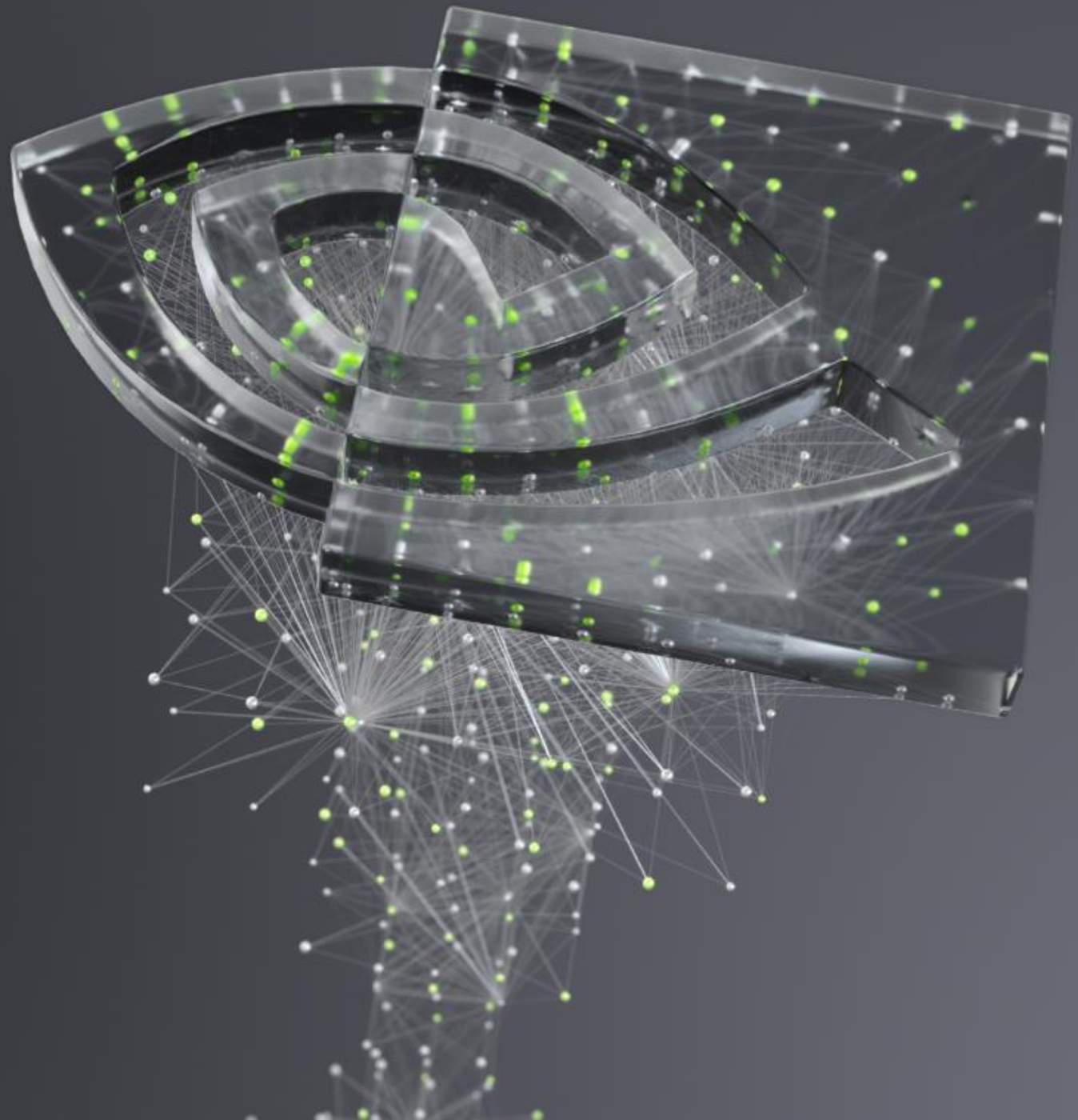


FORWARDING ENHANCEMENTS

Ido Schimmel, September 2024

AGENDA

Past Activities

Multipath hash seed

