Netlink2 as ForCES Protocol

`<draft-jhsrha-forces-netlink2-02.txt>`

Presentation available at:

`http://www.petri-meat.com/slblake/networking/drafts/netlink2.pdf`

Authors

- Jamal Hadi Salim <hadi@znyx.com>
- Robert Haas <rha@zurich.ibm.com>
- Steven Blake <steven.blake@ericsson.com>

Content

- Summary of draft changes
- Summary of Netlink2 and ForCES requirements
- Netlink2 TLVs encapsulation
- Netlink2 LFBs
- Netlink2 security
- Netlink2 reference code

Changes in -02

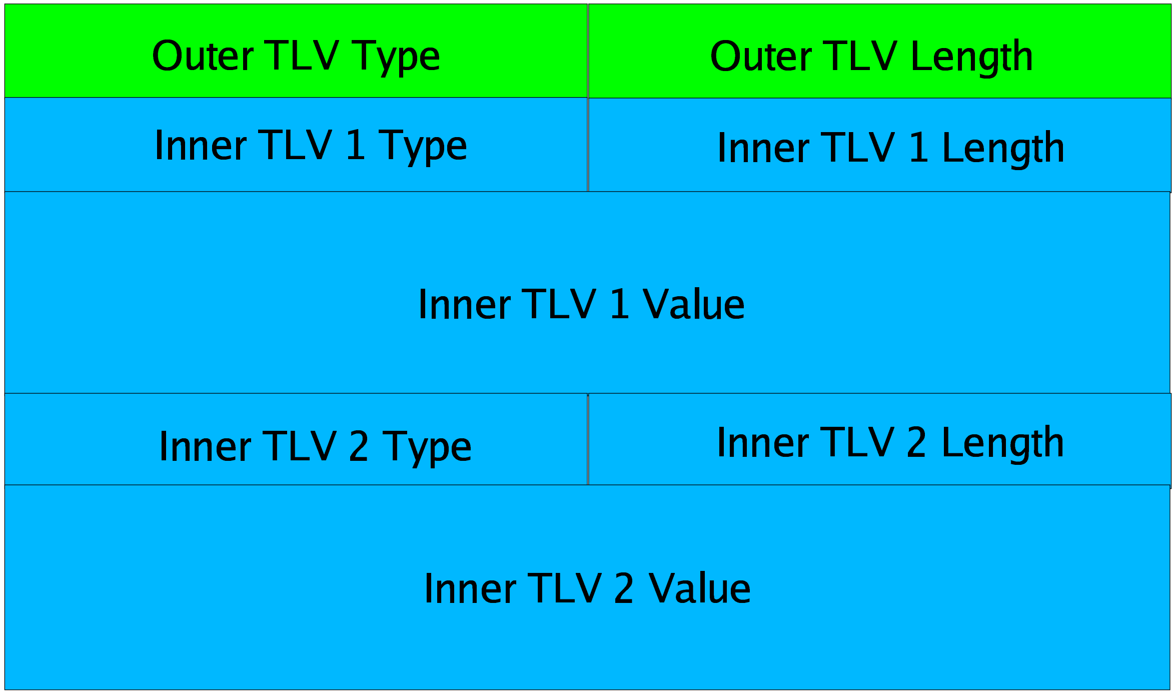
- Last version publicly presented was `<draft-jhsrha-forces-netlink2-01.txt>` in Vienna.
- Changed TLV encapsulation to permit multiple Netlink2-header extension TLVs or Service TLVs in the same Netlink2 message.
- Provide state transition diagrams for FE-CE protocol association.
- Added LFB definitions derived from Netlink:
  - RFC 3549 on "Netlink as IP Services Protocol"
- Extended security measures for local scope (single-hop) and global scope (multi-hop) environments.

Netlink2 and ForCES requirements

- Solves one of the scalability requirements by supporting multicast wires.
  - Ex/ CE multicast global routing table updates to every FE.
  - Can address all instances of a LFB class on all FEs.
- Can be layered directly on top of a link layer (e.g., Ethernet, Infiniband):
  - PID addressing
  - Checksum
  - SYN/FIN
- Netlink2 meets all of the other requirements.

TLV encapsulation

- Allow nested TLV encapsulations.



Outer TLV Type

Outer TLV Length

Inner TLV 1 Type

Inner TLV 1 Length

Inner TLV 1 Value

Inner TLV 2 Type

Inner TLV 2 Length

Inner TLV 2 Value

Netlink2 Extension TLV

- Outer TLV encapsulates one or more inner extension TLVs.
- Used to provide network-layer features if not running over IP + UDP/DCCP/TCP (e.g., raw Ethernet)
- Inner extension TLVs
  - Authentication (to be defined)
  - Checksum
  - Message priority
  - SYN cookie (to be defined)
  - Name
- Optional

Netlink2 LFBs

- Documented in draft:
  - Interface Service
  - Address Service
  - IPv4/IPv6 Forwarding LFB
  - Neighbor Discovery
- Documented elsewhere/implemented:
  - Physical port and address
  - Filtering
  - QoS
  - IPsec
  - Packet sampling
  - Packet mirroring

Security

- Threats at local scope:
  - DoS (Denial of Service) with data packets forwarded to the CE.
    - ➔FE must support policer LFB for traffic redirected to CE.
- Threats at global scope:
  - Netlink2 SYN flood attacks
    - ➔New Netlink2 SYN cookie TLV
  - Fake FEs or CEs
    - ➔Perform authentication based on qualified names.

Netlink2 reference code

- Announcement will be made on the list within the next month.
- Open source
- Demo at the end of this meeting.

Conclusions

- Key differentiators of Netlink2:
  - Based on widely used Netlink.
  - Scalability ensured with use of multicast.
- Many FE-level and LFB-level TLVs are still to be defined, preferrably in separate drafts.
  - Add ForCES-specific TLV(s)

Upcoming work items for -03

- Finish definition of extension TLVs:
  - Authentication
  - SYN cookie
- Finish LFB descriptions
- Editorial cleanups

Backup

Motivation: Why Netlink derived?

- Linux Netlink sockets proven mechanism
  - Derived from BSD routing sockets
  - Running code since Linux 2.1.x
  - Issues related to ForCES addressed over the years from operational experiences
    - User Space (CE) to Kernel (FE) communication
- Many existing services using Netlink
  - IP v4 and v6 forwarding (unicast, multicast, policy routing)
  - Classification, QoS, Packet redirection, IPSec, etc

