Forces Protocol

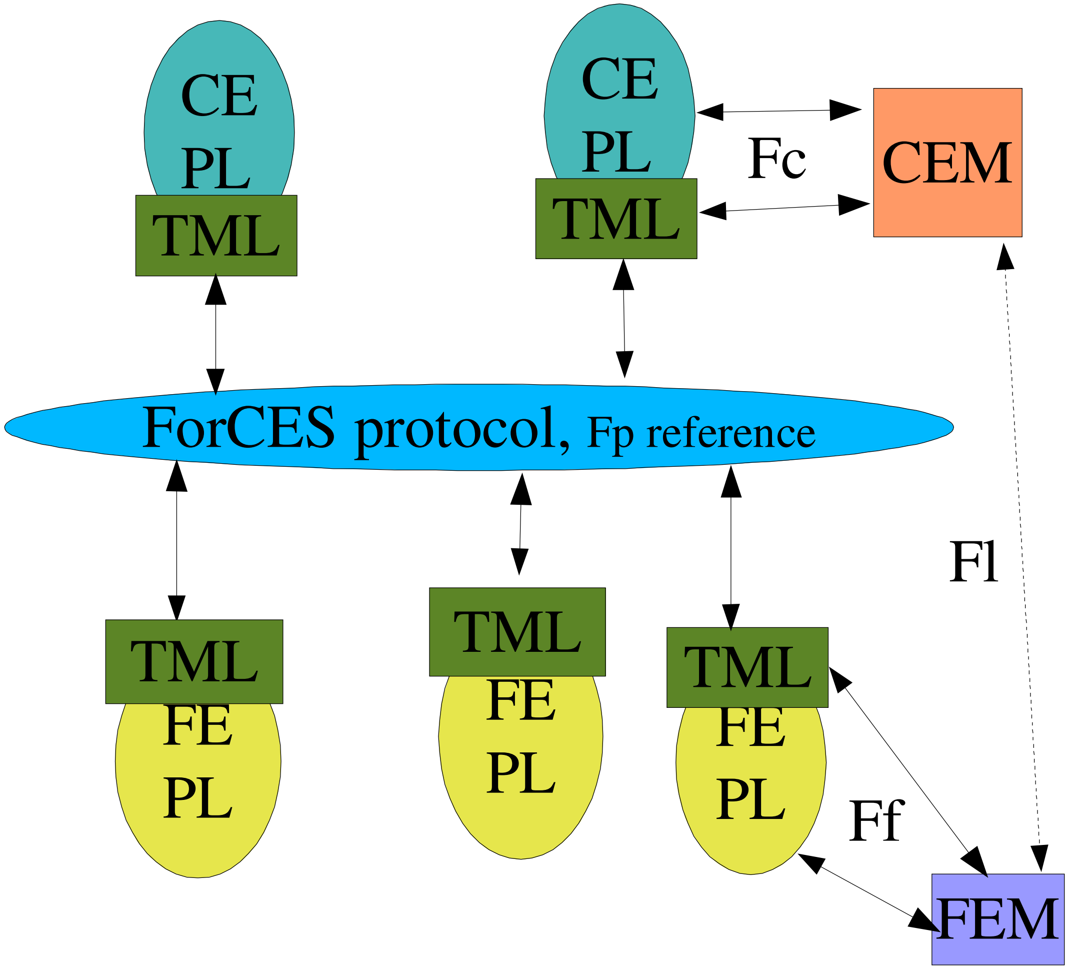
Jamal Hadi Salim
(for the ForCES Protocol Team)

IETF61, Washington, DC
November 2004

Team Members (alphabetical order)

- Ligang Dong, Zhejiang Gongshang University
- Avri Doria, ETRI
- Ram Gopal, Nokia
- Robert Haas, IBM
- Jamal Hadi Salim, ZNYX Networks
- Hormuzd M Khosravi, Intel
- Weiming Wang, Zhejiang Gongshang University

ForCES overview



Outside of ForCES charter:

- CE-CE communication
- FE-FE communication
- CEM-FEM
- CEM
- FEM

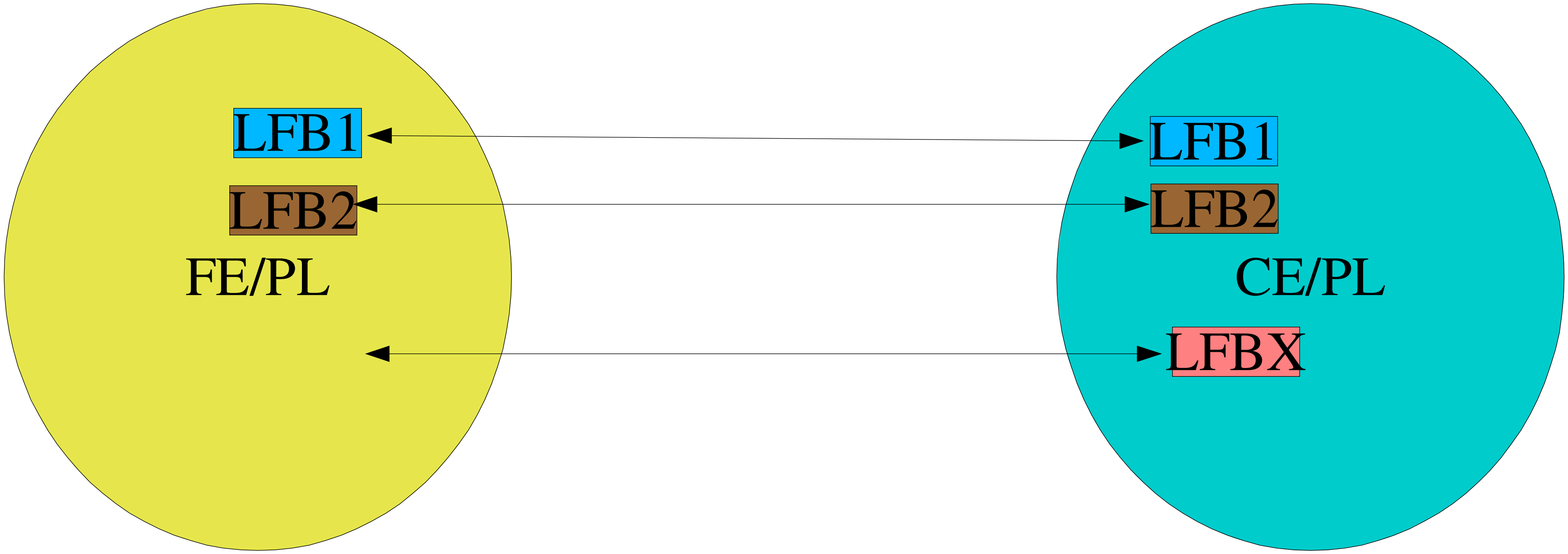
TML

- Resolves underlying Transport issues
  - Reliability
  - Security
  - Congestion control
  - Address Mapping
  - Timeliness
  - Prioritization
  - Encapsulation
  - High Availability facilitator

TML

- Several proposals made to date
  - TIPC: L2/L1
    - Draft-maloy-tipc-01
      - Unicast, multicast, broadcast
  - draft-audu-forces-iptml-00
    - IP level
  - Draft-khosravi-forces-tcptml-00
    - IP Level

PL-PL Logical Interaction



- PL Termination typically at LFB
  - Simplifies grammar and architecture
  - There are a few exceptions such as a heartbeat and association messages
    - Still needs to be discussed

Core LFBs

- Needed for termination of messages addressed to the FE or CE
- Three LFBs at the moment
  - FE Object LFB
    - Maintains attributes relavant to the FE
  - FE Protocol Object LFB
    - Maintains attributes relevant to the ForCES protocol
  - CE Object LFB
    - Maintains attributes relevant to the CE

Core LFBs: FE Protocol Object LFB

- Assigned class ID 1 and instance Id 1
- Responsible for:
  - Protocol events that can be subscribed to
    - Heartbeats, TML events, etc
  - Protocol capabilities
    - Version, TML capability
  - Protocol attributes
    - Association timer, hearbeat interval, Primary CE, alternate
    - Alternate CEs, failover policy, restart policy, etc

Core LFBs: FE Object LFB

- Assigned class ID 2 and instance Id 1
- Responsible for
  - FE Events management
    - FE Status changes, DOS alerts, Capability changes
  - FE attributes
    - FE status, Instantiated LFBs and topologies, FE Model etc
  - FE Capabilities
    - Supported LFBs by FE, their occurance limits, etc

