Internet Engineering Task Force (IETF)
Request for Comments: 8013
Category: Standards Track
ISSN: 2070-1721

D. Joachimpillai
Verizon
J. Hadi Salim
Mojatatu Networks

February 2017

Forwarding and Control Element Separation (ForCES) Inter-FE Logical Functional Block (LFB)

Abstract

This document describes how to extend the Forwarding and Control Element Separation (ForCES) Logical Functional Block (LFB) topology across Forwarding Elements (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

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 7841.

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/rfc8013.

Table of Contents

1. Introduction

In the ForCES architecture, a packet service can be modeled 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 Functional Blocks (LFBs), including provision for the Control Element (CE) to establish and modify that processing sequence, and the parameters of the individual LFBs.

Under some circumstances, 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 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.

2. Terminology and Conventions

2.1. 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].

2.2. Definitions

This document depends on the terms (below) defined in several ForCES documents: [RFC3746], [RFC5810], [RFC5811], [RFC5812], [RFC7391], and [RFC7408].

Control Element (CE)

Forwarding Element (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)

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 that is in close proximity.

- The FEs are already interconnected using Ethernet. We focus on Ethernet because it is commonly used for FE interconnection. 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 and 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 a 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 purpose of the illustration in Figure 1 is 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 to the LFBLibrary 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, tunneling information, etc. These metadata are lumped together as "ingress metadata".

Once the IPv4 validator vets the packet (for example, it ensures that there is no expired TTL), it feeds the packet and inherited metadata into the IPv4 unicast LPM (Longest-Prefix-Matching) LFB.

