Internet Engineering Task Force
Internet-Draft
Updates: 7121 (if approved)
Intended status: Standards Track
Expires: December 5, 2014

J. Hadi Salim
Mojatatu Networks

June 3, 2014

ForCES Protocol Extensions
draft-ietf-forces-protoextension-02

Abstract

Experience in implementing and deploying ForCES architecture has demonstrated need for a few small extensions both to ease programmability and to improve wire efficiency of some transactions. This document describes extensions to the ForCES Protocol Specification[RFC 5810] semantics to achieve that end goal.

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 December 5, 2014.

Table of Contents

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 reiterates the terminology defined by the ForCES architecture in various documents for the sake of 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.

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.

ForCES Protocol - Protocol that runs in the Fp reference points in the ForCES Framework [RFC3746].

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 as defined in the ForCES Protocol Specification [RFC5810].

ForCES Protocol Transport Mapping Layer (ForCES TML) - A layer in ForCES protocol architecture that uses the capabilities of existing transport protocols to specifically address protocol message transportation issues, such as how the protocol messages are mapped to different transport media (like TCP, IP, ATM, Ethernet, etc.), and how to achieve and implement reliability, ordering, etc. the ForCES SCTP TML [RFC5811] describes a TML that is mandated for ForCES.

2. Introduction

Experience in implementing and deploying ForCES architecture has demonstrated need for a few small extensions both to ease programmability and to improve wire efficiency of some transactions. This document describes a few extensions to the ForCES Protocol Specification [RFC5810] semantics to achieve that end goal.

This document describes and justifies the need for 2 small extensions which are backward compatible. The document also clarifies details of how dumping of a large table residing on an FE is achieved. To summarize:

1. A table range operation to allow a controller or control application to request an arbitrary range of table rows is introduced.

2. Additional error codes returned to the controller (or control application) by an FE are introduced. Additionally a new extension to carry details on error codes is introduced.

3. While already supported FE response to a GET request of a large table which does not fit in a single PL message is not described in [RFC5810]. This document clarifies the details.

3. Problem Overview

In this section we present sample use cases to illustrate each challenge being addressed.

3.1. Table Ranges

Consider, for the sake of illustration, an FE table with 1 million reasonably sized table rows which are sparsely populated. Assume, again for the sake of illustration, that there are 2000 table rows sparsely populated between the row indices 23-10023.

Implementation experience has shown that existing approaches for retrieving or deleting a sizeable number of table rows is at the programmatically level (from an application point of view) unfriendly, tedious, and abusive of both compute and bandwidth resources.

By Definition, ForCES GET and DEL requests sent from a controller (or control app) are prepended with a path to a component and sent to the FE. In the case of indexed tables, the component path can either point to a table or a table row index.

As an example, a control application attempting to retrieve the first 2000 table rows appearing between row indices 23 and 10023 can achieve its goal in one of:

- Dump the whole table and filter for the needed 2000 table rows.
- Send upto 10000 ForCES PL requests with monotonically incrementing indices and stop when the needed 2000 entries are retrieved.
- If the application had knowledge of which table rows existed (not unreasonable given the controller is supposed to be aware of state within an NE), then the application could take advantage of ForCES batching to send fewer large messages (each with different path entries for a total of two thousand).

As argued, while the above options exist - all are tedious.

3.2. Error codes

[RFC5810] has defined a generic set of error codes that are to be returned to the CE from an FE. Deployment experience has shown that it would be useful to have more fine grained error codes. As an example, the error code E_NOT_SUPPORTED could be mapped to many FE error source possibilities that need to be then interpreted by the caller based on some understanding of the nature of the sent request. This makes debugging more time consuming.

4. Protocol Update Proposal

This section describes proposals to update the protocol for issues discussed in Section 3

4.1. Table Ranges

We propose to add a Table-range TLV (type ID 0x117) that will be associated with the PATH-DATA TLV in the same manner the KEYINFO-TLV is.