Core LFBs: CE Object LFB

- Assigned class ID 3 and instance ID 1
- Responsible for
  - CE Events management
    - CE Status changes, Capability changes
- This LFB is still under discussion
- Maybe even a CE Protocol LFB maybe needed to reciprocate FE protocol Object

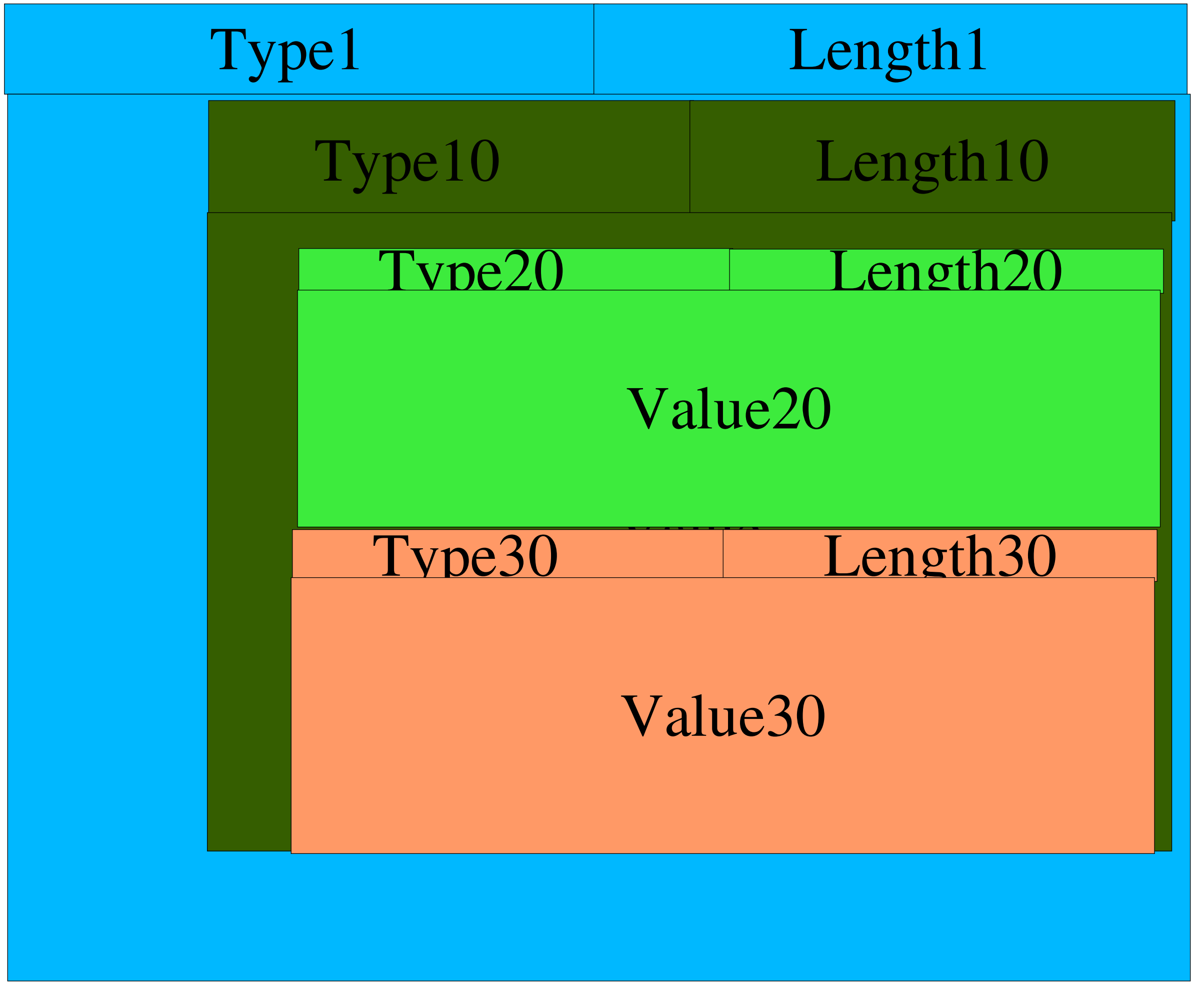
Protocol Grammar

- PL level PDU := PLHdr<*LFBSelect*>+
- LFBSelect := LFBClassID LFBInstance <*OPER*>+
- OPER := <OPERATION [<*path-data*>]*>+

Protocol Grammar

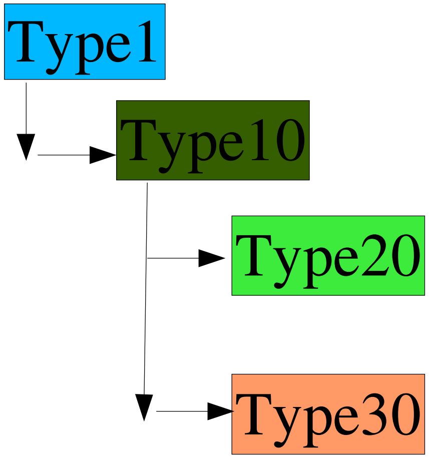
- PLHdr defines message type and CE/FE Ids, etc
- LFBClassID is a 32 bit unique identifier given at LFB definition time
- LFBInstance is a 32 bit ID for an instance of LFB class creates at runtime
- Operation is one of *ADD*, *DEL*, *GET*, *ADVERTISE*, *CANCEL*, etc
- *Path-data* is what the operation is applied on
  - Could be attribute, capability, event

Nested TLVs



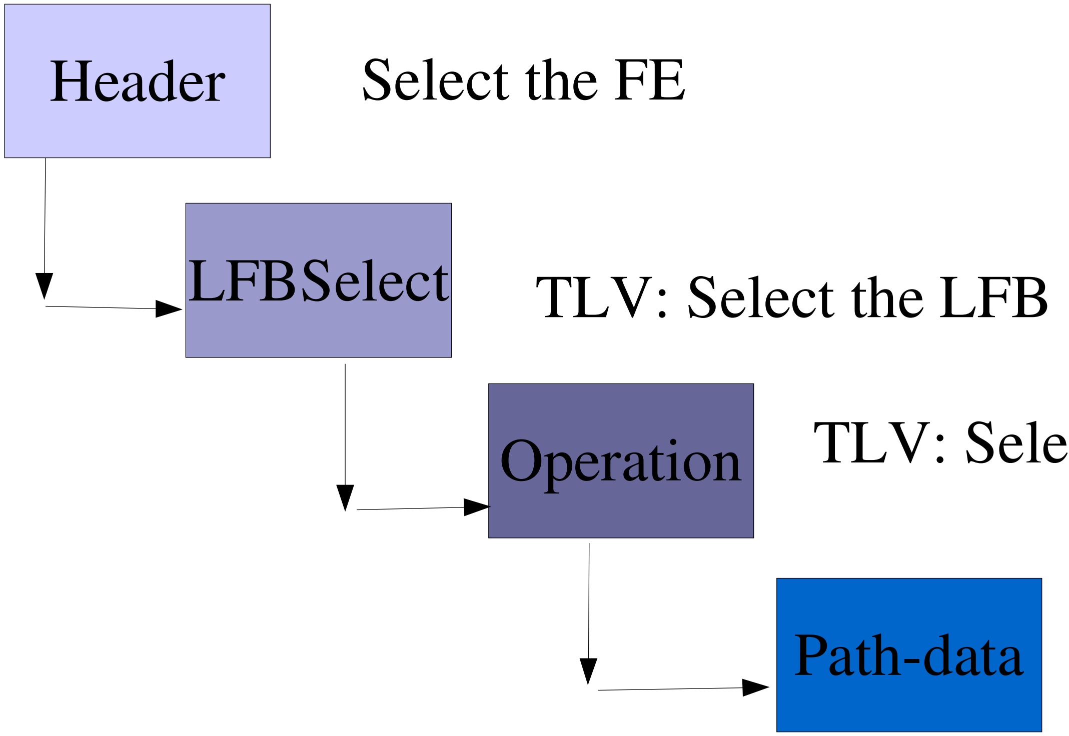
Type1
Length1
Type10
Length10
Type20
Length20
Value20
Type30
Length30
Value30

- Very extensible
  - Fits very nicely in BNF description of PL protocol
  - Maps nicely to xml definitions/layering



