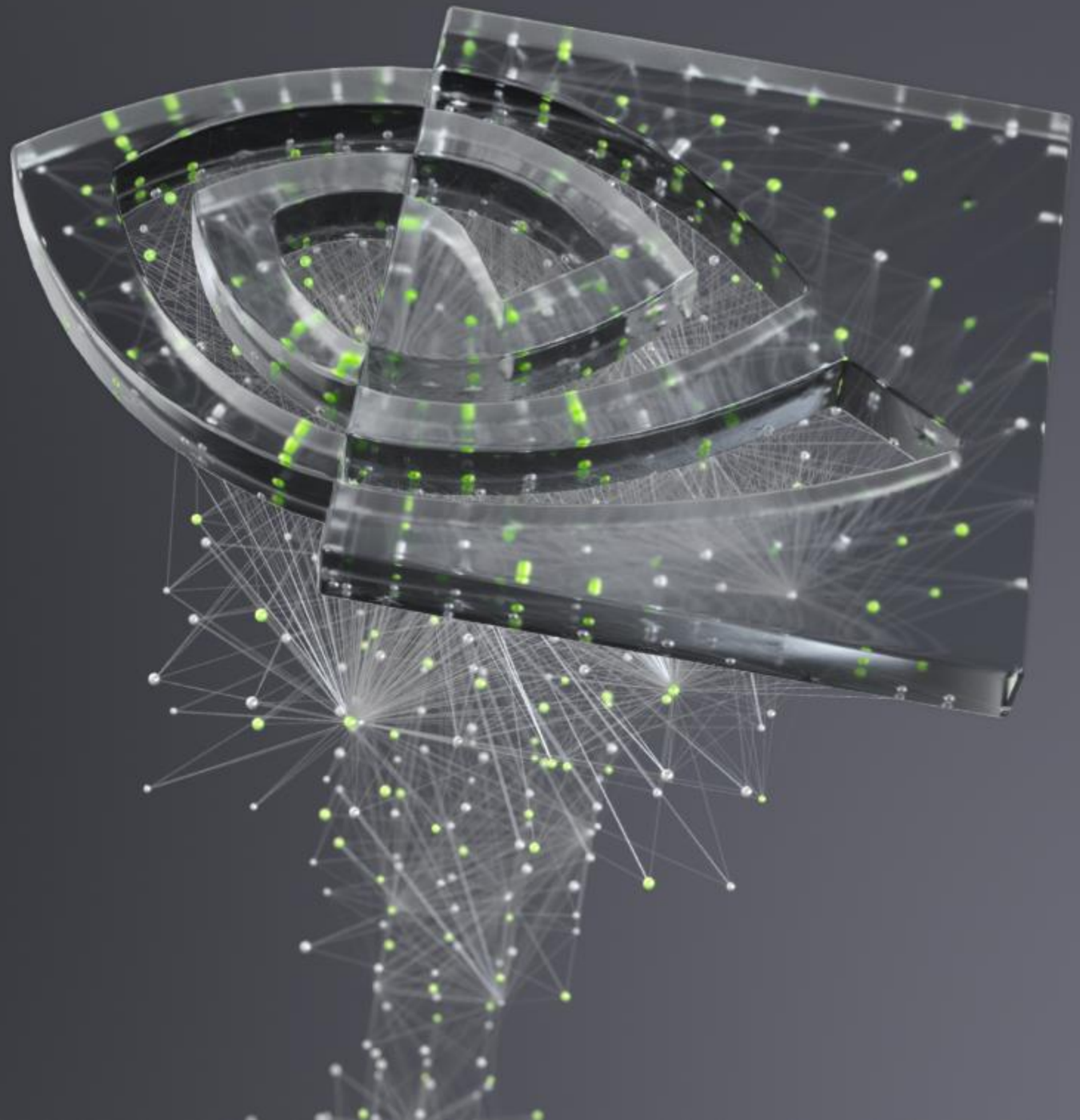


MLXSW UPDATES

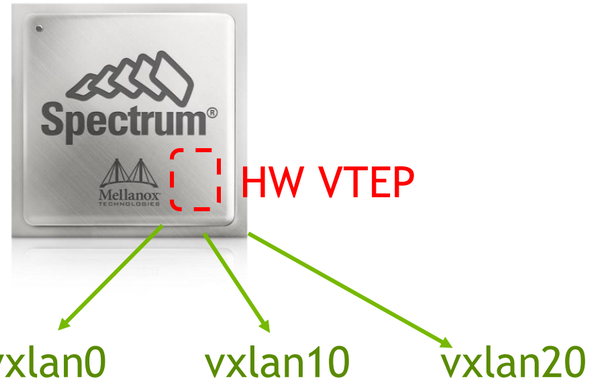
August 2020

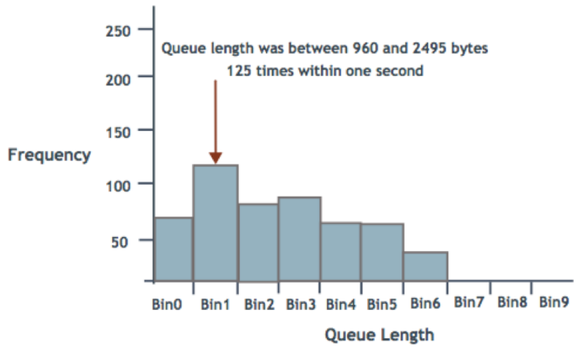
PLANNED FEATURES

DEVICE METRICS

- Netdev-centric metrics (rtnetlink / ethtool)
- Not configurable (e.g., enable / disable, histograms)
- Hardware-specific metrics, not mapped to software objects

