Internet Engineering Task Force (IETF)
Request for Comments: 6053
Category: Informational
ISSN: 2070-1721

E. Haleplidis
University of Patras
K. Ogawa
NTT Corporation
W. Wang
Zhejiang Gongshang University
J. Hadi Salim
Mojatatu Networks

November 2010

Implementation Report for Forwarding and Control Element Separation (ForCES)

Abstract

Forwarding and Control Element Separation (ForCES) defines an architectural framework and associated protocols to standardize information exchange between the control plane and the forwarding plane in a ForCES network element (ForCES NE). RFC 3654 has defined the ForCES requirements, and RFC 3746 has defined the ForCES framework.

This document is an implementation report for the ForCES Protocol, Model, and the Stream Control Transmission Protocol-based Transport Mapping Layer (SCTP TML) documents, and includes a report on interoperability testing and the current state of ForCES implementations.

Status of This Memo

Table of Contents

1. Introduction

This document is an implementation report for the ForCES protocol, model, and the SCTP TML documents, and includes an interoperability report.

It follows the outline suggested by [RFC5657].

ForCES defines an architectural framework and associated protocols to standardize information exchange between the control plane and the forwarding plane in a ForCES network element (ForCES NE). [RFC3654] has defined the ForCES requirements, and [RFC3746] has defined the ForCES framework.

1.1. ForCES Protocol

The ForCES protocol works in a master-slave mode in which forwarding elements (FEs) are slaves and control elements (CEs) are masters. The protocol includes commands for transport of Logical Functional Block (LFB) configuration information, association setup, status, event notifications, etc. The reader is encouraged to read the ForCES Protocol Specification [RFC5810] for further information.

1.2. ForCES Model

The ForCES Model [RFC5812] presents a formal way to define FE Logical Functional Blocks (LFBs) using XML. LFB configuration components, capabilities, and associated events are defined when the LFB is formally created. The LFBs within the FE are accordingly controlled in a standardized way by the ForCES protocol.

1.3. Transport Mapping Layer

The TML transports the protocol layer (PL) messages [RFC5810]. The TML is where the issues of how to achieve transport-level reliability, congestion control, multicast, ordering, etc. are handled. All ForCES protocol layer implementations MUST be portable across all TMLs. Although more than one TML may be standardized for the ForCES protocol, all implementations MUST implement SCTP TML [RFC5811].

2. Terminology and Conventions

2.1. Requirements Language

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].

2.2. Definitions

This document follows the terminology defined by the ForCES requirements in [RFC3654] and by the ForCES framework in [RFC3746]. The definitions are repeated below for clarity.

Control Element (CE) - A logical entity that implements the ForCES protocol and uses it to instruct one or more FEs on how to process packets. CEs handle functionality such as the execution of control and signaling protocols.

Forwarding Element (FE) - A logical entity that implements the ForCES protocol. FEs use the underlying hardware to provide per-packet processing and handling as directed/controlled by one or more CEs via the ForCES protocol.

LFB (Logical Functional Block) - The basic building block that is operated on by the ForCES protocol. The LFB is a well defined, logically separable functional block that resides in an FE and is controlled by the CE via the ForCES protocol. The LFB may reside at the FE’s datapath and process packets or may be purely an FE control or configuration entity that is operated on by the CE. Note that the LFB is a functionally accurate abstraction of the FE’s processing capabilities, but not a hardware-accurate representation of the FE implementation.

LFB Class and LFB Instance - LFBs are categorized by LFB Classes. An LFB Instance represents an LFB Class (or Type) existence. There may be multiple instances of the same LFB Class (or Type) in an FE. An LFB Class is represented by an LFB Class ID, and an LFB Instance is represented by an LFB Instance ID. As a result, an LFB Class ID associated with an LFB Instance ID uniquely specifies an LFB existence.

LFB Metadata - Metadata is used to communicate per-packet state from one LFB to another, but is not sent across the network. The FE model defines how such metadata is identified, produced, and consumed by the LFBs. It defines the functionality but not how metadata is encoded within an implementation.

