                 Forwarding and Control Element Separation (ForCES)
                 Logical Functional Block (LFB) Subsidiary Management

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 modeled 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 class, an LFB class that specifies the configuration
   parameters of an FE. The configuration parameters include new LFB
   class loading and CE associations; they also provide manipulation of
   debug mechanisms along with a general purpose attribute definition to
   describe configuration information.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7729.

Copyright Notice

Table of Contents

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 modeled 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 class, 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 instance.

```
                           ForCES Network Element
                        +-------------------------------------+
                        |       +---------------------+       |
                        |       | Control Application |       |
                        |       +--+--------------+---+       |
                        |          |              |           |
                        |          |              |           |
     --------------  Fc |  ---------+--+      +-----+------+  |
    | CE Manager |--------+-|    CE 1   |------|    CE 2    | |
     --------------       | |           |  Fr  |            | |
         |                | +-+---------+-+    +-----------+  |
         |                |   |          | Fp      /         |
       | Fl               |   |          +--------+ /         |
         |                |   |  Fp               |/          |
         |                |   |               Fp  /|----+      |
         |                |   |            /-------/   |      |
     --------------  Ff  |  ---+---------     -------------  |
    | FE Manager |--------+-|   FE 1    | Fi |    FE 2    | |
     --------------       | |           |------|            | |
         |                |  -------------     -------------  |
         |                |   | | | | |         | | | | |    |
                        ----+--+--+--+---------+--+--+--+-----
                            | | | | |         | | | | |
                            | | | | |         | | | | |
                             Fi/f                 Fi/f
```
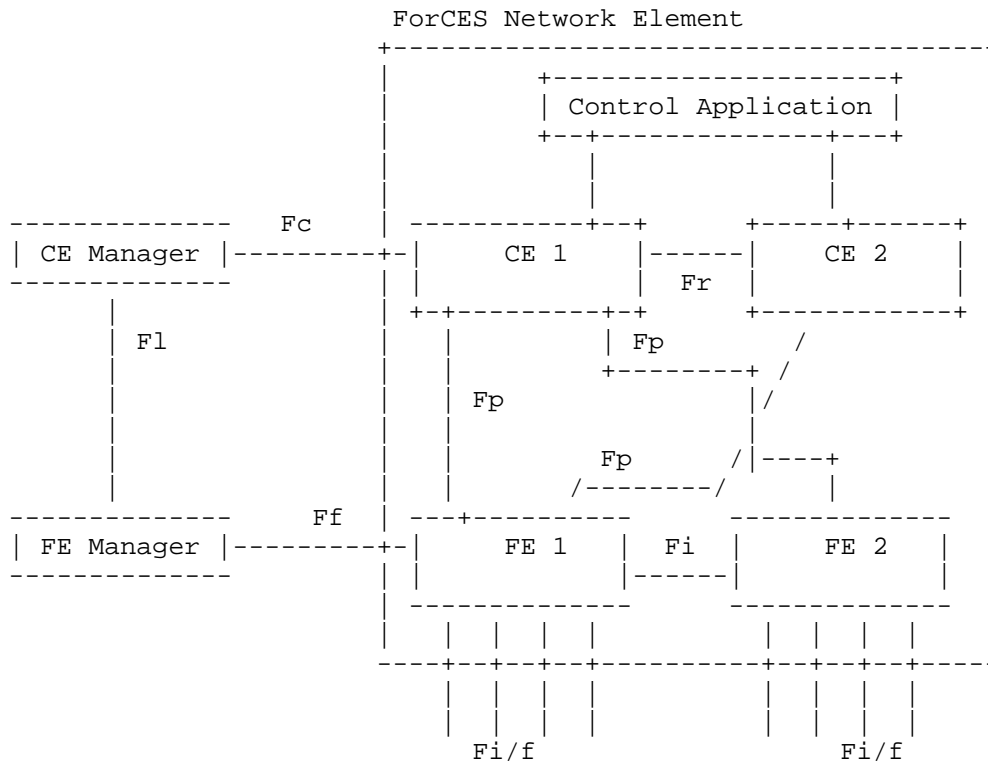
           Fp: CE-FE interface
           Fr: CE-CE interface
           Fc: Interface between the CE Manager and a CE
           Ff: Interface between the FE Manager and an FE
           Fl: Interface between the CE Manager and the FE Manager
           Fi/f: FE external interface

                  Figure 1: ForCES Architectural Diagram

   Figure 1 shows a control application manipulating, at runtime, FE
   configuration via the SM LFB control. It would appear that this
   control application is playing the part of the FE Manager and 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 class 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 configuration information, as well as the
   ability to manipulate the 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 configuration 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 FEMs 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:

   o Logical Functional Block (LFB)

   o Forwarding Element (FE)

   o Control Element (CE)

   o ForCES Network Element (NE)

   o FE Manager (FEM)

   o CE Manager

   o ForCES Protocol

   o ForCES Protocol Layer (ForCES PL)

   o 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 be
   associated to another CE as a backup. To achieve this goal, the CE
   management application specifies the Control Element ID (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 to multiple CEs, possibly
   in an active backup setup. Assume that system analytics discover
   that the CE is becoming a bottleneck. A new CE could 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 DynamicLFBLoading capability of the 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:

o LFB class - The LFB class ID

o LFB version - The version of the LFB class

o LFB class name - Optional, the LFB name

o 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 parameters are fields that need to be described in documentation,
depending on the implementation; one example is 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. These parameters
are placeholders for future use, in order not to redefine the LFB
class versions each time. They are simple strings that define the
parameters supported by the LFB. The CE is expected to read this
capability in order to understand the parameters it can use.

