Network Working Group
Internet-Draft
Expires: October 9, 2006

S. Kiesel
University of Stuttgart
April 7, 2006

SIMCO over SCTP
draft-kiesel-midcom-simco-sctp-01.txt

Status of this Memo

Copyright Notice

Abstract

This document specifies how to use SCTP for the transport of the SIMCO (Version 3.0) protocol. SIMCO (SImple Middlebox COnfiguration) is a protocol that implements the MIDCOM semantics. It can be used for controlling middleboxes such as firewalls and network address translators. SCTP (Stream Control Transmission Protocol) is a transport layer protocol that is expected to have advantages for this type of application, compared to TCP, which is the default transport layer protocol for SIMCO. The specific requirements for SIMCO when using SCTP instead of TCP are specified in this document.

Table of Contents

1. Introduction

The SIMCO (SImple Middlebox COnfiguration) protocol [I-D.stiemerling-midcom-simco] is a signaling protocol that implements the MIDCOM protocol semantics [RFC3989]. In the context of the MIDCOM architecture [RFC3303], it can be used for controlling middleboxes [RFC3234] such as firewalls and network address translators (NATs).

As outlined in [RFC3303], firewalls and NATs are potential obstacles to packet streams, for example if dynamically negotiated UDP or TCP port numbers are used, as in many peer-to-peer communication applications. SIMCO allows applications to communicate with middleboxes on the datagram path in order to request a dynamic configuration at the middlebox that enables datagram streams to pass the middlebox. Applications can request pinholes at firewalls and address bindings at NATs.

The SIMCO specification [I-D.stiemerling-midcom-simco] mandates TCP (Transmission Control Protocol) [RFC0793] as the default transport for SIMCO.

The Stream Control Transmission Protocol (SCTP) [RFC2960] has originally been designed as a part of the SIGTRAN architecture [RFC2719], for the transport of Signaling System No. 7 (SS7) messages over IP. However, this rather special purpose is achieved by adaptation layers on top of SCTP. SCTP itself has been designed as a generic transport protocol for IP networks, at the same layer in the protocol stack as TCP or UDP.

SCTP offers several advantages compared to TCP for the transport of signaling protocols, especially in scenarios with high reliability requirements or high signaling traffic between two endpoints.

This document supplements the SIMCO Version 3.0 specification [I-D.stiemerling-midcom-simco] by itemizing the requirements for transporting SIMCO over SCTP.

2. Requirements notation

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 [RFC2119].

3. Potential benefits

A summary of SCTP's properties and advantages is given in [RFC3257]. Two of these properties are of special benefit, especially when SIMCO is used in large-scale deployments.

3.1. Multiple streams for reduced head-of-line blocking

SCTP allows to send user messages over several streams within a single SCTP association. SCTP ensures in-order delivery only for those messages that are sent through the same stream. This limits the impact of the so-called head-of-line blocking problem to one stream.

Head-of-line blocking occurs if a reliable transport layer protocol has to retransmit a message due to packet loss or bit errors. Subsequent messages that have already arrived at the receiving side cannot be delivered to the upper protocol layers until the retransmission is completed. Instead, they have to be buffered, in order to ensure in-order delivery. Unlike TCP, which has to buffer the whole connection, SCTP has to buffer only one stream, while messages from other streams still can be delivered to the upper layer protocol.

SIMCO can profit from SCTP's multiple streams feature as a typical SIMCO session consists of several message threads, each of them requiring in-order delivery, but being independent of each others.

3.2. Multihoming Support

SCTP transparently supports endpoints with several IP addresses. This allows for having several physical network interfaces connected to different networks. The availability of all paths between these interfaces is monitored with keepalive messages. If the primary path, which is normally used for message transmission, becomes unavailable an automatic changeover to one of the backup paths is performed.

4. Usage of SCTP as transport protocol for SIMCO

4.1. Leveraging SCTP's multiple streams feature

A central concept of SIMCO are the so-called policy rules. Policy rules correspond to pinholes on firewalls, and to address bindings on NATs. They are created, modified, and deleted by means of SIMCO transactions, i.e., a request sent from the SIMCO agent to the middlebox plus a positive or negative reply.

All SIMCO messages bear a Transaction Identifier (TID) field that identifies to which transaction the message belongs to. The TID is uniquely assigned by the entity that sends the transaction's first message (usually the SIMCO agent, in case of asynchronous notifications the middlebox). Policy rules are identified by a numerical policy rule identifier (PID). When a request creates a new policy rule at the middlebox, the middlebox assigns a unique PID and returns it in the positive reply. All subsequent transactions that modify or delete the policy rule contain the respective PID.

It is important that the transport layer protocol preserves the order of transactions that refer to the same policy rule, i.e., to the same PID. For example, it would be bad if a SIMCO agent requested a lifetime extension (PLC with non-zero value) for a specific policy rule and immediately afterwards requested to delete it (PLC with zero value), but the delete request was delivered to the middlebox before the lifetime extension request. However, there is no requirement that prohibits reordering of SIMCO messages that refer to different policy rules.

The basic idea is therefore to have several bidirectional pairs of streams within the SCTP association. All SIMCO messages belonging to one transaction, i.e., all messages that have the same TID value shall be sent over the same stream pair. Furthermore, all SIMCO transactions that refer to the same PID shall be transported over the same stream pair. Transactions that refer to other PIDs may be transported over other streams and are therefore not affected if one stream suffers from head-of-line blocking.

