Internet Engineering Task Force D. Joachimpillai
Internet-Draft Verizon
Intended status: Standards Track J. Hadi Salim
Expires: December 24, 2016 Mojatatu Networks
June 22, 2016

ForCES Inter-FE LFB
draft-ietf-forces-interfelfb-05

Abstract

This document describes how to extend the ForCES LFB topology across FEs by defining the Inter-FE LFB Class. The Inter-FE LFB Class provides the ability to pass data and metadata across FEs without needing any changes to the ForCES specification. The document focuses on Ethernet transport.

Status of This Memo

Copyright Notice

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 depends on the terminology defined in several ForCES documents [RFC3746], [RFC5810], [RFC5811], and [RFC5812] [RFC7391] [RFC7408] for the sake of contextual clarity.

Control Engine (CE)

Forwarding Engine (FE)

FE Model

LFB (Logical Functional Block) Class (or type)

LFB Instance

LFB Model

LFB Metadata

ForCES Component

LFB Component

ForCES Protocol Layer (ForCES PL)

ForCES Protocol Transport Mapping Layer (ForCES TML)

2. Introduction

In the ForCES architecture, a packet service can be modelled by composing a graph of one or more LFB instances. The reader is referred to the details in the ForCES Model [RFC5812].

The ForCES model describes the processing within a single Forwarding Element (FE) in terms of logical forwarding blocks (LFB), including provision for the Control Element (CE) to establish and modify that processing sequence, and the parameters of the individual LFBs.

Under some circumstance, it would be beneficial to be able to extend this view, and the resulting processing across more than one FE. This may be in order to achieve scale by splitting the processing

across elements, or to utilize specialized hardware available on specific FEs.

Given that the ForCES inter-LFB architecture calls for the ability to pass metadata between LFBs, it is imperative therefore to define mechanisms to extend that existing feature and allow passing the metadata between LFBs across FEs.

This document describes how to extend the LFB topology across FEs i.e inter-FE connectivity without needing any changes to the ForCES definitions. It focuses on using Ethernet as the interconnection between FEs.

3. Problem Scope And Use Cases

The scope of this document is to solve the challenge of passing ForCES defined metadata alongside packet data across FEs (be they physical or virtual) for the purpose of distributing the LFB processing.

3.1. Assumptions

- The FEs involved in the Inter-FE LFB belong to the same Network Element(NE) and are within a single administrative private network which is in close proximity.

- The FEs are already interconnected using Ethernet. We focus on Ethernet because it is a very common setup as an FE interconnect. Other higher transports (such as UDP over IP) or lower transports could be defined to carry the data and metadata, but these cases are not addressed in this document.

3.2. Sample Use Cases

To illustrate the problem scope we present two use cases where we start with a single FE running all the LFBs functionality then split it into multiple FEs achieving the same end goals.

3.2.1. Basic IPv4 Router

A sample LFB topology depicted in Figure 1 demonstrates a service graph for delivering basic IPv4 forwarding service within one FE. For the purpose of illustration, the diagram shows LFB classes as graph nodes instead of multiple LFB class instances.

Since the illustration on Figure 1 is meant only as an exercise to showcase how data and metadata are sent down or upstream on a graph of LFB instances, it abstracts out any ports in both directions and

talks about a generic ingress and egress LFB. Again, for illustration purposes, the diagram does not show exception or error paths. Also left out are details on Reverse Path Filtering, ECMP, multicast handling etc. In other words, this is not meant to be a complete description of an IPv4 forwarding application; for a more complete example, please refer the LFBlib document [RFC6956].

The output of the ingress LFB(s) coming into the IPv4 Validator LFB will have both the IPv4 packets and, depending on the implementation, a variety of ingress metadata such as offsets into the different headers, any classification metadata, physical and virtual ports encountered, tunnelling information etc. These metadata are lumped together as "ingress metadata".

