Stream Control Transmission Protocol (SCTP)

Definition

Stream control transmission protocol (SCTP) is an end-to-end, connection-oriented protocol that transports data in independent sequenced streams. SCTP endpoints support multi-homing; therefore, interface redundancy is built into the protocol. Through selective transmission mechanisms, SCTP resolves errors and buffers the data transmission process.

Overview

SCTP provides applications with enhanced performance, reliability, and control functions. This protocol is essential where detection of connection failure and associated monitoring is mandatory. Furthermore, SCTP could be implemented in network systems and applications that deliver voice/data and support quality real-time services (e.g., streaming video and multimedia).

The Signaling Transport (SIGTRAN) group of the Internet Engineering Task Force (IETF) defines SCTP standards in RFC 2960. The underlying mechanism of SCTP is fairly complex and incorporates a number of validation procedures, path-management practices, and security measures.

This tutorial begins by discussing the evolution of SCTP and continues with an explanation of the fundamental concepts of the protocol. Following the explanation, the discussion elaborates on the unique features that distinguish SCTP from other transport protocols. The tutorial ends by covering the basics of message formats, the data transmission process, and the application programming interface (API) calls.

Topics

1. Introduction

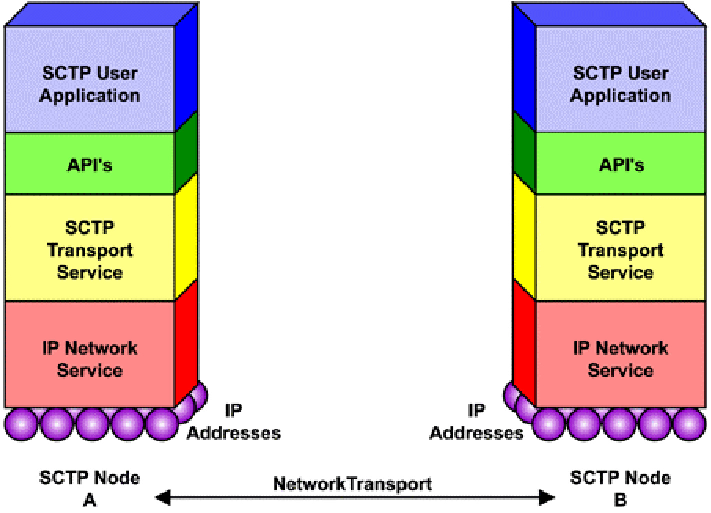
Over the past few decades, packet-switched networks have merged with new technologies to facilitate more efficient communication channels within network systems. The popularity of Internet protocol (IP)–based networks is attributed to the emergence of the Internet and a host of highly popular network applications. IP networks and telephony-based networks are converging to support the inter-working of applications and services. In addition, integrated services digital networks (ISDN), asynchronous transfer mode (ATM) networks, and mobile networks are increasingly routing their signaling traffic via IP networks. SCTP provides flexible delivery and reliable transfer within IP networks.

Large-scale interexchange carriers (IXCs) are realizing that more and more of their network traffic is data rather than voice. Therefore, the scope of present networks must be extended to accommodate application signaling and data services. The primary goal of several IXCs is to carry data and voice using the same transport, thereby reducing additional infrastructure costs.

SCTP provides numerous advantages over user datagram protocol (UDP) and transmission control protocol (TCP). For instance, SCTP combines the datagram orientation of UDP with the sequencing and reliability of TCP. Additionally, SCTP uses multi-stream, message-oriented routing in multi-homed environments.

2. Stream Control Transmission Protocol

Figure 1. SCTP Overview



As illustrated in *Figure 1*, the SCTP transport service layer is positioned between the SCTP user application and the network service being used. Since SCTP is based on interfacing two SCTP endpoints, there are certain application programming interfaces (APIs) that run in between the transport service layer and SCTP user layer. In addition, each endpoint hosts multiple IP addresses.