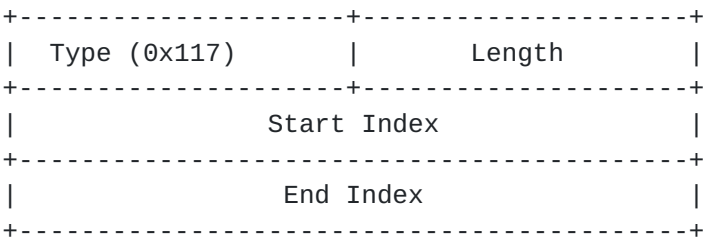

Figure 1: ForCES table range request Layout

Figure 1 shows how this new TLV is constructed.

```
OPER = GET
       PATH-DATA:
         flags = F_SELTABRANGE,  IDCount = 2, IDs = {1,6}
         TABLERANGE-TLV content = {11,23}
```

Figure 2: ForCES table range request

Figure 2 illustrates a GET request for a range of rows 11 to 23 of a table with component path of "1/6".

Path flag of F_SELTABRANGE (0x2 i.e bit 1, where bit 0 is F_SELKEY as defined in RFC 5810) MUST be set to indicate the presence of the

TABLERANGE-TLV. The pathflag bit F_SELTABRANGE can only be used in a GET or DEL and is mutually exclusive with F_SELKEY. The FE MUST enforce the path flag constraints and ensure that the selected path belongs to a defined indexed table component. Any violation of these constraints MUST be rejected with an error code of E_INVALID_TFLAGS with a description of what the problem is when using extended error reporting (refer to Section 4.2).

The TABLERANGE-TLV contents constitute:

- A 32 bit start index. An index of 0 implies the beggining of the table row.

- A 32 bit end index. A value of 0xFFFFFFFFFFFFFFFF implies the last entry.

The response for a table range query will either be:

- The requested table data returned (when at least one referenced row is available); in such a case, a response with a path pointing to the table and whose data content contain the row(s) will be sent to the CE. The data content MUST be encapsulated in sparsedata TLV. The sparse data TLV content will have the "I" (in ILV) for each table row indicating the table indices.

- An EXTENDEDRESULT-TLV (refer to Section 4.2.3) when:

  * Response is to a range delete request. The Result will either be:

    + A success if any of the requested-for rows is deleted

    + A proper error code if none of the requested for rows can be deleted

  * data is absent where the result code of E_EMPTY with an optional content string describing the nature of the error (refer to Section 4.2).

  * When both a path key and path table range are reflected on the the pathflags, an error code of E_INVALID_TFLAGS with an optional content string describing the nature of the error (refer to Section 4.2).

  * other standard ForCES errors (such as ACL constraints trying to retrieve contents of an unreadable table), accessing unknown components etc.

4.2. Error Codes

We propose several things:

1. A new set of error codes.

2. Allocating some reserved codes for vendor use.

3. A new TLV, EXTENDEDRESULT-TLV (0x118) that will carry a code (which will be a superset of what is currently specified in [RFC5810]) but also an optional cause content. This is illustrated in Figure 3.

4.2.1. New Codes

EXTENDEDRESULT-TLV Result Value is 32 bits and is a superset of RFC 5810 Result TLV Result Value. The new version code space is 32 bits as opposed to the RFC 5810 code size of 8 bits. The first 8 bit values are common to both old

Code	Mnemonic	Details
0x18	E_TIMED_OUT	A time out occured while processing the message
0x19	E_INVALID_TFLAGS	Invalid table flags
0x1A	E_INVALID_OP	Requested operation is invalid
0x1B	E_CONGEST_NT	Node Congestion notification
0x1C	E_COMPONENT_NOT_A_TABLE	Component not a table
0x1D	E_PERM	Operation not permitted
0x1E	E_BUSY	System is Busy
0x1F	E_EMPTY	Table is empty
0x20	E_UNKNOWN	A generic catch all error code. Carries a string to further extrapolate what the error implies.

Table 1: New codes

4.2.2. Vendor Codes

Codes 0x100-0x200 are reserved for use as vendor codes. Since these are freely available it is expected that the FE and CE side will both understand/interpret the semantics of any used codes.

