Network Working Group
Request for Comments: 3257
Category: Informational

L. Coene
Siemens

April 2002

Stream Control Transmission Protocol Applicability Statement

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

This document describes the applicability of the Stream Control Transmission Protocol (SCTP). It also contrasts SCTP with the two dominant transport protocols, User Datagram Protocol (UDP) & Transmission Control Protocol (TCP), and gives some guidelines for when best to use SCTP and when not best to use SCTP.

Table of contents

1 Introduction

SCTP is a reliable transport protocol [RFC2960], which along with TCP [RFC793], RTP [RFC1889], and UDP [RFC768], provides transport-layer services for upper layer protocols and services. UDP, RTP, TCP, and SCTP are currently the IETF standards-track transport-layer protocols. Each protocol has a domain of applicability and services it provides, albeit with some overlaps.

By clarifying the situations where the functionality of these protocols are applicable, this document can guide implementers and protocol designers in selecting which protocol to use.

Special attention is given to services SCTP provides which would make a decision to use SCTP the right one.

Major functions provided by SCTP can be found in Appendix A.

1.1 Terminology

The following terms are commonly identified in this work:

Association: SCTP connection between two endpoints.

Transport address: A combination of IP address and SCTP port number.

Upper layer: The user of the SCTP protocol, which may be an adaptation layer, a session layer protocol, or the user application directly.

Multihoming: Assigning more than one IP network interface to a single endpoint.

2 Transport protocols

2.1 TCP service model

TCP is a connection-oriented (a.k.a., session-oriented) transport protocol. This means that it requires both the establishment of a connection prior to the exchange of application data and a connection tear-down to release system resources after the completion of data transfer.

TCP is currently the most widely used connection-oriented transport protocol for the Internet.

TCP provides the upper layer with the following transport services:

- data reliability;

- data sequence preservation; and

- flow and congestion control.

2.2 SCTP service model

SCTP is also connection-oriented and provides all the transport services that TCP provides. Many Internet applications therefore should find that either TCP or SCTP will meet their transport requirements. Note, for applications conscious about processing cost, there might be a difference in processing cost associated with running SCTP with only a single ordered stream and one address pair in comparison to running TCP.

However, SCTP has some additional capabilities that TCP lacks and This can make SCTP a better choice for some applications and environments:

- multi-streams support:

SCTP supports the delivery of multiple independent user message streams within a single SCTP association. This capability, when properly used, can alleviate the so-called head-of-line-blocking problem caused by the strict sequence delivery constraint imposed to the user data by TCP.

This can be particularly useful for applications that need to exchange multiple, logically separate message streams between two endpoints.

- multi-homing support:

SCTP provides transparent support for communications between two endpoints of which one or both is multi-homed.

SCTP provides monitoring of the reachability of the addresses on the remote endpoint and in the case of failure can transparently failover from the primary address to an alternate address, without upper layer intervention.

This capability can be used to build redundant paths between two SCTP endpoints and can be particularly useful for applications that seek transport-level fault tolerance.

Achieving path redundancy between two SCTP endpoints normally requires that the two endpoints being equipped with multiple interfaces assigned with multiple addresses and that routing is configured appropriately (see Section 3).

- preservation of message boundaries:

SCTP preserves application messages boundaries. This is useful when the application data is not a continuous byte stream but comes in logical chunks that the receiver handles separately.

In contrast, TCP offers a reliable data stream that has no indication of what an application may consider logical chunks of the data.

- unordered reliable message delivery:

SCTP supports the transportation of user messages that have no application-specified order, yet need guaranteed reliable delivery.

Applications that need to send un-ordered reliable messages or prefer using their own message sequencing and ordering mechanisms may find this SCTP capability useful.

2.3 UDP Service model

UDP is connectionless. This means that applications that use UDP do not need to perform connection establishment or tear-down.

As transport services to its upper layer, UDP provides only:

- best-effort data delivery, and

- preservation of message boundaries.

Applications that do not require a reliable transfer of more than a packet's worth of data will find UDP adequate. Some transaction-based applications fall into this category.

3 SCTP Multihoming Issues

SCTP provides transport-layer support for multihoming. Multihoming has the potential of providing additional robustness against network failures. In some applications, this may be extremely important, for example, in signaling transport of PSTN signaling messages [RFC2719].

It should be noted that SCTP multihoming support only deals with communication between two endpoints of which one or both is assigned with multiple IP addresses on possibly multiple network interfaces. It does NOT deal with communication ends that contain multiple endpoints (i.e., clustered endpoints) that can switch over to an alternate endpoint in case of failure of the original endpoint.

