Backend (NCCL) RoCE Fabric- High Level Design

High Performance, Lossless Ethernet Design for AI/HPC Workloads

Akash Patel

TABLE OF CONTENTS

1. CORE PRINCIPLES OF LOSSLESS FABRIC DESIGN

Objective: Guarantee zero packet loss for AI training traffic (e.g., NCCL over RDMA) while maintaining high throughput and low latency.

Key Technologies:

- **RoCEv2** – RDMA over Converged Ethernet v2 (UDP/IP encapsulation for routing).
- **Unnumbered eBGP** – underlay routing
- **PFC (Priority Flow Control)** – Per-priority link-level pause to avoid loss.
- **ECN (Explicit Congestion Notification)** – Congestion signaling without dropping packets.

.

---

2. FABRIC TOPOLOGY CONSIDERATIONS

- **Clos/Leaf-Spine or Fat-Tree** for predictable oversubscription (typically 1:1).
- **Consistent hashing (ECMP)** to evenly distribute RDMA flows.
- **Dual-rail fabrics** for redundancy and increased throughput.
- **Non-blocking core** to minimize congestion points.

3. INTER GPU COMMUNICATION

NVLINK/NVSWITCH (INTRA NODE)

Scope: Inside a single GPU server or chassis.

Function: Provides direct, lossless GPU-to-GPU links at hundreds of GB/s per GPU.

Topology: Fixed “islands” of 8, 16, or 32 GPUs depending on server architecture. Our solution is based on 8 GPUs per node

Design Goals:

- Maximize job placement within a single NVLink island.
- Zero packet drops/replays via vendor firmware and driver alignment.
- Maintain topology consistency across cluster.

Operational Notes:

- Monitor link counters via DCGM.
- Align CPU NUMA to GPU mapping for locality.
- Validate with NCCL intra-node tests.

ROCE

RoCEv2 Backend Fabric (Inter-node)

Scope: Between GPU servers/chassis across the data center.

Function: Carries RDMA traffic (NCCL collectives) over Ethernet in a lossless class.

Topology: Dual-rail L3 Clos using leaf–spine–core; separate physical fabrics for compute and storage.

Design Goals:

- <300 ms failure detection/recovery using BFD with eBGP unnumbered.
- Lossless QoS: Single PFC-enabled traffic class (PCP 3), ECN tuned below PFC XOFF, DCQCN on NICs.
- Wide ECMP groups (64–128) with resilient hashing for stable NCCL flows.

Operational Notes:

- Use GSHUT for maintenance to drain traffic cleanly.
- Maintain *MTU 9216* end-to-end with appropriate FEC for optics.
- Validate with ib_write_bw, ib_read_lat, and nccl-tests across no

4. CONGESTION AVOIDANCE

PFC, DCQCN (WITH ECN) & PFC WATCHDOG

1. PFC – PRIORITY FLOW CONTROL

What it is: IEEE 802.1Qbb.

- A link-level pause mechanism that applies flow control per traffic class instead of pausing the entire link.

Why for RoCE:

- RoCEv2 uses UDP over IP, so there's no built-in TCP congestion control. Without PFC, any loss in the network can cause GPU training stalls or RDMA retransmissions (very costly in HPC).

How it works:

- Switch sends a pause frame only for the lossless queue carrying RDMA traffic (e.g., PCP=3 or DSCP=26/28), while other queues (best effort, storage TCP, etc.) keep flowing.

Design note:

- Enable only on lossless class to avoid head-of-line blocking across all traffic.
- Configure symmetric PFC (both ends).
- Tune pause_quanta to match buffer depth.

2. ECN – EXPLICIT CONGESTION NOTIFICATION

What it is: RFC 3168

- Marks packets instead of dropping them when congestion is detected (via queue thresholds).

Why for RoCE:

- When combined with DCQCN, ECN marks trigger end-to-end rate control in RDMA flows, reducing congestion proactively.

How it works in your backend fabric:

- Switch detects queue depth over a threshold → marks ECN bits in IP header (CE codepoint).
- RDMA endpoints read these marks and slow down send rates (instead of waiting for drops).

Design note:

- Configure per-queue ECN thresholds carefully—too low and you under-utilize, too high and you risk microbursts.

Must be enabled end-to-end (every hop).

