VxLan and Multicast

Roopa Prabhu, Nikolay Aleksandrov
Cumulus Networks
Linux Plumbers, 2019

Agenda

- VxLAN and Flooding
- VxLAN and multicast
- State of VxLAN with Multicast for Flooding
- Limitations to current VxLan and multicast support
- Fixes and Futures

Terminologies

- VTEP - VxLan Termination End Point
- BUM flooding - Broadcast, Unknown unicast and Multicast flooding
- PIM - Protocol independent multicast (multicast control plane protocol)
- ipmr - IP multicast routing. Kernel net/ipv[46]/ipmr*
- OIL - Outgoing interface list
- OIF - Outgoing interface
- FDB - layer 2 forwarding database
- E-VPN - ethernet VPN control plane (with VxLAN as the data plane)

VxLAN Overlays

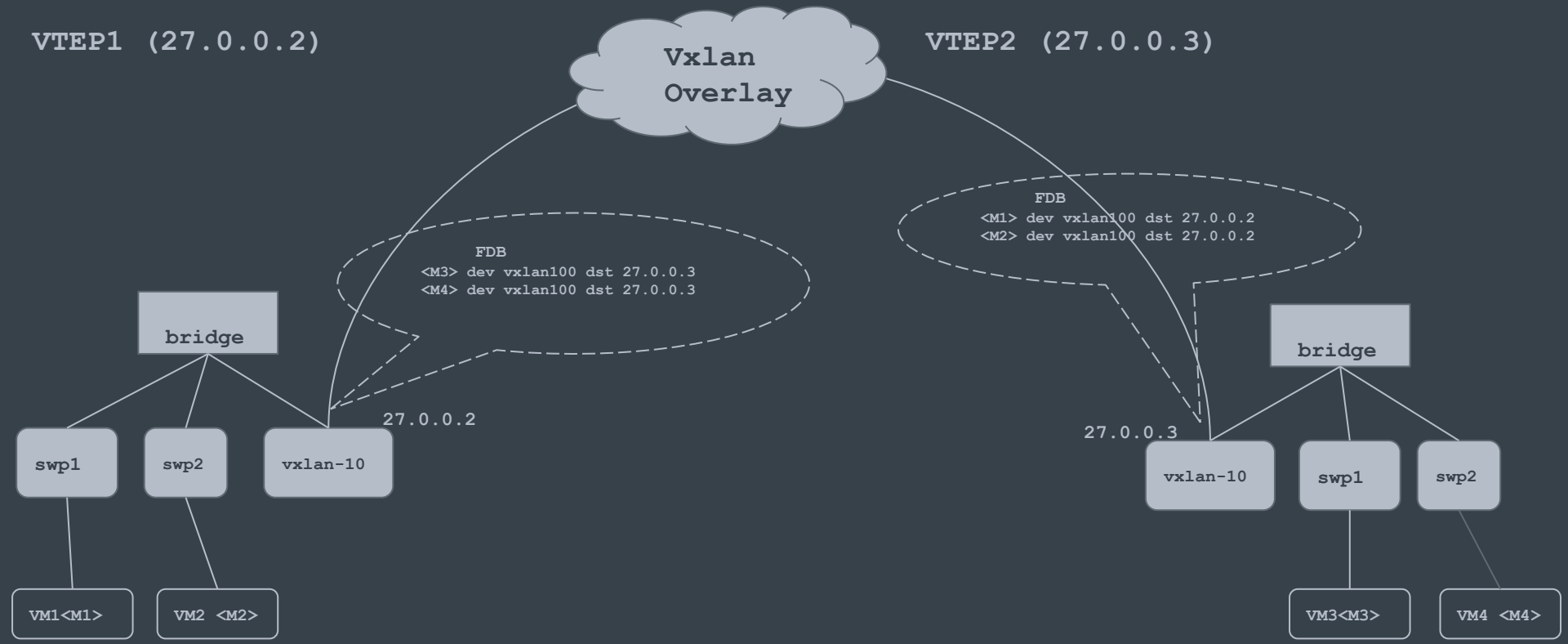
- L2 and L3 traffic encapsulated in VxLAN headers
- VTEP's are VxLAN tunnel endpoints: initiate and terminate VxLAN tunnels
- VTEPs can be deployed:
  - On the Host/Hypervisor/container-OS/Cloud etc
  - HW accelerated VTEP on the Top-Of-The-Rack switch

