Traffic shaping and ETS

Carolina Jubran

Netdev 0x19 conference

Zagreb, Croatia

March 2025

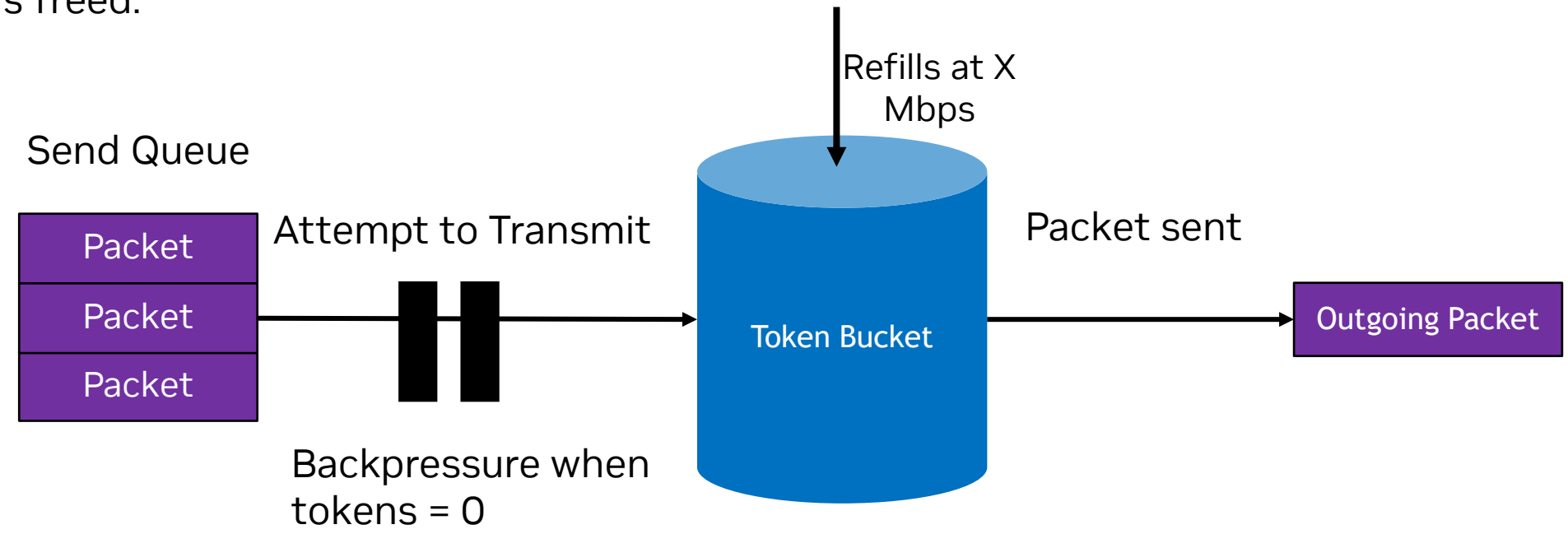
Agenda

- Background
- Use-case description
- Solution
- Timeline

Background

Traffic Shaping Basics

- Traffic shaping enforces rate limits to manage how data is sent across networks.
  - **minimum_rate:** The minimum guaranteed bandwidth for a given node.
  - **maximum_rate:** The maximum bandwidth the node is allowed to transmit.
- Backpressure ensures that once a node hits its rate limit, no more data can flow until capacity is freed.



