56th IETF, March 18th, 2003

Netlink2 as ForCES protocol

draft-jhsrha-forces-netlink2-00.txt

Robert Haas, IBM Research
Jamal Hadi, Znyx Networks

Outline

- Motivation: why Netlink derived ?
- Changes from Netlink to Netlink2
  - Message header format
- Addressing: Wires and bundles
  - 3 examples
- Reliability, prioritization, availability, atomicity, batching.

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

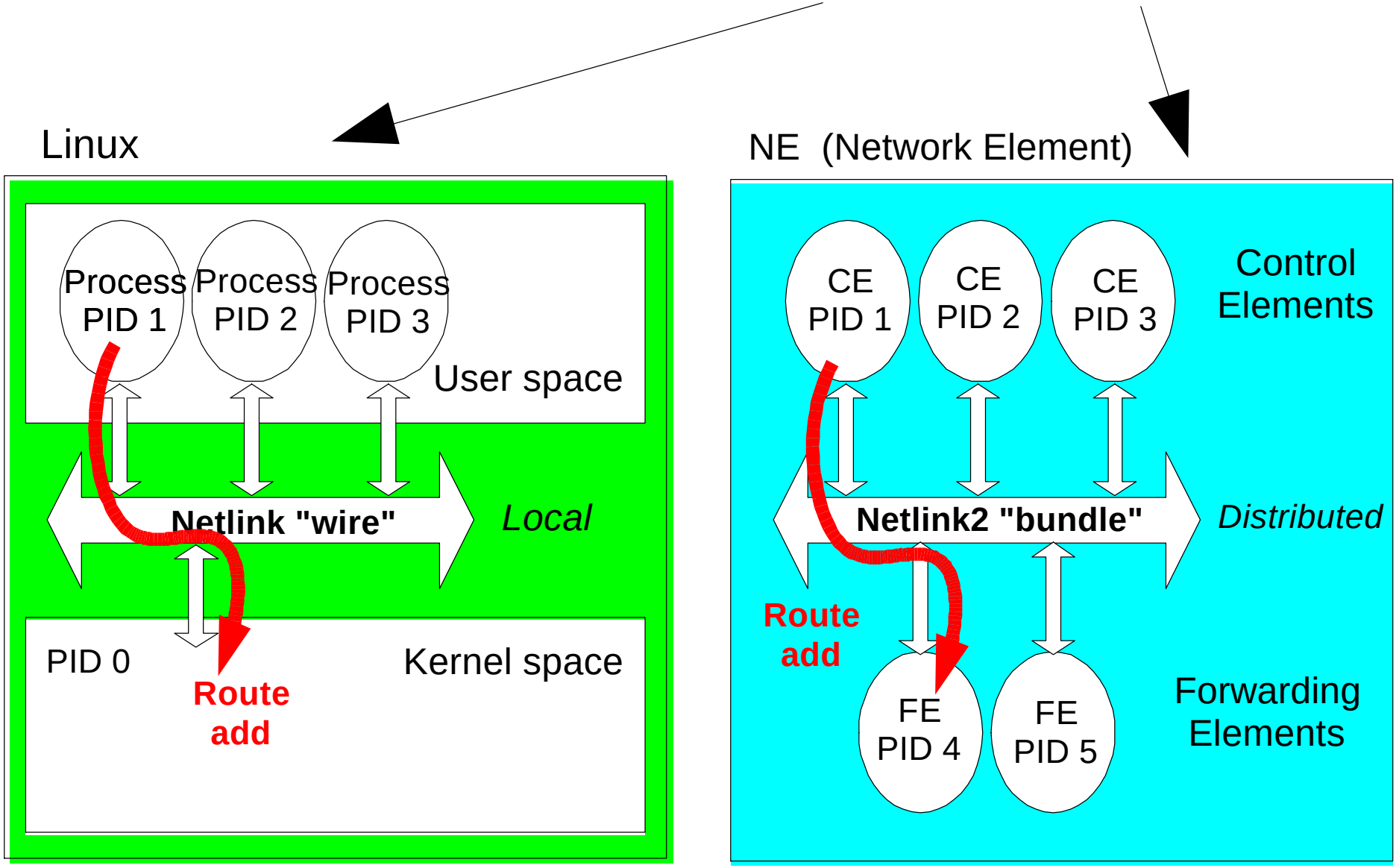
Motivation: Why Netlink derived ?

- 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 ?