SCTP deploys multiple paths and streams to transport messages across two endpoints. In SCTP, data is transmitted between endpoints through a connection referred to as an "association." An association begins with an "initiation" and is maintained until all data has been successfully transmitted and received. Once all data is successfully received, the association is gracefully terminated through a "shutdown."

Within SCTP, user data and control information are assembled in chunks. Multiple chunks and a common header comprise a protocol data unit (PDU), also referred to as an "SCTP packet." SCTP packets contain data chunks and control chunks. SCTP provides ordered message delivery within "SCTP streams" and supports network fault tolerance in multi-homing environments.

The fundamental SCTP properties include the following:

- **Validation and Acknowledgement Mechanisms**—During initiation, the validation mechanism bundles the data into a "cookie" that includes a secure hash of values and a secret key. Cookies are digitally signed with message authentication codes (MAC), which are used to prevent denial-of-service attacks. Within an association, selective acknowledgement (SACK) chunks acknowledge the receipt of data chunks. SACK chunks are also used to inform the endpoint of duplicated or missing chunks.

- **Path Selection and Monitoring**—SCTP packets are routed to the destination IP address of a peer endpoint through a "primary path." The primary path allows the user to determine the primary route for data flow. In addition, alternate paths exist for each IP address that the peer endpoint supports. SCTP closely monitors the transmission paths to the peer endpoint using HEARTBEAT chunks that test the connectivity of a path. In SCTP, a path is considered "active" when it has been acknowledged by the peer endpoint or has been used previously for SCTP packet transfer. A path is considered "inactive" if previous path transmissions have failed.

- **Flow and Congestion Control**—While SCTP flow control is based on each association, congestion control is established within each transmission path. The peer endpoint assigns a receiver-window variable for flow control. The receiver-window variable alerts the endpoint of the amount of space available in the peer endpoint's inbound buffer. SCTP deploys congestion control within each stream using a congestion-window variable. This variable limits the number of bytes that may be sent before an acknowledgement is received. A set of flow and congestion control parameters is subtly retained within the association and each transmission path.

These features give SCTP numerous advantages over TCP and UDP, as the following section elaborates.

3. SCTP Advantages: Multi-Homing, Multi-Streaming, and Other Features

SCTP gains advantage over TCP by the virtue of its unique features. This section explores how multi-homing, multi-streaming, and other SCTP features contribute to the SCTP advantage.

SCTP Multi-Homing

The multi-homing feature enables SCTP endpoints to support multiple IP addresses. Multi-homing protects an association from potential network failures by steering traffic to alternate IP addresses. During the initiation of an association, SCTP endpoints exchange lists of IP addresses. Therefore, each endpoint can send and receive messages from any of the IP addresses listed at the remote endpoint. For example, one of the listed IP addresses will be designated as the primary address during the initiation. If the primary address repeatedly drops chunks, however, all chunks will be transmitted to an alternate address until a connection to the primary address can be reestablished.

Multi-homing is a step above conventional single-homed data exchange sessions (i.e., TCP). In single-homed environments, loss of session could be triggered by core network failures or by isolation of endpoints. Since multi-homing directs traffic on different paths to separate IP addresses, loss of session due to physical network failure is virtually non-existent in SCTP.

SCTP Multi-Streaming

The multi-streaming feature separates and transmits user data on multiple SCTP streams. These streams are capable of independent, sequenced delivery. Message loss in a particular stream will only hinder delivery within that stream. Therefore, other streams within an association are not affected.

Through multi-streaming, SCTP eliminates unnecessary blocking that often occurs in TCP transmissions. In TCP, a stream is defined as a sequence of bytes that conform to strict in-sequence delivery. In-sequence delivery results in a major drawback known as "head-of-the-queue blocking," where messages within a stream are not allowed to bypass each other. Since SCTP streams are independent, retransmitted and high-priority messages can bypass less significant messages.

SCTP Features

In the three stages of association, SCTP applies mechanisms that set it apart from TCP and UDP.