Algorithmic TCAM





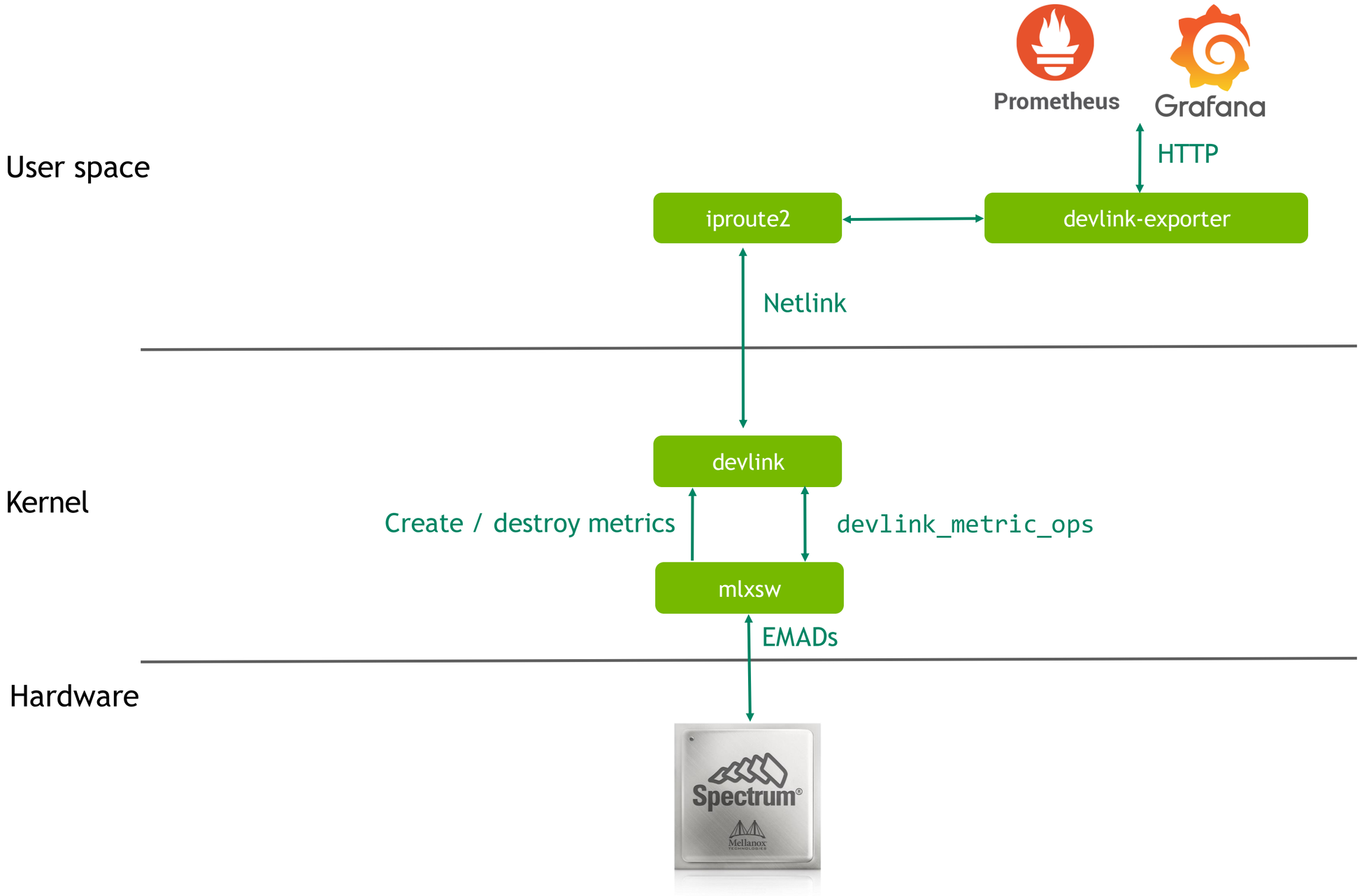
DEVICE METRICS (CONT)

- Debugfs is not an option:
  - Driver-specific (code duplication)
  - Not a stable interface
  - Not acceptable upstream

> Frankly, all of this debugfs crap in the DSA drivers smells like poo.
> I don't like it _AT_ _ALL_, and I shouldn't have allowed any of it
> into the tree in the first place.
>
> I might just remove it all myself, it bothers me so much.
>
> Fetching information should be done by well typed, generic, interfaces
> that apply to any similar device or object. All of this debugfs stuff
> smells of hacks and special case crap that's only usable for one
> device type and that makes it the single most terrible interface to
> give to users.