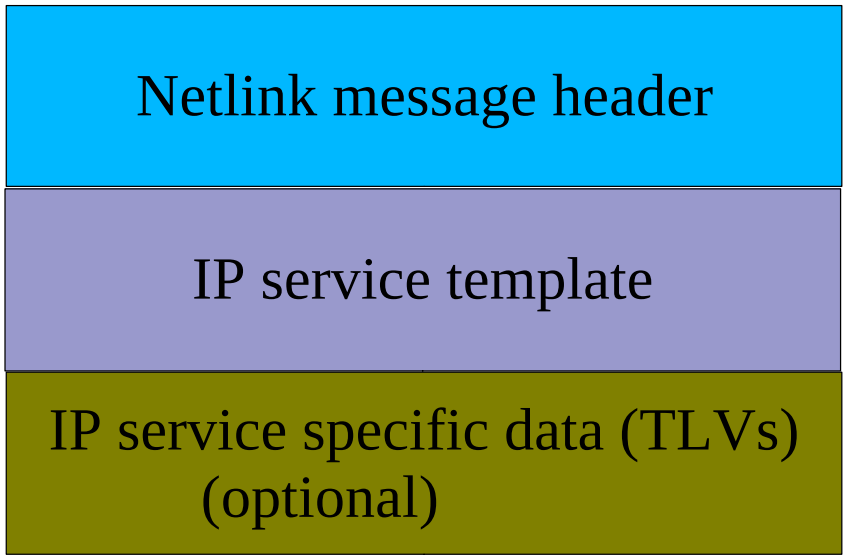
- Netlink Framing mostly complete for ForCES:
  - CE - FE addressing
    - for local, single FE, single CE case
  - Extensibility (use of TLVs)
  - Many services relevant to ForCES already defined
    - IPv4 forwarding service header covers RFC1812 completely
    - Refer to Netlink draft for examples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

