ForCES WG
Internet-Draft
Intended status: Standards Track
Expires: November 17, 2015

B. Khasnabish
ZTE TX, Inc.
E. Haleplidis
University of Patras
J. Hadi Salim, Ed.
Mojatatu Networks
May 16, 2015

IETF ForCES Logical Function Block (LFB) Subsidiary Management
draft-ietf-forces-lfb-subsidiary-management-01

Abstract

Deployment experience has demonstrated the value of using the Forwarding and Control Element Separation (ForCES) architecture to manage resources other than packet forwarding. In that spirit, the Forwarding Element Manager (FEM) is modelled by creating a Logical Functional Block (LFB) to represent its functionality. We refer to this LFB as the Subsidiary Mechanism (SM) LFB. A Control Element (CE) that controls a Forwarding Element's (FE) resources can also manage its configuration via the SM LFB. This document introduces the SM LFB, an LFB that specifies the configuration parameters of an FE.

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 November 17, 2015.

Copyright Notice

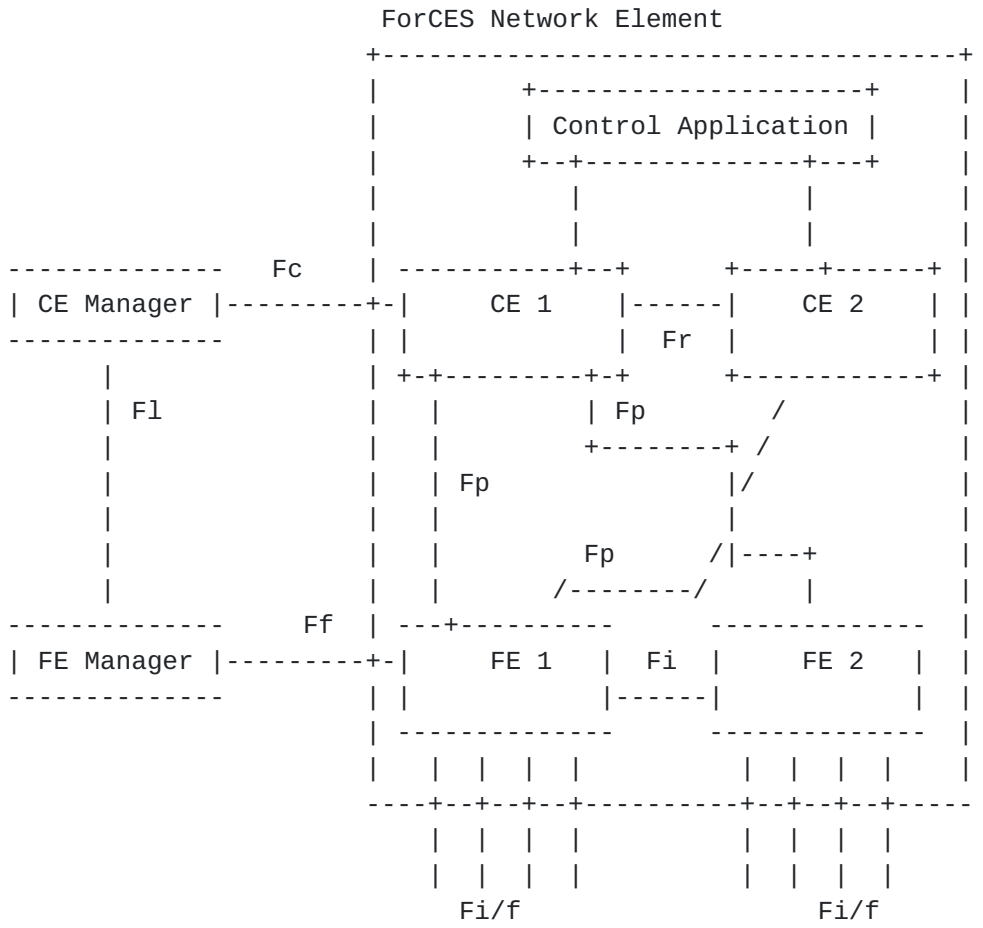
Table of Contents

1. Introduction 3
1.1. Requirements Language 5
1.2. Definitions 5
2. Use cases 6
2.1. High Availability 6
2.2. Scalability 6
2.3. Adding New Resources To An NE 6
2.4. New LFB class installation 6
2.5. Logging Mechanism 7
2.6. General Purpose Attribute Definition 7
3. Applicability statement 8
3.1. FE Integrated 8
3.2. Virtual FEs 8
4. SM Library 8
4.1. Frame Definitions 8
4.2. Datatype Definitions 8
4.3. Metadata Definitions 9
4.4. SM 9
4.4.1. Data Handling 9
4.4.2. Components 10
4.4.3. Capabilities 10
4.4.4. Events 10
5. XML for SM LFB 11
6. Security Considerations 17
7. IANA Considerations 17
7.1. LFB Class Names and LFB Class Identifiers 17
8. Acknowledgments 17
9. References 18
9.1. Normative References 18
9.2. Informative References 18
Authors' Addresses 18

1. Introduction

Deployment experience has demonstrated the value of using the Forwarding and Control Element Separation (ForCES) architecture to manage resources other than packet forwarding. In that spirit, the Forwarding Element Manager (FEM) is modelled by creating a Logical Functional Block (LFB) to represent its functionality. We refer to this LFB as the Subsidiary Mechanism (SM) LFB. A Control Element (CE) that controls a Forwarding Element's (FE) resources can also manage its configuration via the SM LFB. This document introduces the SM LFB, an LFB that specifies the configuration parameters of an FE.

On a running FE, a CE application may update an FE's runtime configuration via the SM LFB.



Fp: CE-FE interface
Fc: Interface between the CE Manager and a CE
Ff: Interface between the FE Manager and an FE
Fl: Interface between the CE Manager and the FE Manager
Fi/f: FE external interface

Figure 1: ForCES Architectural Diagram

Figure 1 shows a control application manipulating, at runtime, FE config via the SM LFB control. It would appear that that control application is playing the part of the FE Manager thus appears as the messaging for Ff (FEM to FE interface) going via the standard Fp plane. However the SM LFB describes a subset of the operations that can be performed over Ff; it does not suggest moving away from the Ff interface.

The SM LFB describes the configuration parameters of an FE, namely the LFB classes it should load, the CEs it should be associated with as well the respective CE IP addresses. Additionally the SM LFB provides a general purpose attribute definition to describe config

information, as well as the ability to manipulate debug logging mechanism.

This document assumes that FEs are already booted. The FE's configuration can then be updated at runtime via the SM LFB for runtime config purposes. This document does not specify or standardize the FEM-FE (Ff) interface as depicted in [RFC3746]. This document describes a mechanism with which a CE can instruct the SM for FE management using ForCES.

This work item makes no assumption of whether FE resources are physical or virtual. In fact, the LFB library provided here is applicable to both. Thus it can also be useful in addressing control of virtual FEs where individual FEM Managers can be addressed to control the creation, configuration, and resource assignment of such virtual FEs within a physical FE.

1.1. 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].

1.2. Definitions

This document follows the terminology defined by [RFC3654], [RFC3746], [RFC5810] and [RFC5812]. In particular, the reader is expected to be familiar with the following terms:

- Logical Functional Block (LFB)
- Forwarding Element (FE)
- Control Element (CE)
- ForCES Network Element (NE)
- FE Manager (FEM)
- CE Manager
- ForCES Protocol
- ForCES Protocol Layer (ForCES PL)
- ForCES Protocol Transport Mapping Layer (ForCES TML)

2. Use cases

In this section we present sample use cases to illustrate the need and usefulness of the SM LFB.

All use cases assume that an FE is already booted up and tied to at least one CE. A control application can delete a CE from an FE's table of CEs which instructs the FE to terminate the connection with that removed CE. Likewise, the control application via the master CE instructs an FE to establish a ForCES association with a new CE by adding a particular CE to the FE's CEs table.

2.1. High Availability

Assume an FE associated to only one CE. At runtime, a CE management application may request for redundancy reasons that an FE to be associated to another CE as a backup. To achieve this goal, the CE management application specifies the CEID of the new backup CE (to be uniquely identified within the NE) and the CE's IP address (IPv4 or IPv6).

2.2. Scalability

Assume an NE cluster that has FEs connected possibly in an active backup setup to multiple CEs. Assume that system analytics discover that the CE is becoming a bottleneck. A new CE could be be booted and some FEs moved to it. To achieve this goal, the CE management application will first ask an FE to connect to a new CE and would then instruct that FE to change its master to the new CE as described in [RFC7121].