David S. Miller, July 2015, https://lkml.org/lkml/2015/7/11/8

DEVICE METRICS – PROPOSED SOLUTION



DEVICE METRICS - PROPOSED INTERFACE

Current interface:

```
devlink [-s] dev metric show [ DEV metric METRIC | group GROUP ]
devlink dev metric set DEV metric METRIC [ group GROUP ]
```

Future extensions (bold):

```
devlink dev metric set DEV metric METRIC [ group GROUP ]
              [ enable { true | false } ] [ hist_type { linear | exp } ]
              [ hist_min MIN ] [ hist_max MAX ] [ hist_buckets BUCKETS ]
              [ hist_sample_interval SAMPLE ]

devlink [-s] port metric show [ DEV/PORT_INDEX metric METRIC | group GROUP ]
devlink port metric set DEV/PORT_INDEX metric METRIC [ group GROUP ]
              [ enable { true | false } ] [ hist_type { linear | exp } ]
              [ hist_min MIN ] [ hist_max MAX ] [ hist_buckets BUCKETS ]
              [ hist_sample_interval SAMPLE ]
```

DEVICE METRICS - PROPOSED INTERFACE

```
[root@r-mgtswd-262 ~]# devlink -s dev metric show
pci/0000:01:00.0:
   metric nve_vxlan_encap type counter group 4 value 97287
   metric nve_vxlan_decap type counter group 0 value 851051
   metric nve_vxlan_decap_errors type counter group 0 value 753747
   metric nve_vxlan_decap_discards type counter group 0 value 0
[root@r-mgtswd-262 ~]#
[root@r-mgtswd-262 ~]# devlink dev metric show pci/0000:01:00.0 metric nve_vxlan_encap
pci/0000:01:00.0:
   metric nve_vxlan_encap type counter group 4
[root@r-mgtswd-262 ~]#
[root@r-mgtswd-262 ~]# devlink dev metric set pci/0000:01:00.0 metric nve_vxlan_encap group 5
[root@r-mgtswd-262 ~]# devlink dev metric set pci/0000:01:00.0 metric nve_vxlan_decap group 5
[root@r-mgtswd-262 ~]#
[root@r-mgtswd-262 ~]# devlink -s dev metric show group 5
pci/0000:01:00.0:
   metric nve_vxlan_encap type counter group 5 value 97910
   metric nve_vxlan_decap type counter group 5 value 858002
```

Dump all existing metrics

Get a specific metric

Bind metrics to a group

Dump all metrics in a group

DEVICE METRICS - PROPOSED INTERFACE

Kenel documentation

Metrics

List of metrics registered by `mlxsw`

Name	Type	Supported platforms	Description
`nve_vxlan_encap`	`counter`	Spectrum-1 only	Counts number of packets that were VXLAN encapsulated by the device. A packet sent to multiple VTEPs is counted multiple times
`nve_vxlan_decap`	`counter`	Spectrum-1 only	Counts number of VXLAN packets that were decapsulated (successfully or otherwise) by the device
`nve_vxlan_decap_errors`	`counter`	Spectrum-1 only	Counts number of VXLAN packets that encountered decapsulation errors. This includes overlay packets with a VLAN tag, ECN mismatch between overlay and underlay, multicast overlay source MAC, overlay source MAC equals overlay destination MAC and packets too short to decapsulate
`nve_vxlan_decap_discards`	`counter`	All	Counts number of VXLAN packets that were discarded during decapsulation. In Spectrum-1 this includes packets that had to be VXLAN decapsulated when VXLAN decapsulation is disabled and fragmented overlay packets. In Spectrum-2 this includes `nve_vxlan_decap_errors` errors and a missing mapping between VNI and filtering identifier (FID)

RESILIENT HASHING

- The objective of resilient hashing is to minimize the impact on flows bound to unaffected nexthops when nexthops are added or deleted from a multipath group (e.g., ECMP)
- The multipath algorithm implemented in Linux (IPv4 & IPv6) is "Hash-Threshold" described in RFC 2992:

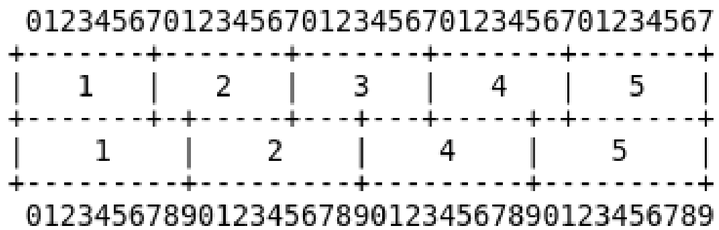
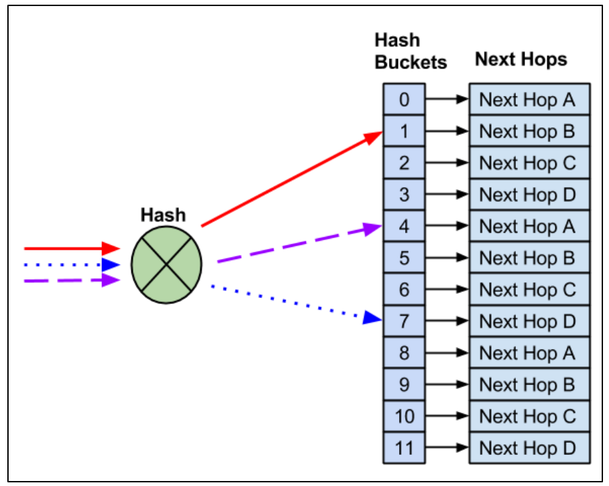


