ForCES WG
Internet-Draft
Intended status: Standards Track
Expires: April 10, 2015

B. Khasnabish
ZTE TX, Inc.
E. Haleplidis
University of Patras
J. Hadi Salim, Ed.
Mojatatu Networks
October 7, 2014

IETF ForCES Logical Function Block (LFB) Subsidiary Management
draft-khs-forces-lfb-subsidiary-management-03.txt

Abstract

This document discusses ForCES Logical Function Block (LFB) Subsidiary Management (SM). Note that LFB SM is useful for introducing and supporting virtualization of ForCES Network Element (NE) including control Element (CE) and Forwarding Element (FE). The objective is to standardize an LFB to support this virtualization.

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 April 10, 2015.

Table of Contents

1. Introduction

This document discusses ForCES Logical Function Block (LFB) Subsidiary Management (SM). Note that LFB SM is useful for

introducing and supporting virtualization of ForCES Network Element (NE) including control Element (CE) and Network Element (NE).

Deployment experience has demonstrated the value of using ForCES to control the Forwarding Element Manager (FEM) by creating an LFB to represent its function using the same encoding rules as for any other LFB. This allows it to be controlled by the same Control Element (CE).

This work item assumes the presence of an initially booted FE whose configuration could then be updated at runtime via an FEM LFB for runtime config purposes (e.g., by adding a new CE and its associated IP address).

This work item 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. This work would result in a standards track LFB FEM library RFC.

1.1. Scope

The scope of this document is discussion (and standardization) of utilizing virtualized NEs (VNEs)for virtual CEs (VCEs) and virtual FEs (VFEs). Subsidiary management refers to utilizing (via e.g., proper bootstraping) VCEs and VFEs from the pools of these elements. The intention is to form pools of VCEs and VFEs rather than discussing the details of addition/deletion of VCE/VFE to these pools.

Note that the currently existing techniques and solutions may be either slow or not directly applicable to ForCES LFB subsidiary management.

1.2. Abbreviations

- API: Application Programming Interface
- CE: Control Element
- CEM: CE Manager
- CEV: CE Visor
- FE: Forwarding Element
- FEM: FE Manager

- FEV: FE Visor
- FEVM: FE Virtualization Manager
- ForCES: Forwarding and Control Element Separation
- LFB: Logical Functional Block
- NE: Network Element
- PL: Protocol Layer
- SFC: Service Function Chaining
- TML: Transport Mapping Layer
- VCE: Virtual CE
- VFE: Virtual FE
- VNE: Virtual NE

1.3. Conventions and Definitions

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].

The following definitions are taken from [7], [8] and [1]. They are repeated here for convenience as needed, but the normative definitions are found in the referenced RFCs:

- Logical Functional Block (LFB) -- A template that represents a fine-grained, logically separate aspects of FE processing. This is also an FB that standardizes and assists the creation of NEs.
- 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 by a CE via the ForCES Protocol.
- 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.
- ForCES Network Element (NE) - An entity composed of one or more CEs and one or more FEs. An NE usually hides its internal

organization from external entities and represents a single point of management to entities outside the NE.

- FE Manager (FEM) - A logical entity that operates in the pre-association phase and is responsible for determining to which CE(s) an FE should communicate. This process is called CE discovery and may involve the FE manager learning the capabilities of available CEs. This is also a logical entity that is responsible for generic FE management tasks such as the settings of TML protocol parameters and secure parameters; the functions of FEM are open and extensible.

- FE Virtualization Manager(FEVM) - This is a Virtualization Manager. Note that there is an inheritance relationship between FEM LFB and FEVM LFB.

- CE Manager - A logical entity that operates in the pre-association phase and is responsible for determining to which FE(s) a CE should communicate. This process is called FE discovery and may involve the CE manager learning the capabilities of available FEs.