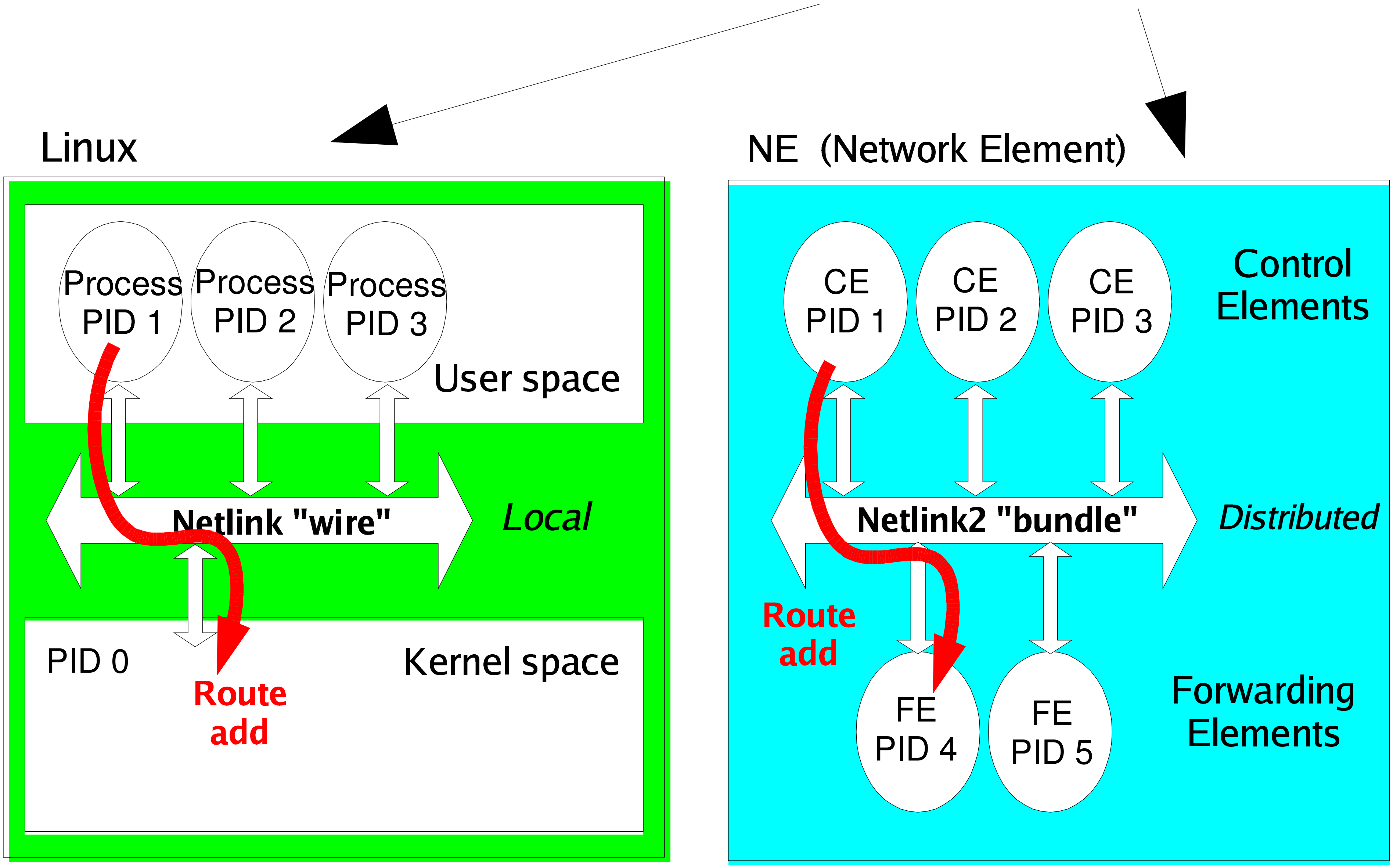
Motivation: Why Netlink derived ?

- Netlink already has relevant protocol features:
  - Connectionless
  - Asynchronous oriented
  - Unicast or Multicast (one FE to many CEs)
  - Ability to run both in reliable and unreliable modes
  - Event handling
    - Port events, table events, etc

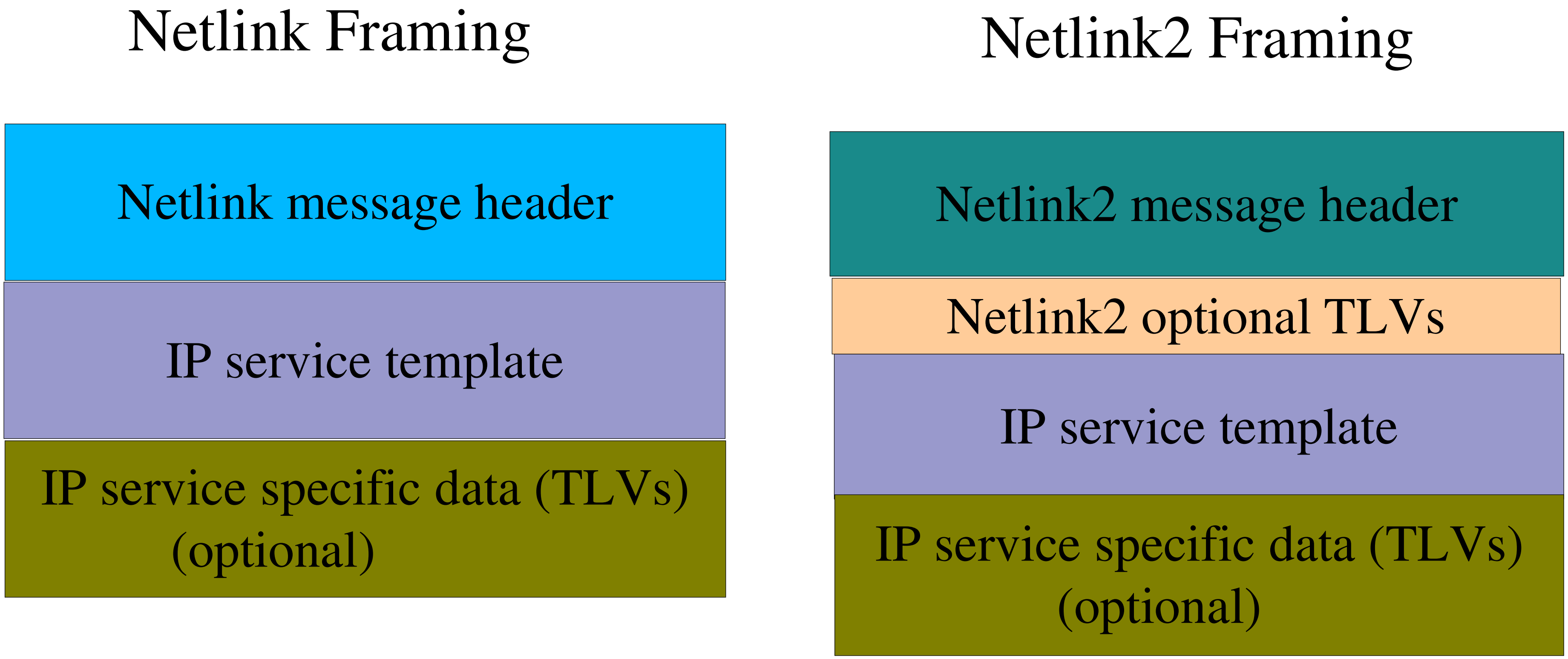
Motivation: Why Netlink derived ?

- Netlink Framing mostly complete for ForCES:
  - CE - FE addressing
    - for local, single FE, single CE case
  - Extensibility (use of TLVs)
  - Many services relevant to ForCES already defined
    - IPv4 forwarding service header covers RFC1812 completely
    - Refer to Netlink draft for examples and latest linux kernel.
    - http://www.ietf.org/internet-drafts/draft-ietf-forces-netlink-04.txt

Architecture: From Netlink to Netlink2



Netlink2: General Framing changes



Netlink Framing

Netlink message header

IP service template

IP service specific data (TLVs) (optional)

Netlink2 Framing

Netlink2 message header

Netlink2 optional TLVs

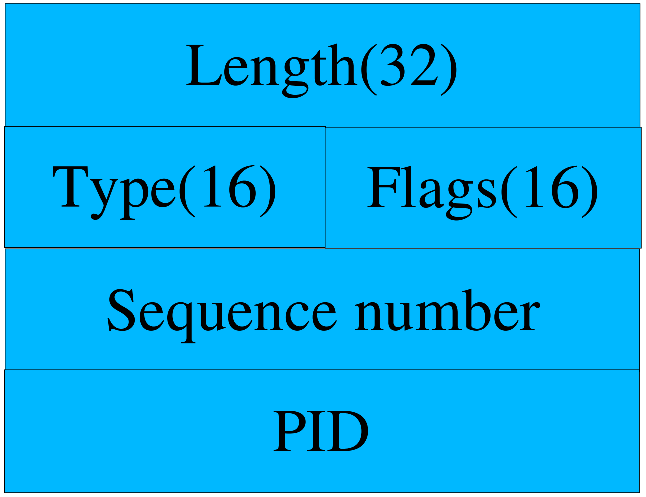
IP service template

IP service specific data (TLVs) (optional)