LFB Components - Operational parameters of the LFBs that must be visible to the CEs are conceptualized in the FE model as the LFB components. The LFB components include, for example, flags, single-parameter arguments, complex arguments, and tables that the CE can read and/or write via the ForCES protocol (see below).

ForCES Protocol - While there may be multiple protocols used within the overall ForCES architecture, the term "ForCES protocol" and "protocol" refer to the "Fp" reference points in the ForCES framework in [RFC3746]. This protocol does not apply to CE-to-CE communication, FE-to-FE communication, or to communication between FE and CE managers. Basically, the ForCES protocol works in a master-slave mode in which FEs are slaves and CEs are masters.

ForCES Protocol Transport Mapping Layer (ForCES TML) - A layer in ForCES protocol architecture that uses the capabilities of existing transport protocols to specifically address 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. The ForCES TML specifications are detailed in separate ForCES documents, one for each TML.

3. Summary

Three independent implementations, NTT Japan, the University of Patras, and Zhejiang Gongshang University, were surveyed and found to already implement all the major features. All implementors mentioned they will be implementing all missing features in the future.

An interop test was conducted in July 2009 for all three implementations. Two other organizations, Mojatatu Networks and Hangzhou Baud Information and Networks Technology Corporation, which independently extended two different well known public domain protocol analyzers, Ethereal/Wireshark and tcpdump, also participated in the interop for a total of five independent organizations implementing. The two protocol analyzers were used to verify the validity of ForCES protocol messages (and in some cases semantics).

There were no notable difficulties in the interoperability test, and almost all issues were code bugs that were dealt with mostly on site; tests repeated successfully, as stated in Section 6.2.3.

4. Methodology

This report describes an implementation experience survey as well as the results of the interoperability test.

The survey information was gathered after implementors answered a brief questionnaire regarding all ForCES Protocol, Model, and SCTP TML features. The results can be seen in Section 6.1.

The interoperability results were part of the interoperability test. Extended Ethereal and extended tcpdump were used to verify the results. The results can be seen in Section 6.2.

5. Exceptions

The core features of the ForCES Protocol, Model, and SCTP TML were implemented and assessed in an interop test in July 2009. The intention of the interop testing was to validate that all the main features of the three core documents were interoperable amongst different implementations. The tested features can be seen in Section 6.2.2.

Different organizations surveyed have implemented certain features but not others. This approach is driven by the presence of different LFBs that the different organizations currently implement. All organizations surveyed have indicated their intention to implement all outstanding features in due time. The implemented features can be seen in Section 6.1.

The mandated TML security requirement, IP security (IPsec), was not validated during the interop and is not discussed in this document. Since IPsec is well known and widely deployed, not testing in the presence of IPsec does not invalidate the tests done. Note that Section 6.1.3.3 indicates that none of the implementations reporting included support for IPsec, but all indicated their intention to implement it.

Although the SCTP priority ports have changed since the interoperability test with the version of the SCTP TML draft available prior to the publication of RFC 5811, the change has no impact on the validity of the interoperability test.

6. Detail Section

6.1. Implementation Experience

Three different organizations have implemented the ForCES Protocol, Model, and SCTP TML and answered a questionnaire. These are:

- NTT Japan
- University of Patras
- Zhejiang Gongshang University

Extensions to protocol analyzers capable of understanding ForCES protocol messages are considered part of an implementation, since these analyzers can now understand and validate ForCES protocol message that have been exchanged. Two such extensions have been created:

- o Extension to Ethereal/Wireshark [ethereal].

- o Extension to tcpdump [tcpdump].

All implementors were asked about the ForCES features they have implemented. For every item listed, the respondents indicated whether they had implemented, will implement, or won't implement at all.

