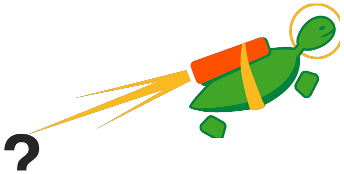


What is an L3 Master Device?

David Ahern — Cumulus Networks

Netdev 1.2, October 2016

Layer 3 Master Device (l3mdev)

Evolved from VRF implementation

Core network stack API

- Can be leveraged by drivers that operate at Layer 3
- Influence FIB lookups
- Access to packets at layer 3

CONFIG_NET_L3_MASTER_DEV

- Kernel config must be set to enable drivers using API (VRF, IPvlan)

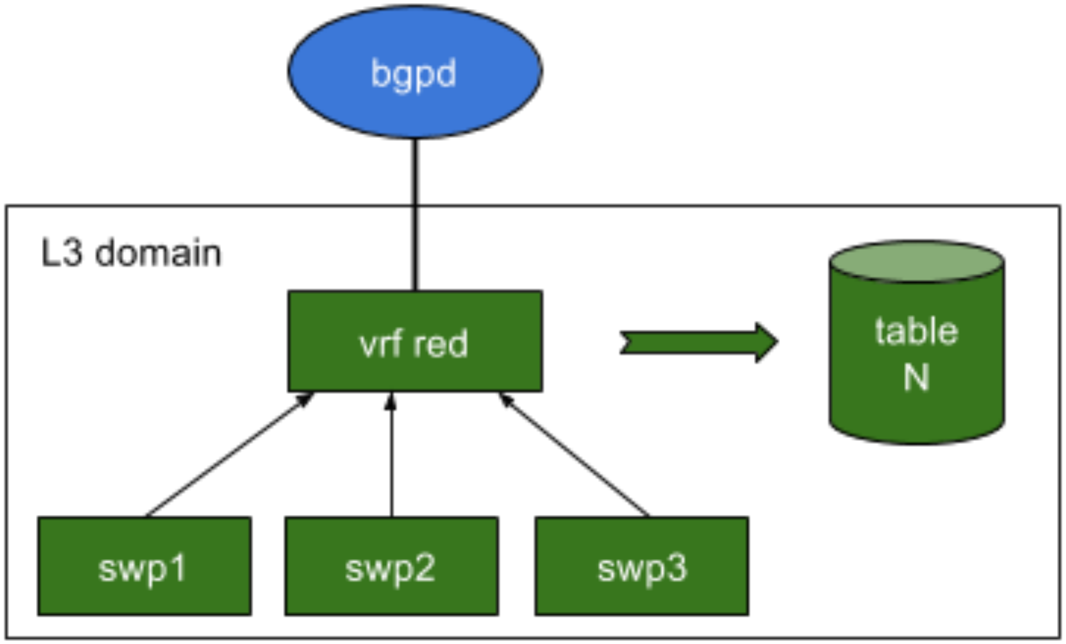
L3 Domains

Primary motivation for L3 master devices

- L3 domains associated with a FIB table

Operational model similar to bridges

- enslave devices to associate with domain
- only L3 decisions affected



L3 domain as a net_device

net_device is a core networking construct

Device-based features that apply to L3 domain

- qdisc, tc filters, netfilter rules, packet capture, domain loopback

Existing policy routing based on oif / iif

Existing userspace APIs

- Bind IPv4/IPv6 socket to l3mdev device to specify L3 domain of interest

Existing operational semantics

- create, delete, show, monitor, enslave

FIB Table for L3 domain

l3mdev_fib_table operation to return table id for device

- Called in fast path; pull table id from private data on device

Contains all routes for domain

- Local, unicast and broadcast
- Host and connected routes moved to table on link up

Additional routes can be added statically or via routing protocol (e.g., bgp)

Policy Routing and FIB Lookups

FIB rules per-device

```
$ ip rule add oif blue table 1001
$ ip rule add iif blue table 1001
```

Single l3mdev rule for all l3mdev devices

```
$ ip rule add l3mdev pref 1000
```

l3mdev_fib_table operation to return table id for device

l3mdev APIs update oif / iif in flow struct

Network Addresses

Source address selection only considers devices in L3 domain

l3mdev is loopback device for L3 domain

- IPv4 loopback address allowed
- Addresses on l3mdev device included in selection

