

SCTP IN BATTLEFIELD NETWORKS

Phillip T. Conrad Gerard J. Heinz

Computer and Information Science Department
Temple University, Philadelphia, PA 19122 USA
Email: conrad@joda.cis.temple.edu, gheinz@unix.temple.edu

Armando L. Caro Jr. Paul D. Amer

Computer and Information Science Department
University of Delaware, Newark, DE 19716 USA
Email: {acarо,amer}@cis.udel.edu

John Fiore

School of Engineering and Applied Science
University of Pennsylvania, Philadelphia, PA 19104 USA
Email: jfiore@seas.upenn.edu

ABSTRACT

The Stream Control Transmission Protocol (SCTP) is a new Internet standards track transport layer protocol. SCTP was originally designed to transport PSTN signaling messages over IP networks, but is also capable of serving as a general purpose transport protocol. As such, SCTP provides an alternative that may be better able to satisfy the requirements of future battlefield networks than the traditional transport protocols, TCP and UDP. Unlike traditional transport protocols, SCTP allows multiple streams of messages within a single connection (or, in SCTP terminology, a single association). As the results in this paper show, this ability is particularly helpful in reducing latency for streaming multimedia in high loss environments. SCTP also provides features for multi-homing that may be helpful in high-mobility environments and additional security against denial-of-service attacks based on SYN flooding.^a

1 INTRODUCTION

The future battlefield environment will include mobile ad-hoc and wireless sensor nodes which deliver streaming real-time multimedia to end users. Traditional trans-

port protocols are not well suited to the relatively high loss rates of battlefield networks. New protocols which are designed to handle flexible service requirements can offer better QoS tradeoffs for future army networks.

Previous work has shown that traditional transport protocols, such as TCP and UDP, are less robust to packet loss than protocols incorporating partial order and partial reliability [3]. Previously, partial order and partial reliability were only implemented in experimental protocols. However, the telecommunication industry is strongly backing a new Internet standards track protocol, the *Stream Control Transmission Protocol (SCTP)* (RFC2690) [6], which incorporates partial order and has extensions for partial reliability in the Internet Draft stage.

Section 2 provides an overview of SCTP. The merits of a partially-ordered service will be covered in Section 3. Section 4 presents results from previous study of partially ordered vs. ordered service. These results indicate the nature of the expected performance gains SCTP may be able to provide. Section 5 ends the paper with some concluding remarks and ideas for future work.

2 SCTP OVERVIEW

SCTP is a reliable transport protocol operating over the IP network layer (i.e., a connectionless packet-switched network). SCTP emerged from the need for telecommunications companies to manage SS7 applications and services over an IP infrastructure. SS7 is a protocol suite

^aPrepared through collaborative participation in the Advanced Telecommunications/Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0002. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

for managing PSTNs and other telecommunication networks. The upper layers of SS7 are designed to operate over a circuit-switched control channel of the industry's TDM phone system network. Therefore, SCTP is oriented towards providing connection-oriented reliable message streams between communication endpoints.

While SCTP was originally designed for signaling transport, and much of the current work on SCTP in the IETF is centered around this application, the protocol designers recognize that SCTP *is capable of broader applications* [6]. Therefore, it is important to compare SCTP's transport service with that of UDP and TCP.

UDP is an unreliable, yet fast connectionless datagram service. Delay-sensitive messages find UDP suitable in that regard, but many applications cannot tolerate the fact that UDP does not provide ordered delivery, loss recovery, duplicate detection, congestion control, and flow control.

At the other extreme, TCP does provide a reliable transport service with the aforementioned features lacking in UDP. However, many applications find TCP too restrictive. Some of the drawbacks of TCP which applications have wanted to bypass are as follows:

- TCP is byte-stream-oriented, which means that applications are responsible for tracking message boundaries and using the push mechanism to ensure messages are transferred in reasonable time.
- TCP preserves order. While strict order-of-transmission data delivery is a restriction for some applications, for many other applications, unordered or partially ordered data delivery is sufficient. For such applications, TCP's strict ordering causes unnecessary delays.
- TCP does not transparently support multihomed hosts (see Section 2.1).
- TCP is vulnerable to denial-of-service attacks, which makes it a risky protocol to use in mission critical applications.