- **Initiation Features**—In contrast to the three-way handshake that occurs in TCP, SCTP uses a four-way handshake to initiate an association. This four-way handshake defends against denial-of-service attempts caused by attackers bombarding the SCTP nodes with counterfeit PDUs. In addition, SCTP packets that contain invalid verification tags are identified during initiation and removed from the transmission path. The verification tag values and the cookie mechanism shield the initiation procedure from SYN-type attacks (commonly known as blind attacks) that are commonplace in TCP.

- **Data Transmission Features**—During data transmission, the chunk-bundling feature allows DATA chunks to be multiplexed with control chunks. The peer endpoint acknowledges the receipt of a data chunk by sending a SACK chunk. SACK chunks contain transmission sequence numbers (TSN) that reveal any gaps in the sequence of data chunks. Within each stream, SCTP packets are also assigned stream sequence numbers (SSN). The SSN determines the sequence of data delivery within each independent stream. If the peer endpoint indicates gaps in the SSN, then the message will not be delivered until the gap is filled.

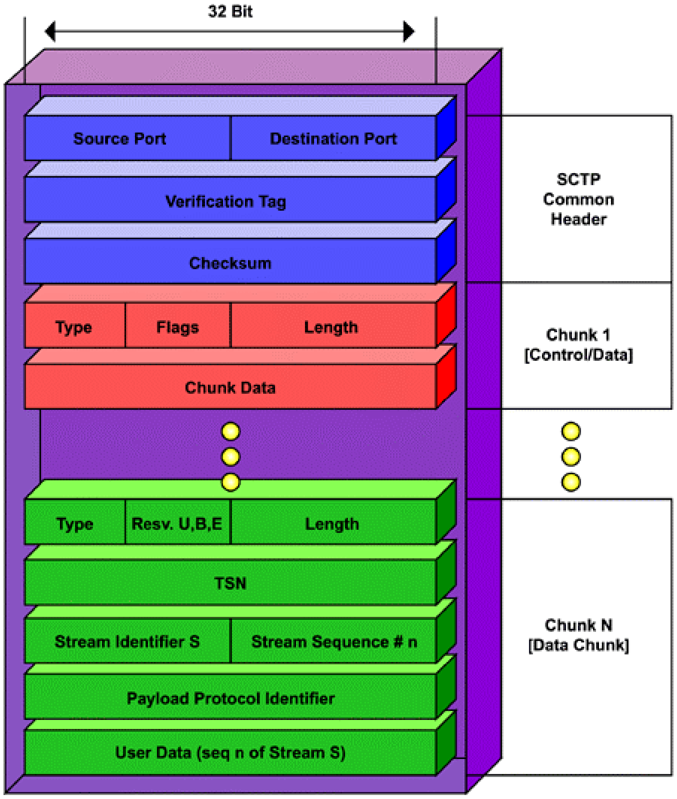
- **Shutdown Features**—The SCTP shutdown procedure has some significant advantages over TCP. For instance, a TCP connection is considered "half-open" when one endpoint continues to send data though the peer endpoint is no longer transmitting data. In contrast, SCTP implements a graceful close of an association by exchanging three messages. These messages acknowledge that both endpoints will cease in their transmissions of data.

In addition to understanding these numerous features, it is also important for one to understand the unique SCTP packet structure.

4. SCTP Packet Format: A Structural Overview

This section discusses the structure of the SCTP packets. A detailed illustration of the SCTP packet is illustrated in *Figure 2*.

Figure 2. SCTP Packet Format



The common header includes the following:

- The source port address
- The destination port address
- The verification tag
- The checksum of the entire packet

The source port number is used by the receiving endpoint to identify the association to which the SCTP packet belongs. The destination port number is the SCTP receiver's address for which the packet is destined. Each endpoint assigns a verification tag (32-bit value) that identifies the association. The checksum acts as a data integrity tool for each SCTP packet.