Figure 1. Before and after deletion of region 3

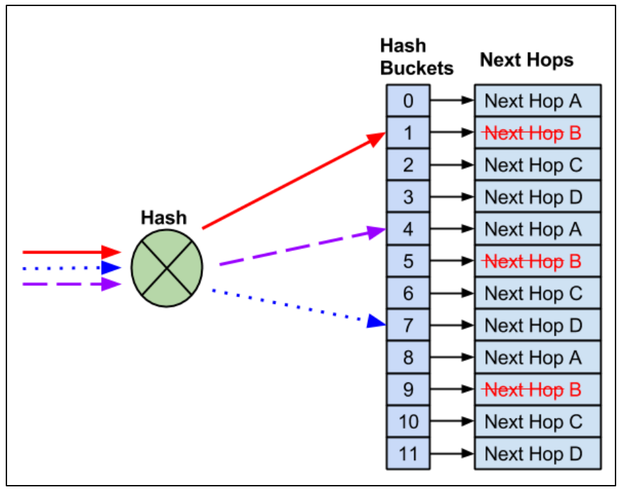
- Flows hashed to areas near region boundaries are remapped even if they were initially mapped to unaffected nexthops (regions)
- Another algorithm described in RFC 2992 is "Modulo-N". More disruptive than "Hash-Threshold".

RESILIENT HASHING (CONT)

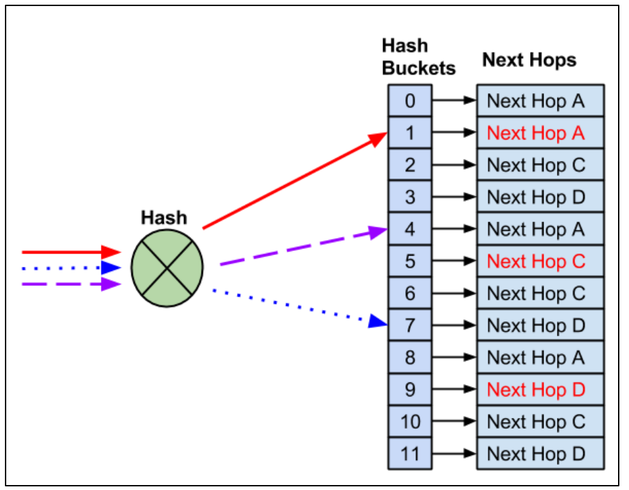
- Resilient hashing can be achieved by populating nexthops in a more sophisticated way
- Nexthop removal example:



t0: Initial state



t1: Nexthop B goes down

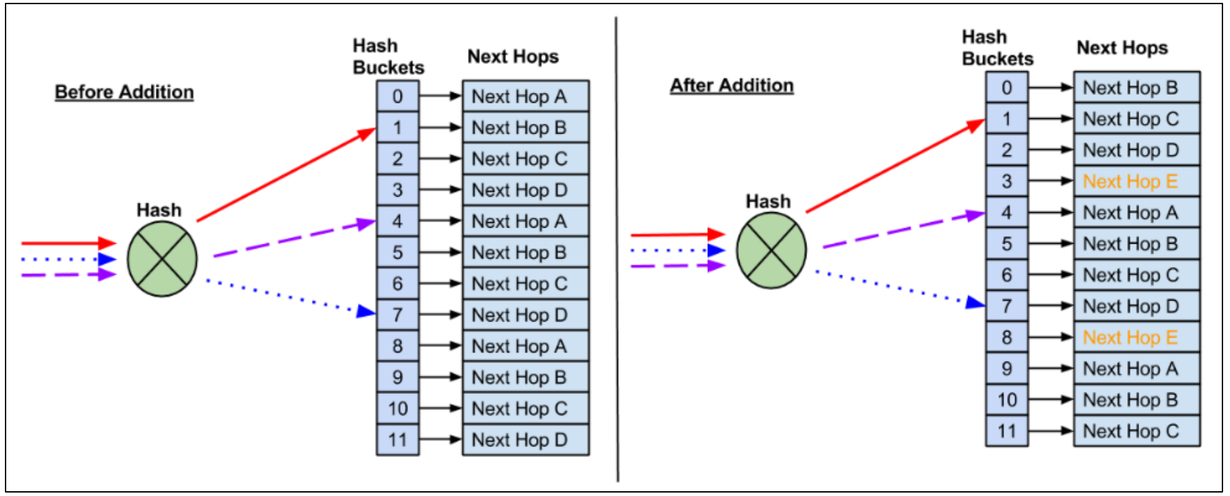


t2: Group rebalanced

- Flows mapped to unaffected nexthops are not impacted

RESILIENT HASHING (CONT)

- Nexthop addition example:



- To minimize impact, nexthop activity is taken into account in order to decide when and how to perform the replacement

RESILIENT HASHING (CONT)

- Resilient hashing can be achieved in the kernel's data path by using the nexthop API, which breaks out the management of nexthops from the routes bound to them
- Two proposals:
  - User space solution
  - Kernel solution

USER SPACE SOLUTION

- Nexthop IDs become hash buckets. Cannot be shared by multiple groups
- User space controls:
  - Number of buckets in a group
  - Mapping of logcial nexthops (gateway + device) to buckets
  - When and how to perform nexthops replacement
- Nexthop removal: Partially addressed by active-backup groups. RFC from David Ahern
- Nexthop addition: User space needs activity information from the kernel per nexthop ID (bucket)

USER SPACE SOLUTION (CONT)

