ACCEPTED FROM OPEN CALL

SCTP: State of the Art in Research, Products, and Technical Challenges

Shaojian Fu and Mohammed Atiquzzaman, University of Oklahoma

ABSTRACT

The Stream Control Transmission Protocol (SCTP) is being standardized by the IETF as a reliable transport protocol to transport SS7 signaling messages over IP networks. Due to its attractive features such as multistreaming and multihoming, SCTP has received much attention from the network community, in terms of both research and development. This article introduces the main features of SCTP, and discusses the state of the art in SCTP research and development activities. We also provide a survey of the available products that use SCTP. Finally, with a view to stimulating further research in this area, the challenges faced by the SCTP research community are identified.

INTRODUCTION

The last few years have witnessed a strong trend of convergence in public switched telephone networks (PSTNs), integrated services digital network (ISDN), and IP-based networks, resulting in the bloom of IP telephony or voice over IP (VoIP) applications. To reduce communication costs, it is now a common practice to transport voice over wide-area IP networks. However, many of the important services provided by PSTN networks need the support of an SS7 signaling network, which is a separate network (from voice circuits) used to carry setup and teardown messages, billing information, routing queries, and so on. SS7 is designed as an open-ended common channel signaling standard, and is currently deployed by virtually all telephone service providers and interexchange carriers. To achieve complete IP telephony, in addition to just transporting raw voice streams between VoIP gateways, one more task needs to be accomplished: transporting SS7 signaling messages over IP networks.

The transport of SS7 signaling messages has stringent requirements for reliable and timely delivery, since the information carried is critical to the operation of the network. Any error or excessive delay in the transport of these messages may result in circuit establishment failure or billing errors. Until late 2000, the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) were the only available standard transport layer protocols in the TCP/IP protocol suite. Since UDP is not a connection-oriented reliable protocol, it cannot be used as the transport protocol for signaling messages. The Internet Engineering Task Force Signaling Transport (IETF SIGTRAN) working group (founded in November 1998) also evaluated the applicability of TCP for transporting signaling messages, and identified several deficiencies of TCP [1]:

- TCP's strict byte-order delivery gives rise to head-of-line (HOL) blocking in some applications.
- TCP is stream-oriented instead of message-oriented.
- TCP does not support multihoming, which is crucial in high-availability environments such as SS7 signaling transport.
- TCP is vulnerable to blind denial of service (DoS) attacks by SYN segments.

To overcome the above limitations of TCP for transport of signaling messages, a new transport protocol, Stream Control Transmission Protocol (SCTP), was proposed by the IETF in October 2000 to accomplish signaling transport over IP networks [1]. It was soon noticed that SCTP should be useful in a wider range of applications than just for signaling transport; as a result, the standardization work of SCTP was moved from SIGTRAN to the Transport Area Working Group (TSVWG) of the IETF in February 2001.

The design of SCTP absorbed many of the strengths of TCP, such as window-based congestion control, error detection, and retransmission, that led to its success during the explosive growth of the Internet. Moreover, SCTP incorporated several new features that are not available in TCP. The two most prominent of these, which will be discussed in more detail later, are:

Multihoming: Multihoming allows two endpoints to set up an association with multiple IP addresses for each endpoint (In SCTP, *association* is the name for the communication relationship between endpoints; it is loosely comparable to *connection* in TCP). This built-in support for multihomed endpoints can utilize redundancy in the network, and allow high-

The research reported in this article was partially funded by National Aeronautics and Space Administration (NASA) Grants NAG3-2528 and NAG3-2922.

availability applications perform switchovers during link failure situations without interrupting data transfer.

Multistreaming: Multistreaming is used to alleviate the head-of-line (HOL) blocking effect resulting from TCP's strict byte-order delivery policy. Each stream is a *subflow* within the overall data flow, and the delivery of each subflow is independent of each other.

Due to its new attractive features, SCTP has received much attention from the research community, and has become one of the hot topics in networking technology. Two excellent tutorials have been published recently introducing SCTP to the research community [2, 3].

The main objective of this article is to provide readers with a comprehensive review of the most recent research activities related to SCTP, survey currently available free and commercial products, and outline problems and issues that remain open for research and development. The contributions of this article can be summarized as follows:

- Provide a comprehensive survey of the state-of-the-art SCTP research in congestion control, multihoming, multistreaming, out-of-order service, partial reliability, and application in the wireless/mobile/satellite environment.
- Summarize the various implementations of SCTP and commercially available products.
- Identify some of the challenges and issues faced by the SCTP research community.

The rest of the article is organized as follows. The main features of SCTP are reviewed to familiarize the readers with the fundamental concepts of SCTP. We discuss the state of the art in SCTP research activities and provide a survey of available products. To stimulate future research in the area of SCTP, we identify a number of issues and challenges in further development and widespread deployment of SCTP.

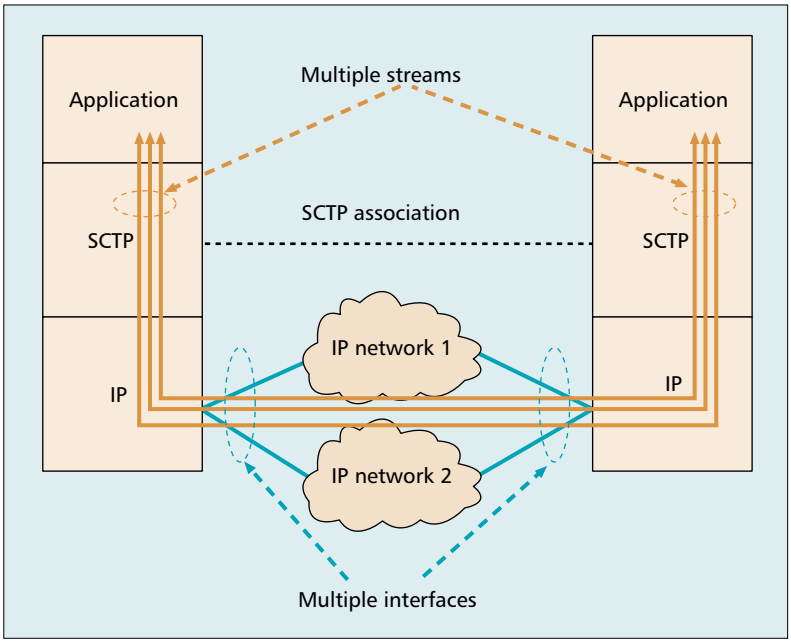

Figure 1. *A schematic view of an SCTP association.*

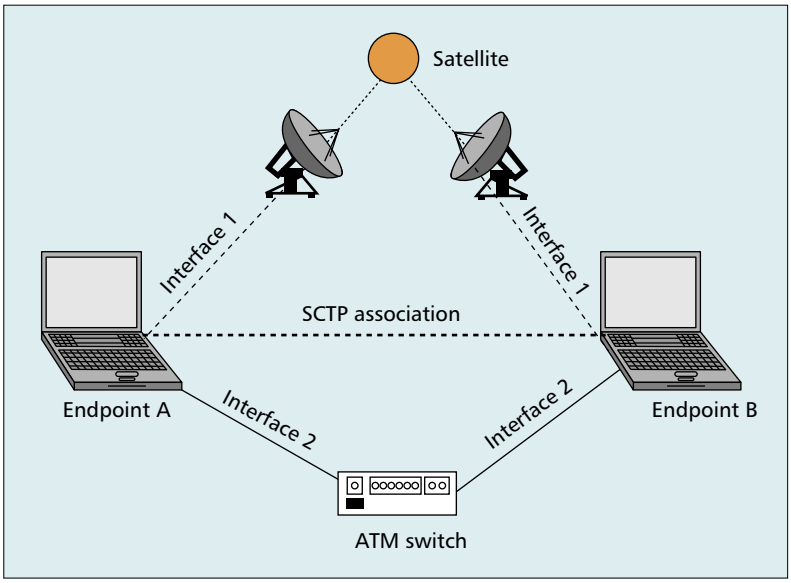

Figure 2. *An SCTP association with multihomed endpoints.*

Main Features of SCTP

Like TCP, SCTP resides in the transport layer of the Internet protocol stack as shown in Fig. 1, which also illustrates an SCTP association using multihoming and multistreaming.

