Network Working Group
Internet-Draft
Intended status: Experimental
Expires: October 30, 2010

X. Fu
C. Dickmann
University of Goettingen
J. Crowcroft
University of Cambridge

April 28, 2010

General Internet Signaling Transport (GIST) over Stream Control Transmission Protocol (SCTP) and Datagram Transport Layer Security (DTLS)

draft-ietf-nsis-ntlp-sctp-11.txt

Abstract

The General Internet Signaling Transport (GIST) protocol currently uses TCP or Transport Layer Security (TLS) over TCP for connection mode operation. This document describes the usage of GIST over the Stream Control Transmission Protocol (SCTP) and Datagram Transport Layer Security (DTLS). The use of SCTP can take advantage of features provided by SCTP, namely streaming-based transport, support of multiple streams to avoid head of line blocking, the support of multi-homing to provide network level fault tolerance, as well as partial reliability extension for partially reliable data transmission. This document also specifies how to establish GIST security over datagram transport protocols using an extension to DTLS.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 30, 2010.

Table of Contents

1. Introduction

This document describes the usage of the General Internet Signaling Transport (GIST) protocol [1] and Datagram Transport Layer Security (DTLS) over the Stream Control Transmission Protocol (SCTP) [2].

GIST, in its initial specification for connection mode operation, runs on top of a byte-stream oriented transport protocol providing a reliable, in-sequence delivery, i.e., using the Transmission Control Protocol (TCP) [7] for signaling message transport. However, some Next Steps in Signaling (NSIS) Signaling Layer Protocol (NSLP) [8] context information has a definite lifetime, therefore, the GIST transport protocol could benefit from flexible retransmission, so stale NSLP messages that are held up by congestion can be dropped. Together with the head-of-line blocking and multihoming issues with TCP, these considerations argue that implementations of GIST should support the Stream Control Transport Protocol (SCTP)[2] as an optional transport protocol for GIST. Like TCP, SCTP supports reliability, congestion control and fragmentation. Unlike TCP, SCTP provides a number of functions that are desirable for signaling transport, such as multiple streams and multiple IP addresses for path failure recovery. Furthermore, SCTP offers an advantage of message-oriented transport instead of using the byte stream oriented TCP where one has to provide its own framing mechanisms. In addition, its Partial Reliability extension (PR-SCTP) [3] supports partial retransmission based on a programmable retransmission timer. Furthermore, Datagram Transport Layer Security (DTLS) [4] provides a viable solution for securing SCTP [5], which allows SCTP to use almost all its transport features and its extensions.

This document defines the use of SCTP as a transport protocol and the use of DTLS as a security mechanism for GIST Messaging Associations and discusses the implications on GIST state maintenance and API between GIST and NSLPs. Furthermore, this document describes how GIST should be interfaced to SCTP and used by NSLPs in order to exploit the additional capabilities offered by SCTP to deliver GIST C-mode messages more effectively. More specifically:

- How to use the multiple streams feature of SCTP.
- How to use the PR-SCTP extension of SCTP.
- How to take advantage of the multi-homing support of SCTP.

The methods of using an unchanged SCTP with GIST described in this document do not require any changes to the high level operation and structure of GIST. Addition of new transport options requires additional interface code and configuration support to allow applications to exploit the additional transport when appropriate. In addition, SCTP over GIST implementations MUST support the optional feature of fragmentation of SCTP user messages.

Additionally, this document also specifies how to establish GIST security using DTLS for use in combination with e.g., SCTP and UDP.

2. Terminology and Abbreviations

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [6]. Other terminologies and abbreviations used in this document are taken from related specifications (e.g., [1] and [2]).

3. GIST Over SCTP

This section defines a new MA-Protocol-ID type, "Forwards-SCTP", for using SCTP as GIST transport protocol.