- Initial state

```
id 101 group 1/2 active-backup
id 102 group 3/4 active-backup
id 103 group 5/6 active-backup
id 104 group 7/8 active-backup
id 105 group 9/10 active-backup
id 106 group 11/12 active-backup
id 107 group 13/14 active-backup
id 108 group 15/16 active-backup
id 109 group 17/18 active-backup
id 110 group 19/20 active-backup
id 111 group 21/22 active-backup
id 112 group 23/24 active-backup
id 10001 group 101/102/103/104/105/106/107/108/109/110/111/112
```

USER SPACE SOLUTION (CONT)

- After nexthop B was removed

```
id 101 group 1 active-backup
id 102 group 4 active-backup
id 103 group 5/6 active-backup
id 104 group 7/8 active-backup
id 105 group 9/10 active-backup
id 106 group 12 active-backup
id 107 group 13 active-backup
id 108 group 15 active-backup
id 109 group 17/18 active-backup
id 110 group 20 active-backup
id 111 group 21/22 active-backup
id 112 group 23/24 active-backup
id 10001 group 101/102/103/104/105/106/107/108/109/110/111/112
```

- Number of buckets did not change
- Does not work when multiple nexthops go down

USER SPACE SOLUTION (CONT)

- After nexthop E was added

```
id 101 group 1/2 active-backup
id 102 group 3/4 active-backup
id 103 group 5/6 active-backup
id 104 group 7/8 active-backup
id 105 group 9/10 active-backup
id 106 group 11/12 active-backup
id 107 group 13/14 active-backup
id 108 group 15/16 active-backup
id 109 group 17/18 active-backup
id 110 group 19/20 active-backup
id 111 group 21/22 active-backup
id 112 group 23/24 active-backup
id 10001 group 101/102/103/104/105/106/107/108/109/110/111/112
```

- Number of buckets did not change. Individual nexthops (IDs 1-24) were replaced

USER SPACE SOLUTION – ACTIVITY INDICATION

- A new nexthop should only be mapped to inactive buckets to minimize impact on active flows
- Possible race: By the time user space decides to perform the replacement, bucket can become active again
- Kernel needs to support atomic replacement
- Two options:
  - Activity flag
  - Used time

USER SPACE SOLUTION – ACTIVITY FLAG

- Each nexthop ID (bucket) reports a new active flag (e.g., RTNH_F_ACTIVE)

```
id 1 via 2.2.2.2 dev dummy_b scope link active
```

- Periodically queried and cleared by user space

```
ip nexthop list_clear
```

- New keyword is added to communicate an atomic replacement

```
ip nexthop replace atomic id 3 via 2.2.2.2 dev dummy_b
```

- Kernel will reject the replacement if provided nexthop ID has active flag set

USER SPACE SOLUTION – USED TIME

- Each nexthop ID (bucket) reports time since last used

```
id 1 via 2.2.2.2 dev dummy_b scope link used 5
```

- Cached by user space and used to perform an atomic replacement

```
ip nexthop replace used 5 id 3 via 2.2.2.2 dev dummy_b
```

- Kernel compares current used time with provided one. If the former is smaller, replacement is rejected

KERNEL SOLUTION – NEW GROUP TYPE

- Resilient hashing can be implemented in the kernel by adding a new group type (e.g., NEXTHOP_GRP_TYPE_RESILIENT)

```
Usage: ip nexthop { list | flush } [ protocol ID ] SELECTOR
       ip nexthop { add | replace | append } id ID NH [ protocol ID ]
       ip nexthop { get| del } id ID
SELECTOR := [ id ID ] [ dev DEV ] [ vrf NAME ] [ master DEV ]
            [ groups ]
NH := { blackhole | [ via ADDRESS ] [ dev DEV ] [ onlink ]
      [ encap ENCAPTYPE ENCAPHDR ] | [ group GROUP GROUPTYPE ]
      [ num_buckets NUM_BUCKETS ] [ resilient_hash_active_timer ACTIVE_TIMER ]
      [ resilient_hash_max_unbalanced_timer UNBALANCED_TIMER ] }
GROUP := [ id[,weight]>/<id[,weight]>/... ]
ENCAPTYPE := [ mpls ]
ENCAPHDR := [ MPLSLABEL ]
GROUPTYPE := { multipath | active-backup | multipath-resilient }
```

KERNEL SOLUTION (CONT)