4.2.3. Extended Result TLV

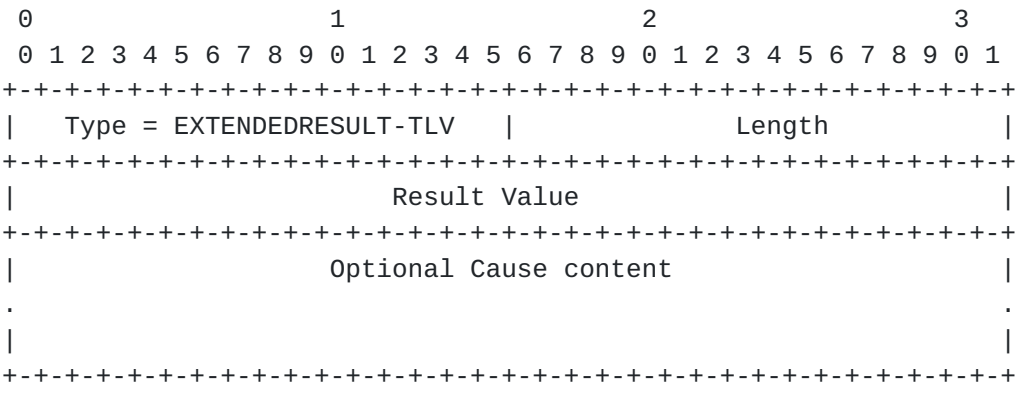

Figure 3: EXTENDEDRESULT-TLV

- Like all other ForCES TLVs, the EXTENDEDRESULT-TLV is expected to be 32 bit aligned.

- The EXTENDEDRESULT-TLV Result Value derives and extends from the same current namespace that is used by RESULT-TLV Result Value as specified in RFC 5810, section 7.1.7. The main difference is that we now have 32 bit result value (as opposed to the old 8 bit).

- The optional result content is defined to further disambiguate the result value. It is expected Utf-8 string values to be used. However, vendor specific error codes may choose to specify different contents. Additionally, future codes may specify cause contents to be of types other than string..

- It is recommended that the maximum size of the cause string should not exceed 32 bytes. We do not propose the cause string be standardized.

4.2.3.1. Extended Result Backward compatibility

To support backward compatibility, we add a new component ID 16 (named EResultAdmin), a capability Component ID 32 (named EResultCapab), and updated the version to 1.2 for the FEPO LFB Appendix A.

An FE will advertise its capability to support extended TLVs via the EResultCapab table. When an FE is capable of responding with both extended results and older result TLVs, it will have two table rows one for each supported value. By default an FE capable of supporting both modes will assume the lowest common denominator i.e EResultAdmin will be EResultNotSupported; and will issue responses using RESULT-

TLVs.

A master CE can turn on support for extended results by setting the value to 2 in which case the FE MUST switch over to sending only EXTENDEDRESULT-TLVs. Likewise a master CE can turn off extended result responses by writting a 1 to the EResultAdmin. An FE that does not support one mode or other MUST reject setting of EResultAdmin to a value it does not support by responding with an error code of E_NOT_SUPPORTED.

4.3. Large Table Dumping

Imagine a GET request to a path that is a table i.e a table dump. Such a request is sent to the FE with a specific correlator, say X. Imagine this table to have a large number of entries at the FE. For the sake of illustration, lets say millions of rows. This requires that the FE delivers the response over multiple messages, all using the same correlator X.

The protocol document [RFC5810] does not adequately describe how a GET response to such a large message is delivered. The text in this section clarifies. We limit the discussion to a table object only.

Implementation experience of dumping large tables indicates we can use the transaction flags to indicate that a GET response is the beginning, middle or end of a multi-part message. In other words we mirror the effect of an atomic transaction sent by a CE to an FE.