Multihoming

Multihoming allows an association between two endpoints span across multiple IP addresses or network interface cards. An example of SCTP multihoming is shown in Fig. 2, where both endpoints A and B have two interfaces bound to an SCTP association. The two endpoints are connected through two types of links: satellite at the top and asynchronous transfer mode (ATM) at the bottom. One of the addresses is designated as the primary, while the other can be used as a backup in case of failure of the primary address, or when the upper-layer application explicitly requests use of the backup. Retransmission of lost packets can also be done over the secondary address. The built-in support for multihomed endpoints by SCTP is especially useful in environments that require high availability of the applications, such as SS7 signaling transport. A multihomed SCTP association can speed up recovery from link failure situations without interrupting any ongoing data transfer.

Multistreaming

Multistreaming allows data from the upper layer application to be multiplexed onto one channel (called association in SCTP) as shown in Fig. 3. Sequencing of data is done within a stream; if a segment belonging to a certain stream is lost, segments (from that stream) following the lost one will be stored in the receiver's stream buffer until the lost segment is retransmitted from the source. However, data from other streams can still be passed to the upper-layer application. This avoids the HOL blocking found in TCP, where a single stream carries data from all the

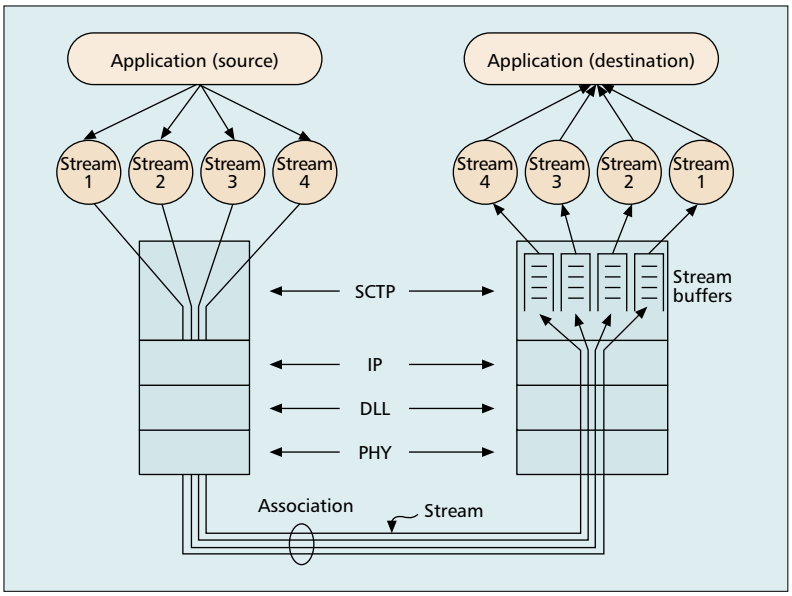

Figure 3. *An SCTP association consisting of four streams carrying data from one upper layer application.*

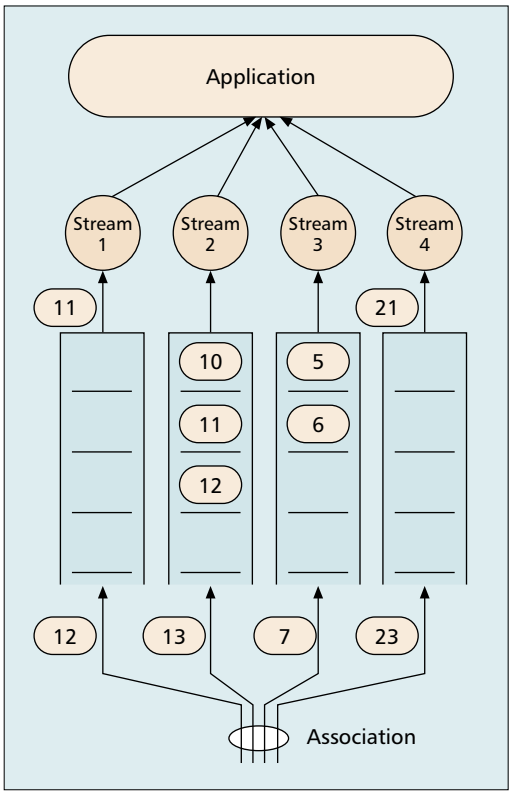

Figure 4. *An illustration showing HOL blocking of individual streams at the receiver.*

upper-layer applications. In other words, the HOL effect is limited within the scope of individual streams, but does not affect the entire association.

Multistreaming and HOL blocking are illustrated in Fig. 4 where an SCTP association consisting of four streams is shown. Segments are identified by stream sequence numbers (SSNs) [1] that are unique within a stream, but different streams can have the same SSN. In the figure, SSN 11 in stream 1 has been delivered to the upper-layer application, and SSN 9 of the second stream is lost in the network; SSNs 10, 11, 12 are therefore queued in the buffer of the second stream, waiting for retransmitted SSN 9 to arrive. Arriving SSN 13 at stream 2 will also be queued. Similarly, SSN 4 of stream 3 is missing during the transmission resulting in the blocking of SSNs 5, 6, and 7. For stream 4, SSN 21 is being delivered to the upper-layer application, while arriving SSN 23 will be queued in the buffer because of missing SSN 22. Note that when SSN 12 arrives at the buffer of stream 1, it can be delivered immediately even if the other streams are blocked. This illustrates that segments arriving on stream 1 can still be delivered to the upper-layer application, although streams 2 and 3 are (and stream 4 will be) blocked because of lost segments.

An example application of using SCTP multistreaming in Web browsing is shown in Fig. 5. Here, an HTML page is split into five objects: a Java applet, an ActiveX control, two images, and plain text. Instead of creating a separate connection for each object as in TCP, SCTP makes use of its multistreaming feature to speed up the transfer of HTML pages. By transmitting each object in a separate steam, the HOL effect between different objects can be eliminated. If one object is lost during the transfer, the others can still be delivered to the Web browser at the upper layer while the lost object is being retransmitted from the Web server. This results in a better response time to users while opening only one SCTP association for a particular HTML page.

Congestion Control

SCTP congestion control is based on the well proven rate-adaptive window-based congestion control scheme of TCP. This ensures that SCTP will reduce its sending rate during network congestion and prevent congestion collapse in a shared network. SCTP provides reliable transmission and detects lost, reordered, duplicate, or corrupt packets. It provides reliability by retransmitting lost or corrupt packets. However, there are several major differences between TCP and SCTP:

- SCTP incorporates a fast retransmit algorithm based on SACK gap reports similar to that of TCP SACK. This mechanism speeds up loss detection and increases the bandwidth utilization. One of the major differences between SCTP and TCP is that SCTP does not have an explicit fast recovery phase. SCTP achieves fast recovery automatically with the use of SACK [1].
- Compared to TCP, the use of SACK is mandatory in SCTP, which allows more robust reaction in the case of multiple losses from a single window of data. This avoids a time-consuming slow start stage after multiple segment losses, thus saving bandwidth and increasing throughput.
- During slow start or congestion avoidance of SCTP, the congestion window (*cwnd*) is increased by the number of acknowledged bytes; in TCP it is increased by the number of ACK segments received. Since the TCP sender

increases the size of *cwnd* based on the number of arriving ACKs, the widely used delayed ACK will reduce the number of ACKs, which in turn slows the *cwnd* growth rate. In long propagation delay scenarios such as satellite networks, this effect is especially serious. Mark Allman has also proposed a byte counting algorithm for TCP; it was incorporated into IETF RFC 2581, which allows a TCP sender to use byte counting to increase *cwnd* during congestion avoidance, although still not during a slow start.

- During congestion avoidance of SCTP, *cwnd* can only be increased when the full *cwnd* is utilized; this restriction does not exist in TCP. The rationale behind this restriction is as follows: *cwnd* not utilized fully means that the endpoint has not used all the network resources available, so why should the network allocate more resources to it? Otherwise, if the endpoint maintains a low sending rate without fully utilizing the available *cwnd*, there may be no loss indication (DupACKs or timeout) generated by the network, and the *cwnd* will continue increasing to a large value. If the endpoint suddenly sends out a big burst of data, it will probably cause congestion in the network.
- TCP begins fast retransmission after receipt of three DupACKs; SCTP begins after four DupACKs. SCTP is able to clock out new data on receipt of the first three DupACKs and retransmit a lost segment by ignoring whether the flight size is less than *cwnd*; TCP can only begin data retransmission on receipt of the third DupACK.