2.3. Adding New Resources To An NE

Assume a resource pooling setup with multiple FEs belonging to a resource pool all connected to a dormant resource pool CE. An NE system manager by demand could move an FE from the resource pool to a working NE by asking it first to connect to a CE on the working NE and then asking it to disconnect from the resource pool manager CE.

2.4. New LFB class installation

A CE can learn via the capability of SM LFB whether an FE is capable of loading new LFB classes. Provided that the FE supports new LFB class loading, the CE can request a new LFB to be installed and supported by the FE.

To load an LFB class on an FE, the CE will have to provide the following parameters:

LFB class - The LFB class ID

LFB version - The version of the LFB class

LFB class name - Optional, the LFB name

Parameters - Optional parameters. These parameters are implementation specific, for example in one implementation they may contain the path where the LFB class implementation resides.

The parameter are fields which will be need to be described in documentation, depending on the implementation. As an example the location of the LFB Class to be installed and/or mechanism to download it. The exact detail of the location semantics is implementation specific and out of scope of this document. However this LFB library provides a placeholder, namely the SupportedParameters capability, which will host any standardized parameters. This document does not standardize these parameters. It is expected that some future document will perform that task.

2.5. Logging Mechanism

The SM LFB library also provides a useful log level manipulation. Experience has proven that the CE may require to increase or decrease the debug levels of parts of the FE, whether that be LFBs or portions of LFBs or generic processing code (all called modules). The module granularity is implementation specific and is not discussed in this document.

2.6. General Purpose Attribute Definition

Experience has shown that a generic attribute name-value pair is useful for describing config information. This LFB library defines such a generic attribute name-value pair defined as a table of attribute-name values. The attribute name-value pair is implementation specific and at the moment there is nothing to standardize. As an example consider switches which have exactly the same LFB classes and capabilities but needing to be used in different roles. A good example would be a switch which could be used either as Spine or ToR in data-centre setups. An attribute which defines the role could be retrieved from the FE which will then dictate how it is controlled/configured. However, as in the case of LFB class loading parameters this LFB class library provides a placeholder, namely the SupportedArguments capability, which will host any standardized arguments. This document does not standardize these parameters. It is expected that some future document(s) help standardize or define good practise of such attributes.

3. Applicability statement

Examples of SM usage are the following, but not limiting, two usage scenarios. These two, but not limiting, scenarios are not implementation details, but rather depict how the SM class can be used to achieve the intended subsidiary mechanism for manipulating the configuration of FEs.

3.1. FE Integrated

Only one instance of the SM class can exist and is directly related to the FE.

3.2. Virtual FEs

In the case of the FE software that has hierarchical virtual FEs, multiple instances of the SM class can exist, one per each virtual FE.

4. SM Library

4.1. Frame Definitions

This LFB does not define any frames

4.2. Datatype Definitions

This library defines the following datatypes.

DataType Name	Type	Synopsis
loglevels	An enumerated char based atomic datatype.	The possible debug log levels. Derived from syslog.
LogRowType	A struct containing three components. The LogModule (string), the optional ModuleFilename (string) and optional DebugLevel which is one of the enumerated loglevels.	The logging module row
CERow	A Struct that contains three components. The address family of the CE IP (uchar), the CE's IPs (octetstring[16] and the CE's ID (uint32)	A struct that defines the CE table row.
LCRowtype	A Struct that contains four components. The LFB Class ID (uint32), the LFB version (string[8]), the optional LFB Name (string) and optional Parameters (string).	The LFB Class Config Definition
NameVal	A Struct that contains two components. An attribute name (string) and an attribute value (string)	Arbitrary Name Value struct

FEM Data Types

4.3. Metadata Definitions

This LFB does not define any metadata definition

4.4. SM

The Subsidiary Mechanism LFB is an LFB that standardizes configuration of the FE parameters.

4.4.1. Data Handling

The SM LFB does not handle any packets. It's function is to provide the configuration parameters to the CE to be updated at runtime.

4.4.2. Components

This LFB has four components specified.

The Debug component (ID 1) is a table to support changing of an FE's module debug levels. Changes in an FE's debug table rows will alter the debug level of the corresponding module.