Type1
Type10
Type20
Type30

PL Message Layout



Header

Select the FE

LFBSelect

TLV: Select the LFB Class and instance

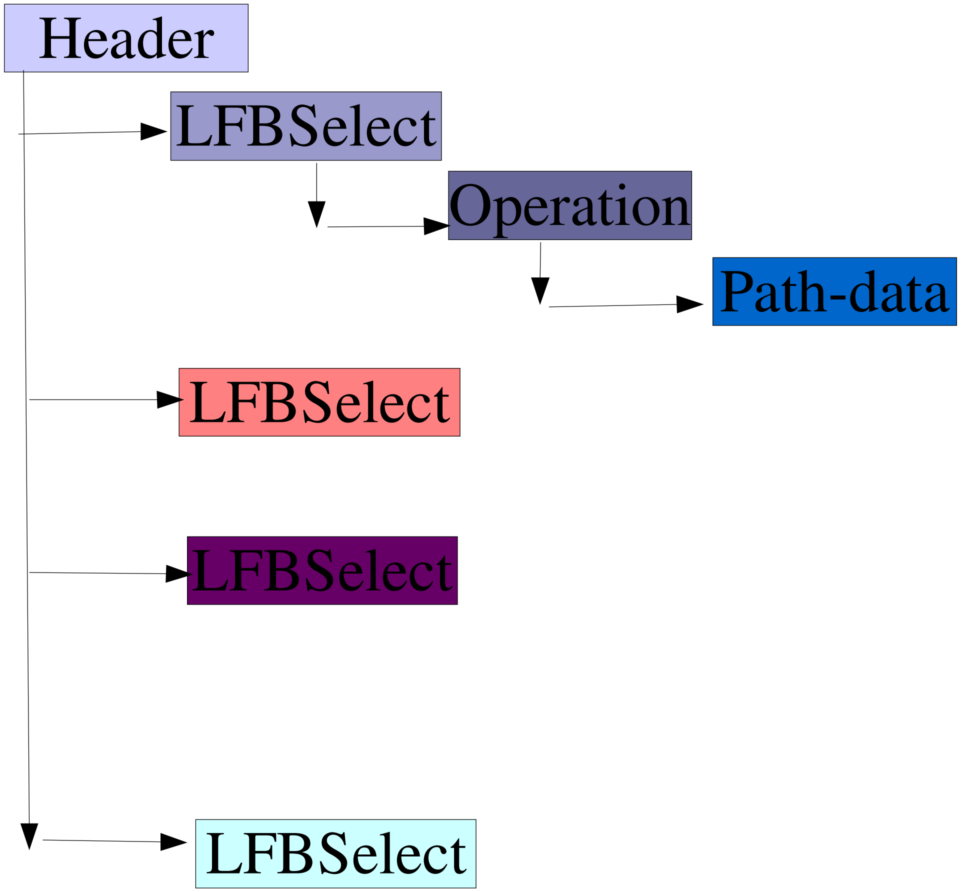
Operation

TLV: Select the operation

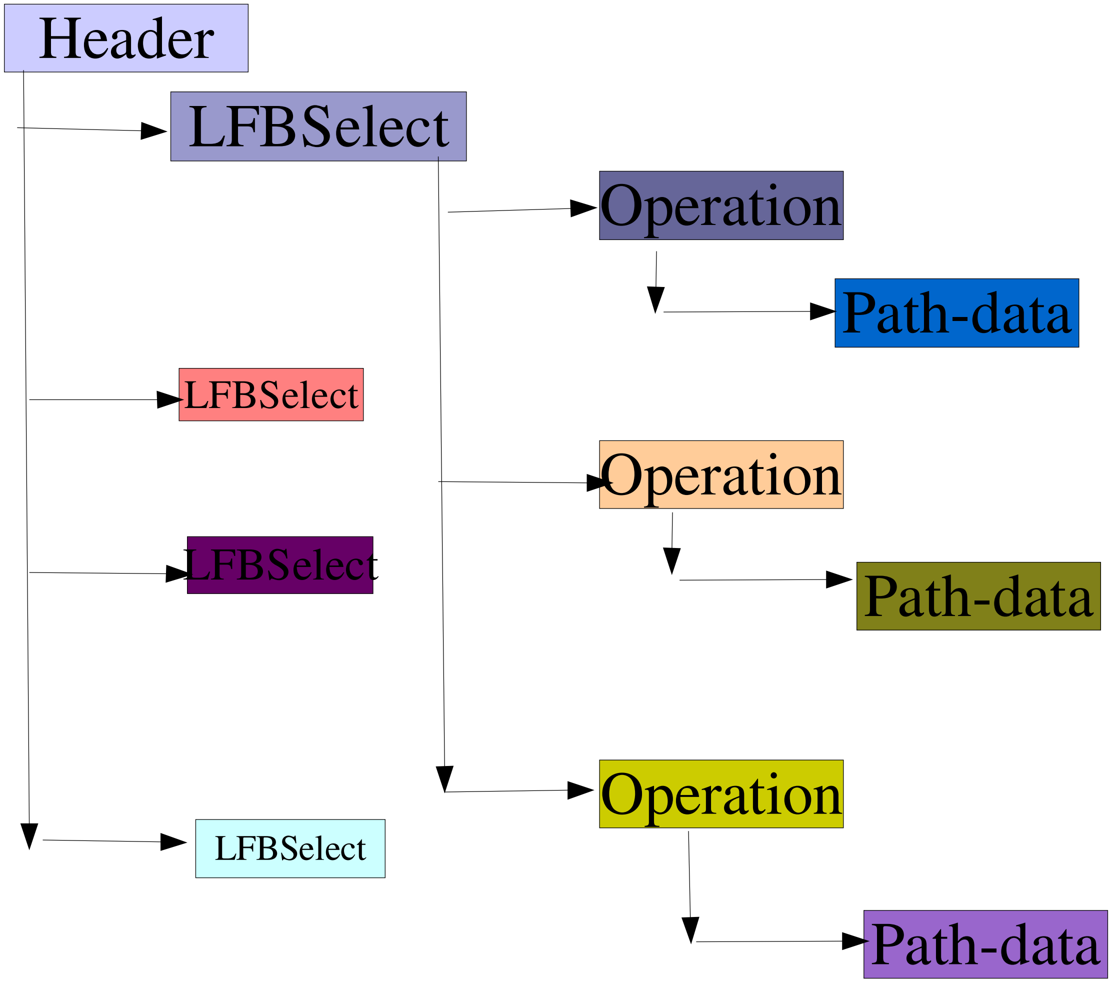
Path-data

TLV: The operand(s)
- path to entity
- data used

Batching: Multiple LFB selections



Batching: Multiple operations



Header

LFBSelect

Operation

Path-data

LFBSelect

Operation

LFBSelect

Path-data

Operation

LFBSelect

Path-data

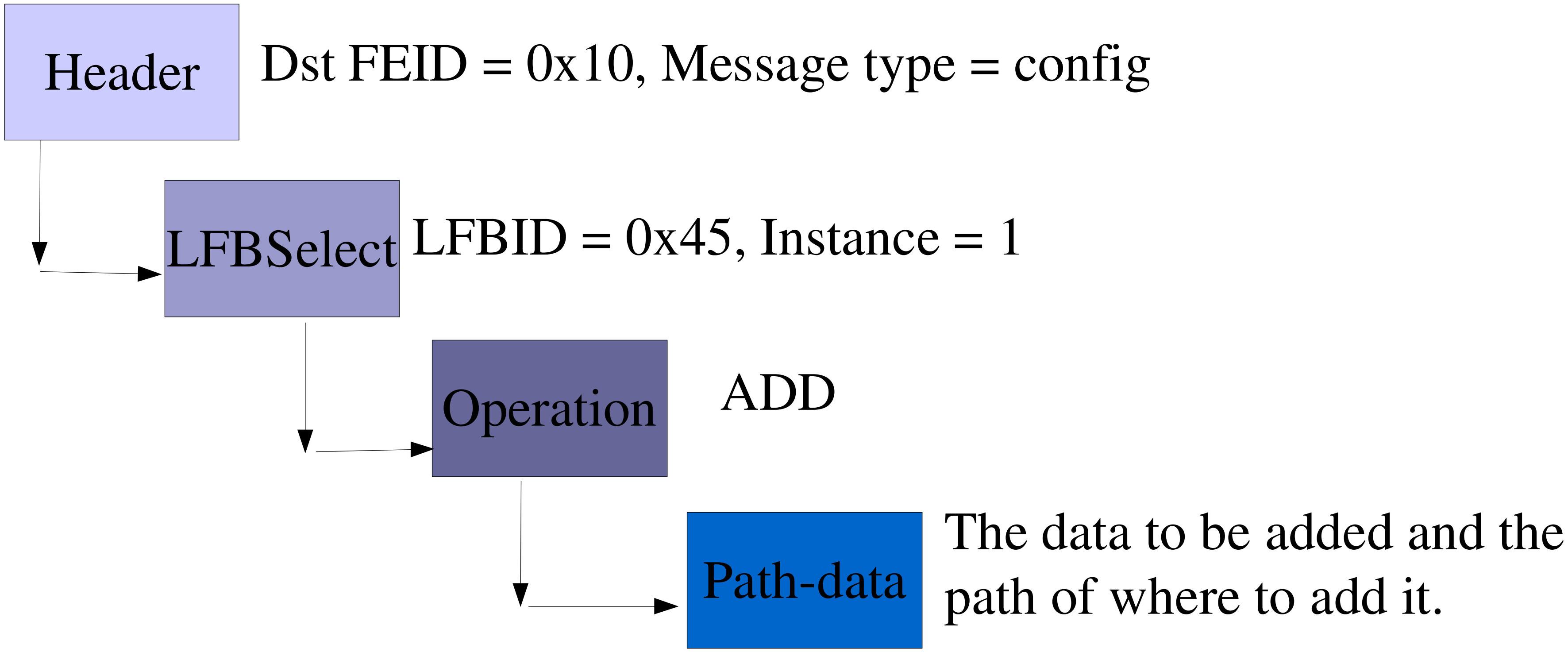
PL Message Types

- Association
  - To join and leave NE (by FE)
- Configuration
  - To configure attributes, capabilities and events in an LFB instance
- Query
  - To query for configured attributes, capabilities and events