- Changes:
  - Netlink header extension
  - ➔Additional optional Netlink2 TLVs

Netlink Header extension

Netlink Header



Length(32)

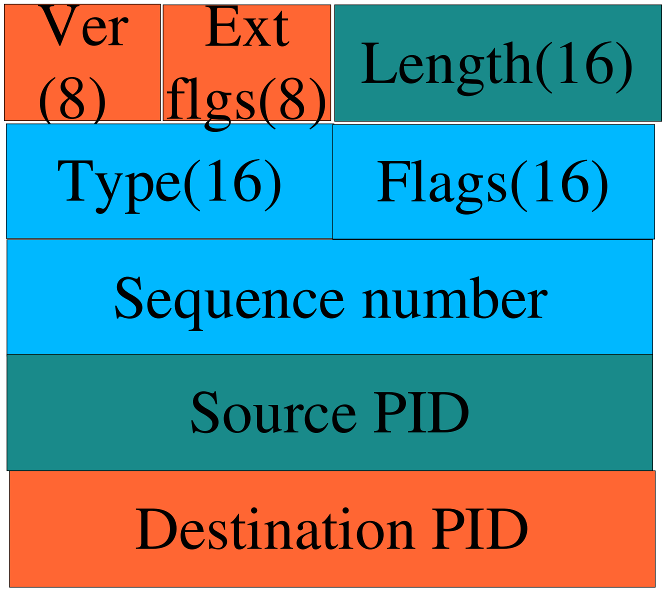
Type(16) Flags(16)

Sequence number

PID

- Length Field reduced to 16 bits
- New Extended flags
  - ➔*NLM_F_SYN* Join message
  - ➔*NLM_F_FIN* Departure message
  - ➔*NLM_F_ETLV* Extended TLVs on
  - ➔*NLM_F_PRIO* Message Priority
  - ➔*NLM_F_ASTR* ACK strategy

Netlink2 Header



Ver (8) Ext flgs(8) Length(16)

Type(16) Flags(16)

Sequence number

Source PID

Destination PID

- Version
- PID renamed Source PID
- New Destination PID

Optional TLVs in Netlink2 Header

- Checksum (see RFC3358)

Type = 12	Length = 2	Value = 16 bit checksum

- Message Priority

Type = 13	Length =2	Value = 16 bit priority

Netlink2 Addressing: Wires and Bundles

- Use IP addressing
- A Netlink2 wire is:
  - Pair of unicast IP addresses and ports, or
  - An IP multicast address and UDP port.
- A Netlink2 bundle is:
  - One or more Netlink2 wires
- Use UDP/TCP/SCTP for transport
- Encapsulation for global scope (out of black box)

Netlink2 Addressing: PIDs

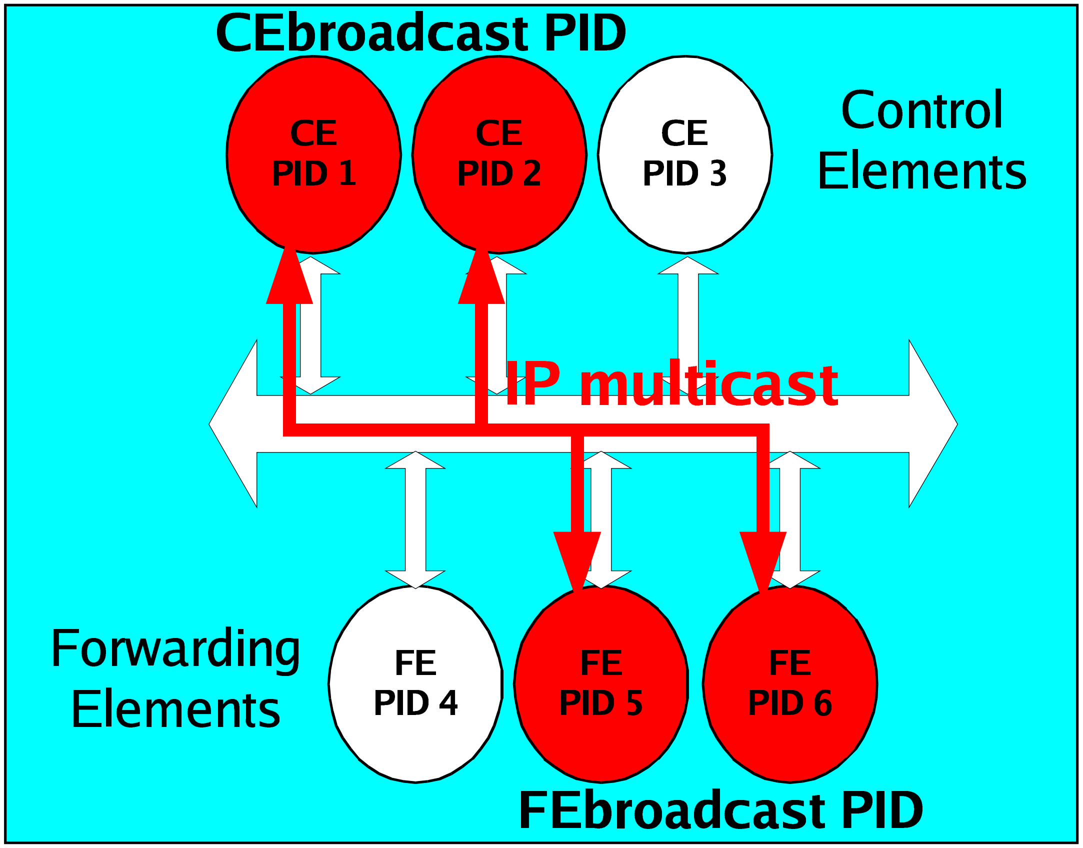
- An FE/CE must process an incoming message if the destination PID is:
  - The unicast PID of the FE/CE, or
  - A logical PID to which the FE/CE belongs to, or
  - The broadcast PID

Netlink2 Addressing: how it works

- A Netlink2 message placed on a Netlink2 wire is delivered to all parties connected to this wire.
  - Parties that have a suitable PID MUST actively process the message
  - Other parties MAY passively process messages for redundancy and HA (High Availability) state maintenance reasons
- Sequencing per wire, ACKs per bundle