IPv6 linklocal addresses

- no linklocal address on l3mdev device
- no multicast route inserted
- VRF specifically fails lookup for these addresses

Userspace API

Bind socket to l3mdev device

POSIX APIs

- SO_BINDTODEVICE
- cmsg / IP_PKTINFO
  - IP_PKTINFO - can use enslaved device

tcp_l3mdev_accept sysctl

- Allows services to use listen socket across all domains with child sockets attached to specific domain

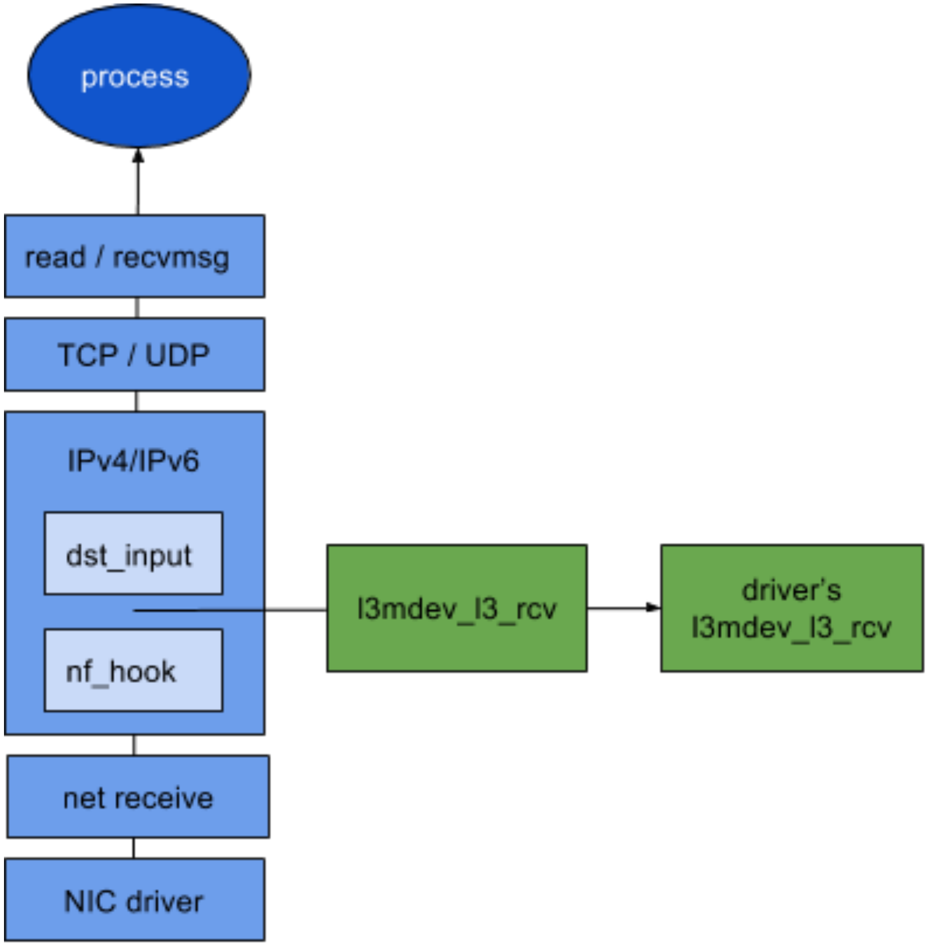
Rx Packet Path

Hook in ingress packet path at L3