Architecture: From Netlink to Netlink2

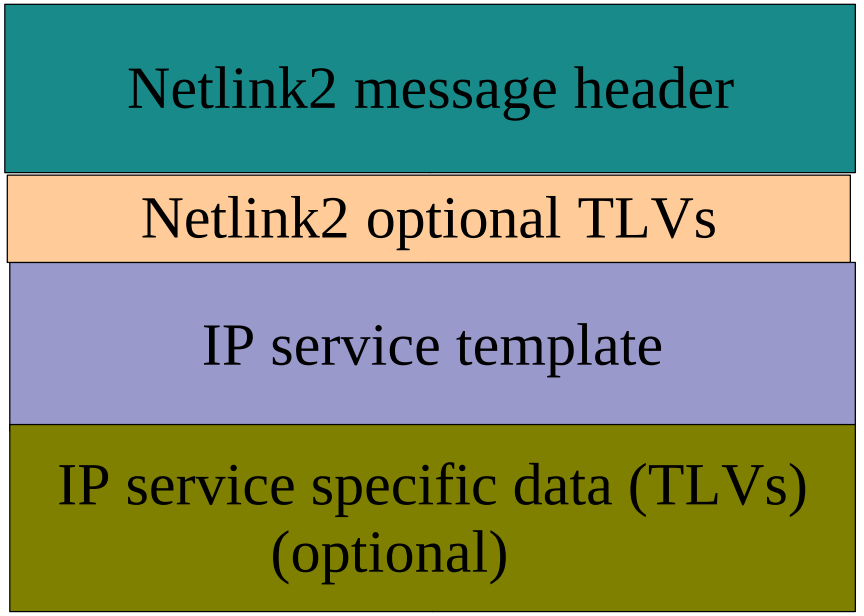


Netlink2: General Framing changes

Netlink Framing



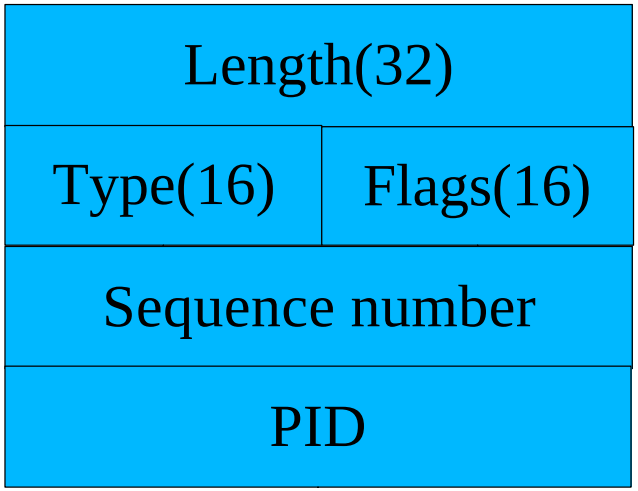
Netlink2 Framing



- Changes:
  - Netlink header extension
  1. Additional optional Netlink2 TLVs

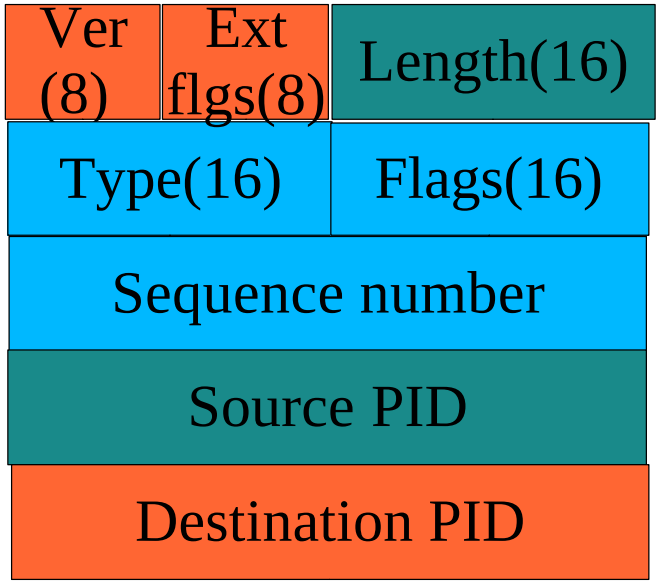
Netlink Header extension

Netlink Header



- 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



- Version
- PID renamed Source PID
- New Destination PID

Optional TLVs in Netlink2 Header

- Checksum (see RFC3358)

- Message 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

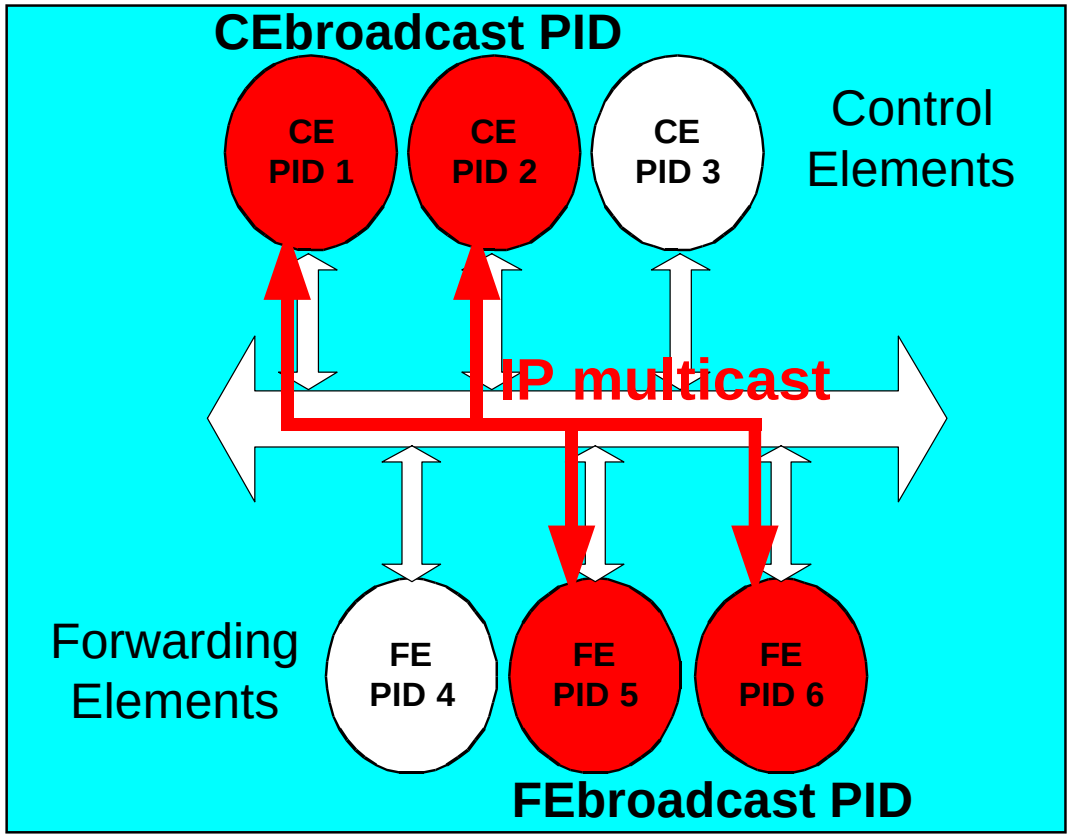
- An FE/CE must process an incoming message if the destination PID is:
  - The unicast PID of the FE/CE, or
  - A logical PID to which the FE/CE belongs to, or
  - The broadcast PID