VxLAN L2 Overlays forwarding information

- Flood and learn (most basic case)
  - Flooded BUM traffic is carried across the multicast underlay
- End point Orchestrator/provisioning controller based FDB programming
- Control plane learning:
  - Local or distributed
- FDB: [**<Mac> <vni> <dst_port> <dst_ip>**]

  dst_ip is remote VTEP ip

VTEPs connecting L2 segments



Linux kernel VxLAN and Flooding

Flooding of Broadcast, Unknown unicast and Multicast traffic:

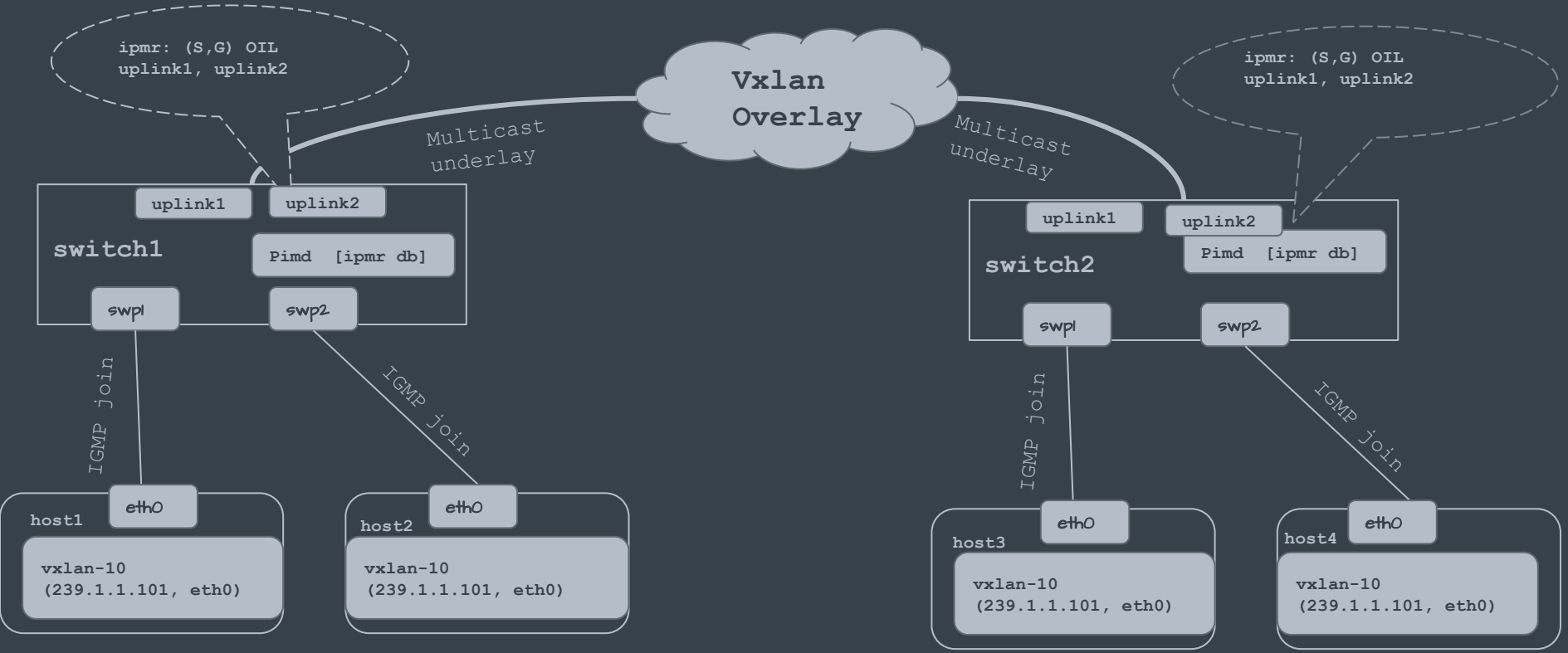
- VxLAN FDB flood entries are all-zero MAC FDB entries
- VxLAN driver supports:
  - Head End Replication by allowing multiple all-zero MAC FDB entries to remote unicast VTEP ips
    - Replication at Source VTEP
    - Eg (bridge fdb show)
      - 00:00:00:00:00:00 dev vxlan-10 dst 27.0.0.3
      - 00:00:00:00:00:00 dev vxlan-10 dst 27.0.0.4
  - Multicast Replication by allowing a single all-zero MAC FDB entry with multicast group
    - Replication at Destination VTEP (optimized)
    - Eg (bridge fdb show)
      - 00:00:00:00:00:00 dev vxlan100 dst 239.1.1.101
- Optimized flooding helps scale the overlay network