Transceiver firmware update

EVPN MH

Current Activities

DSCP matching

Future Activities

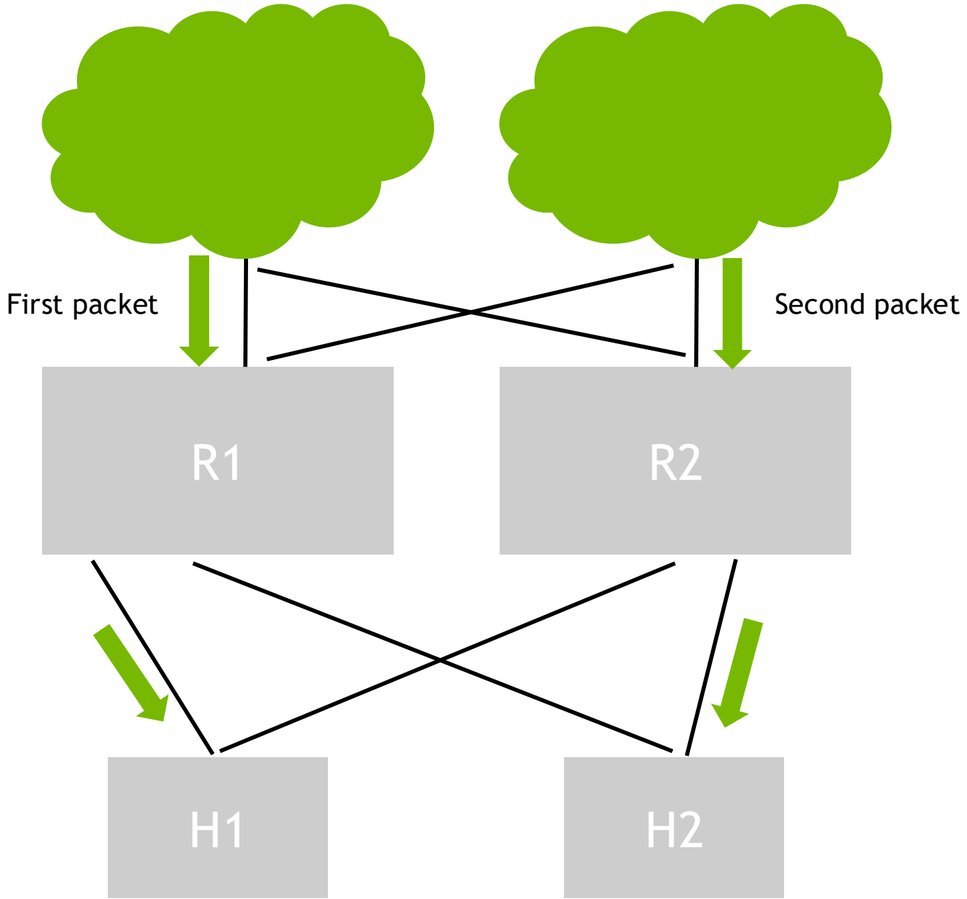
XDP metadata for telemetry