The chunk fields within a chunk can be described as follows:

- The chunk-type field: identifies the type of chunk being transmitted
- The chunk flag: specifies whether bits will be used in the association
- The chunk length: determines the size of the entire chunk in bytes
- The chunk data: includes the actual data payload of the chunk

As indicated in *Figure 2*, there are N chunks (number of chunks) indicated in a single SCTP packet. The number N is determined by the maximum transmission unit (MTU) size of the transmission path. SCTP allows chunks to be multiplexed in one packet to full MTU capacity, with the exception of initiation (INIT) and initiation-acknowledgement (INIT ACK) chunks. There are 14 types of chunks in all, including one DATA chunk and 13 types of control chunks. The DATA chunk contains the actual data payload. The definition and parameters of the control chunks are summarized in *Table 1*.

Table 1: Control Chunk Types and Their Definitions by Function

CHUNK	DEFINITION
Initiation INIT	The INIT chunk is sent in order to initiate a SCTP association between two endpoints.
Initiation Acknowledgement INIT ACK	INIT ACK chunk acknowledges the receipt of an INIT chunk. The receipt of the INIT ACK chunk establishes an association.
Selective Acknowledgement SACK	SACK chunks acknowledge the receipt of DATA chunks.
Cookie Echo COOKIE ECHO	The COOKIE ECHO chunk is used exclusively during the initiation process and is sent to the peer endpoint.
CookieAcknowledgement COOKIE ACK	The COOKIE ACK chunk acknowledges receipt of the COOKIE ECHO chunk. The COOKIE ACK chunk must take precedence over any DATA chunk or SACK chunk sent in the association. The COOKIE ACK chunk may be bundled with DATA chunks or SACK chunks
Heartbeat Request HEARTBEAT	HEARTBEAT chunks are sent from one SCTP endpoint to its peer in order to test the connectivity of a specific destination address in the association.
Heartbeat Acknowledgement HEARTBEAT ACK	Every time a HEARTBEAT chunk is received by an endpoint, a HEARTBEAT ACK chunk is sent to the source IP address in order to acknowledge receipt of the HEARTBEAT chunk.
Abort Association ABORT	The ABORT chunk is an indication to the peer endpoint to close the association. In addition, the ABORT chunk informs the receiver of the reason for aborting the association in the cause parameters.
Operation Error ERROR	The ERROR chunk is sent to the peer endpoint to report certain error conditions that may exist. The ERROR chunk may contain parameters that determine the type of error that has taken place.
Shutdown Association SHUTDOWN	The SHUTDOWN chunk triggers a graceful close of an association with a peer endpoint.
Shutdown Acknowledgement SHUTDOWN ACK	A SHUTDOWN ACK is used to acknowledge the receipt of the SHUTDOWN chunk at the end of the shutdown process.
Shutdown Complete SHUTDOWN COMPLETE	The SHUTDOWN COMPLETE concludes the shutdown procedure.

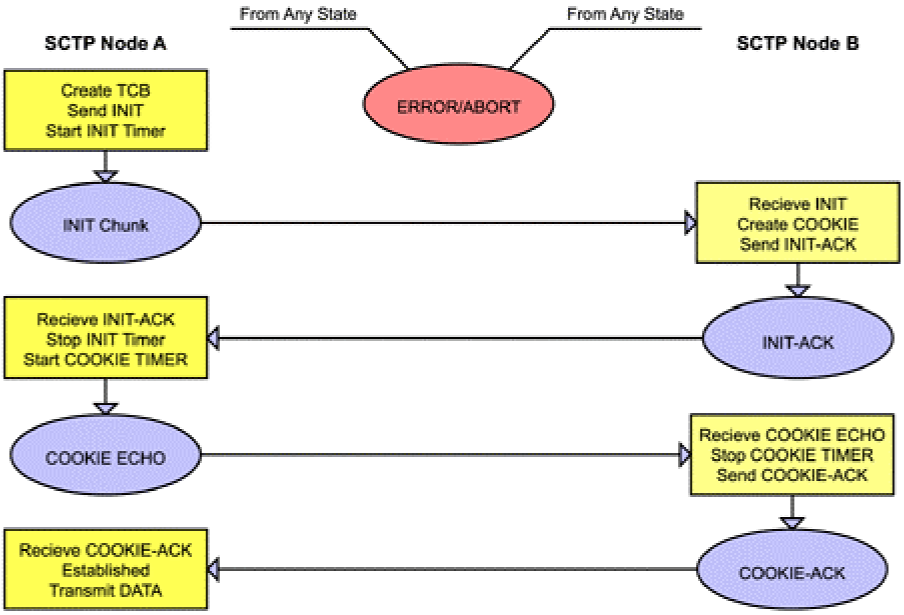
5. SCTP Functions: Tracking the Transmission Process

