ForCES LFB Library
<draft-ietf-forces-lfb-lib-00 >

Authors
Weiming Wang, wmwang@mail.zjgsu.edu.cn
Evangelos Haleplidis, ehalep@ece.upatras.gr
Kentaro Ogawa, ogawa.kentaro@lab.ntt.co.jp
Fenggen Jia, jfg@mail.ndsc.com.cn
Halpern Joel, jhalpern@redback.com

Contributors
Jamal Hadi Salim, hadi@mojatatu.com
Ligang Dong, donglg@mail.zjgsu.edu.cn

IETF 75th Meeting
July 28, 2009, Stockholm, Sweden
Draft Status

• Initial draft

• Rough merge of two documents and new contributions from authors
  – draft-dong-forces-lfblib-00
  – draft-halpern-forces-lfblibrary-base-00

• More editorial work to be done
  – Better text organization.
  – Better descriptions on the design logics of the LFBs.

• More issues to be settled
  – Need to review and modify LFB components etc. one by one

• LFB use cases to be done
Draft Contents

• Base definitions
  – frame types
  – data types
  – metadata types

• Descriptions of 29 LFBs grouped into 6 categories by functionality
  – Core LFBs
  – Port LFBs
  – Address LFBs
  – Forwarding LFBs
  – Queue manager and scheduler LFBs
  – Miscellaneous LFBs

• Complete XML definitions of the library
  – Already over 60 pages of text
Types Definitions

• Frame types defined, like
  – EthernetII - An Ethernet II frame type.

• Based on base atomic data types defined by FE model, new data types defined, such as:
  – ifIndex - A Port Identifier.
  – IEEEMAC - IEEE MAC Address.
  – NetSpeedType - Network speed values.

• Meta data types defined, like:
  – ErrorId - Error Type.
  – QueueID - The queue ID
  – NextHopID - An index into a Next Hop entry in NextHop table

• Need review and modification
• Need more definitions
LFB Design Logics - Core LFBs

- Referred to ForCES protocol and FE model for the core LFBs
  - FE Object LFB
  - FE Protocol LFB
LFB Design Logics - Port LFBs

• GenericConnectivityLFB
  – Now it is vacuum in components, may put base and common port components, capabilities, and events into it.

• By inheritance, defined port LFBs for Ethernet, like
  – EtherPort
  – EtherDecap
  – EtherEncap

• Questions and comments:
  – Should the three LFBs be condensed into one Ethernet port LFB?
  – Joel: When a construct can be and frequently is modeled as two separate pieces, and can also be modeled as one piece, it is probably better to simply model it as two pieces.
# Port LFBs components

| EtherPort   | IfIndex          |
|            | IfName           |
|            | LinkSpeed        |
|            | MTU              |
|            | OperaStatus      |
|            | AdminStatus      |
|            | PromiscuousMode  |
|            | CarrierStatus    |
|            | OperMode         |
|            | SrcNegotiationTypeMACAddr |
|            | MacAliasTable    |
|            | StatsEnable      |
|            | PortStats        |
| EtherEncap  | DCHostTablev4/DCHostTablev6 |
| EtherDecap  | ArpTable/NbrTable |
|            | DispatchTable    |
LFB Design Logics - Address LFBs

- LFBs for address handling tasks such as:
  - ARP
  - IPv6AddrResolution
  - ICMPGenerator
  - ICMPv6Generator
  - IPv4Validator
  - IPv6Validator

- Questions:
  - Should the Validators belong to the Forwarding LFBs group?
LFB Design Logics - Forwarding LFBs (1)

- The following forwarding LFBs are defined:
  - IPv4UcastLPM
  - IPv4NextHopApplicator
  - IPv6UcastLPM
  - IPv6UcastNexthopApplicator
LFB Design Logics - Forwarding LFBs (2)

• Accordingly LFB components defined are
  – Fib
  – PrefixTable
  – NextHopTable
  – LocalIpAddrTable

• Actually two FIB modes applied: integrate mode and diverse mode.
  – Integrate mode FIB: All forwarding information condensed into one FIB entry. (Linux kernel forwarding & Click modular router.)
  – Diverse mode FIB: forwarding information is separated into two tables: Prefix table and Nexthop table, and an index is used to link these two tables. (Intel NP)

• Questions&Comments:
  – Joel: Unless there is a difference in actual functionality, we should not represent the same thing in two different fashions. That makes the job of implementing a CE harder, because it will have to deal with both constructs.
LFB Design Logics - Queue and scheduler LFBs (1)

- Initial work only
  - Scheduler LFB
    - defined an Input Port group called Watchers to get the queues state they watch, and an Output Port group called Controllers to output control command for the queues
  - Queue LFB
    - Have a packet input, a packet output, a control input, and a group of control outputs.
    - The control ports represent the control relationships with schedulers
  - WRRSched LFB

- Questions:
  - Queue and scheduler have complicated associations, is the current method valid?
LFB Design Logics - Miscellaneous LFBs (1)

- Classifier LFBs
  - ArbitraryClassifierLFB
    - A classifier which can test packet or metadata, and on that basis set meta-data a pick an output port.
    - Components
      - ClassifierTable
  - MetadataClassifier
    - Classify packets according to the meta data. Now only works on one meta data
    - Components
      - MetadataClassifyTable
LFB Design Logics - Miscellaneous LFBs (2)

- **Packet processing LFBs**
  - OptionProc
    - Processes on the IPv4 router-alert option
  - ExtendHeaderProc
    - Processes IPv6 packets with extended header
  - PacketTrimmer
    - Removes data from the front of a packet.
  - Duplicator
    - Duplicates packets

- **For CE-FE connection**
  - RedirectLFB
    - Represents a point of exchange of data packets between CE and FE. Packets with meta-data are exchanged.
LFB use case - IPv4 Router
Thanks!