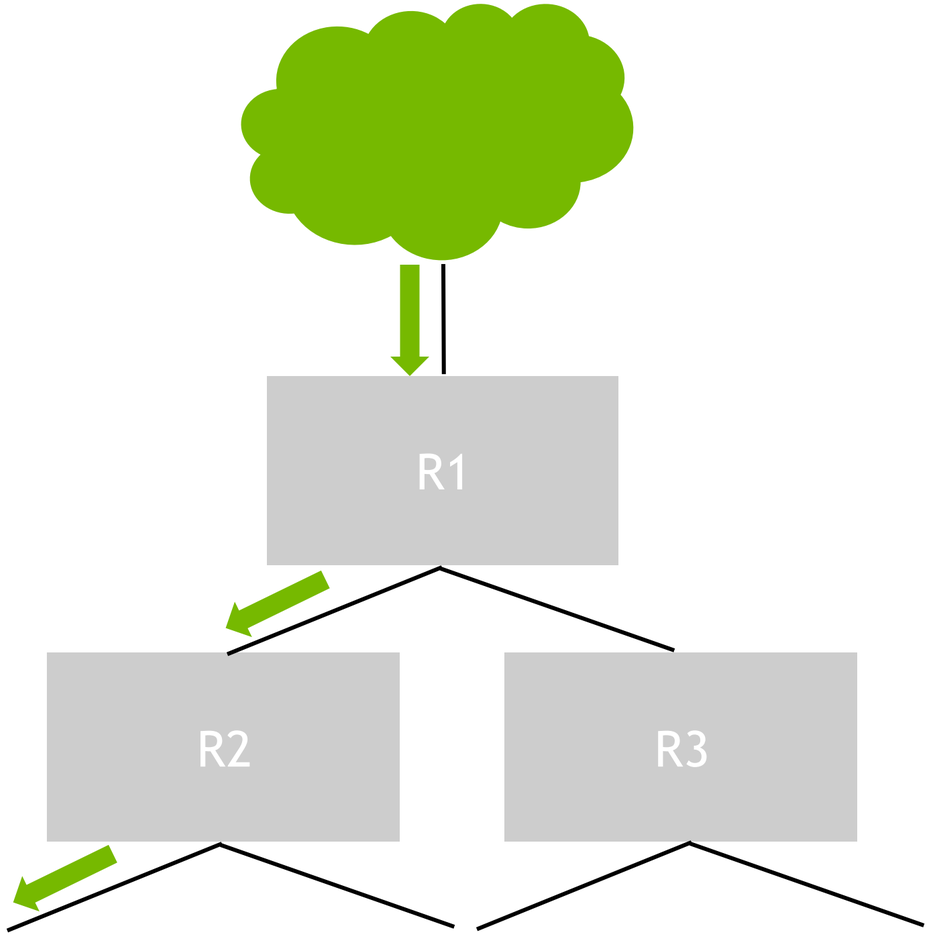
Multicast routing extensions

Drop reasons

PAST ACTIVITIES

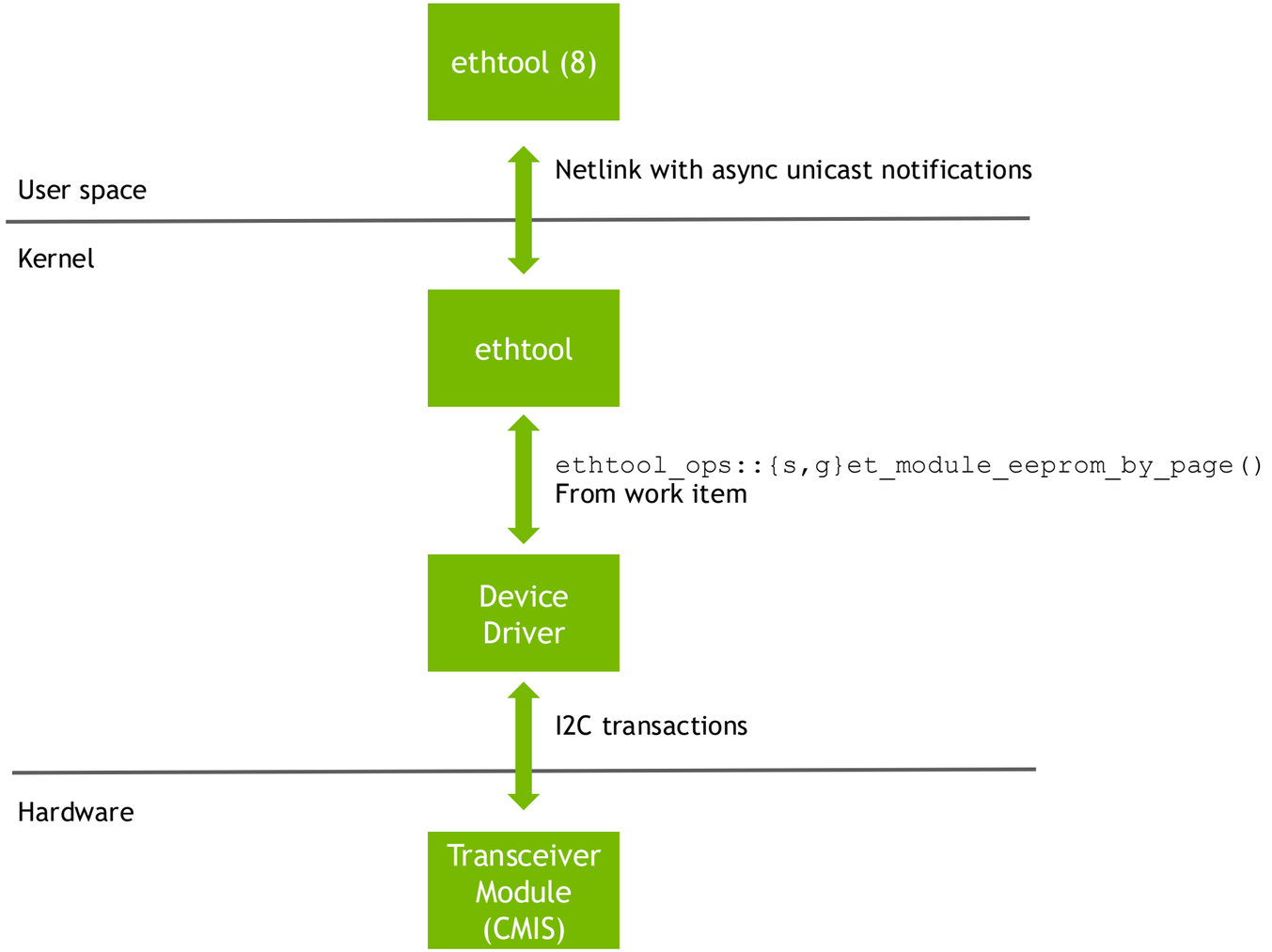