This section describes the data transport process within the three main phases of an SCTP association: initiation, data transmission and shutdown. The endpoint that initiates the association will be referred to as "Node A"; the peer endpoint that receives the association establishment requests will be referred to as "Node B."

Note: ABORT chunks and ERROR chunks can be generated by either endpoint and can be sent at any time during the association. In these two cases, the endpoints undergo immediate shutdown.

Association Initiation

Figure 3. The Four-Way Handshake in the Initiation of SCTP Association



1. Node A generates an INIT chunk and sends it to Node B. Node A starts the INIT timer.

2. If Node B wishes to accept the association, it generates an INIT ACK chunk that includes a cookie. It then sends the INIT ACK chunk, along with a cookie, back to Node A.

3. Node A receives the INIT ACK chunk and stops INIT timer. Node A generates a COOKIE ECHO chunk, which is then sent to Node B. Node

A starts the cookie timer. DATA chunks may be also be bundled in this packet.

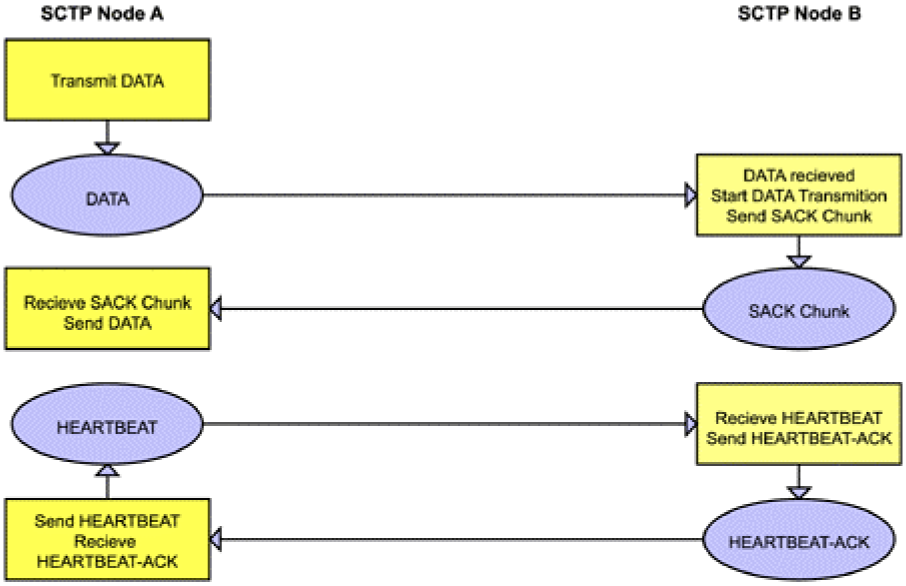
4. Node B checks the validity of the cookie. Following validation it sends a COOKIE ACK back to Node A.

5. Node A receives the COOKIE ACK and enters into the next phase of data transmission.

Association Data Transmission

Node A and Node B seamlessly transmit data. Throughout the data transmission process, HEARTBEAT and HEARTBEAT ACK chunks are exchanged between the nodes at regular-time intervals regulated by the heartbeat timer. These chunks test the connectivity of the endpoints, thereby preserving the validity of the data transmission.