SECURITY

Because a transport protocol could carry sensitive information like billing data or critical signaling messages, the developers of SCTP paid attention to the security mechanisms of the protocol. SCTP [1] identified the following two security objectives:

- The *service availability* of reliable and timely data transport
- The *integrity* of the user-to-user information carried by SCTP

Protecting Availability of Services — One of the common threats to the first objective is blind DoS attacks by flooding the target host with continuous connection setup requests (e.g., SYN attacks in the case of TCP). The root of this problem is that the attacked host maintains in its memory useless state information regarding each pending connection, which will eventually exhaust the memory space of the system. SCTP eliminates the risk of DoS attacks by utilizing a *four-way handshake* sequence and a *cookie* mechanism to avoid maintaining state information for incomplete associations.

The messages exchanged and states of endpoints during an association setup are shown in Fig. 6, where we can see that the SCTP server remains in the *closed* state, and does not store any information regarding the association until the receipt of the COOKIE ECHO message. The cookie is transferred between the endpoints within the INIT ACK and COOKIE ECHO messages. The cookie should include information about endpoint IP addresses, stream numbers, advertised receiver windows, initiation tags, timestamp, time to live (TTL) of the cookie, and a digital signature to authenticate the cookie. The server can extract all the information it needs to complete the association setup from the COOKIE ECHO. The main idea of the cookie mechanism is to store the state information on either the client side or the network, rather than in the memory of the server. The use of a cookie defers the resource reservation at the server until the completion of authentication of the information echoed back by the cookie. This is a simple but powerful design to resist DoS attacks.

Protecting the Integrity of User-to-User Information — If the objective of the attack is to break the integrity or confidentiality of the

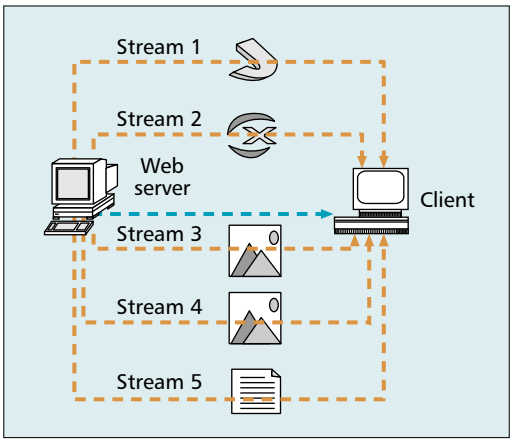

Figure 5. *Multistreaming in Web browsing.*

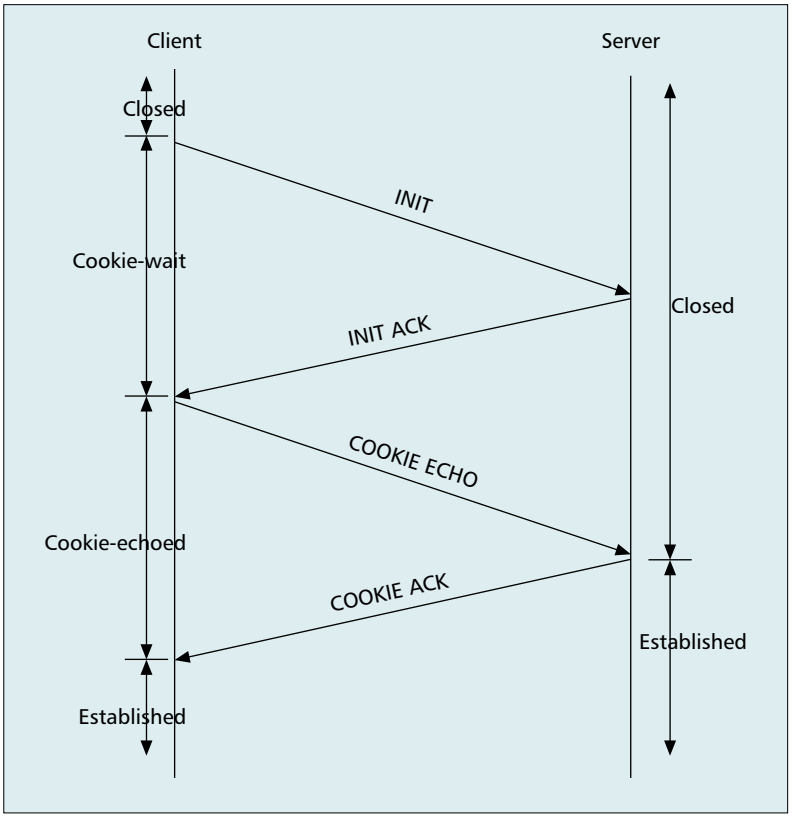

Figure 6. *SCTP association setup message sequence.*

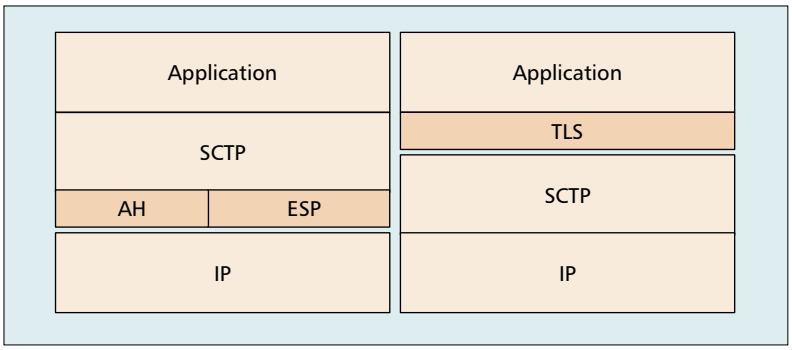

Figure 7. *Usage of IPSec and TLS with SCTP.*

Protocol	TCP	SCTP
Setup messages	Three-way handshake	Four-way handshake
Shutdown messages	Four-way handshake	Three-way handshake
Half-open support	Supported	Not supported
Ordered delivery	Strict ordered	Ordered within a stream
Unordered delivery	Not supported	Supported
Message boundary	No boundary Stream-oriented	Boundary preserved Message-oriented
Multihoming	Not supported	Supported
SACK support	Optional	Mandatory
Keep-alive heartbeat	Optional	Mandatory
Heartbeat interval	≥ Two hours	30 seconds by default

Table 1. *Comparison of TCP and SCTP.*

user-to-user information transfer, the payload of SCTP will be the target of the attack. In this case, IPSec (RFC 2401 and RFC 2406) or Transport Layer Security (TLS, RFC 2246) should be used to protect the confidentiality and integrity of the payload.

IPSec is designed to provide an interoperable security architecture for IPv4 and IPv6, based on cryptography at the network layer. IPSec provides security services at the IP layer by allowing an endpoint to select the required security protocols, determine the algorithms to use, and exchange cryptographic keys required to provide the requested services. In the IPSec protocol suite, there are two security protocols, Authentication Header (AH) and Encapsulating Security Payload (ESP). ESP provides data integrity, authentication, and secrecy services, while AH is less complicated and only provides the first two services.

The protocol stacks when IPSec and TLS are used with SCTP are shown in Fig. 7. Note that in this figure, SCTP can be used with both AH and ESP, but ESP must be supported by the endpoints if the application is to transport SS7 signaling messages. Some issues in using IPSec and Internet Key Exchange (IKE) with SCTP's multihoming will be described later.

The TLS protocol provides an interoperable and extensible security framework based on transport layer services. The position of TLS in the protocol stack is shown in Fig. 7. The TLS protocol can provide the following services:

- Server authentication: Allows the user to confirm the server's identity, and prevents a masquerading attack.
- Client authentication: Allows a server to authenticate the user's identity.
- Encrypted message transfer: All information sent between the client and server is encrypted, and thus provides confidentiality protection.

The specific requirements for using TLS over SCTP can be found in RFC 3436.

Differences from TCP

We have described the differences between the congestion control mechanisms of TCP and SCTP. In Table 1 we describe other differences between them. The first three rows compare the messages exchanged during TCP connection/SCTP association setup and shutdown. "Half-open" in the third row represents a situation where one endpoint has finished its data transfer while expecting to receive further data from its correspondent endpoint (i.e., the connection/association is open only in one direction). TCP supports half-open connection through a four-way handshake shutdown sequence; SCTP uses a three-way handshake for shutdown and does not support half-open association.