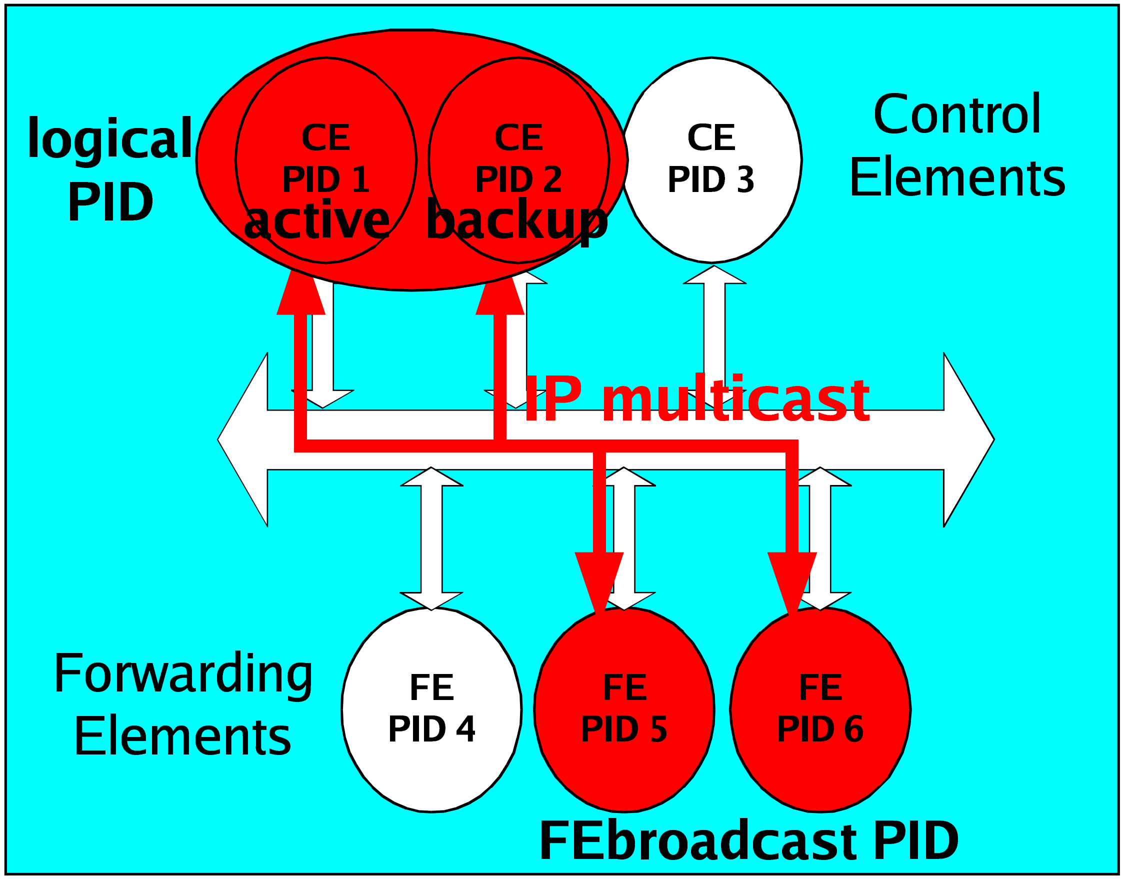
Examples of Netlink2 wires and bundle

Bundle:
IP mcast+port for CEs 1,2 and FEs 5,6



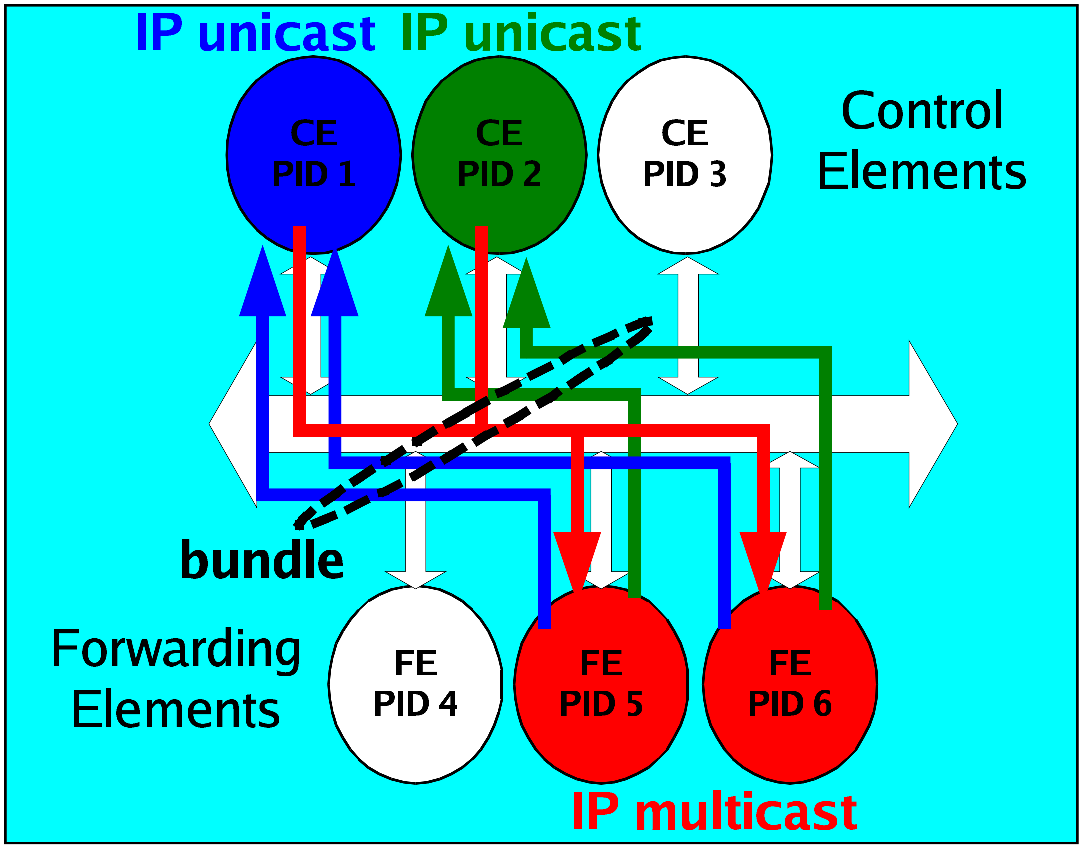
Examples of Netlink2 wires and bundle

HA scenario: logical PID for CEs 1 and 2



Examples of Netlink2 wires and bundle

Bundle:
IP unicast+port for CE 1
IP unicast+port for CE 2
IP mcast+port for FEs 5,6



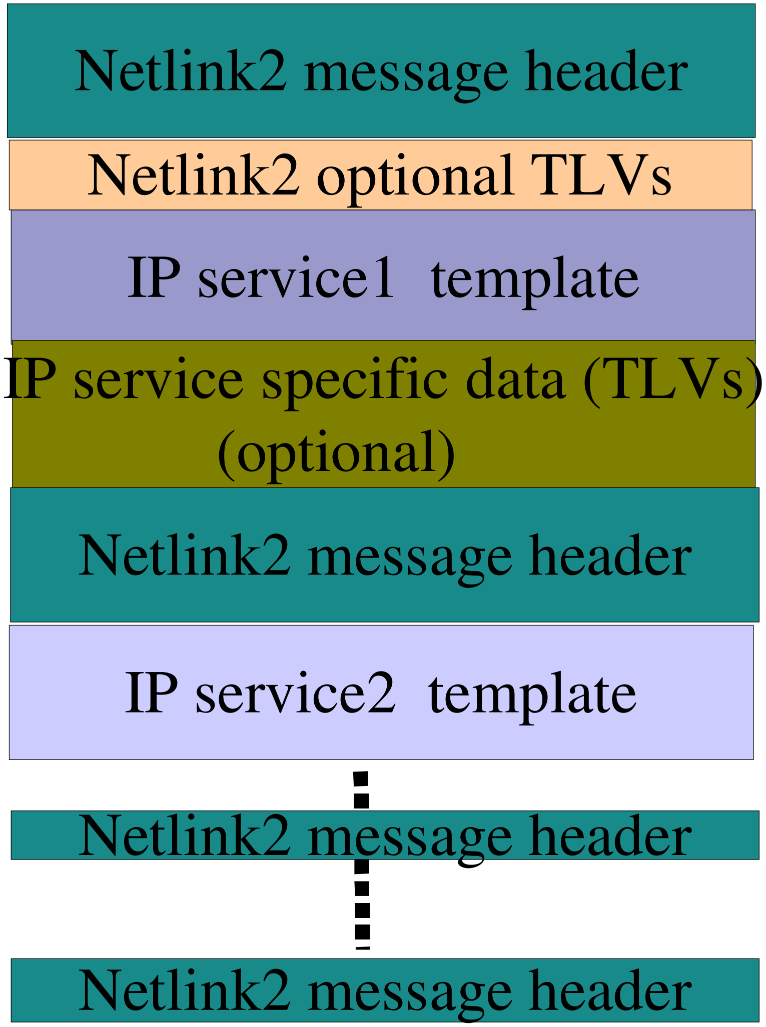
Netlink2: mechanisms for creating protocols