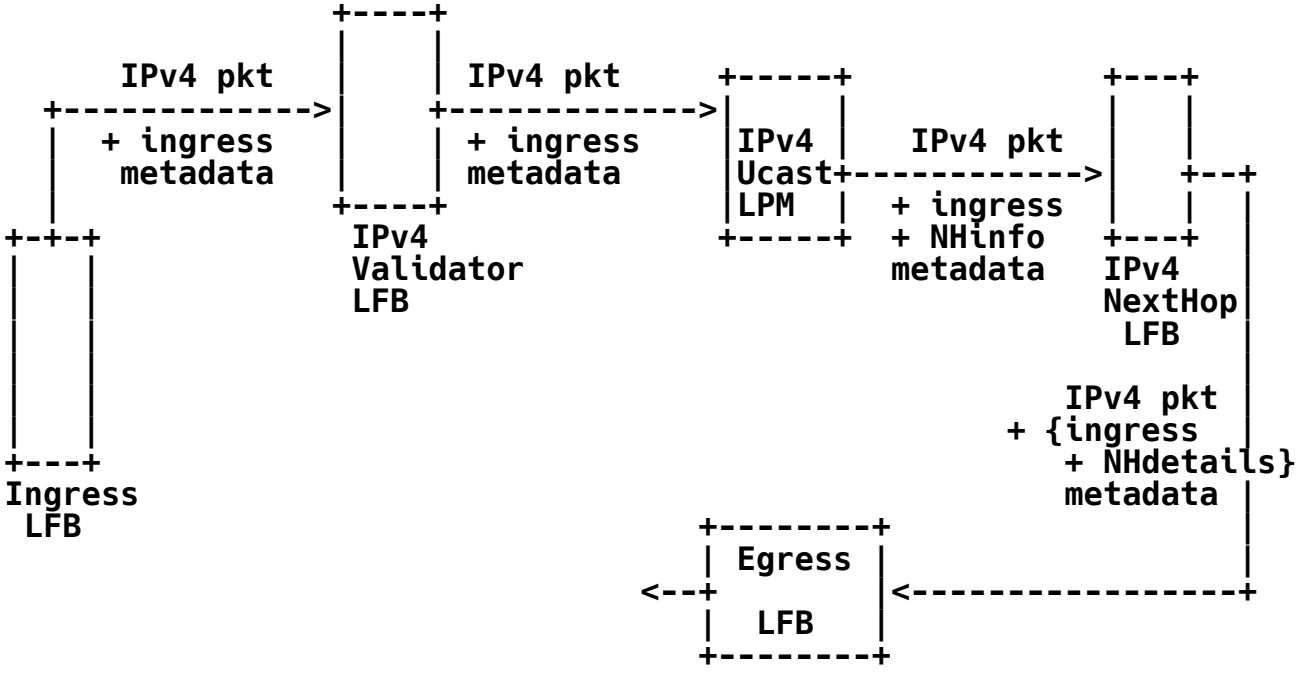

Figure 1: Basic IPv4 Packet Service LFB Topology

The IPv4 unicast LPM LFB does an LPM 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 associated next-hop (NH) information metadata. The NextHop LFB consumes the NH information metadata and derives a table index from it 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, which Media Access Control (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 to say that somewhere within or beyond the Egress LFB, the IPv4 packet will be sent out a port (e.g., Ethernet, virtual or physical).

3.2.1.1. Distributing the Basic IPv4 Router

Figure 2 demonstrates one way that the router LFB topology in Figure 1 may be split across two FEs (e.g., two Application-Specific Integrated Circuits (ASICs)). Figure 2 shows the LFB topology split across FEs after the IPv4 unicast LPM LFB.