6.1.1. ForCES Protocol Features

6.1.1.1. Protocol Messages

Protocol Message	NTT Japan	University of Patras	Zhejiang Gongshang University
Association Setup	Implemented	Implemented	Implemented
Association Setup Response	Implemented	Implemented	Implemented
Association Teardown	Implemented	Implemented	Implemented
Config	Implemented	Implemented	Implemented
Config Response	Implemented	Implemented	Implemented
Query	Implemented	Implemented	Implemented
Query Response	Implemented	Implemented	Implemented
Event Notification	Implemented	Will Implement	Implemented
Packet Redirect	Implemented	Will Implement	Implemented
Heartbeat	Implemented	Implemented	Implemented

ForCES Protocol Messages

6.1.1.2. MainHeader Handling

Header Field	NTT Japan	University of Patras	Zhejiang Gongshang University
Correlator	Implemented	Implemented	Implemented
ACK Indicator Flag	Implemented	Implemented	Implemented
Priority Flag	Will Implement	Implemented	Implemented
Execution Mode Flag	Will Implement	Will Implement	Implemented
Atomic Transaction Flag	Will Implement	Will Implement	Implemented
Transaction Phase Flag	Will Implement	Will Implement	Implemented

MainHeader Handling

6.1.1.3. TLV Handling

TLV	NTT Japan	University of Patras	Zhejiang Gongshang University
REDIRECT-TLV	Implemented	Will Implement	Implemented
ASResult-TLV	Implemented	Implemented	Implemented
ASTReason-TLV	Implemented	Implemented	Implemented
LFBSelect-TLV	Implemented	Implemented	Implemented
OPER-TLV	Implemented	Implemented	Implemented
PATH-DATA-TLV	Implemented	Implemented	Implemented
KEYINFO-TLV	Will Implement	Will Implement	Implemented
FULLDATA-TLV	Implemented	Implemented	Implemented
SPARSEDATA-TLV	Will Implement	Will Implement	Implemented
ILV	Will Implement	Will Implement	Implemented
METADATA-TLV	Will Implement	Will Implement	Implemented
RESULT-TLV	Implemented	Implemented	Implemented
REDIRECTDATA-TLV	Implemented	Will Implement	Implemented

TLVs Supported

6.1.1.4. Operation Types Supported

Operation	NTT Japan	University of Patras	Zhejiang Gongshang University
SET	Implemented	Implemented	Implemented
SET-PROP	Will Implement	Will Implement	Implemented
SET-RESPONSE	Implemented	Implemented	Implemented
SET-PROP-RESPONSE	Will Implement	Will Implement	Implemented
DEL	Implemented	Will Implement	Implemented
DEL-RESPONSE	Implemented	Will Implement	Implemented
GET	Implemented	Implemented	Implemented
GET-PROP	Will Implement	Will Implement	Implemented
GET-RESPONSE	Implemented	Implemented	Implemented
GET-PROP-RESPONSE	Will Implement	Will Implement	Implemented
REPORT	Implemented	Implemented	Implemented
COMMIT	Will Implement	Will Implement	Implemented
COMMIT-RESPONSE	Will Implement	Will Implement	Implemented
TRCOMP	Will Implement	Will Implement	Implemented

Operation Types Supported

6.1.1.5. ForCES Protocol Advanced Features

Feature	NTT Japan	University of Patras	Zhejiang Gongshang University
Execute Mode	Will Implement	Will Implement	Implemented
Transaction	Will Implement	Will Implement	Implemented
Batching	Will Implement	Implemented	Implemented
Command Pipelining	Will Implement	Will Implement	Will Implement
Heartbeats	Implemented	Implemented	Implemented

ForCES Protocol Advanced Features

6.1.2. ForCES Model Features

6.1.2.1. Basic Atomic Types Supported

Atomic Type	NTT Japan	University of Patras	Zhejiang Gongshang University
char	Implemented	Implemented	Will Implement
uchar	Implemented	Implemented	Implemented
int16	Implemented	Implemented	Will Implement
uint16	Implemented	Implemented	Will Implement
int32	Implemented	Implemented	Implemented
uint32	Implemented	Implemented	Implemented
int64	Implemented	Implemented	Will Implement
uint64	Implemented	Implemented	Will Implement
boolean	Implemented	Implemented	Implemented
string[N]	Implemented	Implemented	Implemented
string	Implemented	Implemented	Implemented
byte[N]	Implemented	Implemented	Implemented
octetstring[N]	Implemented	Implemented	Will Implement
float32	Implemented	Implemented	Will Implement
float64	Implemented	Implemented	Will Implement