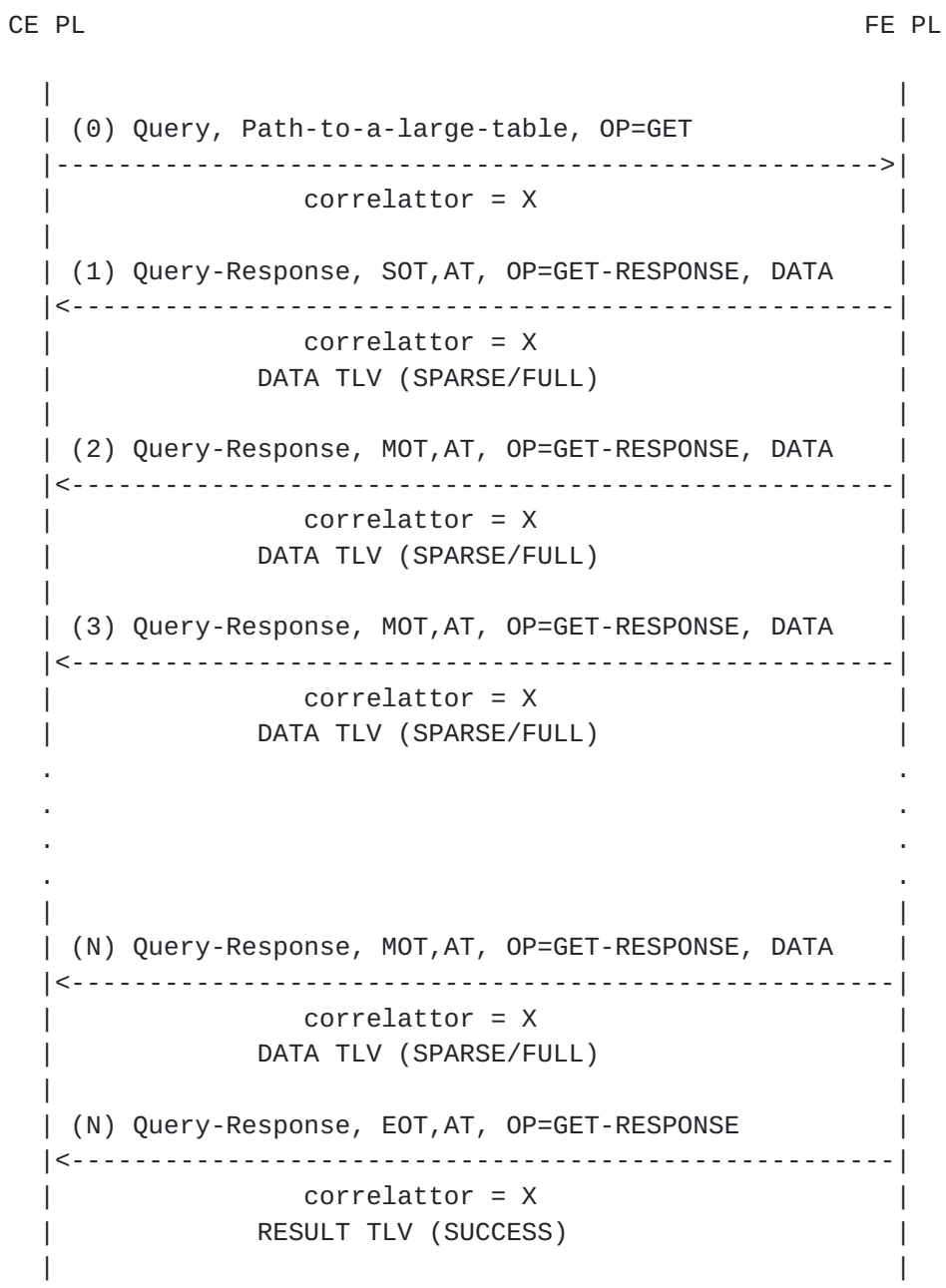

Figure 4: EXTENDEDRESULT-TLV

The last message which carries the EOT flag to go the CE MUST not carry any data. This allows us to mirror ForCES 2PC messaging [RFC5810] where the last message is an empty commit message. GET response will carry a result code TLV in such a case.

