Nokia Technical Brief

Congestion management in AI/ML networks

3HE-21914-AAAA-TQZZA
Issue 1
August 2025
Author: Vivek V

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

Contents

3HE-21914-AAAA-TQZZA
August 2025
Issue 1

1 Introduction

The purpose of this document is to explain the congestion detection, management, and avoidance in artificial intelligence and machine learning (AI/ML) networks. Congestion management involves multiple technologies working in tandem with each other and, when paired with network design principles meant to provide a collision-free environment, we can achieve a lossless AI fabric, which is essential to AI/ML networks.

The scope of this document is confined to the Nokia 7220 IXR H4, which is a Broadcom tomahawk-based router. The internal specifications such as buffer size, queueing, and so on, will vary for other platforms.

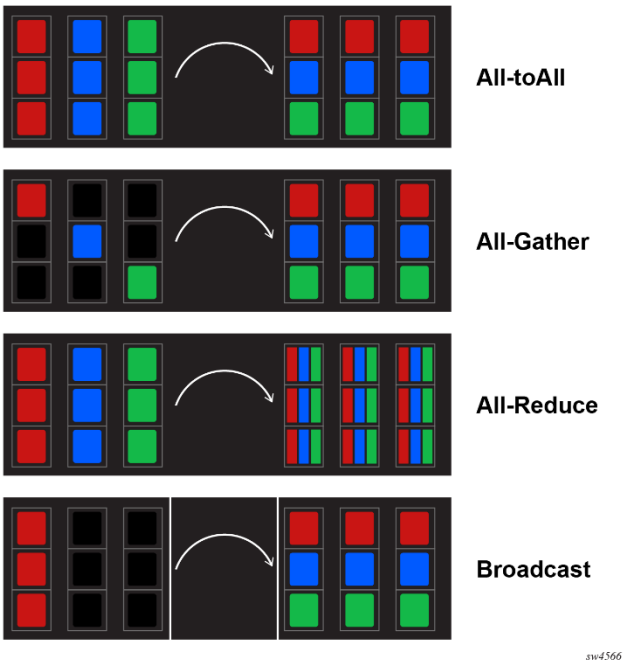
3HE-21914-AAAA-TQZZA
August 2025
Issue 1

2 Traffic in AI/ML networks

The traffic patterns in AI/ML networks are defined by the collective communication operations, a few examples of which are described below. The collective communication operations are further related to the distributed computing and parallel processing mechanisms that have been employed by the application (learning model).

2.1 AI/ML collective communication operations

Collective libraries break a job into pieces and orchestrate the GPUs to work in synchronous fashion, some of these mechanisms are:



All-to-All – This operation allows each participating process (or GPU) to send and receive data from every other process. Each process provides a specific amount of data that is distributed to all other processes.

All-Gather – operation collects data from all processes and distributes the combined result to every process. Each process contributes its own data, and the output is an aggregation of all contributions.

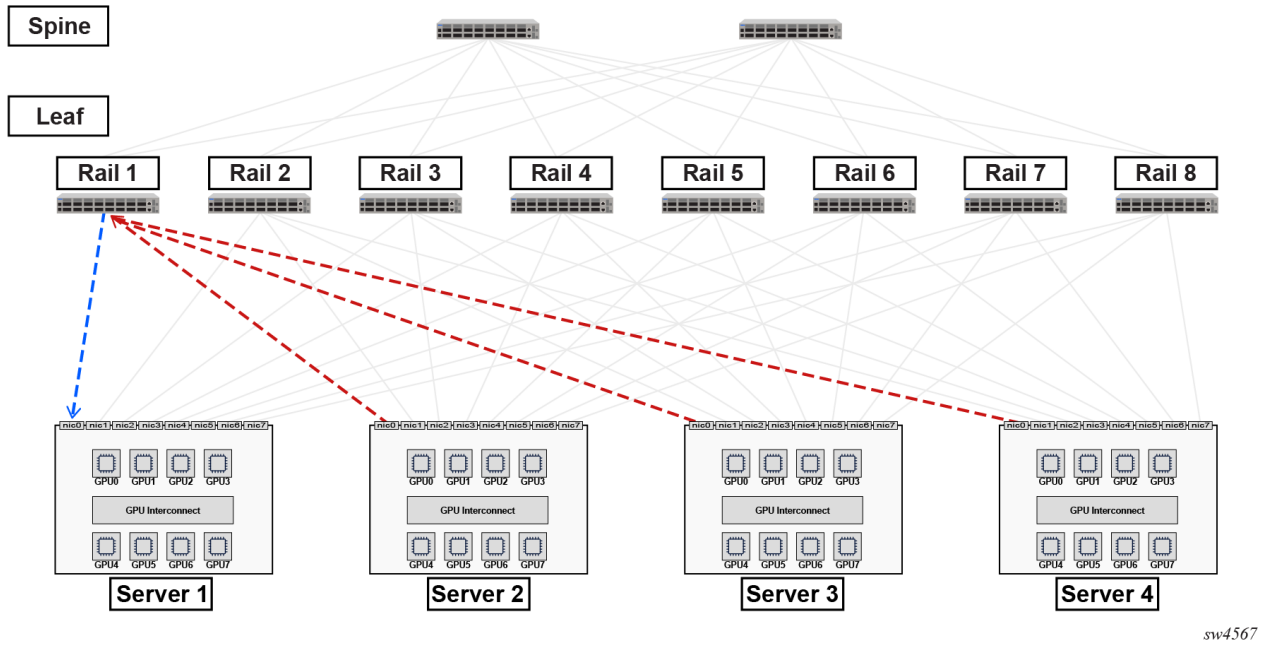
All-Reduce – operation performs a reduction (such as summation or averaging) on data across all processes and then distributes the result back to all processes. Each process starts with its own data, and the result is stored in each process's memory.

