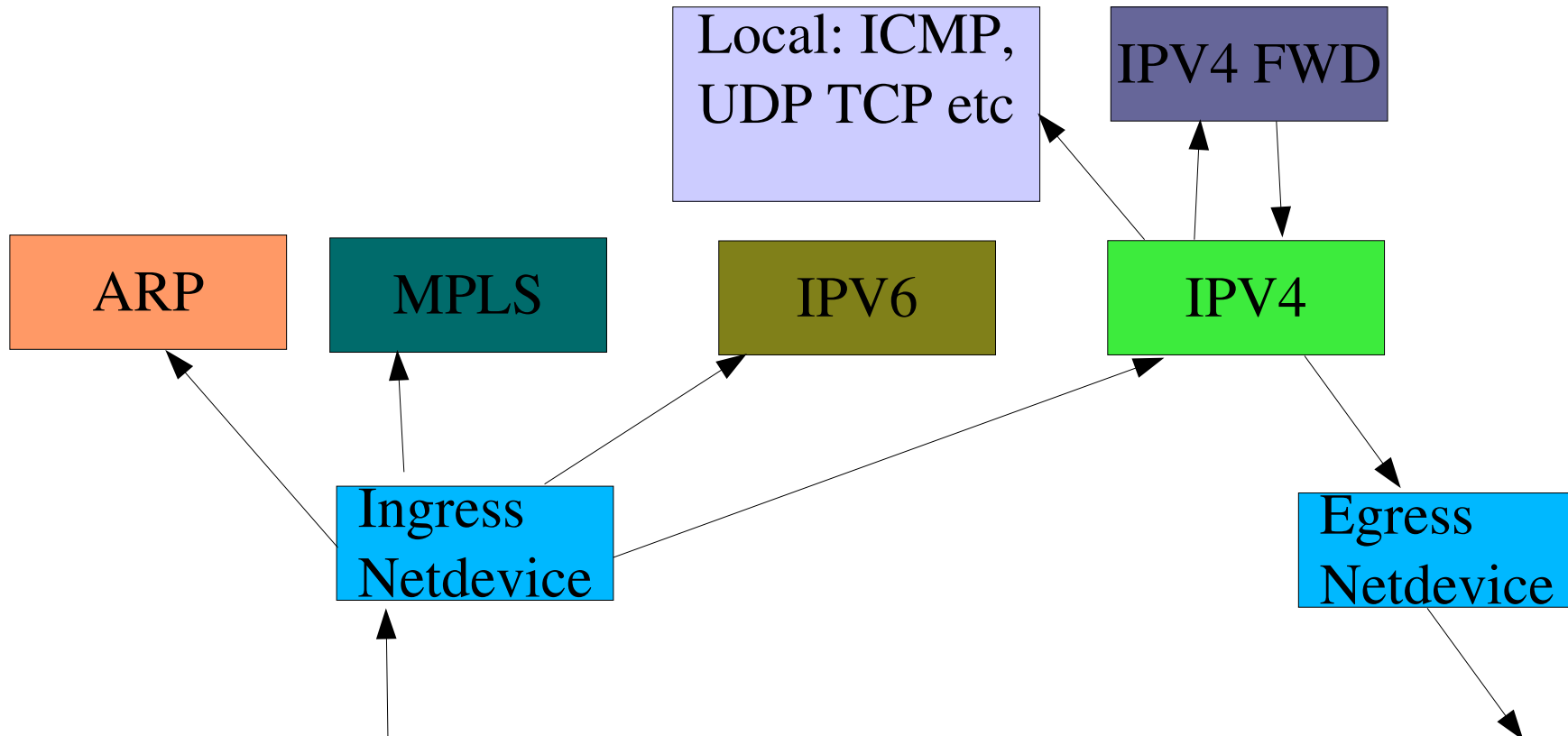


Some Sample LFBs: Netdevice, IPV4, and IPV6

Jamal Hadi Salim <hadi@znyx.com>

Sample LFB topology



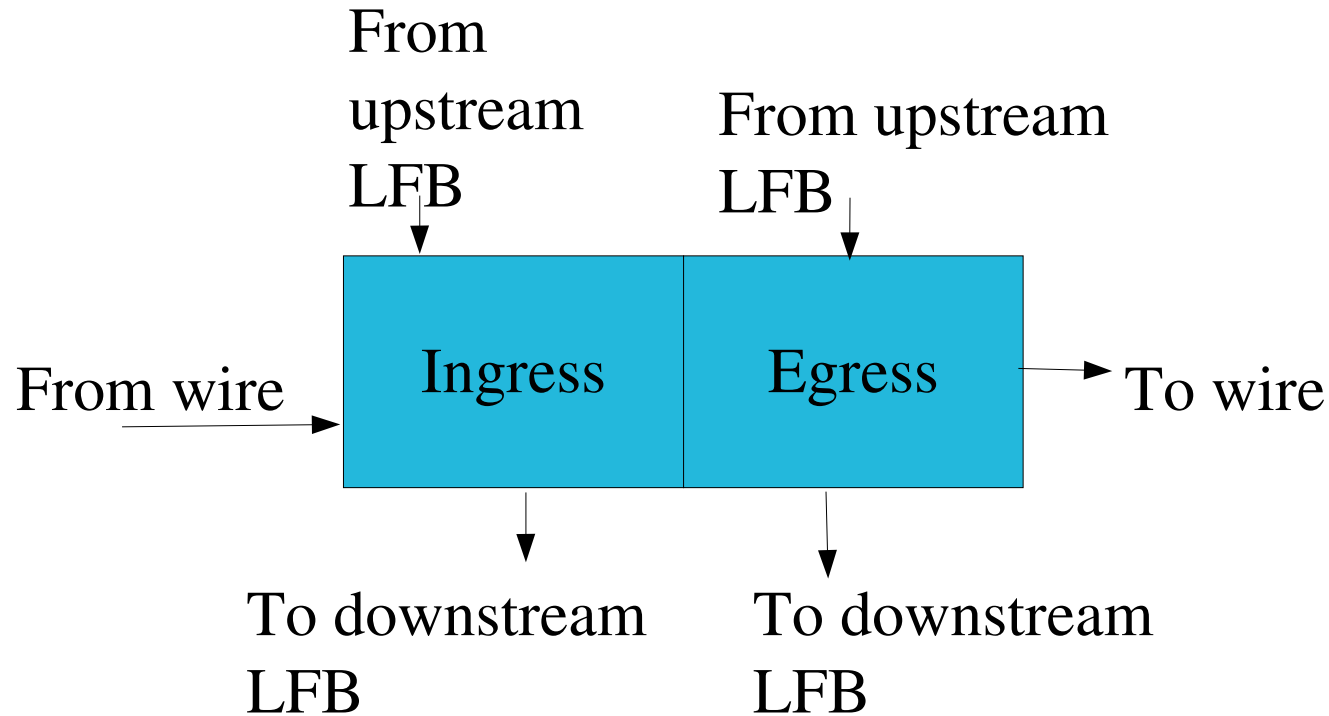
Goal to show A simple example topology

- then focus on Netdevice and IPV4 LFBs for this presentation
- Only 15 minutes allocated (not much time to go into details)

Netdevice LFB

- What has so far been referred to as the *Port LFB*
 - Noun *Netdevice* chosen because *port* is confusing
- The netdevice LFB abstraction does L1 and L2 processing
 - This simplifies things and rids of the need for an L1 LFB
 - So we can lump things like SPI and XAUI in netdevice
 - L2 processing includes checking for header validity, etc but not things like L2 forwarding (which is a separate LFB)

Netdevice Model: Capabilities



- A Netdevice can have either an ingress and/or egress capability
- A Netdevice can have one or all the input and output ports
 - Typical: ingress To-downstream and egress from-upstream ports

Netdevice Model: Capabilities

- A netdevice
 - Has a type { Ethernet, POS, etc }
 - Capabilities adapt on a per type basis
 - Mention LFBs that can occur before/after
 - A netdevice can be connected in a topology to:
 - any other netdevice for stacking purposes
 - Any other LFB(L2, L3 etc) determined by configuration
 - Receives packets from upstream LFBs or the wire
 - Sends packets to downstream LFBs or the wire
 - Up/Downstream LFBs could be other netdevices

