Editor: Barry Leiba • leiba@computer.org

SCTP: What, Why, and How

Preethi Natarajan and Fred Baker • *Cisco Systems*
Paul D. Amer and Jonathan T. Leighton • *University of Delaware*

The Stream Control Transmission Protocol (SCTP) is a general-purpose IETF transport protocol with kernel implementations on various platforms. Similar to TCP, SCTP provides a connection-oriented, reliable, congestion and flow-controlled layer 4 channel. Unlike both TCP and UDP, however, SCTP offers new delivery options that better match diverse applications' needs.

Similar to TCP and UDP, the Stream Control Transmission Protocol (SCTP)[1] is a transport protocol providing end-to-end communication. SCTP was originally designed within the IETF Signaling Transport (SIGTRAN) working group to address TCP's shortcomings relating to telephony signaling over IP networks. SCTP has since evolved into a general-purpose IETF transport protocol with kernel implementations on various platforms. Similar to TCP, SCTP provides a connection-oriented, reliable, full-duplex, congestion and flow-controlled layer 4 channel. Unlike both TCP and UDP, however, SCTP offers new delivery options that better match diverse applications' needs. Here, we introduce SCTP, discuss its innovative services, and outline ongoing SCTP-related research and standardization activities.

Association Setup and Data Transfer

Similar to TCP, the SCTP end points first negotiate and set up an SCTP *association* (SCTP's term for a transport connection) before sending or receiving any data. Once data transfer is complete, the end points terminate the association.

Association Establishment

Unlike TCP, SCTP's association establishment involves a four-way handshake with a cookie mechanism. This mechanism protects the server (listener) from denial-of-service attacks based on spoofed connection setup requests (under TCP, these are called TCP-SYN attacks). As Figure 1 shows, when an SCTP server receives an association setup request (SCTP `INIT` chunk), the server doesn't immediately allocate resources for the association. Instead, the server returns an acknowledgment (SCTP `INIT-ACK` chunk) with a cookie that contains

- the necessary details required to identify and process the association,
- the cookie's life span, and
- a signature to verify the cookie's integrity and authenticity.

A legitimate client will echo the cookie back to the server (SCTP `COOKIE_ECHO` chunk). The server then uses the information in the cookie to allocate resources and establish the association. Given that data might be transmitted from the third leg (`COOKIE_ECHO`), the initial delay before data transmission due to SCTP's four-way handshake is no worse than the delay due to TCP's three-way handshake.

Message-Based Data Transfer

Similar to UDP and unlike TCP, SCTP preserves application message boundaries — messages are delivered in their entirety to the receiving application. By default, data from each `send` system call is considered an application message. However, an application can use SCTP's `EXPLICIT_EOR` socket option to write a single