PL Message Types

- Event Notification
  - Used to notify about events to registered users
- Packet redirection
  - Packet redirects from FE to registered LFBs in CE
- Heartbeat
  - Heartbeat between CE and FE

An Example PL Message Layout



Header

Dst FEID = 0x10, Message type = config

LFBSelect

LFBID = 0x45, Instance = 1

Operation

ADD

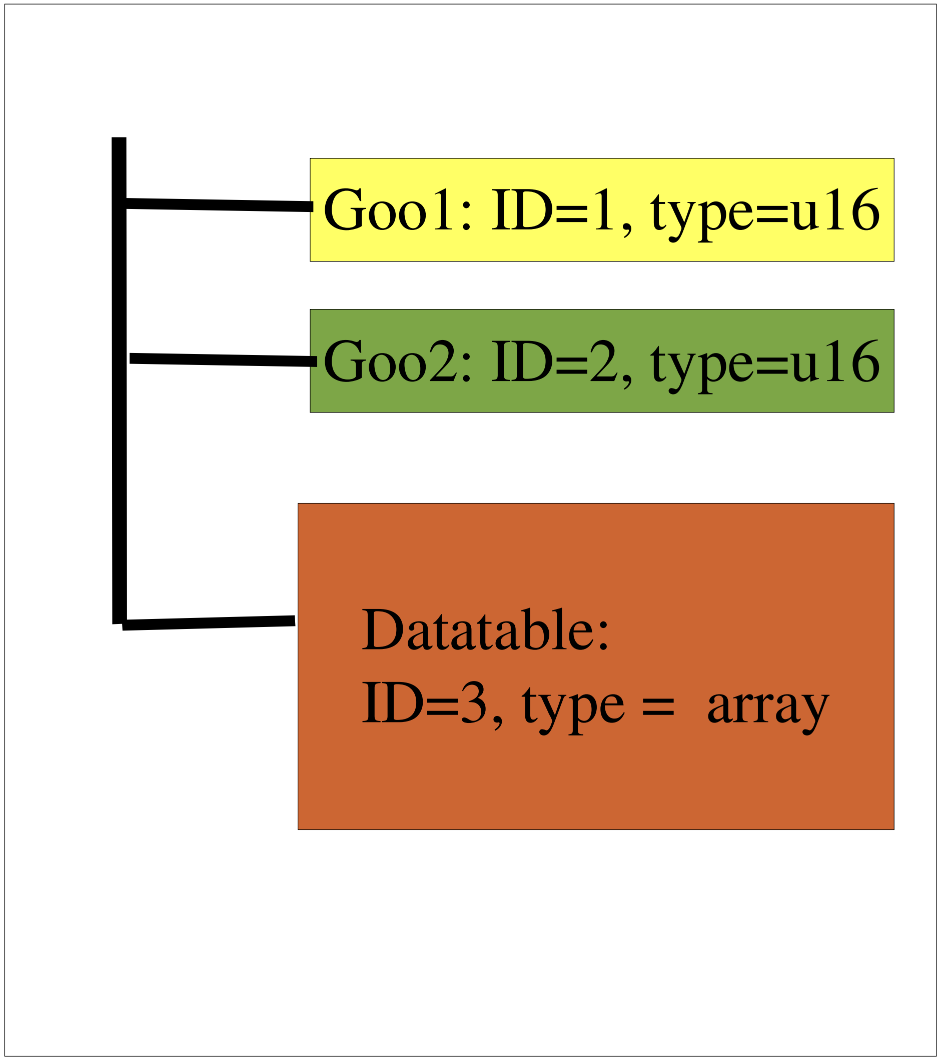
Path-data

The data to be added and the path of where to add it.

Path Data

- A path is a *map* to targeted element/entity
  - Path is a series of 32 bit Identifiers
    - XML definition of LFB requires all elements that can be targeted in a path to have a name and a 32 bit id.
    - A path is not unlike a SNMP OID
  - Element targeted maybe capability, attribute, or event(under discussion)
- Carries associated data where needed
  - Example: Config will have data, but not query.

Simple Example: Attributes for NOP LFB



Goo1: ID=1, type=u16

Goo2: ID=2, type=u16

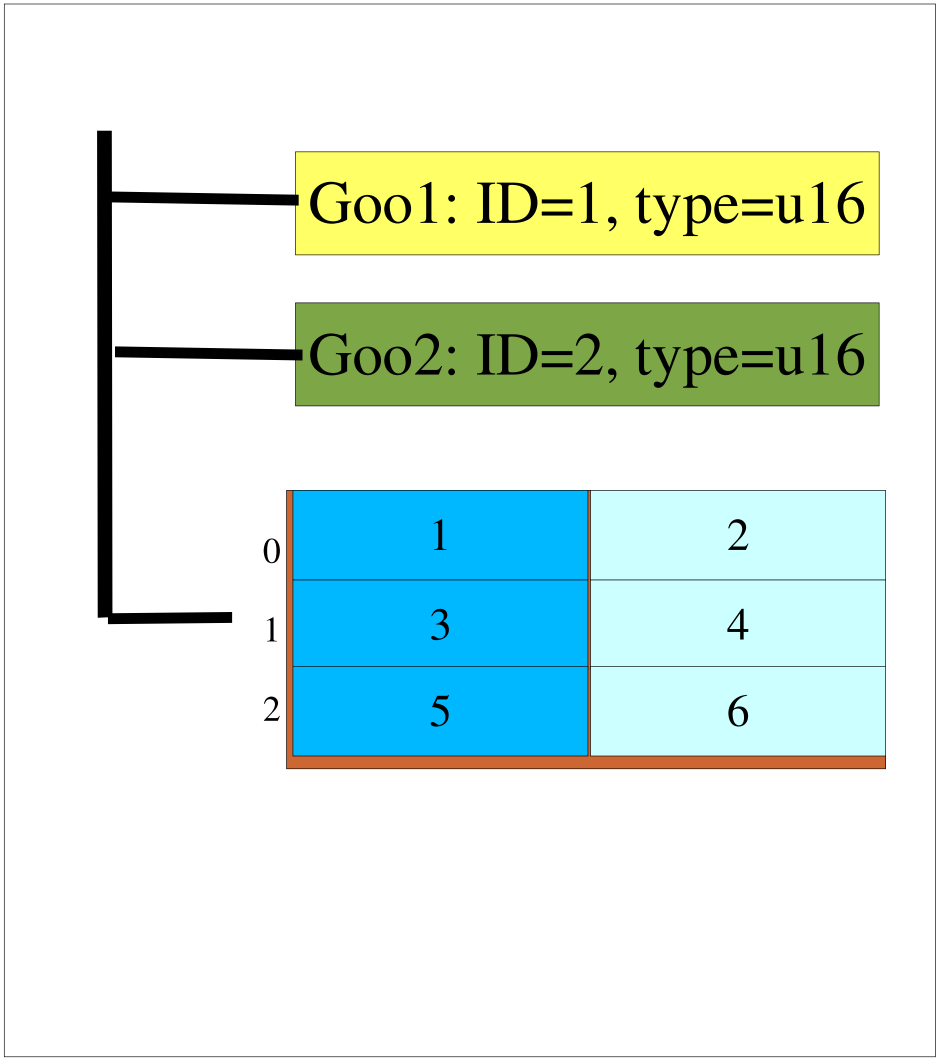
Datatable:
ID=3, type = array

Datatable row constitutes

Foo1: ID=1, type u32	Foo2: ID=2, type u32

- Table entries are created at runtime
- Table entries of arrays have index references