The LFBLoad component (ID 2) is a table of LFBs classes that the FE loads. Adding new rows in this table instructs the FE to load new LFB classes, and removing rows will unload them when possible. These two actions will in effect alter the SupportedLFBs capabilities table of FEObject LFB [RFC5812]. Each such row MUST provide (and is specified by this library) the LFB Class ID. Optionally the LFB class ID version may be specified, the FE MUST assume that version 1.0 is used when the version is unspecified.

The AttributeValues component (ID 3) is the AttributeValues table, a generic attribute-value pair.

The CEs (ID 4) is the table of runtime CEs we are asking the FE to be able to connect with. By adding a row in this table, the CE instructs the FE to be able to connect with the specified CE. By doing a delete on this table, the CE instructs the FE to terminate any connection with that CE. How the FE interacts with the new CEs is dependent on the operations discussed in [RFC7121]

It is worth noting that the generic attribute value pairs, the LFBload parameters and the module information are all strings. To cope with string sizes, a CE application can extract that information from the component properties as defined in [RFC5812]

4.4.3. Capabilities

This LFB provides three capabilities. The first, DynamicLFBLoading, specifies whether this FE supports dynamic loading of new LFB classes. The second, SupportedParameters, is a placeholder and will store all the supported parameters for LFB class loading. The final, SupportedAttributes, is also a placeholder and will store all the supported attributes for the attribute-value pair table.

4.4.4. Events

This LFB has four events specified.

Two events reflect CE additions and report to the CE whether an entry of the CEs information has been added or deleted. In both cases the event report constitutes the added or deleted row contents.

The other two events reflect LFB class loading and notify whether an entry of the LFBLoad table is added or deleted.