VxLAN with multicast underlay

- VTEPs source and receive multicast traffic:
  - They are both source and receivers in a multicast distribution tree
- IGMP [1] is used by VTEP to join the multicast distribution tree (for receive)
- VTEPs on routers will use underlay ip multicast routing to route originated multicast traffic
- PIM [2] is used between routers to set up the multicast distribution tree

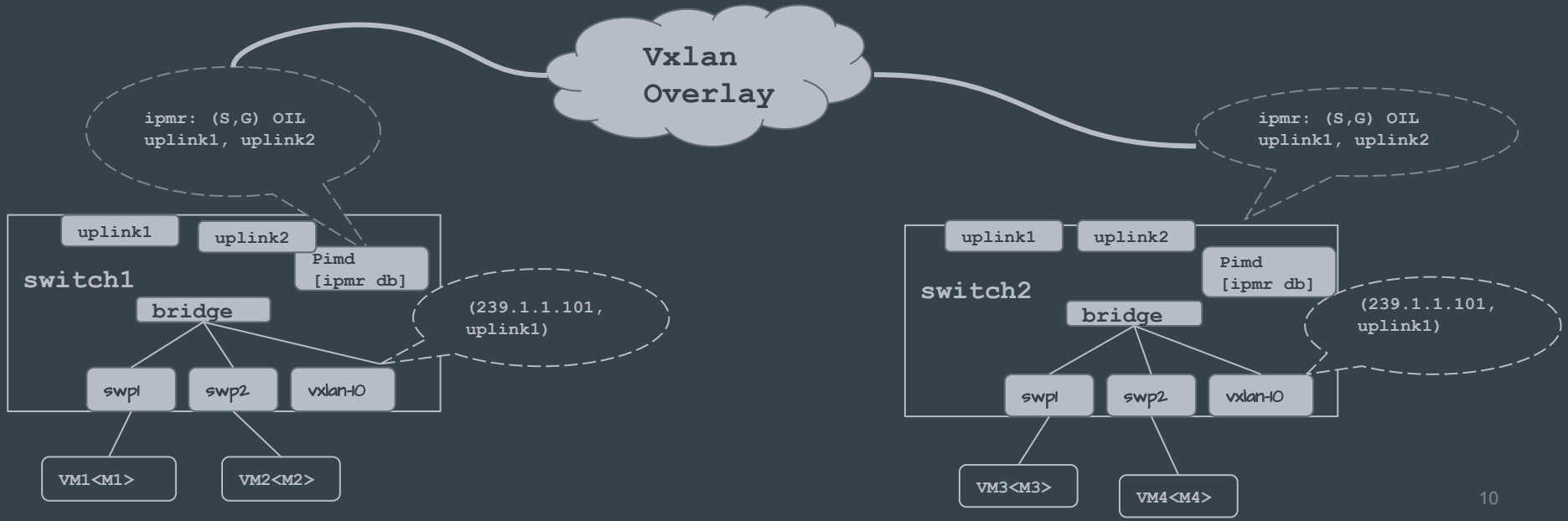
VxLAN with multicast underlay (Contd)

- VTEP on the host (VxLAN termination on the host)



VxLAN with multicast underlay (Contd)

- **VTEP on the switch today:**
  - VxLAN driver is configured with the OIF for the multicast group
  - Ipmr knows the OIL
  - Note that Ipmr and VxLAN driver are not in sync on the OIL