 1089-7801/09/$26.00 © 2009 IEEE

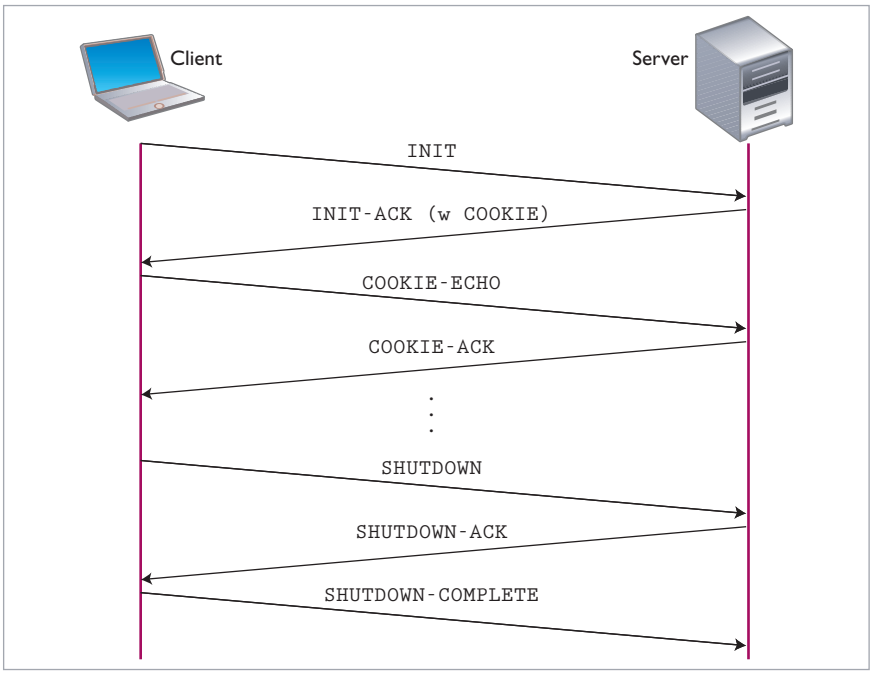

Figure 1. Stream Control Transmission Protocol (SCTP) association establishment and termination. Unlike TCP, SCTP's association establishment involves a four-way handshake with a cookie mechanism, and association termination involves a three-way handshake.

CISCO

Cisco Systems, Inc. is accepting resumes for the following position in:

Austin, TX

Customer Support Engineer

(Ref#: AUS5)

Responsible for providing technical support regarding the company's proprietary systems and software to field engineers, technicians, product support and company customers who are diagnosing, troubleshooting, repairing and debugging complex electro/mechanical equipment, computer systems and/or complex software.

Please mail resumes with reference number to Cisco Systems, Inc., Attn: J51W, 170 W. Tasman Drive, Mail Stop: SJC 5/1/4, San Jose, CA 95134. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

www.cisco.com

message over multiple `send` calls.[2] To achieve message-based transfer, SCTP first discovers the path's maximum transmission unit (MTU) and then employs a fragmentation and reassembly algorithm: the sender fragments application messages larger than the path MTU, and the receiver reassembles them. Given that message boundaries are preserved, SCTP-based applications don't require explicit message delimiters, simplifying application-level message parsing.

Partial-Reliability Extension

In TCP and plain SCTP, all transmitted data are guaranteed to be delivered to the application. RFC 3758[3] describes PR-SCTP, a *partial reliability* extension to SCTP that enables partially reliable data transfer between the transport sender and receiver. With the PR-SCTP extension, an application specifies how persistent the transport sender should be while communicating a particular message to the receiver. The application defines a *lifetime* for each message. The PR-SCTP sender tries to transmit the message during this lifetime. Upon lifetime expiration, the sender discards the message, irrespective of whether the message was successfully transmitted or acknowledged. This timed reliability is useful for applications that regularly generate new data that makes earlier data obsolete — for example, an online gaming application in which a player frequently generates new position coordinates or other data with ephemeral significance.

Unordered Data Delivery

Apart from ordered delivery, SCTP also offers the unordered data delivery service. An application message marked for unordered delivery is delivered to the receiving application as soon as the message arrives at the SCTP receiver, even if that message is out of order with respect to previously transmitted messages. Note that, unlike UDP, SCTP provides reliability, congestion, and flow control for unordered data. RFC 4168[4] proposes using SCTP's unordered service to efficiently transmit Session Initiation Protocol (SIP) signaling messages.

Association Termination

Applications can use a single SCTP association to transfer reliable, partially reliable, ordered, and unordered data. Once data transfer is complete, the SCTP end points terminate the association using a three-way handshake (see Figure 1). Unlike TCP, the SCTP association termination doesn't involve the half-closed state.[5] Consequently, when a peer closes its socket, both end points are required to close the SCTP association. Also, SCTP allows for reusing port numbers immediately after association termination, given that SCTP's *verification tag* helps correlate an incoming SCTP protocol data

unit to the correct SCTP association. Note that TCP end points must wait for twice the *maximum segment lifetime* (MSL) in the `TIME_WAIT` state before port reuse. Table 1 summarizes the various transport-level services and features offered by SCTP, TCP, and UDP.

Table 1. Comparing SCTP, TCP, and UDP.