MULTIPATH HASH SEED





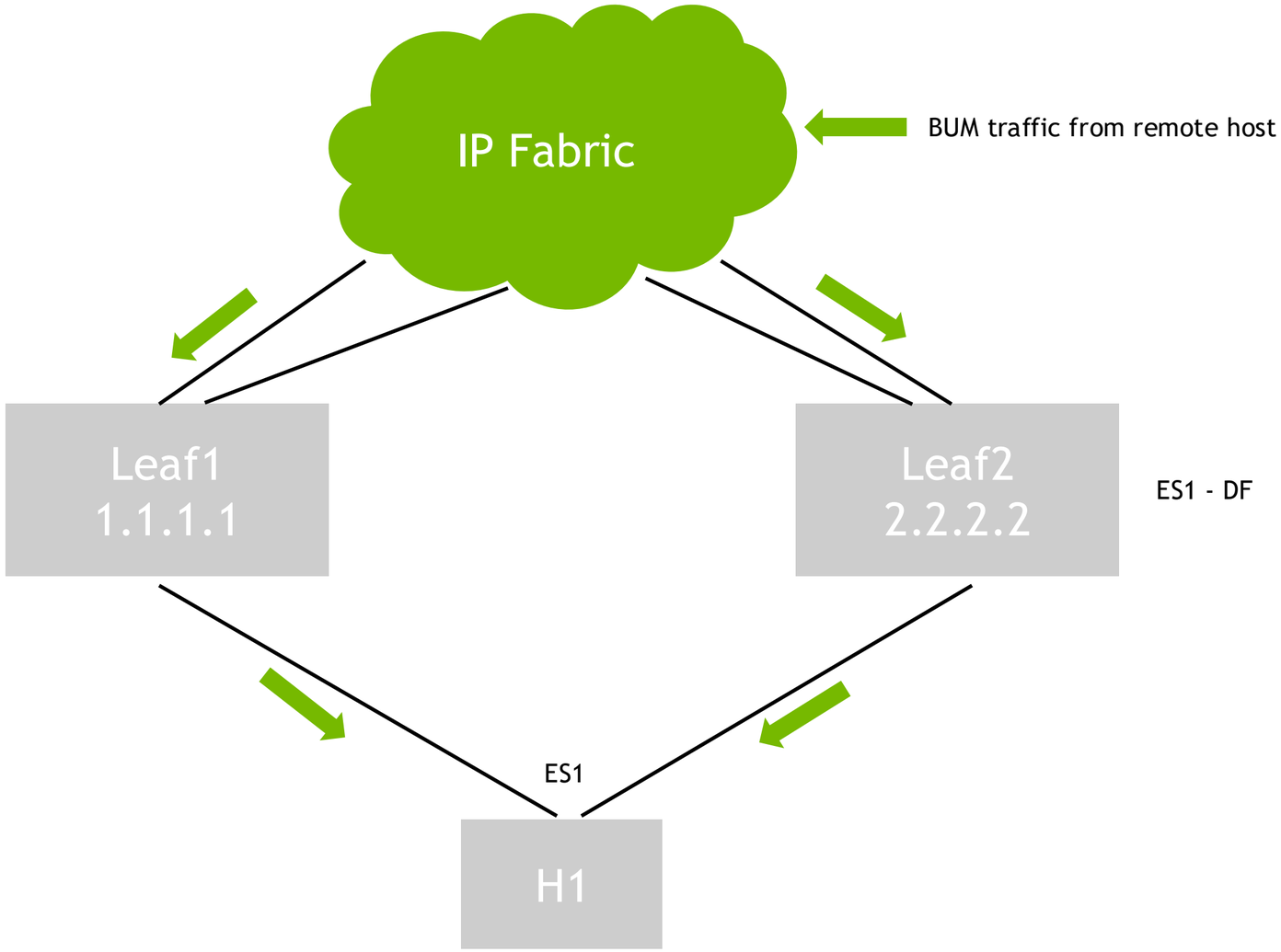
https://lore.kernel.org/all/20240607151357.421181-1-petrm@nvidia.com/