- Building reliability
  - ACKs can be requested on sending msg
  - Netlink(2) has sequence numbers
  - Retransmit timers
- Prioritization
  - If out of resources respond to higher priority messages
- ACK strategy
  - Partial ACKs (or ACK "slotting and damping") to save resources

Netlink2: mechanisms for creating protocols

- Building availability
  - As shown earlier multicasting for multiple listener synchronization
  - NLMSG_NOOP and NLM_F_ECHO for heartbeats
- Atomicity and ordering
  - NLM_F_ATOMIC is essentially a lock
  - NLMSG_DONE translates to an unlock
  - Two phase commit:
    - Send a message with transaction and NLM_F_ATOMIC
    - Send a NLMSG_DONE to commit or discard

Netlink2: mechanisms for creating protocols: Batching



Netlink2 message header

Netlink2 optional TLVs

IP service1 template

IP service specific data (TLVs) (optional)

Netlink2 message header

IP service2 template

Netlink2 message header

Netlink2 message header

- NLM_F_MULTI flag on all Netlink2 headers except for last one
- Last Netlink2 message is of type NLMSG_DONE
- NLMSG_DONE could be in a different packet if MTU boundaries exceeded

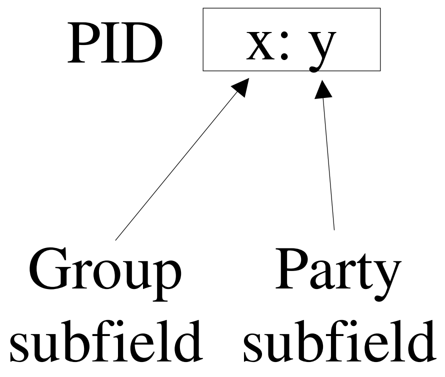
Conclusion

- Netlink2 as ForCES protocol
  - Based on proven and available Netlink
  - Many existing service templates / models
  - Scalability & HA (High Availability) thanks to multicast
  - Flexible wires and bundles of wires
- Discovery of topology, capabilities, etc, will be addressed in revised draft

Summary of draft changes

draft-jhsrha-forces-netlink2-01.txt

- Changed *FEC* (FE Component) to *LFB* (Logical Functional Block), i.e., an FE stage.
  - Netlink2 provides addressing up to the level of LFBs and CECs (CE Components, or processes)
- Structuring of PIDs into *group* and *party* subfields.
- Considerable editorial improvments



ForCES in an nutshell

- A base protocol
  - To move messages on the wire between CEs and FEs, more specifically between CECs and LFBs.
  - With its own addressing
- A set of TLVs derived from
  - FE-level models and LFB-level models
    - To declare and allow manipulation of topology/capabilities/resources of the data path

  ➔Include ForCES-specific LFBs, with:
    - Set of transport protocol(s) available for CE to FE comm,
    - Action(s) when failover, etc.

Netlink2 summary

- Netlink2 is a base protocol between CEs and FEs
- Netlink-derived: CE = user space, FE = kernel
    - Allows reuse of many existing services using Netlink (see RFC 3549)
- Changes from Netlink to Netlink2
    - Message header format extended
- Room for new services
    - Such as topology/capabilities discovery
    - Should be addressed in separate drafts
- Provides transaction reliability, prioritization, availability, atomicity, batching.

ForCES protocol requirements

draft-ietf-forces-requirements-09.txt

- Scalability, 100s of FEs with 100s of ports each
- CE redundancy
- Multiple FEs and CEs, dynamic join/leave
- Encryption/authentication of ForCES messages
- ForCES message priority
- Reliability (built-in: transaction-level reliability)
- Run over various interconnect technologies
- Command bundling and all-or-nothing (atomicity)

Netlink2 addressing

- Goal: have a flexible CE-FE addressing
  - Own CE/FE and CEC/LFB addressing
  - Support for multicast groups
  - Support for transparent HA (active/backup)
  - Mapping groups to IP unicast/multicast, or any other interconnect addressing method (Infiniband, PCI-X, etc).

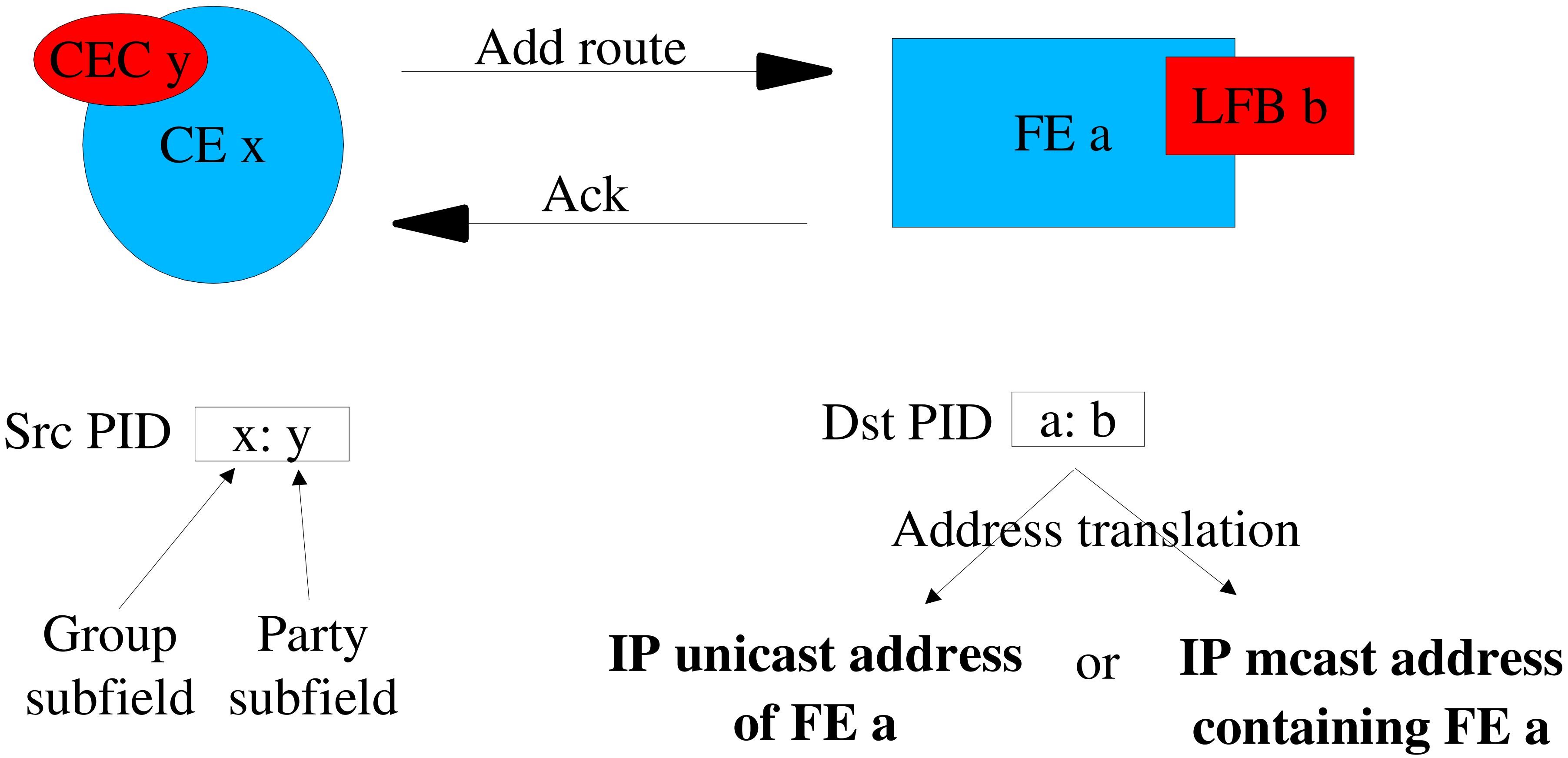
Netlink2 groups

- Allows to address a single or a group of elements (CEs, CECs, FEs, LFBs)
  - Groups can be created
    - By PE (FE or CE)
    - By service type
    - Arbitrarily
- Example of groups:
  - All LFBs instances of type "IPv4_Routing" in the NE
  - All LFBs instances in FE x.
  - Two FEs or CEs running in HA mode.