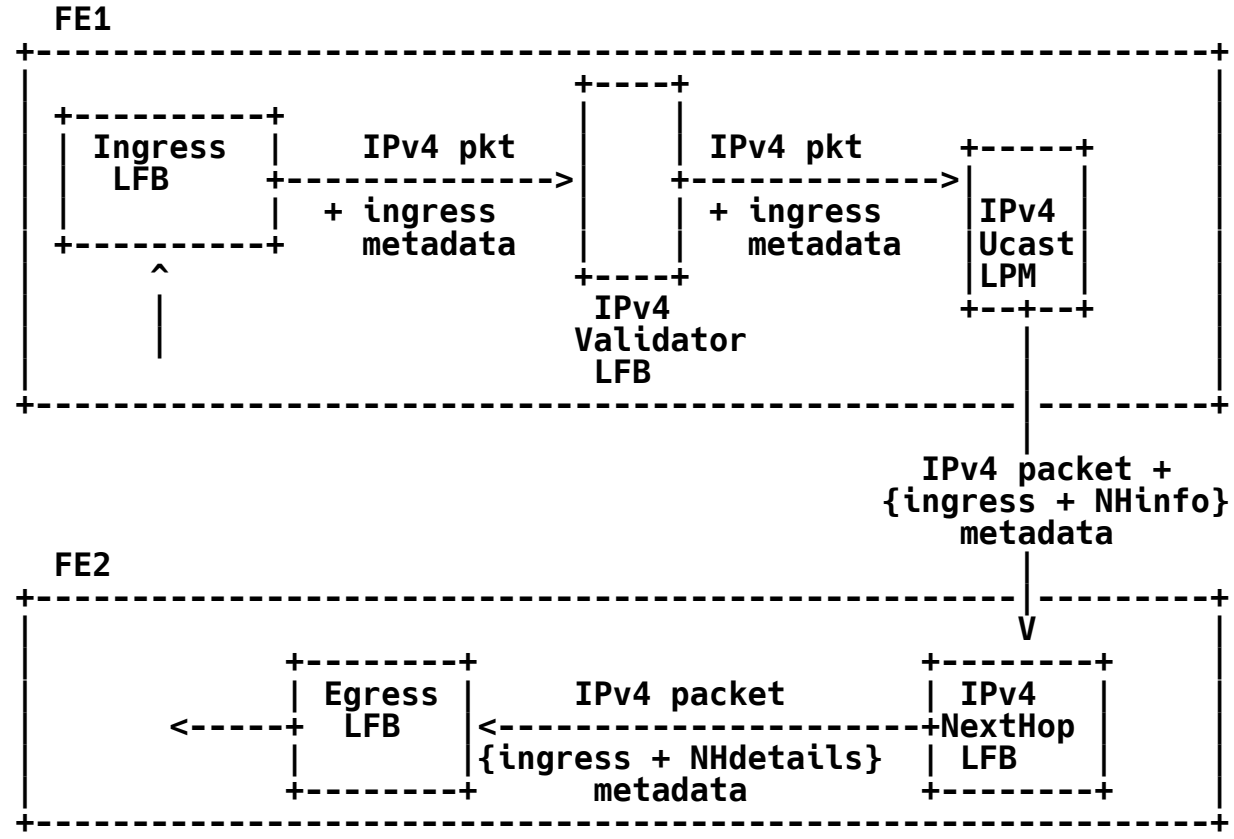

Figure 2: Split IPv4 Packet Service LFB Topology

Some proprietary interconnections (for 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 IPv4NextHop 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 that is more coarsely grained in terms of functionality. Each Network Function may constitute more than one LFB.

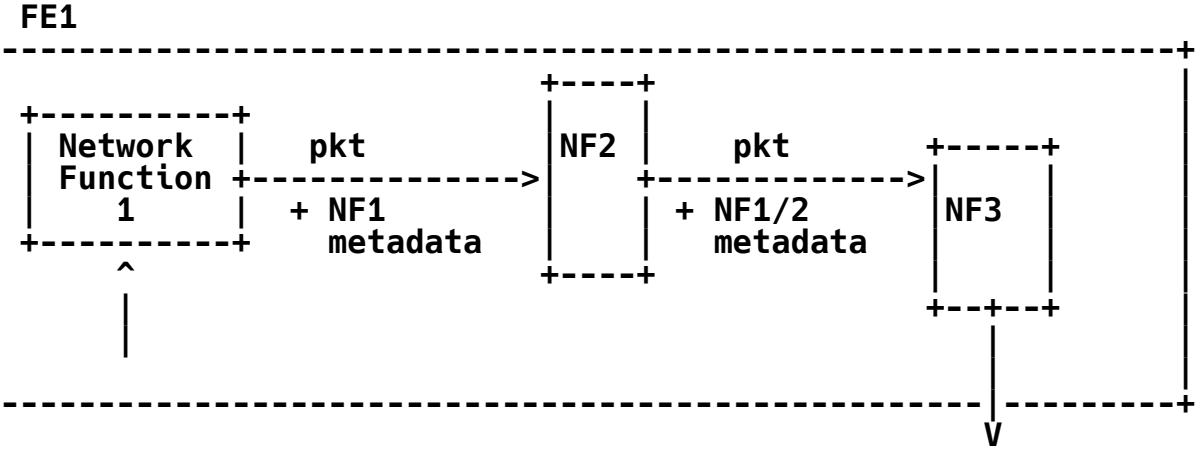

Figure 3: A Network Function Service Chain within One FE

The setup in Figure 3 is typical of most packet processing boxes where we have functions like deep packet inspection (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 across three 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 having the same packet service delivered to the different flows passing through.

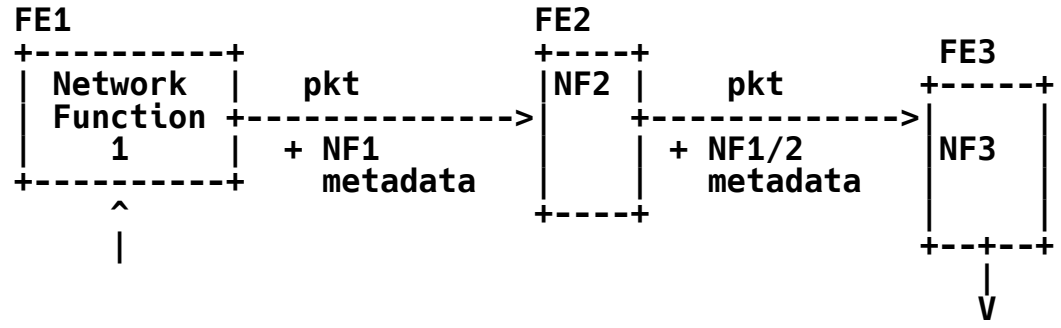

Figure 4: A Network Function Service Chain Distributed across Multiple FEs

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) and the LFBTopology component to describe the inter-FE connectivity capabilities as well as the runtime topology of the LFB instances.