TRANSCEIVER MODULE FIRMWARE UPDATE



https://lore.kernel.org/all/20240627140857.1398100-1-danieller@nvidia.com/

EVPN MH – NON-DF FILTERING



https://lore.kernel.org/netdev/20230529114835.372140-1-idosch@nvidia.com/

CURRENT ACTIVITIES

DSCP MATCHING

Background

- Type of service (ToS) field in the IPv4 header was redefined over the years
- Differentiated services code point (DSCP) was introduced in RFC 2474 (1998)
- Explicit Congestion Notification (ECN) was introduced in RFC 3168 (2001)
- IPv4 stack is mostly using old macros:
  - `IPTOS_RT_MASK`: RFC 791 (1981) - `000xxx00`
  - `RT_TOS()`: RFC 1349 (1992) - `000xxxx0`
- Additional info: "Untangling DSCP, TOS and ECN bits in the kernel", Guillaume Nault, Red Hat, LPC 2021

DSCP MATCHING

Problem

- FIB lookup requires an IPv4 flow key (`struct flowi4`)
- Most callers mask upper DSCP bits when initializing TOS field in the key (`flowi4_tos`)
- Therefore, the kernel rejects IPv4 FIB rules that match on those bits

```
# ip -4 rule add tos 0x1c table 100
# ip -4 rule add tos 0x3c table 100
Error: Invalid tos.
```

- Impossible to redirect traffic to a routing table based on DSCP
- Yet another difference between IPv4 and IPv6

DSCP MATCHING

Solution

- Fix remaining bugs regarding masking of ECN bits
- Align two callers to mask upper DSCP bits
- Move masking of upper DSCP bits to the core
- Unmask upper DSCP bits in all call sites
- Add new FIB rule DSCP selector

```
# ip -4 rule add dscp 63 table 100
# ip -6 rule add dscp 63 table 100
```