The fourth and fifth rows of the table relate to the delivery of segments to the application at the receiver. TCP only supports strict ordered delivery, and can result in HOL blocking in some cases. SCTP can independently deliver to the application layer received segments belonging to different streams, provided that the sequence within the stream is preserved; SCTP can also optionally support unordered delivery, which is not possible in TCP.

The comparison in the sixth row concerns message boundary after transmission by the transport layer protocols. TCP is a stream-oriented protocol, and application data are treated as a continuous byte stream instead of discrete messages. Therefore, application developers must add their own markings between messages, and must use the TCP PUSH flag to ensure that the complete message is received within a reasonable time. In comparison, SCTP is message-oriented; as long as there is space in the receiver buffer, the whole message is delivered by itself without getting mixed with another message.

The last two rows of Table 1 relate to the keep-alive messages. A keep-alive mechanism periodically probes the other end of a connection when the connection is otherwise idle, even when there is no data to be sent. In TCP the question of whether this mechanism should be implemented by the transport layer or the application itself is highly controversial. The opponents of implementing keep-alive in TCP think this mechanism will waste bandwidth unnecessarily. If a specific TCP implementation chooses to implement a keep-alive mechanism, the default value of the heartbeat interval should not be less than 2 h (RFC 1122). SCTP designers believe that the ability to monitor the peer address's reachability is

crucial in high-availability applications. For example, in the SS7 network it is desirable to get a link failure alarm as soon as possible so that the problem can be taken care of immediately. In this sense, conserving bandwidth is not a principal consideration. Therefore, keep-alive heartbeat is provided in SCTP as a standard mechanism instead of depending on the implementation, as is done in TCP. Moreover, the default heartbeat interval is also reduced to 30 s.

The differences revealed in comparing the two transport layer protocols reflect understanding of the deficiencies of TCP by the research community during the past 20 years of practice.

The State of the Art in Research Activities

In this section we provide a comprehensive survey of the current research activities in the area of SCTP, and attempt to provide readers with a clear vision of the state of the art in SCTP research. We discuss various aspects of research in SCTP, and cite recently published papers when applicable.

Congestion Control

SCTP's congestion control mechanisms are slightly different from that of TCP. When SCTP is used as a general-purpose transport protocol, one of the concerns is whether it can coexist fairly with TCP in a shared network such as the Internet. If the difference in the congestion control mechanisms causes SCTP to act more aggressively than TCP, SCTP traffic will consume more network resources than TCP traffic, which could result in unfairness when users compete for network resources. Jungmaier *et al.* [4] investigated the flow control and bandwidth-sharing behavior of SCTP when SCTP associations and TCP connections share common wide area network links, as shown in Fig. 8.

By measuring the link layer load imposed by the different flows of the two protocols, it was shown that SCTP traffic has the same impact on the congestion control decision of TCP connections as normal TCP traffic; if the transport protocol for some of the existing applications was changed from TCP to SCTP, other TCP applications' performance would not be affected. This ensures that the introduction of SCTP traffic into an existing TCP/IP network will not degrade the performance of TCP traffic, and the traffic of the two protocols can share network resources fairly. This is a desirable property that helps in the gradual and seamless deployment of SCTP in the Internet without affecting existing traffic.

To investigate whether this fairness property between SCTP and TCP still holds in a network containing high bandwidth-delay product paths, Alamgir *et al.* [5] compared the congestion control mechanisms of TCP and SCTP in a satellite environment, which is typical of this kind of network. The simulation scenario is shown in Fig. 9.

The study presented a detailed case study of the retransmission policies of the two protocols,

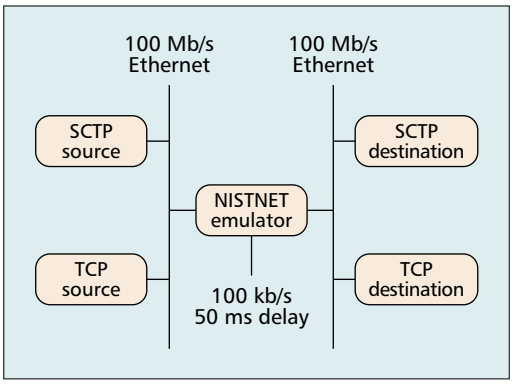

Figure 8. *TCP and SCTP sharing a common WAN link.*

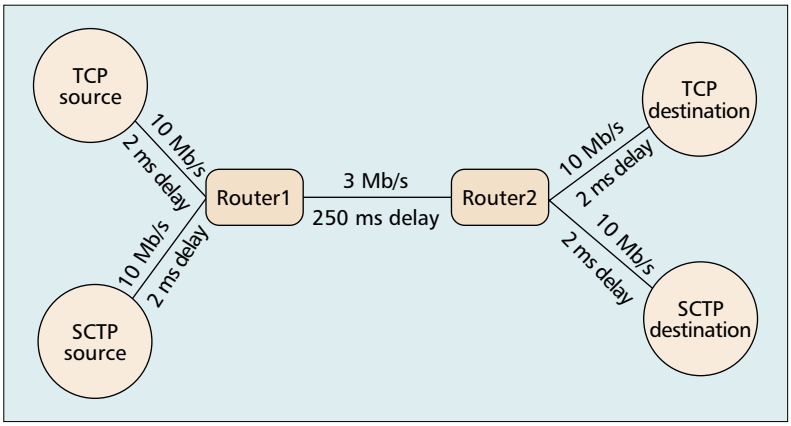

Figure 9. *TCP and SCTP sharing a common satellite link.*

and showed that under certain network scenarios SCTP can achieve better performance than TCP even when both the protocols share a satellite path fairly. The throughput improvement, which was reported to be up to 30.6 percent, resulted from the different retransmission mechanisms of TCP and SCTP during the congestion avoidance phase.

Multihoming

The primary objective of SCTP multihoming is to achieve fast recovery from fault conditions in high-availability environments. Jungmaier *et al.* [6] investigated the effect of SCTP multihoming in the recovery of SS7 network linkset failures. The authors compared transport-layer switchover based on SCTP multihoming against manual user-layer switchover. The two approaches are shown in Fig. 10. In the top part of the figure, only one association is set up across the two available links. In the lower part of the figure two associations are established, one association for each of the link paths. The first approach is called *transport-layer* switchover because the failover is done by the SCTP layer, and is transparent to user applications. The second approach requires the application to be aware of the failover, and is thus called *user-layer* switchover. It was found that the first approach offers a smoother transition by keeping the average segment delay during failover at a much lower value than the second approach. The multihoming feature of

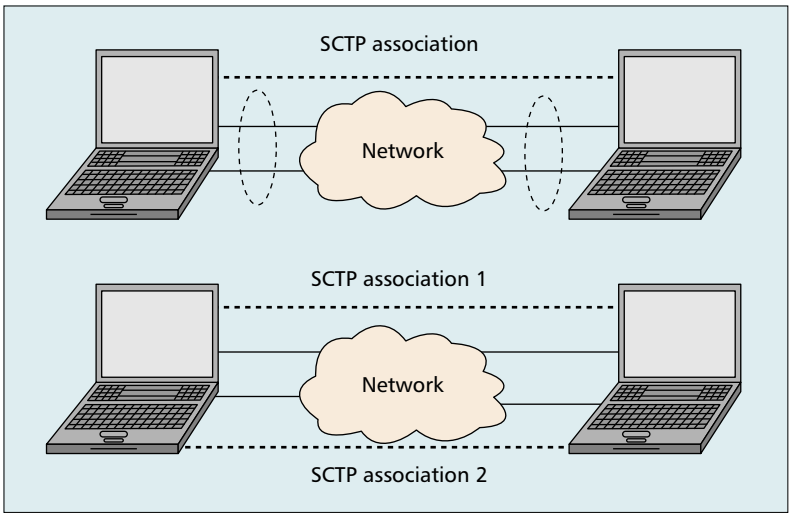

Figure 10. *Two approaches to link failover protection.*

SCTP can help the endpoints detect the link failure earlier.

The current SCTP standard [1] does not recommend the use of SCTP multihoming for load balancing over multiple network paths. It recommends that the backup path only be used when the primary path fails or to carry retransmitted segments. Simultaneous data transfer over multiple destination addresses could cause packet reordering in some scenarios (depending on network conditions, e.g., propagation delay, bandwidth and path maximum transmission unit, MTU). This effect was investigated in detail by Iyengar *et al.* [7]. The main reason for this reordering problem is that the congestion control mechanism of the sender is unaware of the destination address change. It will therefore mistakenly interpret the packet reordering introduced by address change as packet loss in the network. To solve the above problem, an algorithm called Changeover Aware Congestion Control (CACC) has been proposed to let the sender maintain a per-destination state about the segment sequence number, which will eliminate the unnecessary retransmissions after the address switchover. Use of SCTP multihoming to improve mobile data transmission is also being explored by a number of research groups, as discussed in detail later.