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

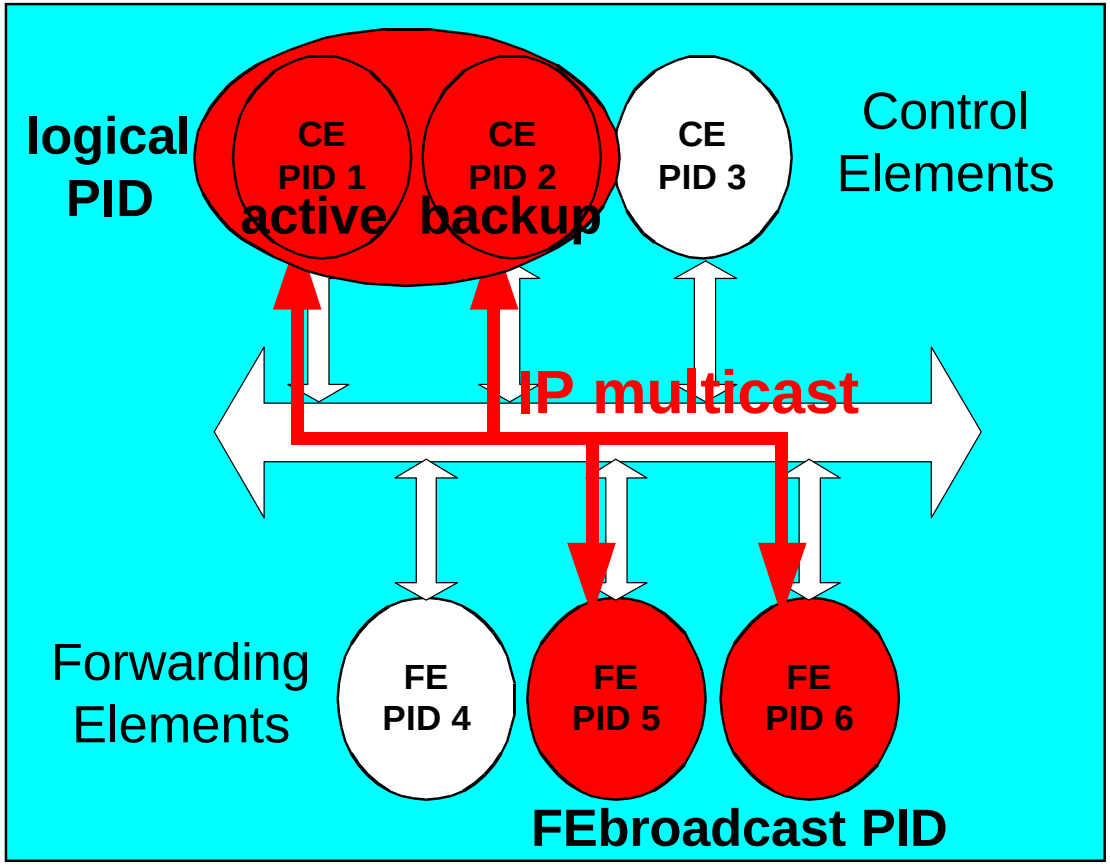
Examples of Netlink2 wires and bundle

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



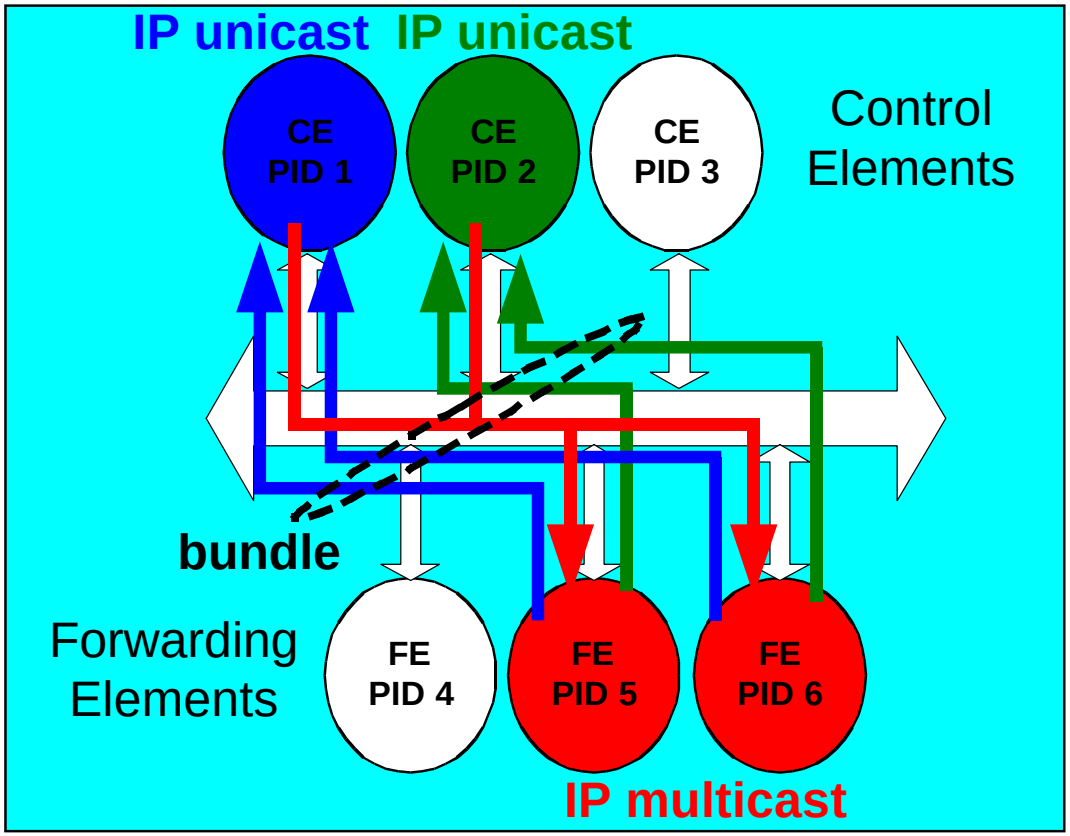
Examples of Netlink2 wires and bundle