- l3mdev_l3_rcv

L3 equivalent of rx-handler

NULL return means skb consumed by handler



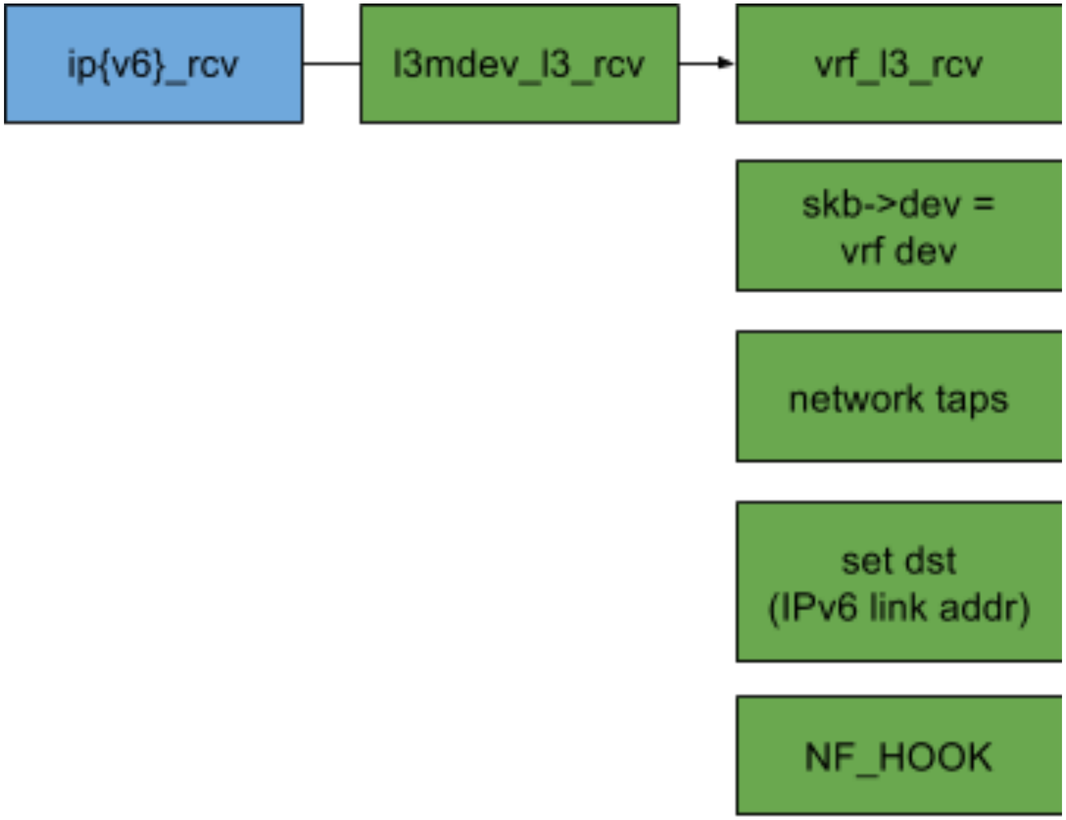
VRF Rx Hook

Switches skb->dev to its device

- original ingress device already saved to skb->cb

Implement device based features

Special case handling of IPv6 linklocal addresses



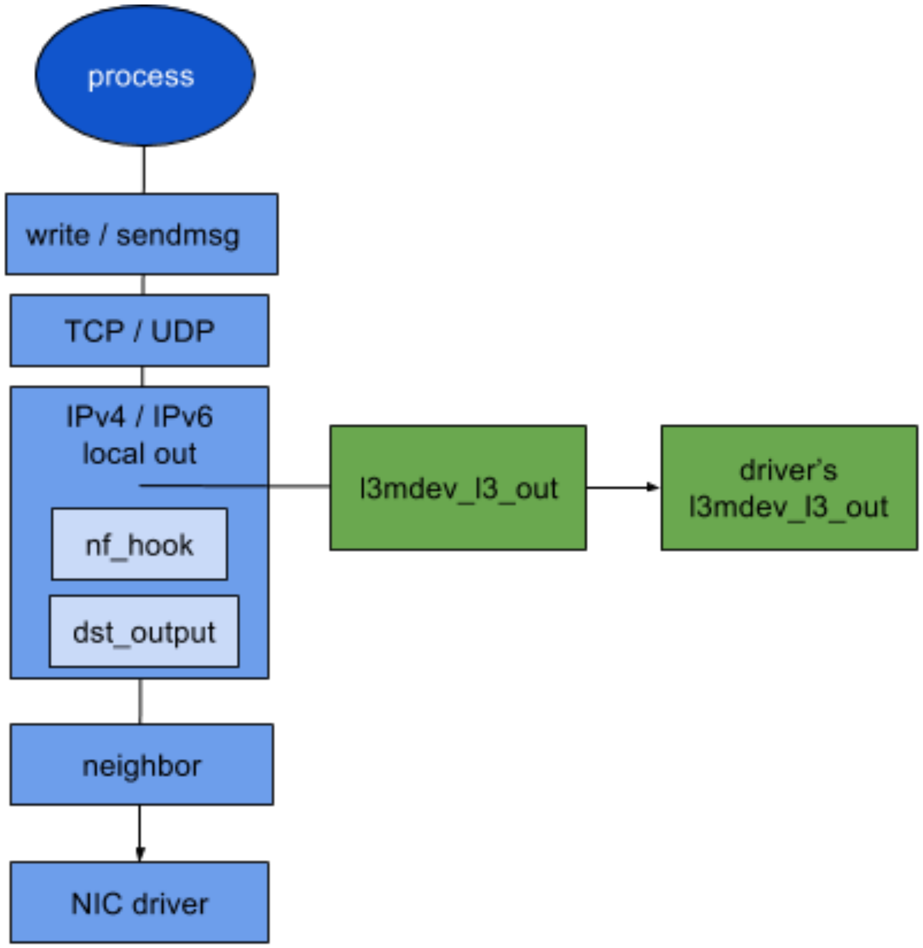
Tx Packet Path

Hook in egress packet path at L3

- l3mdev_l3_out

Called for local traffic before dst->output

NULL return means skb consumed by handler

