    linecode ami
!
controller T1 6/1
    framing sf
    linecode ami
!
interface FastEthernet0/0
    ip address 10.21.0.3 255.255.0.0 secondary
    ip address 10.21.0.2 255.255.0.0
    speed 10
    half-duplex
!
interface FastEthernet0/1
    no ip address
    shutdown
    duplex auto
    speed auto
!
interface Serial1/0:23
    no ip address
    ip mroute-cache
    no logging event link-status
    isdn switch-type primary-5ess
    isdn incoming-voice voice
    isdn bind-13 iua-backhaul as1
    no cdp enable
!
interface Serial1/1:23
    no ip address
    ip mroute-cache
    no logging event link-status
    isdn switch-type primary-5ess
    isdn incoming-voice voice
    isdn guard-timer 3000
    isdn T203 10000
    isdn bind-13 iua-backhaul as1
    no cdp enable
!
interface Serial2/0:23
    no ip address
    ip mroute-cache
    no logging event link-status
    isdn switch-type primary-5ess
    isdn incoming-voice voice
```

```

isdn guard-timer 3000
isdn T203 10000
isdn bind-13 iua-backhaul as1
no cdp enable
!
interface Serial2/1:23
no ip address
ip mroute-cache
no logging event link-status
isdn switch-type primary-5ess
isdn incoming-voice voice
isdn T203 10000
isdn bind-13 iua-backhaul as1
no cdp enable
!
ip classless
ip route 10.0.0.0 255.0.0.0 10.21.0.17
ip route 11.0.0.10 255.255.255.255 FastEthernet0/0
ip route 172.0.0.0 255.0.0.0 172.18.194.1
ip http server
!
snmp-server manager
!
call rsvp-sync
!
voice-port 1/0:23
!
voice-port 1/1:23
!
voice-port 2/0:23
!
voice-port 2/1:23
!
no mgcp timer receive-rtcp
!
mgcp profile default
!
dial-peer cor custom
!
line con 0
  transport input none

line aux 0
line vty 0 4
  login
!
end

```

PRI Group on an MGC: Example

To modify a PRI group on a third-party call agent (MGC), the **isdn bind** commands must be removed from the D channel. The binding of the NFAS groups now takes place when you use the **pri-group (pri-slt)** command for IUA with SCTP.

Use the following examples to help you with your configuration:

- Controller configuration for primary span in an NFAS group for RLM. You can choose any time slot other than 24 to be the virtual container for the D channel parameters for ISDN:

```

controller T1 3/0:1
  framing esf
  pri-group timeslots 1-23 nfas-d primary nfas-int 0 nfas-group 1

```

- Controller configuration for primary span in an NFAS group for IUA:

```
controller T1 3/0:1
 framing esf
 pri-group timeslots 1-23 nfas-d primary nfas-int 0 nfas-group 1 iua as-1
```

SCTP Configuration: Example

You can implicitly configure the number of streams in SCTP by specifying only the serial interfaces that are configured to use IUA. The number of streams is bound to the actual number of interfaces supporting IUA. To support Cisco MGC solutions, you can configure any number of streams for each NFAS D channel, up to the total number of interfaces available in a given GW. For platforms using the PRI backhaul with SCTP and the ISDN Q.921 User Adaptation Layer (UAL), such as the Cisco 3660, you can configure the number of streams to match the number of PRIs that are actually backhauled to the Telcordia session manager.

The following example sets the failover time (in milliseconds) between 1 and 10 seconds. Entering a value of 1000 would equal one second. Entering a value of 10000 would equal 10 seconds. In this example, the failover timer has been set to 10 seconds. The default value is 4000 msec. Once you have set the failover timer to a value, you can return it to its default of 4000 msec by using the **no** form of this command.

```
Router(config-iaa)# as as5400-3 fail-over 10000
```

The following example sets the SCTP maximum startup retransmission interval. Valid values are from 2 to 20:

```
Router(config-iaa)# as as1 sctp-startup-rtx 20
```

The following example specifies the number of SCTP streams for an association. Valid values are from 1 to 56:

```
Router(config-iaa)# as as1 sctp-streams 56
```

The following example sets the SCTP T1 initiation timer in milliseconds. Valid values are from 1000 to 60000:

```
Router(config-iaa)# as as1 sctp-tlinit 60000
```

SCTP Migration from RLM to IUA: Example

The following changes have been made between RLM and IUA with SCTP. Use the examples in this section to help you with your configuration:

- The D channel **interface serial** commands are now replaced by **interface D channel** commands.

For RLM, the following format was used:

```
interface Serial3/0:1:23
```



Note

The :23 in the RLM example above, which typically corresponds with T1 configuration (:15 for E1 configuration), is no longer used.

For IUA, the following format is used:

```
interface Dchannel3/0:1
```

- The RLM group configuration must be removed from the D channel configuration.

For RLM, remove the “**isdn rlm-group 1**” line shown in bold:

```
interface Serial3/0:1:23
  no ip address
  isdn switch-type primary-ni
  isdn incoming-voice modem
  isdn T321 30000
  isdn T303 20000
  isdn T200 2000
  isdn rlm-group 1
  isdn negotiate-bchan resend-setup
  isdn bchan-number-order ascending
  no cdp enable
```

For IUA, use the following format:

```
interface Dchannel3/0:1
  no ip address
  isdn timer t309 100
  isdn timer t321 30000
  isdn incoming-voice modem
  isdn T303 20000
  no isdn send-status-enquiry
  isdn negotiate-bchan resend-setup
  isdn bchan-number-order ascending
  no cdp enable
```

Trunk Group Bound to an Application Server: Example

You can configure the NFAS primary D channel on one channelized T1 controller, and bind the D channel to an IUA application server by using the **pri-group (pri-slt)** command.

This example uses a Cisco AS5400 and applies to T1, which has 24 timeslots and is used mainly in North America and Japan. You can choose any timeslot other than 24 to be the virtual container for the D channel parameters for ISDN.

```
Router(config-controller)# pri-group timeslots 1-23 nfas-d primary nfas-int 0 nfas-group 1
iua as5400-4-1
```

The following example applies to E1, which has 32 timeslots and is used by countries other than North America and Japan. You can choose any timeslot other than 32 to be the virtual container for the D channel parameters for ISDN.

```
Router(config-controller)# pri-group timeslots 1-31 nfas-d primary nfas-int 0 nfas-group 1
iua as5400-4-1
```

Additional References

General ISDN References

- [“ISDN Features Roadmap” on page 1](#)—Describes how to access Cisco Feature Navigator; also lists and describes, by Cisco IOS release, ISDN features for that release
- [“Overview of ISDN Voice Interfaces” on page 3](#)—Describes relevant underlying technology; lists related documents, standards, MIBs, and RFCs; and describes how to obtain technical assistance
- [“Additional References” section on page 64](#)—Lists additional ISDN references

References Mentioned in This Chapter

- Cisco 2600 Series Routers documentation at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis2600/index.htm
- Cisco 3600 Series Routers documentation at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/index.htm
- Cisco 3700 Series Routers documentation at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3700/index.htm
- Cisco AS5300 documentation at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/5300/sw_conf/index.htm
- Cisco AS5400 documentation at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/as5400/index.htm
- Cisco IAD2420 Series IADs documentation at <http://www.cisco.com/univercd/cc/td/doc/product/access/iad/iad2420/index.htm>
- *Cisco IOS Voice, Video, and Fax Command Reference*, Release 12.2 T at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fvfvfax_r/index.htm
- *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2T at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fvfvfax_c/index.htm
- *Cisco Media Gateway Controller Software Release 9 Installation and Configuration Guide* at <http://www.cisco.com/univercd/cc/td/doc/product/access/sc/rel9/swinstl/index.htm>
- *Cisco Media Gateway Controller Software Release 9 Messages Reference Guide* at <http://www.cisco.com/univercd/cc/td/doc/product/access/sc/rel9/errmsg/index.htm>
- *Cisco Media Gateway Controller Software Release 9 MML Command Reference* at <http://www.cisco.com/univercd/cc/td/doc/product/access/sc/rel9/mmlref/index.htm>
- *Cisco Media Gateway Controller Software Release 9 Operations, Maintenance, and Troubleshooting Guide* at <http://www.cisco.com/univercd/cc/td/doc/product/access/sc/rel9/omts/index.htm>
- *Cisco Media Gateway Controller Software Release 9 Provisioning Guide* at <http://www.cisco.com/univercd/cc/td/doc/product/access/sc/rel9/prvgde/index.htm>
- *Integrated Signaling Link Terminal*, Cisco IOS Release 12.2(11)T at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t11/ftintslt.htm>
- *IP Transfer Point (ITP)*, Cisco IOS Release 12.2(2)MB at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122limit/122mb/122mb2/itp20/index.htm>
- *PRI Backhaul Using the Stream Control Transmission Protocol and the ISDN Q.921 User Adaptation Layer* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t4/ft_0546.htm
- *Stream Control Transmission Protocol (SCTP) feature* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/ft_sctp2.htm
- *Stream Control Transmission Protocol (SCTP)*, RFC 2960, at <http://rfc2960.x42.com/>
- *Support for IUA with SCTP* at <http://www.cisco.com/univercd/cc/td/doc/product/access/sc/rel9/mgcfm/941fm/fmiua.htm>

- *Support for IUA with SCTP for Cisco Access Servers* at <http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t15/ftgkrup.htm>
- *Troubleshooting and Fault Management Commands* (chapter in the System Management Commands part of the *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.2) at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_r/ffrprt3/frf013.htm