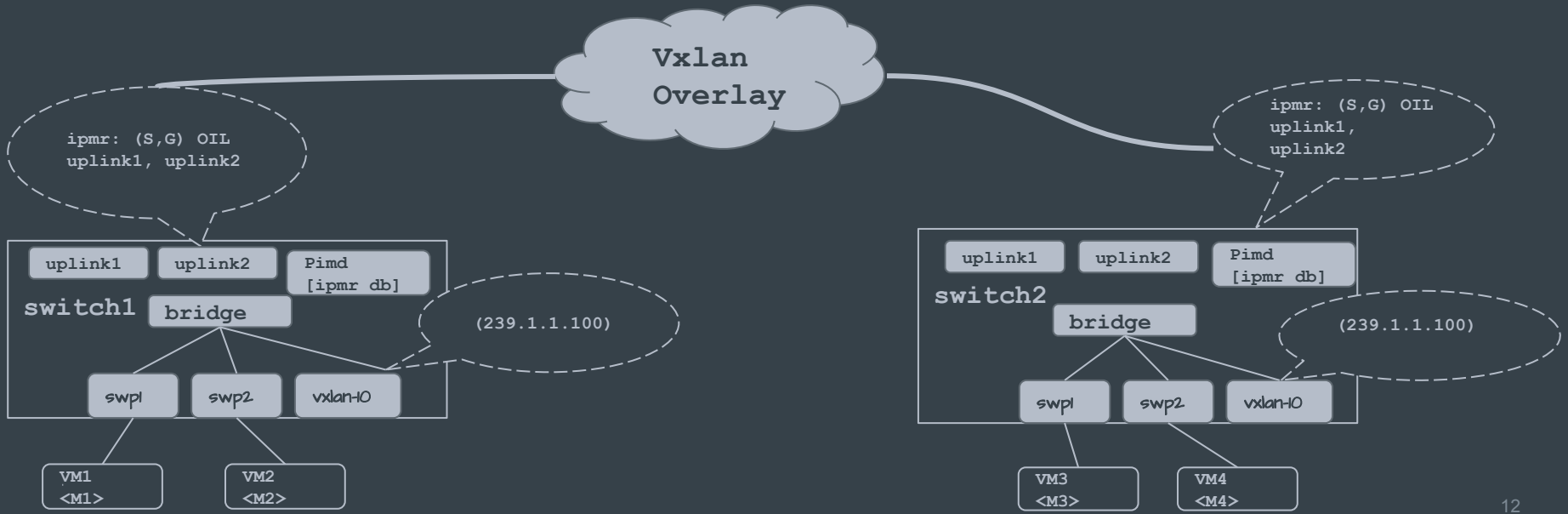


Linux Kernel VxLAN with multicast underlay

- Mostly designed for the Host VTEP case
- API:
  - Multicast group + OIF
- No multicast routing (ipmr) lookup performed for generated multicast traffic: OIF is used on TX
- Essentially supports only static multicast OIL
- Does not work in cases where VTEPs are located on a multicast router:
  - Example with PIM control plane

VxLAN with multicast underlay (Contd)

- **VTEP on the switch ideal (this is what we want to get to)**
  - VxLAN driver only knows the multicast group to replicate to
  - Ipmr knows the exact OIL to replicate the packet to



Multicast Routing

- Multicast IP Routing protocols are used to distribute data to multiple recipients

- A source can send a single copy of data to a single multicast address, which is then distributed to an entire group of recipients

- Protocol independent multicast (PIM) [2] - is a multicast control plane that advertises multicast sources and receivers over a routed layer 3 network