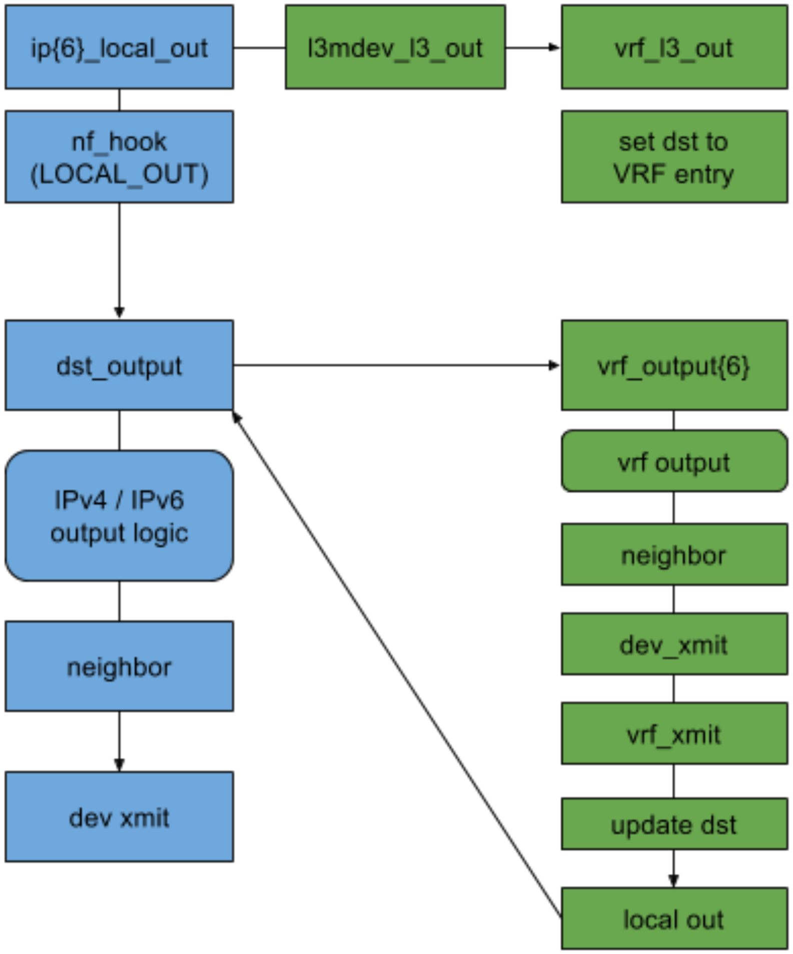


VRF Tx Hook

Sets dst on skb

- Sends packet back to VRF driver after netfilter hook

Basis for device based features on VRF device



l3mdev Driver Operations

Drivers only need to implement operations of interest

- l3mdev_fib_table – returns FIB table for L3 domain
- l3mdev_l3_rcv – Rx hook in network layer
- l3mdev_l3_out – Tx hook in network layer
- l3mdev_link_scope_lookup – route lookup for IPv6 link local and multicast addresses