Netlink2 addressing examples

- Communication scenarios
  - Show duality of groups: PE or service-oriented.
  - Show 4 examples
    - Unicast
    - Multicast
    - Multicast with partial ACKs (avoids ACK implosion)
    - HA (High-Availability)
  - Mapping of PIDs to wires (= IP addresses and ports)

Netlink2 addressing: unicast example



CEC y

CE x

Add route

Ack

FE a

LFB b

Src PID x: y

Group subfield

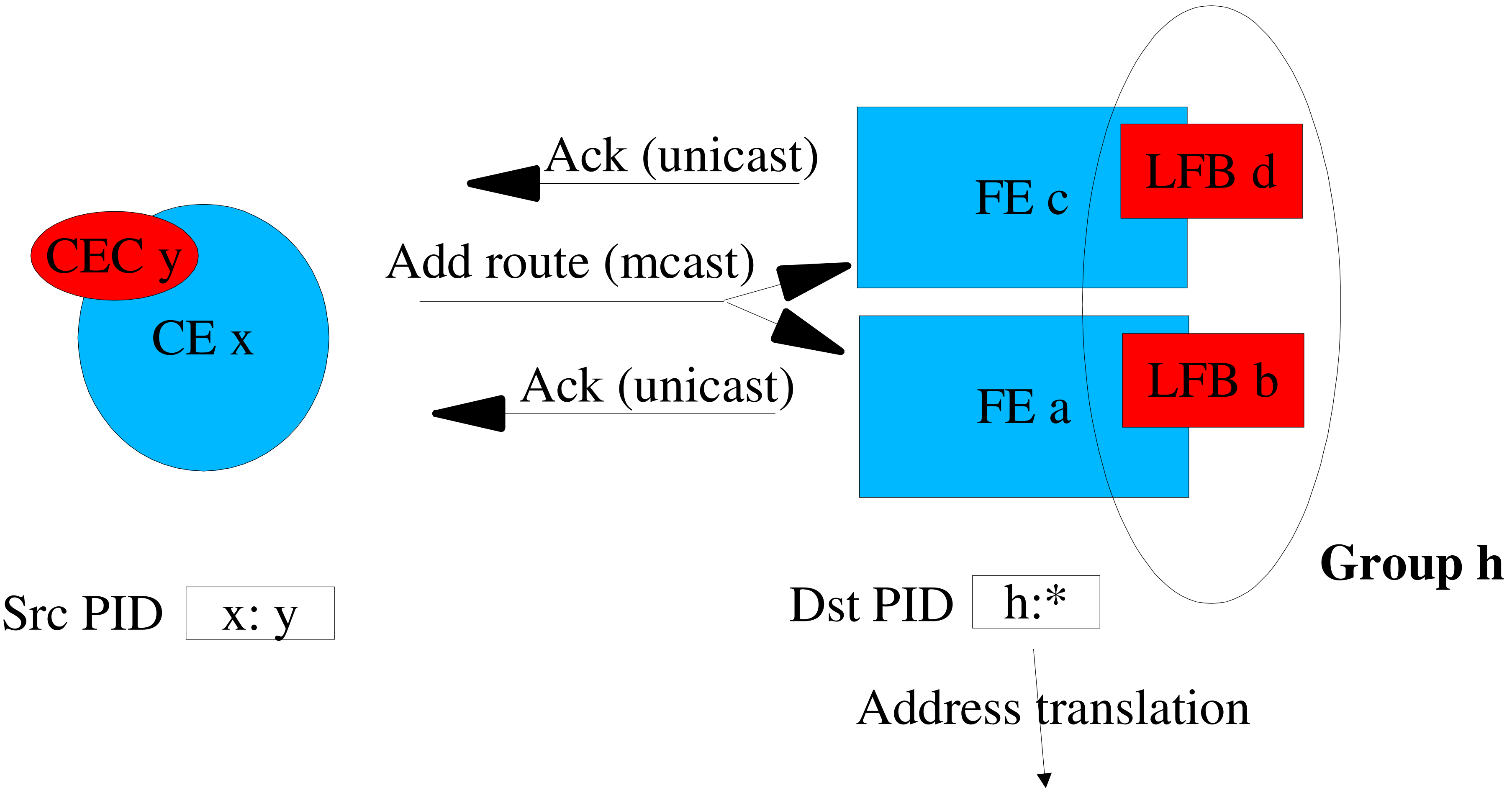
Party subfield

Dst PID a: b

Address translation

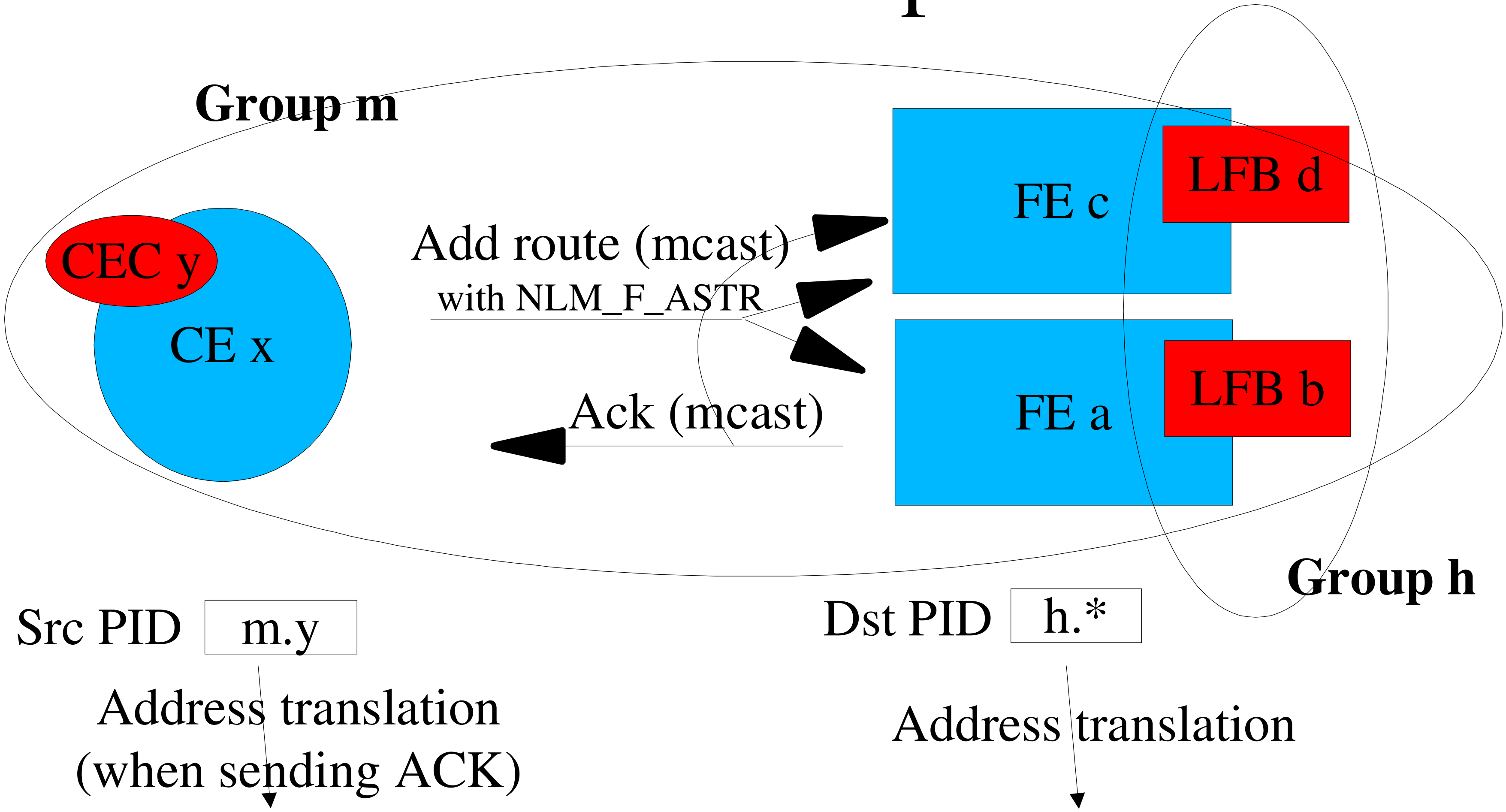
IP unicast address of FE a or **IP mcast address containing FE a**

