Network Working Group
Request for Comments: 3286
Category: Informational

L. Ong
Ciena Corporation
J. Yoakum
Nortel Networks

May 2002

An Introduction to the Stream Control Transmission Protocol (SCTP)

Status of this Memo

Copyright Notice

Abstract

This document provides a high level introduction to the capabilities supported by the Stream Control Transmission Protocol (SCTP). It is intended as a guide for potential users of SCTP as a general purpose transport protocol.

1. Introduction

The Stream Control Transmission Protocol (SCTP) is a new IP transport protocol, existing at an equivalent level with UDP (User Datagram Protocol) and TCP (Transmission Control Protocol), which provide transport layer functions to many Internet applications. SCTP has been approved by the IETF as a Proposed Standard [1]. The error check algorithm has since been modified [2]. Future changes and updates will be reflected in the IETF RFC index.

Like TCP, SCTP provides a reliable transport service, ensuring that data is transported across the network without error and in sequence. Like TCP, SCTP is a session-oriented mechanism, meaning that a relationship is created between the endpoints of an SCTP association prior to data being transmitted, and this relationship is maintained until all data transmission has been successfully completed.

Unlike TCP, SCTP provides a number of functions that are critical for telephony signaling transport, and at the same time can potentially benefit other applications needing transport with additional performance and reliability. The original framework for the SCTP definition is described in [3].

2. Basic SCTP Features

SCTP is a unicast protocol, and supports data exchange between exactly 2 endpoints, although these may be represented by multiple IP addresses.

SCTP provides reliable transmission, detecting when data is discarded, reordered, duplicated or corrupted, and retransmitting damaged data as necessary. SCTP transmission is full duplex.

SCTP is message oriented and supports framing of individual message boundaries. In comparison, TCP is byte oriented and does not preserve any implicit structure within a transmitted byte stream without enhancement.

SCTP is rate adaptive similar to TCP, and will scale back data transfer to the prevailing load conditions in the network. It is designed to behave cooperatively with TCP sessions attempting to use the same bandwidth.

3. SCTP Multi-Streaming Feature

The name Stream Control Transmission Protocol is derived from the multi-streaming function provided by SCTP. This feature allows data to be partitioned into multiple streams that have the property of independently sequenced delivery, so that message loss in any one stream will only initially affect delivery within that stream, and not delivery in other streams.

In contrast, TCP assumes a single stream of data and ensures that delivery of that stream takes place with byte sequence preservation. While this is desirable for delivery of a file or record, it causes additional delay when message loss or sequence error occurs within the network. When this happens, TCP must delay delivery of data until the correct sequencing is restored, either by receipt of an out-of-sequence message, or by retransmission of a lost message.

For a number of applications, the characteristic of strict sequence preservation is not truly necessary. In telephony signaling, it is only necessary to maintain sequencing of messages that affect the same resource (e.g., the same call, or the same channel). Other messages are only loosely correlated and can be delivered without having to maintain overall sequence integrity.

Another example of possible use of multi-streaming is the delivery of multimedia documents, such as a web page, when done over a single session. Since multimedia documents consist of objects of different sizes and types, multi-streaming allows transport of these components to be partially ordered rather than strictly ordered, and may result in improved user perception of transport.

At the same time, transport is done within a single SCTP association, so that all streams are subjected to a common flow and congestion control mechanism, reducing the overhead required at the transport level.

SCTP accomplishes multi-streaming by creating independence between data transmission and data delivery. In particular, each payload DATA "chunk" in the protocol uses two sets of sequence numbers, a Transmission Sequence Number that governs the transmission of messages and the detection of message loss, and the Stream ID/Stream Sequence Number pair, which is used to determine the sequence of delivery of received data.

This independence of mechanisms allows the receiver to determine immediately when a gap in the transmission sequence occurs (e.g., due to message loss), and also whether or not messages received following the gap are within an affected stream. If a message is received within the affected stream, there will be a corresponding gap in the Stream Sequence Number, while messages from other streams will not show a gap. The receiver can therefore continue to deliver messages to the unaffected streams while buffering messages in the affected stream until retransmission occurs.

4. SCTP Multi-Homing Feature

Another core feature of SCTP is multi-homing, or the ability for a single SCTP endpoint to support multiple IP addresses. The benefit of multi-homing is potentially greater survivability of the session in the presence of network failures. In a conventional single-homed session, the failure of a local LAN access can isolate the end system, while failures within the core network can cause temporary unavailability of transport until the IP routing protocols can reconverge around the point of failure. Using multi-homed SCTP, redundant LANs can be used to reinforce the local access, while various options are possible in the core network to reduce the dependency of failures for different addresses. Use of addresses with different prefixes can force routing to go through different carriers, for example, route-pinning techniques or even redundant core networks can also be used if there is control over the network architecture and protocols.