Figure 4. Tracking The Data Transmission Process in SCTP



6. Node A and Node B exchange DATA chunks.

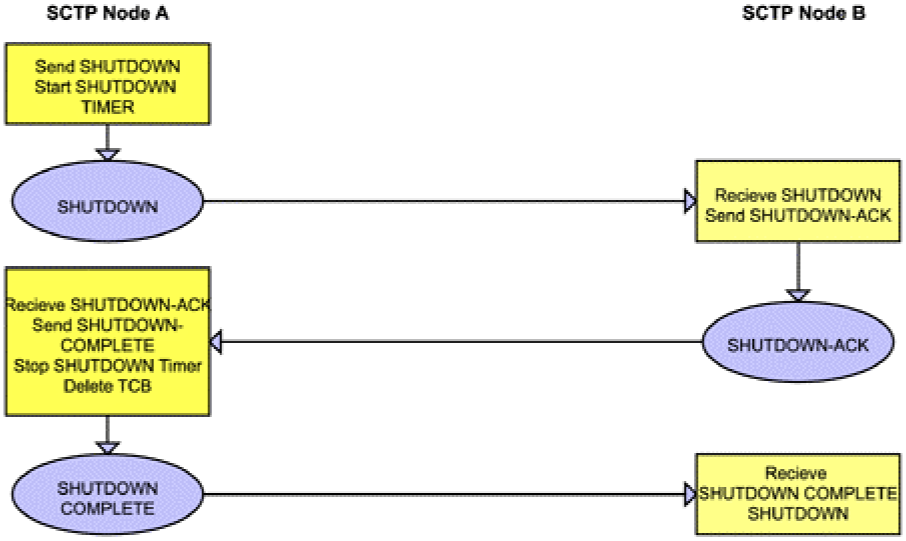
7. After each DATA chunk is received, the receiving endpoint returns a SACK chunk to acknowledge the receipt.

8. Data is transmitted until one of the endpoints decides to shutdown the association by sending a SHUTDOWN chunk bundled in one of the packets.

Association Shutdown

The SHUTDOWN chunk can be sent by any one of the two nodes. For illustrative reasons, Node A is considered here to be the initiator if the Shutdown process.

Figure 5. Graceful Termination of an SCTP Association


SCTP Node A
SCTP Node B
Send SHUTDOWN Start SHUTDOWN TIMER
SHUTDOWN
Recieve SHUTDOWN Send SHUTDOWN-ACK
Recieve SHUTDOWN-ACK Send SHUTDOWN-COMPLETE Stop SHUTDOWN Timer Delete TCB
SHUTDOWN-ACK
SHUTDOWN COMPLETE
Recieve SHUTDOWN COMPLETE SHUTDOWN

9. Node A sends a SHUTDOWN chunk to Node B and starts the shutdown timer.

10. Node B acknowledges the receipt of the SHUTDOWN chunk through generation of SHUTDOWN ACK chunk, which is sent to Node A.

11. Node A receives the SHUTDOWN ACK and responds by stopping the SHUTDOWN timer. Then, Node A generates a SHUTDOWN COMPLETE chunk and sends the chunk to Node B.

In all three stages—initiation, data transmission and shutdown—the SCTP API interfaces with the User Application Layer and the Transport Service Layer.

6. SCTP Application Programming Interface (API)

Transport protocols in IP networks, such as TCP and UDP, benefit from the sockets API. A sockets API provides a standard design interface across diverse operating systems and multiple platforms. This section describes the socket's API that is used to introduce SCTP services into TCP and UDP based applications.

In the following applications, the active primary endpoint that issues establishment requests has been denoted as the "client." The passive endpoint that receives establishment requests has been denoted as the "server."