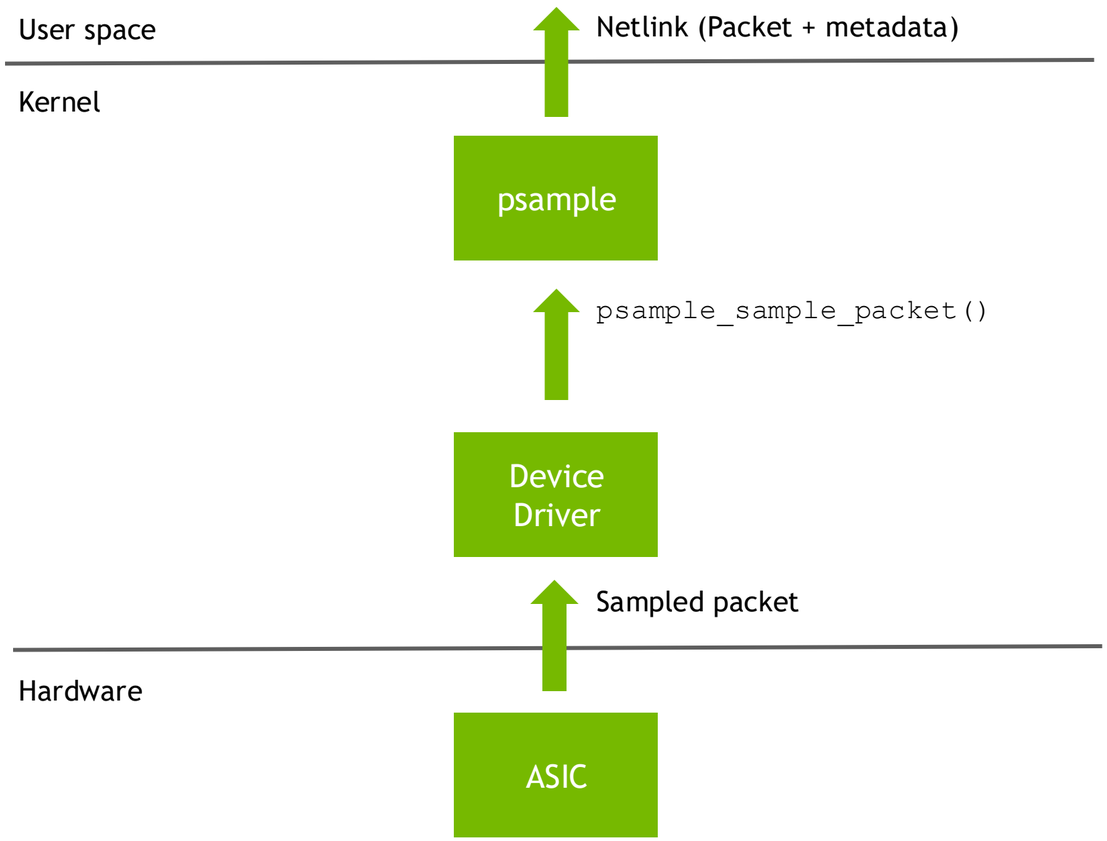
- Future work (?): Convert '`u8 flowi4_tos`' to '`dscp_t dscp`'

FUTURE ACTIVITIES

XDP METADATA FOR TELEMETRY

Packet sampling today

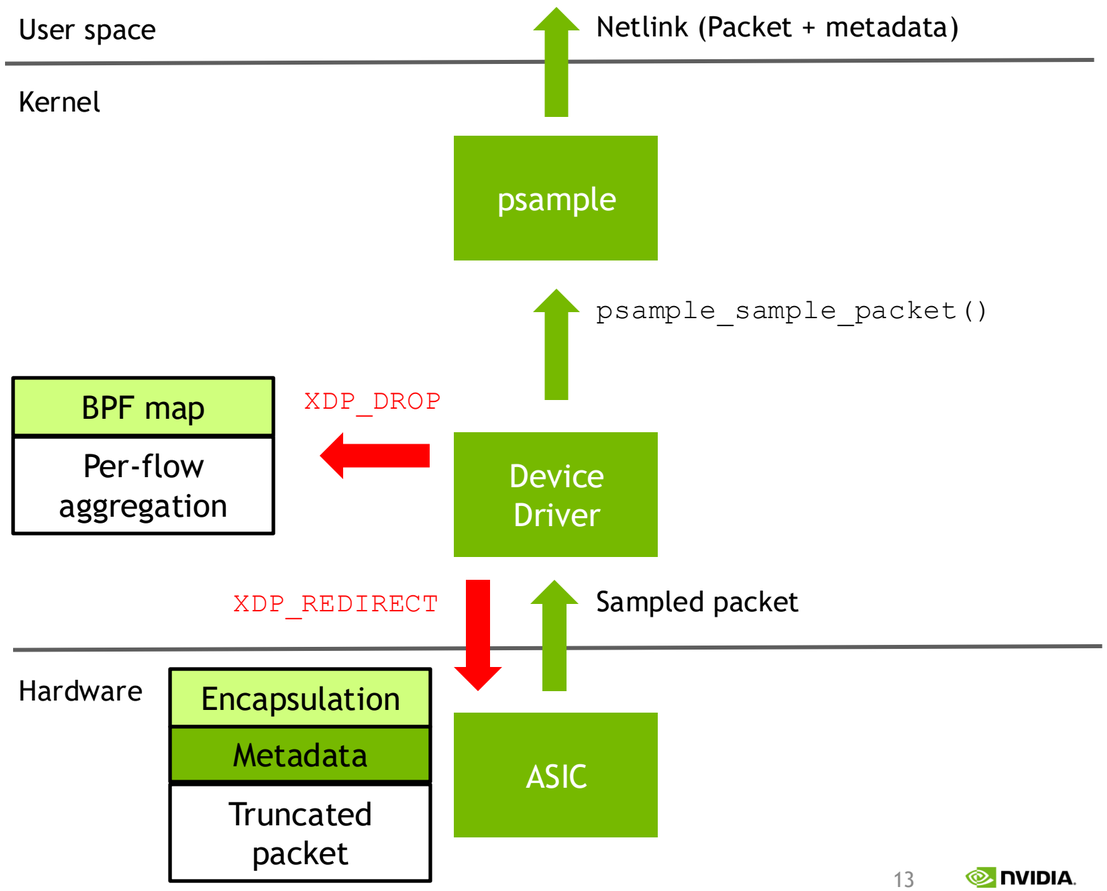
- Control plane via `tc`
- Data plane via `psample` generic netlink family notifications
- Metadata is encoded in netlink attributes. Examples:
  - `PSAMPLE_ATTR_IIFINDEX`
  - `PSAMPLE_ATTR_OIFINDEX`
  - `PSAMPLE_ATTR_OUT_TC`
  - `PSAMPLE_ATTR_OUT_TC_OCC`
  - `PSAMPLE_ATTR_LATENCY`