Generally, for truly fault resilient communication between two endpoints, the multihoming feature needs more than one IP network interface for each endpoint. The number of paths used is the minimum of network interfaces used by any of the endpoints. When an endpoint selects its source address, careful consideration must be taken. If the same source address is always used, then it is possible that the endpoint will be subject to the same single point of failure. When the endpoint chooses a source address, it should always select the source address of the packet to correspond to the IP address of the Network interface where the packet will be emitted subject to the binding address constraint. The binding address constraint is, put simply, that the endpoint must never choose a source address that is not part of the association i.e., the peer endpoint must recognize any source address used as being part of the association.

The availability of the association will benefit greatly from having multiple addresses bound to the association endpoint when the endpoint is on a multi-homed host.

4 SCTP Network Address Translators (NAT) issues [RFC2663]

When two endpoints are to setup an SCTP association and one (or both) of them is behind a NAT (i.e., it does not have any publicly available network addresses), the endpoint(s) behind the NAT should consider one of the following options:

(1) When single homed sessions are to be used, no transport addresses should be sent in the INIT or INIT ACK chunk(Refer to section 3.3 of RFC2960 for chunk definitions). This will force the endpoint that receives this initiation message to use the source address in the IP header as the only destination address for this association. This method can be used for a NAT, but any multi-homing configuration at the endpoint that is behind the NAT will not be visible to its peer, and thus not be taken advantage of. See figure 1.

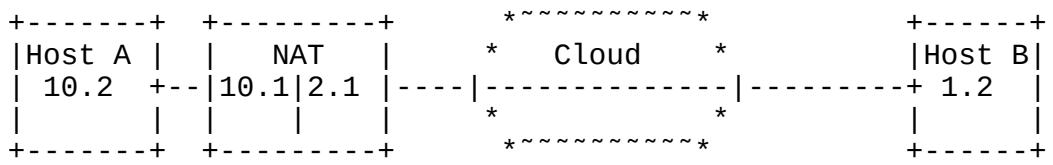

Fig 1: SCTP through NAT without multihoming

For multihoming the NAT must have a public IP address for each represented internal IP address. The host can preconfigure an IP address that the NAT can substitute, or, the NAT can have internal Application Layer Gateway (ALG) which will intelligently translate the IP addresses in the INIT and INIT ACK chunks. See Figure 2.

If Network Address Port Translation is used with a multihomed SCTP endpoint, then any port translation must be applied on a per-association basis such that an SCTP endpoint continues to receive the same port number for all messages within a given association.

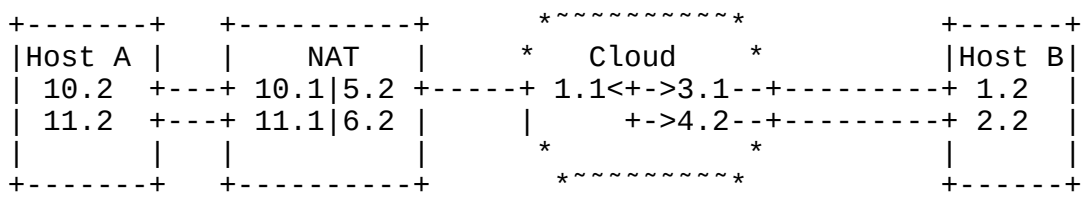

Fig 2: SCTP through NAT with multihoming

(2) Another alternative is to use the hostname feature and DNS to resolve the addresses. The hostname is included in the INIT of the association or in the INIT ACK. The hostname must be resolved by DNS before the association is completely set up. There are special issues regarding NAT and DNS, refer to RFC2694 for details.

5 Security Considerations

In this section, some relevant security issues found in the deployment of the connection-oriented transport protocols will be discussed.

5.1 Security issues with TCP

Some TCP implementations have been known to be vulnerable to blind denial of service attacks, i.e., attacks that had been executed by an attacker that could not see most of the traffic to or from the target host.

The attacker would send a large number of connection establishment requests (TCP-SYN packets) to the attacked target, possibly from faked IP source addresses. The attacked host would reply by sending SYN-ACK packets and entering SYN-received state, thereby allocating space for a TCB. At some point the SYN-queue would fill up, (i.e., the number of connections waiting to be established would rise to a limit) and the host under attack would have to start turning down new connection establishment requests.

TCP implementations with SYN-cookies algorithm [SYN-COOK] reduce the risk of such blind denial of service attacks. TCP implementations can switch to using this algorithm in times when their SYN-queues are filled up while still fully conforming to the TCP specification [RFC793]. However, use of options such as a window scale [RFC1323], is not possible, then. With the SYN-cookie mechanism, a TCB is only created when the client sends back a valid ACK packet to the server, and the 3-way handshake has thus been successfully completed.

Blind connection forgery is another potential threat to TCP. By guessing valid sequence numbers, an attacker would be able to forge a connection. However, with a secure hashsum algorithm, for some of the current SYN-cookie implementations the likelihood of achieving this attack is on the order of magnitude of 1 in 2^24, i.e., the attacker would have to send 2^24 packets before obtaining one forged connection when SYN-cookies are used.