Simple Example: Attributes for NOP LFB



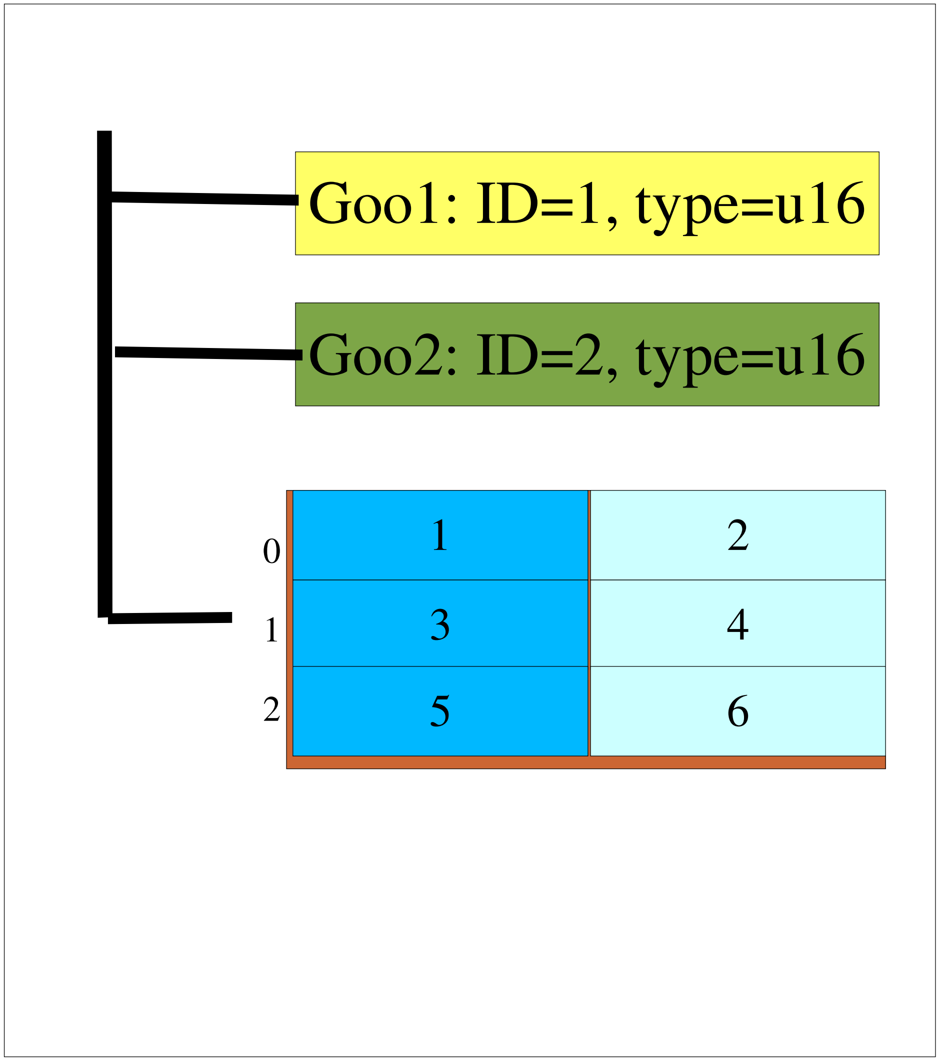
Goo1: ID=1, type=u16

Goo2: ID=2, type=u16

0	1	2
1	3	4
2	5	6

- To access goo1
  - ID = 1
- To acces goo2
  - ID=2
- To access all of datatable
  - ID = 3
- To access first row of datatable
  - ID = 3,0
- To access foo2 in second row of datatable
  - ID=3,1,2

Table operations under discussion



Goo1: ID=1, type=u16

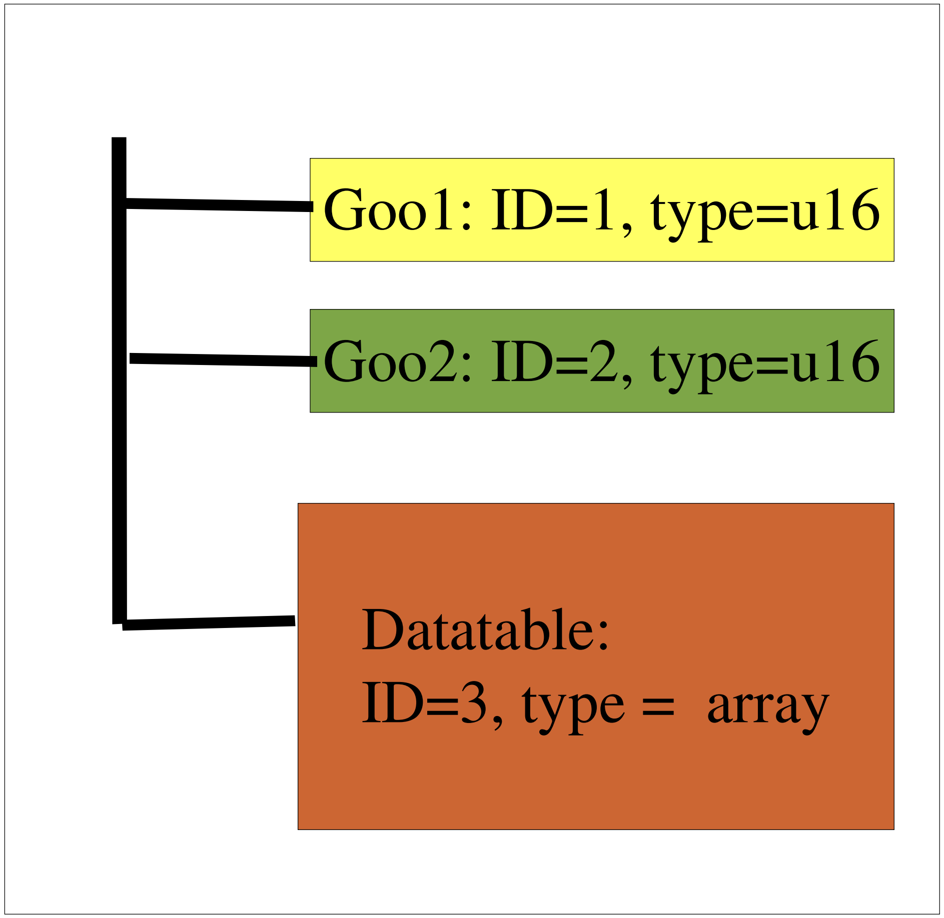
Goo2: ID=2, type=u16

0	1	2
1	3	4
2	5	6

Based on doc posted by Steve

- To access multiple rows
  - Eg: ID = 3, {startrow=1,endrow=2}
  - Eg: ID=3, { startrow=0, count=2}
- To access first column of datatable
  - Eg foo1: ID = 3, {startrow=0, endrow=-1},1
- Content based
  - Eg: occurance of all entries with foo2=1
  - ID=3, [key=foo2, value=1]