Multistreaming

For applications that have independent elements, such as multimedia, SCTP's multistreaming feature can be used to segment the elements into separate streams and thereby eliminate the HOL effect described earlier. In [8], Caro *et al.* shows the ability of SCTP to reduce the latency of streaming multimedia in high-loss environments. The experiment uses the standard GIF compression format and eight parallel streams for the transmission of images. A relatively negative network condition is used: 9.6 kb/s bandwidth and 10 percent loss rate in the network. The experimental results show that multistreaming results in slower degradation of network throughput as the loss rate increases. Moreover, user satisfaction is increased with the improved multimedia quality provided by this feature. This effect is a result of partial-order delivery (maintaining segment sequence only within streams but not within the overall association) of SCTP when losses occurs during the transmission.

For the first time, Atiquzzaman *et al.* [9] showed that multistreaming results in higher goodput than a single stream when the receiver buffer is constrained, as in the case of wireless handheld devices. The study also demonstrated that the multistreaming feature of SCTP results in reduced buffer requirements at the receiver in the presence of losses in satellite networks. The above advantages make SCTP an attractive transport protocol for wireless handheld devices.

Out-of-Order Service

Previous studies have mostly used UDP for the session initiation protocol (SIP), an application layer signaling protocol to establish multimedia sessions. SIP proxies are used to aggregate the traffic from one service provider to another, and provide call routing capabilities to maximize network performance in packet voice networks. Camarillo *et al.* [10] have investigated the use of SCTP's out-of-order service for the transport of SIP messages between two proxies with a view to investigating the effectiveness of SCTP in reducing HOL blocking. The authors used only one stream per association, and the stream was configured to deliver segments to upper layers completely out of order.

By comparing the packet delay introduced by SCTP, TCP, and UDP, they have shown that for the above proxy-to-proxy scenario and under moderate packet losses, SCTP does not provide a statistically substantial improvement over TCP. They have also shown that UDP's late packet loss detection, lack of congestion control mechanisms, and lack of transport layer fragmentation make UDP unsuitable for proxy-to-proxy communication in SIP.

Partial Reliability Extension

The current SCTP only specifies the reliable transport of messages. However, future Internet applications, such as real-time multimedia traffic (e.g., VoIP), may require partial reliable transport of messages. To accommodate partial reliable transport, a new IETF draft specifies the use of SCTP as a partial reliable transport protocol (like UDP) while still maintaining the network-friendly congestion control mechanisms of SCTP. This will allow SCTP to carry traffic requiring partial reliability (e.g., real-time multimedia traffic) in addition to traffic requiring full reliability (FTP, HTTP, etc.).

In SCTP, the cumulative ACK point at the receiver is advanced when new data is received immediately following the previous cumulative ACK point. This extension allows an SCTP sender to signal its peer receiver to move the cumulative ACK point forward even without receiving any new data. When both sides of an SCTP association support this extension, it can be used by an SCTP implementation to provide partially reliable data transmission service to an upper-layer protocol.

Application in the Wireless/Mobile Environment

Although SCTP was initially designed primarily as a transport protocol for signaling messages, there has been significant research activity in the application of SCTP in wireless/mobile networks. Current research in this area can be classified into the following categories.

The Effect of Delay Spikes on SCTP — Like TCP, SCTP is also designed with wired networks in mind. There are a number of problems in wireless communications, one of which is the spurious timeout and retransmission caused by sudden long delays, called *delay spikes*. In a previous paper [11] we studied the effect of delay spikes on SCTP in a wireless mobile environment. We showed that, like TCP, SCTP also suffered go-back-*N* behavior after a delay spike. We further showed that SCTP SACK could be used to eliminate spurious fast retransmission in SCTP. In a lossy network with small bandwidth and receivers with large buffers, SCTP has been shown to perform better than TCP in the presence of delay spikes.

SCTP over Mobile IP — The performance of SCTP in Mobile IP was investigated by Fu *et al.* [12]. Using ns-2 simulation, it was shown that the support of a large number of SCTP GapACK blocks in its SACK chunks can expedite error discovery and lost packet retransmission, and result in better performance than TCP-Reno and TCP-SACK. Simulation results have shown that the throughput improvement is especially prominent when network bandwidth is low.

The possibility of using SCTP multihoming to reduce the load on the home agent after a Mobile IP handover has been evaluated by Noonan *et al.* [13]. This can be achieved by assigning two IP addresses to the SCTP association at the mobile host: a permanent home address and a care-of address. The home address is kept unchanged throughout the whole life of the SCTP association, while the care-of address will be assigned by the current point of attachment to the network. As shown in Fig. 11, the home address is always used to locate the mobile host and as the primary destination during data transmission. Most packets sent from the correspondent node (CN) to the mobile host (MH) are forwarded by the home agent (HA) using tunneling.

The difference between the above scheme and standard Mobile IP is that the retransmitted packets (due to packets lost during handover) are sent to the care-of address directly instead of through the HA. Although the packets retransmitted directly via the care-of address do not constitute the majority of the packets sent by the CN (i.e., triangular routing is still not fully avoided), the reduction in the transmission delay of the retransmitted packets will improve the overall performance. When the packet error rate is 5 percent, the authors [13] reported a throughput increase of up to 41.4 percent from that of standard Mobile IP.

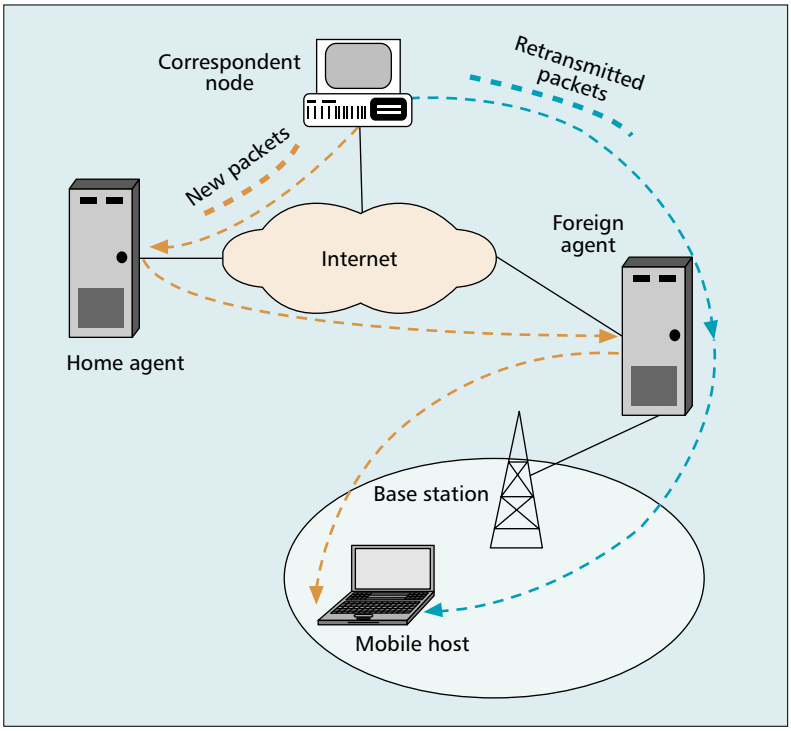

Figure 11. *Reducing Mobile IP home agent load using SCTP multihoming.*

Mobile Handover Based on SCTP Multihoming — Using SCTP's multihoming feature, researchers at the University of Oklahoma and elsewhere are investigating new handover schemes in mobile computing. A new scheme, called Transport Layer Seamless Handover (TraSH) [14], is proposed in this context, where the handover is accomplished at the transport layer without requiring any modification to the IP infrastructure.