3.1. Message Association Setup

3.1.1. Overview

The basic GIST protocol specification defines two possible protocols to be used in Messaging Associations, namely Forwards-TCP and TLS. This information are main part of the Stack Configuration Data (SCD) [1]. This document adds Forwards-SCTP and DTLS as another two possible protocol options. In Forwards-SCTP, analog to Forwards-TCP, connections between peers are opened in the forwards direction, from the querying node, towards the responder.

3.1.2. Protocol-Definition: Forwards-SCTP

This MA-Protocol-ID "Forwards-SCTP" denotes a basic use of SCTP between peers. Support for this protocol is OPTIONAL. If this protocol is offered, MA-protocol-options data MUST also be carried in the SCD object. The MA-protocol-options field formats are:

- in a Query: no information apart from the field header.
- in a Response: 2 byte port number at which the connection will be accepted, followed by 2 pad bytes.

The connection is opened in the forwards direction, from the querying node towards the responder. The querying node MAY use any source address and source port. The destination for establishing the message association MUST be derived from information in the Response: the address from the interface- address from the Network-Layer-Information object and the port from the SCD object as described above.

Associations using Forwards-SCTP can carry messages with the transfer attribute Reliable=True. If an error occurs on the SCTP connection such as a reset, as can be reported by an SCTP socket API notification[9], GIST MUST report this to NSLPs as discussed in Section 4.1.2 of [1]. For the multi-homing scenario, when a destination address of a GIST over SCTP peer encounters a change, the SCTP API will notify GIST about the availability of different SCTP endpoint addresses and possible change of the primary path.

3.2. Effect on GIST State Maintenance

As SCTP provides additional functionality over TCP, this section discusses the implications of using GIST over SCTP on GIST State Maintenance.

While SCTP defines uni-directional streams, for the purpose of this document, the concept of a bi-directional stream is used. Implementations MUST establish downstream and upstream (uni-directional) SCTP streams always together and use the same stream identifier in both directions. Thus, the two uni-directional streams (in opposite directions) form a bi-directional stream.

Due to the multi-streaming support of SCTP, it is possible to use different SCTP streams for different resources (e.g., different NSLP sessions), rather than maintaining all messages along the same transport connection/association in a correlated fashion as TCP (which imposes strict (re)ordering and reliability per transport level). However, there are limitations to the use of multi-streaming. All GIST messages for a particular session MUST be sent over the same SCTP stream to assure the NSLP assumption of in-order delivery. Multiple sessions MAY share the same SCTP stream based on local policy.

The GIST concept of Messaging Association re-use is not affected by this document or the use of SCTP. All rules defined in the GIST specification remain valid in the context of GIST over SCTP.

3.3. PR-SCTP Support

A variant of SCTP, PR-SCTP [3] provides a "timed reliability" service, which would be particular useful for delivering GIST Connection mode messages. It allows the user to specify, on a per message basis, the rules governing how persistent the transport service should be in attempting to send the message to the receiver. Because of the chunk bundling function of SCTP, reliable and partially reliable messages can be multiplexed over a single PR-SCTP association. Therefore, a GIST over SCTP implementation SHOULD attempt to establish a PR-SCTP association using "timed reliability"

service instead of a standard SCTP association, if available, to support more flexible transport features for potential needs of different NSLPs.

In a standard SCTP, instead, if a node has sent the first transmission before the lifetime expires, then the message MUST be sent as a normal reliable message. During episodes of congestion this is particularly unfortunate, as retransmission wastes bandwidth that could have been used for other (non-lifetime expired) messages. The "timed reliability" service in PR-SCTP eliminates this issue and is hence RECOMMENDED to be used for GIST over PR-SCTP.