Broadcast - operation sends data from a designated root process to all other processes. The root process holds the original data, which is copied to all other ranks.

3 Congestion scenarios in AI-DC clusters

There are many scenarios in production AI/ML data center (DC) clusters that can lead to temporary or permanent congestion scenarios; a few of them have been described below.

3.1 Scenario 1: Intra-rail – simultaneous microburst



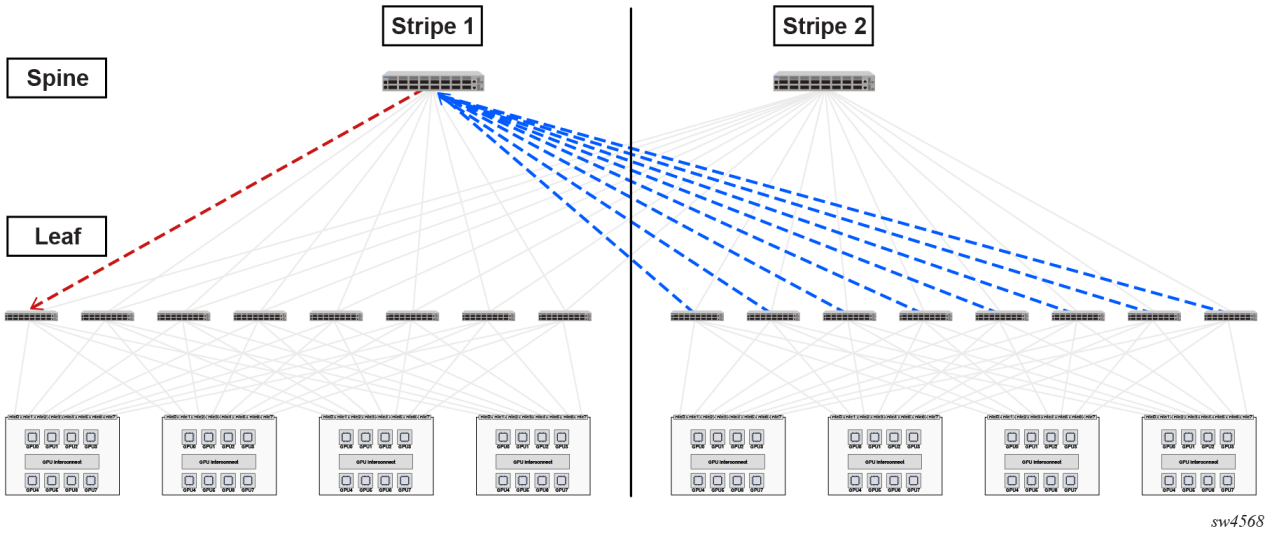
The traffic pattern in AI/ML training clusters is based on the data, model, or pipeline parallelism that is employed to distribute the workload; based on this, the individual GPU receives a piece of the workload, finishes processing it, places the entire payload onto the network queue, and sends it out all at once to the other GPUs to synchronize the state.

Note: This is done via RDMA over converged ethernet (RoCEv2) for Ethernet-based fabrics.

Example: As shown in the figure above, all communication for Rail 1, that is, if all GPU 0s need to synchronize with each other, they need to go through Leaf 1, which is Rail 1 in this case.

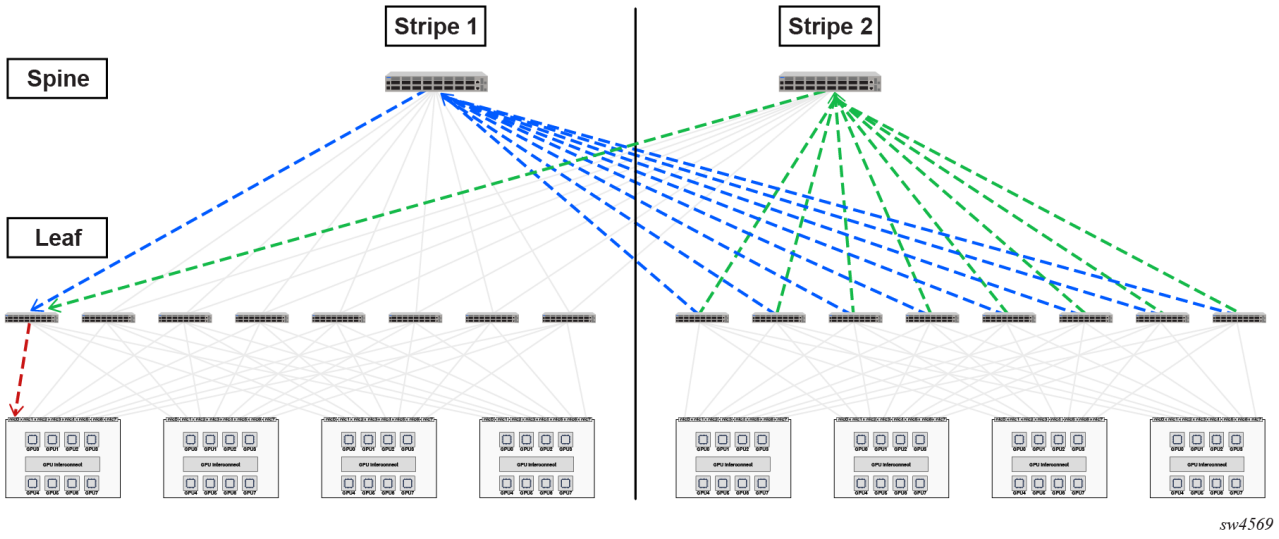
If all the servers' GPU 0s sync with GPU 0 of server 1 at the same time and send out a micro burst at the same time, there is a chance that the dedicated buffer on the switch may get overwhelmed and the shared buffer utilization will also be impacted and lead to congestion.

