Network Working Group
Internet-Draft
Expires: December 20, 2006

J. Hadi Salim
ZNYX Networks
June 18, 2006

SCTP based TML (Transport Mapping Layer) for ForCES protocol
draft-hadi-forces-sctptml-00

Status of this Memo

Copyright Notice

Abstract

This document defines the SCTP based TML (Transport Mapping Layer) for the ForCES protocol. It explains the rationale for choosing the SCTP (Stream Control Transmission Protocol) [RFC2960] and also describes how this TML addresses all the requirements described in [RFC3654] and the ForCES protocol [FE-PROTO] draft.

Table of Contents

1. Definitions

The following definitions are taken from [RFC3654]and [RFC3746]:

ForCES Protocol -- The protocol used at the Fp reference point in the ForCES Framework in [RFC3746].

ForCES Protocol Layer (ForCES PL) -- A layer in ForCES protocol architecture that defines the ForCES protocol architecture and the state transfer mechanisms as defined in [FE-PROTO].

ForCES Protocol Transport Mapping Layer (ForCES TML) -- A layer in ForCES protocol architecture that specifically addresses the protocol message transportation issues, such as how the protocol messages are mapped to different transport media (like TCP, IP, ATM, Ethernet, etc), and how to achieve and implement reliability, multicast, ordering, etc.

2. Introduction

The ForCES (Forwarding and Control Element Separation) working group in the IETF is defining the architecture and protocol for separation of control and forwarding elements in network elements such as routers. [RFC3654] and [RFC3746] respectively define architectural and protocol requirements for the communication between CE and FE. The ForCES protocol layer specification [FE-PROTO] describes the protocol semantics and workings. The ForCES protocol layer operates on top of an inter-connect hiding layer known as the TML. The relationship is illustrated in Figure 1.

This document defines the SCTP based TML for the ForCES protocol layer. It also addresses all the requirements for the TML including security, reliability, etc as defined in [FE-PROTO].

3. Protocol Framework Overview

The reader is referred to the Framework document [RFC3746], and in particular sections 3 and 4, for an architectural overview and explanation of where and how the ForCES protocol fits in.

There may be some content overlap between the ForCES protocol draft [FE-PROTO] and this section in order to provide clarity.

The ForCES layout constitutes two pieces: the PL and TML layer. This is depicted in Figure 1.