4.2. Establishment of SCTP association

In order to setup a SIMCO session, an agent establishes one SCTP association to port 7626/SCTP at the middlebox.

During establishment of the SCTP association, the agent and the middlebox will negotiate the number of streams to use, according to section 5.1.1 of [RFC2960]. After the negotiation the SCTP association has N1 streams from the agent to the middlebox and N2

streams from the middlebox to the agent. From these streams, only the first N = min(N1,N2) streams in each direction (i.e., streams with stream identifiers 0 to (N-1)) will be used. A pair consisting of two streams with the same stream identifier but running in opposite directions is considered and used as one bi-directional stream pair.

If the middlebox has accepted fewer streams as requested by the agent, the agent MAY decide to shutdown the SCTP association without sending any SIMCO messages.

The N bidirectional stream pairs are used for the transmission of policy rule related transactions and policy rule related asynchronous notifications. SIMCO session related transactions and notifications are always transmitted over stream pair 0.

4.3. Encapsulation of SIMCO messages into SCTP data chunks

Every single SIMCO message MUST be passed as exactly one user message to the SCTP layer. SCTP implementations used in conjunction with SIMCO MUST support fragmentation of user messages that exceed the MTU (see [RFC2960], section 6.9).

The SCTP data chunk's Payload Protocol Identifier is set to 15 (SIMCO).

4.4. Mapping of SIMCO Session Control Messages into SCTP streams

The objective of the following requirements is to reduce the impact of head-of-line blocking by distributing transactions on several SCTP streams while retaining ordering where needed.

4.4.1. SIMCO Agent behavior requirements

4.4.1.1. Session Control Messages

Session control messages (SE, SA, ST) MUST be sent over stream 0.

4.4.1.2. Policy Rule Control Messages that do not refer to a PID

The SIMCO agent may use any implementation specific strategy to distribute policy rule control messages that do not include a PID attribute (PRR, PER, PRL, PDR) over the N streams.

Possible strategies are, for example:

- round robin
- SIMCO transaction identifier (TID) modulo N

If the SIMCO agent receives a PRR positive reply, PER positive reply, or PDR positive reply, the SIMCO agent MUST internally store the mapping between the PID contained in the reply and the number of the stream it was received on.

4.4.1.3. Policy Rule Control Messages that refer to a PID

The SIMCO agent MUST send requests that include a PID attribute (PEA, PLC, PS) using the stream number that was saved for this PID, according to Section 4.4.1.2.

An agent may want to send one of these requests (especially PS) referring to a policy rule that was not established by this agent, e.g., if the agent has learned the PID by means of a PRL transaction. In this case, the agent has no mapping from this PID to an SCTP stream number. The agent may send this request over any stream, according to a strategy as described in Section 4.4.1.2. Then, the SIMCO agent MUST internally store the mapping between the PID and the stream number it had chosen, for later usage as described in the first paragraph of this section.

4.4.2. SIMCO server (middlebox) requirements

4.4.2.1. Replies to transactions initiated by Agent

For sending (positive or negative) replies to requests issued by the MIDCOM agent, the middlebox MUST use the same outbound stream number as the number of the inbound stream on which the request was received on.

4.4.2.2. Asynchronous Rule Event notifications

For sending ARE messages, the middlebox may use any implementation specific strategy, such as round robin, to distribute the ARE notification messages over all N streams.

4.4.2.3. Asynchronous Session Termination notifications

AST messages MUST be sent over stream 0.

5. Security Considerations

SCTP has certain advantages over TCP with respect to the protection against denial of service (DoS) attacks (state cookie mechanism, verification tags) [RFC2960]. However, the main threats that are identified in [I-D.stiemerling-midcom-simco] apply for using SIMCO over SCTP as well. Therefore, the requirements given in the section "Security Considerations" of [I-D.stiemerling-midcom-simco] MUST be fulfilled when using SCTP instead of TCP, too. Information on known security threats and countermeasures specific to SCTP can be found in [I-D.ietf-tsvwg-sctpthreat].

6. IANA considerations

For the transport of SIMCO over SCTP a user port number (see Section 4.2) and a SCTP Payload Protocol Identifier (see Section 4.3) have already been registered with IANA.

No further IANA action is required by this document.

7. References

[I-D.ietf-tsvwg-sctpsocket]
Stewart, R., "Sockets API Extensions for Stream Control Transmission Protocol (SCTP)", draft-ietf-tsvwg-sctpsocket-12 (work in progress), February 2006.

[I-D.ietf-tsvwg-sctpthreat]
Stewart, R., "Stream Control Transmission Protocol (SCTP) Security Threats", draft-ietf-tsvwg-sctpthreat-00 (work in progress), January 2006.

[I-D.stiemerling-midcom-simco]
Stiemerling, M., "Simple Middlebox Configuration (SIMCO) Protocol Version 3.0", draft-stiemerling-midcom-simco-08 (work in progress), December 2005.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2719] 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.

[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.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[RFC3257] Coene, L., "Stream Control Transmission Protocol Applicability Statement", RFC 3257, April 2002.

[RFC3303] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A., and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002.

[RFC3989] Stiemerling, M., Quittek, J., and T. Taylor, "Middlebox Communications (MIDCOM) Protocol Semantics", RFC 3989, February 2005.

Author's Address

Sebastian Kiesel
University of Stuttgart
Pfaffenwaldring 47
Stuttgart 70569
Germany

Phone: +49 711 685 7992
Email: kiesel@ikr.uni-stuttgart.de
URI: http://www.ikr.uni-stuttgart.de/en/~kiesel

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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