5. XML for SM LFB

```
<LFBLibrary xmlns="urn:ietf:params:xml:ns:forces:lfbmodel:1.1"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" provides="SM">
  <!-- XXX  -->
  <dataTypeDefs>
    <dataTypeDef>
      <name>loglevels</name>
      <synopsis>The possible debug log levels. Derived from syslog.
      </synopsis>
      <atomic>
        <baseType>char</baseType>
        <specialValues>
          <specialValue value="-1">
            <name>DEB_OFF</name>
            <synopsis> The logs are totally turned off </synopsis>
          </specialValue>
          <specialValue value="0">
            <name>DEB_EMERG</name>
            <synopsis> Emergency level </synopsis>
          </specialValue>
          <specialValue value="1">
            <name>DEB_ALERT</name>
            <synopsis> Alert level </synopsis>
          </specialValue>
          <specialValue value="2">
            <name>DEB_CRIT</name>
            <synopsis> Critical level </synopsis>
          </specialValue>
          <specialValue value="3">
            <name>DEB_ERR</name>
            <synopsis> error level </synopsis>
          </specialValue>
          <specialValue value="4">
            <name>DEB_WARNING</name>
            <synopsis> warning level </synopsis>
          </specialValue>
          <specialValue value="5">
            <name>DEB_NOTICE</name>
            <synopsis>Notice level </synopsis>
          </specialValue>
          <specialValue value="6">
            <name>DEB_INFO</name>
            <synopsis>Info level </synopsis>
          </specialValue>
```

```
            <specialValue value="7">
              <name>DEB_DEBUG</name>
              <synopsis>Debug level </synopsis>
            </specialValue>
          </specialValues>
        </atomic>
      </dataTypeDef>
      <dataTypeDef>
        <name>LogRowtype</name>
        <synopsis>The logging module row</synopsis>
        <struct>
          <component componentID="1">
            <name>lmodule</name>
            <synopsis>The LOG Module Name</synopsis>
            <typeRef>string</typeRef>
          </component>
          <component componentID="2">
            <name>filename</name>
            <synopsis>The Module File Name</synopsis>
            <optional/>
            <typeRef>string</typeRef>
          </component>
          <component componentID="3">
            <name>deblvl</name>
            <synopsis>debug level</synopsis>
            <optional/>
            <typeRef>loglevels</typeRef>
          </component>
        </struct>
      </dataTypeDef>
      <dataTypeDef>
        <name>CERow</name>
        <synopsis>The CE Table Row</synopsis>
        <struct>
          <component componentID="1">
            <name>AddressFamily</name>
            <synopsis>The address family</synopsis>
            <atomic>
              <baseType>uchar</baseType>
              <specialValues>
                <specialValue value="2">
                  <name>IFA_AF_INET</name>
                  <synopsis>IPv4</synopsis>
                </specialValue>
                <specialValue value="10">
                  <name>IFA_AF_INET6</name>
                  <synopsis>IPv6</synopsis>
                </specialValue>
```

```
            </specialValues>
          </atomic>
        </component>
        <component componentID="2">
          <name>CEIP</name>
          <synopsis>CE ip v4 or v6(selected by family)</synopsis>
          <typeRef>octetstring[16]</typeRef>
        </component>
        <component componentID="3">
          <name>CEID</name>
          <synopsis>The CE ID</synopsis>
          <optional/>
          <typeRef>uint32</typeRef>
        </component>
      </struct>
    </dataTypeDef>
    <dataTypeDef>
      <name>LCRowtype</name>
      <synopsis>The LFB Class Config Definition</synopsis>
      <struct>
        <component componentID="1">
          <name>LFBClassID</name>
          <synopsis>The LFB Class ID</synopsis>
          <typeRef>uint32</typeRef>
        </component>
        <component componentID="2">
          <name>LFBVersion</name>
          <synopsis>The LFB Class Version</synopsis>
          <optional/>
          <typeRef>string</typeRef>
        </component>
        <component componentID="3">
          <name>LFBName</name>
          <synopsis>The LFB Class Name</synopsis>
          <optional/>
          <typeRef>string</typeRef>
        </component>
        <component componentID="4">
          <name>Parameters</name>
          <synopsis>Optional parameters such as where the LFB is
          located</synopsis>
          <optional/>
          <typeRef>string</typeRef>
        </component>
      </struct>
    </dataTypeDef>
    <dataTypeDef>
      <name>NameVal</name>
```

```
        <synopsis>Arbitrary Name Value struct</synopsis>
        <struct>
          <component componentID="1">
            <name>AttrName</name>
            <synopsis>The Attribute Name</synopsis>