Once the IPv4 validator vets the packet (example ensures that no expired TTL etc), it feeds the packet and inherited metadata into the IPv4 unicast LPM LFB.

```
                      +----+
                      |    |
           IPv4 pkt   |    | IPv4 pkt     +-----+             +---+
       +------------->|    +------------->|     |             |   |
       |  + ingress   |    | + ingress    |IPv4 |   IPv4 pkt  |   |
       |   metadata   |    | metadata     |Ucast+------------>|   +--+
       |              +----+              |LPM  |  + ingress  |   |  |
     +-+-+             IPv4               +-----+  + NHinfo   +---+  |
     |   |             Validator                   metadata   IPv4   |
     |   |             LFB                                    NextHop|
     |   |                                                     LFB   |
     |   |                                                           |
     |   |                                                  IPv4 pkt |
     |   |                                               + {ingress  |
     +---+                                                  + NHdetails}
     Ingress                                                metadata |
      LFB                                +--------+                  |
                                         | Egress |                  |
                                      <--+        |<-----------------+
                                         |  LFB   |
                                         +--------+
```

Figure 1: Basic IPv4 packet service LFB topology

The IPv4 unicast LPM LFB does a longest prefix match lookup on the IPv4 FIB using the destination IP address as a search key. The result is typically a next hop selector which is passed downstream as metadata.

The Nexthop LFB receives the IPv4 packet with an associated next hop info metadata. The NextHop LFB consumes the NH info metadata and derives from it a table index to look up the next hop table in order to find the appropriate egress information. The lookup result is used to build the next hop details to be used downstream on the egress. This information may include any source and destination information (for our purposes, MAC addresses to use) as well as egress ports. [Note: It is also at this LFB where typically the forwarding TTL decrementing and IP checksum recalculation occurs.]

The details of the egress LFB are considered out of scope for this discussion. Suffice it is to say that somewhere within or beyond the Egress LFB the IPv4 packet will be sent out a port (Ethernet, virtual or physical etc).

3.2.1.1. Distributing The Basic IPv4 Router

Figure 2 demonstrates one way the router LFB topology in Figure 1 may be split across two FEs (eg two ASICs). Figure 2 shows the LFB topology split across FEs after the IPv4 unicast LPM LFB.

```
     FE1
   +-------------------------------------------------------------+
   |                            +----+                           |
   | +----------+               |    |                           |
   | | Ingress  |    IPv4 pkt   |    | IPv4 pkt     +-----+      |
   | |  LFB     +-------------->|    +------------->|     |      |
   | |          |  + ingress    |    | + ingress    |IPv4 |      |
   | +----------+    metadata   |    |   metadata   |Ucast|      |
   |      ^                     +----+              |LPM  |      |
   |      |                      IPv4               +--+--+      |
   |      |                     Validator              |         |
   |                             LFB                   |         |
   +---------------------------------------------------|---------+
                                                       |
                                                  IPv4 packet +
                                                {ingress + NHinfo}
                                                    metadata
     FE2                                               |
   +---------------------------------------------------|---------+
   |                                                   V         |
   |             +--------+                       +--------+     |
   |             | Egress |     IPv4 packet       | IPv4   |     |
   |       <-----+  LFB   |<----------------------+NextHop |     |
   |             |        |{ingress + NHdetails}  | LFB    |     |
   |             +--------+      metadata         +--------+     |
   +-------------------------------------------------------------+
```

Figure 2: Split IPv4 packet service LFB topology

Some proprietary inter-connect (example Broadcom HiGig over XAUI [brcm-higig]) are known to exist to carry both the IPv4 packet and the related metadata between the IPv4 Unicast LFB and IPv4 NextHop LFB across the two FEs.

This document defines the inter-FE LFB, a standard mechanism for encapsulating, generating, receiving and decapsulating packets and associated metadata FEs over Ethernet.

3.2.2. Arbitrary Network Function

In this section we show an example of an arbitrary Network Function which is more coarse grained in terms of functionality. Each Network Function may constitute more than one LFB.

```
     FE1
   +-------------------------------------------------------------+
   |                            +----+                           |
   | +----------+               |    |                           |
```

```
   | | Network  |   pkt         |NF2 |    pkt       +-----+      |
   | | Function +-------------->|    +------------->|     |      |
   | |    1     |  + NF1        |    | + NF1/2      |NF3  |      |
   | +----------+    metadata   |    |   metadata   |     |      |
   |      ^                     +----+              |     |      |
   |      |                                         +--+--+      |
   |      |                                            |         |
   |                                                   |         |
   +---------------------------------------------------|---------+
                                                       V
```

Figure 3: A Network Function Service Chain within one FE

The setup in Figure 3 is a typical of most packet processing boxes where we have functions like DPI, NAT, Routing, etc connected in such a topology to deliver a packet processing service to flows.