3. DCQCN – DATA CENTER QUANTIZED CONGESTION NOTIFICATION

What it is:

- NVIDIA/Mellanox implementation of congestion control for RoCEv2, using ECN marks to adjust RDMA QP send rates.

Why it matters:

- PFC alone prevents drops but doesn't stop queues from filling. DCQCN uses ECN signals to back off before PFC pause storms occur.

How it works:

- Switch ECN-marks packets.
- NIC receives ECN marks → sends Congestion Notification Packets (CNP) to sender.
- Sender NIC rate-limits that RDMA flow.

Design note:

- Enable DCQCN per lossless traffic class in NIC firmware.
- Tune min_rate, rate_increase, rate_decrease parameters for your fabric RTT

4. PFC WATCHDOG

What it is:

- A safeguard that detects and breaks PFC deadlocks caused by bugs or misconfiguration.

Why it's critical:

- PFC deadlocks can bring an entire fabric class to a halt—watchdog forces recovery.

How it works:

- Monitors paused queues for a timeout (e.g., >500ms).
- If timeout hit → flushes queue, drops frames, logs event.

Design note:

- Enable on all switches in lossless queues.
- Tune timeout so it doesn't trigger on normal pauses, only pathological cases.

How They Work Together in Your RoCE Backend

Function	Scope	Prevents	Side Effect if Misused
PFC	Link-level	Packet loss in lossless queues	Head-of-line blocking, deadlocks
ECN	End-to-end	Long-term queue buildup	Under-utilization if threshold too low
DCQCN	Endpoint rate control	PFC storms, persistent congestion	Lower throughput if too aggressive
PFC Watchdog	Switch queue safety	Fabric lock-up from PFC deadlock	Possible packet loss during flush

HOW TO IMPLEMENT (STEP-BY-STEP)

1) Classify traffic (one lossless class only)

- Mark **RoCEv2** as **DSCP 26 (AF31)** → **PCP 3** (example).
- Keep storage/control/management **lossy** (no PFC).

2) MTU & hashing

- Set MTU 9000/9216 end-to-end (hosts + switches).
- Ensure ECMP/LAG hashing has good entropy (5-tuple including UDP).

3) Buffers & PFC thresholds (XOFF/XON)

Goal

Size headroom so a receiver (switch/NIC) can absorb all bytes "in flight" after it sends PFC pause—without loss—then pick sane XOFF/XON marks for the lossless priority queue.

Headroom formula

Headroom≥ (WireRate (bits/s)×ReactionWindow (s))/8 +WorstCaseBurst (bytes)

- ReactionWindow ≈ 25–35 µs (pause frame TX + propagation + upstream stop + drain start)
- WorstCaseBurst: 64–128 KB (multi-sender fan-in, hash collisions, pipeline flush, etc.)

Mapping Headroom → XOFF/XON

- Pick XOFF a bit below headroom to account for jitter/measurement error.
- Pick XON so there's enough drain (XOFF − XON) to clear the queue before senders resume.

Rule of thumb

- XOFF ≈ 70–85% of Headroom
- XON ≈ 35–50% of Headroom
- Drain time ≈ (XOFF − XON) / LineRate (bytes/s)

examples

100 G

- Headroom target: ≈ 450–500 KB
- Suggested marks: XOFF = 320–384 KB, XON ≈ 192 KB

- Drain ≈ 130–190 KB → ~10–15 µs at 100 G)

200 G

- Headroom target: ≈ 900–1000 KB
- Suggested marks: XOFF = 700–850 KB, XON ≈ 400 K
- Drain ≈ 300–450 KB → ~12–18 µs at 200 G)

Notes / definitions

- PFC TX: Receiver (switch/NIC) detects its egress queue for the lossless priority approaching XOFF, then transmits a PFC pause frame upstream.
- Drain time: Time (and bytes) between XOFF and XON while the queue empties.
- Upstream stop: Time for senders to process pause, flush TX pipelines, and cease new frames.
- Worst-case burst: Additional data that can still arrive after pause due to multiple senders, pipeline, and scheduling artifacts (assume 64–128 KB).

Practical tips

- Size per priority (not global buffer).
- Re-verify after changing MTU, number of senders, or ECMP fan-in.
- If using dual-rail fabrics, validate each rail independently.
- Watch tail latency: too-high XOFF/XON can amplify PFC storms; too-low risks drops.

