

The background features a vibrant, abstract design with a color gradient from dark blue on the left to bright yellow and white on the right. The design consists of overlapping, wavy horizontal bands and a radial pattern of lines emanating from a bright white point on the right side, creating a sense of motion and energy.

CISCO *Live!*

Let's go



The bridge to possible

Network Best Practices for Artificial Intelligence Data Centre

Nemanja Kamenica, Technical Marketing Engineer

Agenda

- Why AI is important today and, in the future
- Network For AI Cluster
- Automation and Visibility
- The Blueprint For Today

Why AI is important today and, in the future



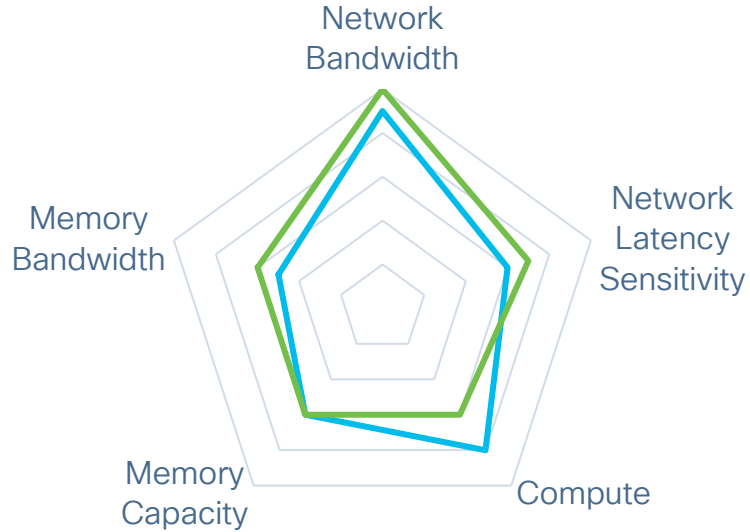
AI/ML Can Help Many Industries

Healthcare	Financial Services	Public Sector	Media and Entertainment	Manufacturing	Retail
Medical Risk Prediction	High Frequency Trading Analysis	Intelligent Public Transport	Speech Recognition	Visual Inspection	Personalized Recommendation
Early diagnostics	Quant Research	Security Log Analytics	Natural Language Processing	Anomaly Detection	Demand Forecasting
Medical Research	Fraud and Risk Analytics	Disaster Recovery Assistance	Content Classification	Asset Management	Visual Search

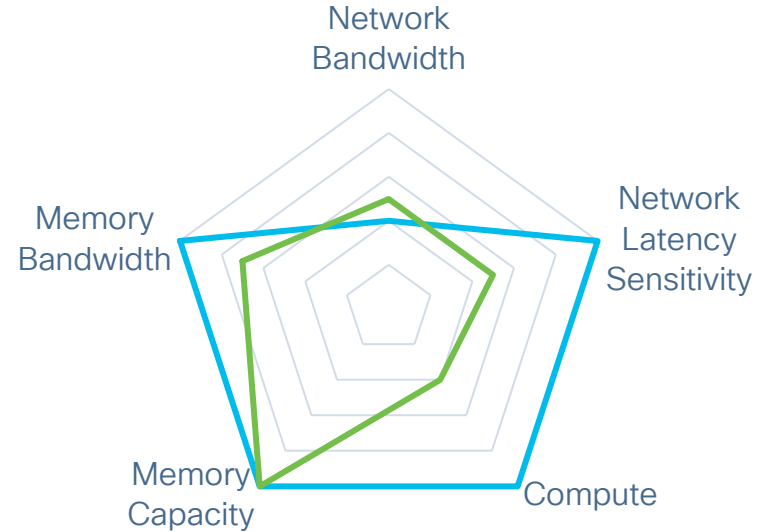
AI cluster types and interaction with network

	Distributed Training	Production Inference
Node to Node Bandwidth	High	Low
Key Metric	Training time of a model	High Availability and Latency
Operational Mode	Model training is offline	Usually online, requires real time response
Infrastructure requirement	Large network with many GPU/CPU hosts	Smaller network with mid size of CPU/GPU hosts

Training vs Inference



— LLM Trainig — Ranking Training



— LLM Inference — Ranking Inference

Large Scale Distributed Training

- Key Challenge of Training Cluster
 - Model Doubles every 2 months
 - Bigger model, higher accuracy
 - Most common single training runs on 512 GPUs

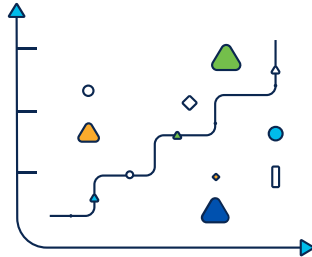


- Cluster Key Components
 - Compute Nodes
 - Network
 - Distributed File System/Storage
 - Job Scheduling and Orchestration
 - Software Framework for AI model



Large Scale Distributed Training

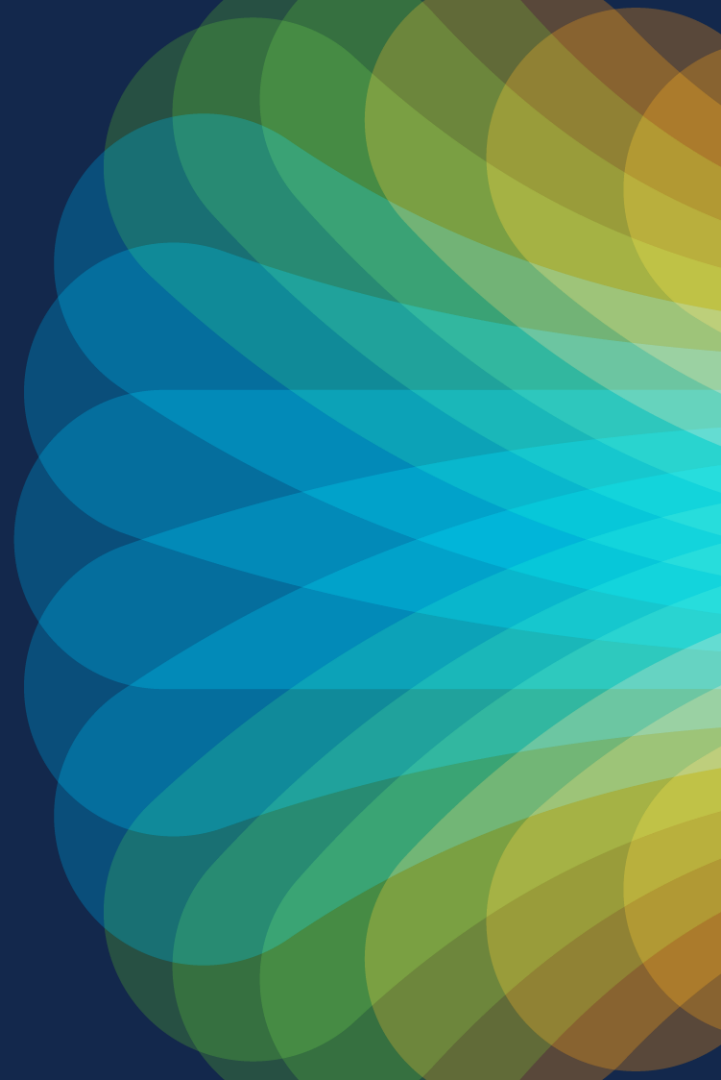
- Key Challenge of Training Cluster
 - Model Doubles every 2 months
 - Bigger model, higher accuracy
 - Most common single training runs on 512 GPUs



- Cluster Key Components
 - Compute Nodes
 - Network
 - Distributed File System/Storage
 - Job Scheduling and Orchestration
 - Software Framework for AI model



Network For AI Cluster



AI Training Network



For RoCEv2 transport the network must provide high throughput and low latency by passing CPU/GPU

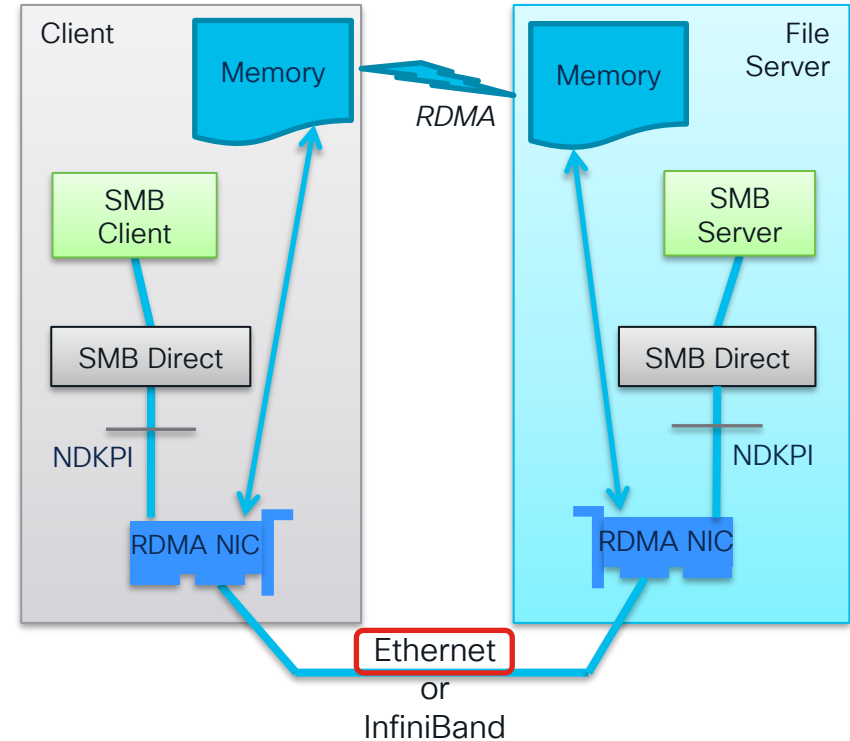


The Cisco Nexus 9000 switches, provide the required low latency, and with up to 25.6Tbps of bandwidth per ASIC, to satisfy AI/ML clusters requirements