A typical handover scenario based on TraSH and using SCTP's multihoming feature is illustrated in Fig. 12. Initially the MH is in the coverage of the previous IP domain's BS; as it enters the overlapping area while moving toward a base station (BS) belonging to a new IP domain, the MH can obtain a new IP address from the new domain, while the CN can still reach the MH using the previous IP address. The MH then notifies the CN about the availability of the new IP address. When the CN finds out that the MH's new IP address should be used as the primary destination address, it begins sending data through the MH's new IP address. This eliminates the infamous triangular routing problem encountered by Mobile IP. Note that the retransmitted packets from the CN in this scheme should also be directed to the MH's new IP address since the old IP address is very likely no more reachable because of the MH's movement.

In contrast to Mobile IP, there are no HAs or foreign agents in TraSH; the scheme, however, requires a location manager for the CN to locate the current position of the MH when a new association is to be set up by the CN. In addition to the University of Oklahoma researchers, similar schemes are being explored by groups at ETRI (Korea), Technical Univer-

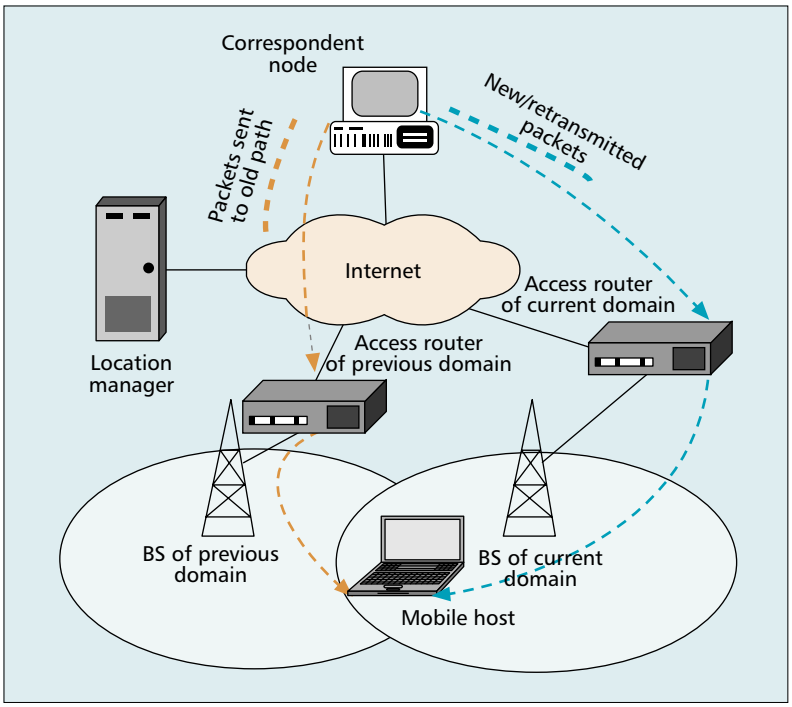

Figure 12. *Handover in TraSH [14] using SCTP's multihoming.*

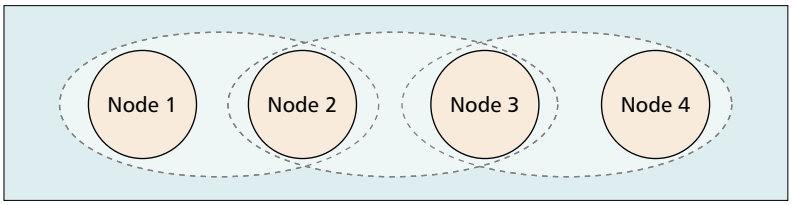

Figure 13. *Simulation topology for SCTP over multihop networks.*

sity of Berlin (Germany), Georgia Institute of Technology (United States), and Siemens. They are all based on the use of SCTP's multihoming feature to assist in data transfer during handover.

SCTP over Wireless Multihop Networks — Ye *et al.* [15] evaluated SCTP's performance in wireless multihop networks in the context of the IEEE 802.11 wireless LAN (WLAN). One important 802.11 parameter investigated was the request to send (RTS) threshold. Before sending data frames with sizes larger than the RTS threshold, the exchange of control frames — RTS/CTS, clear to send, sequence) is required. Generally speaking, a large RTS threshold will result in a high collision rate, whereas a small value will incur high signaling cost since virtually every data packet needs to use RTS/CTS signaling. Using the string simulation topology in Fig. 13 (where the dashed lines denote the radio coverage range), the authors have shown that the throughput of SCTP association degrades when the number of hops between the sender and receiver increases, mainly due to the hidden node and exposed node problems. The simulation results also show that when the hop count is less that three, the use of a low RTS threshold will reduce the collision occurring between SACK packets and RTS for DATA packets.

The *small window syndrome* (SWS) that happens when the SCTP receiver window is too small was also illustrated in the above paper. When SWS happens, the sender cannot get enough DupACKs to trigger a fast retransmit, and therefore must wait for a coarse timeout. Thus, the SCTP sender will experience a long idle period. By assuming that most of the data losses are caused by the medium access control (MAC) layer collision instead of wireless random loss or network congestion, the authors proposed to transmit the data packets with the lowest unreceived TSN (reported in the SACK Gap Block) during the idle period. This algorithm can partially overcome the SWS problem and speed up the error recovery caused by MAC collisions at the risk of pumping more data into an already congested network when the above assumption is not valid.

To summarize, the research endeavors in SCTP over wireless networks are aimed at exploiting SCTP's current capabilities, or designing new features that can make SCTP more suitable for wireless channels and mobile scenarios arising from third-generation (3G) and beyond wireless networks.

SCTP over Satellite Networks

Satellite networks are an indispensable part of the global Internet to provide broadband data, television, telephony, and navigation services. A number of satellite link characteristics, however, may limit the performance of transport protocols over satellite networks. Fu *et al.* [16] investigated and evaluated the SCTP features that can be exploited to increase SCTP's utilization of precious satellite network bandwidth, while at the same time preventing congestion collapse in the Internet. In addition to evaluating the SCTP features that *currently exist* in the TCP or its enhancements for satellite networks, they also investigated the *unique* SCTP features that can help SCTP to achieve high throughput in satellite networks. These unique features include multihoming, multistreaming, byte counting, large initial congestion window, and ECN. The results and recommendations provided in [16] can be used to increase SCTP throughput over satellite networks.

Available SCTP Products

A number of SCTP products are already available for research work and commercial use, as described in this section.

Reference Implementation

Implementing SCTP in the operating system kernel instead of in the user space opens the door to SCTP becoming a major transport protocol competing with TCP. The kernel reference implementation of SCTP in several popular UNIX operating systems, including BSD/OS 4.3, FreeBSD 4.7, NetBSD 1.6, and OpenBSD 3.2, are already available (www.sctp.org).

SCTP Patch to the ns-2 Simulator

Ns-2, a discrete event simulator targeted at networking research, has becomes one of the most popular research tools in networking. Ns-2 pro-

vides substantial support for simulation of transport, routing, and multicast protocols over wired and wireless (local area and satellite) networks. An SCTP patch to the ns-2 simulator has been contributed by a group at the University of Delaware [17]. The patch provides the main SCTP features specified in RFC 2960 [1], including multistreaming, multihoming, congestion control, and chunk bundling. This patch, which is still being developed, made it possible for various research groups to evaluate the performance of SCTP using ns-2.

Linux Kernel SCTP

This project is an open source implementation under GNU General Public License (GPL) to provide an SCTP module in a Linux kernel (lksctp.sourceforge.net). The LKSCTP project migrated to SourceForge in 2001 and now provides support for Linux kernel 2.6.0-test4. The implemented SCTP features in this project include association setup and takedown, sequenced delivery within streams, unordered messages within streams, data fragmentation and reassembly, congestion control, heartbeat, chunk bundling, packet validation, multistreaming, multihoming with failover, IPv4 and partial IPv6 addressing support, CRC32C checksum, and UDP-style socket application programming interface (API). A number of features, such as an ICMP error handler, IPv4-mapped-IPv6 address support, support of dynamically adding/deleting IP in an association, full IPv6 support, and support of a large number of simultaneously active associations, need to be added in future enhancements to the implementation.

Commercial Products

Most commercial products implementing SCTP are for signaling transport solutions. SCTP can transport various SS7 protocol types, such as MTP3, ISUP, SCCP, and TCAP. Some terminologies related to the SS7 signaling network is provided in Table 2.

In Fig. 14 we show the architecture where SCTP is used to transport TCAP messages. The application server process (ASP) is where the TCAP resides, and the SS7 SCCP-user adaptation (SUA) layer completes the adaptation from TCAP to SCTP. The transport of the SS7 signaling message over IP networks occurs between the ASP and signaling gateway (SG), following which the messages are transferred through the interworking function (IWF) to the SS7 network.