4) ECN/DCQCN first, PFC second

- Enable ECN on the lossless queue to mark before PFC triggers.
- Start Kmin ~10–20% below XOFF; Kmax ~2–3× Kmin but < headroom
- 100G example: Kmin 64 KB, Kmax 192 KB
- 200G example: Kmin 128 KB, Kmax 320 KB
- Enable DCQCN on hosts (ConnectX): conservative target rates so queues stay shallow.

5) Enable PFC on links carrying RoCE

- Turn **PFC on only for that one priority** (e.g., PCP 3). Everything else remains lossy.

6) Enable a PFC watchdog (deadlock breaker)

- Detect continuous pause on the lossless class and **temporarily drop or make that class lossy** to drain.
- Start with **detect 200 ms**, **restore 2 s**, **action: drop** (or “lossy”).

7) Host (NIC) alignment

- Map DSCP 26 → PCP 3.
- Enable RoCEv2 + DCQCN; PFC only on the RoCE priority.
- Verify jumbo MTU and GID/IP config.

8) Validate

- Run incast/NCCL tests at ~90–95% link rate.
- Watch per-priority PFC rx/tx, ECN CE marks, watchdog events, RDMA retrans/timeouts.
- Good sign: steady ECN, near-zero PFC, watchdog never triggers, zero RDMA drops.

EXAMPLE CONFIGS (TEMPLATES; ADAPT TO PLATFORM/VERSIONS)

Cisco NX-OS

```
class-map type qos match-any ROCE-CLASS
  match dscp 26

policy-map type qos ROCE-QOS
  class ROCE-CLASS
    set qos-group 3

policy-map type network-qos ROCE-NQ
  class type network-qos class-default
    mtu 9216
  class type network-qos qos-group 3
    pause no-drop
    pfc-cos 3
    ecn
    ecn mark-probability 100 min-threshold 64K max-threshold 192K
system qos
  service-policy type qos input ROCE-QOS
  service-policy type network-qos ROCE-NQ
# Set XOFF/XON for the no-drop class (ASIC/queue CLI varies by platform)
```

NVIDIA Spectrum (SONiC example for watchdog; Cumulus shown for QoS/PFC)

Cumulus/NCLU (QoS/PFC/ECN):

```
net add dcb pfc priority 3 enable
net add qos map dscp-to-tc 26:3
net add qos ecn enable
net add qos ecn queue 3 min-threshold 64k max-threshold 192k mark-probability 100
net add interface swp1-48 mtu 9216
net pending; net commit
```

SONiC (PFC watchdog):

```
pfcwd start --action drop --restoration-time 2000
pfcwd interval 100
pfcwd start --priority 3 --detection-time 200
```

Arista EOS

```
qos map dscp-to-traffic-class 26 3
priority-flow-control enable
priority-flow-control priority 3 enable
queue-monitor length ecn 3 min-threshold 64 kilobytes max-threshold 192 kilobytes
```

 # Platform-specific queue headroom / xoff/xon commands here

Host (Linux / ConnectX)

```
# Map DSCP 26 → PCP 3
dcb app add dev <ifc> dscp 26 prio 3
```

```
# Enable RoCEv2 + DCQCN (exact knobs vary by NIC fw)
mlxconfig -d <pci> set ROCE_ACCL=1
mlxconfig -d <pci> set DCBX_PFC_MODE=1 ETS_PFC_ENABLE=0x08   # PFC on prio 3
ip link set dev <ifc> mtu 9216
```

5. END-HOST (NIC)

PF, VF, AND SR-IOV

1. PF – Physical Function

- The real PCIe device, as seen by the OS.
- Has full hardware control: can configure QoS, PFC, DCQCN, SR-IOV settings, VLAN mappings, firmware features.
- Each physical NIC port has one PF.
- In Mellanox/NVIDIA terms:
    - The PF is where you run mlxconfig, enable PFC, set DSCP→PCP mappings, etc.
    - Whatever you configure here can be pushed down to VFs.

---

2. VF – Virtual Function

- A lightweight “slice**”** of the physical NIC created by SR-IOV.
- Appears as an independent PCIe device to a VM, container, or tenant OS.
- Has its own MAC, queue pairs, interrupts, but cannot change deep hardware policy (that’s the PF’s job).
- In RoCE terms:
    - A VF can do RDMA (with QoS and congestion control inherited from the PF).
    - You can map different VFs to different VLANs/priorities if the PF is set up to allow it.