Complex Example: Attributes for NOOOP LFB



Goo1: ID=1, type=u16

Goo2: ID=2, type=u16

Datatable:
ID=3, type = array

A more complex example will have f002 have its own table. Not showing that

Datatable row constitutes

Foo1: ID=1, type u32	Foo2: ID=2, type array

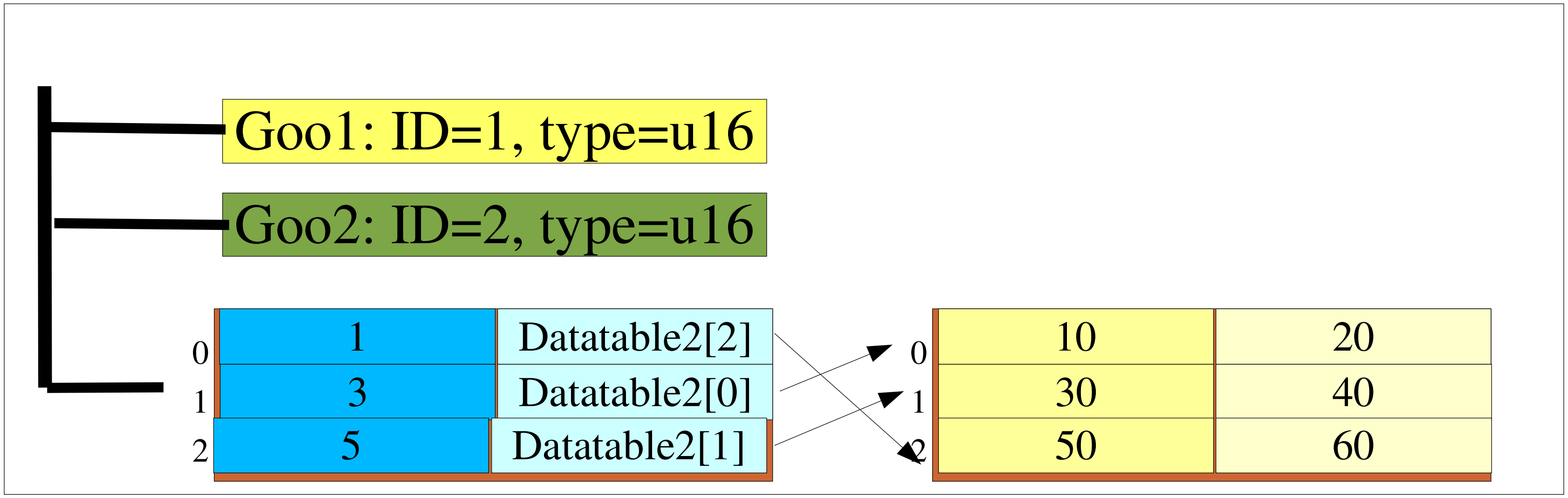
Introduce Datatable2 which has ID 4

Datatable2 row constitutes

Foo3: ID=1, type u32	Foo4: ID=2, type u32

- Note we have nested tables now
- Datatable cell Foo2 nests into datatable2
  - Example consider LPM table with nexthop table

Complex Example: Attributes for NOOOP LFB



0	1	Datatable2[2]
1	3	Datatable2[0]
2	5	Datatable2[1]

0	10	20
1	30	40
2	50	60

- To access datatable2: ID= 4
- To access datatable2, first row: ID = 4,0
- Contents of row 2 in datatable = { 3, datatable2[0])} = {3,10,20}
  - Note encoding is still under cloudy discussion
- To access foo3 of datatable2 that is pointed to by datatable[1]: 3,1,2,1
- Content access: for all occurances where datatable foo1=3, and datatable2 foo4=10
  - ID=3, [[key=foo1, key-val=3][key=foo4,key-val=10]]

Open Issues: Show stoppers

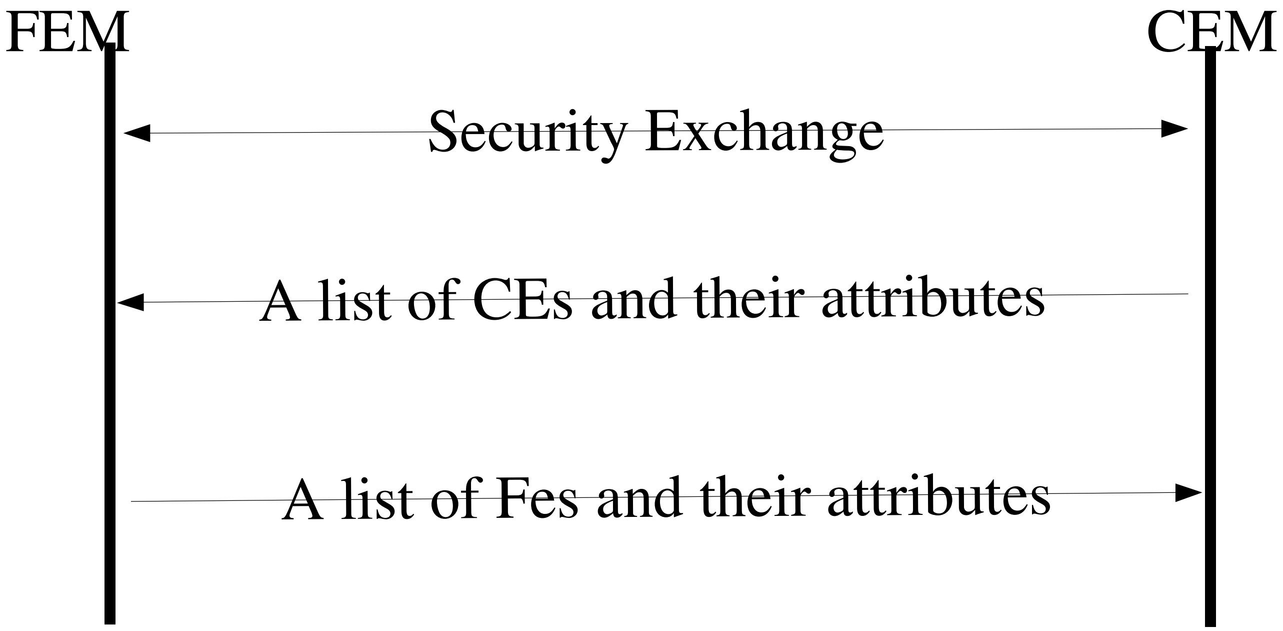
- Consensus on path
  - Two proposals in place at the moment
- Packing/transporting of path referenced data
  - A scheme proposed by Steve/Zsolt insufficient for table hierachies
  - TLV for referenced data (may not be efficient)

Open Issues: Resolvable

- Operations on block data and content based
  - Briefly discussed is access by name
- Multicasting to LFB instances (Robert will present)
- Whether to split LFBSelector into two hierachies (Robert will present)
- All messages terminate at LFB