Netlink2 addressing: mcast example



IP mcast address containing FE a and FE c

Netlink2 addressing: mcast example without ACK implosion



Group m

CEC y

CE x

Add route (mcast)
with NLM_F_ASTR

Ack (mcast)

FE c

LFB d

FE a

LFB b

Group h

Src PID m.y

Dst PID h.*

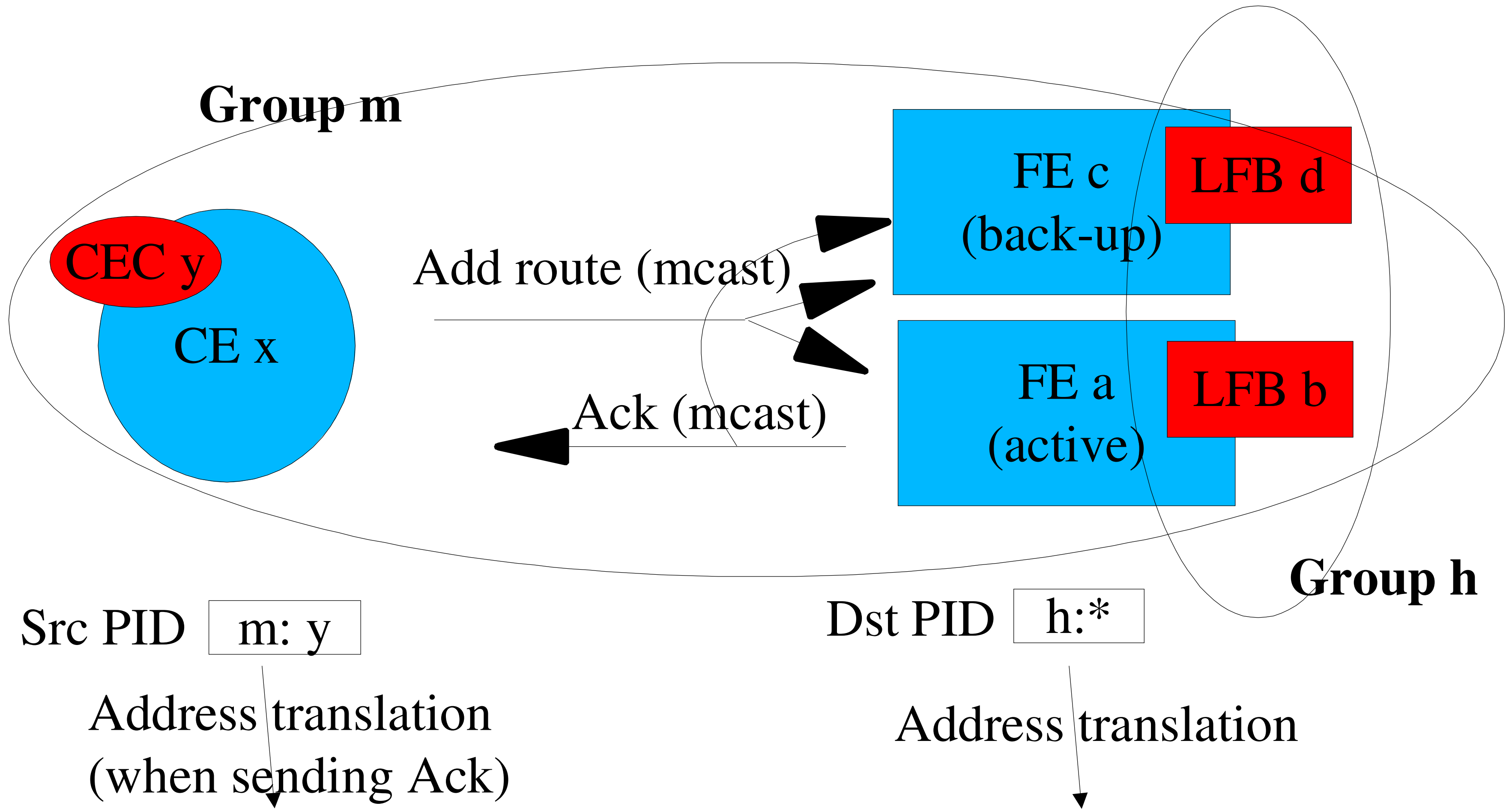
Address translation
(when sending ACK)

Address translation

IP mcast address containing CE x, FE c and FE a.

IP mcast address containing FE a and FE c

Netlink2 addressing: HA example



Group m

CEC y

CE x

Add route (mcast)

Ack (mcast)

FE c
(back-up)

LFB d

FE a
(active)

LFB b

Group h

Src PID m: y

Address translation
(when sending Ack)