Services/features	SCTP	TCP	UDP
Connection-oriented	Yes	Yes	No
Message-based transfer	Yes	No	Yes
Reliable data transfer	Yes	Yes	No
Partially reliable data transfer	Yes	No	No
Ordered data delivery	Yes	Yes	No
Unordered data delivery	Yes	No	Yes
Congestion and flow control	Yes	Yes	No
Protection from spoofed SYN attacks	Yes	No	NA
Allows half-closed connections	No	Yes	NA
Multistreaming	Yes	No	No
Multihoming	Yes	No	No
Dynamic address reconfiguration	Yes	No	No

Multihoming

Multihoming among networked machines and devices is a technologically feasible and increasingly economical proposition. To benefit from network interface redundancy and provide end-to-end network fault tolerance, SCTP supports multihoming at the transport layer. Unlike TCP and UDP, an SCTP association can bind multiple IP addresses at each peer. Referring to Figure 2, four distinct TCP connections would be possible between the client and server: (C_1, S_1), (C_1, S_2), (C_2, S_1), (C_2, S_2). On the other hand, SCTP end points aren't required to choose a specific IP address. Instead, a single SCTP association can consist of multiple IP addresses at each end point, which in this example would be ({C_1, C_2}, {S_1, S_2}).

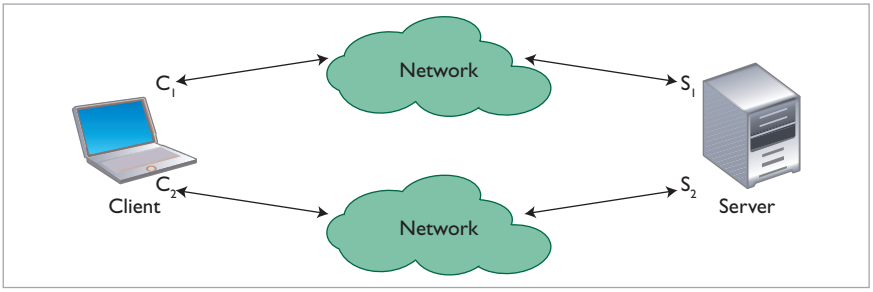

Figure 2. Stream Control Transmission Protocol (SCTP) supports transport-layer multihoming. In this figure, the SCTP association binds multiple IP addresses at each end point.

An SCTP sender chooses a single destination address as the *primary* one and transmits new data to this primary destination. The sender constantly monitors all destinations' reachability. Failure to reach the primary destination results in *failover*, such that the sender dynamically chooses an alternate destination to transmit the data until the primary destination becomes reachable again. Thus, SCTP exploits multihoming to provide failover and association survivability in the face of hardware failures.

Dynamic Address Reconfiguration

Many modern computers allow for the dynamic addition and deletion of network cards, also known as hot-pluggable interfaces. RFC 5061[6] extends SCTP's multihoming feature and allows on-the-fly layer 3 address reconfiguration. With Dynamic Address Reconfiguration (DAR), an SCTP end point can dynamically add or delete IP addresses and change the primary destination in an ongoing SCTP association. Apart from faster failover, DAR also provides efficient hand-off management in the context of mobility.[7]

Concurrent Multipath Transfer

Multihomed nodes can be simultaneously connected through multiple end-to-end paths to increase resilience to path failure (see Figure 2). For instance, users might be simultaneously connected through dial-up/broadband or via multiple wireless technologies such as 802.11b and GPRS. *Concurrent Multipath Transfer* (CMT)[8] is an experimental extension to SCTP that assumes multiple independent paths and exploits them for simultaneous transfer of new data to increase application throughput.

Multistreaming

An SCTP association can consist of multiple streams; each stream is a logical, unidirectional data flow within the association. Data ordering is preserved intrastream but not interstream — each stream has its own sequencing space, and the receiver delivers data in sequence within the stream without regard to the relative order of data arriving on other streams. Thus, independent application messages can be transmitted on different SCTP streams to preserve their logical separation during transmission and delivery. The next section discusses how

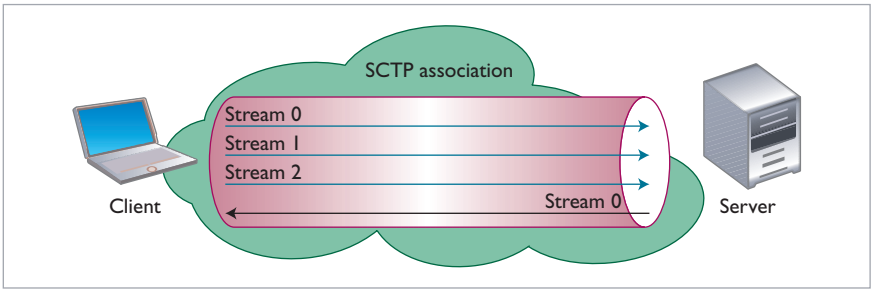

Figure 3. Multistreamed Stream Control Transmission Protocol (SCTP) association. This association consists of three outbound streams (from client to server) and one inbound stream.

independent HTTP transactions can leverage SCTP streams to improve user-perceived Web performance.

