



---

# Update of ForCES LFB Library Draft

**<draft-ietf-forces-lfb-lib-08 >**

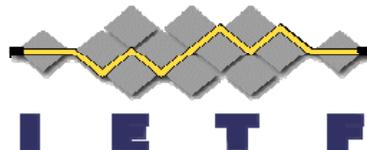
## Authors

Weiming Wang , [wmwang@zjgsu.edu.cn](mailto:wmwang@zjgsu.edu.cn)  
Evangelos Haleplidis , [ehalep@ece.upatras.gr](mailto:ehalep@ece.upatras.gr)  
Kentaro Ogawa, [ogawa.kentaro@lab.ntt.co.jp](mailto:ogawa.kentaro@lab.ntt.co.jp)  
Chuanhuang Li, [chuanhuang\\_li@zjgsu.edu.cn](mailto:chuanhuang_li@zjgsu.edu.cn)  
Halpern Joel, [joel.halpern@ericsson.com](mailto:joel.halpern@ericsson.com)

## Contributors

Jamal Hadi Salim, [hadi@mojatatu.com](mailto:hadi@mojatatu.com)  
Ligang Dong, [donglg@zjgsu.edu.cn](mailto:donglg@zjgsu.edu.cn)  
Fenggen Jia, [jfg@mail.ndsc.com.cn](mailto:jfg@mail.ndsc.com.cn)

IETF 83<sup>th</sup> Meeting  
Paris, March 25-30, 2012





## Draft Status

---

- Current Version: 08, updated February 29, 2012
  - Version 07 -- an interim version
  - Major updates since Version 06
    - Modified some definitions
    - Resolved a few issues
  
- WG Last Call
  - Version 08
  - Initiated by WG Chair on March 5
  - Review comments received



## Updates since Version 06 (1)

---

- Modified some definitions
  - LANSpeedType
    - added the special value 'LAN\_SPEED\_NONE' for not having link.
  - In the definition of some data types, added 'Reserved' field for ease of bits management to implementers:
    - VlanInputTableEntryType
    - IPv4PrefixInfoType
    - IPv6PrefixInfoType
    - EncapTableEntryType



## Updates since Version 06 (2)

---

- Resolved a few issues:
  - Reviewed and modified all ‘synopsis’ for the XML of Base Type Lib and LFB Lib.
  - Modified the type name:
    - PortStatusValues -> PortStatusType
  - Added the output port ‘ExceptionOut’ in EtherClassifier and BasicMetadataDispatch LFBs.



## Review Comments During WG LC

---

- Special thanks to Edward Crabbe for valuable reviews
- Some editorial, others technical
- Have resolved some, a few pending



## Technical issues (1)

---

- **(Pending)** Section 5 (on LFB class description):
  - Also note that, as a default provision of [RFC5812], in FE model, all metadata produced by upstream LFBs will pass through all downstream LFBs by default without being specified by input port or output port. Only those metadata that will be used (consumed) by an LFB will be explicitly marked in input of the LFB as expected metadata. For instance, ... PHYPortID ....

**Q/S:** if it's not used it's effectively invisible, and implementations in a HAL shouldn't pass this through. Given that you effectively have a mcast model for metadata at this point, I don't understand the point of making this a bus.

**WRe:** We have a quite discussion on this issue. The problem is if we only let explicit metadata passed through an LFB, downstream LFBs become hard to access any metadata upstream of upstream LFBs generated while upstream LFB does not allow them to pass.

**Need more explanation from authors**



## Technical issues (2)

---

- (Pending) Section 5.1.2.3(on EtherMACIn capability):
  - Q/S : isn't the ability to send pause frames a capability? if you're supporting standard 802.3 then this is a negotiated option. if this is not what you mean by flow control, then there should be more of a definition of what is meant.

**WRe:** : We currently define a flow control as a configurable element. A pause frame sending has been considered as one of the method of flow control implementation.

We may don't have to specifically define a pause sending as a capability if we have defined a flow control capability for the EtherMACIn LFB.

To conclude, I think we may need to consider to add a "FlowControl" capability for the LFB.



## Technical issues (3)

---

- **(Pending)** Section 5.1.3.1 (on Ethernet Classifier LFB):
  - **Q/S** : ALSO how are multiple logical ports handled? What if we have qinq or ethernet LAG being used as a trunk? It looks like only a single LogicalPortID can be carried in metadata currently which is problematic. This is very limiting as far as reuse of any of these primitives goes.  
also effectively mandating use of VlanInputTable is kludgy and will diminish reuse of the primitive.
  - **Chuanhuang Re:** The “IncomingPortID” is a global one-dimensional port number. When MAC-in-MAC, "IncomingPortID" is the logical inner MAC port. "PHYPortID" and "IncomingPortID" use non-overlapping portions of the same number space.
  - 
  - **Need more discussions to resolve this issue.**



## Technical issues (4)

---

- **(Pending)** Section 5.1.4.1 (on EtherEncap LFB):
  - The upstream LFB may be programmed by the CE to pass along a MediaEncapInfoIndex that does not exist in the EncapTable. That is to allow for resolution of the L2 headers, if needed, to be made at the L2 encapsulation level in this case (Ethernet) via ARP, or ND (or other methods depending on the link layer technology) when a table miss occurs.