---

3. SR-IOV – Single Root I/O Virtualization

- PCIe standard that allows one PF to present multiple VFs to the system.
- Lets each VF bypass the host kernel’s slow path and talk directly to NIC hardware (hardware offload).
- Benefits in AI/ML and HPC fabrics:
- Low latency (no software switching in the hypervisor).
- High throughput (direct DMA from VF to NIC queues).
- Hardware isolation per VF (each VM/container sees “its own” NIC).

CONFIGURATION

Two clear classes at the NIC:

- **Lossless class**: RDMA QPs for RoCEv2 traffic → Priority 3 (example).
- **Lossy class**: Everything else (TCP/IP, management).

Example of **NVIDIA/Mellanox ConnectX-4 Lx and newer** (ConnectX-4/5/6/6 Dx/7)

1. Classify RDMA traffic

Map RoCEv2 DSCP (26) to lossless PCP (3)

- dcb app add dev eth0 dscp 26 prio 3

2. Map other traffic to lossy priorities
   Example: map management and TCP/IP to prio 0 (lossy)
   - dcb app add dev eth0 dscp 0 prio 0

3. Enable PFC only on lossless prio
   - mlxconfig -d <pci_bus> set DCBX_PFC_MODE=1 ETS_PFC_ENABLE=0x08   # bit mask for prio 3

4. Confirm mappings
   - dcb app show dev eth0

NVIDIA BLUEFIELD-3 DPUS

Nvidia Bluefield 3 (BF3) use essentially the same RoCEv2, DCQCN, and per-priority PFC configuration model as the ConnectX NICs, because the DPU's embedded networking pipeline is ConnectX-7–class under the hood. Here is the comparison

Feature / Step	ConnectX-5/6/7 NIC	BlueField-3 DPU
Where you run commands	Host OS (x86/Arm server)	On the DPU Arm cores (Ubuntu/DPU-OS) or via DOCA
Firmware tool	mlxconfig (part of MFT package)	mlxconfig on DPU OS (same syntax)
Verify RoCE support	`mlxconfig -d <pci> query	grep ROCE`
Enable RoCEv2	mlxconfig -d <pci> set ROCE_ACCL=1 ROCE_IPV6_ONLY=0	Same
Enable DCQCN	mlxconfig -d <pci> set CNP_PACING_MODE=1	Same
Map DSCP→PCP	On host: dcb app add dev <ifc> dscp 26 prio 3	On DPU OS: same dcb or tc command for PFs; for VFs, map in PF and propagate to VF
Enable PFC (lossless class only)	mlxconfig -d <pci> set DCBX_PFC_MODE=1 ETS_PFC_ENABLE=0x08 (bitmask for PCP 3)	Same on DPU OS; applies to PF; ensure VF inherits or override in VF config
ECN thresholds (switch side)	Kmin ≈ 64 KB, Kmax ≈ 192 KB for 100 G (below XOFF)	Same — BlueField NIC pipeline behaves like ConnectX-7; thresholds set on switch, not DPU

PFC watchdog	On switch OS (e.g., SONiC/Arista/Cisco); not NIC-local	Same — controlled in the switch, not the DPU
SR-IOV mapping	PF owns PFC/DCQCN config; VFs inherit	PF config on DPU controls VF behavior; can also isolate via DOCA Flow
Telemetry	ethtool -S, mlxstat on host	ethtool -S, mlxstat on DPU OS; DOCA Telemetry for per-priority counters
Special notes	Runs entirely in host NIC	DPU may terminate control/management traffic locally; ensure only RDMA data path is in lossless class

---

6. VALIDATION & ACCEPTANCE TESTING

OBJECTIVE

Verify that the backend RoCE fabric (NCCL traffic domain) meets design intent for lossless transport, resiliency, and performance under operational conditions before being placed into production.

SCOPE

Covers dual-rail backend fabric interconnecting GPU nodes across 98 pods / 392 SU. Tests validate:

- Correct physical & logical connectivity (port, VLAN/VRF assignments).
- RoCEv2 lossless behavior (PFC, ECN, DCQCN tuning).
- BGP underlay stability and fast reconvergence.
- NCCL collective communication performance and path diversity.