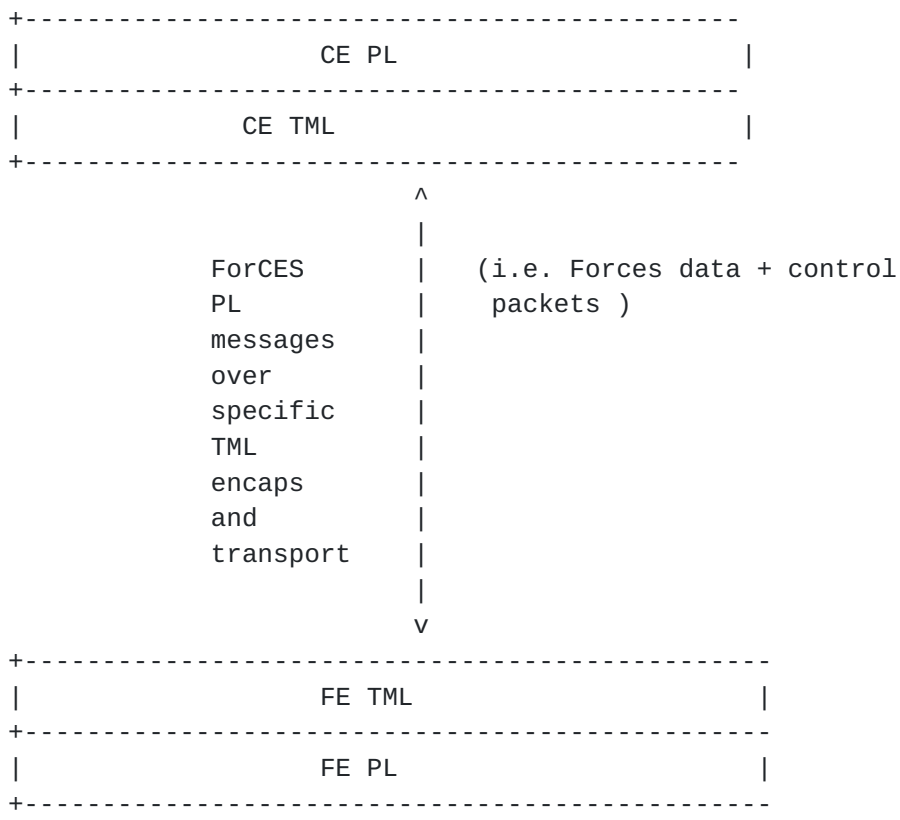

Figure 1: Message exchange between CE and FE to establish an NE association

The PL layer is in charge of the ForCES protocol. Its semantics and message layout are defined in [FE-PROTO]. The TML Layer is necessary to connect two ForCES PL layers as shown in Figure 1.

Both the PL and TML are standardized by the IETF. While only one PL is defined, different TMLs are expected to be standardized. The TML at each of the peers (CE and FE) is expected to be of the same definition in order to inter-operate.

When transmitting, the PL delivers its messages to the TML. The TML then delivers the PL message to the destination peer TML(s) as defined by the addressing in the PL message.

On reception of a message, the TML delivers the message to its destination PL layer(s).

3.1. The PL

The PL is common to all implementations of ForCES and is standardized

by the IETF [FE-PROTO]. The PL layer is responsible for associating an FE or CE to an NE. It is also responsible for tearing down such associations. An FE uses the PL layer to throw various subscribed-to events to the CE PL layer as well as respond to various status requests issued from the CE PL. The CE configures both the FE and associated LFBs attributes using the PL layer. In addition the CE may send various requests to the FE to activate or deactivate it, reconfigure its HA parameterization, subscribe to specific events etc.

3.2. The TML layer

The TML layer is responsible for transport of the PL layer messages. The TML provides the following services on behalf of the ForCES protocol:

1. Reliability
   As defined by RFC 3654, section 6 #6.

2. Security
   TML provides security services to the ForCES PL. The TML definition needs to define how the following are achieved:

   * Endpoint authentication of FE and CE

   * Message authentication

   * Confidentiality service

3. Congestion Control
   The congestion control mechanism defined by the TML should prevent the FE from being overloaded by the CE. Additionally, the circumstances under which notification is sent to the PL to notify it of congestion must be defined.

4. Uni/multi/broadcast addressing/delivery, if any
   If there is any mapping between PL and TML level uni/multi/ broadcast addressing it needs to be defined.

5. Transport High Availability
   It is expected that availability of transport links is the TML's responsibility. However, on config basis, the PL layer may wish to participate in link failover schemes and therefore the TML must allow for this.

6. Encapsulations used
   Different types of TMLs will encapsulate the PL messages on different types of headers. The TML needs to specify the

encapsulation used.

7. Prioritization
   The TML SHOULD will be able to handle up to 8 priority levels needed by the PL and will provide preferential treatment.
   The TML needs to define how this is achieved.

8. Protection against DoS attacks
   As described in the Requirements RFC 3654, section 6

It is expected more than one TML will be standardized. The different TMLs each could implement things differently based on capabilities of underlying media and transport. However, since each TML is standardized, interoperability is guaranteed as long as both endpoints support the same TML.

3.2.1. TML Parameterization

It is expected that it should be possible to use a configuration reference point, such as the FEM or the CEM, to configure the TML.

Some of the configured parameters may include:

- PL ID
- Connection Type and associated data. For example if a TML uses IP/TCP/UDP then parameters such as TCP and UDP ports and IP addresses need to be configured.
- Number of transport connections
- Connection Capability, such as bandwidth, etc.
- Allowed/Supported Connection QoS policy (or Congestion Control Policy)

3.3. The TML-PL interface

[TML-API] defines an interface between the PL and the TML layers. The end goal of [TML-API] is to provide a consistent top edge semantics for all TMLs to adhere to. Conforming to such an interface makes it easy to plug in different TMLs over time. It also allows for simplified TML parameterization requirement stated in Section 3.2.1.