In the rest of this section we describe six currently available commercial SCTP products, and compare the different SCTP features supported by each product in Table 3.

APS-SCTP/T: The APS-SCTP/T software module from Adax is part of the Adax Protocol Software (APS) product family designed for signaling transport. APS-SCTP/T provides a signaling framework that enables IP telephony networks to achieve the same levels of service quality and reliability as those expected from the PSTN.

IP Transfer Point (ITP): Cisco implemented SCTP as part of its ITP product family and distributed it in the Cisco IOS Software Releases 12.2. ITP is a comprehensive product for transporting SS7 traffic over traditional time-division multiplexing (TDM) networks or advanced SS7-over-IP (SS7oIP) networks. In the SS7oIP mode, ITP connects to traditional SS7 signaling points or IP-enabled signaling points, and offloads the SS7 traffic to cost-efficient IP networks. The Cisco ITP is also capable of operating in a mode that mixes TDM and SS7oIP.

IN7: IN7 from Hewlett-Packard provides end-node connectivity to an SS7 network, as well as an SCCP relay point functionality. It also addresses next-generation IP networks with SS7 over IP capabilities such as media gateway controllers, media servers, signaling transfer point (STP), and signaling gateways.

HSS SIGTRAN Suite: The HSS SIGTRAN

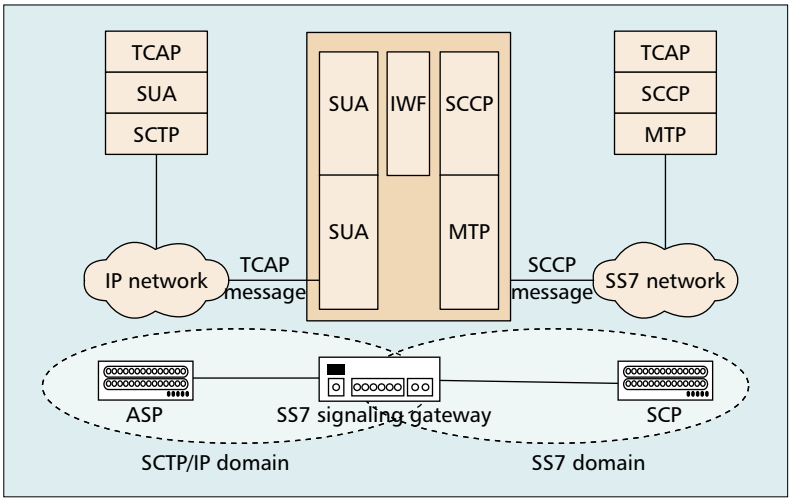

Figure 14. *Transport of TCAP messages using SCTP.*

Abbreviation	Meaning
SG	Signaling gateway
MG	Media gateway
MGC	Media gateway controller
SEP	Signaling endpoint
STP	Signaling transfer point
SCP	Service control point
MTP	Message transfer part (level 1, 2, 3)
SCCP	Signaling connection control part
TCAP	Transaction capabilities part
UAL	User adaptation layer
M2PA	MTP2-user peer-to-peer sdaptation
M2UA	MTP2 user adaptation
M3UA	MTP3 user adaptation layer
SUA	SCCP user adaptation
IWF	Interworking function

Table 2. *SS7 terminology.*

	Product Name					
Supported SCTP features	**Adax APS-SCTP/T**	**Cisco ITP**	**HP IN7**	**HSS SIGTRAN Suite**	**Performance Technologies SEGway**	**Ulticom SignalWare**
Multihoming	Yes	Yes	Yes	Yes	Yes	Yes
Multistreaming	Yes	Yes	Yes	Yes	Yes	Yes
Unordered delivery	Yes	Yes	Yes	Yes	Partial[a]	Yes
Cookie mechanism	Yes	Yes	Yes	Yes	Yes	Yes
Keep-alive heartbeat	Yes	Yes	Yes	Yes	Yes	Yes
Message fragmentation	Yes	Yes	Yes	Yes	Partial[a]	Yes
PMTU discovery	No	Yes	Yes	Yes	No	Yes
Socket API	No (STREAMs API used)	No information	Yes[b]	Yes	Proprietary C++ API	Partial
User adaption layers (M2PA, M3UA, SUA, M2UA)	M2PA, M3UA supported; SUA work in progress	Yes	M2PA: no; M3UA: partial[c]; SUA: partial[c]; M2UA: yes	Yes	M2PA: yes; M3UA: yes; SUA: partial	M2PA, M3UA, SUA: yes; M2UA will be available in mid-2004
SNMP support	Work in progress	No information	No	Yes	Yes	Yes

[a] Cannot handle messages that are both unordered and fragmented

[b] Support is at draft-ietf-tswg-sctpsocket-02 level

[c] Only on the signaling gateway side

Table 3. *A comparison of different commercial products.*

Suite from Hughes Software Systems implements the SCTP and user adaptation layers to enable transport of signaling information over packet-based networks. These software components can be integrated into end products, such as the media gateway controller, soft switch, signaling gateways, and IP-based service control point.

SEGway: SEGway signaling gateways from Performance Technologies provide a signaling bridge between traditional telephone networks and the growing packet-switched network architectures of today. When used in conjunction with softswitches, media gateways and application servers, signaling gateways can provide the call control functionality or service processing capabilities of traditional PSTN switches. The SEGway gateway can be installed as a standalone or an embedded product.

Signalware SIGTRAN and Signalware Gateway: These software stacks from Ulticom utilize an integrated streams-based SCTP implementation as the transport protocol to enable carrier-grade signaling solutions in next generation networks. Signalware SIGTRAN provides a platform to host signaling applications, while Signalware Gateway provides a bridge between traditional SS7 networks and IP networks.

The authors have made every effort to make the information in Table 3 accurate based on personal communications with each vendor and the brochures from the vendors' Web sites. Readers are encouraged to obtain the most up-to-date information about the products mentioned above at the Web sites of the respective vendors, as given in Table 4.

Implementation Considerations

SCTP is being implemented by an increasing number of groups. This section explains a number of issues that need to be considered by implementors.

Socket API

The socket API is probably the most commonly used API to access the services of TCP and UDP in the Internet. Providing a socket API in SCTP will help application developers who are familiar with TCP and UDP adapt to SCTP more quickly. The following are some of the goals that should be kept in mind while implementing a socket API in SCTP [18]:

- Maintain consistency with existing socket APIs
- Support TCP-style interface
- Support UDP-style interface

SCTP-Aware Applications

For some applications designed for SCTP, the standard TCP or UDP-style socket API is not enough to utilize the full power of SCTP. An implementation should provide an API to enable developers to specify some parameters peculiar to SCTP, such as the number of out-

going streams to set up during negotiation, number of outgoing streams that are unreliable, stream IDs used, and whether unordered delivery is allowed. Providing this kind of API will give application developers more flexibility in controlling the behavior of SCTP.

SCTP Implementor's Guide

The IETF published the SCTP implementor's guide containing a compilation of all defects found in SCTP, RFC 2960 [1]. This document may be thought of as a companion document to be used in the implementation of SCTP in order to clarify errors in the original SCTP RFC.

Issues and Challenges

SCTP is a relatively new protocol which still needs work to resolve a number of issues. Following are three major issues that need to be addressed.

Meeting the Reliability Requirements of SS7

An SS7 network has stringent reliability requirements, and there is much concern regarding functional specifications for overcoming linkset failures and congestion in the signaling network. Here, a linkset is defined as the set of all links between two signaling points in an SS7 network. Some of the major requirements are summarized below (from International Telecommunication Union — Telecommunication Standardization Union [ITU-T] Recommendation Q.706):

- The time needed to switch to another link when link failure occurs should be less than 800 ms.
- The availability of communication service between two signaling points should be at least 99.9988 percent, or a maximum downtime of 10 min/yr.
- No more than one in 10^7 messages may be lost due to failure in the message transfer part (MTP) layer.
- No more than one in 10^{10} messages may be delivered out of sequence to the user part due to failure in the MTP layer.

There are still no results available from large-scale experiments to verify that the current SCTP standard meets these requirements. Much simulation and experimental work still needs to be carried out in this regard.