TEST CATEGORIES

1. PHYSICAL & LOGICAL VALIDATION

Checkpoint	Method	Acceptance Criteria
Port link-up / MTU	ethtool, show interfaces	All fabric links operational at 400G; MTU=9216

VLAN / VRF mapping	net show bridge vlan, ip vrf show	Backend VRF correctly applied to all RoCE interfaces
Optics & FEC	ethtool -m, NOS optics diag	Optics match design spec; FEC enabled as per vendor guidance

---

2. CONTROL PLANE VALIDATION

Checkpoint	Method	Acceptance Criteria
eBGP unnumbered peering	net show bgp summary	All leaf–spine–core adjacencies up
BFD timers	net show bfd	Sub-300 ms failure detection
ECMP path count	net show bgp ipv4 unicast	ECMP width matches ASIC limit (≥64)
Graceful maintenance	Simulated GSHUT	Traffic drains cleanly with no packet loss

---

3. DATA PLANE VALIDATION (ROCEV2 LOSSLESS)

Checkpoint	Method	Acceptance Criteria
PFC pause frame counters	ethtool -S or mlxreg	No sustained pause storm; counters stable. Mlxreg for advance troubleshooting
ECN marking	ethtool -S, telemetry collector	ECN marks in acceptable range under load
DCQCN behavior	Flow replay under congestion	Congestion window ramps down/up smoothly

---

4. PERFORMANCE BENCHMARKING

Tool	Purpose	Acceptance Criteria
ib_write_bw / ib_write_bw -d mlx5_0 -F/ ib_read_lat	RDMA microbenchmarks	Bandwidth ≥ link-rate efficiency; latency < design target ** for ConnectX-5(6/7) install perftest using apt-get (ubuntu/Debian) or yum (RHEL/Cent-OS) before writing this If BlueField3, then you can directly SSH into it

nccl-tests (all_reduce, all_gather)	Application-level NCCL validation	Throughput within 5% of lab reference; no retries triggered
Multipath diversity test	Parallel NCCL jobs	Flows spread evenly across ECMP paths

---

5. STRESS & FAILOVER TESTING

Scenario	Method	Acceptance Criteria
Link failure	Disable leaf–spine link	No job failure; reconvergence < 300 ms
Node reboot	Restart GPU node	NCCL jobs recover without hang
Multi-link failure (within ECMP width)	Disable multiple spines	Bandwidth reduction proportional; no packet loss in remaining paths

---

DOCUMENTATION & SIGN-OFF

- All test results recorded in Validation Report with:
  - Test date/time
  - Node pair / fabric rail tested
  - Tool output logs
  - Pass/Fail summary

---

7. OPERATIONAL BEST PRACTICES

- Keep firmware and switch OS updated for latest RDMA/PFC fixes.
- Monitor per-priority buffer utilization, ECN marks, and pause events.
- Avoid enabling PFC globally—scope it tightly to RDMA priorities.
- If congestion hotspots appear, evaluate load balancing (e.g., adaptive routing).

APPENDIX A- BGP DESIGN AND SAMPLE CONFIGURATION)

Design intention (why)	**Implementation on Spectrum/Cumulus**

Fast failure detection < 200–300 ms	BFD on every eBGP peer: tx/rx 50–100 ms, multiplier 3
Blast radius containment	Dual rails, separate ASNs per rail; no interconnect between rails
L3 CLOS with wide load-balancing	eBGP unnumbered on all leaf↔spine↔core links; ECMP 64; multipath-relax
Minimal flow churn on path changes	Enable resilient/consistent ECMP hashing (if exposed); include UDP L4 ports in hash
Preserve dataplane during control restarts	BGP Graceful Restart (GR) on all tiers (Leaf/Spine/Core)
Immediate removal of failed paths	Fast external fallover; drop eBGP adjacency on link-down
Maintenance without packet loss	GSHUT (graceful-shutdown) before interface or process work
Lossless only for RoCE	One lossless class (PCP 3) mapped from DSCP (26/40/44 → 3); PFC only on that class
Keep queues short, avoid PFC churn	ECN on TC3 with Kmin/Kmax below PFC XOFF; hosts/DPUs run DCQCN
Jumbo & physical hygiene	MTU 9216 end-to-end; optics-appropriate FEC; watch micro-loss
Simple, scoped routing	Redistribute only SVIs (leaf) and loopbacks; no non-underlay routes
Observability	Stream per-priority PFC/ECN + queue stats; alarms for BFD down, ECMP shrink, prefix churn