Basic Atomic Types Supported

6.1.2.2. Compound Types Supported

Compound Type	NTT Japan	University of Patras	Zhejiang Gongshang University
structs	Implemented	Implemented	Implemented
arrays	Implemented	Implemented	Implemented

Compound Types Supported

6.1.2.3. LFBs Supported

6.1.2.3.1. FE Protocol LFB

Protocol Datatypes	NTT Japan	University of Patras	Zhejiang Gongshang University
CEHBPolicy	Implemented	Implemented	Implemented
FEHBPolicy	Implemented	Implemented	Implemented
FERestartPolicy	Implemented	Implemented	Implemented
CEFailoverPolicy	Implemented	Implemented	Implemented
FEHACapab	Implemented	Implemented	Will Implement

FE Protocol LFB Datatypes

Protocol Components	NTT Japan	University of Patras	Zhejiang Gongshang University
CurrentRunningVersion	Implemented	Implemented	Implemented
FEID	Implemented	Implemented	Implemented
MulticastFEIDs	Implemented	Implemented	Implemented
CEHBPolicy	Implemented	Implemented	Implemented
CEHDI	Implemented	Implemented	Implemented
FEHBPolicy	Implemented	Implemented	Implemented
FEHI	Implemented	Implemented	Implemented
CEID	Implemented	Implemented	Implemented
BackupCEs	Implemented	Will Implement	Will Implement
CEFailoverPolicy	Implemented	Implemented	Implemented
CEFTI	Implemented	Implemented	Implemented
FERestartPolicy	Implemented	Implemented	Will Implement
LastCEID	Implemented	Implemented	Will Implement

FE Protocol LFB Components

Capabilities	NTT Japan	University of Patras	Zhejiang Gongshang University
SupportableVersions	Implemented	Implemented	Implemented
HACapabilities	Implemented	Implemented	Will Implement

Capabilities Supported

Events	NTT Japan	University of Patras	Zhejiang Gongshang University
PrimaryCEDown	Will Implement	Will Implement	Will Implement

Events Supported

6.1.2.3.2. FE Object LFB

Object Datatypes	NTT Japan	University of Patras	Zhejiang Gongshang University
LFBAdjacencyLimitType	Implemented	Implemented	Implemented
PortGroupLimitType	Implemented	Implemented	Implemented
SupportedLFBType	Implemented	Implemented	Implemented
FEStateValues	Implemented	Implemented	Implemented
FEConfiguredNeighborType	Implemented	Implemented	Implemented
LFBSelectorType	Implemented	Implemented	Implemented
LFBLinkType	Implemented	Implemented	Implemented

FE Object LFB Datatypes

Object Components	NTT Japan	University of Patras	Zhejiang Gongshang University
LFBTopology	Implemented	Implemented	Implemented
LFBSelectors	Implemented	Implemented	Implemented
FEName	Implemented	Implemented	Implemented
FEID	Implemented	Implemented	Implemented
FEVendor	Implemented	Implemented	Implemented
FEModel	Implemented	Implemented	Implemented
FEState	Implemented	Implemented	Implemented
FENeighbors	Implemented	Implemented	Implemented

FE Object LFB Components

Capabilities	NTT Japan	University of Patras	Zhejiang Gongshang University
ModifiableLFBTopology	Implemented	Implemented	Implemented
SupportedLFBs	Implemented	Implemented	Implemented

Capabilities Supported

6.1.3. ForCES SCTP TML Features

6.1.3.1. TML Priority Ports

Port	NTT Japan	University of Patras	Zhejiang Gongshang University
High priority (6700)	Implemented	Implemented	Implemented
Medium priority (6701)	Implemented	Implemented	Implemented
Low priority (6702)	Implemented	Implemented	Implemented