3.2 Scenario 2: Spine polarization



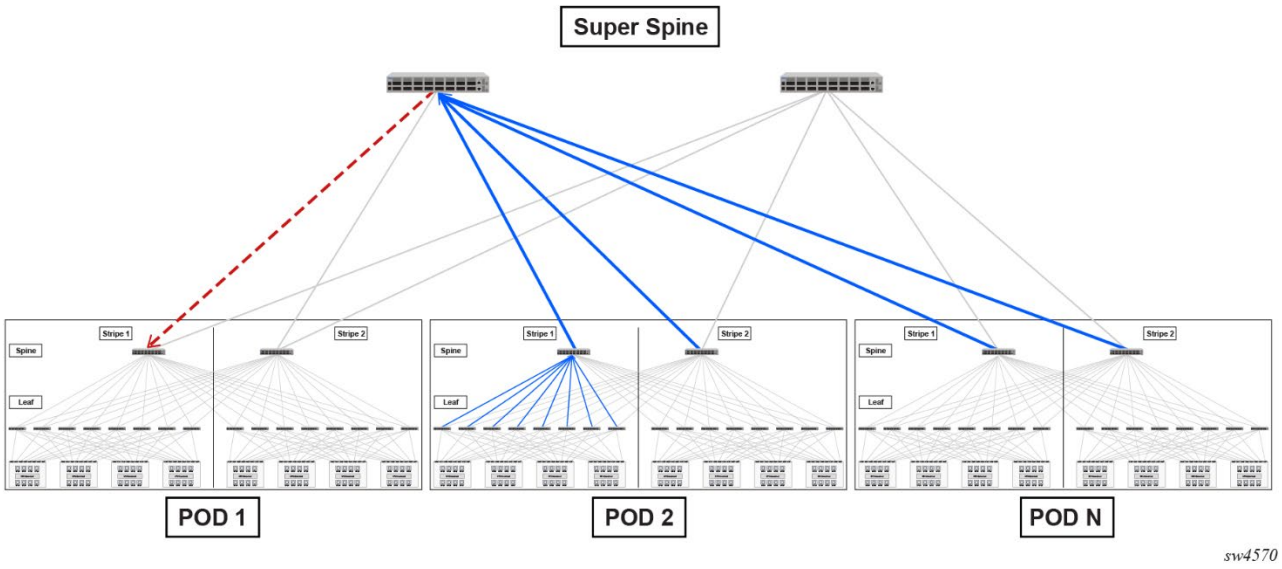
In the above scenario, consider an ALL-to-ALL operation where every GPU in the high-bandwidth zone must synchronize with every other GPU, and for an instance in time, all the GPUs polarize to a single spine to send it across to a rail in another stripe. This can lead to congestion at the spine layer.

3.3 Scenario 3: Inter-stripe leaf-GPU congestion



The workload distributions may happen over several stripes, which can lead to a scenario when multiple stripes try to synchronize with GPU 0 of 1 server at the same time, as shown in the figure above. This can lead to congestion in the GPU southbound ports.

3.4 Inter-pod super-spine polarization



In the scenario shown above, where the workload is distributed among multiple pods that have multiple stripes, there are two possible scenarios:

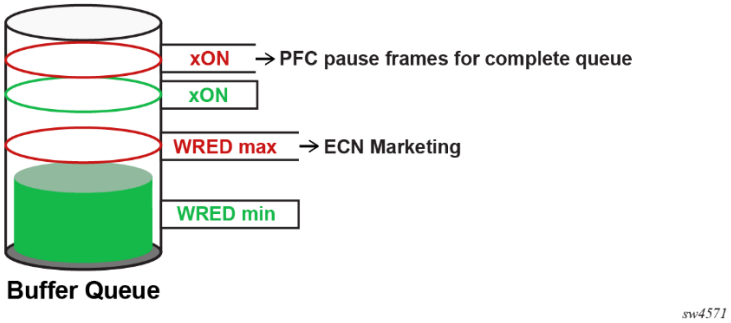
- The first scenario is where multiple pods polarize to the same spine, as shown in the figure above, which causes buffer overflow on the super-spine due to polarization and the simultaneous burst.
- The other scenario is where the oversubscription to the super-spine layer is greater than 1, which means that there are more leaf-to-spine ports than there are spine-to-super-spine ports, which can cause a scenario where the rate of traffic coming into a spine exceeds the rate at which it can send it out, which can cause congestion

Note: Many probable congestion scenarios can be avoided by using non-blocking architectures and maintaining an oversubscription ratio of 1.

4 DCQCN

One mechanism that can be used to identify and manage congestion is Data Center Quantized Congestion Notification (DCQCN), which is an end-to-end congestion control mechanism designed for RoCEv2 networks.