---

SAMPLE CONFIGS — DUAL-RAIL ROCE BACKEND (RAIL-A SHOWN)

Mirror the same pattern for **Rail-B** with its **own ASN**, VLAN/SVI space, and physical links.
Backend VRF: vrf-backend-nccl. Lossless class: **TC/PCP 3**. DSCP for RoCE: **26** (or **40/44** if you prefer). This is using Nvidia Cumulus (NCLU) for illustration purpose, but similar configuration can be applied to FRR or other vendor switching solution

A) LEAF (RAIL-A) — NCLU

Assumptions

- Leaf-A ASN 65011; loopback 10.0.11.1/32
- Host RDMA VLAN/SVI (this SU/rail): VLAN 1005, 10.100.5.1/26, MTU 9216
- Uplinks to spines: swp33–swp34 (routed)

- Host ports: swp1–swp32(access VLAN 1005)

System & Loopback

net add loopback lo ip address 10.0.11.1/32

```
# VRF & SVI (backend RDMA for this SU/rail)

net add vlan 1005
net add interface vlan1005 ip address 10.100.5.1/26
net add interface vlan1005
net add interface vlan1005 mtu 9216

# Host access ports

net add bridge bridge vids 1005
net add bridge bridge ports swp1-32
net add interface swp1-32 bridge access 1005
net add interface swp1-32mtu 9216



# Uplinks to Spines (routed, unnumbered eBGP)

net add interface swp33-64 mtu 9216

# QoS / Lossless isolation (RoCE only)
net add dcb pfc priority 3 enable
net add qos map dscp-to-tc 26:3 40:3 44:3
net add qos ecn enable
net add qos ecn queue 3 min-threshold 64k max-threshold 192k mark-probability
100



# BGP core: eBGP unnumbered + BFD + ECMP + GR + GSHUT (ready)
net add bgp autonomous-system 65011
net add bgp router-id 10.0.11.1
net add bgp bestpath as-path multipath-relax
net add bgp graceful-restart                    # if control plan
crash/restart , the data plane continue to work (FIB)
net add bgp graceful-shutdown             # enable/disable during drains ,
used during maintenance window


net add bgp neighbor SPINES peer-group
net add bgp neighbor SPINES remote-as external
net add bgp neighbor SPINES timers 60 180

for i in {33..64}; do
  net add bgp neighbor swp$i interface peer-group SPINES
  net add bgp neighbor swp$i bfd
done


# Address-family
```

```
net add bgp address-family ipv4 unicast maximum-paths 64
net add bgp address-family ipv4 unicast redistribute connected

# Commit

net commit
```

Operational drain (GSHUT)

```
# Start graceful shutdown to drain traffic
net add bgp graceful-shutdown
net commit

# After maintenance:
net del bgp graceful-shutdown
net commit
```

B) SPINE (RAIL-A) — NLCU

Assumptions

- Spine ASN 65100; loopback 10.0.100.1/32
- Downlinks to leaves: swp1–swp32 (routed)
- Uplinks to core: swp49–swp64 (routed)

```
# System & Loopback

net add loopback lo ip address 10.0.100.1/32



# Links

net add interface swp1-64 mtu 9216



# QoS policy (mirror lossless class across fabric)
net add dcb pfc priority 3 enable
net add qos map dscp-to-tc 26:3 40:3 44:3
net add qos ecn enable
net add qos ecn queue 3 min-threshold 64k max-threshold 192k mark-probability 100



# BGP core

net add bgp autonomous-system 65100
net add bgp router-id 10.0.100.1
net add bgp bestpath as-path multipath-relax
net add bgp graceful-restart
net add bgp graceful-shutdown
```

```
# Peer-groups
net add bgp neighbor LEAFS peer-group
net add bgp neighbor LEAFS remote-as external
net add bgp neighbor LEAFS timers 60 180

net add bgp neighbor CORES peer-group
net add bgp neighbor CORES remote-as external
net add bgp neighbor CORES timers 60 180



# Attach downlinks to LEAFS + BFD
for i in {1..32}; do
  net add bgp neighbor swp$i interface peer-group LEAFS
  net add bgp neighbor swp$i bfd
done



# Attach uplinks to CORES + BFD
for i in {33..64}; do
  net add bgp neighbor swp$i interface peer-group CORES
  net add bgp neighbor swp$i bfd
done



# Address-family
net add bgp address-family ipv4 unicast maximum-paths 64
net add bgp address-family ipv4 unicast redistribute connected

net commit
Operational drain (GSHUT)
net add bgp graceful-shutdown
net commit

# remove after:
net del bgp graceful-shutdown
net commit
```