Other known netdevice types

- Derived from ARP protocol identifiers
 - Netrom, ax25, pronet token ring, chaosnet, arcnet, appletalk, frame relay DLCI, ATM, metricom STRIP, IEEE1394(firewire), EUI-64, Infiniband, SLIP, CSLIP, SLIP6, CSLIP6, ROSE, AX25, HWX25, PPP, CISCO HDLC, RAW HDLC, Digital's DDCMP, IPIP tunnel, IP6IP6 tunnel, Frame Relay Access Device, SKIP virtual interface, Loopback device, Localtalk, FDDI, AP1000 BIF, SIT tunnel, IP over DDP tunnel, GRE over IP tunnel, PIMSM register interface, HIPPI, Nexus 64Mbps Ash, Acorn Econet, Infrared, Point to point FB, FB arbitrated loop, FB public loop, FB fabric, 802.11 Prism
 - Some of these maybe aggregated (example “ethernet-like”)
 - Capture here to make sure they are not lost
 - Need also to look at Interface MIB to see if can use anything

Related LFBs in a topology

- Relationship mapping between netdevice and other LFBs is via topology layout
 - ARP
 - IPV4 LPM
 - IPV6 LPM
 - QoS/network scheduling
 - classifier
 - other netdevices, etc
- Do we need a Mux right after netdevice?

Netdevice Model: Generic attributes

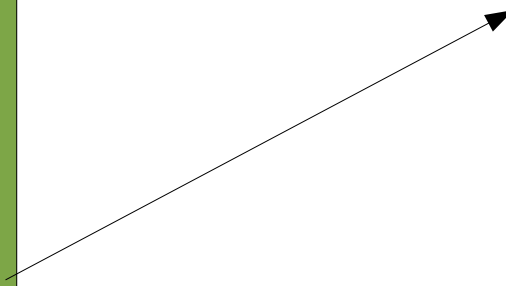
- MIB derived (RFC 2863, 1213, 2233 etc)
 - ifindex
 - name
 - description
 - stats
- flags
 - admin, operational, loopback, point-to-point
- device type indicator
- mtu

Netdevice Model: Attributes

For Ethernet-like device types

addr_len
L2Address
broadcast_addr
mc_count
promiscuity
allmutli
Optional Link Attribs