5.2 Security issues with SCTP

SCTP has been designed with the experiences made with TCP in mind. To make it hard for blind attackers (i.e., attackers that are not man-in-the-middle) to inject forged SCTP datagrams into existing associations, each side of an SCTP association uses a 32 bit value called "Verification Tag" to ensure that a datagram really belongs to the existing association. So in addition to a combination of source and destination transport addresses that belong to an established association, a valid SCTP datagram must also have the correct tag to be accepted by the recipient.

Unlike in TCP, usage of cookie in association establishment is made mandatory in SCTP. For the server, a new association is fully established after three messages (containing INIT, INIT-ACK, COOKIE-ECHO chunks) have been exchanged. The cookie is a variable length parameter that contains all relevant data to initialize the TCB on the server side, plus a HMAC used to secure it. This HMAC (MD5 as per [RFC1321] or SHA-1 [SHA1]) is computed over the cookie and a secret, server-owned key.

As specifically prescribed for SCTP implementations [RFC2960], additional resources for new associations may only be reserved in case a valid COOKIE-ECHO chunk is received by a client, and the computed HMAC for this new cookie matches that contained in the cookie.

With SCTP the chances of an attacker being able to blindly forge a connection are even lower than in the case of TCP using SYN-cookies, since the attacker would have to guess a correct value for the HMAC contained in the cookie, i.e., lower than 1 in 2^128 which for all practical purposes is negligible.

It should be noted that SCTP only tries to increase the availability of a network. SCTP does not contain any protocol mechanisms that are directly related to user message authentication, integrity and confidentiality functions. For such features, it depends on the IPsec protocols and architecture and/or on security features of the application protocols.

Transport Layer security(TLS)[RFC2246] using SCTP must always use in-order streams.

Currently the IPSEC working group is investigating the support of multi-homing by IPSEC protocols. At the present time to use IPSEC, one must use 2 * N * M security associations if one endpoint uses N addresses and the other M addresses.

5.3 Security Issues with both TCP and SCTP

It is important to note that neither TCP nor SCTP protect itself from man-in-the-middle attacks where an established session might be hijacked (assuming the attacker can see the traffic from and inject its own packets to either endpoints).

Also, to prevent blind connection/session setup forgery, both TCP implementations supporting SYN-cookies and SCTP implementations rely on a server-known, secret key to protect the HMAC data. It must be ensured that this key is created subject to the recommendations mentioned in [RFC1750].

Although SCTP has been designed carefully as to avoid some of the problems that have appeared with TCP, it has as of yet not been widely deployed. It is therefore possible that new security issues will be identified that will have to be addressed in further revisions of [RFC2960].

6 References and related work

[RFC2960] 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.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[RFC2694] Srisuresh, P., Tsirtsis, G., Akkiraju, P. and A. Heffernan, "DNS extensions to Network Address Translators (DNS_ALG)", RFC 2694, September 1999.

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC2719] Ong, L., Rytina, I., Garcia, M., Schwarzbauer, H., Coene, L., Lin, H., Juhasz, I., Holdrege, M. and C. Sharp, "Architectural Framework for Signaling Transport", RFC 2719, October 1999.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[RFC1323] Jacobson, V., Braden, R. and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[RFC1750] Eastlake, D., Crocker, S. and J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994.

[SHA1] NIST FIPS PUB 180-1, "Secure Hash Standard," National Institute of Standards and Technology, U.S. Department of Commerce, April 1995.

[SYNCOOK] Dan J. Bernstein, SYN cookies, 1997, see also <http://cr.yp.to/syncookies.html>

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC1889] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

7 Acknowledgments

This document was initially developed by a design team consisting of Lode Coene, John Loughney, Michel Tuexen, Randall R. Stewart, Qiaobing Xie, Matt Holdrege, Maria-Carmen Belinchon, Andreas Jungmaier, Gery Verwimp and Lyndon Ong.

The authors wish to thank Renee Revis, I. Rytina, H.J. Schwarzbauer, J.P. Martin-Flatin, T. Taylor, G. Sidebottom, K. Morneault, T. George, M. Stillman, N. Makinae, S. Bradner, A. Mankin, G. Camarillo, H. Schulzrinne, R. Kantola, J. Rosenberg, R.J. Atkinson, and many others for their invaluable comments.

Appendix A: Major functions provided by SCTP

- Reliable Data Transfer
- Multiple streams to help avoid head-of-line blocking
- Ordered and unordered data delivery on a per-stream basis
- Bundling and fragmentation of user data
- TCP friendly Congestion and flow control
- Support continuous monitoring of reachability
- Graceful termination of association
- Support of multi-homing for added reliability
- Some protection against blind denial-of-service attacks
- Some protection against blind masquerade attacks

8 Editor's Address

Lode Coene
Siemens Atea
Atealaan 34
B-2200 Herentals
Belgium

Phone: +32-14-252081
EMail: lode.coene@siemens.atea.be

9. Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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