- ForCES Protocol -- The protocol used for communication communication between CEs and FEs. This protocol does not apply to CE-to-CE communication, FE-to-FE communication, or to communication between FE and CE managers. The ForCES protocol is a master-slave protocol in which FEs are slaves and CEs are masters. This protocol includes both the management of the communication channel (e.g., connection establishment, heartbeats) and the control messages themselves.

- ForCES Protocol Layer (ForCES PL) -- A layer in the ForCES protocol architecture that defines the ForCES protocol messages, the protocol state transfer scheme, and the ForCES protocol architecture itself (including requirements of ForCES TML as shown below). Specifications of ForCES PL are defined in [1]

- 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 SCTP, IP, TCP, UDP, ATM, Ethernet, etc), and how to achieve and implement reliability, security, etc.

2. Use of Virtualized ForCES Elements

Virtualization of ForCES Elements allows efficient, scalable, and robust utilization of network control and transmission resources. Virtualization has been discussed (and deployed) widely in the

Computing Industry (e.g., server) in the context of efficient utilization of server resources.

As mentioned before, the currently existing techniques and solutions may be either slow or not directly applicable to ForCES LFB subsidiary management.

2.1. Use of Virtualized CEs

In this section we discuss the use of virtualized ForCES control elements (CEs). The resulting operating entities in virtualized environment are Virtual CEs of VCEs. The CE Visor (CEV) has the visibility to all of the VCEs in a domain, and can assign one of the VCEs as primary Master-VCE and another as secondary Master-VCE. CEV can dynamically manage the role of primary and secondary master-VCEs from a pool of VCEs.

2.2. Use of Virtualized FEs

In this section we discuss the use of virtualized ForCES forwarding elements (FEs). The resulting operating entities in virtualized environment are Virtual FEs of VFEs. The FE Visor (FEV) has the visibility to all of the VFEs in a domain, and can assign one of the VFEs as primary Master-VFE and another as secondary Master-VFE. FEV can dynamically manage the role of primary and secondary master-VFEs from a pool of VFEs.

2.3. Generic Lifecycle of Physical/Virtual Elements

The generic lifecycle of physical/virtual elements including NEs, FEs, VNEs, VCEs, VFEs, etc. consists of the following FOUR states:

- (a)Instantiation -- This refers to instantiation of CEs and FEs.
- (b) Association -- This refers to associating FEs to the CEs
- (c) Activation -- This refers to activation of CEs and FEs for normal operation. This state may include monitoring as well with an objective to satisfy both scaling and reliability requirements.
- (d) Release -- This refers to releasing resources (both physical and virtual elements) to the pool of available (that is un-assigned) elements, and reporting this to the appropriate (CE or FE) manager. It may be required to cleanse the physical/virtual elements before releasing in order to prevent harvesting of data/ information by the the next user of the CEs/FEs. The details of the cleansing operation is out of scope of this draft.

Figure 1 shows physical/virtual elements states and their transition.