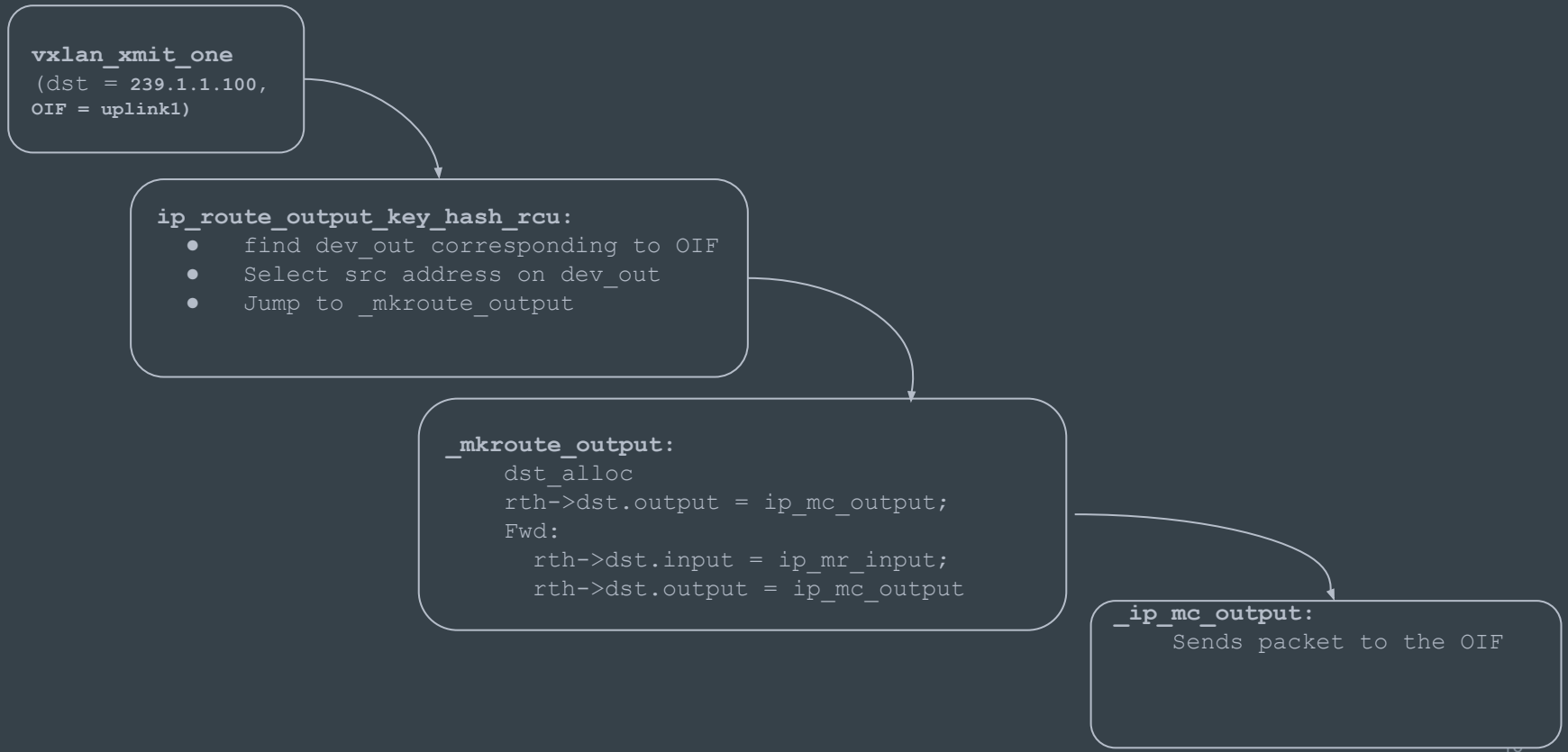
Linux kernel multicast routing

- Code: net/ipv[4,6]/ipmr.c
- Received ip multicast packets get into ip_mr_input:
  - ip_mr_fwd and/or local receive
- Locally generated multicast packets hit ip_mc_output

Linux Kernel locally generated multicast packets

- Locally generated multicast packets don’t go through a multicast routing lookup
- If fl4->flowi4_oif is set and the packet is multicast
  - It will find saddr and jump directly to _mkroute_output
- __mkroute_output: sets dst output to ip_mc_output
- There is ip_mr_input but no equivalent ip_mr_output
  - ip_mc_output directly xmits out of the set OIF

VxLAN multicast kernel TX pkt flow



Problems for VxLAN today

- Does not work with Dynamic Multicast routing control plane for originated multicast traffic on the same VTEP
- Dynamic multicast routing is needed for:
  - Distribution across multiple paths
  - Multihoming:
    - In cases where uplinks are down, multicast control plane with Multihoming capability can re-route multicast traffic via peer switches

Need for ip_mr_output

- Use multicast routing (ipmr) for locally generated multicast packets
- Primary use-case is VxLAN multicast underlay:
  - Locally generated VxLAN encap packets need lookup in ipmr table to route

VxLAN driver changes to use ipmr

- Multicast receive: we still need the dev/OIF to express interest in the multicast group via IGMP (IGMP join)