Priority Ports

6.1.3.2. Message Handling at Specific Priorities

ForCES Message	NTT Japan	University of Patras	Zhejiang Gongshang University
Association Setup	Implemented	Implemented	Implemented
Association Setup Response	Implemented	Implemented	Implemented
Association Teardown	Implemented	Implemented	Implemented
Config	Implemented	Implemented	Implemented
Config Response	Implemented	Implemented	Implemented
Query	Implemented	Implemented	Implemented
Query Response	Implemented	Implemented	Implemented

Message Handling at High-Priority (6700) Port

ForCES Message	NTT Japan	University of Patras	Zhejiang Gongshang University
Event Notification	Implemented	Implemented	Implemented

Message Handling at Medium-Priority (6701) Port

ForCES Message	NTT Japan	University of Patras	Zhejiang Gongshang University
Packet Redirect	Implemented	Implemented	Implemented
Heartbeat	Implemented	Implemented	Implemented

Message Handling at Low-Priority (6702) Port

6.1.3.3. TML Security Feature

Security Feature	NTT Japan	University of Patras	Zhejiang Gongshang University
IPsec	Will Implement	Will Implement	Will Implement

Security Feature Support

6.2. Interoperability Report

The interoperability test took place at the University of Patras, in the Department of Electrical and Computer Engineering.

There were two options for participation in the interoperability test.

1. Locally, on the University of Patras premises.

2. Remotely, via Internet.

Implementations from NTT and the University of Patras were present locally on the University of Patras premises in Greece, while the implementation from Zhejiang Gongshang University, which was behind a NAT, connected remotely from China.

The interoperability test validated the basic functionality of the ForCES protocol, mainly message exchanging and handling.

The following scenarios were tested.

6.2.1. Scenarios

The main goal of the interoperability test was to validate the basic protocol functionality; the test parameters were limited.

1. In the Association Setup message, all report messages were ignored.

2. In the Association Setup stage, the FEO OperEnable Event (FE to CE), Config FEO Adminup (CE to FE), and FEO Config-Resp (FE to CE) messages were ignored. The CEs assumed that the FEs were enabled once the LFB selectors had been queried.

3. Only FULLDATA-TLVs were used and not SPARSEDATA-TLVs.

4. There were no transaction operations.

5. Each message had only one LFBSelect-TLV, one OPER-TLV, and one PATH-DATA-TLV per message when these were used.

6.2.1.1. Scenario 1 - Pre-Association Setup

While the pre-association setup is not in the ForCES current scope, it is an essential step before CEs and FEs communicate. As the first part in a successful CE-FE connection, the participating CEs and FEs had to be configurable.

In the pre-association phase, the following configuration items were set up regarding the CEs:

- The CE ID.

- The FE IDs that were connected to this CE.

- The IP addresses of the FEs that connected to the CE.

- The TML priority ports.

In the pre-association phase, the following configuration items were set up regarding the FEs:

- The FE ID.
- The CE ID to which this FE was connecting.
- The IP address of the CE to which this FE was connecting.
- The TML priority ports.

6.2.1.2. Scenario 2 - TML Priority Channels Connection

For the interoperability test, the SCTP was used as TML. The TML connection with the associating element was needed for Scenario 2 to be successful.

SCTP TML [RFC5811] defines three priority channels, with specific ports:

- High priority - Port number: 6704
- Medium priority - Port number: 6705
- Lower priority - Port number: 6706

However, at the time of the interoperability test, the SCTP ports of the three priority channels were the following:

- High priority - Port number: 6700
- Medium priority - Port number: 6701
- Lower priority - Port number: 6702

As specified in Section 5, "Exceptions", this does not invalidate the results of the interoperability test.

6.2.1.3. Scenario 3 - Association Setup - Association Complete

Once the pre-association phase in the previous two scenarios had completed, CEs and FEs would be ready to communicate using the ForCES protocol and enter the Association Setup stage. In this stage, the FEs would attempt to join the NE. The following ForCES protocol messages would be exchanged for each CE-FE pair in the specified order:

- o Association Setup message (from FE to CE)
- o Association Setup Response message (from CE to FE)
- o Query message: FEO LFB selectors (from CE to FE)
- o Query Response: FEO LFB selectors response (from FE to CE)

6.2.1.4. Scenario 4 - CE Query

Once the Association Setup stage had completed, the FEs and CEs would enter the Established stage. In this stage, the FE will be continuously updated or queried. The CE should query the FE for a specific value from the FE Object LFB and from the FE Protocol LFB. An example from the FE Protocol LFB is the FE Heartbeat Interval (FEHI), and an example from the FE Object LFB is the state of the LFB (FEState).