4.1. Inserting the Inter-FE LFB ne 15

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 inter-FE LFB instance. Each input represents a unique set of selection information.

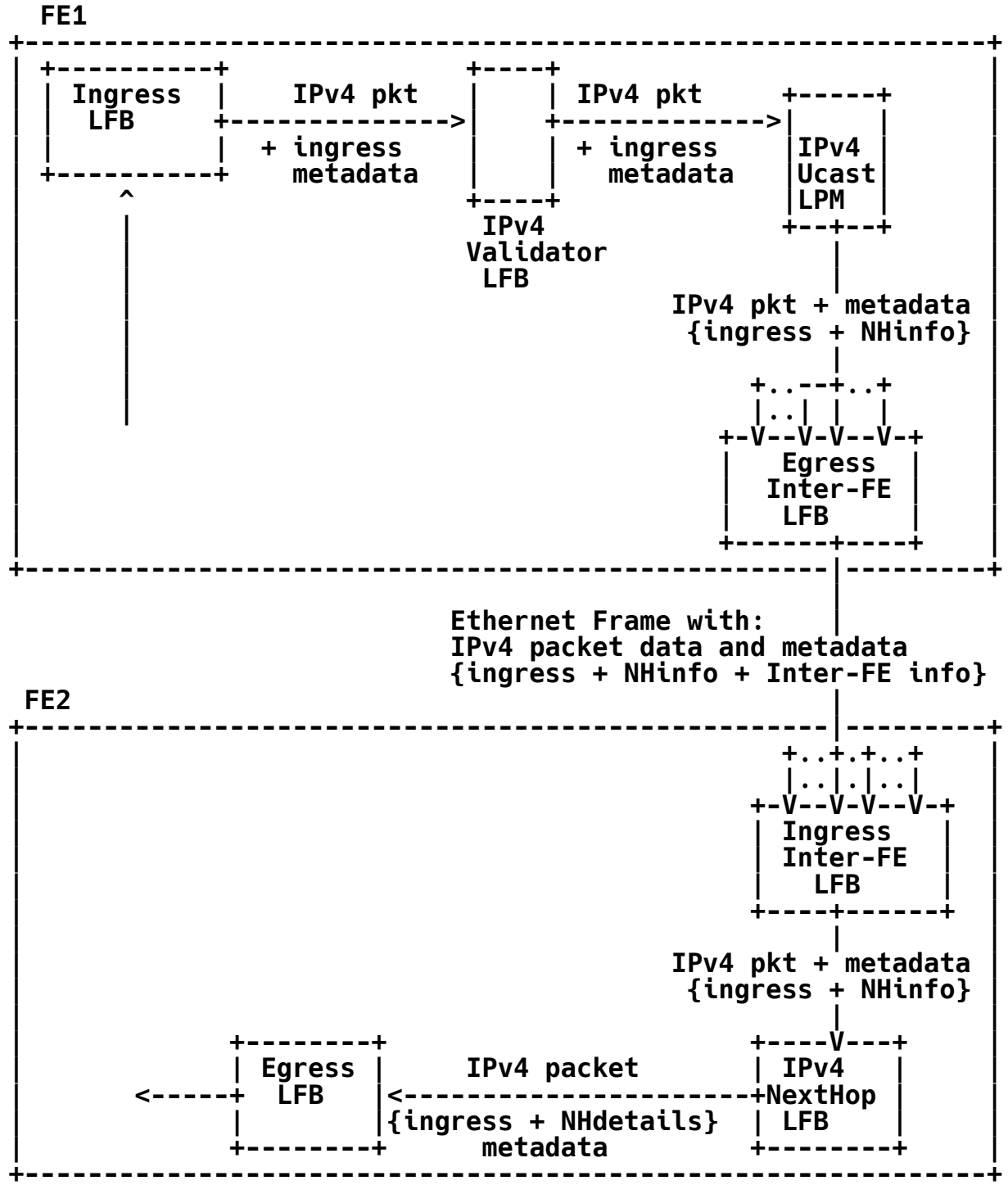

Figure 5: Split IPv4-Forwarding Service with Inter-FE LFB

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 on which next LFB class instance to continue processing. In Figure 5, an IPv4NextHop LFB instance is selected and the appropriate metadata is passed to it.

The ingress side of the inter-FE LFB consumes some of the information passed and passes it the IPv4 packet alongside with the ingress and NHinfo metadata to the IPv4NextHop LFB as was done earlier in both Figures 1 and 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 that runs 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:

- Using large MTUs when possible (example with jumbo frames).
- Limiting 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. Setting the MTU can be achieved by ForCES control of the port LFB (or some other

configuration. 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 resulting reported link MTU could be: for IPv4 packets, the minimum size is 64 octets [RFC791] 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 Class-of-Service (CoS) metadata 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 [UDP-GUIDE]. 