An SCTP association can have multiple (up to 65,536) input or output streams. The SCTP end points negotiate the number of streams during or after association establishment. In Figure 3, the client uses three output streams (numbered 0 to 2) to transmit data to the server and has only one input stream (numbered 0) to receive data from the server. Note that all streams within an SCTP association are subject to shared congestion control, and thus SCTP's multistreaming adheres to TCP's fairness principles.

Case Study: SCTP Streams for HTTP Transfers

HTTP requires a reliable transport protocol for end-to-end communication. Although historically TCP has been used for this purpose, HTTP doesn't require TCP, which offers a single sequential byte stream to a Web server. In the case of HTTP 1.1 with persistence and pipelining, the independent HTTP responses are serialized and sent sequentially over TCP's single bytestream. In addition, TCP provides in-order delivery within the byte stream; a lost piece of one HTTP response delays the delivery of successively received other responses until the sender retransmits the lost piece. This problem, known as *head-of-line* (HOL) *blocking*, is due to the fact that TCP can't logically separate independent HTTP responses in its transport and delivery mechanisms.

When independent HTTP transactions are transmitted over different SCTP streams (within a single association), the SCTP receiver delivers these transactions to the application without inter-transaction HOL blocking. Evaluations using an SCTP-capable Apache Web server and Firefox browser show that SCTP streams eliminate HOL blocking and significantly improve Web response times in end-to-end paths that have high latency and loss rates (see http://pel.cis.udel.edu).[9,10] Ongoing work includes identifying other SCTP services that are useful for HTTP, standardizing HTTP over SCTP,[11] and merging the SCTP-specific modifications to Apache and Firefox into the corresponding source repositories. A patch for adding SCTP support to Mozilla's portable runtime library is currently available.[9]

Socket API Extensions

SCTP provides many TCP services and also incorporates semantics closer to UDP. Therefore, two different SCTP socket models exist:[2]

- *one-to-one model* — an SCTP socket is used to communicate to one SCTP end point. This model is similar to TCP's and can port existing TCP-based applications to use SCTP with very little effort.
- *one-to-many model* — an SCTP socket can be used to communicate to multiple SCTP end points, similar to a UDP socket.

SCTP supports all existing socket API calls such as `bind`, `listen`, `connect`, `read`, and `write` to provide the regular transport services. However, the existing socket API is inefficient and insufficient to take advantage of many of SCTP's innovative services, such as multihoming and multistreaming. Consequently, the IETF has introduced new SCTP-specific APIs. For example,

- `sctp_bindx` lets an application bind an SCTP socket to multiple IP addresses;
- `sctp_connectx` lets the application connect to a peer using multiple IP addresses; and
- `sctp_sendmsg` includes special arguments to send data on a particular SCTP stream, specify flags

CISCO

Cisco Systems, Inc. is accepting resumes for the following position in:

New York, NY

Network Consulting Engineer
(Ref#: NY1)

Provide consultative, proactive and/or reactive support to Company accounts.

Please mail resumes with reference number to Cisco Systems, Inc., Attn: J51W, 170 W. Tasman Drive, Mail Stop: SJC 5/1/4, San Jose, CA 95134. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

www.cisco.com

for unordered delivery, specify a lifetime value for a partially reliable message, and so on.

New SCTP-specific socket options, events, and notifications have also been introduced. SCTP's events and notifications let an application gather (and react to) relevant information about network status changes, peer address changes, remote operational errors, and so on. A complete list of the APIs, socket options, events, and notifications is available elsewhere.[2]

Standardization Activities and Deployment Status

SCTP is maintained within the IETF's Transport Area working group (TSVWG). Ongoing standardization efforts within the TSVWG include specifying modular extensions, socket API extensions (just discussed) and security enhancements for SCTP. Activities at the IETF's Behave working group include specifying *network address translation* (NAT) behavior for SCTP traffic and SCTP over UDP encapsulation to enable SCTP traversal through middleboxes that don't yet understand SCTP.