Background

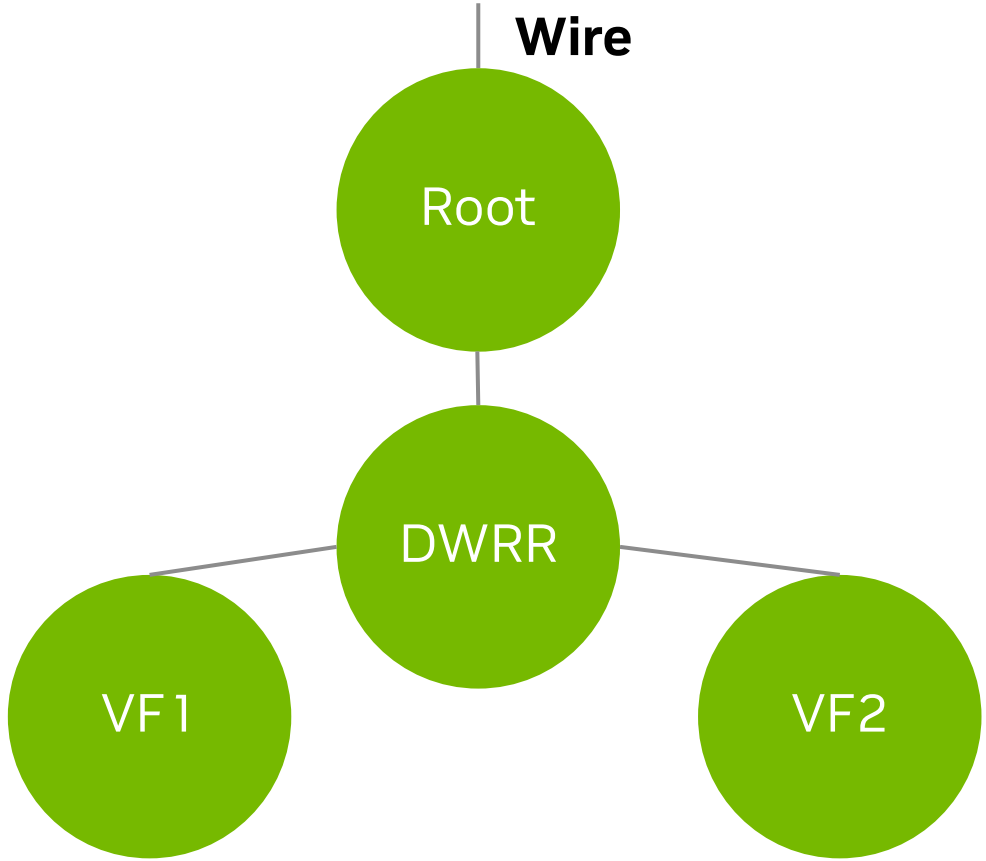
Eswitch VF Traffic Shaping

- Virtual Function shaping:
  - Capping a Virtual Function's total bandwidth as if it were a separate NIC port.
- Methods:
  - ndo_set_vf_rate:
    - ip link set $PF vf $VF max_tx_rate $Rate
  - matchall tc filter and a police:
    - tc filter add dev $VF_rep ingress matchall \
      action police rate $Rate conform-exceed drop/continue
  - devlink-rate:
    - devlink port function rate set pci/$PF/$VF tx_max $Rate