Backtrack

Fl Reference point



FEM

CEM

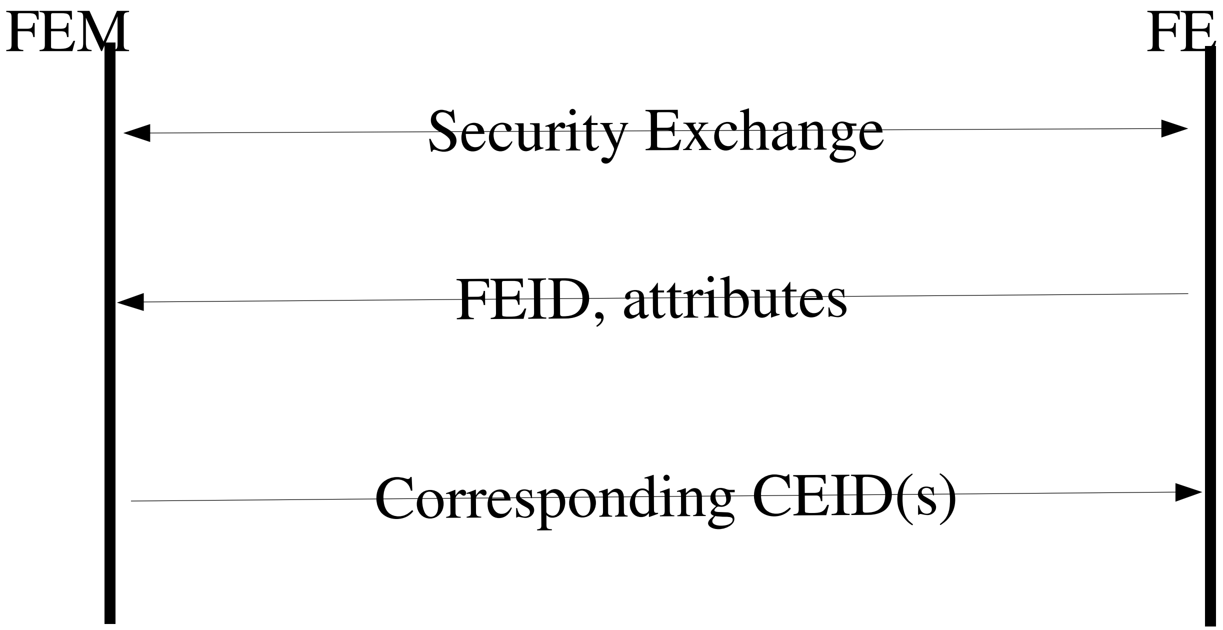
Security Exchange

A list of CEs and their attributes

A list of Fes and their attributes

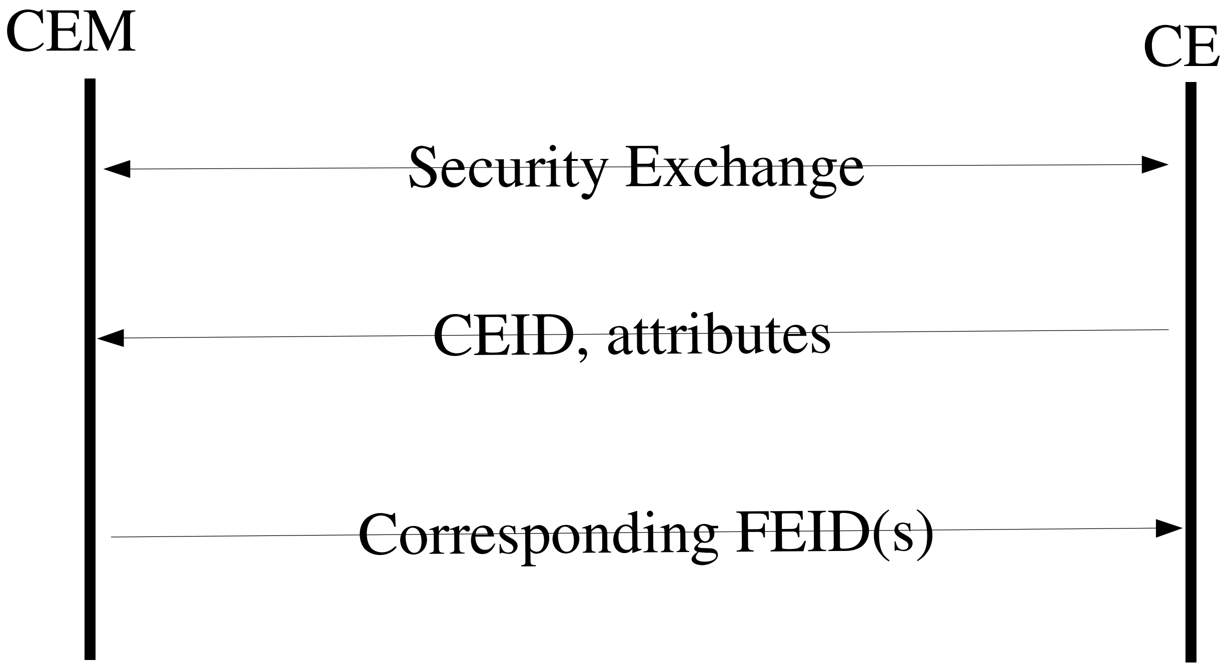
This reference point could be simple, static and based on Fes
reading config files

Ff Reference point



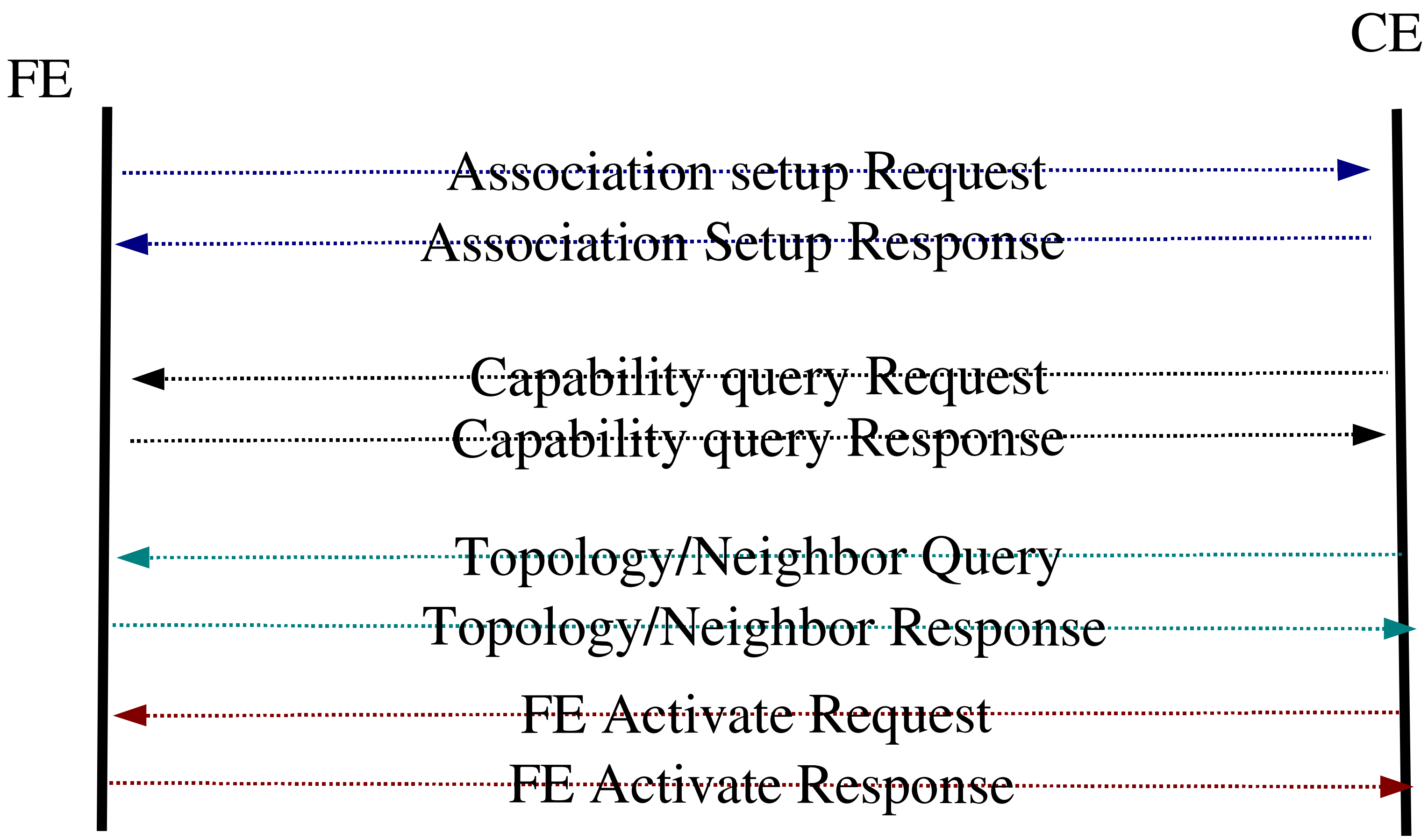
This reference point could be simple, static and based on FE reading config files

Fc Reference point

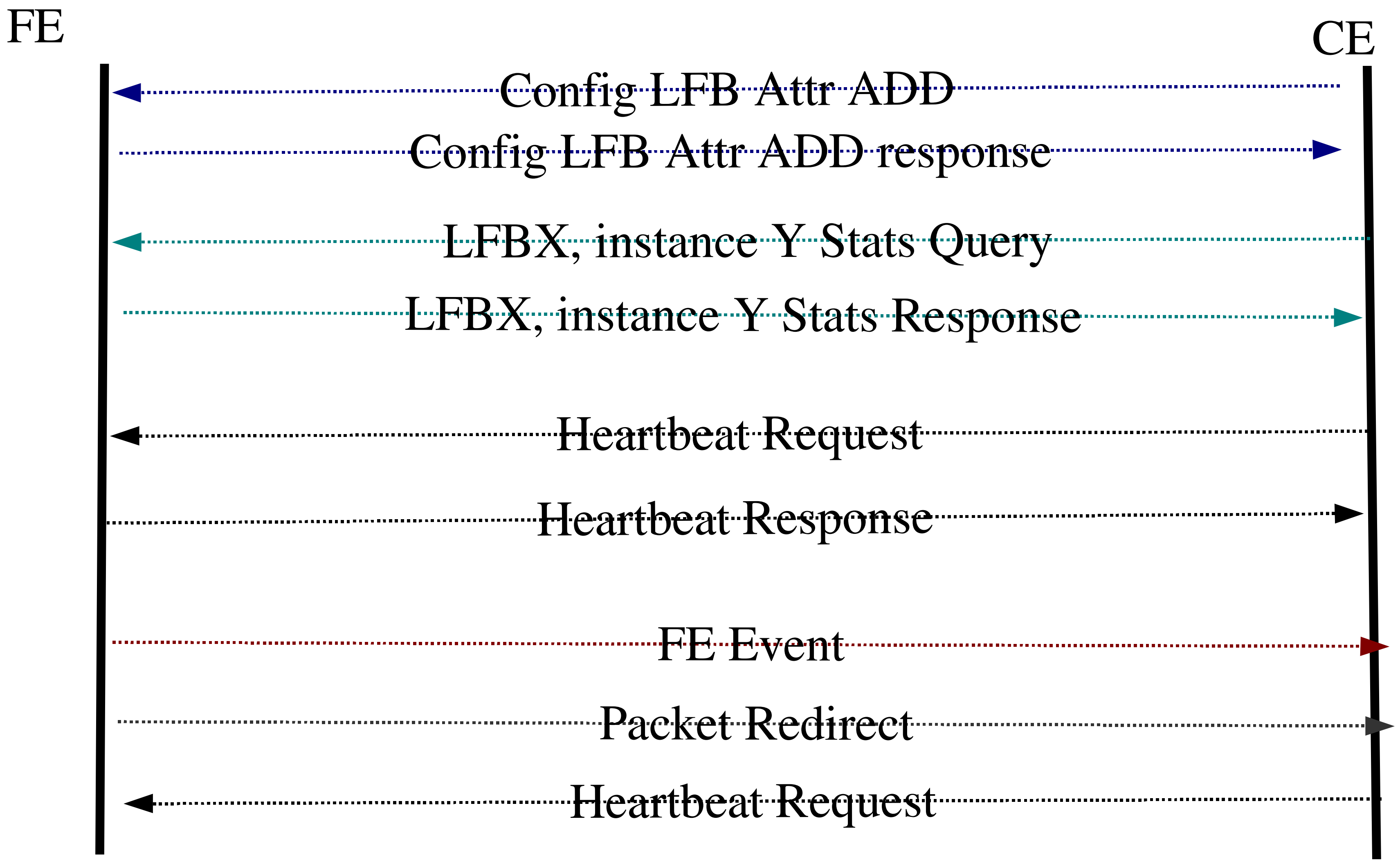


This reference point could be simple, static and based on FE reading config files