Shipping feature set with many customers in production

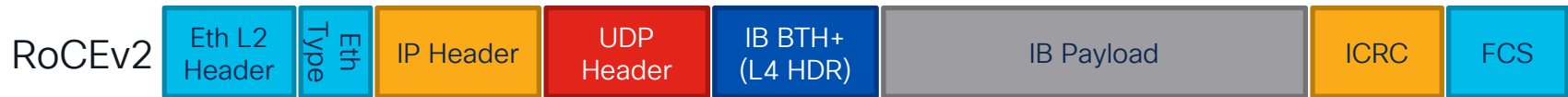
RDMA – Remote Data Memory Access

- Allows application software to communicate directly with the hardware (RDMA NIC)
- Bypasses OS stack
- RDMA delivers, low latency, high throughput, zero copy capabilities
- RDMA Hardware Technologies
 - RoCE: RDMA over Converged Ethernet
 - iWARP: RDMA over TCP/IP
 - Infiniband



RoCEv2 - Basics

- Extension of RoCE protocol that involves a simple modification of the RoCE packet format
- Carry IP header and UDP header that serves as a stateless encapsulation layer for RDMA transport over IP



Source: https://en.wikipedia.org/wiki/RDMA_over_Converged_Ethernet



RoCEv2

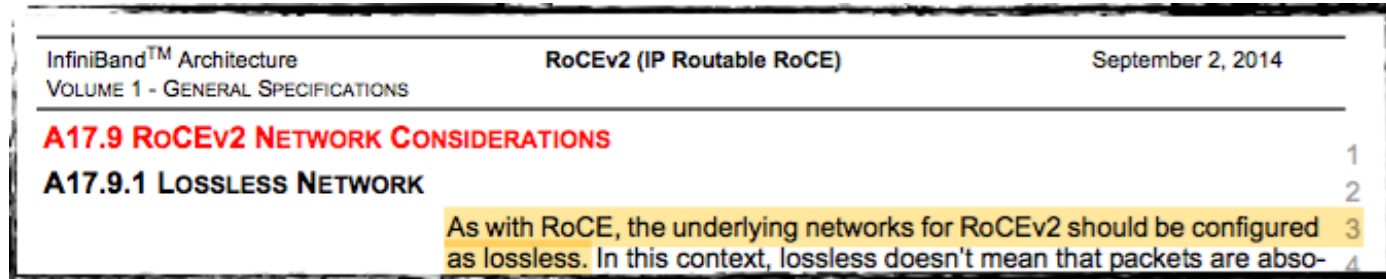
- Uses well-known UDP Destination Port (dport) value 4791
- UDP Source Port (sport) serves as opaque flow identifier that can be used by networking infrastructure for packet forwarding optimizations (e.g., ECMP)
- Supports both IPv4 and IPv6
- Makes use of ECN field in IPv4/6 header for signaling of congestion

CISCO *Live!*

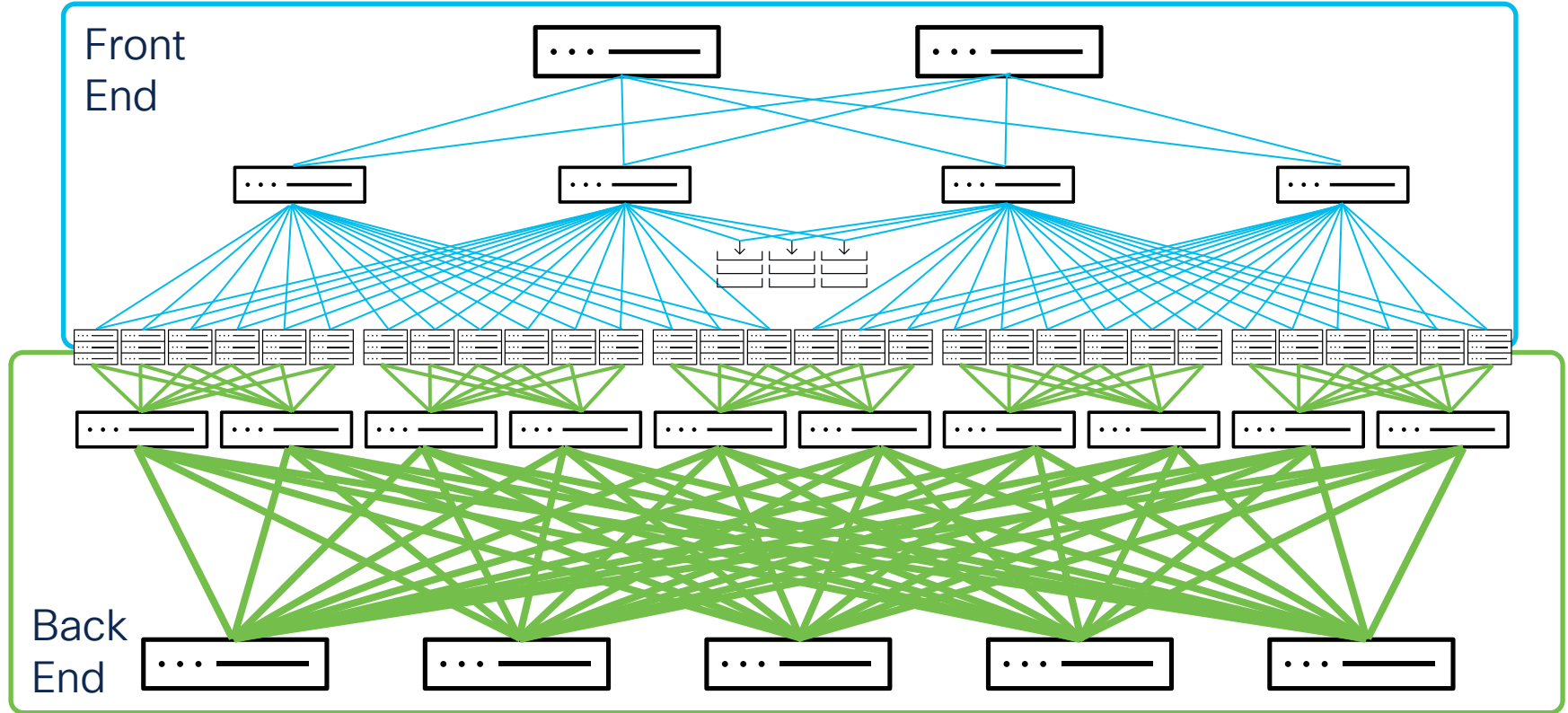


RoCEv2 End-To-End Lossless Behavior

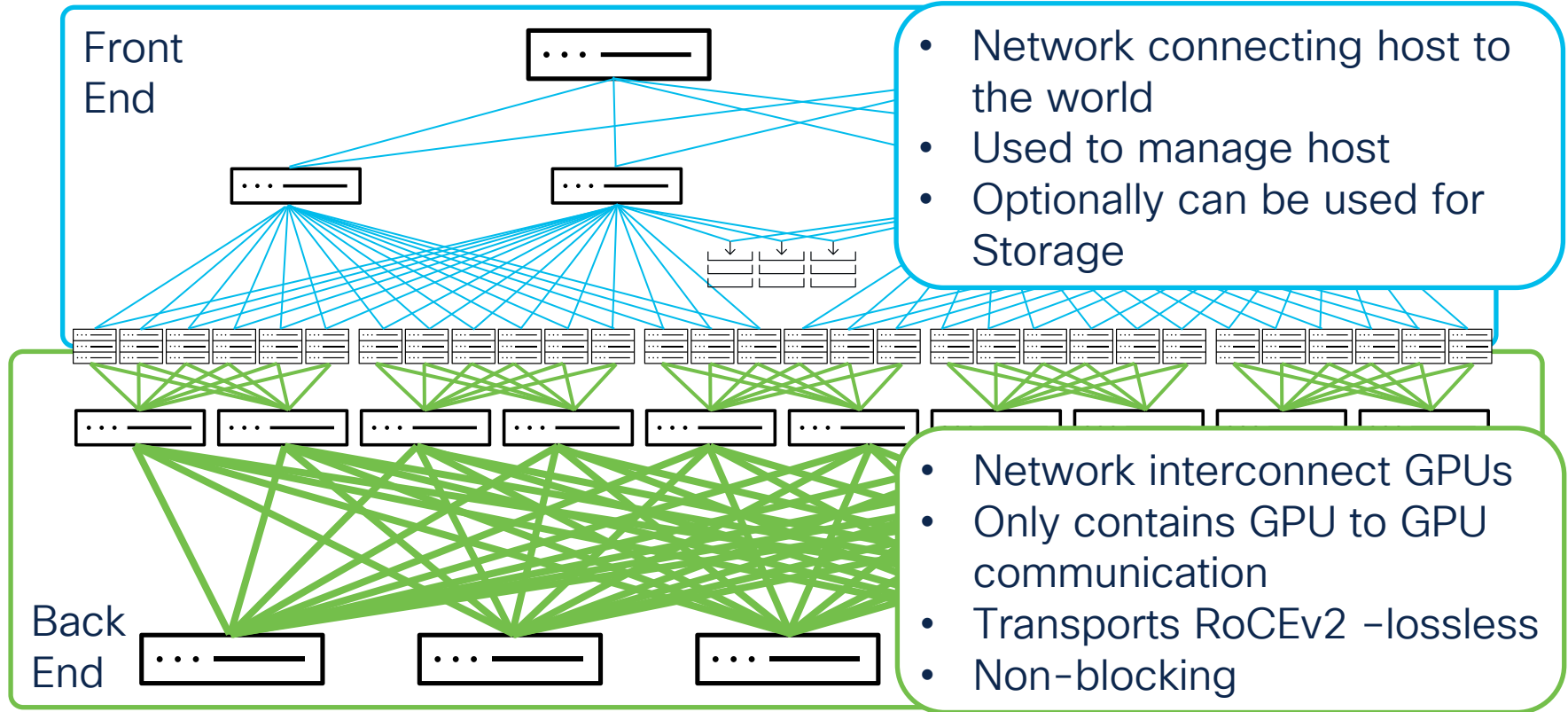
- Requires PFC to be enabled for RoCEv2 transport
- Traffic priority to be preserved between Layer 2 and Layer 3 network
 - Packet/Flow identification follows standard practices of IP/Ethernet networks (i.e., DSCP/802.1Q)
- ECN marking (WRED or consider AFD)
- Configure ETS
 - 802.1Qaz ETS