Background

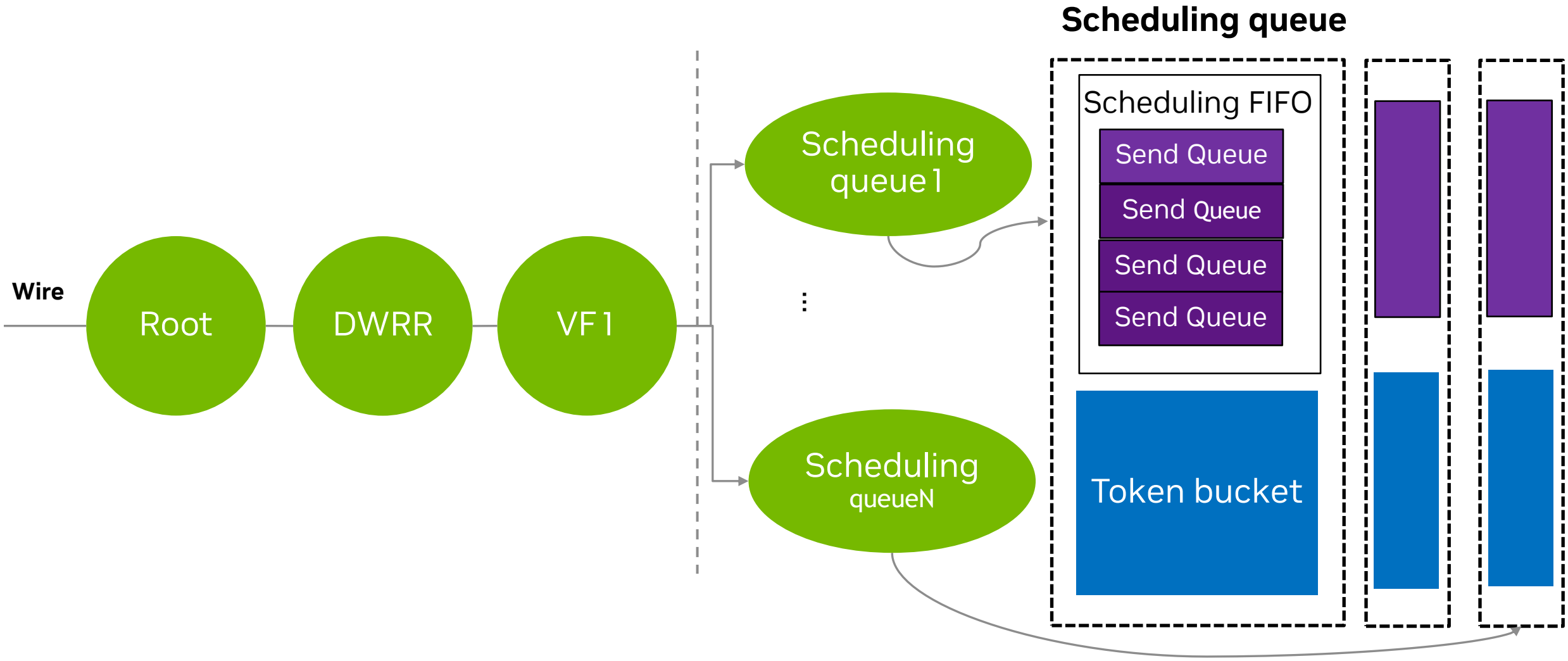
Eswitch QoS

- NVIDIA hardware provides a hierarchical quality of service offload capability.
- The firmware provides a mechanism to configure a QoS tree, where the leaf nodes are exposed to the driver through Virtual Functions.
  - DWRR – Deficit Weighted Round Robin: Functions as an intermediary scheduler that allocates weighted tokens to each connected node.



