Internet Engineering Task Force SIP WG
Internet Draft
draft-rosenberg-sip-sctp-01.txt
May 18, 2001
Expires: November, 2001

Rosenberg/Schulzrinne/Camarillo
dynamicsoft,Columbia U.,Ericsson

SCTP as a Transport for SIP

Abstract

This document specifies a mechanism for usage of SCTP (the Stream Control Transmission Protocol) as the transport between SIP entities. SCTP is a new protocol which provides several features that may prove beneficial for transport between SIP entities which exchange a large amount of messages, including gateways and proxies. As SIP is transport independent, support of SCTP is a relatively straightforward process, nearly identical to support for TCP.

1 Introduction

The Stream Control Transmission Protocol (SCTP) [1] has been designed as a new transport protocol for the Internet (or intranets), at the same layer as TCP and UDP. SCTP has been designed with the transport of legacy SS7 signaling messages in mind. We have observed that many of the features designed to support transport of such signaling are also useful for the transport of SIP (the Session Initiation Protocol) [2], which is used to initiate and manage interactive sessions on the Internet.

SIP itself is transport-independent, and can run over any reliable or unreliable message or stream transport. However, procedures are only defined for transport over UDP and TCP. In order to encourage experimentation and evaluation of the appropriateness of SCTP for SIP transport, this document defines transport of SIP over SCTP.

Note that this is not a proposal that SCTP be adopted as the primary or preferred transport for SIP; substantial evaluation of SCTP, deployment, and experimentation can take place before that happens. This draft is targeted at encouraging such experimentation by enabling it in SIP.

2 Potential Benefits

Coene et. al. present some of the key benefits of SCTP [3]. We summarize some of these benefits here and analyze how they relate to SIP:

2.1 Advantages over UDP

All the advantages that SCTP has over UDP regarding SIP transport are also shared by TCP. Below there is a list of the general advantages that a connection-oriented transport protocol such as TCP or SCTP has over a connection-less transport protocol such as UDP.

Fast Retransmit: SCTP can quickly determine the loss of a packet, as a result of its usage of SACK and a mechanism which sends SACK messages faster than normal when losses are detected. The result is that losses of SIP messages can be detected much faster than when SIP is run over UDP (detection will take at least 500ms, if not more). Note that TCP SACK does exist as well, and TCP also has a fast retransmit option. Over an existing connection, this results in faster call setup times under conditions of packet loss, which is very desirable. This is probably the most significant advantage to SCTP for SIP transport.

Congestion Control: SCTP maintains congestion control over the entire connection. For SIP, this means that the aggregate rate of messages between two entities can be controlled. When SIP is run over TCP, the same advantages are afforded. However, when run over UDP, SIP provides less effective congestion control. That is because congestion state

(measured in terms of the UDP retransmit interval) is computed on a transaction by transaction basis, rather than across all transactions. Congestion control performance is thus similar to opening N parallel TCP connections, as opposed to sending N messages over 1 TCP connection.

Transport layer fragmentation: SCTP and TCP provide transport layer fragmentation. If a SIP message is larger than the MTU size it is fragmented at the transport layer. When UDP is used fragmentation occurs at the IP layer. IP fragmentation increases the likelihood of having packet losses and make it difficult (when not impossible) NAT and firewall traversal. This feature will become important if the size of SIP messages grows dramatically.

2.2 Advantages over TCP

We have shown the advantages of SCTP and TCP over UDP. We now analyze the advantages of SCTP over TCP.

Head of the Line: SCTP is message based as opposed to TCP that is stream based. This allows SCTP to separate different signalling messages at the transport layer. TCP just understands bytes. Assembling received bytes to form signalling messages is performed at the application layer. Therefore, TCP always delivers an ordered stream of bytes to the application. On the other hand, SCTP can deliver signalling messages to the application as soon as they arrive (when using the unordered service). The loss of a signalling message does not affect the delivery of the rest of the messages. This avoids the head of line blocking problem in TCP, which occurs when multiple higher layer connections are multiplexed within a single TCP connection. A SIP transaction can be considered an application layer connection. Between proxies there are multiple transactions running. The loss of a message in one transaction should not adversely effect the ability of a different transaction to send a message. Thus, if SIP is run between entities with many transactions occurring in parallel, SCTP can provide improved performance over SIP over TCP (but not SIP over UDP; but SIP over UDP is not ideal from a congestion control standpoint, see above).

Easier Parsing: Another advantage of message based protocols such as SCTP and UDP over stream based protocols such as TCP is that they allow easier parsing of messages at the application layer. There is not need of establishing boundaries (typically using Content-Length headers) between

different messages. However, this advantage is almost negligible.

Multihoming: An SCTP connection can be associated with multiple IP addresses on the same host. Data is always sent over one of the addresses, but should it become unreachable, data sent to one can migrate to a different address. This improves fault tolerance; network failures making one interface of the server unavailable do not prevent the service from continuing to operate. SIP servers are likely to have substantial fault tolerance requirements. It is worth noting that because SIP is message-oriented, and not stream oriented, the existing SRV procedures defined in [2] can accomplish the same goal, even when SIP is run over TCP. In fact, SRV records allow the "connection" to fail over to a separate host. Since SIP proxies can run statelessly, failover can be accomplished without data synchronization between the primary and its backups. Thus, the multihoming capabilities of SCTP provide marginal benefits.