By comparison, SCTP is a reliable message-based connection-oriented transport protocol that provides the following services:

- acknowledged error-free non-duplicated transfer of user data,

- congestion avoidance behavior,
- data fragmentation to conform to discovered path MTU size,
- optional bundling of multiple user messages into a single SCTP packet,
- network-level fault tolerance through support of multihoming at either or both ends of an association,
- resistance to flooding and masquerade attacks, and
- sequenced delivery of user messages within multiple streams^b (i.e., partially-ordered data delivery), with an option for order-of-arrival delivery of individual user messages.

In the remainder of this paper, we present more information about the last three items in the list above. We first provide a brief description of SCTP's features for multi-homing and increased security. We then present some data concerning the benefit of providing multiple streams (i.e. partial order delivery).

2.1 SCTP multihoming features

Multihomed machines are those that have multiple IP addresses. Routers are always multihomed by necessity, however end systems that do not route packets can also have multiple IP addresses. For example, many hosts provide mission critical services and should therefore provide multiple IP addresses for redundant protection against single points of failure.

A TCP connection is defined by a 4-tuple: <remote-IP-address, remote-port, local-IP-address, local-port>. Suppose that an end system with two local IP addresses *A* and *B* has a TCP connection established to a remote system through local IP address *A*. If connectivity is lost on address *A*, the TCP connection will be aborted and must be reestablished, because TCP provides no capability to migrate an established connection from one IP address to another. Reestablishing a TCP connection is sub-optimal, because:

1. Transparent failover is lost, since the application must be involved.

^bThe term *stream* is used in SCTP to refer to a sequence of user messages that are to be delivered to the upper-layer protocol in order with respect to other messages within the same stream. This is in contrast to its usage in TCP, where it refers to a sequence of bytes. [6]

2. Connection setup adds the cost of one round-trip time.
3. The reestablished connection must begin with slow start.
4. Data buffered at the receiver would need to be discarded and retransmitted.

By contrast, SCTP allows a transport layer connection (an “association” in SCTP terminology) to be defined between a *set* of local IP addresses, and a *set* of remote IP addresses. Heartbeat packets (a form of “ping”) are used to monitor connectivity from the local host to each of the remote IP addresses. If connectivity is lost on the primary IP address being used for the association, the association will seamlessly fail over to an alternate IP address.

SCTP associations have potential to be useful for systems in battlefield networks where a given end system (e.g. a mobile command center) provides a mission critical service and may require seamless failover between multiple network attachments (e.g. satellite, wireless LAN, wired LAN).

2.2 SCTP mobility features

Battlefield networks may have highly mobile systems where IP address are continuously changing. In such situations, SCTP offers an alternative to Mobile IP solutions that require IP-in-IP encapsulation and forwarding through home agents. As mentioned in Section 2.1, SCTP allows associations to be defined between sets of local and remote IP addresses. This feature, in addition to an extension for dynamically adding and deleting IP addresses from these sets, allow mobile systems to hand-off smoothly and transparently. Each time a host moves into a new subnet and obtains a new IP addresses, it can add the new IP address to the its set and delete the old one. Meanwhile, all other transport layer activities remain the same, and the application never gets disrupted.

2.3 Resistance to attacks

Both TCP and SCTP provide connection-oriented data delivery, however, it has been found that TCP’s connection establishment process is particularly vulnerable to denial-of-service attacks[2].

2.3.1 TCP Connection Establishment

Before data is transmitted, TCP hosts must first establish a connection through a 3-way handshake:

1. The initiator of the connection sends a SYN packet.
2. The passive end of the connection allocates resources dedicated to the connection, and then replies with a SYN-ACK.
3. The initiator then acknowledges the SYN-ACK with its own acknowledgment.

TCP implementations limit the number of connections that can be in the SYN_RCVD state, i.e. connections in which the passive host has: received a SYN, allocated resources, and replied with a SYN-ACK, but has not yet received the final ACK of the 3-way handshake.

An attacking system can send a series of SYN packets (usually from forged IP addresses) until the victim has reached its limit of half-open connections, thus leaving it unable to accept any new incoming connections. Although the half-open connections are purged after a period of time, the attacker can send SYNs faster than the timeout period, often at little cost to itself.

This form of attack, known as SYN-flooding, has no widely accepted defense that can be implemented entirely within TCP. At first glance, it may seem as if raising (or eliminating) the limit of half-open connections might help, however this only exacerbates the problem. Because the attacker can send SYNs at will, eliminating the limit would not only render the victim unable to receive new incoming connections, but would also exhaust its supply of memory.