It combines Explicit Congestion Notification (ECN), which is an end-to-end control mechanism, and priority-based flow control (PFC), which is a per-segment control mechanism. A lossless Ethernet fabric is crucial to an AI/ML cluster, and this is an important component to achieve the same.



DCQCN uses ECN and PFC together to build a lossless Ethernet network.

Minor congestion results in moderate shared buffer usage wherein WRED with ECN triggers first to slow down the traffic by getting the receiver to send CNP packets back to the sender.

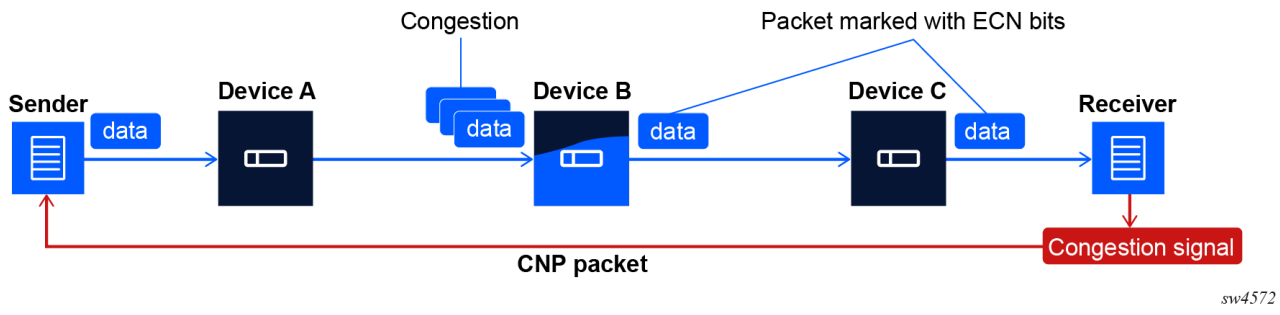
If the scenario is such that the receiver is not sending the CNP packets or the sender is not slowing the traffic, it can lead to more congestion.

Major congestion results in high usage of buffers wherein PFC is triggered, and this operates per queue on the network devices and slows traffic per segment.

The WRED minimum and maximum thresholds are set for lower buffer utilization to mitigate congestion first, and the PFC threshold is set higher as a safety net to mitigate congestion after ECN.

Note: Appropriate thresholds must be set or tuned for every AI/ML model and resultant traffic patterns.

4.1 ECN



ECN **(Explicit Congestion Notification),** as described in RFC 3168**,** facilitates end-to-end congestion management without dropping packets and hence is ideal for lossless AI fabrics.

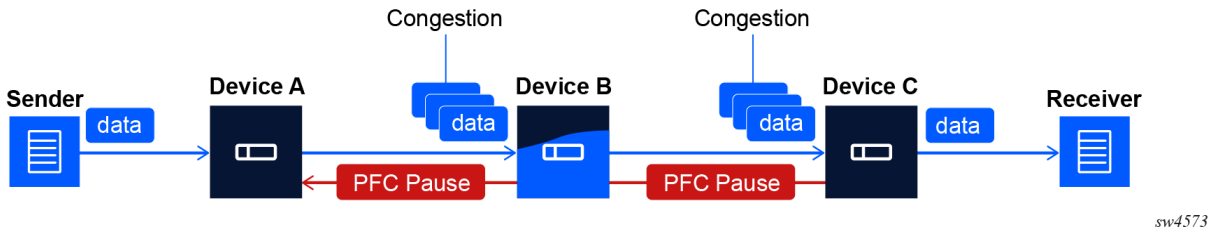
The figure above shows the communication between two endpoints, a sender and a receiver, with the network path having three network devices in between.

When Device B on the network path faces congestion, it marks packets with ECN 11 bits. When a marked packet reaches the receiver, the ECN sensitive receiver sends a congestion notification packet (CNP) back to the sender via a dedicated queue that has been allocated strictly for it on all the network devices in the path.

When the sender receives this packet, it is notified that there is congestion in the network, and it needs to decrease the rate of traffic that is being sent out.

With this mechanism, the traffic rate is controlled solely at the endpoints.