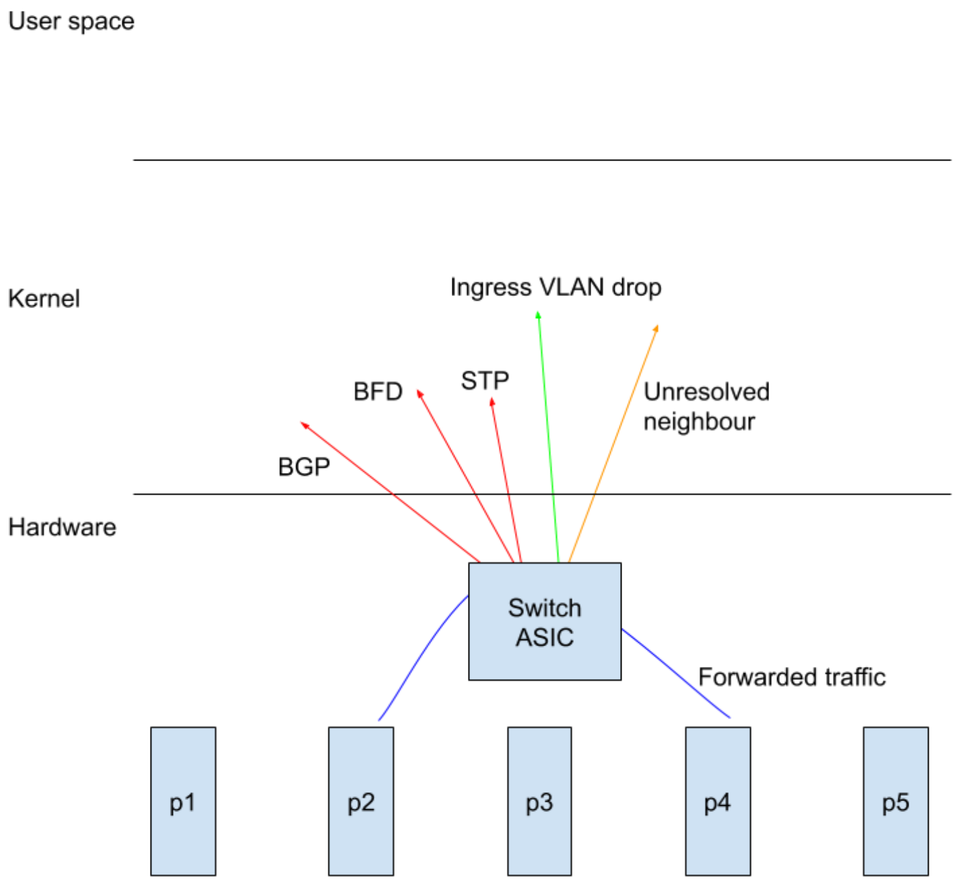
- New attributes:
  - Number of buckets: More buckets reduce impact when nexthop is added. When removed, nexthops are more evenly distributed
  - Active timer: When adding a new nexthop, wait for at least one hash bucket to be inactive for N seconds before performing the replacement
  - Unbalanced timer: Force a rebalance every N seconds
- More attributes required in order to dump buckets to user space. Necessary for testing and visibility
- Appending nexthops to a group?

RECENTLY ADDED FEATURES

CONTROL PLANE POLICING (COPP) - MOTIVATION

- Kernel's data path mirrored to capable hardware
- Hardware able to handle packet rates that are several order of magnitude higher compared to CPU
- Some packets still need to be trapped to the CPU:
  - Control: Required for the correct functioning of the control plane. For example, ARP request and IGMP query packets
  - Exceptions: Not forwarded as intended by the underlying device due to an exception (e.g., TTL error, missing neighbour entry). Need kernel intervention
  - Drops: Dropped by the underlying device. Trapped to the CPU for visibility
- Need to be able to rate limit trapped packets to ensure CPU is not overwhelmed and control plane remains functional

CONTROL PLANE POLICING (COPP) - ILLUSTRATION



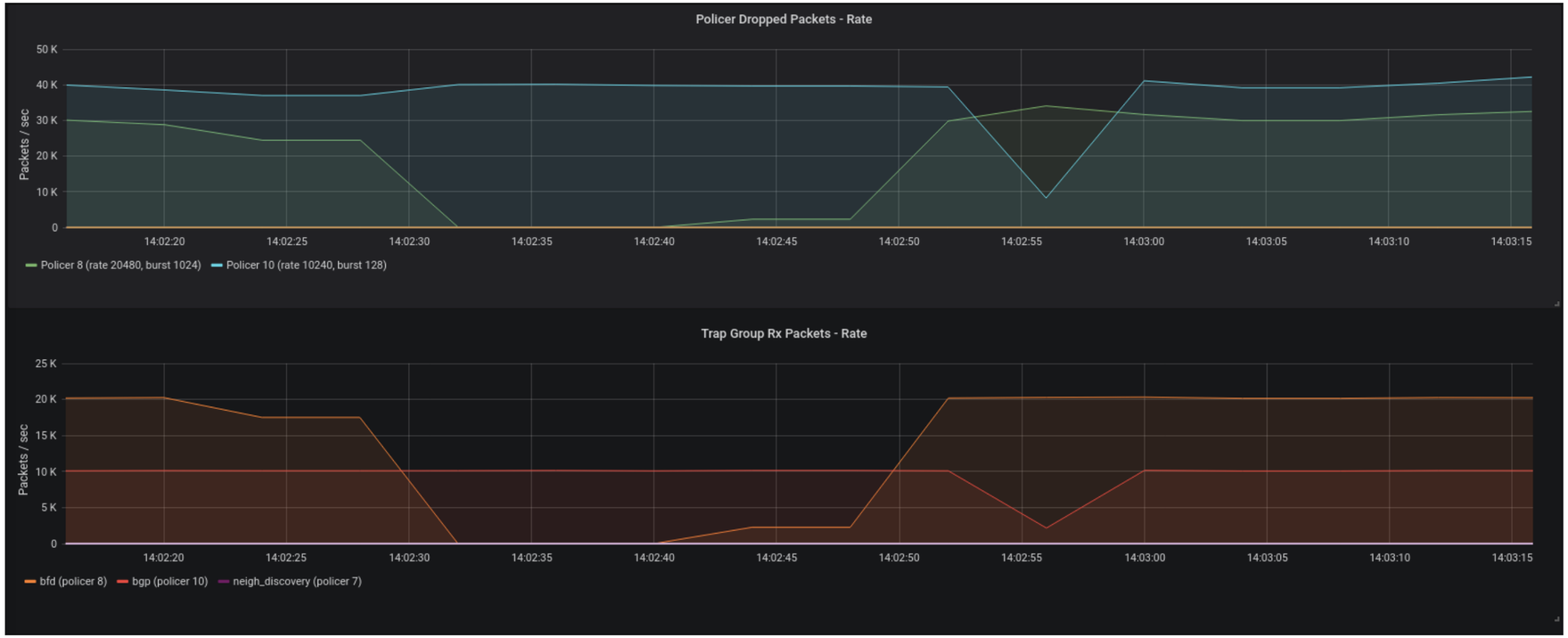
CONTROL PLANE POLICING (COPP) - SOLUTION