In the Internet, security from this attack rests almost solely on internet service providers’ willingness to restrict outgoing IP packets with forged source addresses [13]. Working with ISPs to filter out forged packets may be a practical “partial solution” for the Internet, however, in a battlefield network where an enemy may have the means to forge packets, an alternative solution is needed.

2.3.2 SCTP’s Defense

The four-way handshake used in SCTP’s association setup incorporates the exchange of a Message Authentication Code (MAC), generated by the passive host

through the use of a cryptographic hash algorithm. Because the passive host does not commit any resources to the association until the MAC is exchanged, the target system is much less vulnerable to denial-of-service attacks from spoofed addresses. Additionally, the MAC helps prevent against replay attacks.

In the following example of a typical SCTP association establishment, active host *A* attempts to connect with passive host *P*:

1. *A* dedicates resources to the connection and sends to *P* an INIT packet.
2. *P* replies with an INIT-ACK containing a cookie comprised of:
 - The transmission control block (the state that should be setup for the new connection only *if and when* it is established)
 - The time that the cookie was generated.
 - The time that the cookie expires.
 - The MAC based on a implementation dependent private key hash algorithm [9].
3. *A* sends a COOKIE-ECHO chunk containing a copy of the cookie sent in the INIT-ACK, and optionally piggybacks user data.
4. If the COOKIE-ECHO contains a valid cookie, *P* dedicates resources to the association, and sends a COOKIE-ACK with optionally piggybacked user data. If the cookie is invalid, however, *P* discards the packet.

If, in the above example, *A* had sent an INIT with a spurious IP address, *P* would have sent its INIT-ACK to the spurious address. Since *P* would not have saved state or allocated any resources for the association until it received a valid MAC, *A*'s ability to perform a denial-of-service attack is curtailed.

Traditionally MACs are used between two separate parties which share a private key. In SCTP, however, they are used to ensure that a state cookie being echoed back to the passive host has not been forged. Because *P* generates the MAC through a secure hash function based on its private key, it becomes impossible for *A* to send a counterfeit MAC which *P* will accept, unless *A* can break *P*'s encryption scheme. As a result, SCTP offers enhanced protection against replay and denial-of-service attacks.

Though current SCTP implementations use MD5, the SCTP specification does not endorse any particular secure hash algorithm. This allows protocol implementors to use the strongest, most efficient algorithms at their disposal. The fact that the hash is generated by and verified by the same host (the passive host) eliminates problems associated with interoperability. In the event that a particular algorithm is found to be cryptographically insecure, it may be replaced with a new one with minimal effort. Similarly, SCTP implementations with military encryption may seamlessly integrate with other military and/or civilian SCTP implementations without weakening security.

3 BENEFITS OF PARTIALLY-ORDERED SERVICE

Since traditional protocols (UDP and TCP) offer only the extremes (unordered and strictly ordered service, respectively), application developers with needs in between these extremes are faced with a dilemma. If TCP is chosen, unnecessary performance penalties must be paid. If UDP is chosen, developers must build their own transport protocol over UDP to provide the exact services that are needed.

A flexible transport protocol offering a partially-ordered service is ideal for applications that need flexible control over the ordering of individual elements. A partially-ordered service allows independent Application Data Units (ADUs) to be processed out of sequence, while still maintaining ordering constraints between ADUs that require sequential processing. Such a service is essential for balancing various QoS parameters required by the application, without having to implement a custom protocol for each new application.

Analysis and simulation have shown that partially-ordered transport service can provide improvements in throughput, delay, and buffer utilization for a normalized time-scale [10, 11, 12].

The development of an innovative transport protocol that provides a partially-ordered service (POCv2) and an application which uses the protocol (ReMDoR), demonstrated that the theoretical advantages of a partially-ordered service can be achieved in practice [3]. A subset of these results are presented in section 4.

Diot et al. [5] also investigated PO services, which they refer to as "out-of-sequence delivery". Their results showed that out-of-sequence delivery is beneficial when the following relationships hold:

$$\left(T - \frac{R}{N}\right) < P \leq T$$

$$10T \leq R/2$$

where

- T is packet transmission time (packet length divided by bitrate)
- R is round trip time
- N is $1/\text{LossRate}$
- P is packet processing time

Although the results presented in [5] used experimental data, the models used were too simplified to resemble real network communication. For example, TCP-friendly connection-oriented transport protocols do not have a fixed window size as used in their experiments.