Autonegotiation,
Flow Control
MAC addr List
Link Speed
etc

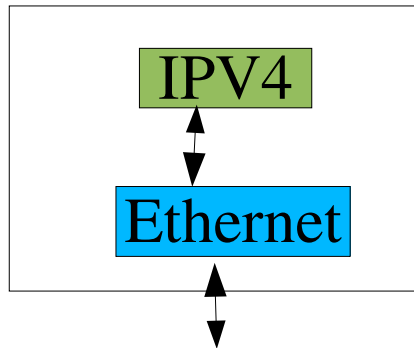


Netdevice Model: Events

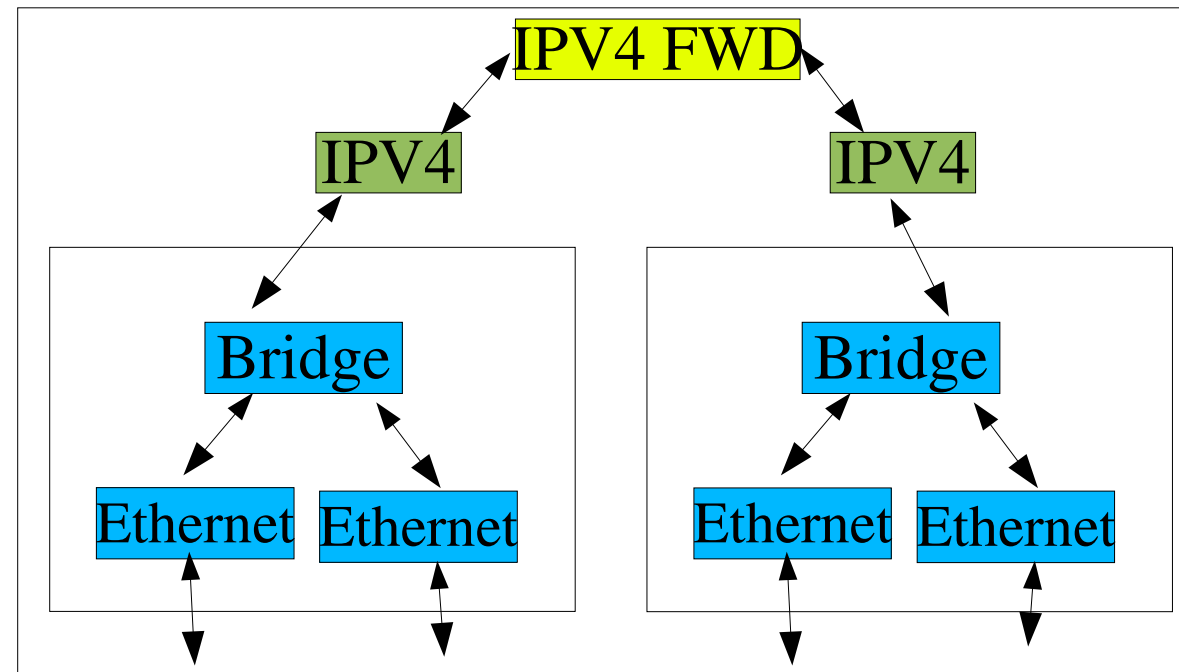
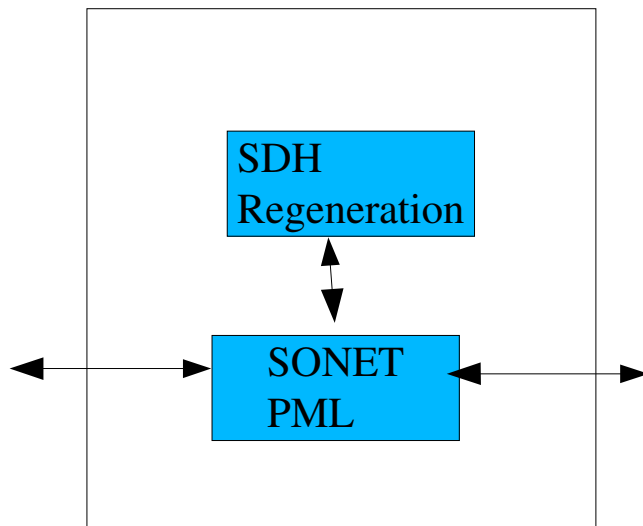
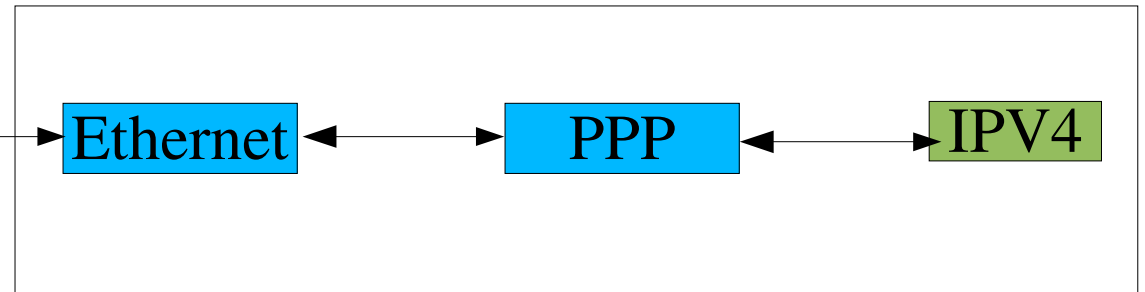
- Generally any attribute modifications could result in an event
 - Admin/Operational
 - up/down
 - Protocol Address
 - Adding/Modifying/Deleting
 - Any other attributes
 - Promiscuity, New MAC address etc
- Other Forces-events
 - Device created, destroyed etc