Dst PID h:*

Address translation

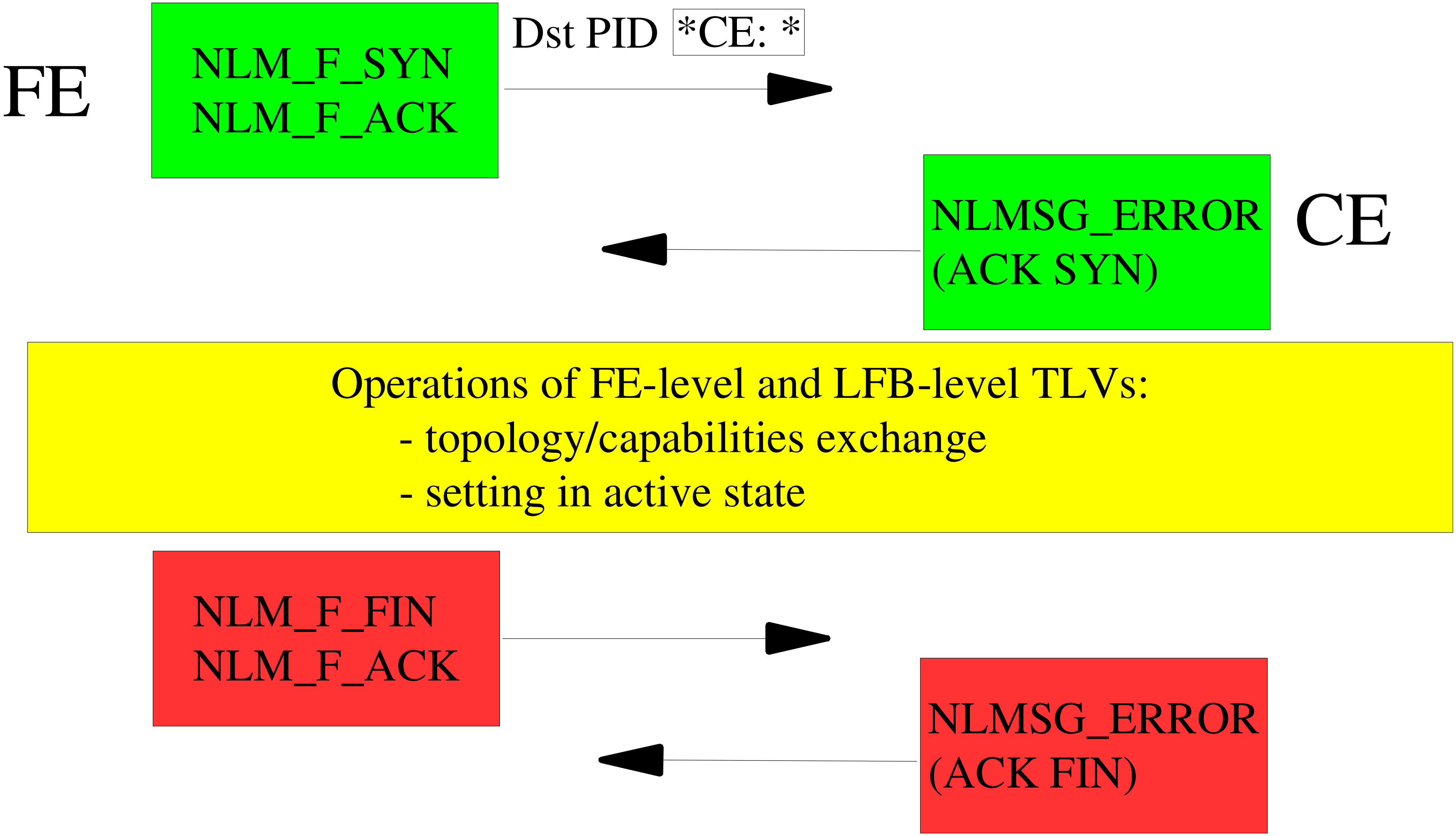
**IP mcast address
containing CE x, FE c and FE a.**

**IP mcast address
containing FE a and FE c**

Netlink2 communication

- Setup of association
  - SYN and FIN messages
- Multipart transaction with two-phase commit
  - Use of NLM_F_MULTI and NLM_F_ATOMIC flags

SYN/FIN messages



FE

NLM_F_SYN
NLM_F_ACK

Dst PID *CE: *

NLMSG_ERROR
(ACK SYN)

CE

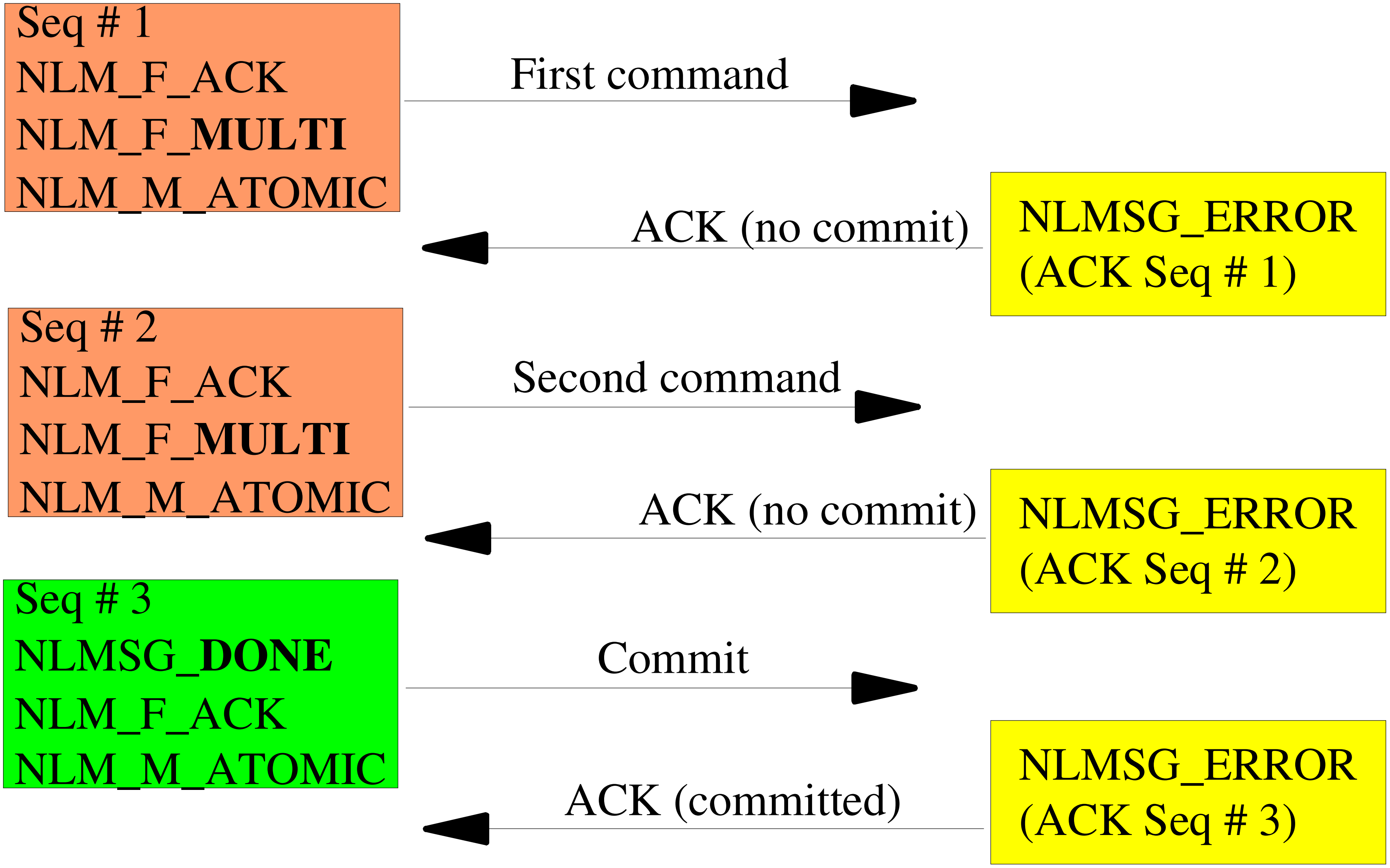
Operations of FE-level and LFB-level TLVs:

- topology/capabilities exchange
- setting in active state

NLM_F_FIN
NLM_F_ACK

NLMSG_ERROR
(ACK FIN)

2-phase commit message exchange



Seq # 1
NLM_F_ACK
NLM_F_**MULTI**
NLM_M_ATOMIC

First command

ACK (no commit)

NLMSG_ERROR
(ACK Seq # 1)

Seq # 2
NLM_F_ACK
NLM_F_**MULTI**
NLM_M_ATOMIC

Second command

ACK (no commit)

NLMSG_ERROR
(ACK Seq # 2)

Seq # 3
NLMSG_**DONE**
NLM_F_ACK
NLM_M_ATOMIC

Commit

ACK (committed)

NLMSG_ERROR
(ACK Seq # 3)

Conclusion

- Netlink2 fulfills ForCES base protocol requirements
  - Key features are scalability and flexibility
    - Use of groups
- FE-level and LFB-level TLVs are to be defined in separate drafts
  - RFC 3549 on "Netlink as IP Services Protocol"
  - Add ForCES-specific TLV(s)