5. Acknowledgements

The author would like to thank Evangelos Haleplidis and Joel Halpern

for discussions that made this document better.

6. IANA Considerations

This document registers two new top Level TLVs and two new path flags and updates an IANA registered FE Protocol object Logical Functional Block (LFB).

XXX: when this document is undergoing IANA review we should update https://www.iana.org/assignments/forces/forces.xml section on FEPO to have the new version reflected.

The following new TLVs are defined:

- TABLERANGE-TLV (type ID 0x117)
- EXTENDEDRESULT-TLV (type ID 0x118)

The following new path flags are defined:

- F_SELTABRANGE (value 0x2 i.e bit 1)

The Defined Result Values are changed:

- codes 0x21-0xFE are reserved.
- codes 0x18-0x20 are defined by this document.
- codes 0x100-0x200 are reserved for vendor use.

7. Security Considerations

The security considerations that have been described in the ForCES protocol [RFC5810] apply to this document as well.

8. References

8.1. Normative References

[RFC3746] Yang, L., Dantu, R., Anderson, T., and R. Gopal, "Forwarding and Control Element Separation (ForCES) Framework", RFC 3746, April 2004.

[RFC5810] Doria, A., Hadi Salim, J., Haas, R., Khosravi, H., Wang, W., Dong, L., Gopal, R., and J. Halpern, "Forwarding and

Control Element Separation (ForCES) Protocol Specification", RFC 5810, March 2010.

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

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

[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.