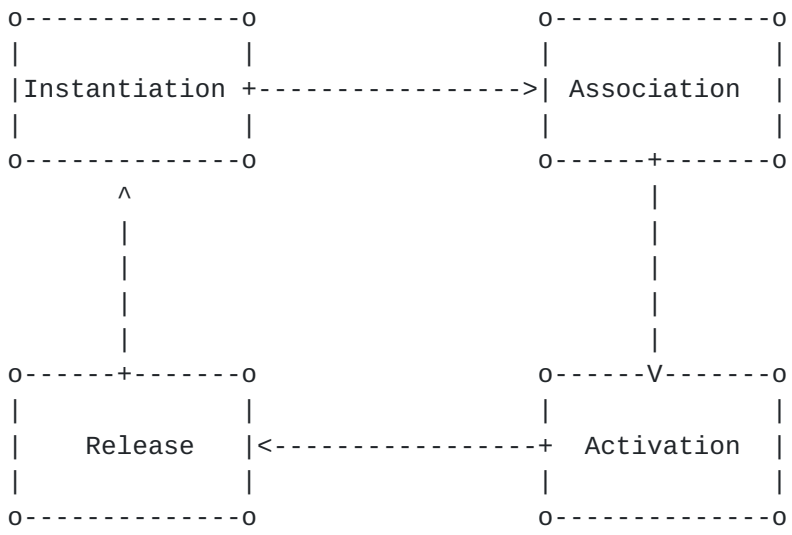

Figure 1: Physical/Virtual Elements States and their Transition

3. Potential Scenarios

In this section we discuss a few potential scenarios that can utilize ForCES LFB subsidiary management for efficient and robust operation of networks without using excessive additional resources.

3.1. Recovery from FE failure

In this section we discuss how virtualization of FEs can be used for efficient recovery from FE failure(s). An FE can initially boot using a default Association and Configuration. The Association and Configuration can be updated at runtime via an FE-Visor or FEM LFB for runtime configuration purposes. This can be achieved, for example, by adding a new CE and its associated IP address. A CE can initially boot using a default Configuration and State(s). The Configuration and State(s) can be updated at runtime via a CE-Visor or CEM LFB to satisfy the runtime requirements.

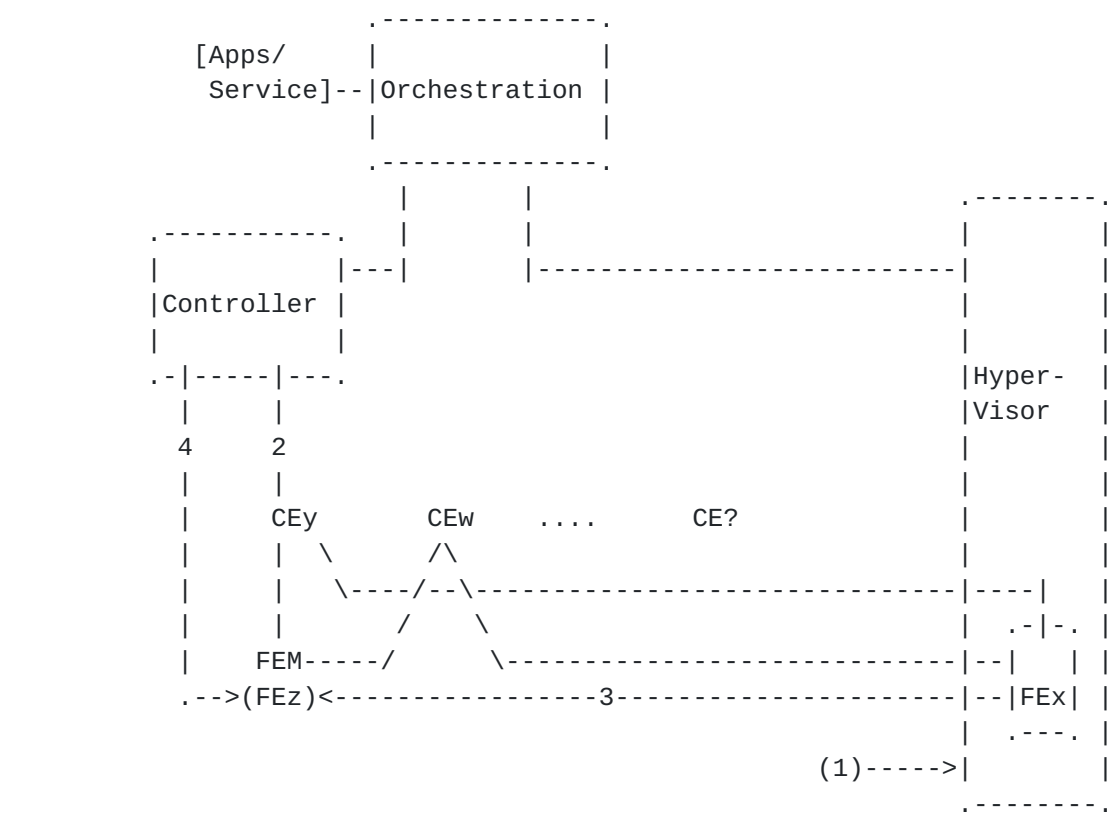

Figure 2: Sequence of Events in FEM for Recovery from FE Failure

Note: 1.(Hyervisor) Boots up FEx, and connects to CEy and CEw, 2. Boot a VM of Type FE 3. FEx Boots FEz, and Connects to CEy, 4.Connect to CEw