3.2.2.1. Distributing The Arbitrary Network Function

The setup in Figure 3 can be split out across 3 FEs instead of as demonstrated in Figure 4. This could be motivated by scale out reasons or because different vendors provide different functionality which is plugged-in to provide such functionality. The end result is to have the same packet service delivered to the different flows passing through.

```
      FE1                        FE2
      +----------+               +----+               FE3
      | Network  |   pkt         |NF2 |    pkt       +-----+
      | Function +-------------->|    +------------->|     |
      |    1     |  + NF1        |    | + NF1/2      |NF3  |
      +----------+    metadata   |    |   metadata   |     |
           ^                     +----+              |     |
           |                                         +--+--+
                                                        |
                                                        V
```

Figure 4: A Network Function Service Chain Distributed Across Multiple FEs

4. Inter-FE LFB Overview

We address the inter-FE connectivity requirements by defining the inter-FE LFB class. Using a standard LFB class definition implies no change to the basic ForCES architecture in the form of the core LFBs (FE Protocol or Object LFBs). This design choice was made after considering an alternative approach that would have required changes

to both the FE Object capabilities (SupportedLFBs) as well LFBTopology component to describe the inter-FE connectivity capabilities as well as runtime topology of the LFB instances.

4.1. Inserting The Inter-FE LFB

The distributed LFB topology described in Figure 2 is re-illustrated in Figure 5 to show the topology location where the inter-FE LFB would fit in.

As can be observed in Figure 5, the same details passed between IPv4 unicast LPM LFB and the IPv4 NH LFB are passed to the egress side of the Inter-FE LFB. This information is illustrated as multiplicity of inputs into the egress InterFE LFB instance. Each input represents a unique set of selection information.

```
     FE1
   +-------------------------------------------------------------+
   | +----------+               +----+                           |
   | | Ingress  |    IPv4 pkt   |    | IPv4 pkt     +-----+      |
   | |  LFB     +--------------->|    +------------->|     |      |
   | |          |  + ingress    |    | + ingress    |IPv4 |      |
   | +----------+    metadata   |    |   metadata   |Ucast|      |
   |      ^                     +----+              |LPM  |      |
   |      |                      IPv4               +--+--+      |
   |      |                     Validator              |         |
   |      |                      LFB                   |         |
   |      |                                  IPv4 pkt + metadata |
   |      |                                   {ingress + NHinfo} |
   |      |                                            |         |
   |      |                                       +..--+..+      |
   |      |                                       |..| |  |      |
   |                                            +-V--V-V--V-+    |
   |                                            |   Egress  |    |
   |                                            |  InterFE  |    |
   |                                            |   LFB     |    |
   |                                            +------+----+    |
   +---------------------------------------------------|---------+
                                                       |
                               Ethernet Frame with:    |
                               IPv4 packet data and metadata
                               {ingress + NHinfo + Inter FE info}
    FE2                                                |
   +---------------------------------------------------|---------+
   |                                                +..+.+..+    |
   |                                                |..|.|..|    |
   |                                              +-V--V-V--V-+  |
   |                                              | Ingress   |  |
```

```
   |                                              | InterFE   |  |
   |                                              |   LFB     |  |
   |                                              +----+------+  |
   |                                                   |         |
   |                                         IPv4 pkt + metadata |
   |                                          {ingress + NHinfo} |
   |                                                   |         |
   |             +--------+                       +----V---+     |
   |             | Egress |     IPv4 packet       | IPv4   |     |
   |       <-----+  LFB   |<----------------------+NextHop |     |
   |             |        |{ingress + NHdetails}  | LFB    |     |
   |             +--------+      metadata         +--------+     |
   +-------------------------------------------------------------+
```

Figure 5: Split IPv4 forwarding service with Inter-FE LFB

The egress of the inter-FE LFB uses the received packet and metadata to select details for encapsulation when sending messages towards the selected neighboring FE. These details include what to communicate as the source and destination FEs (abstracted as MAC addresses as described in Section 5.2); in addition the original metadata may be passed along with the original IPv4 packet.