Early in 2000, the IETF began standardizing a new TCP-friendly transport protocol (SCTP) to provide a PO transport service. Given that SCTP is a standards track protocol rather than an experimental academic protocol, it represents the next logical step for investigating PO transport services. Additionally, the first FreeBSD kernel implementation of SCTP is near completion. Experimenting with a kernel level implementation will give us even more realistic experimental results than with the application level implementation used in [3]. Therefore, SCTP will be the avenue for fully understanding PO services and the applications which use them.

4 EMPIRICAL RESULTS

Reference [3] includes extensive results from sixteen experiments comparing the performance of multimedia document retrieval over reliable transport services providing unordered, partially-ordered, and ordered delivery. In this section, we present results from one of these experiments that highlight the benefits we expect from SCTP's partially-ordered service.

Experiment R1 from [3] compares ordered/reliable service to partially-ordered/reliable service for retrieval of a document with eight images presented in parallel. Experiment R1 uses the standard GIF compression technique rather than a specific network conscious com-

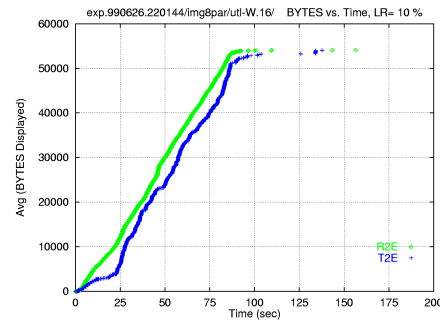


Figure 1: Performance graph: Experiment R1.1 (9.6kbps PPP link at 10% loss)

pression technique [7, 8]. The GIF format requires ordered/reliable delivery for each image, so unordered service cannot be used. However, partially-ordered service can be used because the data for each image can be interleaved in eight parallel streams. This experiment uses the ReMDoR application [1] in addition to the UTL and Lossy Router tools developed by the Protocol Engineering Laboratory at the University of Delaware [4].

The hypothesis for this experiment is that for all loss rates $> 0\%$, partially-ordered/reliable (PO/R) service provides, on average, better progressive display for parallel GIF images than ordered/reliable (O/R) service.

Due to space limitations, this paper will present only a subset of the experimental results of R1, which we will refer to as Experiment R1.1. Experiment R1.1 illustrates the performance of PO/R service versus O/R service. To simulate the low bitrate and high loss associated with battlefield networks such as SINCGARS, a PPP connection at 9.6Kbps was used, and 10% packet loss was introduced. Figure 1 shows an average performance graph for Experiment R1.1. From this graph, we conclude that this set of experimental data supports the hypothesis stated above.

In addition, the results of the complete Experiment R1 presented in [3] show the following:

- At 0% loss, PO/R and O/R have virtually identical performance.
- While both PO/R and O/R experience worse performance as the loss rate increases, the performance of PO/R degrades more slowly than that of O/R.
- At nearly every point in time, on average, PO/R

provides more data (show both in bytes and pixels) to the end-user.

To provide an end-user perspective, Figure 2 shows the difference between PO/R and O/R performance at a few sample points for 10% loss. As can clearly be seen, at each of these points, partially-ordered service provides better performance than totally-ordered service. While human factor studies (which we suggest as future work) would be necessary to establish this scientifically, we hypothesize that the initial delivery of at least a few pixels will prove to be highly correlated with user satisfaction. Seeing at least some progress provides hope to the user, while seeing a screen that does not change for a long period of time (especially a blank one) can be discouraging.

5 CONCLUSION AND FUTURE WORK

Previously, the merits of partial order were theoretical advantages proven by analytical models and simulation only. The results in this paper show, however, that applications requiring such a transport service do actually achieve the benefits in practice. Until recently, partial order has only been implemented in experimental transport protocols for research purposes, but the telecommunication industry is now interested the new and upcoming transport protocol which supports partial order, namely SCTP.