Background

Transmit Schedular



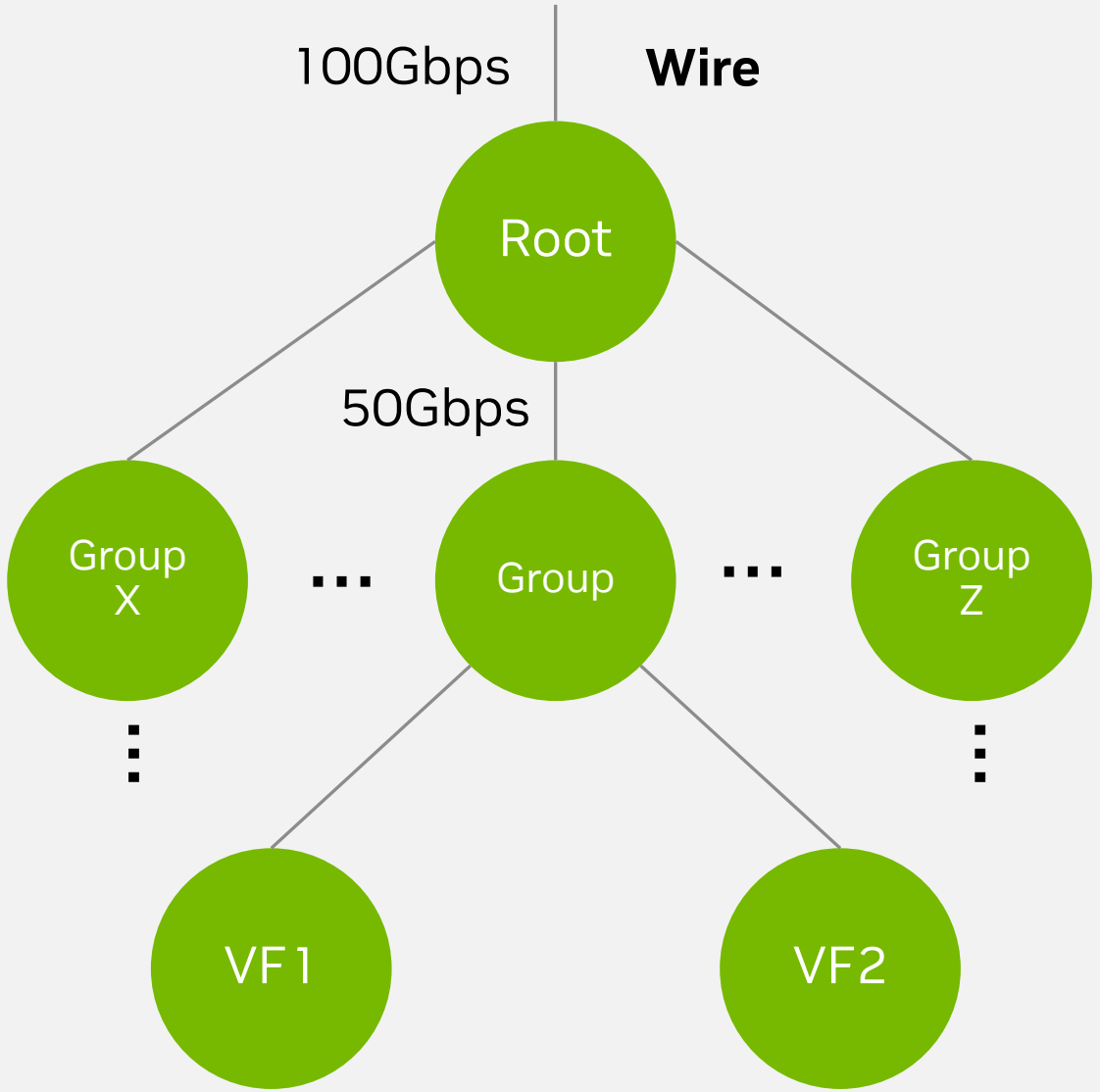
Use-case description

- Multiple **Virtual Functions (VFs)** share a single physical NIC port.
- Need per-class bandwidth guarantees **across a group of VFs** rather than a single limit applied to all traffic.
- **Goal:** Achieve Enhanced Transmission Selection (ETS) on a multi-VF group.
  - **What is ETS?** It is an IEEE (802.1Qaz) feature within Data Center Bridging that allocates bandwidth among multiple traffic classes.