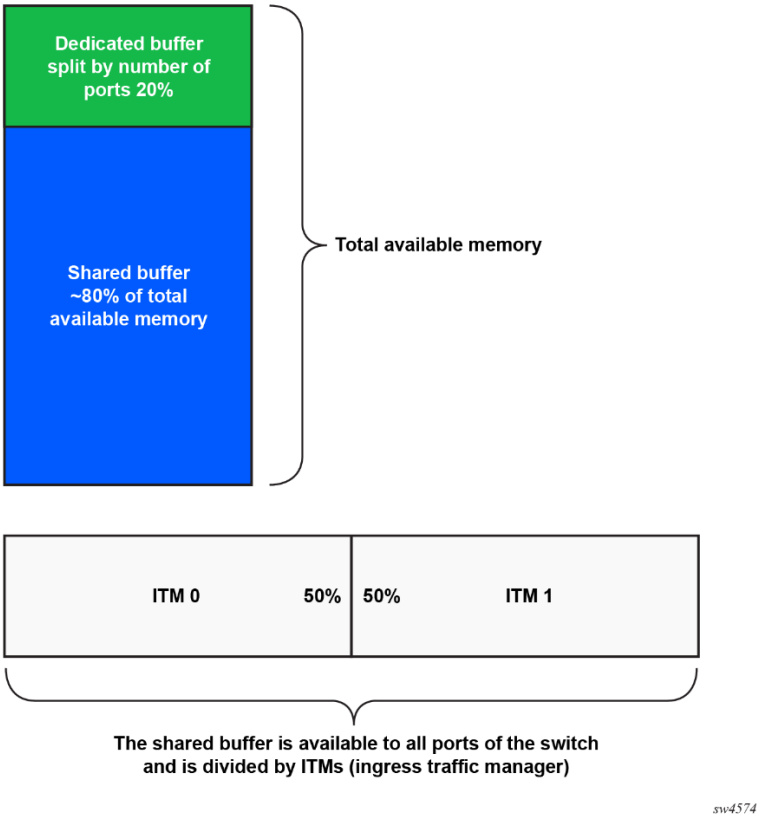
4.2 PFC



PFC **(Priority Flow Control)** unlike ECN, works on a per-segment basis. As shown in the figure above, data is being sent from the sender to the receiver via network devices A, B, and C. Devices B and C are facing congestion. PFC pause frames will be sent from C to B and from B to A. If either C or B stops being congested, the corresponding pause frames will stop as well.

Note: Unlike Ethernet pause, which is sent on the entire segment, PFC pause is only sent on the queue that is facing congestion. For example, if only queue 3 of 0 to 7 is facing congestion, queues 0 to 2 and 4 to 7 do not see the PFC pause and, unlike a CNP packet which flows all the way to the sender, the PFC pause ends at the segment where the congestion was seen.

4.3 ECN and PFC threshold



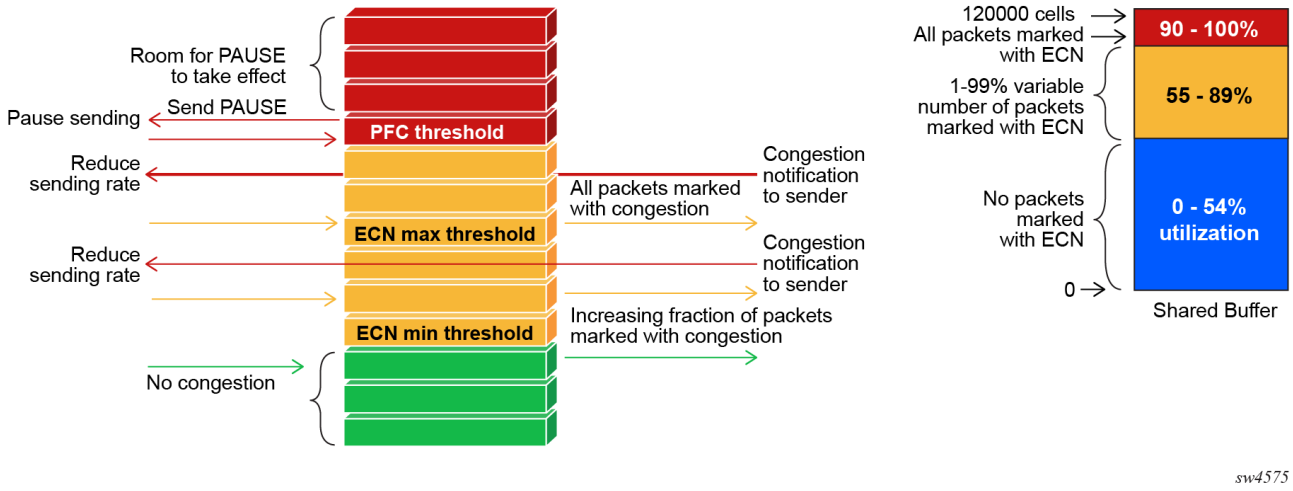
The total available memory of the router is split into two kinds of buffers: the dedicated buffer and the shared buffer.

The dedicated buffer is then further allocated equally to all available ports.

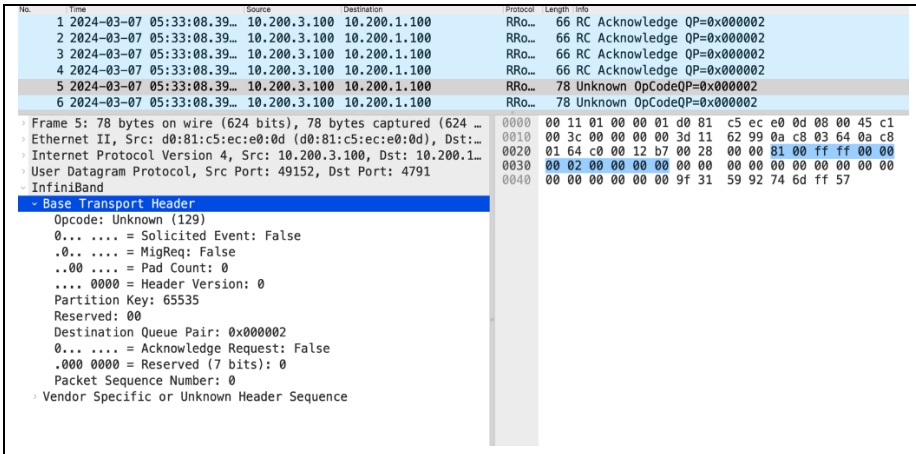
For example, if the total memory is 100 mb, about 20 mb is reserved as the dedicated buffer, and because we have 32 ports, the dedicated buffer available to each port is 20/32 = .625 mb.