Example topologies with netdevices

IPV4 Over Ethernet

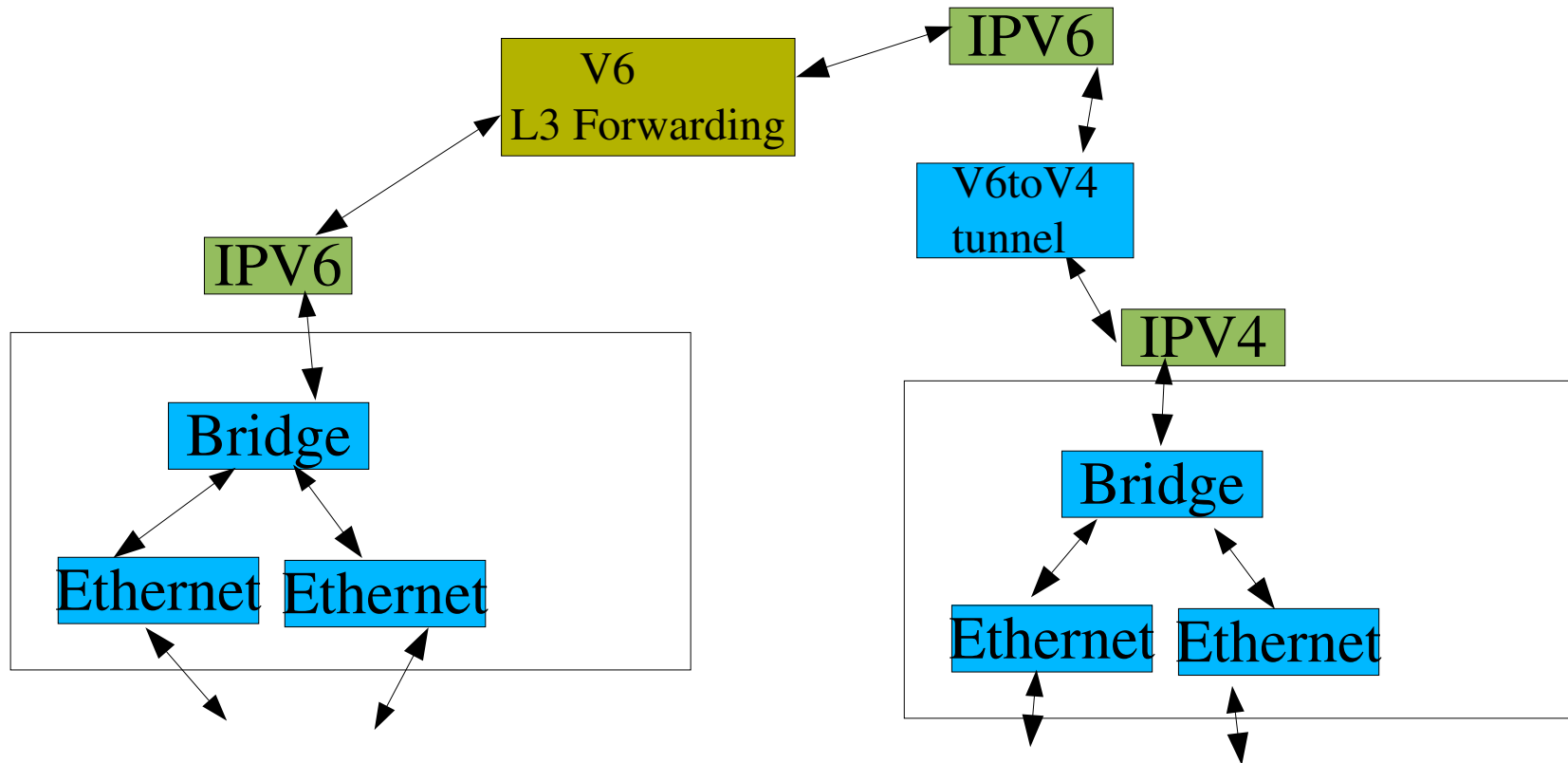


IPV4 over PPP Over Ethernet



L2/L3 Switching

Example Setup



Shows:

- more layers/stacks and more netdevice types
 - Physical(ethernet), Aggregator(Bridge), Tunnel(SIT)
- Basic protocol translations

model definition

...

```
<LFBClassDefs>
  <LFBClassDef>
    <name>netdevice</name>
    <id>6</id>
    <synopsis>Network device LFB</synopsis>
    <version>1.0</version>
    <derivedFrom>baseclass</derivedFrom>

    <capabilities>
      <capability>
        <name>NetdevCount</name>
        <id>1</id>
        <synopsis>Total Number of netdevices</synopsis>
        <typeRef>uint32</typeRef>
        <name>NetdevCapabTable</name>
        <id>2</id>
        <synopsis>Table of Capabilities for netdevs</synopsis>
        <array type="variable-size">
          <typeRef>netdevCapab</typeRef>
        </array>
      </capability>
    </capabilities>

    <Events>
      ..
    </Events>
  ..
  <attributes>
    <attribute access="read-write">
      <name>Netdevtable</name>
      <id>4</id>
      <synopsis>table of netdevices</synopsis>
      <array type="variable-size">
        <typeRef>netdevdesc</typeRef>
      </array>
    </attribute>
  </attributes>
</LFBClassDef>
</LFBClassDefs>
</LFBLibrary>
```

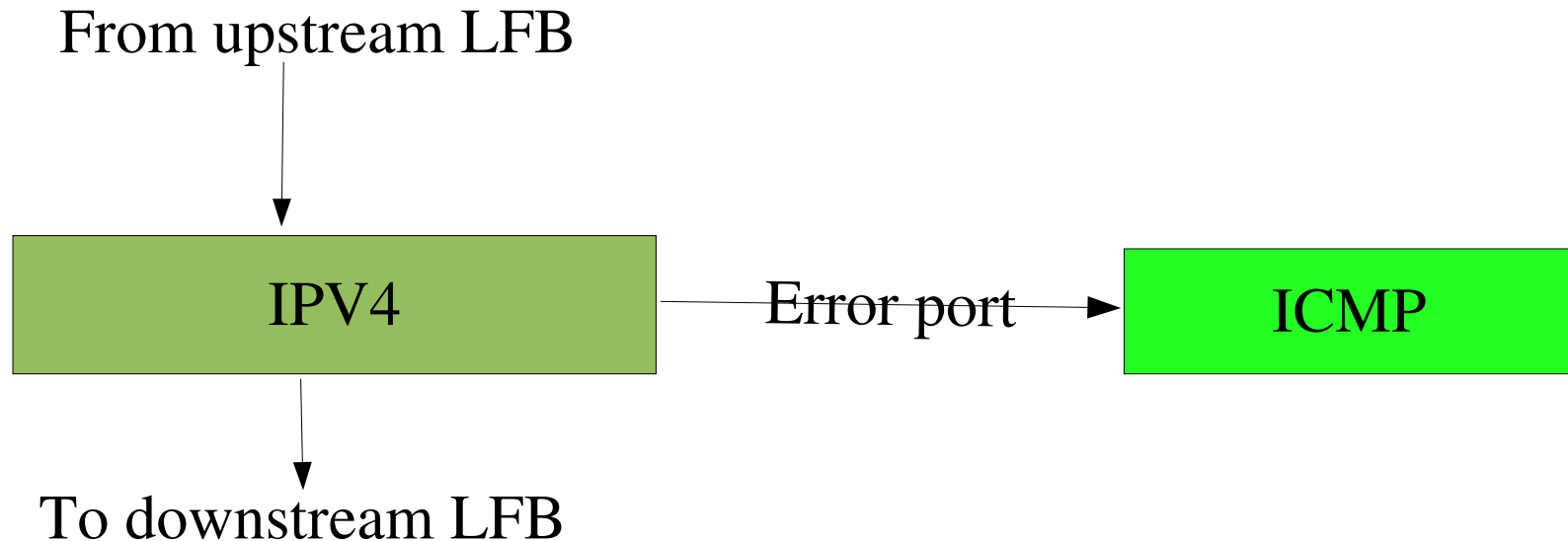
Has been implemented (not converted to new schema yet)

- implementation tried only for ethernet-like netdevices

IPV4 LFB

- The IPv4 LFB abstraction does L3 IPV4 basic processing
 - The basic decrement ttl, validate checksum, recompute checksum
- This LFB is not IPV4 Forwarding although it is related
- It owns the multicast/unicast IPV4 addresses in a node
 - upstream LFBs pass metadata to it to select outgoing IPV4 addresses