Non-blocking Network

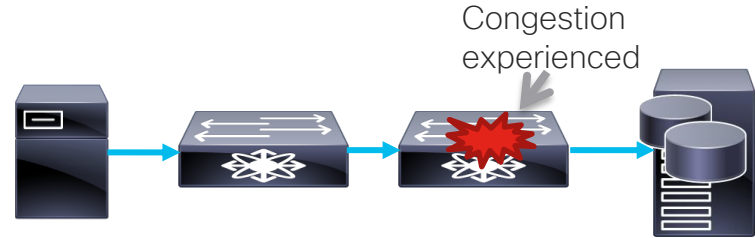


Non-blocking Network



Explicit Congestion Notification (ECN)

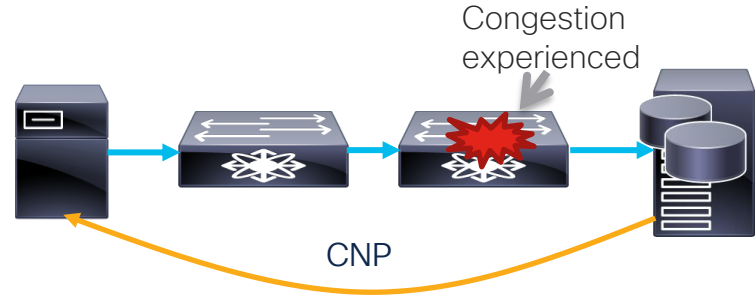
- IP Explicit Congestion Notification (ECN) is used for congestion notification.
- ECN enables end-to-end congestion notification between two endpoints on IP network
- ECN uses 2 LSB of Type of Service field in IP header



ECN	ECN Behavior
00	Non ECN Capable
10	ECN Capable Transport (0)
01	ECN Capable Transport (1)
11	Congestion Encountered

Explicit Congestion Notification (ECN)

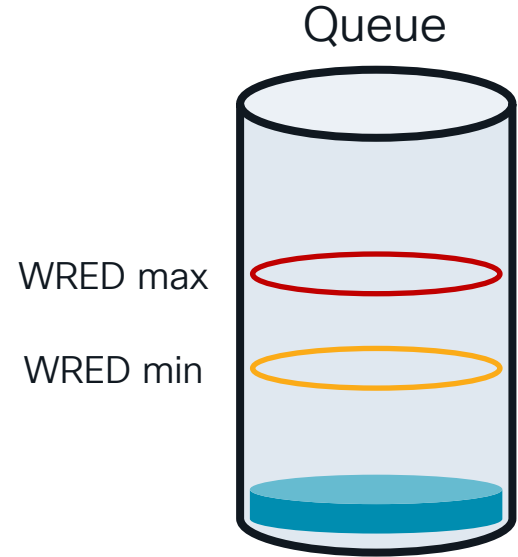
- IP Explicit Congestion Notification (ECN) is used for congestion notification.
- ECN enables end-to-end congestion notification between two endpoints on IP network
- ECN uses 2 LSB of Type of Service field in IP header
- In case of congestion, ECN gets transmitting device to reduce transmission rate using Congestion Notification Packet (CNP) without pausing traffic.



ECN	ECN Behavior
00	Non ECN Capable
10	ECN Capable Transport (0)
01	ECN Capable Transport (1)
11	Congestion Encountered

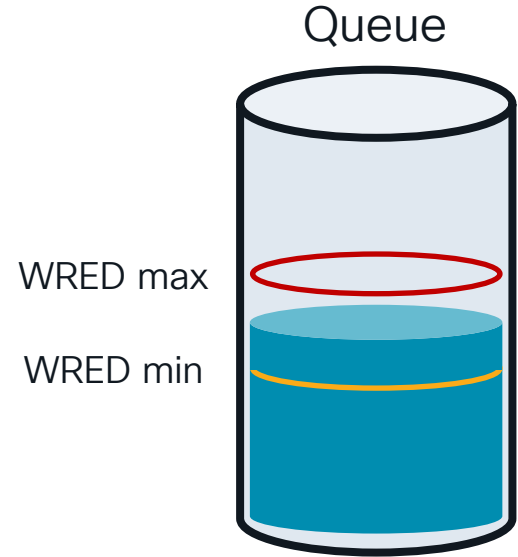
How does WRED ECN work?

- WRED (Weighted Random Early Detection) is used to signalize severity of congestion
- ECN is not marked when buffer usage is below WRED min threshold



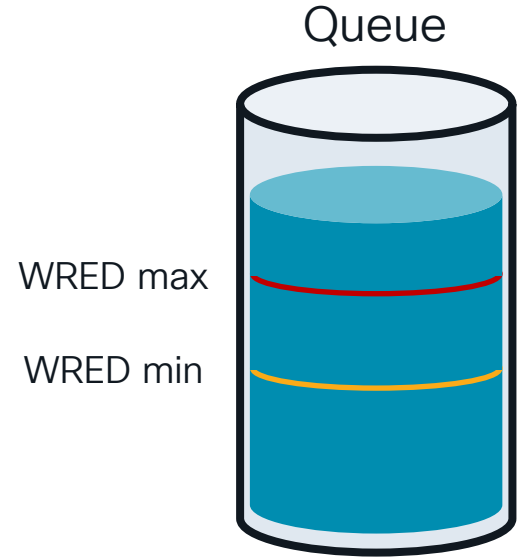
How does WRED ECN work?

- WRED (Weighted Random Early Detection) is used to signalize severity of congestion
- ECN is not marked when buffer usage is below WRED min threshold
- When buffer usage is minimal threshold, Congestion Encountered will be marked on N number of randomly selected packets (probability parameter)

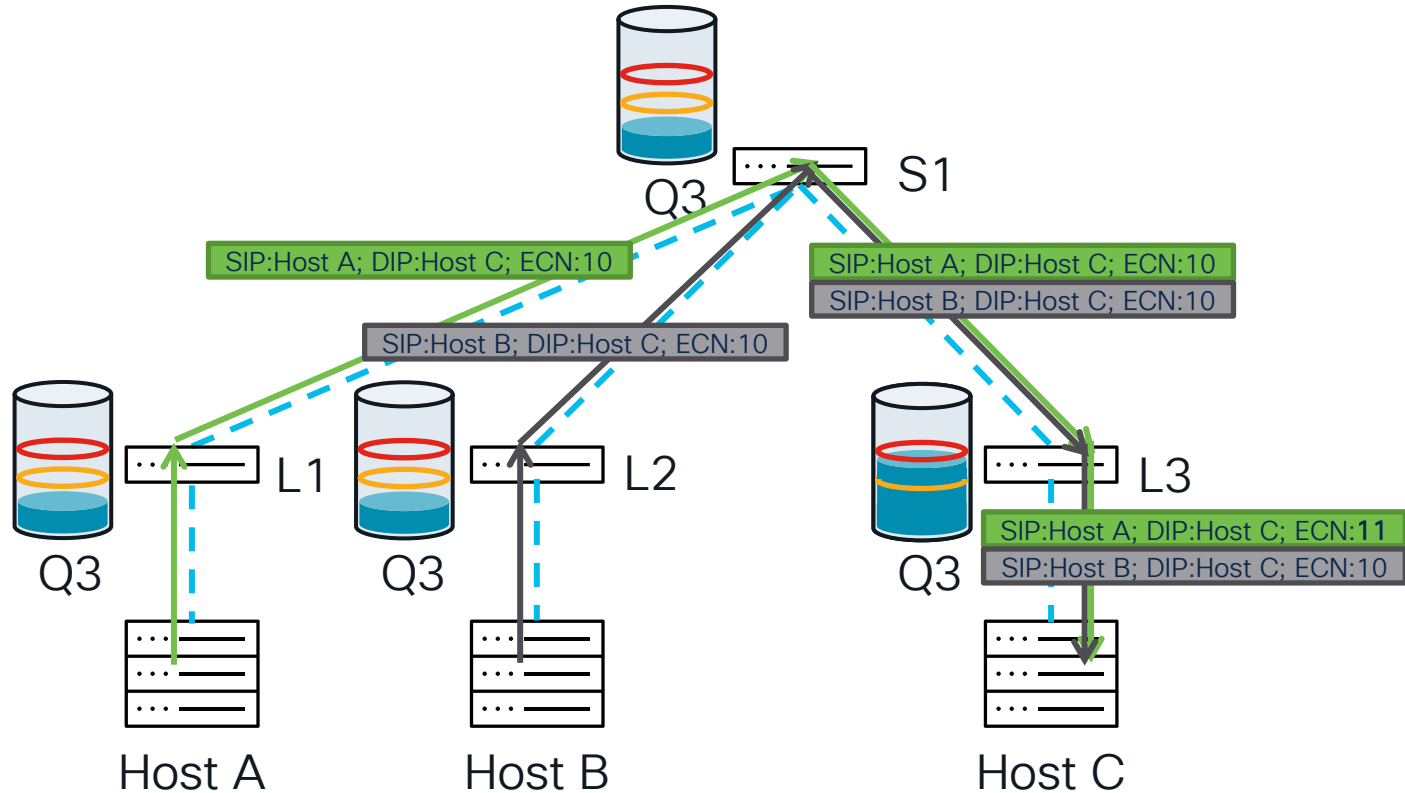


How does WRED ECN work?