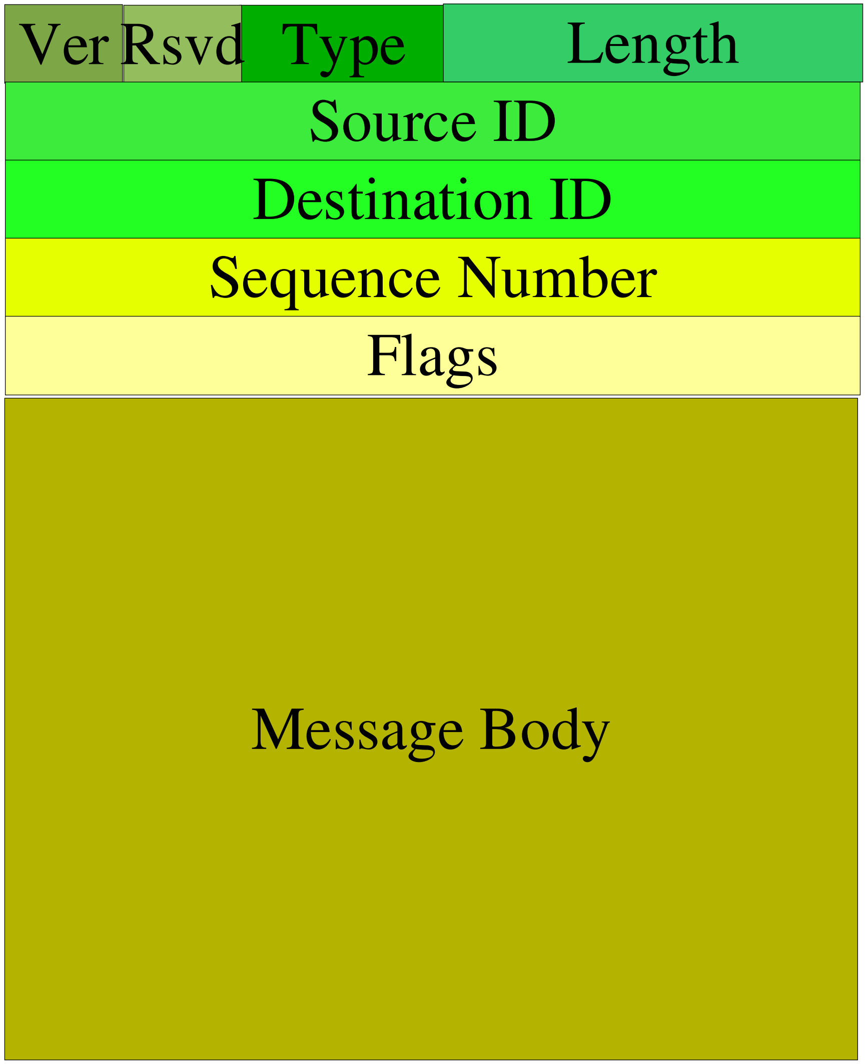
Sample Scenario: FE initialization


FE
CE
Association setup Request
Association Setup Response
Capability query Request
Capability query Response
Topology/Neighbor Query
Topology/Neighbor Response
FE Activate Request
FE Activate Response

Sample Scenario: FE established



PL Message Layout



Ver Rsvd Type Length

Source ID

Destination ID

Sequence Number

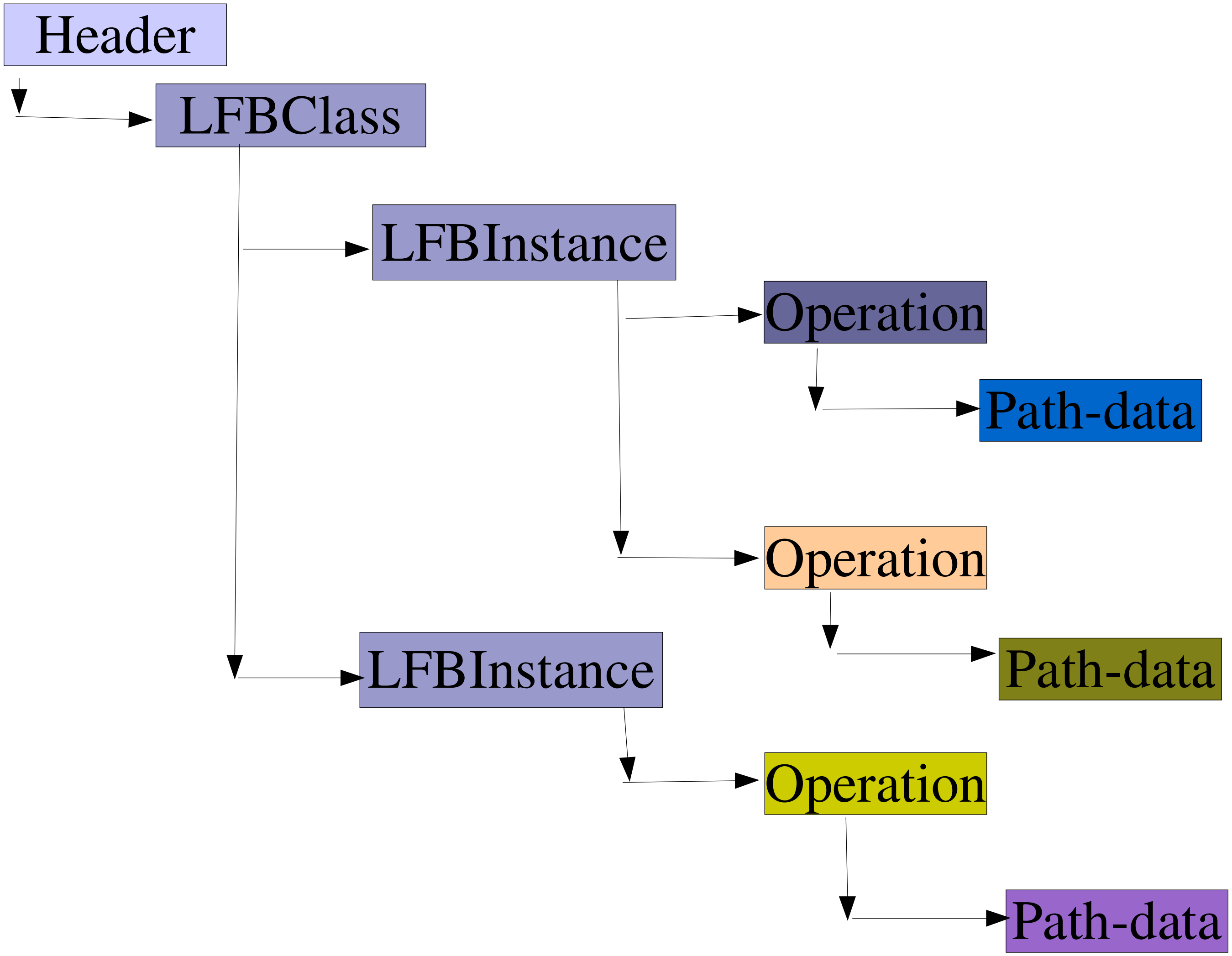
Flags

Message Body

- Type defines content of message
- Length DWORDS includes header + message body
- Source ID: The ID of source CE/FE
- Sequence Number of message
- Flags that further define the content in the body
  - Example ACK requested
- Message Body: Constitutes of multiple TLVs nested or in the same level

Source/Destination ID field breakdown

Batching: Multiple LFB Instances



Header

LFBClass

LFBInstance

Operation

Path-data

Operation

LFBInstance

Path-data

Operation

Path-data