The researchers of the University of Delaware's Protocol Engineering Laboratory in collaboration with Temple's NetLab researchers are investigating SCTP. We plan to evaluate the performance of SCTP when used for streaming real-time multimedia and other partial order benefiting applications. Additionally, we plan to investigate the mobility benefits gained by exploiting SCTP's multihoming features. Our goal is to develop an integrated multimedia transport protocol to fit the needs of future army networks.

6 DISCLAIMER

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the Army Research Laboratory or the U.S. Government.

7 REFERENCES

- [1] A. Caro. Remdor 2.0: Remote multimedia document retrieval over partially-ordered, partially-reliable transport protocols, May 1998. BS Thesis, CIS Dept., University of Delaware.
- [2] Computer Emergency Response Team (CERT). TCP SYN flooding and IP spoofing attacks (CA-1996-21). Carnegie Mellon University, Pittsburgh, PA., September 1996.
- [3] P. Conrad. Order, reliability, and synchronization in transport layer protocols for multimedia document retrieval, 2000. PhD Dissertation, CIS Dept. University of Delaware.
- [4] P. Conrad, P. Amer, M. Taube, G. Sezen, S. Iren, and A. Caro. Testing environment for innovative transport protocols. In *MILCOM '98*, Bedford, MA, October 1998.
- [5] C. Diot and F. Gagnon. Impact of out-of-sequence processing on data transmission performance. *Computer Networks*, 31(5):475–492, March 1999.
- [6] R. Stewart et al. Stream control transmission protocol. RFC 2960, October 2000.
- [7] S. Iren. Network-conscious image compression, 1999. PhD Dissertation, CIS Dept., University of Delaware.
- [8] S. Iren, P. Amer, A. Caro, G. Sezen, M. Taube, and P. Conrad. Network-conscious compressed image transmission over battlefield networks. In *MILCOM '98*, Bedford, MA, October 1998.
- [9] Canetti Krawczyk, Bellare. HMAC: Keyed-hashing for message authentication. Request for comments, Internet Engineering Task Force, February 1997.
- [10] R. Marasli. Partially ordered and partially reliable transport protocols: Performance analysis, 1997. PhD Dissertation, CIS Dept., University of Delaware.
- [11] R. Marasli, P. Amer, and P. Conrad. Retransmission-based partially reliable services: An analytic model. In *IEEE INFOCOM*, San Fransisco, CA, March 1996.
- [12] R. Marasli, P. Amer, and P. Conrad. An analytic model of partially ordered transport service. *Computer Networks and ISDN Systems*, 29(6):675–699, May 1997.
- [13] Schuba, Krsul, Kuhn, Spafford, Sundaram, and Zamboni. Analysis of a denial of service attack on TCP. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy.*, pages 208–223, Los Alamitos, CA, May 1997.

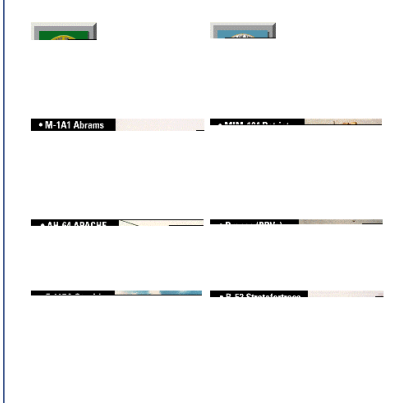
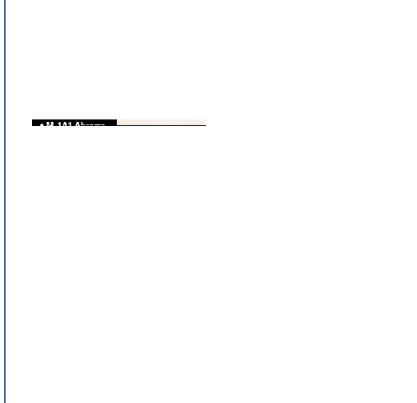




	Partially-ordered service (R2E)	Ordered service (T2E)
25 seconds	 <p>avg. 14447 (13508 pixels shown)</p>	 <p>avg. 1530 (1535 pixels shown)</p>
50 seconds	 <p>avg. 53767 (53294 pixels shown)</p>	 <p>avg. 40437 (41745 pixels shown)</p>
75 seconds	 <p>avg. 87345 (86446 pixels shown)</p>	 <p>avg. 75665 (76506 pixels shown)</p>

Figure 2: Screenshot: Experiment R1.1 (9.6kbps PPP link at 10% loss)