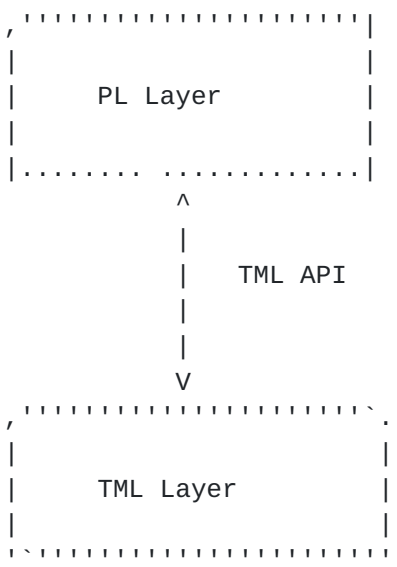

Figure 2: The TML-PL interface

We are going to assume the existence of such an interface and not discuss it further. The reader is encouraged to read [TML-API] as a background.

4. SCTP TML overview

4.1. Introduction to SCTP

SCTP [RFC2960] is an end-to-end transport protocol that is equivalent to TCP, UDP, or DCCP in many aspects. With a few exceptions, SCTP can do most of what UDP, TCP, or DCCP can achieve.

Like TCP, it provides ordered, reliable, connection-oriented, flow-controlled, congestion controlled data exchange. Unlike TCP, it does not provide byte streaming and instead provides message boundaries.

Like UDP, it can provide unreliable, unordered data exchange. Unlike UDP, it does not provide multicast support

Like DCCP, it can provide unreliable, ordered, congestion controlled, connection-oriented data exchange.

SCTP also provides other services that none of the 3 transport protocols mentioned above provide. These include:

- Multi-homing
  An SCTP connection can make use of multiple destination IP

addresses to communicate with its peer.

- Runtime IP address binding
  With the SCTP ADDIP feature, a new address can be bound at runtime. This allows for migration of endpoints without restarting the association (valuable for high availability).

- A range of reliability shades with congestion control
  SCTP offers a range of services from full reliability to none, and from full ordering to none. With SCTP, on a per message basis, the application can specify a message's time-to-live. When the expressed time expires, the message can be "skipped".

- Built-in heartbeats
  SCTP has built-in heartbeat mechanism that validate the reachability of peer addresses.

- Multi-streaming
  A known problem with TCP is head of line (HOL) blocking. If you have independent messages, TCP enforces ordering of such messages. Loss at the head of the messages implies delays of delivery of subsequent packets. SCTP allows for defining upto 64K independent streams over the same socket connection, which are ordered independently.

- Message boundaries with reliability
  SCTP allows for easier message parsing (just like UDP but with reliability built in) because it establishes boundaries on a PL message basis. On a TCP stream, one would have to peek into the message to figure the boundaries.

- Improved SYN DOS protection
  Unlike TCP, which does a 3 way connection setup handshake, SCTP does a 4 way handshake. This improves against SYN-flood attacks because listening sockets do not set up state until a connection is validated.

- Simpler transport events
  An application (such as the TML) can subscribe to be notified of both local and remote transport events. Events such as indication of association changes, addressing changes, remote errors, expiry of timed messages, etc, are off by default and require explicit subscription.

- Simplified replicasting
  Although SCTP does not allow for multicasting it allows for a single message from an application to be sent to multiple peers. This reduces the messaging that typically crosess different memory

domains within a host.

4.2. Rationale for using SCTP for TML

SCTP has all the features required to provide a robust TML. As a transport that is all-encompassing, it negates the need for having multiple transport protocols, as has been suggested so far in the other proposals for TMLs. As a result it allows for simpler coding and therefore reduces a lot of the interoperability concerns.

SCTP is also very mature and widely deployed completing the equation that makes it a superior choice in comparison with other proposed TMLs.