Kernel-level SCTP implementations are available on various platforms, such as AIX, FreeBSD, HP-UX, Linux, Mac OS X, and Solaris/OpenSolaris. SCTP is available as a kernel driver for Windows XP and Vista.

This article discussed SCTP's unique features unavailable in both TCP and UDP. We encourage application developers to explore and experiment with SCTP and consider utilizing its services for superior performance.

References

1. R. Stewart, ed., *Stream Control Transmission Protocol*, IETF RFC 4960, Sept. 2007; http://tools.ietf.org/html/rfc4960.
2. R. Stewart et al., "Sockets API Extensions for Stream Control Transmission Protocol (SCTP)," IETF Internet draft, work in progress, Feb. 2009.
3. R. Stewart et al., *SCTP Partial Reliability Extension*, IETF RFC 3758, May 2004; http://tools.ietf.org/html/rfc3758.
4. J. Rosenberg et al., *SCTP as a Transport for the Session Initiation Protocol*, IETF RFC 4168, Oct. 2005; http://tools.ietf.org/html/rfc4168.
5. J. Postel, *Transmission Control Protocol,* IETF RFC 793, Sept. 1981; http://tools.ietf.org/html/rfc793.
6. R. Stewart et al., *SCTP Dynamic Address Reconfiguration*, IETF RFC 5061, Sept. 2007; http://tools.ietf.org/html/rfc5061.
7. S. Zeadally et al., "An Empirical Analysis of Handoff Performance for SIP, Mobile IP, and SCTP Protocols," *Wireless Personal Communications J.*, vol. 43, no. 2, Oct. 2007, pp. 589–603.
8. J. Iyengar et al., "Concurrent Multipath Transfer using SCTP Multihoming over Independent End-to-end Paths," *IEEE/ACM Trans. Networking*, vol. 14, no. 5, 2006, pp. 951–964.
9. "HTTP over SCTP versus HTTP over TCP Movies," Oct 2007; www.cis.udel.edu/~leighton/.
10. P. Natarajan et al., "Multistreamed Web Transport for Developing Regions," *Proc. ACM SIGCOMM Workshop on Networked Systems for Developing Regions* (NSDR 08), Aug. 2008.
11. P. Natarajan et al., "Using SCTP as a Transport Layer Protocol for HTTP," IETF Internet draft, work in progress, Mar. 2009.

Preethi Natarajan is a software engineer at Cisco Systems. Her research interests include transport protocols, congestion control, and mobility solutions. Natarajan has a PhD in computer Science from the University of Delaware. Contact her at prenatar@cisco.com.

Fred Baker is a fellow at Cisco Systems. He's an active member at the IETF, has chaired several IETF working groups, and was the IETF chair from 1996 to 2001. Baker served on the Internet Architecture Board from 1996 to 2002 and has coauthored or edited roughly 40 RFCs and contributed to others. Contact him at fred@cisco.com.

Paul D. Amer is an Alumni Distinguished Professor and directs the Protocol Engineering Lab at the University of Delaware. His research interests include transport protocols, SCTP, and data compression in multimedia. Amer has a PhD in computer science from the Ohio State University. Contact him at amer@cis.udel.edu.

Jonathan T. Leighton is a member of the technical staff at the University of Delaware's Protocol Engineering Lab. His research interests include transport protocols in general with particular interest in SCTP and its applications. Leighton has an MS in computer science from the University of Delaware and a PhD in solid mechanics from Brown University. Contact him at leighton@cis.udel.edu.

CISCO

Cisco Systems, Inc. is accepting resumes for the following positions in:

San Jose/Milpitas/Santa Clara, CA:

Applications Engineer (Ref#: SJ50)
Manages small, medium, large/complex and multiple projects throughout the application project lifecycle.

Network Consulting Engineer (Ref#: SJ9)
Provide consultative, proactive and/or reactive support to Company accounts.

Service Operations Program Manager (Ref#: SJ32)
Interacts across different project teams to secure resources and analyze business needs.

Database Administrator (Ref#: SJ43)
Provides database design and management function for business and/or engineering computer databases. Includes architecture, design, build and management of databases of varying sizes and complexities.

Please mail resumes with reference number to Cisco Systems, Inc., Attn: J51W, 170 W. Tasman Drive, Mail Stop: SJC 5/1/4, San Jose, CA 95134. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

www.cisco.com