2.5. Logging Mechanism

The SM LFB class also provides a useful log-level manipulation.
Experience has proven that the CE may be required to increase or
decrease the debug levels of parts of the FE, whether that be LFBs,
portions of LFBs, or generic processing code (all called "modules").
The module granularity is implementation specific and is not
discussed in this document. The debug levels are derived from the
"syslog Message Severities" registry
<http://www.iana.org/assignments/syslog-parameters> defined in
[RFC3164].

2.6. General-Purpose Attribute Definition

Experience has shown that a generic attribute name-value pair is
useful for describing configuration information. This LFB class
defines such a generic attribute name-value pair defined as a table
of attribute name-value pair values. The attribute name-value pair
is implementation specific and at the moment there is nothing to
standardize. As an example, consider switches that have exactly the

same LFB classes and capabilities but need to be used in different
roles. A good example would be a switch that could be used either as
Spine or Top-of-Rack (ToR) in data-center setups. An attribute that
defines the role could be retrieved from the FE, which will then
dictate how it is controlled and 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. The CE is expected to read the SupportedArguments
capability in order to know what attributes it can use.

3. Applicability Statement

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

3.1. FE Integrated

   Only one instance of the SM LFB 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 LFB class can exist, one per each
   virtual FE.

4. SM Library

4.1. Frame Definitions

   This LFB class does not define any frames.

4.2. Data Type Definitions

   This library defines the following data types.

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

                           FEM Data Types

4.3. Metadata Definitions

   This LFB does not define any metadata definitions.

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. Its function is to provide
   the configuration parameters to the CE to be updated at runtime.

4.4.2. Components

   This LFB class 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 LFB 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, and 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 Configuration 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 manage the SM for loading
   LFBs and create new connections between FEs and CEs.

   On the issue of trust, a designer should take into account that the
   CE that creates new connections to CEs is either:

   o The FE manager that is responsible for managing the FEs, or

   o An already associated CE

   In both of these cases, the entity making the connections should
   already be trusted to perform such activities. If the entity making
   the connections is faulty, rogue, or hacked, there is no way for the
   FE to know this, and it will perform any action that the CE requests.
   Therefore, 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.

   The reader is also referred to the ForCES framework [RFC3746]
   document, particularly Section 8, for an analysis of potential
   threats introduced by ForCES and how the ForCES architecture
   addresses them.

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. The registration procedure is Standards Action
   for the range 0 to 65535 and First Come First Served with any
   publicly available specification for identifiers over 65535
   [RFC5226]. This specification registers the following LFB class name
   and LFB class identifier in the "Logical Functional Block (LFB) Class
   Names and Class Identifiers" registry:

   +------------+--------+---------+----------------------+-----------+
LFB Class	LFB	LFB	Description	Reference
Identifier	Class	Version		
	Name			
   +------------+--------+---------+----------------------+-----------+
19	SM	1.0	An SM LFB to	RFC 7729
			standardize	(this
			subsidiary management	document)
			for ForCES Network	
			Elements	
   +------------+--------+---------+----------------------+-----------+

        Logical Functional Block (LFB) Class Name and Class Identifier

8. References

8.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <http://www.rfc-editor.org/info/rfc2119>.

   [RFC5810] Doria, A., Ed., Hadi Salim, J., Ed., Haas, R., Ed.,
              Khosravi, H., Ed., Wang, W., Ed., Dong, L., Gopal, R., and
              J. Halpern, "Forwarding and Control Element Separation
              (ForCES) Protocol Specification", RFC 5810,
              DOI 10.17487/RFC5810, March 2010,
              <http://www.rfc-editor.org/info/rfc5810>.

   [RFC5812] Halpern, J. and J. Hadi Salim, "Forwarding and Control
              Element Separation (ForCES) Forwarding Element Model",
              RFC 5812, DOI 10.17487/RFC5812, March 2010,
              <http://www.rfc-editor.org/info/rfc5812>.

   [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,
              DOI 10.17487/RFC7121, February 2014,
              <http://www.rfc-editor.org/info/rfc7121>.

8.2. Informative References

   [RFC3164] Lonvick, C., "The BSD Syslog Protocol", RFC 3164,
              DOI 10.17487/RFC3164, August 2001,
              <http://www.rfc-editor.org/info/rfc3164>.

   [RFC3654] Khosravi, H., Ed. and T. Anderson, Ed., "Requirements for
              Separation of IP Control and Forwarding", RFC 3654,
              DOI 10.17487/RFC3654, November 2003,
              <http://www.rfc-editor.org/info/rfc3654>.

   [RFC3746] Yang, L., Dantu, R., Anderson, T., and R. Gopal,
              "Forwarding and Control Element Separation (ForCES)
              Framework", RFC 3746, DOI 10.17487/RFC3746, April 2004,
              <http://www.rfc-editor.org/info/rfc3746>.

   [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an
              IANA Considerations Section in RFCs", BCP 26, RFC 5226,
              DOI 10.17487/RFC5226, May 2008,
              <http://www.rfc-editor.org/info/rfc5226>.

Authors' Addresses

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

   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 Road
   Ottawa, Ontario K2H 9G1
   Canada

   Email: hadi@mojatatu.com