4.3. Meeting TML requirements

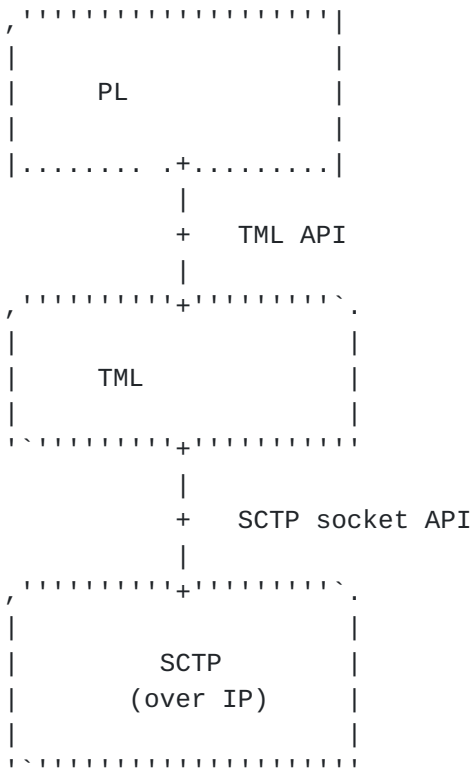

Figure 3: The TML-SCTP interface

Figure 3 above shows the interfacing between the TML and SCTP. There is only one socket connection open with two streams used. The first

stream which is high priority will be dedicated for configuration data and the second lower priority stream is used for data path redirect. The TML will use information passed by the TML API to select which of the two streams to use when sending. The TML will also subscribe to events from SCTP associated with the two streams.

4.3.1. Reliability

As mentioned earlier, a shade of reliability ranges is possible in SCTP. Therefore this requirement is met.

4.3.2. Congestion control

Congestion control is built into SCTP. Therefore, this requirement is met.

4.3.3. Timeliness and prioritization

By using multiple streams in conjunction with the partial-reliability feature, both timeliness and prioritization can be achieved.

4.3.4. Addressing

SCTP can be told to replicast packets to multiple destinations. The TML will translate PL level addresses, to a variety of unicast IP addresses in order to emulate multicast and broadcast. Note, however, that there are no extra headers required.

4.3.5. HA

Transport link resiliency is SCTP's strongest point (where it totally outclasses all other TML proposals). Failure detection and recovery is built in as mentioned earlier.

- With multi-homing, path diversity is provided. Should one of the peer IP addresses become unreachable, the other(s) can be used without involving lower layer (routing, for example) convergence or even the TML becoming aware.

- With heartbeats and data transmission thresholds, on a per peer IP address, reachability faults can be detected. The faults could be a result of an unreachable address or peer, which may be caused by a variety of reasons, like interface, network, or endpoint failures.

- With the ADDIP feature, one can migrate IP addresses to other nodes at runtime. This is not unlike the VRRP protocol use.

4.3.6. DOS prevention

Two separate streams are used within any FE-CE setup: the higher priority one is for configuration and the lower priority one for data redirection. The design is strict priority to further guarantee that lower priority is starved if lack of resources happen.

4.3.7. Encapsulation

There is no extra encapsulation added by this TML. In the future, should the need arise, SCTP provides for extensions to be added to it by defining new chunks.

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

TBA: how to use TLS,IPSEC

7. Acknowledgements

8. References

8.1. Normative References

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[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.

[RFC3654] Khosravi, H. and T. Anderson, "Requirements for Separation of IP Control and Forwarding", RFC 3654, November 2003.

[RFC3746] Yang, L., Dantu, R., Anderson, T., and R. Gopal, "Forwarding and Control Element Separation (ForCES)

Framework", RFC 3746, April 2004.

8.2. Informative References

[FE-MODEL]
Yang, L., Halpern, J., Gopal, R., DeKok, A., Haraszti, Z., and S. Blake, "ForCES Forwarding Element Model", Mar. 2006.

[FE-PROTO]
Doria (Ed.), A., Haas (Ed.), R., Hadi Salim (Ed.), J., Khosravi (Ed.), H., M. Wang (Ed.), W., Dong, L., and R. Gopal, "ForCES Protocol Specification", Mar. 2006.

[TML-API] M. Wang, W. and J. Hadi Salim, "ForCES Transport Mapping Layer (TML) Service Primitives", Apr. 2006.

Author's Address

Jamal Hadi Salim
ZNYX Networks
Ottawa, Ontario
Canada

Email: hadi@znyx.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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