On the ingress side of the inter-FE LFB the received packet and its associated metadata are used to decide the packet graph continuation. This includes which of the original metadata and which next LFB class instance to continue processing on. In the illustrated Figure 5, an IPv4 Nexthop LFB instance is selected and appropriate metadata is passed on to it.

The ingress side of the inter-FE LFB consumes some of the information passed and passes on the IPv4 packet alongside with the ingress and NHinfo metadata to the IPv4 NextHop LFB as was done earlier in both Figure 1 and Figure 2.

5. Inter-FE Ethernet Connectivity

Section 5.1 describes some of the issues related to using Ethernet as the transport and how we mitigate them.

Section 5.2 defines a payload format that is to be used over Ethernet. An existing implementation of this specification on top of Linux Traffic Control [linux-tc] is described in [tc-ife].

5.1. Inter-FE Ethernet Connectivity Issues

There are several issues that may occur due to using direct Ethernet encapsulation that need consideration.

5.1.1. MTU Consideration

Because we are adding data to existing Ethernet frames, MTU issues may arise. We recommend:

- To use large MTUs when possible (example with jumbo frames).

- Limit the amount of metadata that could be transmitted; our definition allows for filtering of select metadata to be encapsulated in the frame as described in Section 6. We recommend sizing the egress port MTU so as to allow space for maximum size of the metadata total size to allow between FEs. In such a setup, the port is configured to "lie" to the upper layers by claiming to have a lower MTU than it is capable of. MTU setting can be achieved by ForCES control of the port LFB(or other config). In essence, the control plane when explicitly making a decision for the MTU settings of the egress port is implicitly deciding how much metadata will be allowed. Caution needs to be exercised on how low the MTU could be: For IPv4 packets the minimum size is 576 octets [RFC 791] and for IPv6 the minimum size is 1280 octets [RFC2460].

5.1.2. Quality Of Service Considerations

A raw packet arriving at the Inter-FE LFB (from upstream LFB Class instances) may have COS metadatum indicating how it should be treated from a Quality of Service perspective.

The resulting Ethernet frame will be eventually (preferentially) treated by a downstream LFB(typically a port LFB instance) and their COS marks will be honored in terms of priority. In other words the presence of the Inter-FE LFB does not change the COS semantics

5.1.3. Congestion Considerations

Most of the traffic passing through FEs that utilize the Inter-FE LFB is expected to be IP based, which is generally assumed to be congestion controlled [draft-ietf-tsvwg-rfc5405bis]. For example if congestion causes a TCP packet annotated with additional ForCES metadata to be dropped between FEs, the sending TCP can be expected to react in the same fashion as if that packet had been dropped at a different point on its path where ForCES is not involved. For this reason, additional Inter-FE congestion control mechanisms are not specified.

However, the increased packet size due to addition of ForCES metadata is likely to require additional bandwidth on inter-FE links by comparison to what would be required to carry the same traffic

without ForCES metadata. Therefore, traffic engineering SHOULD be done when deploying Inter-FE encapsulation.

Furthermore, the Inter-FE LFB MUST only be deployed within a single network (with a single network operator) or networks of an adjacent set of cooperating network operators where traffic is managed to avoid congestion. These are Controlled Environments, as defined by Section 3.6 of [draft-ietf-tsvwg-rfc5405bis]. Additional measures SHOULD be imposed to restrict the impact of Inter-FE encapsulated traffic on other traffic; example:

- rate limiting at an upstream LFB all Inter-FE LFB traffic

- managed circuit breaking[circuit-b].

- Isolating the Inter-FE traffic either via dedicated interfaces or VLANs.

5.2. Inter-FE Ethernet Encapsulation

The Ethernet wire encapsulation is illustrated in Figure 6. The process that leads to this encapsulation is described in Section 6. The resulting frame is 32 bit aligned.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Destination MAC Address                                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Destination MAC Address       |   Source MAC Address          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Source MAC Address                                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Inter-FE ethertype            | Metadata length               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | TLV encoded Metadata ~~~..............~~                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | TLV encoded Metadata ~~~..............~~                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Original packet data ~~................~~                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Packet format definition

The Ethernet header (illustrated in Figure 6) has the following semantics:

- The Destination MAC Address is used to identify the Destination FEID by the CE policy (as described in Section 6).

- The Source MAC Address is used to identify the Source FEID by the CE policy (as described in Section 6).