The remaining ~80% of memory is split, as shown in the diagram, per ITM. Assuming default alpha values, each ITM receives 50%, which is available to all ports that are part of that ITM (16 per ITM out of 32 ports, not in sequential order).

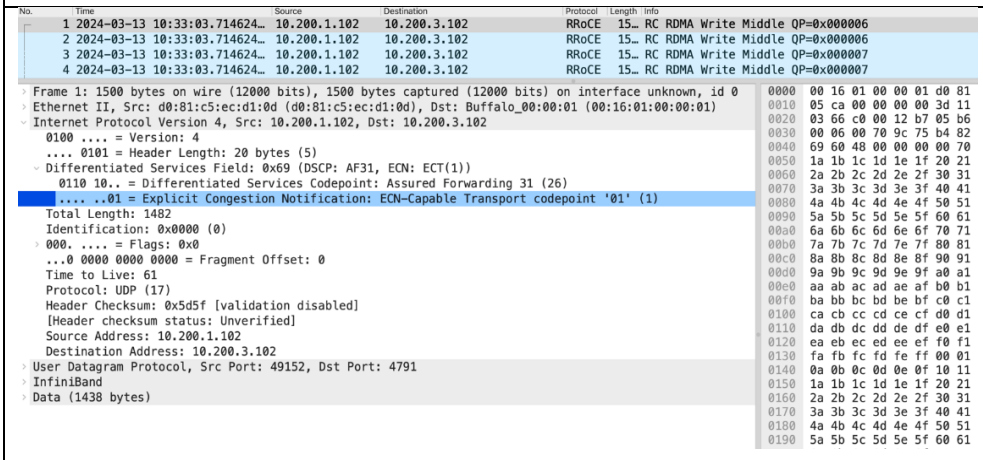
The buffer is further allocated per port per queue. For example, if there are 16 ports and queues 3 and 4 have congestion on all the ports, the buffer allocation is:
40 Mb (per ITM) / 16 ports × (2 queues per port)



ECN and PFC thresholds are all factors of shared buffer utilization, hence the thresholds shown here are percentages of the available shared buffer memory shown in the calculation above.



A CNP packet has an opcode of 0x81 in the InfiniBand BTH (Base Transport Header). An example packet capture is shown in the figure



An ECN packet sets the ECN bits in the DSCP field of the IP header to 11 (the ECN capability is indicated by setting it to 01 or 10). An example packet is shown below shows the capability (which is set to 11 to indicate congestion).

4.4 Configuring DCQCN on SRL

4.4.1 Classification

```
classifiers {
            dscp-policy ingress-backend-backend1 {
                dscp 0 { <<<<<<<<<<<<<< all unmarked traffic
                    forwarding-class fc0
                    drop-probability low
                }
                dscp 26 { <<<<<<<<<<<<< RoCEv2 traffic from the GPU marked with DSCP 26
                    forwarding-class fc3
                    drop-probability low
                }
                dscp 48 {<<<<<<<<<<<<<< CNP packets marked with DSCP 48
                    forwarding-class fc6
                    drop-probability low
                }
            }
        }
```

Example 1

The expectation from the switch is that all RoCEv2 coming in from the GPU servers should be marked as DSCP 26 and all CNP packets must be marked as DSCP 48. The classifier takes these DSCP markings and assigns it to forwarding classes fc3 and fc6 respectively. All unmarked packets go into fc0.

4.4.2 Forwarding classes and queue mapping

```
        queues {
            queue unicast-0 {
                queue-index 0
            }
            queue unicast-3 { <<<<<<< RoCEv2 mapped to queue 3
                queue-index 3
            }
            queue unicast-6 {
                queue-index 6
            }
            pfc-queue pfc-3 { <<<<<<<<< PFC queue also mapped to queue 3
                queue-index 3
            }
        }
forwarding-classes {
            forwarding-class fc0 {
                output {
                    unicast-queue unicast-0
                }
            }
            forwarding-class fc3 {
                output {
                    unicast-queue unicast-3
                }
            }
            forwarding-class fc6 {
```

```
                output {
                    unicast-queue unicast-6
                }
            }
        }
pfc-mapping-profile dcqcn1 {
            received-traffic {
                unicast-mapping {
                    pfc-queue pfc-3 {
                        forwarding-class [
                            fc3
                        ]
                        pfc-pause-frame-priority [
                            3
                        ]
                    }
                }
            }
            received-pfc-pause-frames {
                deadlock {
                    enable true
                    detection-timer 750
                    recovery-timer 750
                }
                queue unicast-3 {
                    enable-pfc true
                    pfc-pause-frame-priority [
                        3
                    ]
                }
            }
        }
```

Example 2

The forwarding classes are mapped to output queues. The Broadcom tomahawks have eight queues, 0 to 7 and, in this case, fc3 -> RoCEv2 is mapped to queue 3, fc6-> CNP is mapped to queue 6, and unmarked traffic is mapped to queue 0.

The “received-pfc-pause-frame” section of Example 2 deals with the action to be taken on receipt of PFC pause.