XDP METADATA FOR TELEMETRY

Packet sampling with XDP

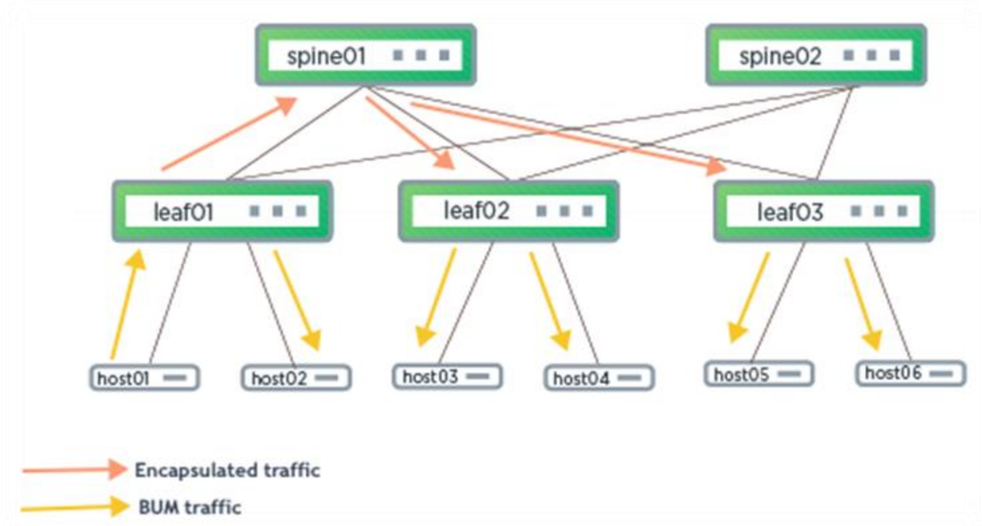
- Control plane still via `tc`
- Expose new metadata via new `xdp_metadata_ops`
- Enables new use cases with better performance. Examples:
  - `XDP_DROP`: Per-flow aggregation
  - `XDP_REDIRECT`: Route towards aggregation server with custom header



MULTICAST ROUTING EXTENSIONS

Routing locally generated traffic

- Unlike unicast, locally generated multicast packets are not routed
- Packets are transmitted via output interface specified in the flow key
- Problematic for VXLAN deployments that use multicast for BUM
  - Output interface can change over time
  - There can be a list of output interfaces