UDP–Style Interface

The UDP–style interface accepts inbound association automatically and supports the delivery of complete messages. In addition, individual interfaces use a set of calls that execute the responses. In an SCTP association, the client and the server use these calls to establish an association. *Table 2* describes the functions of UDP–style-interface calls in sequence.

Table 2: UDP–Style-Interface Calls for the Server to Prepare Itself for Servicing Requests

Sq	CALLS	DEFINITION
1	socket ()	The socket() call creates a socket descriptor to represent SCTP endpoint.
2	bind()	The bind() call designates the primary address from the list of local addresses.
3	setsocketopt()	The setsocketopt() call sets the default association parameters.
4	rcvmsg()	The rcvmsg() call requests that data be received from the endpoint.
5	sendmsg()	The sendmsg() call requests that data be transmitted to the endpoint.
6	close()	The close() call performs a graceful shutdown.

The client uses the same interface calls listed in *Table 2* except for possibly bind() and setsocketopt() calls, which are handled by the server. In case the client or server wants to divert an existing association to a separate socket, the sctp_peeloff() call is available.

TCP–Style Interface

Since TCP connections are more prevalent in IP networks, the TCP–style interface enables connection oriented protocols to be linked with SCTP. Similar to the UDP–style interface, the client and server execute calls in the sequence described in *Table 3*.

Table 3: TCP–Style Interface Calls for the Server to Prepare itself for Servicing Requests

Sq	CALLS	DEFINITION
1	socket ()	The socket() call generates a socket descriptor corresponding to an SCTP endpoint.
2	bind()	The bind() call is used to bypass the primary address associated with the SCTP endpoint.
3	listen()	The listen() call prepares the SCTP endpoint to receive inbound associations.
4	accept()	The accept() call blocks the endpoint until a new association is initialized and a new socket descriptor is returned.
5	close()	The close() call is used to terminate an association.

While the server executes the calls described in *Table 3*, the client uses a separate set of calls to set up an association with a server. These calls are listed in sequence in *Table 4*.

Table 4: TCP–Style-Interface Calls for the client to set up an association with a server.

Sq	CALLS	DEFINITION
1	socket ()	The socket() call generates a socket descriptor corresponding to the SCTP endpoint.
2	connect()	The connect() call is used to initiate an association with the peer endpoint.
3	sendmsg()	The sendmsg() call is used to send a request to the server.
4	recvmsg()	The recvmsg() call is used to receive messages from the server.
5	close()	The close() call is used to terminate an association.

The API sendmsg() and recvmsg() calls contain unique data structures that control endpoints and provide access to supplementary information.

7. Conclusion

SCTP's appeal goes beyond being just a robust transport protocol. SCTP can be seamlessly introduced into present IP networks, simply as a higher layer user of IP services. The applicability and enhanced efficiency of SCTP over existing transport protocols and its conformity with existing systems may establish it as a protocol of choice within present and future networking systems.

Self-Test

1. The Signaling Transport group of the IETF defines SCTP standards in ___________.

   a. RFC 2890

   b. RFC 2780

   c. RFC 2960

   d. RFC 2360

2. In SCTP, data is transmitted only when a connection is established by creating a relationship, which is referred to as ________________.

   a. a link

   b. an association

   c. a collaboration

   d. an affiliation

3. _______ chunks acknowledge the receipt of DATA chunks and inform the endpoint of duplicated or missing chunks.

   a. INIT ACK

   b. COOKIE ACK

   c. HEARTBEAT ACK

   d. SACK

4. SCTP closely monitors the transmission paths to the peer endpoint using __________ chunks that test the connectivity of each path.

   a. INIT

   b. SACK

   c. HEARTBEAT

   d. SHUTDOWN

5. The ___________ feature enables SCTP endpoints to support multiple addresses.

    a. chunk bundling

    b. multi-streaming

    c. address management

    d. multi-homing

6. SCTP deploys the _________ mechanism to shield the initiation procedure from SYN-type attacks (commonly known as blind attacks) that are commonplace in TCP.

