

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 26, 2013

E. Haleplidis
O. Koufopavlou
S. Denazis
University of Patras
October 23, 2012

**Virtualization of the Forwarding Plane Devices with ForCES
draft-haleplidis-forces-virtualization-01**

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). [RFC5812](#) has defined the ForCES Model provides a formal way to represent the capabilities, state, and configuration of forwarding elements within the context of the ForCES protocol, so that control elements (CEs) can control the FEs accordingly. More specifically, the model describes the logical functions that are present in an FE, what capabilities these functions support, and how these functions are or can be interconnected.

The ForCES model provides the necessary abstractions to natively support virtualization of the forwarding plane. This documents describes a formal approach to model the necessary parameters required for defining and managing virtual network forwarding planes to create virtual network elements.

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 26, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Terminology and Conventions	3
1.1.	Requirements Language	3
1.2.	Definitions	3
2.	Introduction	5
3.	Virtualization	6
4.	Virtualization Base Types	8
4.1.	Frame Types	8
4.2.	Data Types	8
4.3.	MetaData Types	8
5.	Virtualization LFBs	9
5.1.	vFE	9
5.1.1.	Data Handling	9
5.1.2.	Components	9
5.1.3.	Capabilities	9
5.1.4.	Events	9
6.	XML for Virtual LFB library	10
7.	Acknowledgements	15
8.	IANA Considerations	16
9.	Security Considerations	17
10.	References	18
10.1.	Normative References	18
10.2.	Informative References	18
	Authors' Addresses	19

1. Terminology and Conventions

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 the ForCES Model in [[RFC5812](#)]. The required definitions are repeated below for clarity.

FE Model - The FE model is designed to model the logical processing functions of an FE. The FE model proposed in this document includes three components; the LFB modeling of individual Logical Functional Block (LFB model), the logical interconnection between LFBs (LFB topology), and the FE-level attributes, including FE capabilities. The FE model provides the basis to define the information elements exchanged between the CE and the FE in the ForCES protocol [[RFC5810](#)].

LFB (Logical Functional Block) Class (or type) - A template that represents a fine-grained, logically separable aspect of FE processing. Most LFBs relate to packet processing in the data path. LFB classes are the basic building blocks of the FE model.

LFB Instance - As a packet flows through an FE along a data path, it flows through one or multiple LFB instances, where each LFB is an instance of a specific LFB class. Multiple instances of the same LFB class can be present in an FE's data path. Note that we often refer to LFBs without distinguishing between an LFB class and LFB instance when we believe the implied reference is obvious for the given context.

LFB Model - The LFB model describes the content and structures in an LFB, plus the associated data definition. XML is used to provide a formal definition of the necessary structures for the modeling. Four types of information are defined in the LFB model. The core part of the LFB model is the LFB class definitions; the other three types of information define constructs associated with and used by the class definition. These are reusable data types, supported frame (packet) formats, and metadata.

Element - Element is generally used in this document in accordance with the XML usage of the term. It refers to an XML tagged part of an XML document. For a precise definition, please see the full set of XML specifications from the W3C. This term is included in

this list for completeness because the ForCES formal model uses XML.

Attribute - Attribute is used in the ForCES formal modeling in accordance with standard XML usage of the term, i.e., to provide attribute information included in an XML tag.

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, but not how the per-packet state is implemented within actual hardware. Metadata is sent between the FE and the CE on redirect packets.

ForCES Component - A ForCES Component is a well-defined, uniquely identifiable and addressable ForCES model building block. A component has a 32-bit ID, name, type, and an optional synopsis description. These are often referred to simply as components.

LFB Component - An LFB component is a ForCES component that defines the Operational parameters of the LFBs that must be visible to the CEs.

LFB Class Library - The LFB class library is a set of LFB classes that has been identified as the most common functions found in most FEs and hence should be defined first by the ForCES Working Group.

2. Introduction

Forwarding plane virtualization is one key ingredient in creating a fully virtualized environment for data centers. One of the main requirements for virtualizing the forwarding plane is to create a complete set of abstractions that can be mapped to the physical devices. The ForCES Model [[RFC5812](#)] is such an abstraction as it presents a formal way to describe the Forwarding Plane's datapath with Logical Function Blocks (LFBs) using XML. This document describes a formal approach to model the necessary parameters required for defining and managing a virtual network forwarding plane. Control Elements virtual or physical can be associated with ForCES protocol to the virtual FEs and create a virtual network element.

3. Virtualization

LFBs are abstraction of the forwarding plane therefore they can be also used as abstractions of the virtual forwarding plane as well.