3.4. API between GIST and NSLP

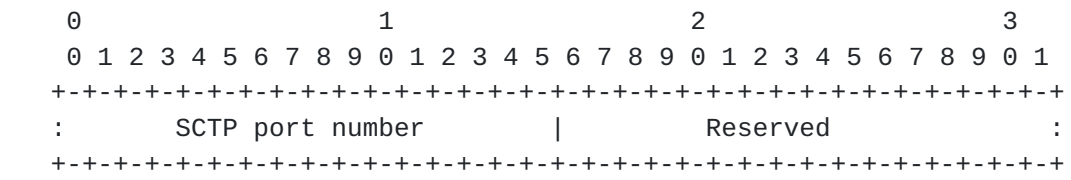
GIST specification defines an abstract API between GIST and NSLPs. While this document does not change the API itself, the semantics of some parameters have slightly different interpretation in the context of SCTP. This section only lists those primitives and parameters, that need special consideration when used in the context of SCTP. The relevant primitives from [1] are as follows:

- The Timeout parameter in API "SendMessage": According to [1], this parameter represents the "length of time GIST should attempt to send this message before indicating an error." When used with PR-SCTP, this parameter is used as the timeout for the "timed reliability" service of PR-SCTP.
- "NetworkNotification": According to [1], this primitive "is passed from GIST to a signalling application. It indicates that a network event of possible interest to the signalling application occurred." Here, if SCTP detects a failure of the primary path, GIST SHOULD also indicate this event to the NSLP by calling this primitive with Network-Notification-Type "Routing Status Change". This notification should be done even if SCTP was able to retain an open connection to the peer due to its multi-homing capabilities.

4. Bit-Level Formats

4.1. MA-Protocol-Options

This section provides the bit-level format for the MA-protocol-options field that is used for SCTP protocol in the Stack-Configuration-Data object of GIST.



SCTP port number = Port number at which the responder will accept SCTP connections

The SCTP port number is only supplied if sent by the responder.

5. Application of GIST over SCTP

5.1. Multi-homing support of SCTP

In general, the multi-homing support of SCTP can be used to improve fault-tolerance in case of a path- or link-failure. Thus, GIST over SCTP would be able to deliver NSLP messages between peers even if the primary path is not working anymore. However, for the Message Routing Methods (MRMs) defined in the basic GIST specification such a feature is only of limited use. The default MRM is path-coupled, which means, that if the primary path is failing for the SCTP association, it most likely is also for the IP traffic that is signaled for. Thus, GIST would need to perform a refresh to the NSIS nodes to the alternative path anyway to cope with the route change. When the two endpoints of a multi-homed SCTP association (but none of the intermediate nodes between them) support NSIS, GIST over SCTP provides a robust means for GIST to deliver NSLP messages even when the primary path fails but at least one alternative path between these (NSIS-enabled) endpoints of the multihomed path is available. Additionally, the use of the multi-homing support of SCTP provides GIST and the NSLP with another source to detect route changes. Furthermore, for the time between detection of the route change and recovering from it, the alternative path offered by SCTP can be used by the NSLP to make the transition more smoothly. Finally, future MRMs might have different properties and therefore benefit from multi-homing more broadly.

5.2. Streaming support in SCTP

Streaming support in SCTP is advantageous for GIST. It allows better parallel processing, in particular by avoiding head of line blocking issue in TCP. Since a same GIST MA may be reused by multiple sessions, using TCP as transport for GIST signaling messages belonging to different sessions may be blocked if another message is dropped. In the case of SCTP, this can be avoided as different

sessions having different requirements can belong to different streams, thus a message loss or reordering in a stream will only affect the delivery of messages within that particular stream, and not any other streams.

6. NAT Traversal Issue

NAT traversal for GIST over SCTP will follow Section 7.2 of [1] and the GIST extensibility capabilities defined in [10]. This specification does not define NAT traversal procedure for GIST over SCTP, although an approach for SCTP NAT traversal is described in [11].