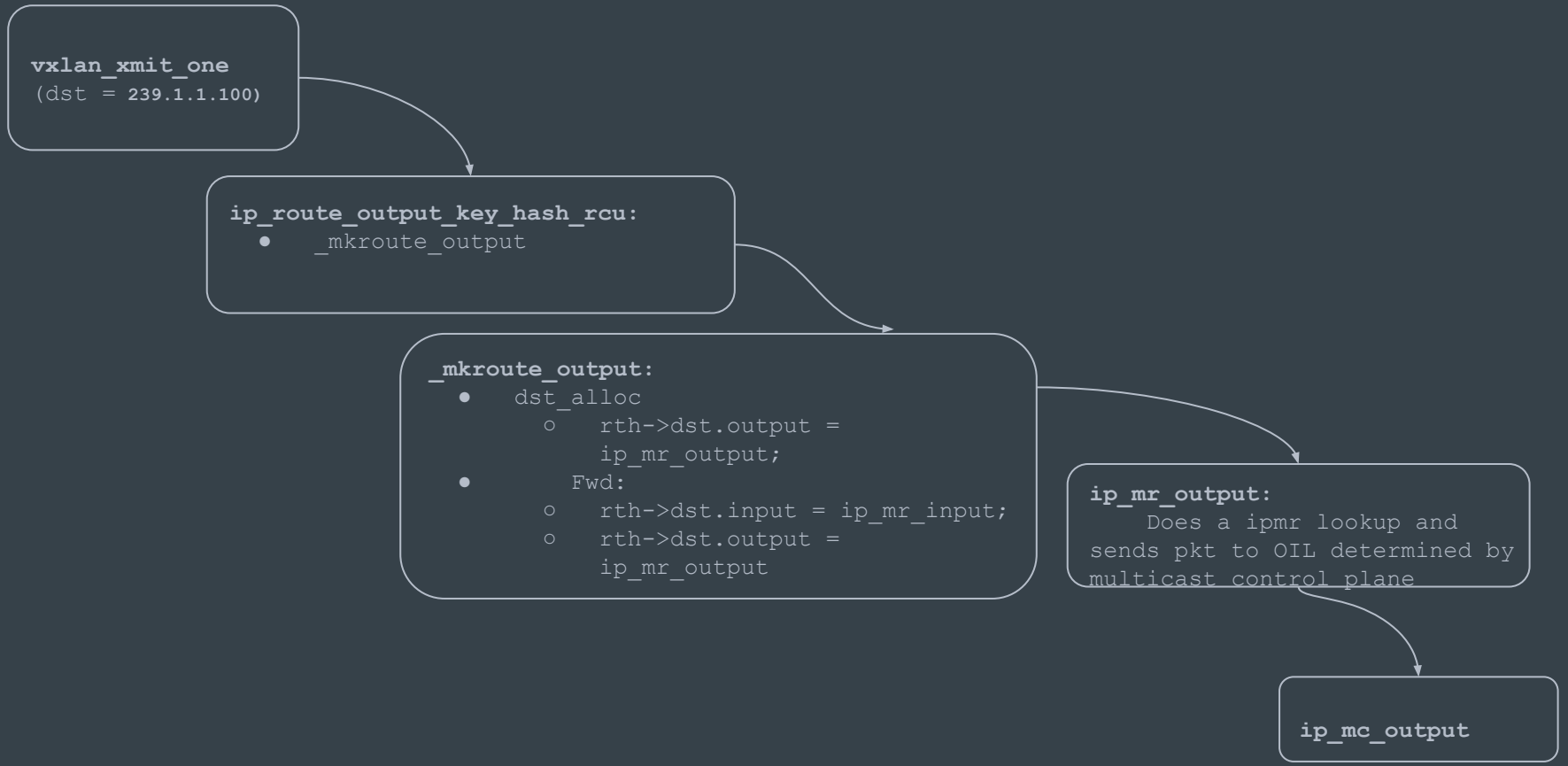
- Multicast transmit: new VxLAN dev flag VXLAN_F_USE_IPMR to not use the dev/OIF in route lookups (This will ensure ip_mr_output gets the right OIL)