How a device is exactly virtualized is out of scope of this document and is considered implementation specific. However an example is shown in Figure 1 where distinct and isolated topologies of LFB instances inside an FE can be virtualiza a physical FE.

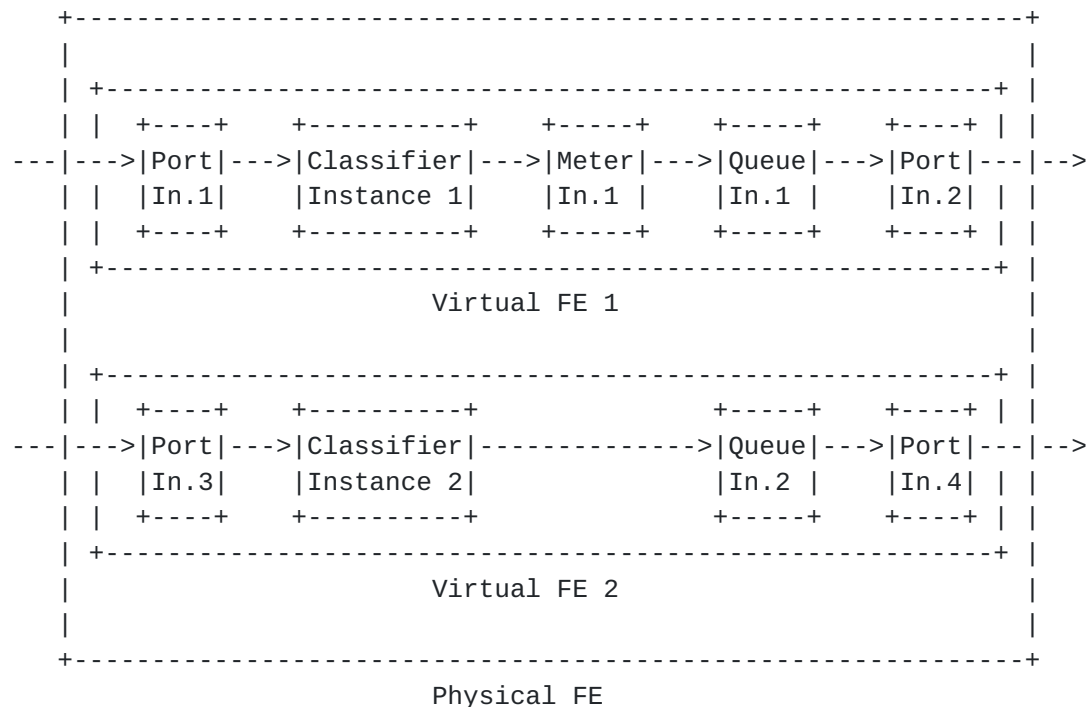


Figure 1: Isolated LFB instances

This document focuses on the definition of an LFB that will allow a CE to deploy and manage virtual FEs. In this approach we try to define parameters of a Virtual Network Element Manager (VNEM), what is commonly called a hypervisor therefore treating it as an FE, in order to be managed by a virtual management software, in this case a CE.

The VNEM in the ForCES model can be a joined Control Element Manager and a Forwarding Element Manager which defines which CEs or vCEs connect to which FEs or vFEs. What is required therefore of this document is a way to define resource allocation to a vFE and the topology of the FE or vFEs. This document introduces a new LFB, called vFE which contains the following details for one tenant of the network:

1. TenantID
2. FEs and resource allocation per FE.
3. FETopology

It is expected that there is one instance of the vFE LFB per tenant.

