

ForCES Forwarding Element Model

`<draft-ietf-forces-model-02.txt>`

Steven Blake

Presentation available at:

[http://www.petri-meet.com/slblake/
networking/drafts/forces-model-02.pdf](http://www.petri-meet.com/slblake/networking/drafts/forces-model-02.pdf)

Design Team

- Lily Yang <lily.y.yang@intel.com>
- Ellen Deleganes <ellen.m.deleganes@intel.com>
- Joel Halpern <jhalpern@megisto.com>
- Ram Gopal <ram.gopal@nokia.com>
- Alan DeKok <alan.dekok@idt.com>
- Zsolt Haraszti <zsolt.haraszti@ericsson.com>
- Steven Blake <steven.blake@ericsson.com>

Changes in -02

- Section 3.2.4 “Metadata” rewritten, incorporating the metadata model submitted to the list.
- Section 4 “Model and Schema for LFB Classes” rewritten, including formal XML schema included in the section.
- Section 5 “FE Attributes and Capabilities” completed, including formal XML schema.
- -01 Section 6 “LFB topology” section deleted, topology is included in Section 5.
- Section 6 “LFB Class Library” expanded to include definitions for 18 LFB classes .
- Section 7 “Satisfying the Requirements on FE Model” revised considerably.

Metadata model

- Metadata are modeled as <tag, value> pairs:
 - tag defines the metadata name.
 - No two metadata will have the same tag value at any particular point in the LFB topology.
- Metadata life-cycle: write, read, consume.
- Metadata operations:
 - Ignore
 - Read
 - Read/Re-write
 - Write (Create)
 - Read/Consume
 - Consume

Metadata model (2)

- Metadata Production and Consumption:
 - Unconditional metadata always written.
 - Conditional metadata may be written under certain conditions.
 - Required metadata is always read.
 - Optional metadata is read if available; a default value is assumed otherwise.

Metadata model (3)

- Fixed, Variable, and Configurable Tag:
 - Fixed: LFB instances always write/read a metadata with a particular tag.
 - Variable: LFB classes allow a metadata to be written/read with any tag value from a set. The tag value written or read will be either a fixed or configurable attribute of a LFB instance.
 - Configurable: LFB instance allows a variable tag value for a metadata to be configured.

Metadata model (4)

- Metadata Usage Categories:
 - Relational (binding):
 - Used to bind two attributes in separate LFB instances.
 - Ex/ bind prefix attribute (LPM Classifier LFB) to next hop attribute (Next Hop LFB).
 - Value is communicated to both LFBs using the ForCES protocol. Numeric value is irrelevant to the CE, which treats it like a handle or file descriptor.
 - Enumerated:
 - Small set of values with an explicit meaning.
 - Ex/ Color
 - Explicit/external value:
 - Value used to overwrite a packet header field.
 - Ex/ Destination MAC address

Metadata model (5)

- Remember, it's only a model:
 - LFB boundaries and the metadata exchanged between them are abstractions useful to the ForCES protocol.
 - Implementations are free to implement (encode, store, prepend) metadata any way they choose.
 - An implementation component may subsume two or more LFBs, so that there is no metadata exchange between these “LFBs” within the implementation.

Model and Schema for LFB Classes

- Re-call:
 - FE is modeled by a graph of LFBs.
 - Attributes of LFBs will be manipulated via the ForCES protocol.
- LFBs are instantiated from templates, called *LFB classes*.
- LFBLibrary schema defines the major elements used when creating the formal definition of a LFB Class.
- Formal specification of LFB classes will be provided as XML documents using the LFBLibrary schema. The actual class documents will be the subject of other I-Ds.

LFBLibrary schema overview

- **Namespace:**
`http://ietf.org/forces/1.0/lfbmodel`
- **Root element:** `<LFBLibrary>`.
- A unique name is assigned to each library file, used for referencing by other XML files.
- `<LFBLibrary>` can refer to definitions in other `<LFBLibrary>`'s, but these files have to be explicitly mentioned in `<load>` statements (similar to `#include` in C).

LFBLibrary schema overview (2)

- Body of the XML file: One or more of the following four blocks:
 - `<frameDefs>`: Declares frame (packet) types and names them. Referred to by LFB input/output port definitions.
 - `<dataTypeDefs>`: Defines common data types and names them. Referred to by metadata definitions and LFB attribute definitions.
 - `<metadataDefs>`: Defines metadata and names them. Referred to by LFB input/output port definitions.
 - `<LFBClassDefs>`: Consists of one or more `<LFBClassDef>` elements, each defining an LFB class.

LFBLibrary schema overview (3)

- **Each** `<LFBClassDef>` **consists of:**
 - `<name>`: **Unique name of LFB class.**
 - `<synopsis>`: **1-line description of LFB class.**
 - `<version>`: **Version number.**
 - `<derivedFrom>`: **Reference to parent class (if any).**
 - `<inputPorts>`: **Definition of input ports of LFB.**
 - `<outputPorts>`: **Definition of output ports.**
 - `<attributes>`: **List of LFB attributes.**
 - `<capabilities>`: **List of LFB capability attributes.**
 - `<description>`: **Detailed description of modeled operation.**

FE Attributes and Capabilities

- XML schema (FEDocument) defines the document structure:
 - Abstract representation: a way to model the FE attributes and capabilities for a particular FE instance.
 - FE instance may not require an FEDocument; protocol will convey equivalent information in some format.
- Includes an example FEDocument:
 - Specifies the connectivity/capabilities of an LFB instance.
 - Specifies several LFB instances.
 - Specifies LFB topology.

FEDocument schema overview

- **Root element:** <FEDocument>.
- <FECapabilities>
 - <ModifiableLFBTopology>
 - <SupportedLFBs> **and** <Supported LFB>
 - <SupportedAttributes>
- <FEAttributes>
 - <FEStatus>
 - <LFBInstances> **and** <LFBInstance>
 - <LFBTopology> **and** <LFBLink>
 - <FEConfiguredNeighbors> **and** <FEConfiguredNeighbor>

LFB Class Library

- Defines 18 LFB classes, including the most commonly found functions for IP forwarding, processing and QoS:
 - Class description
 - Example attributes, static and configurable
 - Potential sub-classes
 - Possible metadata
- Still very preliminary:
 - Library is not exhaustive.
 - Section is meant as an overview; details will be provided in per-class LFBLibrary documents.
 - We won't know if we have the details correct until we work through some sample topologies and then try to model existing implementations.

LFB Class Library (2)

- Port
- L2 Interface
- IP Interface
- Classifier
- Next Hop
- Rate Meter
- Redirector
- Packet Header Rewriter
- Counter
- Dropper
- IPv4 Fragmenter
- L2 Address Resolution
- Queue
- Scheduler
- MPLS ILM/Decapsulation
- MPLS Encapsulation
- Tunnel Encap/Decap
- Replicator

Open Issues

- Is augmentation useful? Necessary?
- How useful are input groups?
- In which document(s) will the protocol payload associated with LFB attributes be defined?
 - ForCES model?
 - LFB Class definitions?
 - Protocol specification?
- Design team feels that there are good answers to other comments on the list.

Pending work items for -03

- Define rules and schema extensions for LFB class derivation via inheritance.
- Support for data type augmentation (if needed).
- Inclusion of type and class IDs (if needed).
- Rules for version handling.
- Provide an example LFB class definition using the LFBLibrary schema.
- Improve schema representation of capability attributes for an associated attribute:
 - Ex/ Express association between an array size capability and the associated array attribute.
- Harmonize LFBLibrary and FEDocument schema.