The following ForCES protocol messages were exchanged:

- o Query message
- o Query Response message

6.2.1.5. Scenario 5 - Heartbeat Monitoring

The Heartbeat (HB) message is used for one ForCES element (FE or CE) to asynchronously notify one or more other ForCES elements in the same ForCES NE of its liveness. The default configuration of the Heartbeat Policy of the FE is set to 0, which means that the FE should not generate any Heartbeat messages. The CE is responsible for checking FE liveness by setting the PL header ACK flag of the message it sends to AlwaysACK. In this scenario, the CE will send a Heartbeat message with the ACK flag set to AlwaysACK, and the FE should respond.

The following type of ForCES protocol message was exchanged:

- o Heartbeat message

6.2.1.6. Scenario 6 - Simple Config Command

A Config message is sent by the CE to the FE to configure LFB components in the FE. A simple Config command, easily visible and metered, would be to change the Heartbeat configuration. This was done in two steps:

1. Change the FE Heartbeat Policy (FEHBPolicy) to value 1, to force the FE to send heartbeats.

2. After some heartbeats from the FE, the FE Heartbeat Interval (FEHI) was changed.

The following ForCES protocol messages were exchanged:

- o Config message

- o Config Response message

6.2.1.7. Scenario 7 - Association Teardown

In the end, the association must be terminated. There were three scenarios by which the association was terminated:

1. Normal teardown, by exchanging an Association Teardown message.

2. Irregular teardown, by stopping heartbeats from an FE or a CE.

3. Irregular teardown, by externally shutting down/rebooting an FE or a CE.

All scenarios were investigated in the interoperability test.

The following type of ForCES protocol message was exchanged:

- o Association Teardown message

6.2.2. Tested Features

The features that were tested are:

6.2.2.1. ForCES Protocol Features

6.2.2.1.1. Protocol Messages

Protocol Message
Association Setup
Association Setup Response
Association Teardown
Config
Config Response
Query
Query Response
Heartbeat

ForCES Protocol Messages

- PASS: All implementations handled the protocol messages, and all protocol analyzers captured them.

6.2.2.1.2. MainHeader Handling

```
+--------------------+
|    Header Field    |
+--------------------+
|     Correlator     |
|                    |
| ACK Indicator Flag |
|                    |
|    Priority Flag   |
+--------------------+
```

MainHeader Handling

- o PASS: All implementations handled these main header flags, and all protocol analyzers captured them.

6.2.2.1.3. TLV Handling

```
+---------------+
|      TLV      |
+---------------+
|  ASResult-TLV |
|               |
| ASTReason-TLV |
|               |
| LFBSelect-TLV |
|               |
|    OPER-TLV   |
|               |
| PATH-DATA-TLV |
|               |
|  FULLDATA-TLV |
|               |
|   RESULT-TLV  |
+---------------+
```

TLVs Supported

- o PASS: All implementations handled these TLVs, and all protocol analyzers captured them.

6.2.2.1.4. Operation Types Supported

Operation
SET
SET-RESPONSE
GET
GET-RESPONSE
REPORT

Operation Types Supported

- o PASS: All implementations handled these operations, and all protocol analyzers captured them.

6.2.2.1.5. ForCES Protocol Advanced Features

Feature
Batching
Heartbeats

ForCES Protocol Advanced Features

Although batching was not initially intended to be tested, it was assessed during the interoperability test.

- o PASS: Two implementations handled batching, and all handled heartbeats. The protocol analyzers captured both.

6.2.2.2. ForCES Model Features

6.2.2.2.1. Basic Atomic Types Supported

Atomic Type
uchar
uint32

Basic Atomic Types Supported

- o PASS: All implementations handled these basic atomic types.