It is important to note that most of the benefits of SCTP for SIP occur under loss conditions. Therefore, under a zero loss condition, SCTP transport of SIP should perform on par with TCP transport. Research is needed to evaluate under what loss conditions the improvements in setup times and throughput will be observed. The purpose of this draft is to enable such experimentation in order to provide concrete data on its applicability to SIP.

3 Usage of SCTP

Usage of SCTP requires no additional headers or syntax in SIP. Below we show an example of a SIP URL with a transport parameter and an example of a via header. In both examples SCTP is the transport protocol.

```
sip:Bob.Johnson@example.com;transport=sctp

Via: SIP/2.0/SCTP ws1234.example.com:5060
```

Rules for sending a request over SCTP are identical to TCP. The only difference is that an SCTP sender has to choose a particular stream within an association in order to send the request.

It is important noting that a receiver uses SIP headers such as Call-ID to demultiplex requests (or responses) that belong to different transactions and are received over the same STCP association. The stream id MUST NOT be used for this purpose.

> This way a receiver is not affected by the way a particular sender maps transactions into streams. The receiver will always be able to properly demultiplex incoming SIP traffic without knowing the mapping policy of the sender.

3.1 Mapping of transactions into streams

The more efficient mapping of transactions into streams consists of sending requests belonging to the same call over the same ordered stream. Requests belonging to different calls will be mapped into different streams. It is RECOMMENDED that some kind of stateless hash be used so that requests for the same call all be mapped into the same stream.

> Note that if the number of calls handled by the SIP entity is larger than the number of available streams some calls would have to share the same stream. This would result in the head of the line blocking problem described previously.

Responses are mapped in the same way - responses that belong to the same call are sent over the same oreered stream. However, final responses should be transmitted with the unordered flag set. This prevents lost provisional responses from delaying the delivery of final responses.

Some implementors might think that this way of mapping transactions into streams is somehow complicated. It is possible to perform this mapping in a much simpler way. Senders can take advantage of the ordering of requests that SIP performs at the application layer. That is, SIP does not send a new request until the last transaction has completed. Therefore, all requests and responses can be sent with the unordered flag set over any stream. Effectively, a single stream can be used for all SIP traffic. This way of performing the mapping is almost as effective as the one described previously and it has the advantage of being simpler.

> The scenarios where this second way of performing the mapping might result in reordering of requests/responses represent corner cases that do not justify the increase in the complexity of the first solution.

It is RECOMMENDED to use the second approach because it combines simplicity and efficiency.

4 Locating a SIP server

The primary issue when sending a request is determining whether the next hop server supports SCTP, so that an association can be opened.

This draft assumes that SRV records are the primary vehicle for such determinations. RFC2543bis [4] describes the process that an entity (UAC or proxy) that wishes to send a request to a particular URI MUST follow.

The format of the SRV RR as described in [5] is shown below:

```
     _Service._Proto.Name TTL Class Priority weight Port Target
```

When SRV records are to be used, the service to use when querying for the SRV record is "sip" and the transport protocol is "sctp". So, a SIP client that wants to discover a SIP server that supports SCTP for the domain example.com does a lookup of

```
     _sip._sctp.example.com
```

SCTP associations that were opened by proxies as the result of a successful SRV query SHOULD remain open after the transaction completes. The amount of time after completion of a transaction, before which the connection is closed, is configurable.

> The rule here for SRV provides a neat way to differentiate among connections between proxies, and between proxies and UAs and UAs and proxies. You really only want and need long lived connections between proxies. It is very likely that only proxies have SRV record entries.

5 Conclusion

This draft has presented a discussion on the applicability of SCTP to SIP transport, and provided an experimental mechanism for allowing two SCTP-capable entities to establish and use an SCTP connection.

6 Author's Addresses

Jonathan Rosenberg
dynamicsoft
200 Executive Drive
Suite 120
West Orange, NJ 07052
email: jdrosen@dynamicsoft.com

Henning Schulzrinne
Columbia University
M/S 0401
1214 Amsterdam Ave.
New York, NY 10027-7003

email: schulzrinne@cs.columbia.edu

Gonzalo Camarillo
Ericsson
Advanced Signalling Research Lab.
FIN-02420 Jorvas
Finland
Phone: +358 9 299 3371
Fax: +358 9 299 3052
Email: Gonzalo.Camarillo@ericsson.com

7 Bibliography

[1] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream control transmission protocol," Request for Comments 2960, Internet Engineering Task Force, Oct. 2000.

[2] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: session initiation protocol," Request for Comments 2543, Internet Engineering Task Force, Mar. 1999.

[3] L. Coene et al. , "Stream control transmission protocol applicability statement," Internet Draft, Internet Engineering Task Force, Apr. 2001. Work in progress.

[4] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session initiation protocol," Internet Draft, Internet Engineering Task Force, Nov. 2000. Work in progress.

[5] A. Gulbrandsen, P. Vixie, and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)," Request for Comments 2782, Internet Engineering Task Force, Feb. 2000.

Table of Contents