8.2. Informative References

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

Appendix A. Appendix A - New FEPO version

```
<LFBLibrary xmlns="urn:ietf:params:xml:ns:forces:lfbmodel:1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="lfb-schema.xsd" provides="FEPO">
   <!-- XXX -->
   <dataTypeDefs>
      <dataTypeDef>
         <name>CEHBPolicyValues</name>
         <synopsis>
            The possible values of CE heartbeat policy
         </synopsis>
         <atomic>
            <baseType>uchar</baseType>
            <specialValues>
               <specialValue value="0">
                  <name>CEHBPolicy0</name>
                  <synopsis>
                      The CE will send heartbeats to the FE
                      every CEHDI timeout if no other messages
                      have been sent since.
                  </synopsis>
               </specialValue>
               <specialValue value="1">
                  <name>CEHBPolicy1</name>
```

```
                  <synopsis>
                      The CE will not send heartbeats to the FE
                  </synopsis>
               </specialValue>
            </specialValues>
         </atomic>
      </dataTypeDef>
      <dataTypeDef>
         <name>FEHBPolicyValues</name>
         <synopsis>
             The possible values of FE heartbeat policy
         </synopsis>
         <atomic>
            <baseType>uchar</baseType>
            <specialValues>
               <specialValue value="0">
                  <name>FEHBPolicy0</name>
                  <synopsis>
                      The FE will not generate any heartbeats
                      to the CE
                  </synopsis>
               </specialValue>
               <specialValue value="1">
                  <name>FEHBPolicy1</name>
                  <synopsis>
                     The FE generates heartbeats to the CE every FEHI
                     if no other messages have been sent to the CE.
                  </synopsis>
               </specialValue>
            </specialValues>
         </atomic>
      </dataTypeDef>
      <dataTypeDef>
         <name>FERestartPolicyValues</name>
         <synopsis>
            The possible values of FE restart policy
         </synopsis>
         <atomic>
            <baseType>uchar</baseType>
            <specialValues>
               <specialValue value="0">
                  <name>FERestartPolicy0</name>
                  <synopsis>
                     The FE restart restats its state from scratch
                  </synopsis>
               </specialValue>
            </specialValues>
         </atomic>
```

```
        </dataTypeDef>
        <dataTypeDef>
           <name>HAModeValues</name>
           <synopsis>
              The possible values of HA modes
           </synopsis>
           <atomic>
              <baseType>uchar</baseType>
              <specialValues>
                 <specialValue value="0">
                    <name>NoHA</name>
                    <synopsis>
                       The FE is not running in HA mode
                    </synopsis>
                 </specialValue>
                 <specialValue value="1">
                    <name>ColdStandby</name>
                    <synopsis>
                       The FE is running in HA mode cold Standby
                    </synopsis>
                 </specialValue>
                 <specialValue value="2">
                    <name>HotStandby</name>
                    <synopsis>
                       The FE is running in HA mode hot Standby
                    </synopsis>
                 </specialValue>
              </specialValues>
           </atomic>
        </dataTypeDef>
        <dataTypeDef>
           <name>CEFailoverPolicyValues</name>
           <synopsis>
              The possible values of CE failover policy
           </synopsis>
           <atomic>
              <baseType>uchar</baseType>
              <specialValues>
                 <specialValue value="0">
                    <name>CEFailoverPolicy0</name>
                    <synopsis>
                        The FE should stop functioning immediate and
                        transition to the FE OperDisable state
                    </synopsis>
                 </specialValue>
                 <specialValue value="1">
                    <name>CEFailoverPolicy1</name>
                    <synopsis>
```

```
                    The FE should continue forwarding even
                    without an associated CE for CEFTI. The
                    FE goes to FE OperDisable when the CEFTI
                    expires and no association. Requires
                    graceful restart support.
                 </synopsis>
              </specialValue>
           </specialValues>
        </atomic>
     </dataTypeDef>
     <dataTypeDef>
        <name>FEHACapab</name>
        <synopsis>
           The supported HA features
        </synopsis>
        <atomic>
           <baseType>uchar</baseType>
           <specialValues>
              <specialValue value="0">
                 <name>GracefullRestart</name>
                 <synopsis>
                    The FE supports Graceful Restart
                 </synopsis>
              </specialValue>
              <specialValue value="1">
                 <name>HA</name>
                 <synopsis>
                    The FE supports HA
                 </synopsis>
              </specialValue>
           </specialValues>
        </atomic>
     </dataTypeDef>
     <dataTypeDef>
        <name>CEStatusType</name>
        <synopsis>Status values. Status for each CE</synopsis>
        <atomic>
           <baseType>uchar</baseType>
           <specialValues>