- The Ethernet type is used to identify the frame as inter-FE LFB type. Ethertype TBA1 is to be used (XXX: Note to RFC editor, likely we wont get that value - update when available).

- The 16-bit metadata length is used to described the total encoded metadata length (including the 16 bits used to encode the metadata length).

- One or more 16-bit TLV encoded Metadatum follows the metadata length field. The TLV type identifies the Metadata id. ForCES IANA-defined Metadata ids will be used. All TLVs will be 32 bit aligned. We recognize that using a 16 bit TLV restricts the metadata id to 16 bits instead of ForCES-defined component ID space of 32 bits. However, at the time of publication we believe this is sufficient to carry all the info we need; this approach has been selected because it saves us 4 bytes per Metadatum transferred.

- The original packet data payload is appended at the end of the metadata as shown.

6. Detailed Description of the Ethernet inter-FE LFB

The Ethernet inter-FE LFB has two LFB input port groups and three LFB output ports as shown in Figure 7.

The inter-FE LFB defines two components used in aiding processing described in Section 6.2.

```
                    +-----------------+
     Inter-FE LFB   |                 |
     Encapsulated   |             OUT2+--> decapsulated Packet
     -------------->|IngressInGroup   |       + metadata
     Ethernet Frame |                 |
                    |                 |
     raw Packet +   |             OUT1+--> Encapsulated Ethernet
     -------------->|EgressInGroup    |           Frame
     Metadata       |                 |
                    |    EXCEPTIONOUT +--> ExceptionID, packet
                    |                 |           + metadata
                    +-----------------+
```

Figure 7: Inter-FE LFB

6.1. Data Handling

The Inter-FE LFB (instance) can be positioned at the egress of a source FE. Figure 5 illustrates an example source FE in the form of FE1. In such a case an Inter-FE LFB instance receives, via port group EgressInGroup, a raw packet and associated metadata from the preceding LFB instances. The input information is used to produce a selection of how to generate and encapsulate the new frame. The set of all selections is stored in the LFB component IFETable described further below. The processed encapsulated Ethernet Frame will go out on OUT1 to a downstream LFB instance when processing succeeds or to the EXCEPTIONOUT port in the case of a failure.

The Inter-FE LFB (instance) can be positioned at the ingress of a receiving FE. Figure 5 illustrates an example destination FE in the form of FE1. In such a case an Inter-FE LFB receives, via an LFB port in the IngressInGroup, an encapsulated Ethernet frame. Successful processing of the packet will result in a raw packet with associated metadata IDs going downstream to an LFB connected on OUT2. On failure the data is sent out EXCEPTIONOUT.

6.1.1. Egress Processing

The egress Inter-FE LFB receives packet data and any accompanying Metadatum at an LFB port of the LFB instance's input port group labelled EgressInGroup.

The LFB implementation may use the incoming LFB port (within LFB port group EgressInGroup) to map to a table index used to lookup the IFETable table.

If lookup is successful, a matched table row which has the InterFEinfo details is retrieved with the tuple {optional IFEtype, optional StatId, Destination MAC address(DSTFE), Source MAC address(SRCFE), optional metafilters}. The metafilters lists define a whitelist of which Metadatum are to be passed to the neighboring FE. The inter-FE LFB will perform the following actions using the resulting tuple:

- o Increment statistics for packet and byte count observed at corresponding IFEStats entry.

- When MetaFilterList is present, then walk each received Metadatum and apply against the MetaFilterList. If no legitimate metadata is found that needs to be passed downstream then the processing stops and send the packet and metadata out the EXCEPTIONOUT port with exceptionID of EncapTableLookupFailed [RFC6956].

- Check that the additional overhead of the Ethernet header and encapsulated metadata will not exceed MTU. If it does, increment the error packet count statistics and send the packet and metadata out the EXCEPTIONOUT port with exceptionID of FragRequired [RFC6956].

- Create the Ethernet header

- Set the Destination MAC address of the Ethernet header with value found in the DSTFE field.

- Set the Source MAC address of the Ethernet header with value found in the SRCFE field.

- If the optional IFETYPE is present, set the Ethernet type to the value found in IFETYPE. If IFETYPE is absent then the standard Inter-FE LFB Ethernet type TBA1 is used (XXX: Note to RFC editor, to be updated).