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 the addition of ForCES metadata is likely to require additional bandwidth on inter-FE links in 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 [UDP-GUIDE]. Additional measures SHOULD be imposed to restrict the impact of inter-FE-encapsulated traffic on other traffic; for example:

- rate-limiting all inter-FE LFB traffic at an upstream LFB
- managing 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.

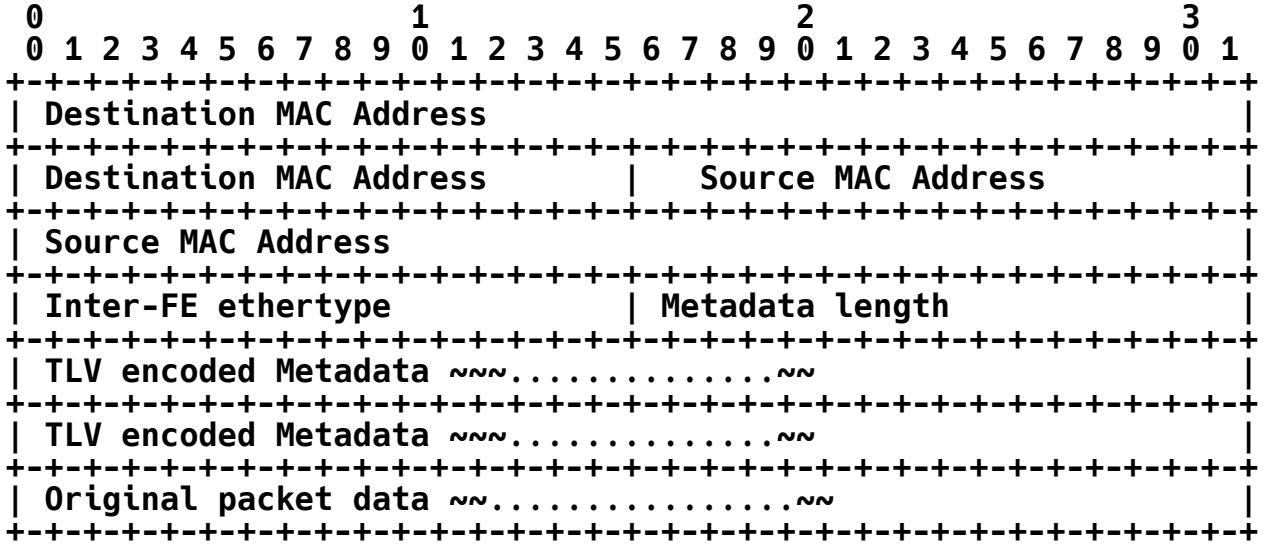

Figure 6: Packet Format Definition

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 ethertype is used to identify the frame as inter-FE LFB type. Ethertype ED3E (base 16) is to be used.
- The 16-bit metadata length is used to describe 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 metadata IDs that have been registered with IANA 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 a ForCES-defined component ID space of 32 bits if an Index-Length-Value (ILV) is used. However, at the time of publication, we believe this is sufficient to carry all the information we need; the TLV approach has been selected because it saves us 4 bytes per metadatum transferred as compared to the ILV approach.