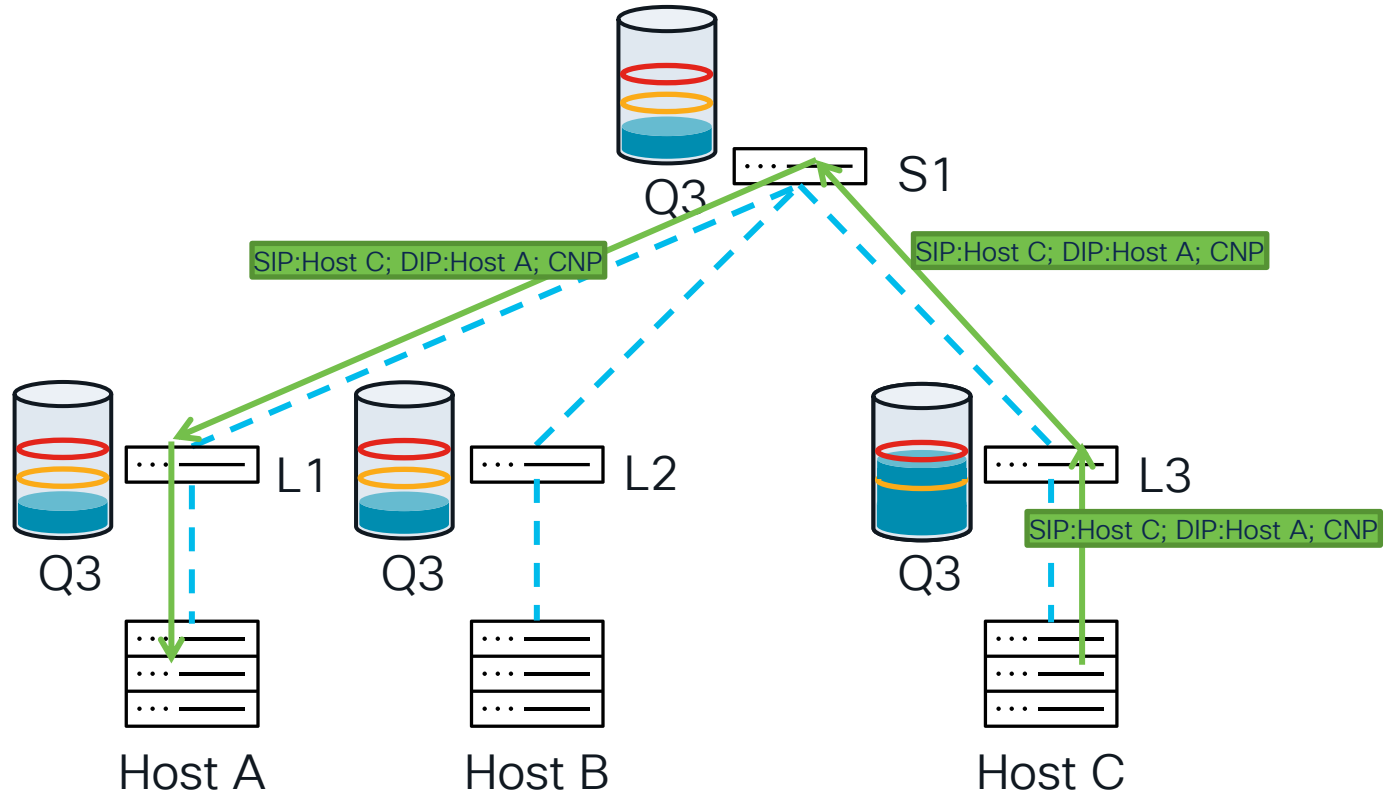
- WRED (Weighted Random Early Detection) is used to signalize severity of congestion
- ECN is not marked when buffer usage is below WRED min threshold
- When buffer usage is minimal threshold, Congestion Encountered will be marked on N number of randomly selected packets (probability parameter)
- After buffer usage crosses MAX threshold, every ECN capable packet will be marked with Congestion Encountered



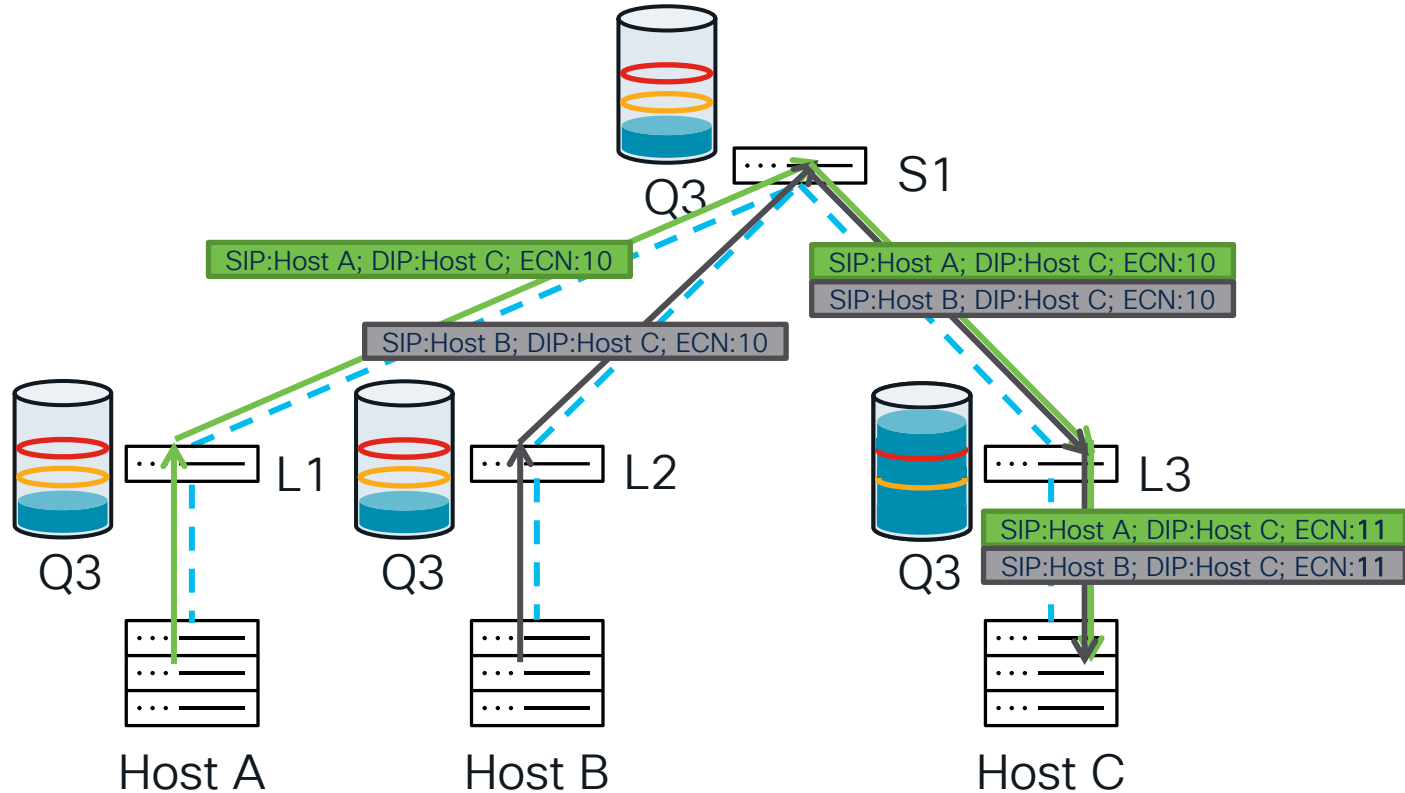
Explicit Congestion Notification – End to End



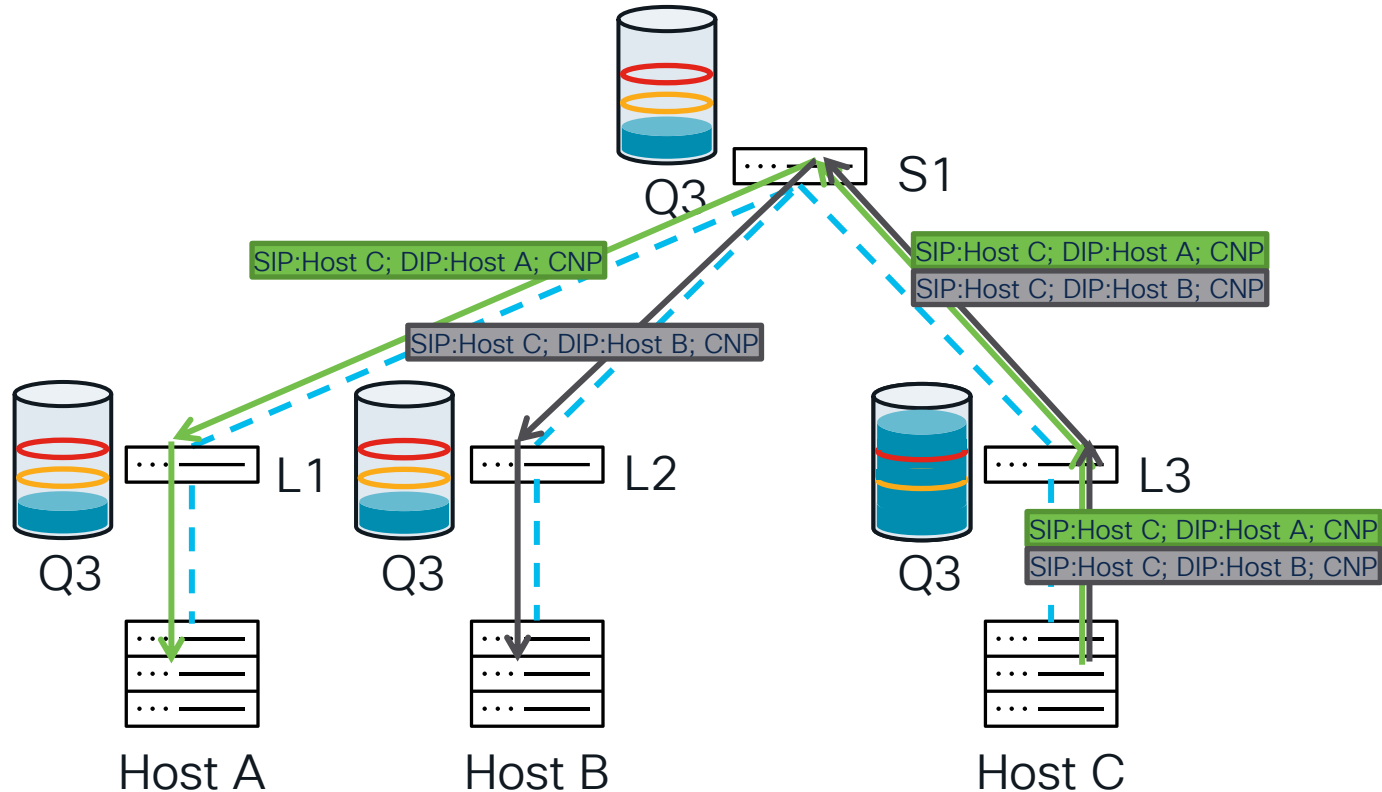
Explicit Congestion Notification – End to End