- Encapsulate each allowed Metadatum in a TLV. Use the Metaid as the "type" field in the TLV header. The TLV should be aligned to 32 bits. This means you may need to add padding of zeroes at the end of the TLV to ensure alignment.

- Update the Metadata length to the sum of each TLV's space plus 2 bytes (for the Metadata length field 16 bit space).

The resulting packet is sent to the next LFB instance connected to the OUT1 LFB-port; typically a port LFB.

In the case of a failed lookup the original packet and associated metadata is sent out the EXCEPTIONOUT port with exceptionID of EncapTableLookupFailed [RFC6956]. Note that the EXCEPTIONOUT LFB port is merely an abstraction and implementation may in fact drop packets as described above.

6.1.2. Ingress Processing

An ingressing inter-FE LFB packet is recognized by inspecting the ethertype, and optionally the destination and source MAC addresses. A matching packet is mapped to an LFB instance port in the IngressInGroup. The IFETable table row entry matching the LFB

instance port may have optionally programmed metadata filters. In such a case the ingress processing should use the metadata filters as a whitelist of what metadatum is to be allowed.

- o Increment statistics for packet and byte count observed.

- o Look at the metadata length field and walk the packet data extracting from the TLVs the metadata values. For each Metadatum extracted, in the presence of metadata filters, the metaid is compared against the relevant IFETable row metafilter list. If the Metadatum is recognized, and is allowed by the filter, the corresponding implementation Metadatum field is set. If an unknown Metadatum id is encountered, or if the metaid is not in the allowed filter list the implementation is expected to ignore it, increment the packet error statistic and proceed processing other Metadatum. XXX: Alia's comment. is it per packet or metadatum..

- o Upon completion of processing all the metadata, the inter-FE LFB instance resets the data point to the original payload (i.e skips the IFE header information). At this point the original packet that was passed to the egress Inter-FE LFB at the source FE is reconstructed. This data is then passed along with the reconstructed metadata downstream to the next LFB instance in the graph.

In the case of processing failure of either ingress or egress positioning of the LFB, the packet and metadata are sent out the EXCEPTIONOUT LFB port with appropriate error id. Note that the EXCEPTIONOUT LFB port is merely an abstraction and implementation may in fact drop packets as described above.

6.2. Components

There are two LFB components accessed by the CE. The reader is asked to refer to the definitions in Figure 8.

The first component, populated by the CE, is an array known as the IFETable table. The array rows are made up of IFEInfo structure. The IFEInfo structure constitutes: optional IFETYPE, optionally present StatId, Destination MAC address(DSTFE), Source MAC address(SRCFE), optionally present array of allowed Metaids (MetaFilterList).

The second component(ID 2), populated by the FE and read by the CE, is an indexed array known as the IFEStats table. Each IFEStats row which carries statistics information in the structure bstats.

A note about the StatId relationship between the IFETable table and IFEStats table: An implementation may choose to map between an IFETable row and IFEStats table row using the StatId entry in the matching IFETable row. In that case the IFETable StatId must be present. Alternative implementation may map at provisioning time an IFETable row to IFEStats table row. Yet another alternative implementation may choose not to use the IFETable row StatId and instead use the IFETable row index as the IFEStats index. For these reasons the StatId component is optional.