              <specialValue value="0">
                 <name>Disconnected</name>
                 <synopsis>No connection attempt with the CE yet
                 </synopsis>
              </specialValue>
              <specialValue value="1">
                 <name>Connected</name>
                 <synopsis>The FE connection with the CE at the TML
                    has been completed
```

```
                  </synopsis>
               </specialValue>
               <specialValue value="2">
                  <name>Associated</name>
                  <synopsis>The FE has associated with the CE
                  </synopsis>
               </specialValue>
               <specialValue value="3">
                  <name>IsMaster</name>
                  <synopsis>The CE is the master (and associated)
                  </synopsis>
               </specialValue>
               <specialValue value="4">
                  <name>LostConnection</name>
                  <synopsis>The FE was associated with the CE but
                     lost the connection
                  </synopsis>
               </specialValue>
               <specialValue value="5">
                  <name>Unreachable</name>
                  <synopsis>The CE is deemed as unreachable by the FE
                  </synopsis>
               </specialValue>
            </specialValues>
         </atomic>
      </dataTypeDef>
      <dataTypeDef>
         <name>StatisticsType</name>
         <synopsis>Statistics Definition</synopsis>
         <struct>
            <component componentID="1">
               <name>RecvPackets</name>
               <synopsis>Packets Received</synopsis>
               <typeRef>uint64</typeRef>
            </component>
            <component componentID="2">
               <name>RecvErrPackets</name>
               <synopsis>Packets Received from CE with errors
               </synopsis>
               <typeRef>uint64</typeRef>
            </component>
            <component componentID="3">
               <name>RecvBytes</name>
               <synopsis>Bytes Received from CE</synopsis>
               <typeRef>uint64</typeRef>
            </component>
            <component componentID="4">
               <name>RecvErrBytes</name>
```

```
                <synopsis>Bytes Received from CE in Error</synopsis>
                <typeRef>uint64</typeRef>
             </component>
             <component componentID="5">
                <name>TxmitPackets</name>
                <synopsis>Packets Transmitted to CE</synopsis>
                <typeRef>uint64</typeRef>
             </component>
             <component componentID="6">
                <name>TxmitErrPackets</name>
                <synopsis>
                   Packets Transmitted to CE that incurred
                   errors
                </synopsis>
                <typeRef>uint64</typeRef>
             </component>
             <component componentID="7">
                <name>TxmitBytes</name>
                <synopsis>Bytes Transmitted to CE</synopsis>
                <typeRef>uint64</typeRef>
             </component>
             <component componentID="8">
                <name>TxmitErrBytes</name>
                <synopsis>Bytes Transmitted to CE incurring errors
                </synopsis>
                <typeRef>uint64</typeRef>
             </component>
          </struct>
       </dataTypeDef>
       <dataTypeDef>
          <name>AllCEType</name>
          <synopsis>Table Type for AllCE component</synopsis>
          <struct>
             <component componentID="1">
                <name>CEID</name>
                <synopsis>ID of the CE</synopsis>
                <typeRef>uint32</typeRef>
             </component>
             <component componentID="2">
                <name>Statistics</name>
                <synopsis>Statistics per CE</synopsis>
                <typeRef>StatisticsType</typeRef>
             </component>
             <component componentID="3">
                <name>CEStatus</name>
                <synopsis>Status of the CE</synopsis>
                <typeRef>CEStatusType</typeRef>
             </component>
```

```
            </struct>
         </dataTypeDef>
         <dataTypeDef>
            <name>ExtendedResultType</name>
            <synopsis>
                Possible extended result support
            </synopsis>
            <atomic>
               <baseType>uchar</baseType>
               <rangeRestriction>
                 <allowedRange min="1" max="2"/>
               </rangeRestriction>
               <specialValues>
                  <specialValue value="1">
                     <name>EResultNotSupported</name>
                     <synopsis>
                         Extended Results are not supported
                     </synopsis>
                  </specialValue>
                  <specialValue value="2">
                     <name>EResultSupported</name>
                     <synopsis>
                         Extended Results are supported
                     </synopsis>
                  </specialValue>
               </specialValues>
            </atomic>
         </dataTypeDef>
      </dataTypeDefs>
      <LFBClassDefs>
         <LFBClassDef LFBClassID="2">
            <name>FEPO</name>
            <synopsis>
               The FE Protocol Object, with EXtended Result control
            </synopsis>
            <version>1.2</version>
            <components>
               <component componentID="1" access="read-only">
                  <name>CurrentRunningVersion</name>
                  <synopsis>Currently running ForCES version</synopsis>
                  <typeRef>uchar</typeRef>
               </component>