    a. fault management

    b. selective retransmission

    c. cookie

    d. chunk bundling

7. The SSN determines the sequence of delivery within the association.

    a. true

    b. false

8. There are ___ types of chunks indicated by the chunk-type field.

    a. 5

    b. 14

    c. 10

    d. 25

9. The COOKIE ACK chunk must take precedence over any DATA chunk or SACK chunk sent in an association.

    a. true

    b. false

10. ________ chunks and _________chunks can be generated by either SCTP endpoint at any time, causing the association to immediately shutdown.

    a. COOKIE ECHO; COOKIE-ACK

    b. ABORT; ERROR

    c. SHUTDOWN; SHUTDOWN ACK

    d. HEARTBEAT; HEARTBEAT ACK

11. SCTP _______ exist between the SCTP user layer and the SCTP packet service layer.

    a. IP addresses

    b. APIs

    c. DATA chunks

    d. PDUs

12. SCTP is a transport protocol that would require changing the infrastructure and network routes within present IP networks. a. IP addresses

    a. true

    b. false

Correct Answers

1. The Signaling Transport group of the IETF defines SCTP standards in ___________.

    a. RFC 2890

    b. RFC 2780

    c. RFC 2960

    d. RFC 2360

    See Overview.

2. In SCTP, data is transmitted only when a connection is established by creating a relationship, which is referred to as _______________.

   a. a link

   b. an association

   c. a collaboration

   d. an affiliation

   See Topic 2.

3. _______ chunks acknowledge the receipt of DATA chunks and inform the endpoint of duplicated or missing chunks.

   a. INIT ACK

   b. COOKIE ACK

   c. HEARTBEAT ACK

   d. SACK

   See Topic 2.

4. SCTP closely monitors the transmission paths to the peer endpoint using __________ chunks that test the connectivity of each path.

   a. INIT

   b. SACK

   c. HEARTBEAT

   d. SHUTDOWN

   See Topic 2.

5. The ___________ feature enables SCTP endpoints to support multiple addresses.

   a. chunk bundling

   b. multi-streaming

   c. address management

d. multi-homing

See Topic 3.

6. SCTP deploys the _________ mechanism to shield the initiation procedure from SYN-type attacks (commonly known as blind attacks) that are commonplace in TCP.

    a. fault management

    b. selective retransmission

    c. cookie

    d. chunk bundling

    See Topic 3.

7. The SSN determines the sequence of delivery within the association.

    a. true

    b. false

    See Topic 3.

8. There are ___ types of chunks indicated by the chunk-type field.

    a. 5

    b. 14

    c. 10

    d. 25

    See Topic 4.

9. The COOKIE ACK chunk must take precedence over any DATA chunk or SACK chunk sent in an association.

    a. true

    b. false

    See Table 1.

10. ________ chunks and _________chunks can be generated by either SCTP endpoint at any time, causing the association to immediately shutdown.

a. COOKIE ECHO; COOKIE-ACK

b. ABORT; ERROR

c. SHUTDOWN; SHUTDOWN ACK

d. HEARTBEAT; HEARTBEAT ACK

See Topic 5.

11. SCTP _______ exist between the SCTP user layer and the SCTP packet service layer.

a. IP addresses

b. APIs

c. DATA chunks

d. PDUs

See Topic 6.

12. SCTP is a transport protocol that would require changing the infrastructure and network routes within present IP networks.

a. true

b. false

See Conclusion.

Glossary

API
application programming interface

ATM
asynchronous transfer mode

IP
Internet protocol

ISDN

integrated services digital network

IXC
interexchange carriers

LAN
local-area network

MAC
message authentication code

MTU
maximum transmission unit

PDU
protocol data unit

RFC

request for comment

SCTP
stream control transmission protocol

SS7
signaling system 7

SSN
stream sequence number

TCB
transmission control block

TCP
transmission control protocol

TSN
transmission sequence number

UDP
user datagram protocol

ULP
upper layer protocol