6.2.2.2.2. Compound Types Supported

Compound Type
structs
arrays

Compound Types Supported

- o PASS: All implementations handled these compound types.

6.2.2.2.3. LFBs Supported

6.2.2.2.3.1. FE Protocol LFB

Protocol Datatypes
CEHBPolicy
FEHBPolicy

FE Protocol LFB Datatypes

- o PASS: All implementations handled these FE Protocol LFB datatypes.

```
+---------------------+
| Protocol Components |
+---------------------+
|         FEID        |
|                     |
|      CEHBPolicy     |
|                     |
|        CEHDI        |
|                     |
|      FEHBPolicy     |
|                     |
|         FEHI        |
|                     |
|         CEID        |
+---------------------+
```

FE Protocol LFB Components

o PASS: All implementations handled these FE Protocol LFB components.

6.2.2.2.3.2. FE Object LFB

```
+------------------+
| Object Datatypes |
+------------------+
|   FEStateValues  |
|                  |
|  LFBSelectorType |
+------------------+
```

FE Object LFB Datatypes

o PASS: All implementations handled these FE Object LFB datatypes.

```
+-------------------+
| Object Components |
+-------------------+
|    LFBSelectors   |
|                   |
|      FEState      |
+-------------------+
```

FE Object LFB Components

o PASS: All implementations handled these FE Object LFB components.

6.2.2.3. ForCES SCTP TML Features

6.2.2.3.1. TML Priority Ports

```
                        +------------------------+
                        |          Port          |
                        +------------------------+
                        |  High priority (6700)  |
                        |                        |
                        | Medium priority (6701) |
                        |                        |
                        |   Low priority (6702)  |
                        +------------------------+
```

Priority Ports

- PASS: All implementations opened and connected to all the SCTP priority ports. The protocol analyzers captured all ports and their corresponding priority.

6.2.2.3.2. Message Handling at Specific Priorities

```
                      +----------------------------+
                      |       ForCES Message       |
                      +----------------------------+
                      |      Association Setup     |
                      |                            |
                      | Association Setup Response |
                      |                            |
                      |    Association Teardown    |
                      |                            |
                      |           Config           |
                      |                            |
                      |       Config Response      |
                      |                            |
                      |            Query           |
                      |                            |
                      |       Query Response       |
                      +----------------------------+
```

Message Handling at High-Priority (6700) Port

- PASS: All implementations handled these messages at this SCTP priority port. The protocol analyzers captured these messages at this priority port.

```
+----------------+
| ForCES Message |
+----------------+
|   Heartbeats   |
+----------------+
```

Message Handling at Low-Priority (6702) Port

- PASS: All implementations handled these messages at this SCTP priority port. The protocol analyzers captured these messages at this priority port.

6.2.3. Interoperability Results

All implementations were found to be interoperable with each other.

All scenarios were tested successfully.

The following issues were found and dealt with.

1. Some messages were sent on the wrong priority channels. There were some ambiguities in the SCTP TML document regarding how to deal with such a situation. The possibilities were an FE response on the same (wrong) channel as a CE query; an FE response on the correctly documented channel for the message; or simply dropping the packet. This has been corrected by mandating the message-to-channel mapping to be a MUST in the SCTP TML document [RFC5811] before it was published as an RFC.

2. At some point, a CE sent a Teardown message to the FE. The CE expected the FE to shut down the connection, and the FE waited for the CE to shut down the connection; both were then caught in a deadlock. This was a code bug and was fixed.

3. Sometimes, only when the CE and FE were remote to each other (one being in China and another in Greece), the Association Setup message was not received by the CE side, and therefore an association never completed. This was not an implementation issue but rather a network issue. This issue was solved with the retransmission of the non-delivered messages.

4. An implementation did not take into account that the padding in TLVs MUST NOT be included in the length of the TLV. This was a code bug and was fixed.