As described in Figure 2, the following is a sequence of events in FEM (an example).

- Step-1: Hypervisor boots up with FEx that connects to CEy and CEw.
- Step-2: The Controller (attached to CEy) instructs FEx to boot an FE-type VM.
- Step-3: FEx boots FEz and instructs it to connect to CEy
- Step-4: The Controller (attached to FEz) instructs FEz to also connect to CEw. This is essentially the "Association" part of Association and Configuration, as discussed earlier.

- Step-5: The Controller (attached to FEz) instructs FEz to increase its Syslog debug level. This is essentially the "Configuration" part of Association and Configuration, as discussed earlier.

Note that the 4th (FEM part of the charter) and 5th steps are what we would like to achieve here. In addition, the FEVM may not need to be aware which Virtual FEs are in one Virtual NE, it only needs know of the information about a Virtual FE in the physical FE. CE Manager may need to have visibility to all Virtual NEs. The component "NE" of the LFB may be considered as Virtual NE as well.

3.2. Recovery from CE failure

In this section we discuss how virtualization of CEs can be used for efficient recovery from CE failure(s).

A CE can initially boot using a default Association, State, and Configuration. The Association and Configuration can be updated at runtime via a CE-Visor or CEM-LFB for runtime configuration purposes, for example, by adding a new CE and its associated IP address.

An FE can initially boot using a default Configuration, Association, with a CE, and State. The Configuration, Association can be updated at runtime via a FE-Visor or FEM LFB to satisfy runtime requirements. The sequence of events, an example, can be as follows.

- Step-1: The CEx is Active with CEy as its Standby with Standby-Active or Active-Active setup.

- Step-2: The CEx controls FEy and FEw with both FEy and FEw having Standby control links to CEy, with Standby-Active or Active-Active setup. Note that CEx and CEy are controlled, assigned, by CE-visor, and may have a common, virtual, IP address.

- Step-3: The Controller is fully aware of the status of all of the CEs, physical and virtual; When CEx fails, its states are fully transferred (may already be synced) to CEy.

- Step-4: The Standby links from CEy to FEy and FEw become fully active, and the control, of FEy and FEw, is fully transferred from CEx and CEy.

- Step-5: A graceful-smooth failover of CEx to CEy is now successfully complete, and SysLog debug level for CEy is increased..

As discussed earlier, the last two steps are concerned with Subsidiary management. Although we discuss the recovery method by

using virtualization of CEs, the role of FEVM in the recovery process will be described further later.

3.3. Load Balancing

In this section we discuss efficient load balancing of both CE and FE in virtualized environment.

3.4. Scalable/Robust Service Function Chaining

In this section we discuss how LFB subsidiary management can contribute to the robust/scalable implementation of Service Function Chaining (SFC).

3.5. Orchestration

In this section we discuss efficient Orchestration of both CE and FE in virtualized multi-admin-domain environment.

3.6. Generic LFB Lifecycle Management

In this section we discuss generic lifecycle management of subsidiaries of LFBs in virtualized environment(s). The typical management activities in the life of FE/CE are discussed in the following sub-sections.

3.6.1. Booting a CE/FE

When an entity needs to boot a CE/FE, if this is a VM, some orchestration would scheme/plan to do this. In case of ForCES, we have a control App that boots a CE or an FE via a management FE. So here we have a management plane details that is described either in FEM or other LFB.