4.4.3 Scheduler policy and buffer management

```
      buffer-management {
            queue-management-profile
            egress-backend-backend1-2 {
                weight-factor 0
                wred {
                    wred-slope all drop-
probability all enable-ecn true {
                        min-threshold-
percent 40
                        max-threshold-
percent 80
                        slope-enabled
false
                        max-drop-
probability-percent 100
```

```
 scheduler-policies {
            scheduler-policy egress-
backend-backend1 {
                scheduler 0 {
                    priority strict
                    input unicast-6 {
                        queue-name
unicast-6
                        peak-rate-percent
10
                    }
                }
                scheduler 1 {
                    input unicast-3 {
```

```
                    }
                }
            }
            buffer-allocation-profile
egress-backend-backend1 {
                queues {
                    queue unicast-0 {
                        maximum-burst-size
52110640
                    }
                    queue unicast-3 {
                        maximum-burst-size
52110640
                    }
                    queue unicast-6 {
                        maximum-burst-size
52110640
                    }
                }
            }
            buffer-allocation-profile
ingress-backend-backend1 {
                queues {
                    pfc-queue pfc-3 {
                        maximum-burst-size
52110640
                    }
                }
            }
```

```
                        queue-name
unicast-3
                        peak-rate-percent
100
                        weight 50
                    }

interfaces {
            interface ethernet-1/1 {
                interface-ref {
                    interface ethernet-1/1
                }
                pfc {
                    pfc-mapping-profile
dcqcn1
                    pfc-enable true
                }
                input {
                    pfc-buffer-allocation-
profile ingress-backend-backend1
                }
                output {
                    buffer-allocation-
profile egress-backend-backend1
                    queues {
                        queue unicast-3 {
                            queue-
management-profile egress-backend-
backend1-2
                        }
                    }
                    scheduler {
                        scheduler-policy
egress-backend-backend1
                    }
```

Example 3

The “buffer-management” section of Example 3 deals with the ECN minimum and maximum thresholds, which define the point at which ECN marking begins and at which point all packets are marked with ECN 11.

The “scheduler profiles” section defines the parameters of the queues and then are mapped to the respective interfaces themselves.

4.4.4 Validation

```
A:h4-spine# info from state qos interfaces interface ethernet-1/1
    qos {
        interfaces {
            interface ethernet-1/1 {
                interface-ref {
                    interface ethernet-1/1
                }
                pfc {
                    pfc-enable true
                    oper-state up
                    deadlock-detection-timer 0
                    statistics {
                        total-pfc-pause-frames-received 0
                        total-pfc-pause-frames-generated 0
                        total-packet-pfc-discards 0
                        pfc-priority 0 {
                            pfc-pause-frames-received 0
                            pfc-pause-frames-generated 0
                            pfc-transitions 0
                            deadlock-recovery-occurrences 0
                        }
                        pfc-priority 3 {
                            pfc-pause-frames-received 0
                            pfc-pause-frames-generated 0
                            pfc-transitions 0
                            deadlock-recovery-occurrences 0
                        }
                    }
                    pfc-queue pfc-3 {
                    }
                }
                input {
                }
                output {
                    buffer-allocation-profile egress-backend-backend1
                    queues {
                        queue unicast-0 {
                            queue-management-profile default
                            forwarding-class [
                                fc0
                            ]
                            active-queue-management {
                                wred-slope tcp drop-probability low enable-ecn false {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope tcp drop-probability medium enable-ecn false {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope tcp drop-probability high enable-ecn false {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope non-tcp drop-probability low enable-ecn false
{
                                    min-threshold-bytes 0
```

```
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope non-tcp drop-probability medium enable-ecn
false {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope non-tcp drop-probability high enable-ecn false
{
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope all drop-probability low enable-ecn true {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope all drop-probability medium enable-ecn true {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope all drop-probability high enable-ecn true {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                            }
                            queue-depth {
                                maximum-burst-size 52110640
                                committed-burst-size 0
                            }
                            scheduling {
                                peak-rate-percent 100
                                peak-rate-bps 400003072000
                                strict-priority true
                                weight 2
                            }
                            queue-statistics {
                                aggregate-statistics {
                                    transmitted-packets 79498236573470
                                    transmitted-octets 26808903327496734
                                    dropped-packets 19874632
                                    dropped-octets 6787310760
                                }
                            }
                        }
--snip--
                        queue unicast-3 {
                            queue-management-profile egress-backend-backend1-2
                            forwarding-class [
                                fc3
                            ]
                            active-queue-management {
                                wred-slope tcp drop-probability low enable-ecn false {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope tcp drop-probability medium enable-ecn false {
```

```
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope tcp drop-probability high enable-ecn false {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope non-tcp drop-probability low enable-ecn false
{
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope non-tcp drop-probability medium enable-ecn
false {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope non-tcp drop-probability high enable-ecn false
{
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope all drop-probability low enable-ecn true {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope all drop-probability medium enable-ecn true {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope all drop-probability high enable-ecn true {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                            }
                            queue-depth {
                                maximum-burst-size 52110640
                                committed-burst-size 0
                            }
                            scheduling {
                                peak-rate-percent 100
                                peak-rate-bps 400003072000
                                strict-priority false
                                weight 50
                            }
                            queue-statistics {
                                aggregate-statistics {
                                    transmitted-packets 59624158489702
                                    transmitted-octets 19468093162138876
                                    dropped-packets 0
                                    dropped-octets 8854974912
                                }
                            }
                        }
--snip--
```

```
                        queue unicast-6 {
                            queue-management-profile default
                            forwarding-class [
                                fc6
                            ]
                            active-queue-management {
                                wred-slope tcp drop-probability low enable-ecn false {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope tcp drop-probability medium enable-ecn false {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope tcp drop-probability high enable-ecn false {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope non-tcp drop-probability low enable-ecn false
{
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope non-tcp drop-probability medium enable-ecn
false {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope non-tcp drop-probability high enable-ecn false
{
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope all drop-probability low enable-ecn true {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope all drop-probability medium enable-ecn true {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                                wred-slope all drop-probability high enable-ecn true {
                                    min-threshold-bytes 0
                                    max-threshold-bytes 0
                                    max-probability 0
                                }
                            }
                            queue-depth {
                                maximum-burst-size 52110640
                                committed-burst-size 0
                            }
                            scheduling {
                                peak-rate-percent 10
                                peak-rate-bps 40000000000
                                strict-priority true
```