Device flags

- Master devices: IFF_L3MDEV_MASTER
- Enslaved devices: IFF_L3MDEV_SLAVE

Overhead of l3mdev API

Compiles out if CONFIG_ L3_MASTER_DEVICE not enabled

Minimal as possible when enabled

Sources of overhead

- Extra device lookups
- Device flag checks
- Master device lookup
- Driver operation

Performance of l3mdev devices dictated by device driver

Performance Comparison

netperf UDP_RR with 1-byte payload

- Stresses FIB lookups and l3mdev Rx/Tx hooks

3 cases:

1. l3mdev compiled out - baseline
2. l3mdev compiled in, not used
3. l3mdev compiled in, VRF configured - activates l3mdev hooks

   VRF module reduced to only influencing FIB lookups

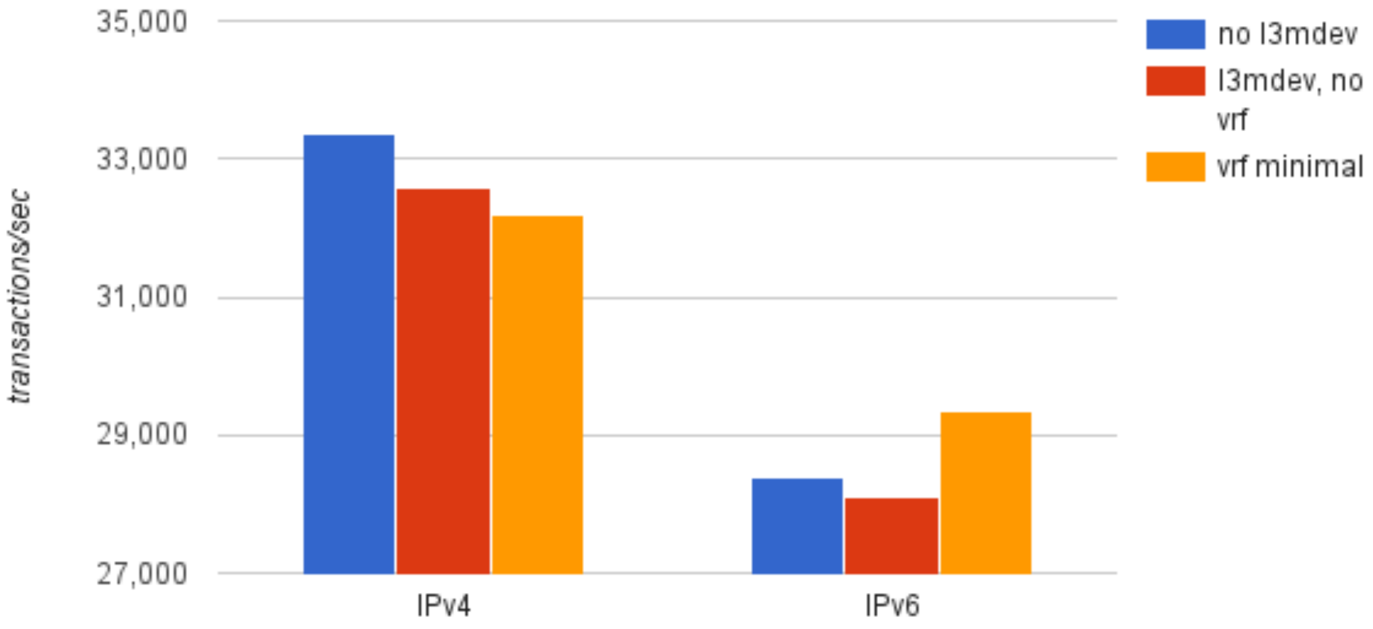
Overhead

Enabling l3mdev

- IPv4: 2.4%
- IPv6: 1.0%

Activating lookups

- IPv4: 3.6%
- IPv6: 3.2% gain



Q & A

Unleashing the Power of Open Networking

Thank You!