- Device drivers register supported packet traps with devlink
- Default control plane policy exposed to user space
- Can be monitored and tuned by user space according to its needs

```
# devlink trap group set pci/0000:01:00.0 group bgp policer 8

# devlink trap policer show pci/0000:01:00.0 policer 8
pci/0000:01:00.0:
  policer 8 rate 20480 burst 1024

# devlink trap policer set pci/0000:01:00.0 policer 8 rate 5000 burst 256

# devlink -s trap policer show pci/0000:01:00.0 policer 8
pci/0000:01:00.0:
  policer 8 rate 5000 burst 256
    stats:
        rx:
          dropped 13522938
```

CONTROL PLANE POLICING (COPP) - MONITORING

- Statistics can be exported from individual switches to a Prometheus server using `devlink-exporter`
- Visualised using Grafana



EXTENDED LINK STATE

- Sometimes a netdev can be administratively up, but operationally down
- Can now be debugged using two new ethtool netlink attributes
  - ETHTOOL_A_LINKSTATE_EXT_STATE
  - ETHTOOL_A_LINKSTATE_EXT_SUBSTATE
- Queried from device drivers using new ethtool operation:

```
int     (*get_link_ext_state)(struct net_device *,
                              struct ethtool_link_ext_state_info *);
```

- Example:

```
# ethtool swp1
Link detected: no (No cable)
```

EXTENDED LINK STATE (CONT)

- Various extended states and extended substates can be reported:

Link extended states:

``ETHTOOL_LINK_EXT_STATE_AUTONEG``	States relating to the autonegotiation or issues therein
``ETHTOOL_LINK_EXT_STATE_LINK_TRAINING_FAILURE``	Failure during link training
``ETHTOOL_LINK_EXT_STATE_LINK_LOGICAL_MISMATCH``	Logical mismatch in physical coding sublayer or forward error correction sublayer
``ETHTOOL_LINK_EXT_STATE_BAD_SIGNAL_INTEGRITY``	Signal integrity issues
``ETHTOOL_LINK_EXT_STATE_NO_CABLE``	No cable connected
``ETHTOOL_LINK_EXT_STATE_CABLE_ISSUE``	Failure is related to cable, e.g., unsupported cable
``ETHTOOL_LINK_EXT_STATE_EEPROM_ISSUE``	Failure is related to EEPROM, e.g., failure during reading or parsing the data
``ETHTOOL_LINK_EXT_STATE_CALIBRATION_FAILURE``	Failure during calibration algorithm
``ETHTOOL_LINK_EXT_STATE_POWER_BUDGET_EXCEEDED``	The hardware is not able to provide the power required from cable or module
``ETHTOOL_LINK_EXT_STATE_OVERHEAT``	The module is overheated

Link extended substates:

Autoneg substates:

``ETHTOOL_LINK_EXT_SUBSTATE_AN_NO_PARTNER_DETECTED``	Peer side is down
``ETHTOOL_LINK_EXT_SUBSTATE_AN_ACK_NOT_RECEIVED``	Ack not received from peer side
``ETHTOOL_LINK_EXT_SUBSTATE_AN_NEXT_PAGE_EXCHANGE_FAILED``	Next page exchange failed
``ETHTOOL_LINK_EXT_SUBSTATE_AN_NO_PARTNER_DETECTED_FORCE_MODE``	Peer side is down during force mode or there is no agreement of speed
``ETHTOOL_LINK_EXT_SUBSTATE_AN_FEC_MISMATCH_DURING_OVERRIDE``	Forward error correction modes in both sides are mismatched
``ETHTOOL_LINK_EXT_SUBSTATE_AN_NO_HCD``	No Highest Common Denominator

QDISC EVENTS

- Tc actions can be executed on packets that were classified by tc filters
- Qdiscs also perform "classification". Examples:
  - RED: Early drop, ECN mark
  - FIFO: Tail drop
- Extend qdiscs to expose "classification" events and attach shared blocks to them
- Only RED supported. FIFO support in the works

```
# tc qdisc replace dev swp1 root handle 1: \
     red limit 2M avpkt 1000 probability 0.1 min 500K max 1.5M \
     qevent early_drop block 10

# tc filter add block 10 matchall skip_sw \
     action mirred egress mirror dev swp6 hw_stats disabled
```

QDISC EVENTS - SAMPLING

- A lot of packets can be dropped / marked by qdiscs during bursts
- No need to act (e.g., mirror / trap) on all the packets
- Sampling allows us to act on only a subset of packets, but still get visibility into both mice and elephant flows
- Current tc-sample API is aimed at sending sampled packets to user space:

```
# tc ... action sample rate RATE group GROUP [ trunc SIZE ] [ index INDEX ]
```