In its current form, SCTP does not do load sharing, that is, multi-homing is used for redundancy purposes only. A single address is chosen as the "primary" address and is used as the destination for all DATA chunks for normal transmission. Retransmitted DATA chunks use the alternate address(es) to improve the probability of reaching the remote endpoint, while continued failure to send to the primary address ultimately results in the decision to transmit all DATA chunks to the alternate until heartbeats can reestablish the reachability of the primary.

To support multi-homing, SCTP endpoints exchange lists of addresses during initiation of the association. Each endpoint must be able to receive messages from any of the addresses associated with the remote endpoint; in practice, certain operating systems may utilize available source addresses in round robin fashion, in which case receipt of messages from different source addresses will be the normal case. A single port number is used across the entire address list at an endpoint for a specific session.

In order to reduce the potential for security issues, it is required that some response messages be sent specifically to the source address in the message that caused the response. For example, when the server receives an INIT chunk from a client to initiate an SCTP association, the server always sends the response INIT ACK chunk to the source address that was in the IP header of the INIT.

5. Features of the SCTP Initiation Procedure

The SCTP Initiation Procedure relies on a 4-message sequence, where DATA can be included on the 3rd and 4th messages of the sequence, as these messages are sent when the association has already been validated. A "cookie" mechanism has been incorporated into the sequence to guard against some types of denial of service attacks.

5.1 Cookie Mechanism

The "cookie" mechanism guards specifically against a blind attacker generating INIT chunks to try to overload the resources of an SCTP server by causing it to use up memory and resources handling new INIT requests. Rather than allocating memory for a Transmission Control Block (TCB), the server instead creates a Cookie parameter with the TCB information, together with a valid lifetime and a signature for authentication, and sends this back in the INIT ACK. Since the INIT ACK always goes back to the source address of the INIT, the blind attacker will not get the Cookie. A valid SCTP client will get the Cookie and return it in the COOKIE ECHO chunk, where the SCTP server can validate the Cookie and use it to rebuild the TCB. Since the server creates the Cookie, only it needs to know the format and secret key, this is not exchanged with the client.

Otherwise, the SCTP Initiation Procedure follows many TCP conventions, so that the endpoints exchange receiver windows, initial sequence numbers, etc. In addition to this, the endpoints may exchange address lists as discussed above, and also mutually confirm the number of streams to be opened on each side.

5.2 INIT Collision Resolution

Multi-homing adds to the potential that messages will be received out of sequence or with different address pairs. This is a particular concern during initiation of the association, where without procedures for resolving the collision of messages, you may easily end up with multiple parallel associations between the same endpoints. To avoid this, SCTP incorporates a number of procedures to resolve parallel initiation attempts into a single association.

6. SCTP DATA Exchange Features

DATA chunk exchange in SCTP follows TCP’s Selective ACK procedure. Receipt of DATA chunks is acknowledged by sending SACK chunks, which indicate not only the cumulative Transmission Sequence Number (TSN) range received, but also any non-cumulative TSNs, implying gaps in the received TSN sequence. Following TCP procedures, SACKs are sent using the "delayed ack" method, normally one SACK per every other received packet, but with an upper limit on the delay between SACKs and an increase to once per received packet when there are gaps detected.

Flow and Congestion Control follow TCP algorithms. The advertised receive window indicates buffer occupancy at the receiver, while a per-path congestion window is maintained to manage the packets in flight. Slow start, Congestion avoidance, Fast recovery and Fast retransmit are incorporated into the procedures as described in RFC 2581, with the one change being that the endpoints must manage the conversion between bytes sent and received and TSNs sent and received, since TSN is per chunk rather than per byte.

The application can specify a lifetime for data to be transmitted, so that if the lifetime has expired and the data has not yet been transmitted, it can be discarded (e.g., time-sensitive signaling messages). If the data has been transmitted, it must continue to be delivered to avoid creating a hole in the TSN sequence.

7. SCTP Shutdown Features

SCTP Shutdown uses a 3-message procedure to allow graceful shutdown, where each endpoint has confirmation of the DATA chunks received by the remote endpoint prior to completion of the shutdown. An Abort procedure is also provided for error cases when an immediate shutdown must take place.

Note that SCTP does not support the function of a "half-open" connection as can occur in TCP, when one side indicates that it has no more data to send, but the other side can continue to send data indefinitely. SCTP assumes that once the shutdown procedure begins, both sides will stop sending new data across the association, and only need to clear up acknowledgements of previously sent data.

8. SCTP Message Format

The SCTP Message includes a common header plus one or more chunks, which can be control or data. The common header has source and destination port numbers to allow multiplexing of different SCTP associations at the same address, a 32-bit verification tag that guards against insertion of an out-of-date or false message into the SCTP association, and a 32-bit checksum (this has been modified to use the CRC-32c polynomial [2]) for error detection.