Use-case Example

The goal

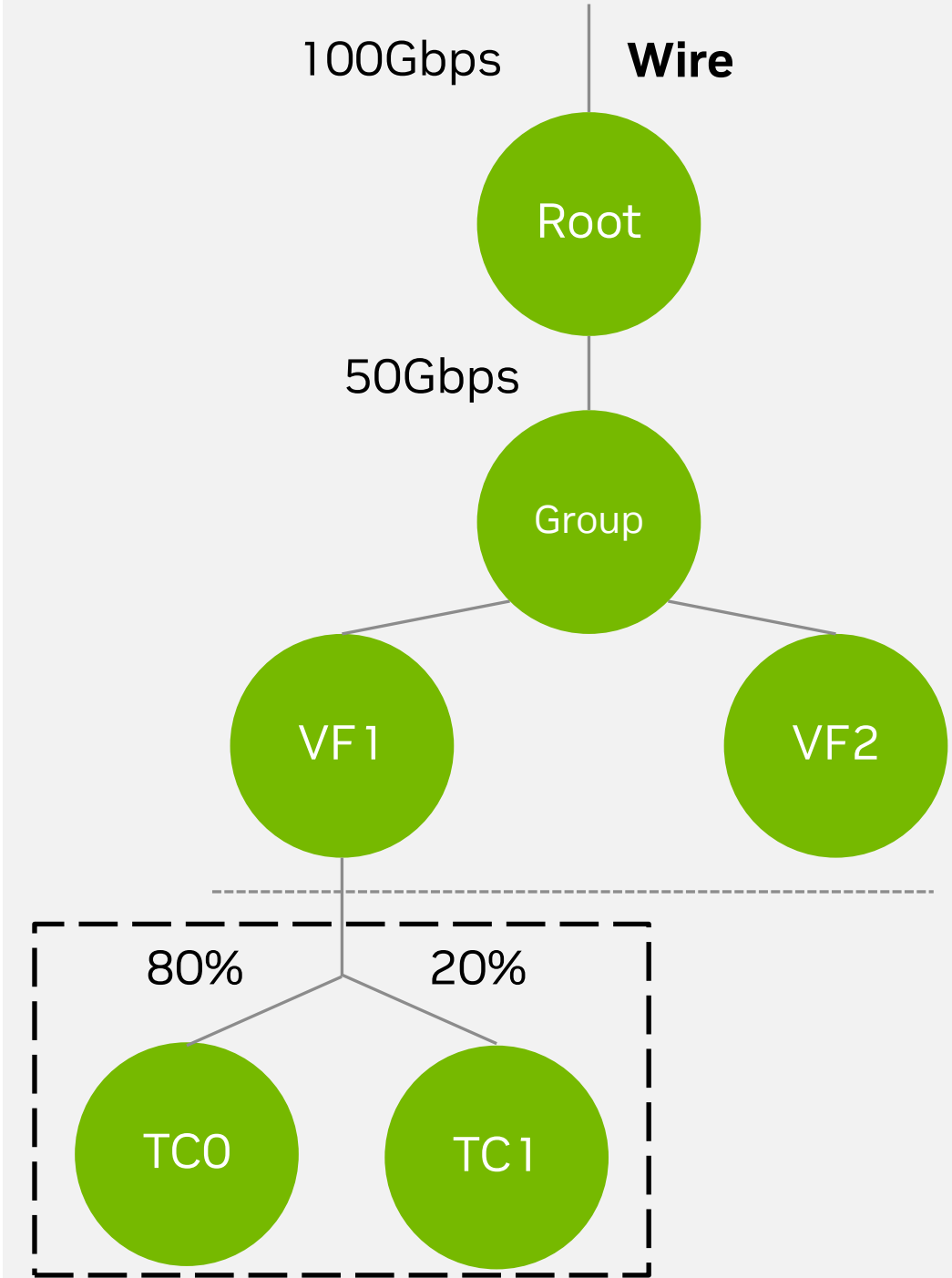
- **Scenario:** 50 Gbps group with 2 VFs (VF1 and VF2), each sending traffic in Class 0 and Class 1
- **Goal**: Treat Class 0 traffic from both VFs as one combined 40 Gbps pool, and Class 1 traffic from both as another combined 10 Gbps pool.