IPv4 Model: Capabilities

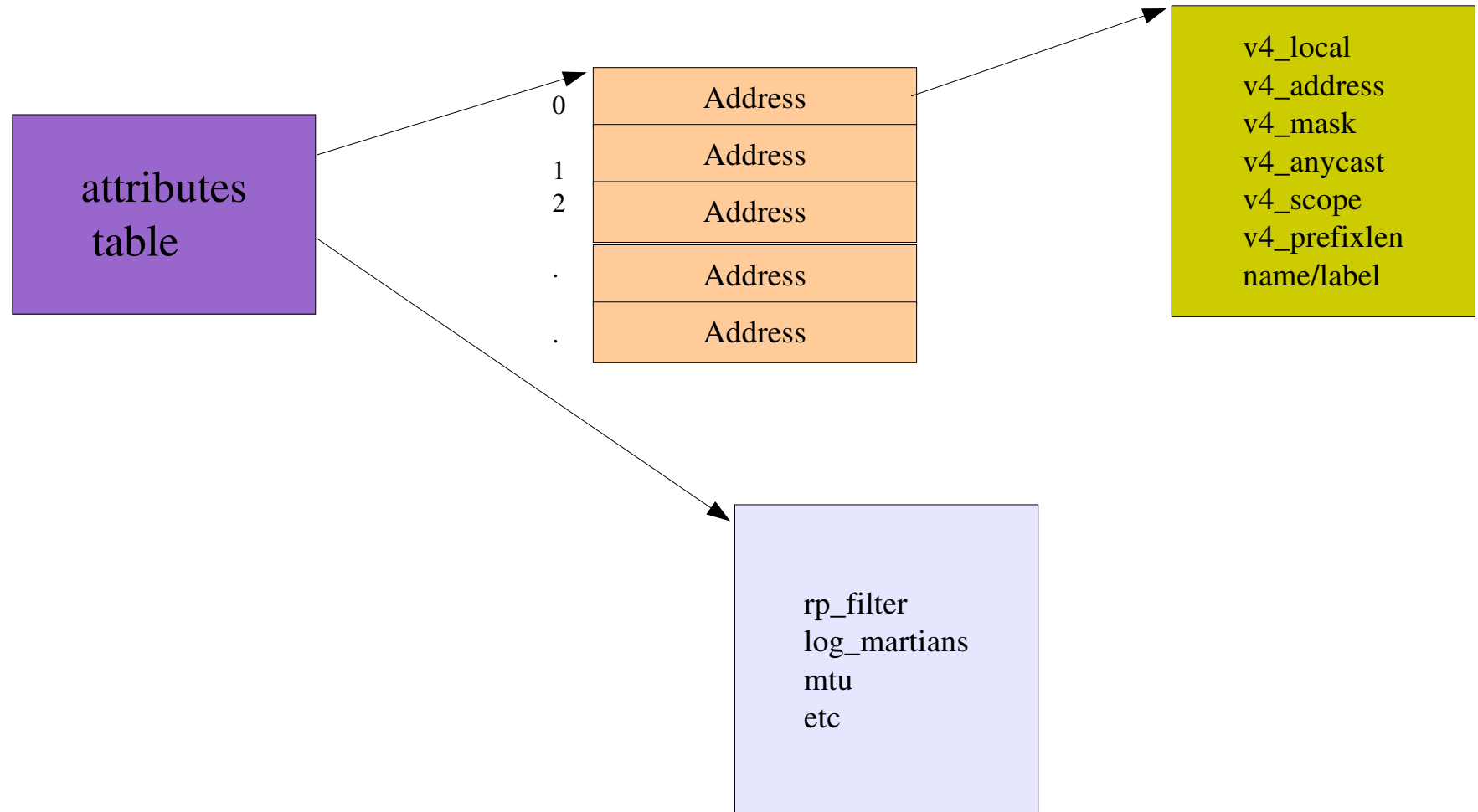


- Implicit tight integration with ICMP
 - ICMP is the error and control path
- Not shown is the exception path (eg processing options)
 - Really part of the LFB internals

IPV4 Model: Capabilities

- Mention LFBs that can occur before/after
- Can do multicast/unicast

IPV4 attributes



IPv6 interface