C) CORE (RAIL-A) — NCLU

Assumptions

- Core ASN 65200; loopback 10.0.200.1/32
- Downlinks to spines: swp1-swp64 (routed)
- Core carries underlay only (no SVIs); optional service loopbacks (telemetry, etc.) are fine to advertise
- Same QoS policy (TC3 lossless) to keep PFC semantics consistent across the fabric

```
# System & Loopback
```

```
net add loopback lo ip address 10.0.200.1/32

# Downlinks to Spines with  Breakout each 800G port into 2x400G
for p in {1..64}; do
    net add interface swp$p breakout 2x400G
done

net pending
net commit

# Set MTU for each breakout child port
for p in {1..64}; do
    for s in 0 1; do
        net add interface swp${p}s${s} mtu 9216
    done
done

net add dcb pfc priority 3 enable
net add qos map dscp-to-tc 26:3 40:3 44:3
net add qos ecn enable
net add qos ecn queue 3 min-threshold 64k max-threshold 192k mark-probability
100

# BGP core
net add bgp autonomous-system 65200
net add bgp router-id 10.0.200.1
net add bgp bestpath as-path multipath-relax
net add bgp graceful-restart
net add bgp graceful-shutdown

# Peer group for spines
net add bgp neighbor SPINES peer-group
net add bgp neighbor SPINES remote-as external
net add bgp neighbor SPINES timers 60 180

# Attach all spine-facing interfaces + BFD
for p in {1..64}; do
  for s in 0 1; do
    net add bgp neighbor swp${p}s${s} interface peer-group SPINES
    net add bgp neighbor swp${p}s${s} bfd
  done
done

# Address-family
net add bgp address-family ipv4 unicast maximum-paths 64
# Only advertise loopbacks (and any explicit service loopbacks)
net add bgp address-family ipv4 unicast redistribute connected

net commit
Operational drain (GSHUT)
net add bgp graceful-shutdown
net commit
# remove after:
net del bgp graceful-shutdown
net commit
```

RAIL-B -CONFIGURATION

- Rail-B: duplicate each block with its own ASNs (e.g., Leaf-B 65012, Spine-B 65101, Core-B 65201), its own VLAN/SVI space (e.g., 10.100.6.0/26 for the Leaf-B SU), and separate physical links.
- No inter-rail connectivity: rails are independent failure domains. Any cross-rail services (rare) should live outside the RoCE underlay.

** *Same design and configuration with different Loopback & /26 subnets (NVM-of RoCE) would be applicable for Storage Fabric*

QUICK VALIDATION (ALL TIERS)

```
# BGP/BFD health
net show bgp summary
net show bfd peers

# ECMP width / routes
net show route 10.100.5.0/26
net show route summary

# QoS / lossless signals
sudo ethtool -S swp1 | egrep -i 'pfc|prio|pause'
sudo cl-qos-show
```

DESIGN AND CONFIGURATION OPTIMIZATION WITH CUMULUS

BGP unnumbered

This example leverages BGP Unnumbered to optimize the configuration and save thousands of /31 IP addresses.

- No IPv4 / IPv6 global address needed on the interface
- The switch interface only uses IPv6 link-local addresses (FE80::/10), which are automatically created for every L3 interface.
- These link-local addresses exist without you explicitly assigning anything.
- When you do bfd on that neighbor, it also uses IPv6 link-local over the same interface.

BGP external

Normal BGP requires you to define the remote AS number as an example *net add bgp neighbor swp33 remote-as 65102*

- However, in our sample configuration, remote-as external tells FRR:
- “This neighbor’s AS number is different from local AS. Learn it dynamically during session establishment.”
- This is valid for eBGP sessions only, because eBGP expects AS numbers to differ.
- The local router’s ASN is already set (e.g., net add bgp autonomous-system 65101).
- On TCP session open, FRR inspects the remote’s BGP Open message, reads the ASN from the packet, and uses it without you hardcoding it