3.6.2. Bootstrapping the Configuration

The FE, e.g., the VM which has just been booted, as described in the previous sub-section, needs initial bootstrap configuration (e.g., what CEs to connect to etc). This clearly falls in the FEM LFB domain.

3.6.3. Runtime Management

At runtime of the FE, for example, the management could introduce a new CE for the FE to associate with; it may also be for an FE to dissociate from a CE, and so on.

4. FEM 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
IPs	A Struct of 2 components. IPv4 (byte[4]) and IPv6 (byte[16]) addresses.	A struct that defines an IPv4 and an IPv6 address
LFBDefs	A Struct that contains three components. The LFB Class ID (uint32), the LFB version (string)and the LFB name (string)	A struct that defines basic LFB definitions
CEParams	A Struct that contains two components. A CE's ID (uint32) and the CE's IPs (array of IPs)	A struct that defines CE parameters.
FEParams	A Struct that contains four components. An FE's ID (uint32), the FE's IPs (array of IPs), the LFBs this FE supports (array of LFBDefs) and the CEs this FE is part of (array of CEParams).	A struct that defines the FE parameters.

FEM Data Types

4.3. Metadata Definitions

This LFB does not define any metadata definition

4.4. FEM

The LFB is an LFB that standardizes and assists creation of NEs.

4.4.1. Data Handling

The FEM LFB does not handle any packets. It's function is to subsidize creation of NEs. A CE or a CEM will request from the FEM the creation of the NE, it will provide the requirements, e.g. FEs, LFBs in FEs etc..., and depending on the implementation the FEM may create these FE instances, or if these instances exist, provide the bootstrap information to FEs how and where to connect to the CEs.

4.4.2. Components

This LFB has only one component specified. The NEs component, is a component that contains all the Network Elements this FEM is responsible for maintaining. It is a table and each row is a struct of the NEID, a uint32 and an array of FE parameters.

4.4.3. Capabilities

This LFB has no Capabilities specified.

4.4.4. Events

This LFB has three events specified. These events notify the CE whether an NE has been added, deleted or changed. The event report is the NEs row that created the event.