- o 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 6.1.

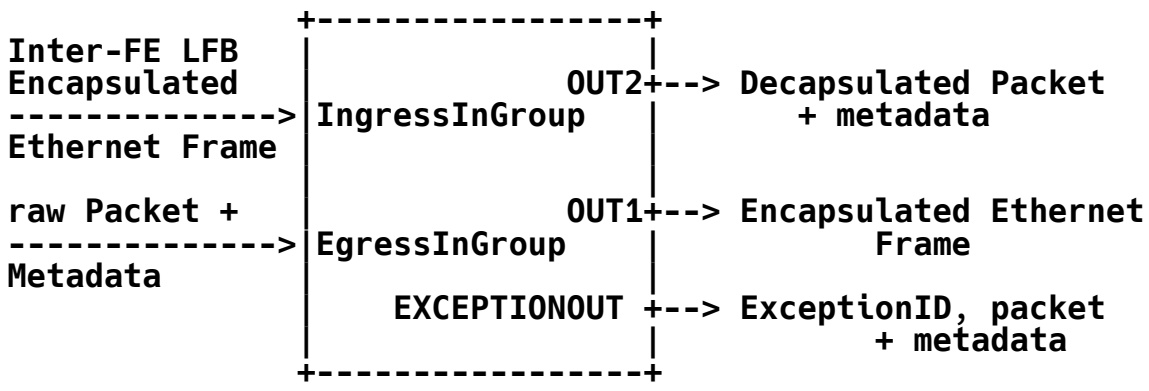

Figure 7: Inter-FE LFB

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 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 labeled EgressInGroup.

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

If the lookup is successful, a matched table row that has the IFEInfo details is retrieved with the tuple (optional IFETYPE, optional StatId, Destination MAC address (DSTFE), Source MAC address (SRCFE), and 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:

- Increment statistics for packet and byte count observed at the corresponding IFEStats entry.
- When the MetaFilterList is present, walk each received metadatum and apply it against the MetaFilterList. If no legitimate metadata is found that needs to be passed downstream, then the processing stops and the packet and metadata are sent out the EXCEPTIONOUT port with the 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 the exceptionID of FragRequired [RFC6956].
- Create the Ethernet header.
- Set the Destination MAC address of the Ethernet header with the value found in the DSTFE field.

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

- If the optional IFETYPE is present, set the ethertype to the value found in IFETYPE. If IFETYPE is absent, then the standard inter-FE LFB ethertype ED3E (base 16) is used.

- 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 a 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 (a 16-bit space for the Metadata length field).

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 the 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.

- Increment statistics for packet and byte count observed.

- Look at the metadata length field and walk the packet data, extracting the metadata values from the TLVs. 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 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, then the implementation is expected to ignore it, increment the packet error statistic, and proceed processing other metadatum.

- 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 a processing failure of either ingress or egress positioning of the LFB, the packet and metadata are sent out the EXCEPTIONOUT LFB port with the 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 the optional IFETYPE, the optionally present StatId, the Destination MAC address (DSTFE), the Source MAC address (SRCFE), and an 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 carries statistics information in the structure bstats.

A note about the StatId relationship between the IFETable table and the 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. An alternative implementation may map an IFETable row to an IFEStats table row at provisioning time. 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 encapsulated 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 ethertype 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 the 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 inter-FE 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. IANA Considerations

IANA has registered the following LFB class name in the "Logical Functional Block (LFB) Class Names and Class Identifiers" subregistry of the "Forwarding and Control Element Separation (ForCES)" registry <https://www.iana.org/assignments/forces>.