Use-case Example

Attempt 1

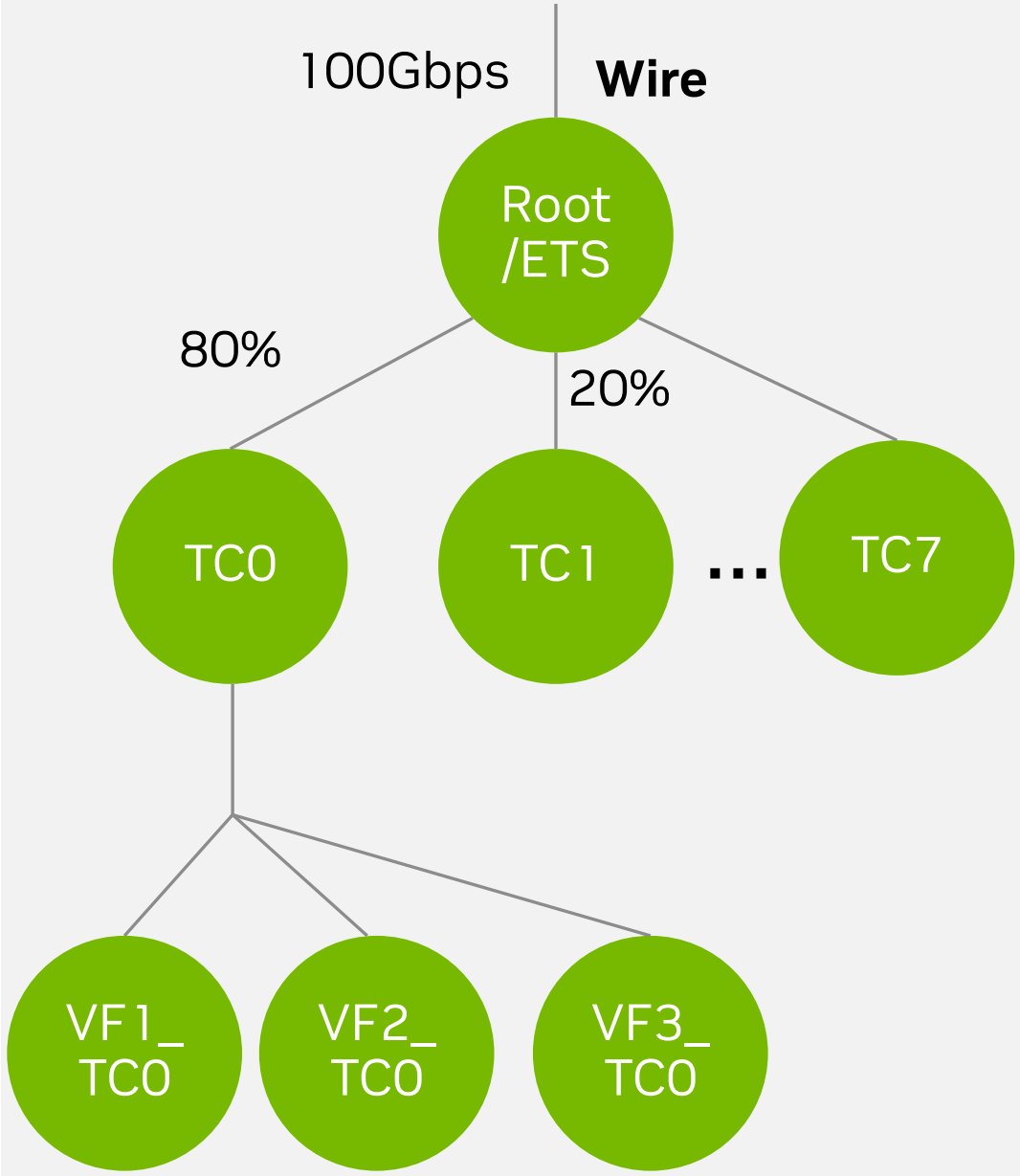
- Split bandwidth inside each VF
- **Problem 1:** This is done from inside the VF.
- **Problem 2:** High-Priority Traffic Can Be Blocked.