- Proposed extension to allow sampled packets to be piped to other actions:

```
# tc ... action sample rate RATE { group GROUP | nogroup } [ trunc SIZE ] [ index INDEX ] [ CONTROL ]
```

- Example:

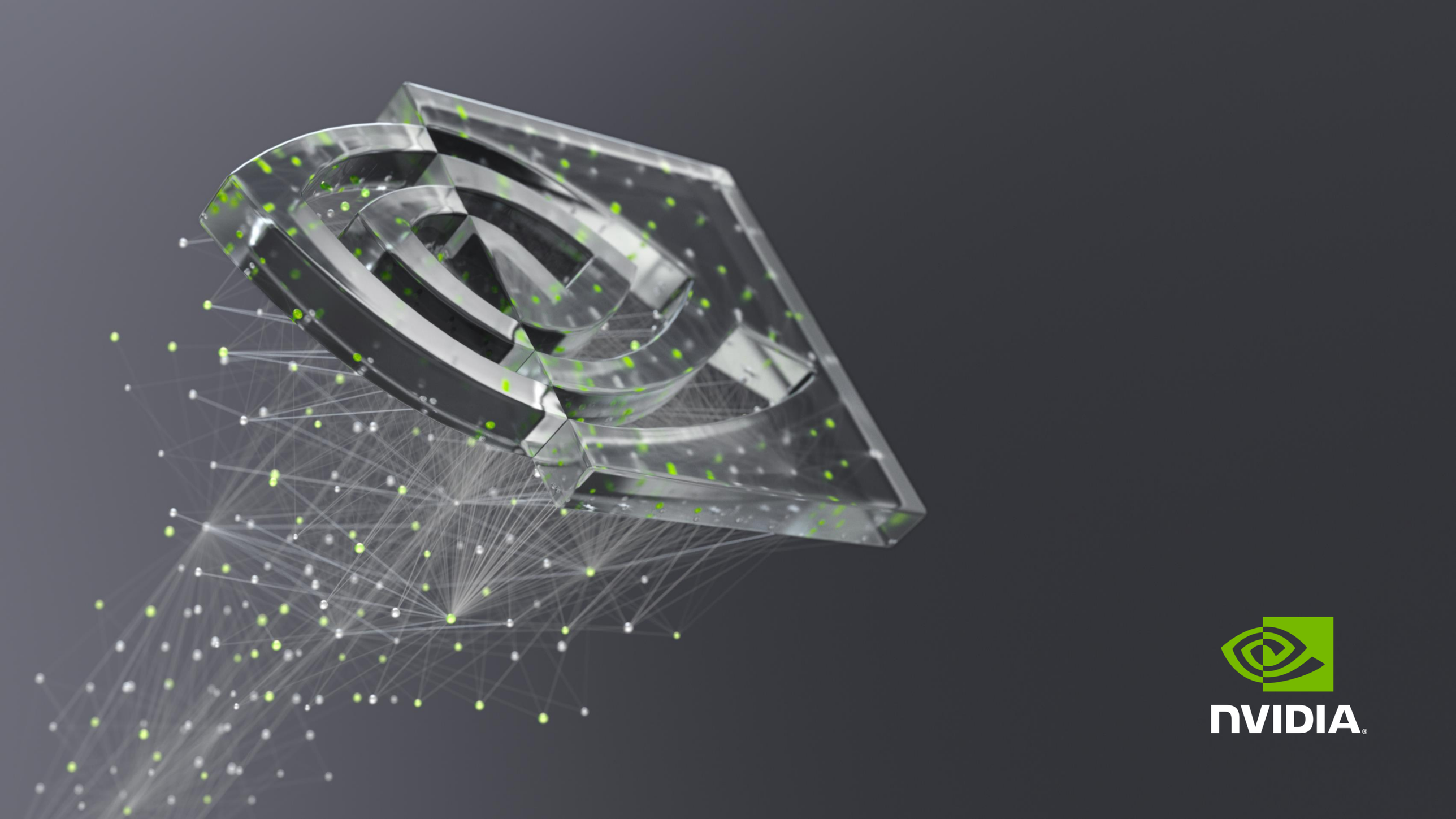
```
# tc filter add block 10 matchall skip_sw \
    action sample rate 1000 nogroup pipe \
    action mirred egress mirror dev swp6
```

REFERENCES

- https://www.kernel.org/doc/html/latest/networking/devlink/devlink-trap.html
- https://github.com/Mellanox/mlxsw/wiki/Quality-of-Service#control-plane-policing-copp
- man devlink-trap
- https://github.com/Mellanox/mlxsw/wiki/Switch-Port-Configuration#link-down-reason
- https://git.kernel.org/pub/scm/linux/kernel/git/netdev/net-next.git/tree/Documentation/networking/ethtool-netlink.rst
- https://github.com/Mellanox/mlxsw/wiki/Queues-Management#qevents
- man tc-red
- man tc-sample

REFERENCES

- https://tools.ietf.org/html/rfc2992
- https://docs.cumulusnetworks.com/cumulus-linux-41/Layer-3/Equal-Cost-Multipath-Load-Sharing-Hardware-ECMP/
- man ip-nexthop
- https://lore.kernel.org/netdev/20200610034953.28861-1-dsahern@kernel.org/


NVIDIA