6.3. Inter-FE LFB XML Model

```
   <LFBLibrary xmlns="urn:ietf:params:xml:ns:forces:lfbmodel:1.1"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          provides="IFE">
     <frameDefs>

        <frameDef>
        <name>PacketAny</name>
         <synopsis>Arbitrary Packet</synopsis>
        </frameDef>
        <frameDef>
        <name>InterFEFrame</name>
        <synopsis>
            Ethernet Frame with encapsulate IFE information
        </synopsis>
        </frameDef>

     </frameDefs>

     <dataTypeDefs>

       <dataTypeDef>
          <name>bstats</name>
          <synopsis>Basic stats</synopsis>
       <struct>
       <component componentID="1">
        <name>bytes</name>
        <synopsis>The total number of bytes seen</synopsis>
        <typeRef>uint64</typeRef>
       </component>

       <component componentID="2">
        <name>packets</name>
        <synopsis>The total number of packets seen</synopsis>
        <typeRef>uint32</typeRef>
       </component>
```

```
       <component componentID="3">
        <name>errors</name>
        <synopsis>The total number of packets with errors</synopsis>
        <typeRef>uint32</typeRef>
       </component>
       </struct>

      </dataTypeDef>

        <dataTypeDef>
           <name>IFEInfo</name>
       <synopsis>Describing IFE table row Information</synopsis>
           <struct>
              <component componentID="1">
                <name>IFETYPE</name>
            <synopsis>
            the ethernet type to be used for outgoing IFE frame
            </synopsis>
            <optional/>
                <typeRef>uint16</typeRef>
              </component>
              <component componentID="2">
                <name>StatId</name>
            <synopsis>
            the Index into the stats table
            </synopsis>
            <optional/>
                <typeRef>uint32</typeRef>
              </component>
              <component componentID="3">
                <name>DSTFE</name>
            <synopsis>
                the destination MAC address of destination FE
            </synopsis>
                <typeRef>byte[6]</typeRef>
              </component>
              <component componentID="4">
                <name>SRCFE</name>
            <synopsis>
                the source MAC address used for the source FE
            </synopsis>
                <typeRef>byte[6]</typeRef>
              </component>
              <component componentID="5">
                <name>MetaFilterList</name>
            <synopsis>
                the allowed metadata filter table
            </synopsis>
```

```
            <optional/>
                <array type="variable-size">
                  <typeRef>uint32</typeRef>
                </array>
               </component>


           </struct>
        </dataTypeDef>

     </dataTypeDefs>

     <LFBClassDefs>
       <LFBClassDef LFBClassID="18">
         <name>IFE</name>
         <synopsis>
            This LFB describes IFE connectivity parameterization
         </synopsis>
         <version>1.0</version>

       <inputPorts>

         <inputPort group="true">
          <name>EgressInGroup</name>
          <synopsis>
              The input port group of the egress side.
              It expects any type of Ethernet frame.
          </synopsis>
          <expectation>
           <frameExpected>
           <ref>PacketAny</ref>
           </frameExpected>
          </expectation>
         </inputPort>

         <inputPort  group="true">
          <name>IngressInGroup</name>
          <synopsis>
              The input port group of the ingress side.
              It expects an interFE encapsulated Ethernet frame.
           </synopsis>
          <expectation>
           <frameExpected>
           <ref>InterFEFrame</ref>
           </frameExpected>
          </expectation>
       </inputPort>
```

```
          </inputPorts>

          <outputPorts>

            <outputPort>
              <name>OUT1</name>
              <synopsis>
                   The output port of the egress side.
              </synopsis>
              <product>
                 <frameProduced>
                   <ref>InterFEFrame</ref>
                 </frameProduced>
              </product>
           </outputPort>

           <outputPort>
             <name>OUT2</name>
             <synopsis>
                 The output port of the Ingress side.
             </synopsis>
             <product>
                <frameProduced>
                  <ref>PacketAny</ref>
                </frameProduced>
             </product>
          </outputPort>

          <outputPort>
            <name>EXCEPTIONOUT</name>
            <synopsis>
               The exception handling path
            </synopsis>
            <product>
               <frameProduced>
                 <ref>PacketAny</ref>
               </frameProduced>
               <metadataProduced>
                 <ref>ExceptionID</ref>
               </metadataProduced>
            </product>
         </outputPort>

      </outputPorts>

      <components>

         <component componentID="1" access="read-write">
```

```
            <name>IFETable</name>
            <synopsis>
               the table of all InterFE relations
            </synopsis>
            <array type="variable-size">
               <typeRef>IFEInfo</typeRef>
            </array>
         </component>

        <component componentID="2" access="read-only">
          <name>IFEStats</name>
          <synopsis>
           the stats corresponding to the IFETable table
          </synopsis>
          <typeRef>bstats</typeRef>
        </component>

     </components>

    </LFBClassDef>
   </LFBClassDefs>

   </LFBLibrary>
```

Figure 8: Inter-FE LFB XML

7. Acknowledgements

The authors would like to thank Joel Halpern and Dave Hood for the stimulating discussions. Evangelos Haleplidis shepherded and contributed to improving this document. Alia Atlas was the AD sponsor of this document and did a tremendous job of critiquing it. The authors are grateful to Joel Halpern and Sue Hares in their roles as the Routing Area reviewers in shaping the content of this document. David Black put a lot of effort in making sure congestion control considerations are sane. Russ Housley did the Gen-ART review and Joe Touch did the TSV area.