5. XML for FEM LFB

```
<?xml version="1.0" encoding="UTF-8"?>
<LFBLibrary xmlns="urn:ietf:params:xml:ns:forces:lfbmodel:1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" provides="FEM">
   <load library="FEPO"/>
   <dataTypeDefs>
      <dataTypeDef>
         <name>IPs</name>
         <synopsis>IP definition</synopsis>
         <struct>
            <component componentID="1">
               <name>FEIPv4</name>
               <synopsis>The FEs IPv4</synopsis>
               <typeRef>byte[4]</typeRef>
            </component>
            <component componentID="2">
               <name>FEIPv6</name>
               <synopsis>The FEs IPv6</synopsis>
               <typeRef>byte[16]</typeRef>
            </component>
         </struct>
```

```
      </dataTypeDef>
      <dataTypeDef>
         <name>LFBDefs</name>
         <synopsis>LFB parameters inside the FE</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 Version of the LFB</synopsis>
               <typeRef>string</typeRef>
            </component>
            <component componentID="3">
               <name>LFBName</name>
               <synopsis>The name of the LFB</synopsis>
               <optional/>
               <typeRef>string</typeRef>
            </component>
         </struct>
      </dataTypeDef>
      <dataTypeDef>
         <name>CEParams</name>
         <synopsis>CE parameters</synopsis>
         <struct>
            <component componentID="1">
               <name>CEID</name>
               <synopsis>The CE ID</synopsis>
               <typeRef>uint32</typeRef>
            </component>
            <component componentID="2">
               <name>CEIP</name>
               <synopsis>The CEIP</synopsis>
               <array>
                  <typeRef>IPs</typeRef>
               </array>
            </component>
         </struct>
      </dataTypeDef>
      <dataTypeDef>
         <name>FEParams</name>
         <synopsis>FE parameters</synopsis>
         <struct>
            <component componentID="1">
               <name>FEID</name>
               <synopsis>The FEID</synopsis>
```

```
            <typeRef>uint32</typeRef>
         </component>
         <component componentID="2">
            <name>FEIP</name>
            <synopsis>The FE's IP</synopsis>
            <array>
                <typeRef>IPs</typeRef>
            </array>
         </component>
         <component componentID="3">
            <name>LFBparameters</name>
            <synopsis>The LFBs in this FE</synopsis>
            <array>
               <typeRef>LFBDefs</typeRef>
            </array>
         </component>
         <component componentID="4">
            <name>CEs</name>
            <synopsis>The CEs that should be associated with this
               FE</synopsis>
            <array>
               <typeRef>CEParams</typeRef>
            </array>
         </component>
      </struct>
   </dataTypeDef>
</dataTypeDefs>
<LFBClassDefs>
   <LFBClassDef LFBClassID="21">
      <name>FEM</name>
      <synopsis>The Forwarding Element Manager LFB</synopsis>
      <version>1.0</version>
      <components>
         <component componentID="1" access="read-write">
            <name>NEs</name>
            <synopsis>All the Network Elements this FEM is
               responsible for maintaining</synopsis>
            <array>
               <struct>
                  <component componentID="1">
                     <name>NEID</name>
                     <synopsis>ID of the Network Element</synopsis>
                     <typeRef>uint32</typeRef>
                  </component>
                  <component componentID="2">
                     <name>FEs</name>
                     <synopsis>FEs in the Network Element
                     </synopsis>
```

```
                        <array>
                           <typeRef>FEParams</typeRef>
                        </array>
                     </component>
                  </struct>
               </array>
            </component>
         </components>
         <events baseID="10">
            <event eventID="1">
               <name>NEchanged</name>
               <synopsis>The NE definition has changed</synopsis>
               <eventTarget>
                  <eventField>NEs</eventField>
               </eventTarget>
               <eventChanged/>
               <eventReports>
                  <eventReport>
                     <eventField>NEs</eventField>
                     <eventSubscript>_NEsrowid_</eventSubscript>
                  </eventReport>
               </eventReports>
            </event>
            <event eventID="2">
               <name>NEcreated</name>
               <synopsis>An NE has been created</synopsis>
               <eventTarget>
                  <eventField>NEs</eventField>
               </eventTarget>
               <eventCreated/>
               <eventReports>
                  <eventReport>
                     <eventField>NEs</eventField>
                     <eventSubscript>_NEsrowid_</eventSubscript>
                  </eventReport>
               </eventReports>
            </event>
            <event eventID="3">
               <name>NEdeleted</name>
               <synopsis>An NE has been deleted</synopsis>
               <eventTarget>
                  <eventField>NEs</eventField>
               </eventTarget>
               <eventDeleted/>
               <eventReports>
                  <eventReport>
                     <eventField>NEs</eventField>
                     <eventSubscript>_NEsrowid_</eventSubscript>
```

```
                </eventReport>
              </eventReports>
            </event>
          </events>
        </LFBClassDef>
    </LFBClassDefs>
</LFBLibrary>
```

Figure 1: FEM XML LFB library

6. Security Considerations

Security considerations for ForCES LFB subsidiary management will be added in a future version of this daft.

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
21	FEM	1.0	An FEM LFB to standardize creation of ForCES Network Elements	This document

Logical Functional Block (LFB) Class Names and Class Identifiers

8. Acknowledgments

The authors would like to thank DJ, Joel, ChuanhuangLi, and many others for their discussions and support.

9. References

9.1. Normative References

[1] 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.

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

[3] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

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

[5] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.

9.2. Informative References

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

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

[8] 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 400, 303 Moodie Dr.
Ottawa, Ontario K2H 9R4
Canada

Email: hadi@mojatatu.com