**Q/S :** seems to imply a coupling of the returned, albeit unknown, MediaEncapInfoIndex to the future returned state. IF this is the intention, constraining the model in a way that dictates impl. is probably not a good idea. if this is not the case, then why not just return a type indicating l2 entry and leave it at that. not such a great way of doing this.

**Jamal Re:** There are multiple ways to implement; eg in addition to passing the detail to the CE, just drop the packet or send it to the CE or queue it on FE waiting for a resolution etc. Do we need text to state this? Note: I argued for that text because initially ARP/ND was built into the model.

Need a few more discussion to form the text.



## Technical issues (5)

- (Pending) Section 5.1.4.1:

- For neighbor L2 header resolution (table miss exception), the processing LFB may pass this packet to the CE via the redirect LFB or FE software or another LFB instance for further resolution. In such a case the metadata NextHopIPv4Addr or NextHopIPv6Addr generated by next hop LFB is also passed to the exception handling. Such an IP address could be used to do activities such as ARP or ND by the handler it is passed to.

**Q/S :** what about flooding behavior in bridged environments? as written this doesn't seem extensible unless the intent is to send all L2 packets w/o an entry to CE.

**WRe:** Not sure if I followed what you mean well, but what I understand is that all packets with the L2 resolution requirement output from the exception port of the LFB don't have to be delivered to CE all. A classification LFB based on the different exception cases (by exception ID) may follow the output to demux the packets.

**Jamal Re:** We would need to define the behavior on L2 in that perspective when for example some LFB related to bridging was added . At the moment we are dealing with L3 only. [The necessary knobs at L2 would apply: Flood on the broadcast domain, Send the packet to the FE software, Send the packet to the CE etc. Out of scope: but you'd also need knobs for learning the addresses etc (as options since learning could also happen in s/ware at the CE or FE).]

waiting more discussions



## Technical issues (6)

---

- (Resolved?) Section 5.3 (on IP forwarding LFBs):
  - Q/S : probably should be reworded - just need to point out decoupling of model from FE. hw again:

However, there actually exists other models, like one which may only have a forwarding information base that have conjoined next hop information together with forwarding information. In this case, if ForCES technology is to be applied, some translation work will have to be done in the FE to translate attributes defined by this document into attributes related to the implementation.
  - Jamal Re: There's some acrobatics involved in the wording, but the intent is to not have someone implementations feel their approach is to be left out. We optimized for the common approach.
  - Need to form an updated text ?



## Editorial Issues (1)

---

- Section 2:
  - **Question/Suggestion:** mixing of 'class' and 'type' confusing - why not just use class
  - Weiming Re:** the name of LFB class or type were used in previous document like RFC5812, so it was just inherited.
- Section 3.1:
  - **Q/S:** no mention of stats - implicit in (6) but should be called out
  - WRe:** good catch. Plan to update as below.
    - (6) Provide network management and system support facilities, including loading, debugging, status reporting, exception reporting and control, **statistics query**.



## Editorial Issues (2)

---

- Section 3.2.1:
  - o When flexibility is not required, an LFB should take advantage of its independence as much as possible and have minimal coupling with other LFBs. The coupling may be from LFB attributes definitions as well as physical implementations.  
**Q/S:** this should be the case whether or not flexibility is required  
**WRe:** agreed. Plan to remove "When flexibility is not required, "
  - o Unless there is a clear difference in functionality, similar packet processing should not be represented as two or more different LFBs. Or else, it may add extra burden on implementation to achieve interoperability.  
**Q/S:** poorly phrased - you're doing a typical appeal to parsimony but this wording complicates interpretation  
**WRe:** plan to modify it.



## Editorial Issues (3)

---

- Section 4.2:
  - **Q/S** : packet type != frame type
  - **WRe**: to change ‘packet type’ to ‘packet frame type’
- Section 4.3:
  - **Q/S** : seems like 'metadata ID' should be 'metadata type'
  - **WRe**: An element <metadtaID> instead of a ‘<metadataType>’ has been defined in FE model RFC5812.
- Section 5:
  - In this section, the terms "upstream LFB" and "downstream LFB" are used. These are used relative to an LFB to an LFB that is being .....
  - Q/S** : reword - this is unnecessarily convoluted
  - WRe**: try to reword it.



## Editorial issues (4)

---

- Section 5.1 on Ethernet processing LFBs:
  - **Q/S** : then what Ethernet is supported? state explicitly  
**W Re:** Generally speaking, all types may be supported. For individual LFBs, detailed supported types are defined by the input or output frame types of the individual Ethernet LFBs. Note that if all types of Ethernet are allowed, EthernetAll type is applied. When an LFB have to generate a type of Ethernet packet, it is uniformly defined to generate Ethernet II type.



## Editorial issues (5)

---

- Section 5.1.3.1:
  - **Q/S** : use of the term 'decapsulation' here is a bit of an overstatement. you may not actually be decapsulating the packet. you *\*are\** demultiplexing the packet  
**WRe:** decapsulating is included in the LFB, i.e., output of the demuxed packets are network layer packets instead of Ethernet packets.
  
- Section 5.1.4.1:
  - **Q/S** : use of the term 'packet' in terms of the ForCES packet processing model is inaccurate  
**WRe:** can not catch up with it well. Could it be explained more?



## Next Work

---

- Resolve pending issues ASAP
- Update a new version accordingly
- Move forward according to IETF publication process



**Thanks!**