Each chunk includes chunk type, flag field, length and value. Control chunks incorporate different flags and parameters depending on the chunk type. DATA chunks in particular incorporate flags for control of segmentation and reassembly, and parameters for the TSN, Stream ID and Stream Sequence Number, and a Payload Protocol Identifier.

The Payload Protocol ID has been included for future flexibility. It is envisioned that the functions of protocol identification and port number multiplexing will not be as closely linked in the future as they are in current usage. Payload Protocol ID will allow the protocol being carried by SCTP to be identified independent of the port numbers being used.

The SCTP message format naturally allows support of bundling of multiple DATA and control chunks in a single message, to improve transport efficiency. Use of bundling is controllable by the application, so that bundling of initial transmission can be prohibited. Bundling will naturally occur on retransmission of DATA chunks, to further reduce any chance of congestion.

9. Error Handling

9.1 Retransmission

Retransmission of DATA chunks occurs from either (a) timeout of the retransmission timer; or (b) receipt of SACKs indicating the DATA chunk has not been received. To reduce the potential for congestion, the rate of retransmission of DATA chunks is limited. The retransmission timeout (RTO) is adjusted based on estimates of the round trip delay and backs off exponentially as message loss increases.

In an active association with fairly constant DATA transmission, SACKs are more likely to cause retransmission than the timeout. To reduce the chance of an unnecessary retransmission, a 4 SACK rule is used, so that retransmission only occurs on receipt of the 4th SACK that indicates that the chunk is missing. This is intended to avoid retransmits due to normal occurrences such as packets received out of sequence.

9.2 Path Failure

A count is maintained of the number of retransmissions to a particular destination address without successful acknowledgement. When this count exceeds a configured maximum, the address is declared inactive, notification is given to the application, and the SCTP begins to use an alternate address for the sending of DATA chunks.

Also, Heartbeat chunks are sent periodically to all idle destinations (i.e., alternate addresses), and a counter is maintained on the number of Heartbeats sent to an inactive destination without receipt of a corresponding Heartbeat Ack. When this counter exceeds a configured maximum, that destination address is also declared inactive.

Heartbeats continue to be sent to inactive destination addresses until an Ack is received, at which point the address can be made active again. The rate of sending Heartbeats is tied to the RTO estimation plus an additional delay parameter that allows Heartbeat traffic to be tailored according to the needs of the user application.

9.3 Endpoint Failure

A count is maintained across all destination addresses on the number of retransmits or Heartbeats sent to the remote endpoint without a successful Ack. When this exceeds a configured maximum, the endpoint is declared unreachable, and the SCTP association is closed.

10. API

The specification includes a model of the primitives exchanged between the application and the SCTP layer, intended as informational material rather than a formal API statement. A socket-based API is being defined to simplify migration of TCP or UDP applications to the use of SCTP.

11. Security Considerations

In addition to the verification tag and cookie mechanisms, SCTP specifies the use of IPSec if strong security and integrity protection is required. The SCTP specification does not itself define any new security protocols or procedures.

Extensions to IPSec are under discussion to reduce the overhead required to support multi-homing. Also, work is in progress on the use of Transport Layer Security (TLS) over SCTP [4].

12. Extensions

The SCTP format allows new chunk types, flags and parameter fields to be defined as extensions to the protocol. Any extensions must be based on standard agreements within the IETF, as no vendor-specific extensions are supported in the protocol.

Chunk Type values are organized into four ranges to allow extensions to be made with a pre-defined procedure for responding if a new Chunk Type is not recognized at the remote endpoint. Responses include: whole packet discard; packet discard with reporting; ignoring the chunk; ignoring with reporting. Similar pre-defined responses are specified for unrecognized Parameter Type values.

Chunk Parameter Type values are in principle independent ranges for each Chunk Type. In practice, the values defined in the SCTP specification have been coordinated so that a particular parameter type will have the same Chunk Parameter Type value across all Chunk Types. Further experience will determine if this alignment needs to be maintained or formalized.

13. Informative References

[1] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[2] Stewart, Sharp, et. al., "SCTP Checksum Change", Work in Progress.

[3] Ong, L., Rytina, I., Garcia, M., Schwarzbauer, H., Coene, L., Lin, H., Juhasz, I., Holdrege, M. and C. Sharp, "Framework Architecture for Signaling Transport", RFC 2719, October 1999.

[4] Jungmeier, Rescorla and Tuexen, "TLS Over SCTP", Work in Progress.

14. Authors' Addresses

Lyndon Ong
Ciena Corporation
10480 Ridgeview Drive
Cupertino, CA 95014

EMail: lyong@ciena.com

John Yoakum
Emerging Opportunities
Nortel Networks

EMail: yoakum@nortelnetworks.com

15. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.