8. IANA Considerations

This memo includes one IANA request within the registry https://www.iana.org/assignments/forces

The request is for the sub-registry "Logical Functional Block (LFB) Class Names and Class Identifiers" to request for the reservation of LFB class name IFE with LFB classid 18 with version 1.0.

LFB Class Identifier	LFB Class Name	LFB Version	Description	Reference
18	IFE	1.0	An IFE LFB to standardize inter-FE LFB for ForCES Network Elements	This document

Logical Functional Block (LFB) Class Names and Class Identifiers

9. IEEE Assignment Considerations

This memo includes a request for a new ethernet protocol type as described in Section 5.2.

10. Security Considerations

The FEs involved in the Inter-FE LFB belong to the same Network Device (NE) and are within the scope of a single administrative Ethernet LAN private network. Trust of policy in the control and its treatment in the datapath exists already.

This document does not alter [RFC5812] or the ForCES Protocol[RFC5810]. As such, it has no impact on these documents security considerations. This document simply defines the operational parameters and capabilities of an LFB that performs LFB class instance extensions across nodes under a single administrative control. This document does not attempt to analyze the presence or possibility of security interactions created by allowing LFB graph extension on packets. Any such issues, if they exist should be resolved by the designers of the particular data path i.e they are not the responsibility of general mechanism outlined in this document; one such option for protecting Ethernet is the use of IEEE 802.1AE Media Access Control Security [ieee8021ae] which provides encryption and authentication.

11. References

11.1. Normative References

[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>.

[RFC5811] Hadi Salim, J. and K. Ogawa, "SCTP-Based Transport Mapping Layer (TML) for the Forwarding and Control Element Separation (ForCES) Protocol", RFC 5811, DOI 10.17487/RFC5811, March 2010, <http://www.rfc-editor.org/info/rfc5811>.

[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>.

[RFC7391] Hadi Salim, J., "Forwarding and Control Element Separation (ForCES) Protocol Extensions", RFC 7391, DOI 10.17487/RFC7391, October 2014, <http://www.rfc-editor.org/info/rfc7391>.

[RFC7408] Haleplidis, E., "Forwarding and Control Element Separation (ForCES) Model Extension", RFC 7408, DOI 10.17487/RFC7408, November 2014, <http://www.rfc-editor.org/info/rfc7408>.

[draft-ietf-tsvwg-rfc5405bis]
Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", Nov 2015, <https://tools.ietf.org/html/draft-ietf-tsvwg-rfc5405bis-07>.

11.2. Informative 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>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[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>.

[RFC6956] Wang, W., Haleplidis, E., Ogawa, K., Li, C., and J. Halpern, "Forwarding and Control Element Separation (ForCES) Logical Function Block (LFB) Library", RFC 6956, DOI 10.17487/RFC6956, June 2013, <http://www.rfc-editor.org/info/rfc6956>.

[brcm-higig] , "HiGig", <http://www.broadcom.com/products/brands/HiGig>.

[circuit-b] Fairhurst, G., "Network Transport Circuit Breakers", Feb 2016, <https://tools.ietf.org/html/draft-fairhurst-tsvwg-circuit-breaker-13>.

[ieee8021ae] , "IEEE Standard for Local and metropolitan area networks Media Access Control (MAC) Security", IEEE 802.1AE-2006, Aug 2006.

[linux-tc] Hadi Salim, J., "Linux Traffic Control Classifier-Action Subsystem Architecture", netdev 01, Feb 2015.

[tc-ife] Hadi Salim, J. and D. Joachimpillai, "Distributing Linux Traffic Control Classifier-Action Subsystem", netdev 01, Feb 2015.

[vxlan-udp] , "iproute2 and kernel code (drivers/net/vxlan.c)", <https://www.kernel.org/pub/linux/utils/net/iproute2/>.

Authors' Addresses

Damascane M. Joachimpillai
Verizon
60 Sylvan Rd
Waltham, Mass. 02451
USA

Email: damascene.joachimpillai@verizon.com

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

Email: hadi@mojatatu.com