               <component componentID="2" access="read-only">
                  <name>FEID</name>
                  <synopsis>Unicast FEID</synopsis>
                  <typeRef>uint32</typeRef>
               </component>
               <component componentID="3" access="read-write">
```

```
      <name>MulticastFEIDs</name>
      <synopsis>
         the table of all multicast IDs
      </synopsis>
      <array type="variable-size">
         <typeRef>uint32</typeRef>
      </array>
   </component>
   <component componentID="4" access="read-write">
      <name>CEHBPolicy</name>
      <synopsis>
         The CE Heartbeat Policy
      </synopsis>
      <typeRef>CEHBPolicyValues</typeRef>
   </component>
   <component componentID="5" access="read-write">
      <name>CEHDI</name>
      <synopsis>
         The CE Heartbeat Dead Interval in millisecs
      </synopsis>
      <typeRef>uint32</typeRef>
   </component>
   <component componentID="6" access="read-write">
      <name>FEHBPolicy</name>
      <synopsis>
         The FE Heartbeat Policy
      </synopsis>
      <typeRef>FEHBPolicyValues</typeRef>
   </component>
   <component componentID="7" access="read-write">
      <name>FEHI</name>
      <synopsis>
         The FE Heartbeat Interval in millisecs
      </synopsis>
      <typeRef>uint32</typeRef>
   </component>
   <component componentID="8" access="read-write">
      <name>CEID</name>
      <synopsis>
         The Primary CE this FE is associated with
      </synopsis>
      <typeRef>uint32</typeRef>
   </component>
   <component componentID="9" access="read-write">
      <name>BackupCEs</name>
      <synopsis>
         The table of all backup CEs other than the
         primary
```

```
      </synopsis>
      <array type="variable-size">
         <typeRef>uint32</typeRef>
      </array>
   </component>
   <component componentID="10" access="read-write">
      <name>CEFailoverPolicy</name>
      <synopsis>
         The CE Failover Policy
      </synopsis>
      <typeRef>CEFailoverPolicyValues</typeRef>
   </component>
   <component componentID="11" access="read-write">
      <name>CEFTI</name>
      <synopsis>
         The CE Failover Timeout Interval in millisecs
      </synopsis>
      <typeRef>uint32</typeRef>
   </component>
   <component componentID="12" access="read-write">
      <name>FERestartPolicy</name>
      <synopsis>
         The FE Restart Policy
      </synopsis>
      <typeRef>FERestartPolicyValues</typeRef>
   </component>
   <component componentID="13" access="read-write">
      <name>LastCEID</name>
      <synopsis>
         The Primary CE this FE was last associated
         with
      </synopsis>
      <typeRef>uint32</typeRef>
   </component>
   <component componentID="14" access="read-write">
      <name>HAMode</name>
      <synopsis>
         The HA mode used
      </synopsis>
      <typeRef>HAModeValues</typeRef>
   </component>
   <component componentID="15" access="read-only">
      <name>AllCEs</name>
      <synopsis>The table of all CEs</synopsis>
      <array type="variable-size">
         <typeRef>AllCEType</typeRef>
      </array>
    </component>
```

```
        <component componentID="16" access="read-write">
          <name>EResultAdmin</name>
          <synopsis>
              Turn Extended results off or on.
              default to off
          </synopsis>
          <typeRef>ExtendedResultType</typeRef>
          <defaultValue>1</defaultValue>
       </component>
    </components>
    <capabilities>
       <capability componentID="30">
          <name>SupportableVersions</name>
          <synopsis>
             the table of ForCES versions that FE supports
          </synopsis>
          <array type="variable-size">
             <typeRef>uchar</typeRef>
          </array>
       </capability>
       <capability componentID="31">
          <name>HACapabilities</name>
          <synopsis>
             the table of HA capabilities the FE supports
          </synopsis>
          <array type="variable-size">
             <typeRef>FEHACapab</typeRef>
          </array>
       </capability>
        <capability componentID="32">
          <name>EResultCapab</name>
          <synopsis>
             the table of supported result capabilities
          </synopsis>
          <array type="variable-size">
             <typeRef>ExtendedResultType</typeRef>
          </array>
      </capability>
    </capabilities>
    <events baseID="61">
       <event eventID="1">
          <name>PrimaryCEDown</name>
          <synopsis>
             The primary CE has changed
          </synopsis>
          <eventTarget>
             <eventField>LastCEID</eventField>
          </eventTarget>
```

```
                  <eventChanged/>
                  <eventReports>
                     <eventReport>
                        <eventField>LastCEID</eventField>
                     </eventReport>
                  </eventReports>
               </event>
               <event eventID="2">
                  <name>PrimaryCEChanged</name>
                  <synopsis>A New primary CE has been selected
                  </synopsis>
                  <eventTarget>
                     <eventField>CEID</eventField>
                  </eventTarget>
                  <eventChanged/>
                  <eventReports>
                     <eventReport>
                        <eventField>CEID</eventField>
                     </eventReport>
                  </eventReports>
               </event>
            </events>
         </LFBClassDef>
      </LFBClassDefs>
   </LFBLibrary>
```

Author's Address

Jamal Hadi Salim
Mojatatu Networks
Suite 400, 303 Moodie Dr.
Ottawa, Ontario K2H 9R4
Canada

Email: hadi@mojatatu.com