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



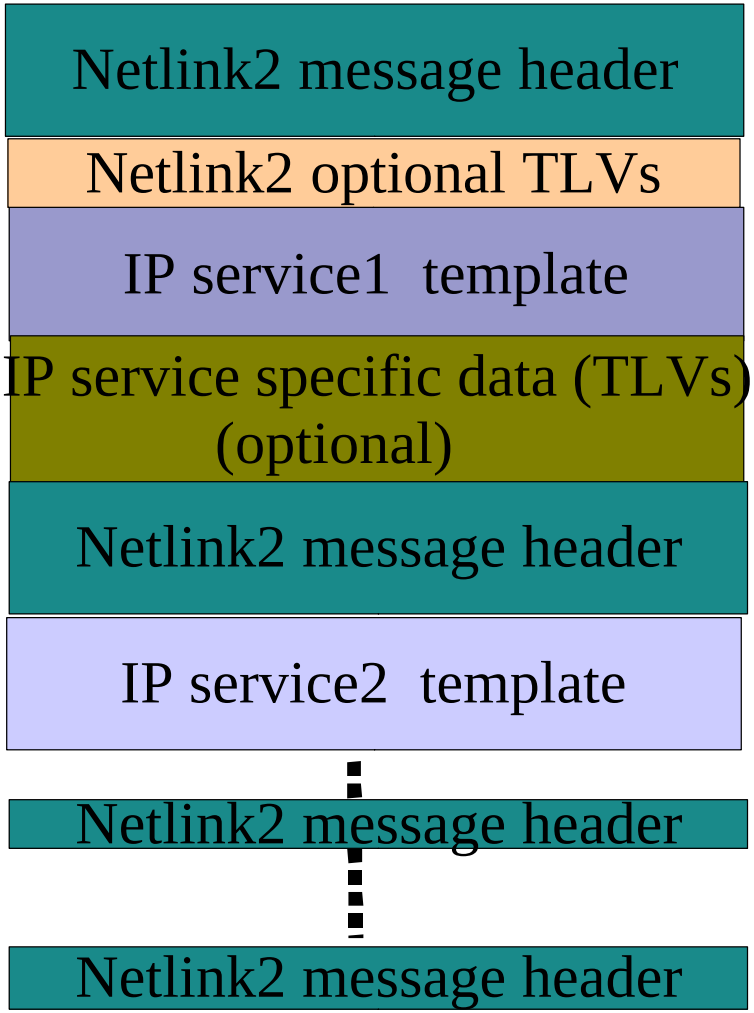
Netlink2: mechanisms for creating protocols

- 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



- 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