Patches

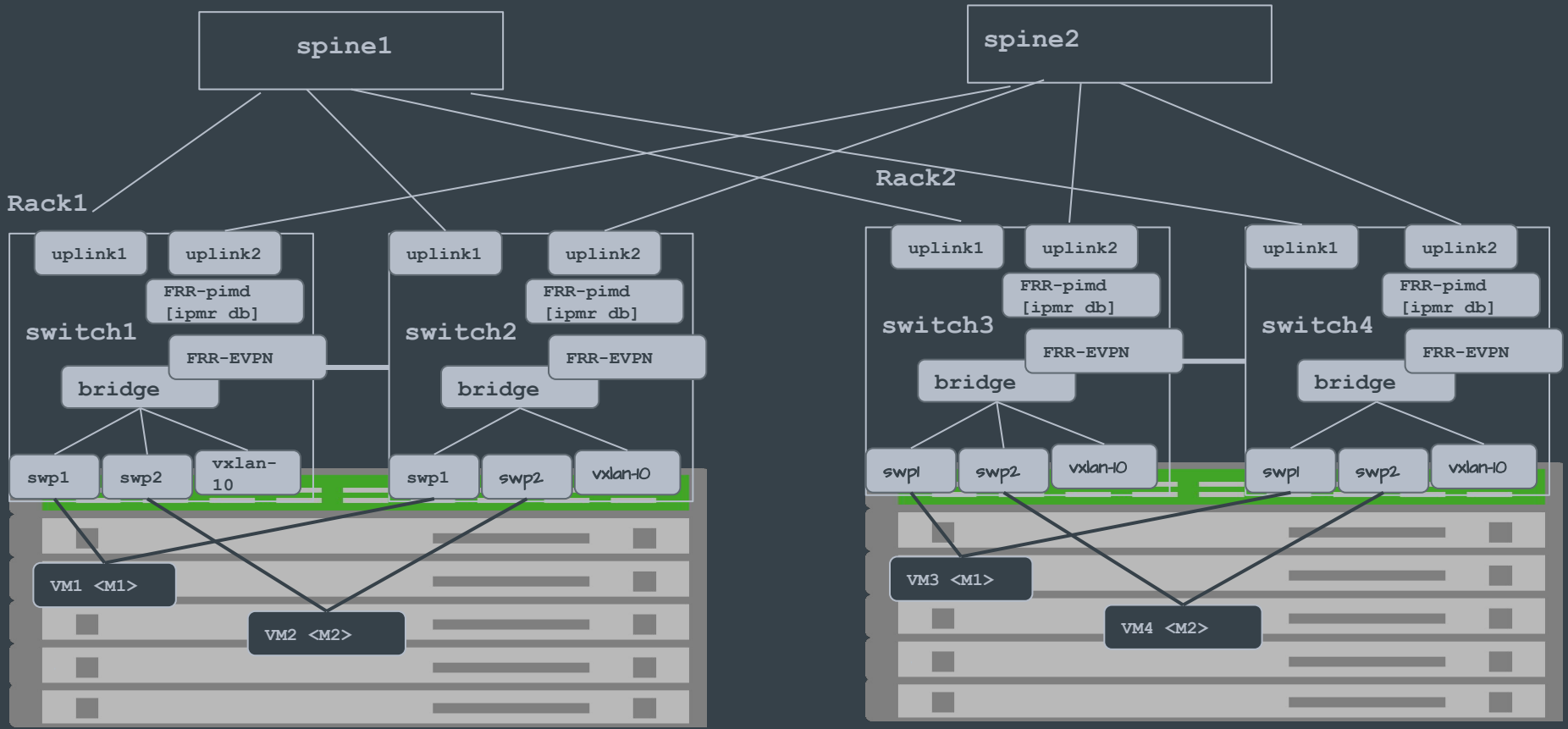
Tree: https://github.com/CumulusNetworks/net-next branch ipmr-vxlan

- Patches waiting to be upstreamed:
  - Needs more IPv6 VxLAN underlay testing
  - Make sure we don't break any existing expectations from multicast apps:
    - We try to fallback to old ip_mc_output when ip_mr_forward fails to find the OIL

VxLAN multicast TX kernel new pkt flow



Bigger Picture



Futures:

New multicast optimizations via control plane in a VxLAN fabric:

- Selective Multicast: IGMP multicast group information propagation using E-VPN protocol [6]
- IGMP/MLD proxy: similar to ARP proxy [7]

References

[1] IGMP https://tools.ietf.org/html/rfc2236

[2] PIM https://tools.ietf.org/html/rfc7761

[3] Nice VxLAN multicast blog: https://vincent.bernat.ch/en/blog/2017-vxlan-linux

[4] EVPN RFC: https://tools.ietf.org/html/rfc8365

[5] EVPN multicast forwarding: https://tools.ietf.org/html/draft-lin-bess-evpn-irb-mc

[6] EVPN selective multicast: https://tools.ietf.org/html/draft-zhang-l2vpn-evpn-selective-mcast-00

[7] EVPN IGMP-MLD proxy https://tools.ietf.org/html/draft-ietf-bess-evpn-igmp-mld-proxy-03

[8] Free range routing (FRR): https://frrouting.org/

[9] Linux bridge, L2-Overlays and E-VPN: https://www.netdevconf.org/2.2/slides/prabhu-linuxbridge-tutorial.pdf

Thank you