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

8. IEEE Assignment Considerations

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

9. Security Considerations

The FEs involved in the inter-FE LFB belong to the same NE and are within the scope of a single administrative Ethernet LAN private network. While trust of policy in the control and its treatment in the datapath exists already, an inter-FE LFB implementation SHOULD support security services provided by Media Access Control Security (MACsec) [ieee8021ae]. MACsec is not currently sufficiently widely deployed in traditional packet processing hardware although it is present in newer versions of the Linux kernel (which will be widely deployed) [linux-macsec]. Over time, we expect that most FEs will be able to support MACsec.

MACsec provides security services such as a message authentication service and an optional confidentiality service. The services can be configured manually or automatically using the MACsec Key Agreement (MKA) over the IEEE 802.1x [ieee8021x] Extensible Authentication Protocol (EAP) framework. It is expected that FE implementations are going to start with shared keys configured from the control plane but progress to automated key management.

The following are the MACsec security mechanisms that need to be in place for the inter-FE LFB:

- Security mechanisms are NE-wide for all FEs. Once the security is turned on, depending upon the chosen security level (e.g., Authentication, Confidentiality), it will be in effect for the inter-FE LFB for the entire duration of the session.

- An operator SHOULD configure the same security policies for all participating FEs in the NE cluster. This will ensure uniform operations and avoid unnecessary complexity in policy configuration. In other words, the Security Association Keys (SAKs) should be pre-shared. When using MKA, FEs must identify themselves with a shared Connectivity Association Key (CAK) and Connectivity Association Key Name (CKN). EAP-TLS SHOULD be used as the EAP method.

- An operator SHOULD configure the strict validation mode, i.e., all non-protected, invalid, or non-verifiable frames MUST be dropped.

It should be noted that given the above choices, if an FE is compromised, an entity running on the FE would be able to fake inter-FE or modify its content, causing bad outcomes.

10. References

10.1. Normative References

[ieee8021ae] IEEE, "IEEE Standard for Local and metropolitan area networks Media Access Control (MAC) Security", IEEE 802.1AE-2006, DOI 10.1109/IEEESTD.2006.245590, <http://ieeexplore.ieee.org/document/1678345/>.

[ieee8021x] IEEE, "IEEE Standard for Local and metropolitan area networks - Port-Based Network Access Control.", IEEE 802.1X-2010, DOI 10.1109/IEEESTD.2010.5409813, <http://ieeexplore.ieee.org/document/5409813/>.

[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>.

[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>.

10.2. Informative References

[brcm-higig]
Broadcom, "HiGig", <http://www.broadcom.com/products/ethernet-communication-and-switching/switching/bcm56720>.

[circuit-b]
Fairhurst, G., "Network Transport Circuit Breakers", Work in Progress, draft-ietf-tsvwg-circuit-breaker-15, April 2016.

[linux-macsec]
Dubroca, S., "MACsec: Encryption for the wired LAN", Netdev 11, Feb 2016.

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

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <http://www.rfc-editor.org/info/rfc791>.

[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>.

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

[UDP-GUIDE]
Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", Work in Progress, draft-ietf-tsvwg-rfc5405bis-19, October 2016.

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 for shaping the content of this document. David Black put in a lot of effort to make sure the congestion-control considerations are sane. Russ Housley did the Gen-ART review, Joe Touch did the TSV area review, and Shucheng LIU (Will) did the OPS review. Suresh Krishnan helped us provide clarity during the IESG review. The authors are appreciative of the efforts Stephen Farrell put in to fixing the security section.

Authors' Addresses

Damascane M. Joachimpillai
Verizon
60 Sylvan Rd
Waltham, MA 02451
United States of America

Email: damascene.joachimpillai@verizon.com

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

Email: hadi@mojatatu.com