Use-case Example

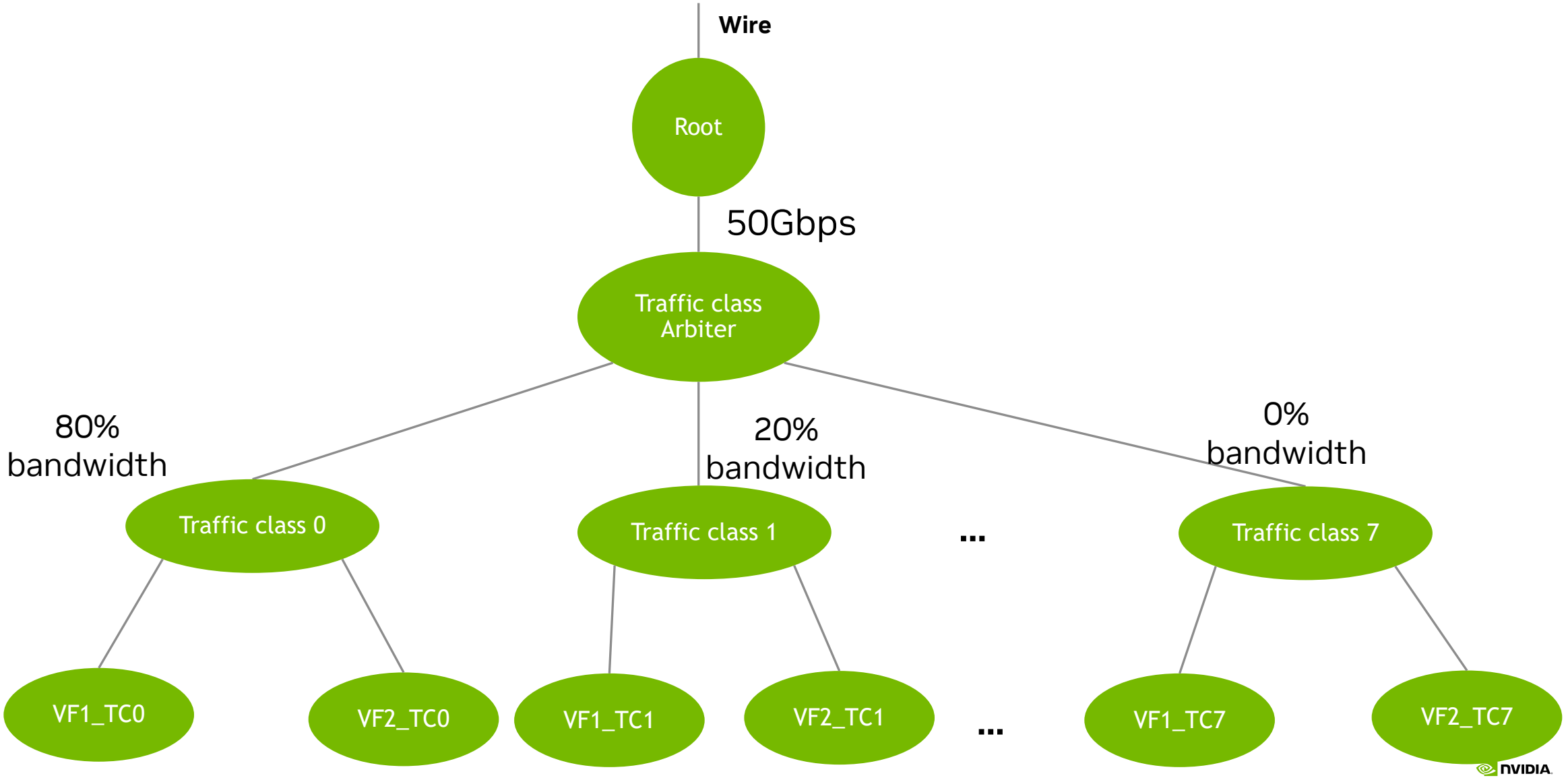
Attempt 2

- Configure ETS on the port and set bandwidth share per traffic class.
- **Problem 3**: No per-group control



Solution

Scheduling Hierarchy



Solution

Ensuring Per-Traffic-Class Queues

- **Issue:** mlx5 driver by default uses one send queue per CPU core, not per TC.
- This means Class 0 and Class 1 traffic could end up on the same hardware queue, defeating separate shaping.
- **Solution:** Use MQPRIO in **DCB mode**, where we create separate TX queues for each traffic class.