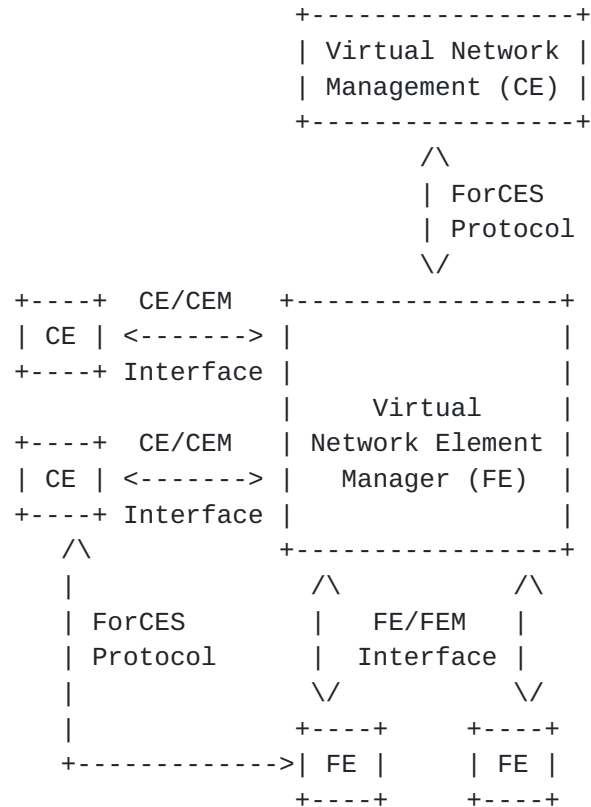


Figure 2: Virtual Network Elements

The Virtual Network Management is able to describe and instantiate FE topologies and assign CEs to control them. The CEs will be able to be configured via the CE/CEM interface and the FEs by the FE/FEM interface

4. Virtualization Base Types

4.1. Frame Types

No frame types has been defined in this library.

4.2. Data Types

TBD

4.3. MetaData Types

No metadata types have been defined in this library.

5. Virtualization LFBs

5.1. vFE

The vFE LFB holds information regarding a tenant in a virtual network device

5.1.1. Data Handling

The vFE LFB does not handle any data. It is similar to the core LFBs, FEObject and FEProtocolObject. It is expected to be one vFE LFB per tenant.

5.1.2. Components

The following components have been defined for this FE:

1. FETopology - The Topology of the FEs. From a FE, To an FE, via port and the link allocation between them.
2. FEs - The FEs supported by this vFE
3. CEs - The CEs, master and backup to control the FEs.
4. TenantID - The tenant ID for this vFE.

5.1.3. Capabilities

The following two capabilities have been defined:

1. ModifiableFETopology - Whether the FE topology is modifiable.
2. SupportedFES - The FEs that are supported by this topology.

5.1.4. Events

This LFB has no events specified.