Explicit Congestion Notification – End to End

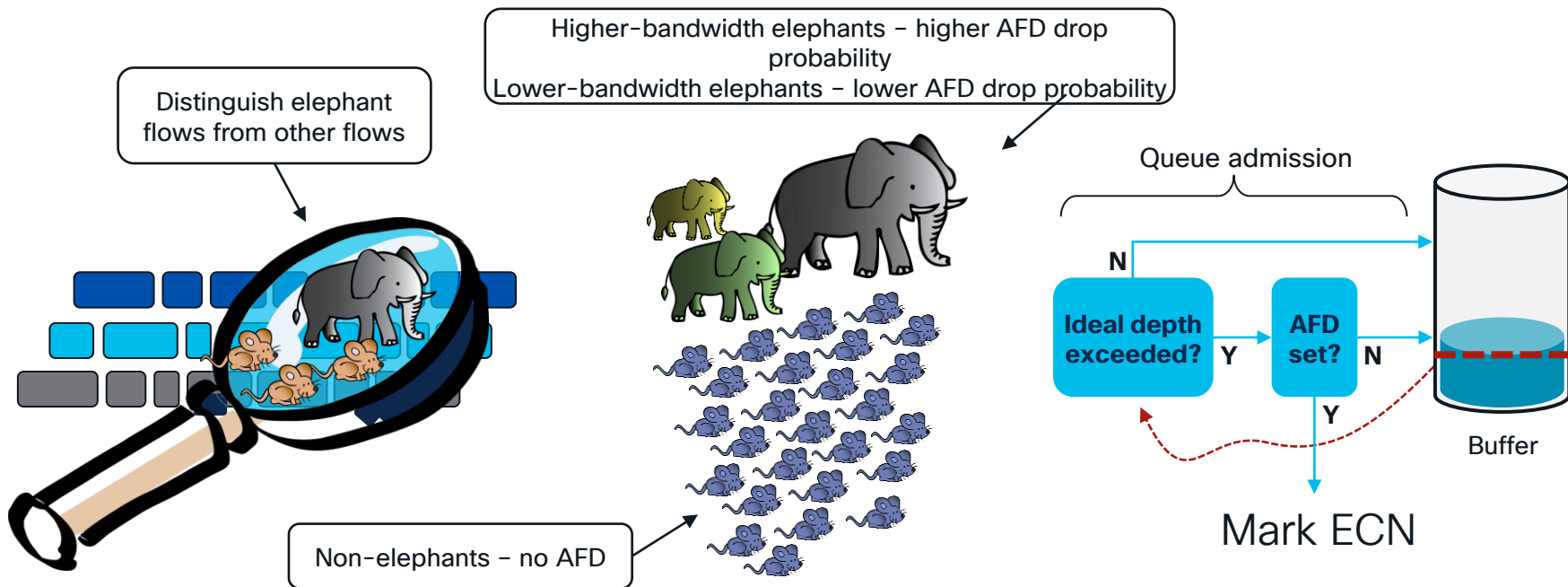


Explicit Congestion Notification – End to End



Approximate Fair Drop (AFD)

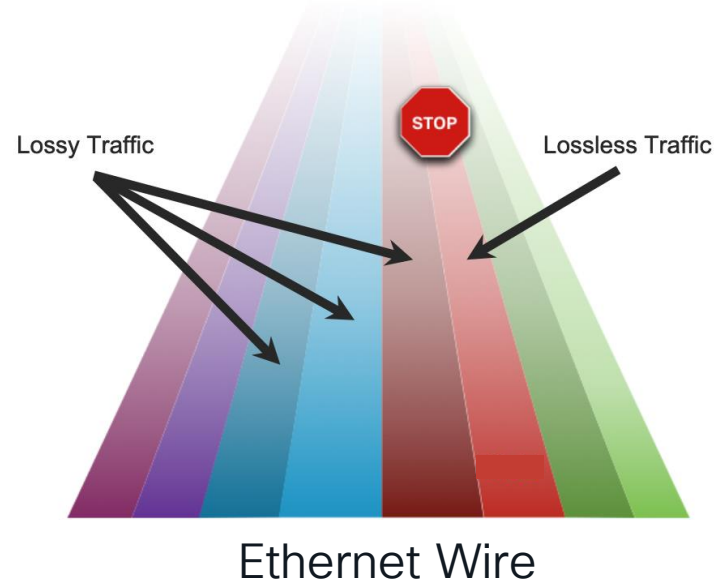
Maintain throughput while minimizing buffer consumption by elephant flows – **keep buffer state as close to the ideal as possible**



Priority Flow Control

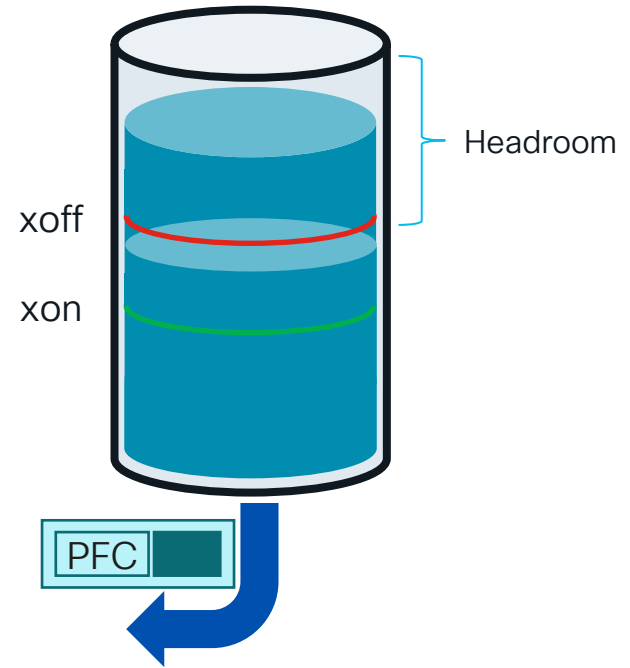
Flow Control Mechanism – 802.1Qbb

- A.k.a "Lossless Ethernet"
- PFC enables Flow Control on a Per-Priority basis
- PFC is also called Per-Priority-Pause
- Therefore, we have the ability to have lossless and lossy priorities at the same time on the same wire
- Allows traffic to operate over a lossless priority independent of other priorities
- Other traffic assigned to other priority will continue to transmit and rely on upper layer protocols for retransmission