Performance in Wireless Environments

SCTP is based on congestion control and retransmission schemes similar to those of TCP. SCTP and TCP are both designed with wireline environments in mind; they assume all losses are caused by congestion, and round-trip time (RTT) changes slowly and gradually. However, wireless mobile networks encounter higher bit error rates (BERs) and more frequent delay spikes [11] than wireline networks. This will cause SCTP to back off unnecessarily and result in poor throughput. Currently, there is no significant finding to solve this kind of problem.

APS-SCTP/T	http://www.adax.com
Cisco ITP	http://www.cisco.com
HP IN7	http://www.hp.com
HSS SIGTRAN Suite	http://www.hssworld.com
SEGway	http://www.pt.com
Ulticom SignalWare	http://www.ulticom.com

Table 4. *Product references.*

Dynamic Address Reconfiguration

The dynamic addition/deletion interface provides a graceful method to modify interfaces to an existing association when one of the endpoints in the association wishes to notify its peer that a new IP address will join the association or one of the old IP addresses will be out of service. It is important in mission-critical applications or mobile environments to support service reconfiguration without interrupting ongoing data transfers. This option, however, needs to define new chunk types and parameter types, and is still in the IETF draft stage.

The dynamic addition/deletion feature creates an extra security risk, the *traffic redirection attack*. An attacker A claims that its IP address should be added to an established association between H_1 and H_2, and further communication should be directed to this IP address. Therefore, an IP authentication process needs to be employed for address reconfiguration signaling. IPSec and IKE were not initially designed to support multihomed sessions efficiently. They need to create a separate database entry or perform a key negotiation for each pair of source/destination address, which is a waste of memory and time. The IETF IPSec working group has a relevant standard (RFC 3554), which provides several functional requirements and recommendations for using IPSec and IKE with SCTP multihoming.

Conclusions

The Stream Control Transmission Protocol is being standardized by the IETF as a reliable transport protocol to address a number of limitations of TCP in terms of transporting signaling messages over IP. Due to its attractive features, such as multihoming and multistreaming, SCTP has received much attention from the research community.

In this article we first summarize several key features of SCTP, then provide a comprehensive categorized survey of the most recent research activities to explore and improve SCTP. We also discuss the state of the art in commercially available products for SCTP users. Finally, with a view to stimulating further research, we identify several issues and challenges that still need to be addressed to improve the performance of SCTP and use it for various applications in the Internet.

There are still no results available from large-scale experiments to verify that the current SCTP standard meets these requirements. Much simulation and experimental work still needs to be carried out in this regard.

References

[1] R. Stewart *et al.*, "Stream Control Transmission Protocol," IETF RFC 2960, Oct. 2000.

SCTP is being standardized by IETF as a reliable transport protocol to address limitations of TCP. Due to its attractive features, SCTP has received much attention from the research community.

[2] R. Stewart and C. Metz, "SCTP: New Transport Protocol for TCP/IP," *IEEE Internet Comp.*, vol. 5, no. 6, Nov./Dec. 2001, pp. 64–69.

[3] A. L. Caro *et al.*, "SCTP: A Proposed Standard for Robust Internet Data Transport," *IEEE Comp.*, vol. 36, no. 11, Nov. 2003, pp. 56–63.

[4] A. Jungmaier, "Performance Evaluation of the Stream Control Transmission Protocol," *Proc. IEEE Conf. on High Perf. Switching and Routing*, Heidelberg, Germany, June 2000, pp. 141–48.

[5] R. Alamgir, M. Atiquzzaman, and W. Ivancic, "Effect of Congestion Control on the Performance of TCP and SCTP over Satellite Networks," *NASA Earth Sci. Tech. Conf.*, Pasadena, CA, June 2002.

[6] A. Jungmaier, E. P. Rathgeb, and M. Tuxen, "On the use of SCTP in Failover-scenarios," *Int'l. Conf. Info. Sys., Analysis and Synthesis*, Orlando, FL, July 2002, pp. 363–68.

[7] J. Iyengar *et al.*, "Preventing SCTP Congestion Window Overgrowth During Changeover," Internet draft,draft-iyengar-sctp-cacc-01txt.

[8] A. L. Caro *et al.*, "Improving Multimedia Performance over Lossy Networks via SCTP," *ATIRP 2001*, College Park, MD, Mar. 2001.

[9] M. Atiquzzaman and W. Ivancic, "Evaluation of SCTP Multistreaming over Wireless/Satellite Links," *12th Int'l. Conf. Comp. Commun. and Networks*, Dallas, TX, Oct. 2003, pp. 591–94.

[10] G. Camarillo, R. Kantola, and H. Schulzrinne, "Evaluation of Transport Protocols for the Session Initiation Protocol," *IEEE Network*, vol. 17, no. 5, Sept./Oct. 2003, pp. 40–46.

[11] S. Fu, M. Atiquzzaman, and W. Ivancic, "Effect of Delay Spike on SCTP, TCP Reno, and Eifel in a Wireless Mobile Environment," *11th Int'l. Conf. Comp. Commun. and Networks*, Miami, FL, Oct. 2002, pp. 575–78.

[12] S. Fu and M. Atiquzzaman, "Improving End-to-end Throughput of Mobile IP using SCTP," *Wksp. High Perf. Switching and Routing*, Torino, Italy, June 2003, pp. 171–76.

[13] J. Noonan, P. Perry, and J. Murphy, "A Study of SCTP Services in a Mobile IP Network," *IT&T Annual Conf.*, Ireland, Oct. 2002.

[14] S. Fu *et al.*, "TraSH: A Transport Layer Seamless Handover Scheme," Tech. rep., Comp. Sci., Univ. of OK, www.cs.ou.edu/~atiq, Nov. 2003.

[15] G. Ye, T. Saadawi, and M. Lee, "SCTP Congestion Control Performance in Wireless Multihop Networks," *MILCOM2002*, Anaheim, CA, Oct. 2002, pp. 934–39.

[16] S. Fu, M. Atiquzzaman, and W. Ivancic, "SCTP over Satellite Networks," *IEEE 18th Annual Wksp. Comp. Commun.*, Dana Point, CA, Oct. 2003, pp. 112–16.

[17] NS-2 SCTP Module Home Page, http://pel.cis.udel.edu

[18] R. Stewart and Q. Xie, *Stream Control Transmission Protocol (SCTP): A Reference Guide*, 1st ed., Addison Wesley, 2001.

BIOGRAPHIES

SHAOJIAN FU (sfu@ou.edu) received a B.E. degree in transportation engineering in 1997, and an M.E. degree in systems engineering in 2000, both from Northern Jiaotong University, Beijing, China. During 2000–2001 he worked with Bell Labs China, Lucent Technologies in the area of network surveillance and performance management. He is currently working toward a Ph.D. degree in the School of Computer Science, University of Oklahoma. His research interests are in the areas of wireless communications, mobility, transport protocols, and embedded reconfigurable computing.

MOHAMMED ATIQUZZAMAN [SM] (atiq@ou.edu) received M.Sc. and Ph.D. degrees in electrical engineering from the University of Manchester, England. Currently he is a professor in the School of Computer Science at University of Oklahoma. He is Co-Editor-in-Chief of *Computer Communications*, and serves on the editorial board of *IEEE Communications Magazine*, and the journals *Telecommunications Systems*, *Wireless and Optical Networks*, and *Real Time Imaging*. He has guest edited many special issues in various journals including feature topics on switching and traffic management for multimedia in *IEEE Communications Magazine*, Architecture, Protocol and Quality of Service in the *European Transactions on Telecommunications*, ATM Switching and ATM Networks in *International Journal of Computer Systems Science & Engineering*, and Parallel Computing on Clusters of Workstations in *Parallel Computing*. He was conference chair of the 2003 Workshop on High Performance Switching and Routing and SPIE's Quality of Service over Next-Generation Data Networks conference. He has also served on the technical program committee of many national and international conferences including IEEE INFOCOM, IEEE GLOBECOM, and IEEE Annual Conference on Local Computer Networks. His current research interests are in QoS for next-generation Internet, broadband networks, multimedia over high-speed networks, wireless and satellite networks, TCP/IP over ATM, multiprocessor systems, and image processing. He has taught many short courses to industry in the area of computer and telecommunication networking. His research has been supported by state and federal agencies like the National Science Foundation, NASA, U.S. Air Force, Ohio Board of Regents, and DITARD (Australia). He has over 130 refereed publications in the above areas, most of which can be accessed at http://www.cs.ou.edu/~atiq