6. XML for Virtual LFB library

```
<?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"
  xsi:schemaLocation="urn:ietf:params:xml:ns:forces:lfbmodel:1.0
  provides="vFE">
  <load library="BaseTypeLibrary"/>
  <dataTypeDefs>
    <dataTypeDef>
      <name>PercentageType</name>
      <synopsis>A datatype that defines a percentage
      </synopsis>
      <atomic>
        <baseType>uchar</baseType>
        <rangeRestriction>
          <allowedRange min="1" max="100"/>
        </rangeRestriction>
      </atomic>
    </dataTypeDef>
    <dataTypeDef>
      <name>FEAdjacencyLimitType</name>
      <synopsis>Describing the Adjacent FE</synopsis>
      <struct>
        <component componentID="1">
          <name>NeighborLFB</name>
          <synopsis>FE ID for that FE</synopsis>
          <typeRef>uint32</typeRef>
        </component>
        <component componentID="2">
          <name>ViaPorts</name>
          <synopsis>the ports on which we can connect
          </synopsis>
          <array>
            <typeRef>string</typeRef>
          </array>
        </component>
      </struct>
    </dataTypeDef>
    <dataTypeDef>
      <name>SupportedFEType</name>
      <synopsis>Table entry for supported FEs</synopsis>
      <struct>
        <component componentID="1">
          <name>FEName</name>
          <synopsis>The name of a supported FE</synopsis>
          <typeRef>string</typeRef>
```



```
</component>
<component componentID="2">
  <name>FEID</name>
  <synopsis>The id of a supported FE</synopsis>
  <typeRef>uint32</typeRef>
</component>
<component componentID="3">
  <name>CanOccurAfters</name>
  <synopsis>List of FEs that this FE class can follow
</synopsis>
  <optional/>
  <array>
    <typeRef>FEAdjacencyLimitType</typeRef>
  </array>
</component>
<component componentID="4">
  <name>CanOccurBefore</name>
  <synopsis>List of FEs that this FE class can follow
</synopsis>
  <optional/>
  <array>
    <typeRef>FEAdjacencyLimitType</typeRef>
  </array>
</component>
</struct>
</dataTypeDef>
<dataTypeDef>
  <name>FELinkType</name>
  <synopsis>Link between two FEs</synopsis>
  <struct>
    <component componentID="1">
      <name>FromFEID</name>
      <synopsis>FE source</synopsis>
      <typeRef>uint32</typeRef>
    </component>
    <component componentID="2">
      <name>ToFEID</name>
      <synopsis>FE destination</synopsis>
      <typeRef>uint32</typeRef>
    </component>
    <component componentID="3">
      <name>ViaPorts</name>
      <synopsis>The interfaces on which the FEs connect
</synopsis>
      <array>
        <typeRef>string</typeRef>
      </array>
    </component>
  </struct>
</dataTypeDef>
```



```
<component componentID="4">
  <name>LinkAllocation</name>
  <synopsis>Percentage of allowed Link usage</synopsis>
  <typeRef>PercentageType</typeRef>
</component>
</struct>
</dataTypeDef>
<dataTypeDef>
  <name>FEType</name>
  <synopsis>An FE inside a virtual forwarding element topology
</synopsis>
  <struct>
    <component componentID="1">
      <name>FEID</name>
      <synopsis>ID of the FE</synopsis>
      <typeRef>uint32</typeRef>
    </component>
    <component componentID="2">
      <name>ResourceAllocation</name>
      <synopsis>Resource Allocation for this FE
</synopsis>
      <struct>
        <component componentID="1">
          <name>Storage</name>
          <synopsis>Storage allocation of this FE
</synopsis>
          <optional/>
          <typeRef>PercentageType</typeRef>
        </component>
        <component componentID="2">
          <name>Memory</name>
          <synopsis>Memory allocation of this FE
</synopsis>
          <optional/>
          <typeRef>PercentageType</typeRef>
        </component>
        <component componentID="3">
          <name>Compuutation</name>
          <synopsis>Computation allocation of this FE
</synopsis>
          <optional/>
          <typeRef>PercentageType</typeRef>
        </component>
        <component componentID="4">
          <name>Bandwidth</name>
          <synopsis>Bandwidth allocation of this FE
</synopsis>
          <optional/>
```



```

        <typeRef>PercentageType</typeRef>
      </component>
    </struct>
  </component>
</struct>
</dataTypeDef>
</dataTypeDefs>
<LFBClassDefs>
  <LFBClassDef LFBClassID="18">
    <name>vFE</name>
    <synopsis>Core LFB:FE Object</synopsis>
    <version>1.0</version>
    <components>
      <component componentID="1" access="read-write">
        <name>FETopology</name>
        <synopsis>The table of known topologies</synopsis>
        <array type="Variable-size">
          <typeRef>FELinkTYpe</typeRef>
        </array>
      </component>
      <component componentID="2" access="read-write">
        <name>FEs</name>
        <synopsis>table of FEs</synopsis>
        <array type="Variable-size">
          <typeRef>FEType</typeRef>
        </array>
      </component>
      <component componentID="3" access="read-write">
        <name>CEs</name>
        <synopsis>table of CEs</synopsis>
        <array>
          <struct>
            <component componentID="1">
              <name>CEID</name>
              <synopsis>The CEID</synopsis>
              <typeRef>uint32</typeRef>
            </component>
            <component componentID="2">
              <name>CEType</name>
              <synopsis>Master or backup</synopsis>
              <atomic>
                <baseType>uchar</baseType>
                <specialValues>
                  <specialValue value="0">
                    <name>Master</name>
                    <synopsis>This CE is the master
                  </synopsis>
                </specialValue>
              </atomic>
            </component>
          </struct>
        </array>
      </component>
    </components>
  </LFBClassDef>
</LFBClassDefs>

```



```

        <specialValue value="1">
            <name>Backup</name>
            <synopsis>This CE is a backup
            </synopsis>
        </specialValue>
    </specialValues>
</atomic>
</component>
</struct>
</array>
</component>
<component componentID="4" access="read-write">
    <name>TenantID</name>
    <synopsis>The tenant ID of this virtual topology of
    FEs</synopsis>
    <typeRef>uint32</typeRef>
</component>
</components>
<capabilities>
    <capability componentID="30">
        <name>ModifiableFETopology</name>
        <synopsis>Whether Modifiable FE topology is supported
        </synopsis>
        <typeRef>boolean</typeRef>
    </capability>
    <capability componentID="31">
        <name>SupportedFEs</name>
        <synopsis>List of all supported FEs</synopsis>
        <array type="Variable-size">
            <typeRef>uint32</typeRef>
        </array>
    </capability>
</capabilities>
</LFBClassDef>
</LFBClassDefs>
</LFBLibrary>

```

Figure 3: Parallel LFB library

[7.](#) Acknowledgements

TBD

8. IANA Considerations

This memo includes no request to IANA.

9. Security Considerations

10. References

10.1. 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.

10.2. Informative References

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

Authors' Addresses

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

Email: ehalep@ece.upatras.gr

Odysseas Koufopavlou
University of Patras
Department of Electrical and Computer Engineering
Patras, 26500
Greece

Email: odysseas@ece.upatras.gr

Spyros Denazis
University of Patras
Department of Electrical and Computer Engineering
Patras, 26500
Greece

Email: sdena@upatras.gr