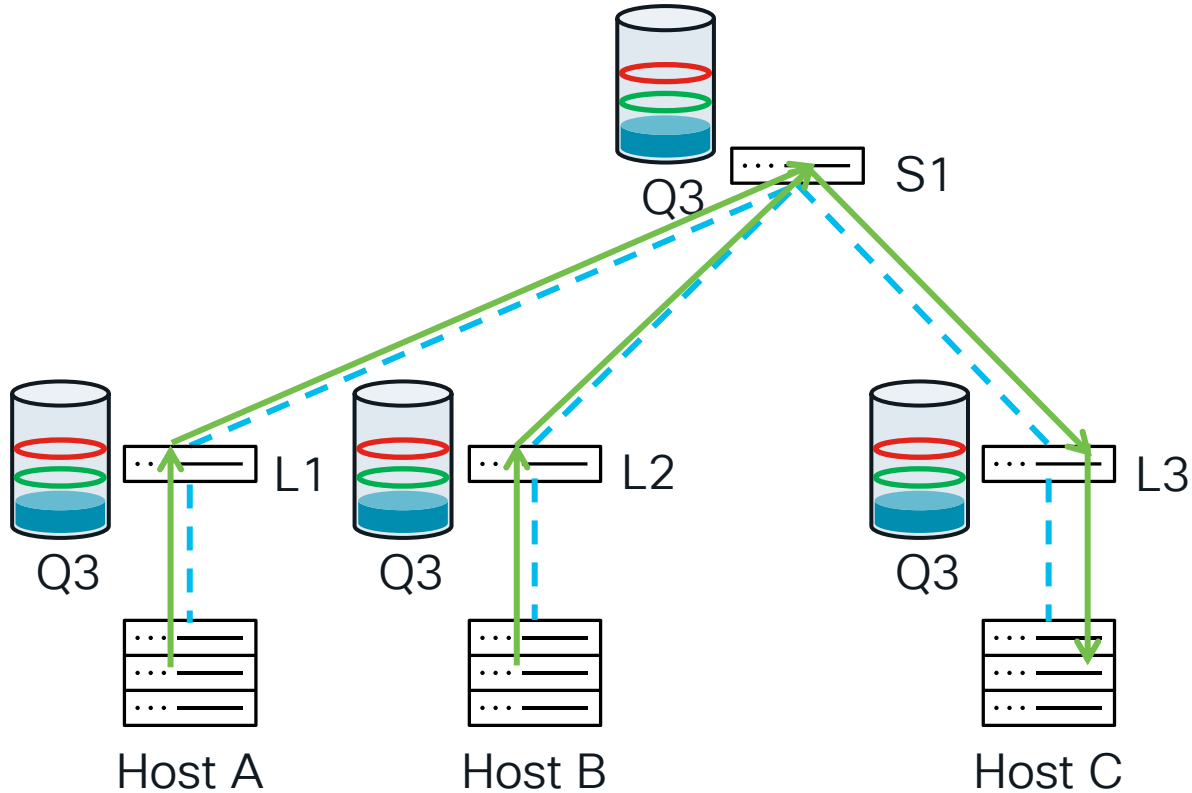


PFC – How pause frames are sent

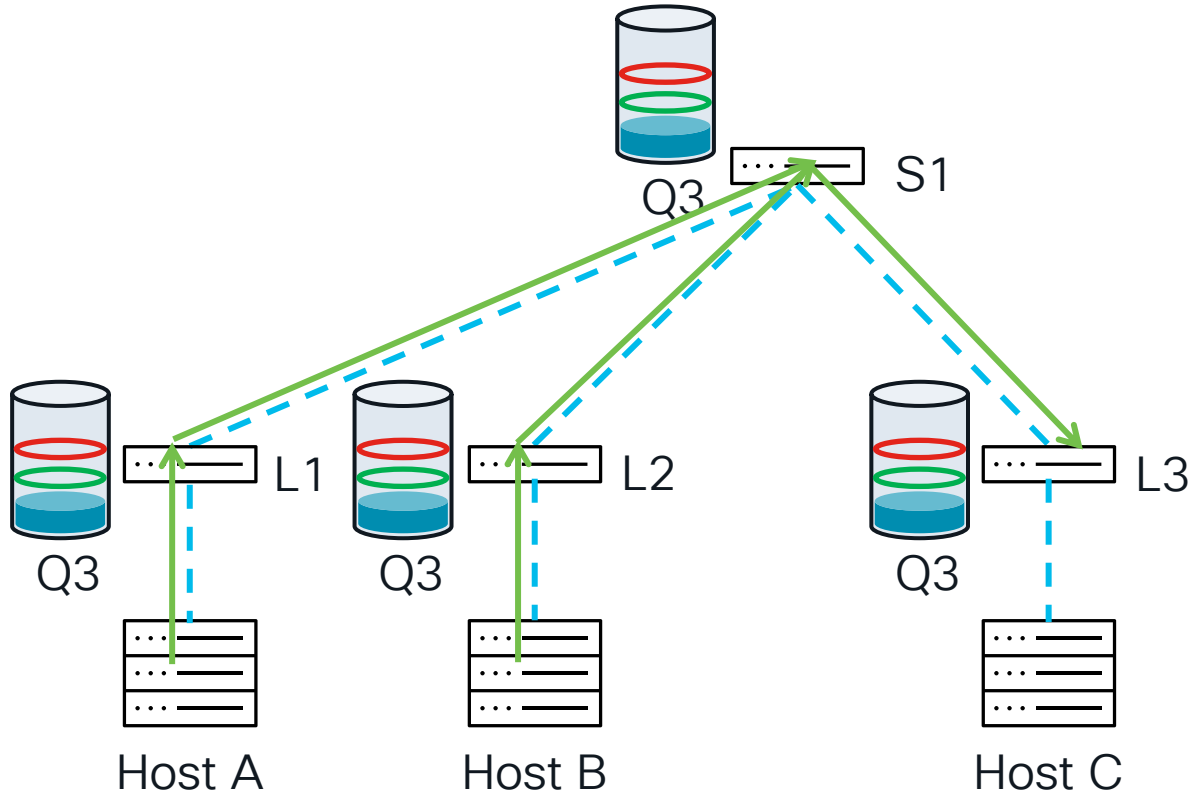
- PFC sets thresholds in no-drop queue
- Headroom is present to accommodate “in flight” packets
- Under congestion, traffic is buffered in non-drop queue
- PFC frames are sent toward sender after queue utilization exceeds *xoff* threshold
- While draining the queue, and utilization is below *xon* threshold system will stop sending PFC frames