5. The Execution Mode flag was set to Reserved by a CE and was not ignored by the FE. This was a code bug and was fixed.

6. After the FEHBPolicy was set to 1, the FE didn't send any heartbeats. This was a code bug and was fixed.

7. Some FEs sent heartbeats with the ACK flag set to a value other than NoACK. The CE responded. This was a code bug and was fixed.

8. When a cable was disconnected, none of the TML implementations detected it. The association was eventually dropped due to heartbeat detection; this test was a success, but this is an implementation issue that implementors should keep in mind. This is an SCTP options issue. Nothing needed to be done.

9. A CE crashed due to unknown LFB selector values. This was a code bug and was fixed.

10. With the remote connection from China (which was behind a NAT) to Greece, there were a lot of ForCES packet retransmissions. The problem was that packets like heartbeats were retransmitted. This was an implementation issue regarding SCTP usage that implementors should keep in mind. The SCTP-PR option needed to be used. Nothing needed to be done.

The interoperability test went so well that an additional extended test was added to check for batching messages. This test was also done successfully.

7. Acknowledgements

The authors would like to give thanks to Professors Odysseas Koufopavlou and Spyros Denazis, and the Department of Electrical and Computer Engineering at the University of Patras, who hosted the ForCES interoperability test.

The authors would also like to give thanks to Chuanhuang Li, Ming Gao, and other participants from Zhejiang Gongshang University, which connected remotely. This allowed the discovery of a series of issues that would have been uncaught otherwise.

The authors would also like to thank Hideaki Iwata and Yoshinobu Morimoto of NTT Japan for participating locally at the interoperability test; as well as Hiroki Date and Hidefumi Otsuka, also of NTT Japan, for contributing to the interoperability test.

Additionally, thanks are given to Xinping Wang for her help in writing the interoperability document and to Fenggen Jia for extending the Ethereal protocol analyzer.

8. Security Considerations

No security elements of the protocol or the SCTP TML [RFC5811] specification were tested.

The survey indicated that no security elements were implemented, but all participants indicated their intention to implement them.

For security considerations regarding the ForCES protocol and SCTP TML, please see [RFC5810] and [RFC5811].

9. References

9.1. Normative References

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

[RFC5810] Doria, A., Hadi Salim, J., Haas, R., Khosravi, H., Wang, W., Dong, L., Gopal, R., and J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification", RFC 5810, March 2010.

[RFC5811] Hadi Salim, J. and K. Ogawa, "SCTP-Based Transport Mapping Layer (TML) for the Forwarding and Control Element Separation (ForCES) Protocol", RFC 5811, March 2010.

[RFC5812] Halpern, J. and J. Hadi Salim, "Forwarding and Control Element Separation (ForCES) Forwarding Element Model", RFC 5812, March 2010.

9.2. Informative References

[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.

[RFC5657] Dusseault, L. and R. Sparks, "Guidance on Interoperation and Implementation Reports for Advancement to Draft Standard", BCP 9, RFC 5657, September 2009.

[ethereal] "Ethereal is a protocol analyzer. The specific Ethereal that was used is an updated Ethereal, by Fenggen Jia, that can analyze and decode the ForCES protocol messages", <http://www.ietf.org/mail-archive/web/forces/current/msg03687.html>.

[tcpdump] "tcpdump is a Linux protocol analyzer. The specific tcpdump that was used is a modified tcpdump, by Jamal Hadi Salim, that can analyze and decode the ForCES protocol messages", <http://www.ietf.org/mail-archive/web/forces/current/msg03811.html>.

Authors' Addresses

Evangelos Haleplidis
University of Patras
Patras
Greece

EMail: ehalep@ece.upatras.gr

Kentaro Ogawa
NTT Corporation
Tokyo
Japan

EMail: ogawa.kentaro@lab.ntt.co.jp

Weiming Wang
Zhejiang Gongshang University
18, Xuezheng Str., Xiasha University Town
Hangzhou, 310018
P.R. China

Phone: +86-571-28877721
EMail: wmwang@mail.zjgsu.edu.cn

Jamal Hadi Salim
Mojatatu Networks
Ottawa, Ontario
Canada

Phone:
EMail: hadi@mojatatu.com