MULTICAST ROUTING EXTENSIONS

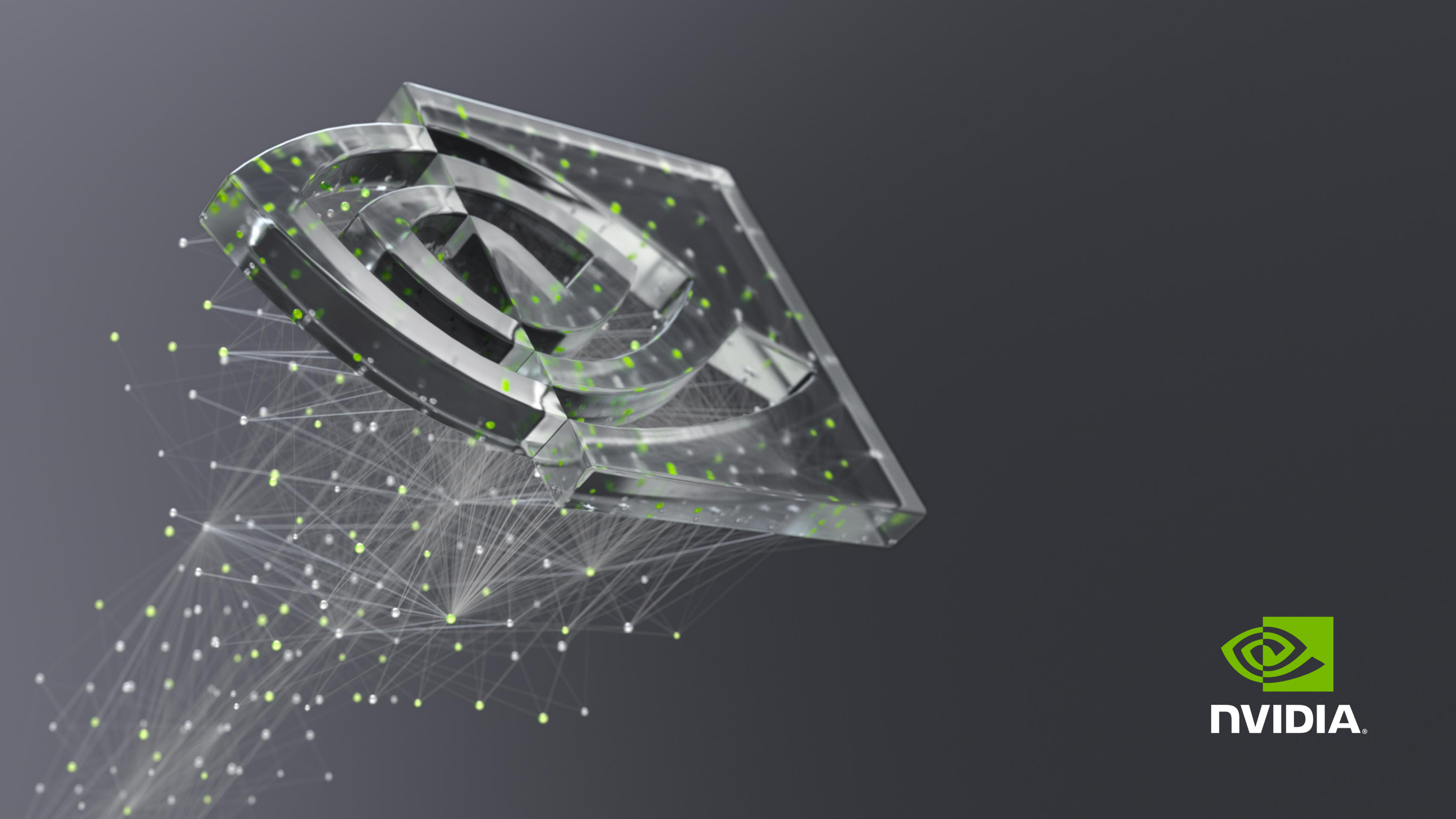
Routing locally generated traffic

- For both VXLAN FDB and MDB we would like to only specify the multicast destination address
- Output interface list will be determined by chosen multicast route
- Make new behavior opt-in to avoid behavior change
  - New flow key flag to request multicast routing
  - New VXLAN device knob
  - New socket option / control message for user space socket (if / when needed)
- Additional info: "VxLan and Multicast", Roopa Prabhu, Cumulus, LPC 2019

DROP REASONS

Guidelines

- Useful for observability, but what are the guidelines?
- Creating a new "subsystem" (e.g., mac80211, openvswitch) vs "core"
- Generic as possible vs specific as possible
- Do we annotate every drop or leave obscure ones as not specified?
- Can we rename / merge reasons when needed (affects tracers)?


NVIDIA