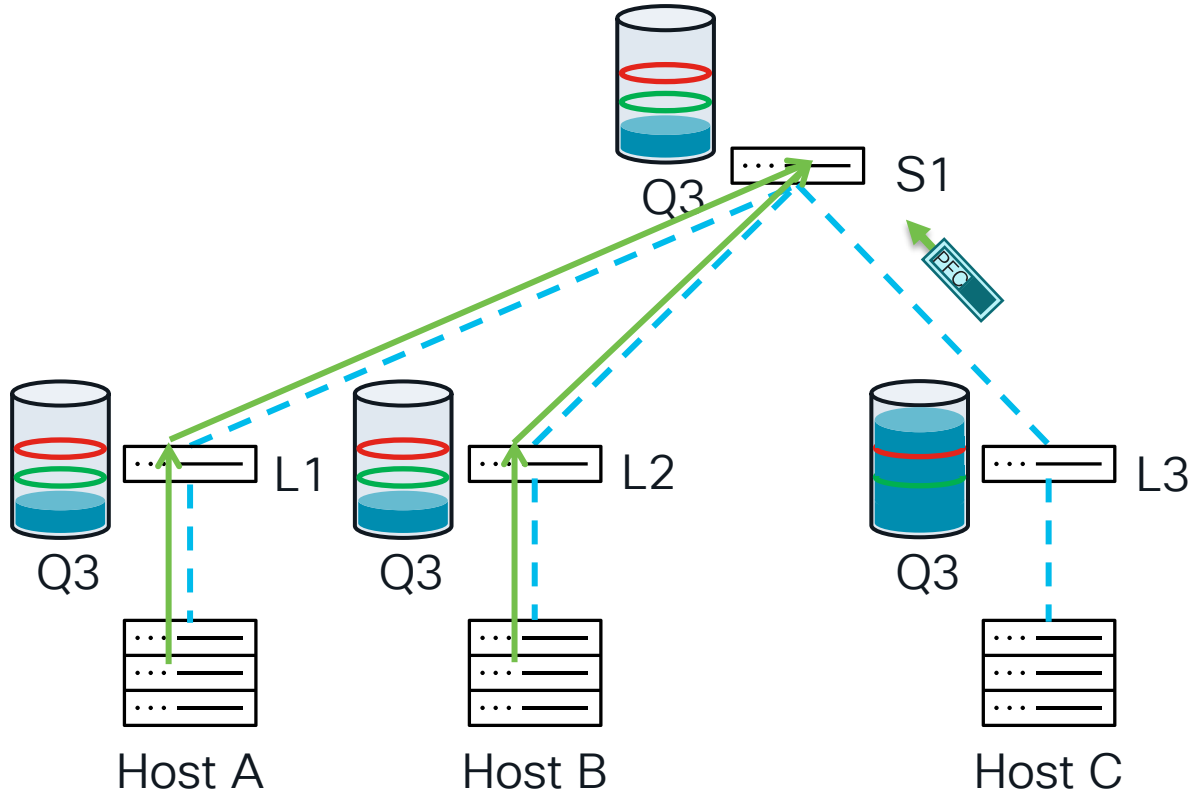
PFC Hop by hop



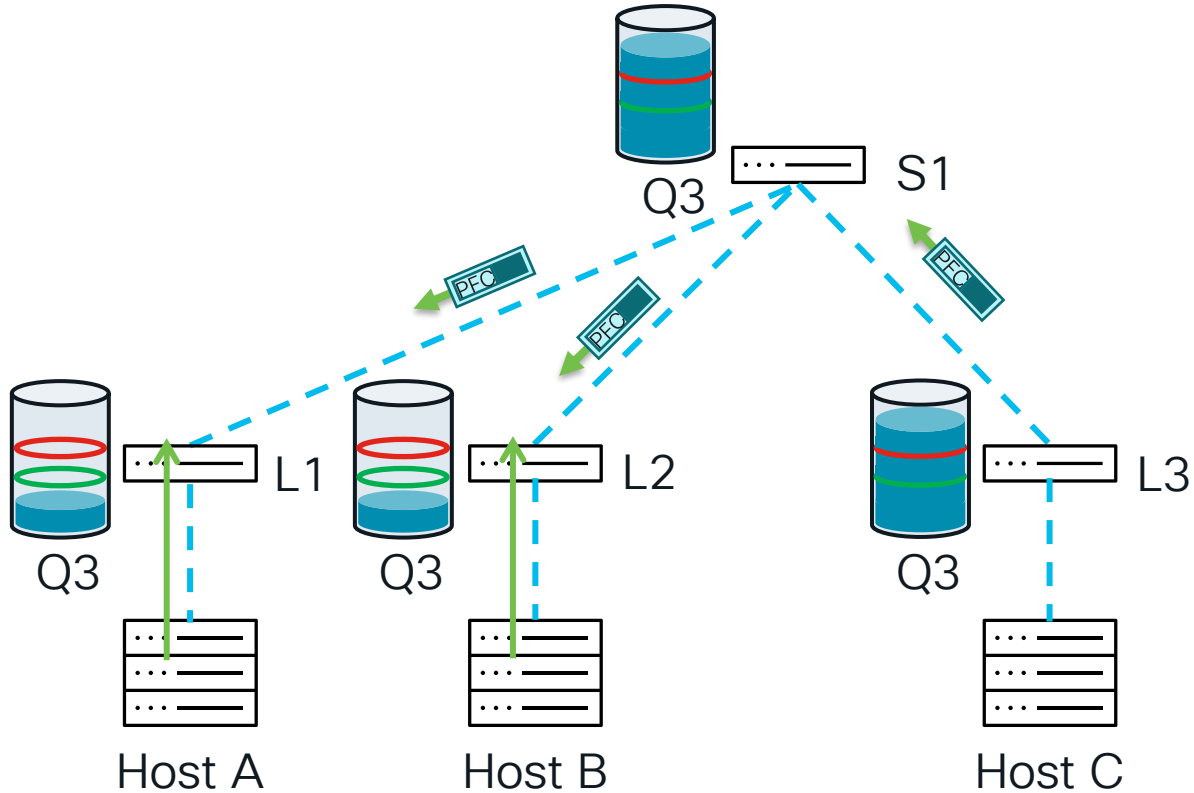
PFC Hop by hop



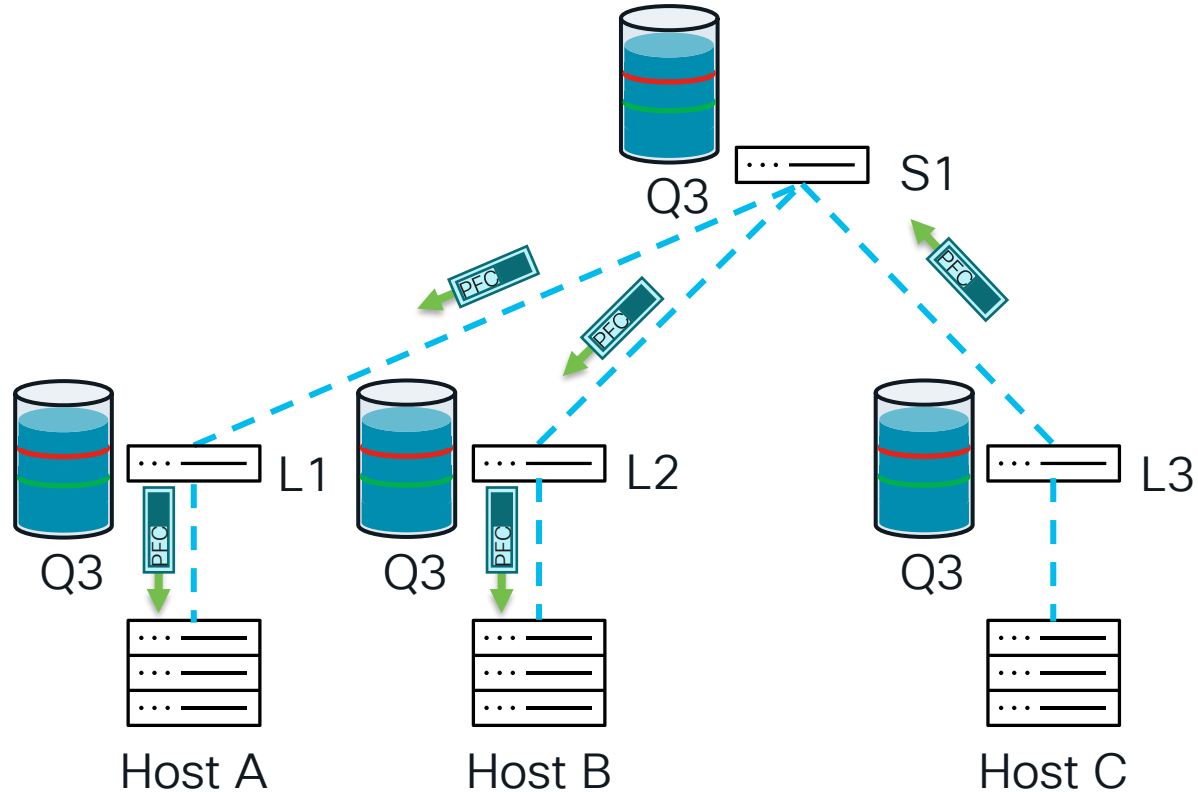
PFC Hop by hop



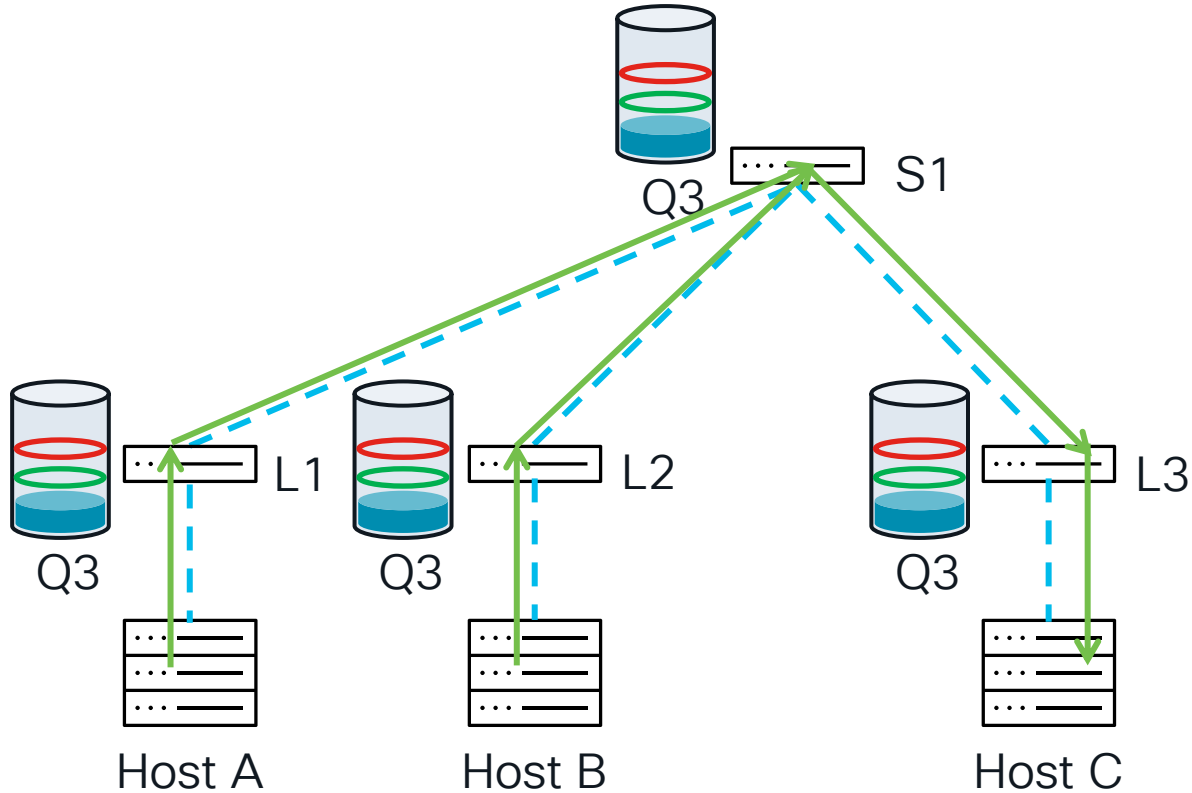
PFC Hop by hop



PFC Hop by hop

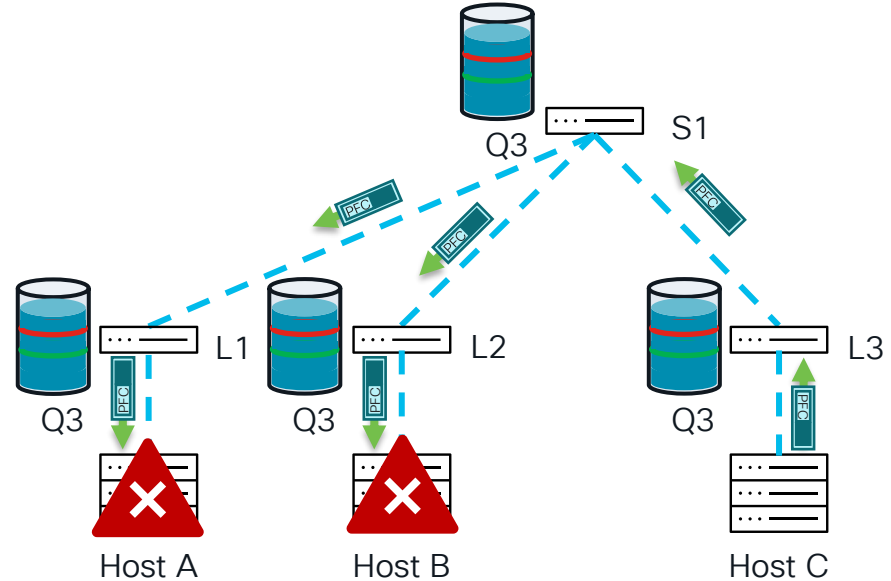


PFC Hop by hop



NIC PFC Storm

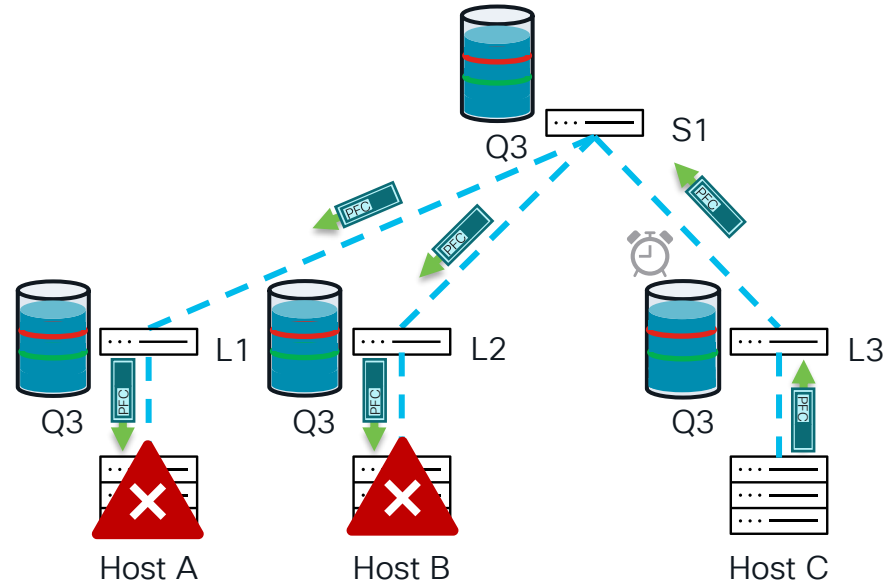
- In occasion of NIC malfunction, PFC storm can be triggered to send continues PFC frames in the network
- Network will propagate those frame to all senders
- PFC storm will stop traffic coming from sender
- PFC watchdog can drain the queue



https://www.microsoft.com/en-us/research/wp-content/uploads/2016/11/rdma_sigcomm2016.pdf