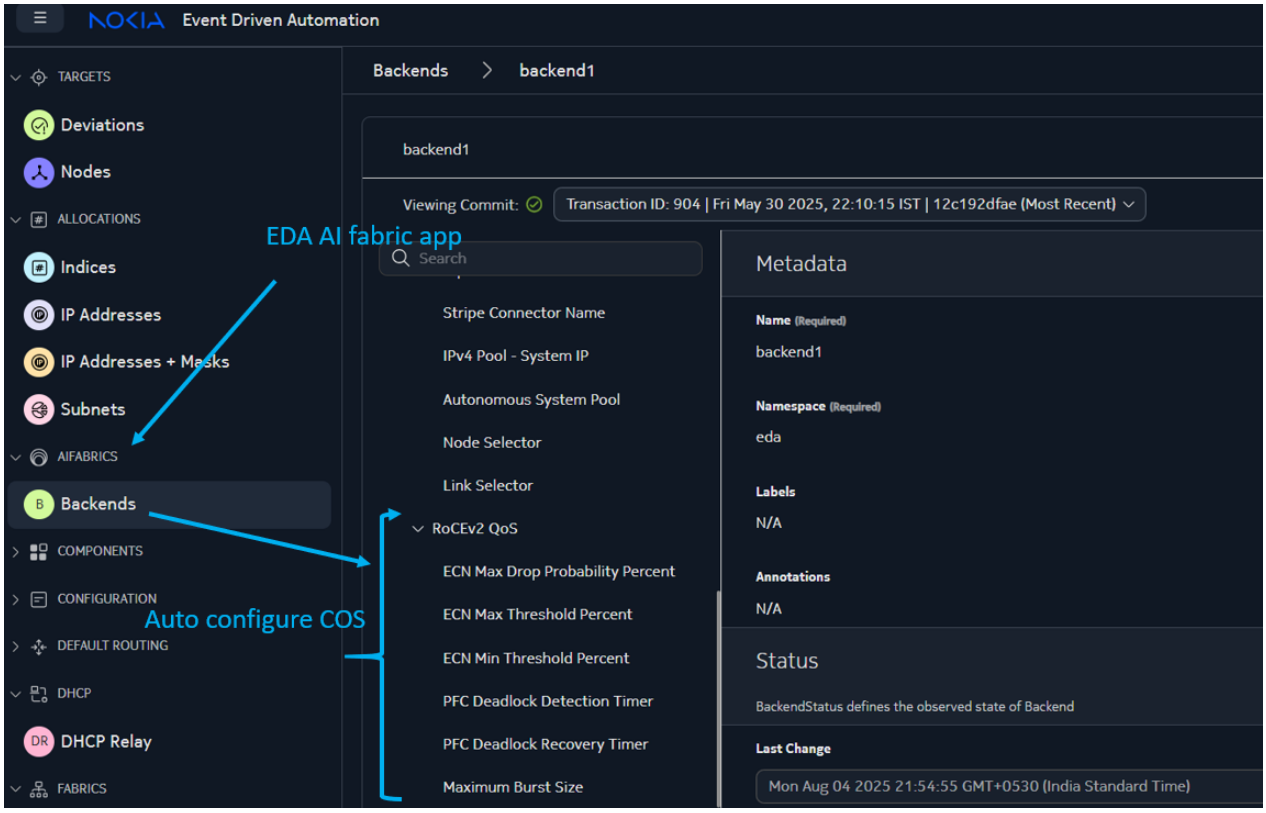
```
                                weight 11
                            }
                            queue-statistics {
                                aggregate-statistics {
                                    transmitted-packets 0
                                    transmitted-octets 0
                                    dropped-packets 0
                                    dropped-octets 0
                                }
                            }
                        }

--{ + running }--[  ]--
A:h4-spine2#
```

Example 4

Example 4 shows interface counters for the respective ECN- and PFC-mapped traffic, and the function can be verified using the same.

5 EDA AI fabric app class of service



The EDA ADI fabric app allows the user to configure the various parameters of class of service (COS) that are needed to orchestrate the lossless fabric and provides default values as a baseline.

6 Summary

AI lossless fabric is an important component of optimization of training and inference workloads. The thresholds of congestion management, such as ECN/PFC marking and drop limits, are subject to customer environments and traffic patterns, and hence one single value does not suit all customers. The strategy is to ensure that the ECN triggers before PFC and PFC triggers before drops while ensuring that ECN does not trigger too early, which can cause sub-par performance. This value varies for every customer based on the apps, workloads, and infrastructure that they have deployed and should be optimized on specific customer deployments.