            <typeRef>string</typeRef>
          </component>
          <component componentID="2">
            <name>AttrVal</name>
            <synopsis>The Attribute Value</synopsis>
            <typeRef>string</typeRef>
          </component>
        </struct>
      </dataTypeDef>
    </dataTypeDefs>
    <LFBClassDefs>
      <LFBClassDef LFBClassID="19">
        <name>SM</name>
        <synopsis>
           The Subsidiary Management LFB
        </synopsis>
        <version>1.0</version>
        <components>
          <component componentID="1" access="read-write">
            <name>Debug</name>
            <synopsis>A table to support changing of all debug levels
            </synopsis>
            <array type="variable-size">
              <typeRef>LogRowtype</typeRef>
            </array>
          </component>
          <component componentID="2" access="write-only">
            <name>LFBLoad</name>
            <synopsis>An LFB Class to Load</synopsis>
            <array type="variable-size">
              <typeRef>LCRowtype</typeRef>
            </array>
          </component>
          <component componentID="3" access="read-write">
            <name>AttributeValues</name>
            <synopsis>Table of general purpose SM attribute Values
            </synopsis>
            <array type="variable-size">
              <typeRef>NameVal</typeRef>
            </array>
          </component>
          <component componentID="4" access="write-only">
            <name>CEs</name>
```

```
            <synopsis>Table of CEs we are asking the FE to associate
             with</synopsis>
            <array type="variable-size">
              <typeRef>CERow</typeRef>
            </array>
          </component>
        </components>
        <!---->
        <capabilities>
          <capability componentID="10">
            <name>DynamicLFBLoading</name>
            <synopsis>This capability specifies whether this FE supports
             dynamic loading of new LFBs</synopsis>
            <typeRef>boolean</typeRef>
          </capability>
          <capability componentID="11">
            <name>SupportedParameters</name>
            <synopsis>This capability contains all the supported
             parameters</synopsis>
            <array type="variable-size">
              <typeRef>string</typeRef>
            </array>
          </capability>
          <capability componentID="12">
            <name>SupportedAttributes</name>
            <synopsis>This capability contains all the supported
             attributes names</synopsis>
            <array type="variable-size">
              <typeRef>string</typeRef>
            </array>
          </capability>
        </capabilities>
        <events baseID="20">
          <event eventID="1">
            <name>CEAdded</name>
            <synopsis>An CE has been added</synopsis>
            <eventTarget>
              <eventField>CEs</eventField>
            </eventTarget>
            <eventCreated/>
            <eventReports>
              <eventReport>
                <eventField>CEs</eventField>
                <eventSubscript>_CEIDsrowid_</eventSubscript>
              </eventReport>
            </eventReports>
          </event>
          <event eventID="2">
```

```
            <name>CEDeleted</name>
            <synopsis>An CE has been deleted</synopsis>
            <eventTarget>
              <eventField>CEs</eventField>
              <eventSubscript>_CEIDsrowid_</eventSubscript>
            </eventTarget>
            <eventDeleted/>
            <eventReports>
              <eventReport>
                <eventField>CEs</eventField>
                <eventSubscript>_CEIDsrowid_</eventSubscript>
              </eventReport>
            </eventReports>
          </event>
          <event eventID="3">
            <name>LFBLoaded</name>
            <synopsis>An LFB has been loaded</synopsis>
            <eventTarget>
              <eventField>LFBLoad</eventField>
            </eventTarget>
            <eventCreated/>
            <eventReports>
              <eventReport>
                <eventField>LFBLoad</eventField>
                <eventSubscript>_LFBLoadrowid_</eventSubscript>
              </eventReport>
            </eventReports>
          </event>
          <event eventID="4">
            <name>LFBUnloaded</name>
            <synopsis>An CE has been unloaded</synopsis>
            <eventTarget>
              <eventField>LFBLoad</eventField>
              <eventSubscript>_LFBLoadrowid_</eventSubscript>
            </eventTarget>
            <eventDeleted/>
            <eventReports>
              <eventReport>
                <eventField>LFBLoad</eventField>
                <eventSubscript>_LFBLoadrowid_</eventSubscript>
              </eventReport>
            </eventReports>
          </event>
        </events>
      </LFBClassDef>
    </LFBClassDefs>
  </LFBLibrary>
```