7. Use of DTLS with GIST

This section specifies a new "MA-Protocol-ID" for the use of DTLS in GIST, which denotes a basic use of datagram transport layer channel security, initially in conjunction with SCTP over GIST. It provides authentication, integrity and optionally replay protection for control packets. The use of DTLS for securing GIST over SCTP allows GIST to take the advantage of features provided by SCTP and its extensions. Note replay protection is not available for DTLS over SCTP [5]. The usage of DTLS for GIST over SCTP is similar to TLS for GIST as specified in [1], where a stack-proposal containing both MA-Protocol-IDs for SCTP and DTLS during the GIST handshake phase.

GIST message associations using DTLS may carry messages with transfer attributes requesting confidentiality or integrity protection. The specific DTLS version will be negotiated within the DTLS layer itself, but implementations MUST NOT negotiate to protocol versions prior to DTLS v1.0 and MUST use the highest protocol version supported by both peers. GIST nodes supporting DTLS MUST be able to negotiate the DTLS NULL and block ciphers and SHOULD be able to negotiate the new cipher suites. They MAY negotiate any mutually acceptable ciphersuite that provides authentication, integrity, and confidentiality. The same rules for negotiating TLS cipher suites as specified in Section 5.7.3 of [1] apply.

No MA-protocol-options field is required for DTLS. The configuration information for the transport protocol over which DTLS is running (e.g. SCTP port number) is provided by the MA-protocol-options for that protocol.

8. Security Considerations

The security considerations of [1], [2] and [4] apply. Following [5], replay detection of DTLS over SCTP is not supported.

The usage of DTLS [4] for securing GIST over datagram transport protocols MUST be implemented and SHOULD be used. An implementation of GIST over SCTP with no PR-SCTP support MAY use TLS for its channel security, when DTLS is not available between two GIST peers.

9. IANA Considerations

This specification requests the following codepoints (MA-Protocol-IDs) be assigned in a registry created by [1]:

MA-Protocol-ID	Protocol
3	SCTP opened in the forwards direction
4	DTLS initiated in the forwards direction

Note that MA-Protocol-ID 4 is never used alone but always coupled with a transport protocol in the stack proposal.

10. Acknowledgments

The authors would like to thank John Loughney, Jukka Manner, Magnus Westerlund, Robert Hancock, Andrew McDonald, Martin Stiemerling, Fang-Chun Kuo, Jan Demter, Lauri Liuhto, Michael Tuexen, and Roland Bless for their helpful suggestions.

11. References

11.1. Normative References

[1] Schulzrinne, H. and M. Stiemerling, "GIST: General Internet Signalling Transport", draft-ietf-nsis-ntlp-20 (work in progress), June 2009.

[2] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[3] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, May 2004.

[4] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[5] Tuexen, M., Seggelmann, R., and E. Rescorla, "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)", draft-ietf-tsvwg-dtls-for-sctp-05 (work in progress), March 2010.

[6] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

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

[8] Hancock, R., Karagiannis, G., Loughney, J., and S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", RFC 4080, June 2005.

[9] Stewart, R., Poon, K., Tuexen, M., Yasevich, V., and P. Lei, "Sockets API Extensions for Stream Control Transmission Protocol (SCTP)", draft-ietf-tsvwg-sctpsocket-22 (work in progress), March 2010.

[10] Manner, J., Bless, R., Loughney, J., and E. Davies, "Using and Extending the NSIS Protocol Family", draft-ietf-nsis-ext-07 (work in progress), April 2010.

[11] Stewart, R., Tuexen, M., and I. Ruengeler, "Stream Control Transmission Protocol (SCTP) Network Address Translation", draft-ietf-behave-sctpnat-02 (work in progress), December 2009.

Authors' Addresses

Xiaoming Fu
University of Goettingen
Institute of Computer Science
Goldschmidtstr. 7
Goettingen 37077
Germany

Email: fu@cs.uni-goettingen.de

Christian Dickmann
University of Goettingen
Institute of Computer Science
Goldschmidtstr. 7
Goettingen 37077
Germany

Email: mail@christian-dickmann.de

Jon Crowcroft
University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
UK

Email: jon.crowcroft@cl.cam.ac.uk