Solution

Traffic Classification

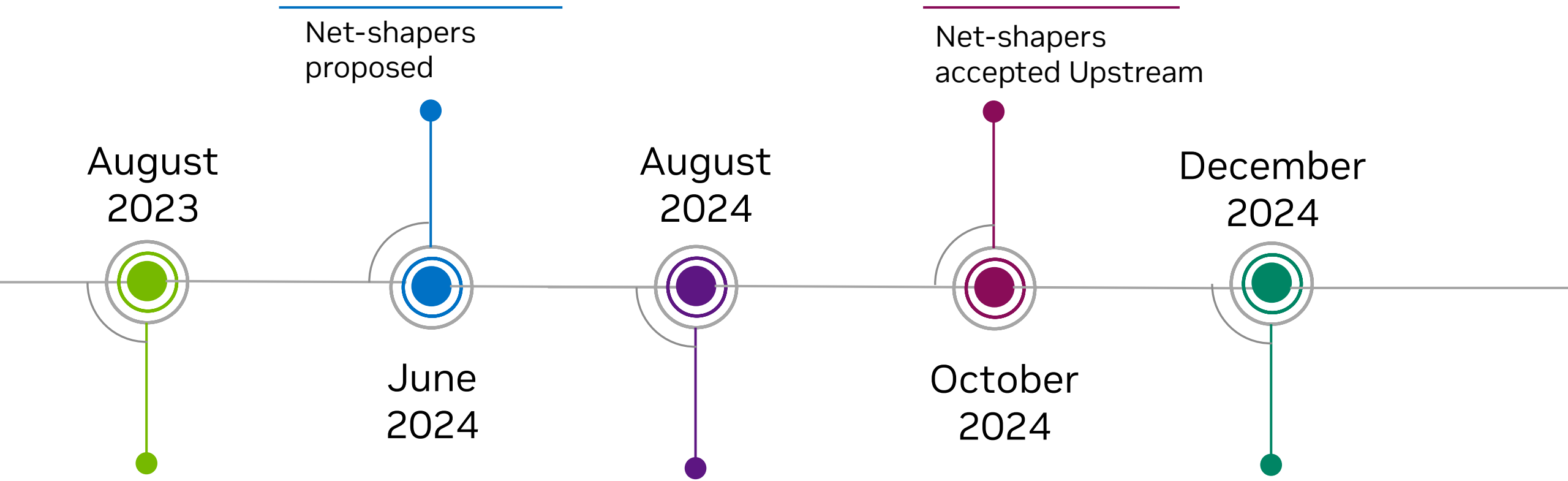
- **Packet Classification**: Partitions network traffic into multiple classes of service.
- **PCP - Priority Code Point**: 3-bit value (0–7) in an IEEE 802.1Q VLAN tag that marks a frame's priority for QoS.
- **DSCP - Differentiated Services Code Point**: 6-bit value in the IP header's Differentiated Services field.
- **Trust Mode**: Configure the device to trust DSCP or PCP for classifying packets into TCs.

Solution

devlink rate

- **Goal**: Extend devlink-rate to specify per-traffic-class bandwidth (tc-bw) on a rate object.
- devlink already supports grouping VFs under a single scheduling node.
- with tc-bw, we assign each traffic class (TC) a percentage.
- Usage example:
  devlink rate set pci/$PF/tcs_group **tc-bw 0:80 1:20 2:0 3:0 4:0 5:0 6:0 7:0**

Timeline


August 2023
Upstream discussion about fragmented traffic shaping methods.
June 2024
Net-shapers proposed
August 2024
RFC: Introducing tc-bw in devlink
October 2024
Net-shapers accepted Upstream
December 2024
Devlink extension patches posted; community requests integration with Net-shapers

References

- Devlink-rate - https://man7.org/linux/man-pages/man8/devlink-rate.8.html
- Net-shapers - https://lore.kernel.org/all/cover.1728460186.git.pabeni@redhat.com/
- Devlink RFC - https://lore.kernel.org/netdev/202408150939.GeZ6uR8S-lkp@intel.com/T/
- Devlink and mlx5 patches - https://lore.kernel.org/netdev/20250209101716.112774-1-tariqt@nvidia.com/


NVIDIA
Questions?
Thank you