Figure 2: FEM XML LFB library

6. Security Considerations

This document does not alter the ForCES Model [RFC5812] or the ForCES Protocol [RFC5810]. As such, it has no impact on their security considerations. This document simply defines the operational parameters and capabilities of an LFB that manages subsidiary mechanism for loading LFBs and create new connections between FEs and CEs. This document does not attempt to analyze the security issues that may arise from misuse of the SM LFB. Any such issues, if they exist, and mitigation strategies are for the designers of the particular SM implementation, not the general mechanism.

7. IANA Considerations

7.1. LFB Class Names and LFB Class Identifiers

LFB classes defined by this document belong to LFBs defined by Standards Track RFCs. According to IANA, the registration procedure is Standards Action for the range 0 to 65535 and First Come First Served with any publicly available specification for over 65535. This specification includes the following LFB class names and LFB class identifiers:

LFB Class Identifier	LFB Class Name	LFB Version	Description	Reference
19	SM	1.0	An SM LFB to standardize subsidiary management for ForCES Network Elements	This document

Logical Functional Block (LFB) Class Names and Class Identifiers

8. Acknowledgments

The authors would like to thank Damascene Joachimpillai, Joel Halpern, Chuanhuang Li, and many others for their discussions and support.

The authors are grateful to Joel Halpern for shepherding this document. The authors would also like to thank Alia Atlas for taking on the role of sponsoring this document.

9. References

9.1. Normative References

[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.

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

[RFC7121] Ogawa, K., Wang, W., Haleplidis, E., and J. Hadi Salim, "High Availability within a Forwarding and Control Element Separation (ForCES) Network Element", RFC 7121, February 2014.

9.2. Informative References

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

[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.

Authors' Addresses

Bhumip Khasnabish
ZTE TX, Inc.
55 Madison Avenue, Suite 160
Morristown, New Jersey 07960
USA

Phone: +001-781-752-8003
Email: vumip1@gmail.com, bhumip.khasnabish@ztetx.com
URI: http://tinyurl.com/bhumip/

Evangelos Haleplidis
University of Patras
Department of Electrical and Computer Engineering
Patras 26500
Greece

Email: ehalep@ece.upatras.gr

Jamal Hadi Salim (editor)
Mojatatu Networks
Suite 200, 15 Fitzgerald Rd,
Ottawa, Ontario K2H 9G1
Canada

Email: hadi@mojatatu.com