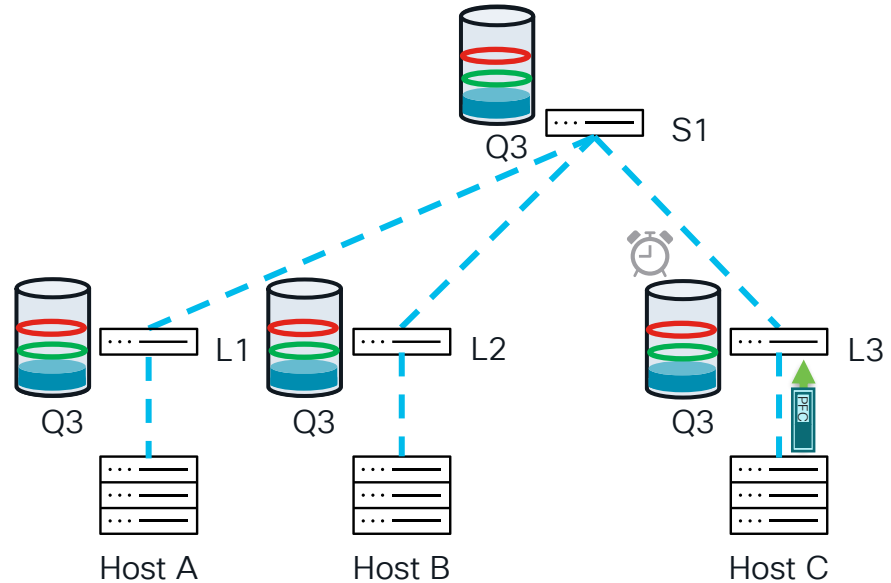
PFC Watchdog

- PFC watchdog sets a timeout, if a packet exceeds time out, all packets from a queue will be cleared
- The watchdog prevents PFC frames propagating to sender and blocking it
- PFC Watchdog is supported on Cisco Nexus 9000 switches



PFC Watchdog

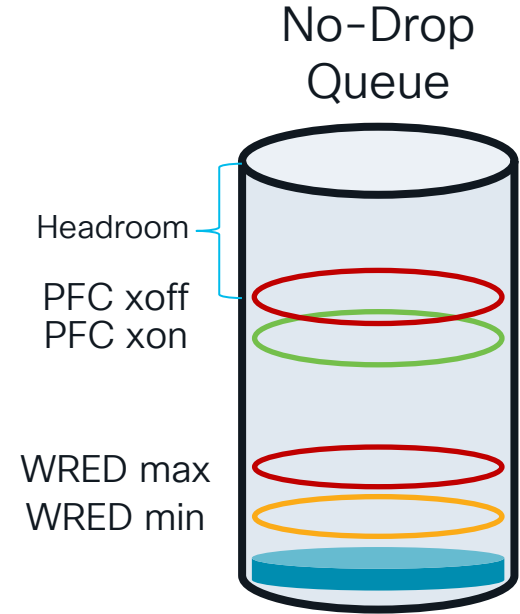
- PFC watchdog sets a timeout, if a packet exceeds time out, all packets from a queue will be cleared
- The watchdog prevents PFC frames propagating to sender and blocking it
- PFC Watchdog is supported on Cisco Nexus 9000 switches



RoCEv2: PFC and ECN together

How does it work?

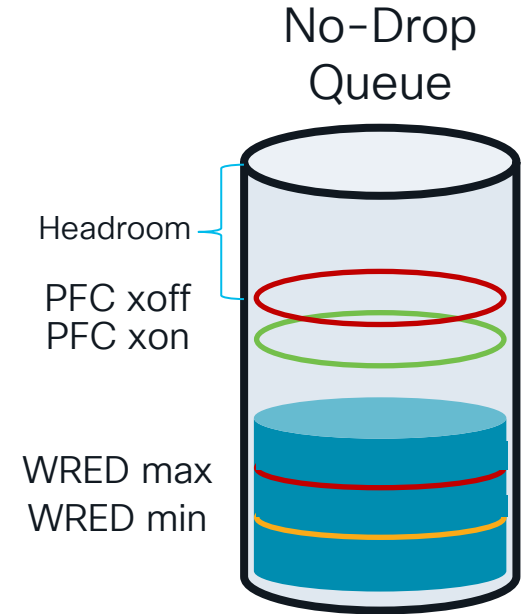
- WRED threshold are set low in no-drop queue
 - Signalize early for congestion, give enough time for end points to react
- PFC threshold are set higher than ECN
 - In case oversubscription buffers can be filled quickly without giving time to ECN to react
 - PFC will react and mitigate congestion



RoCEv2: PFC and ECN together

How does it work?

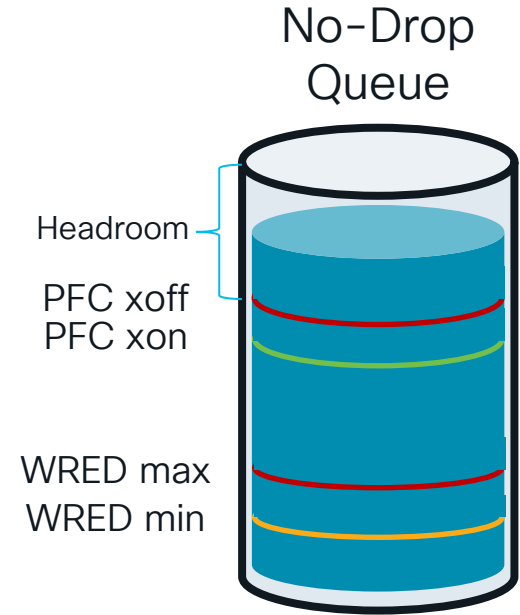
- WRED threshold are set low in no-drop queue
 - Signalize early for congestion, give enough time for end points to react
- PFC threshold are set higher than ECN
 - In case oversubscription buffers can be filled quickly without giving time to ECN to react
 - PFC will react and mitigate congestion



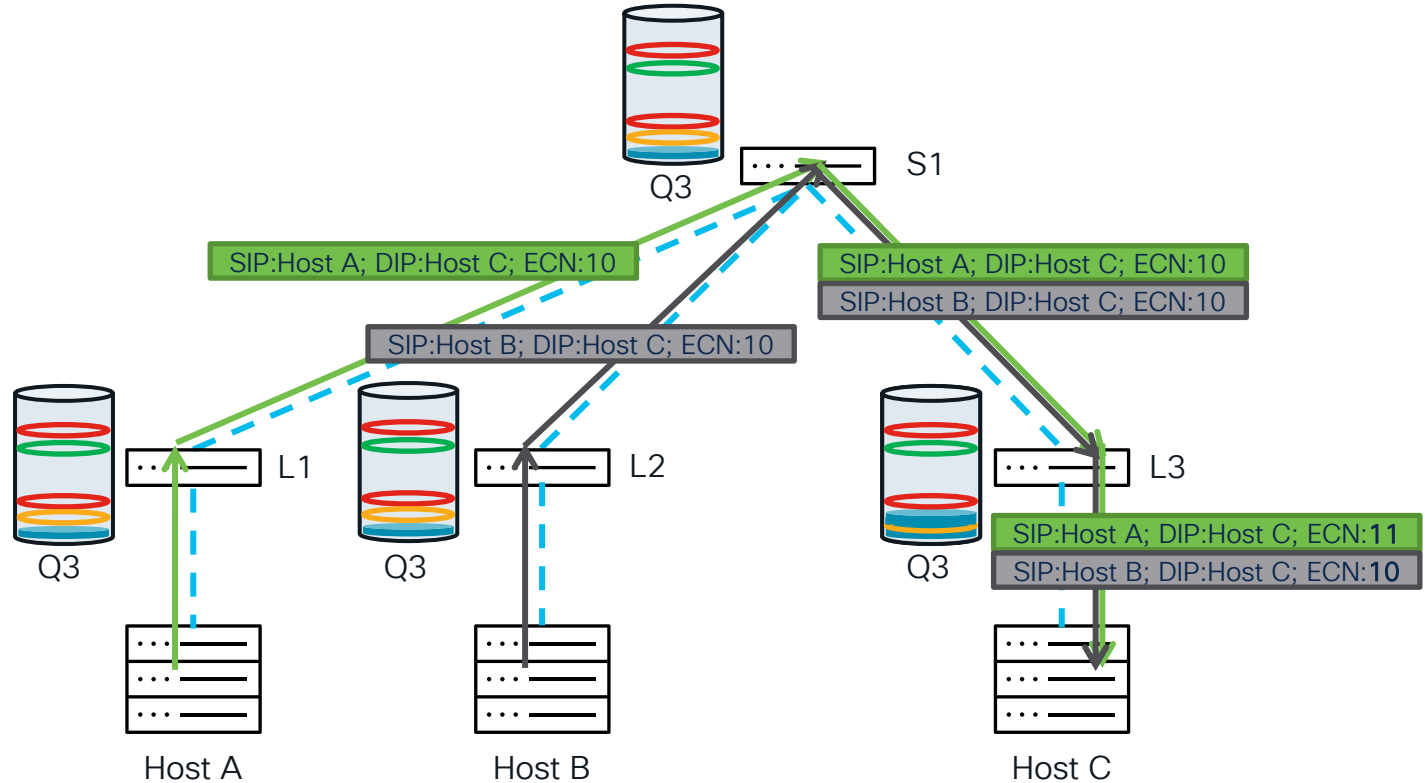
RoCEv2: PFC and ECN together

How does it work?

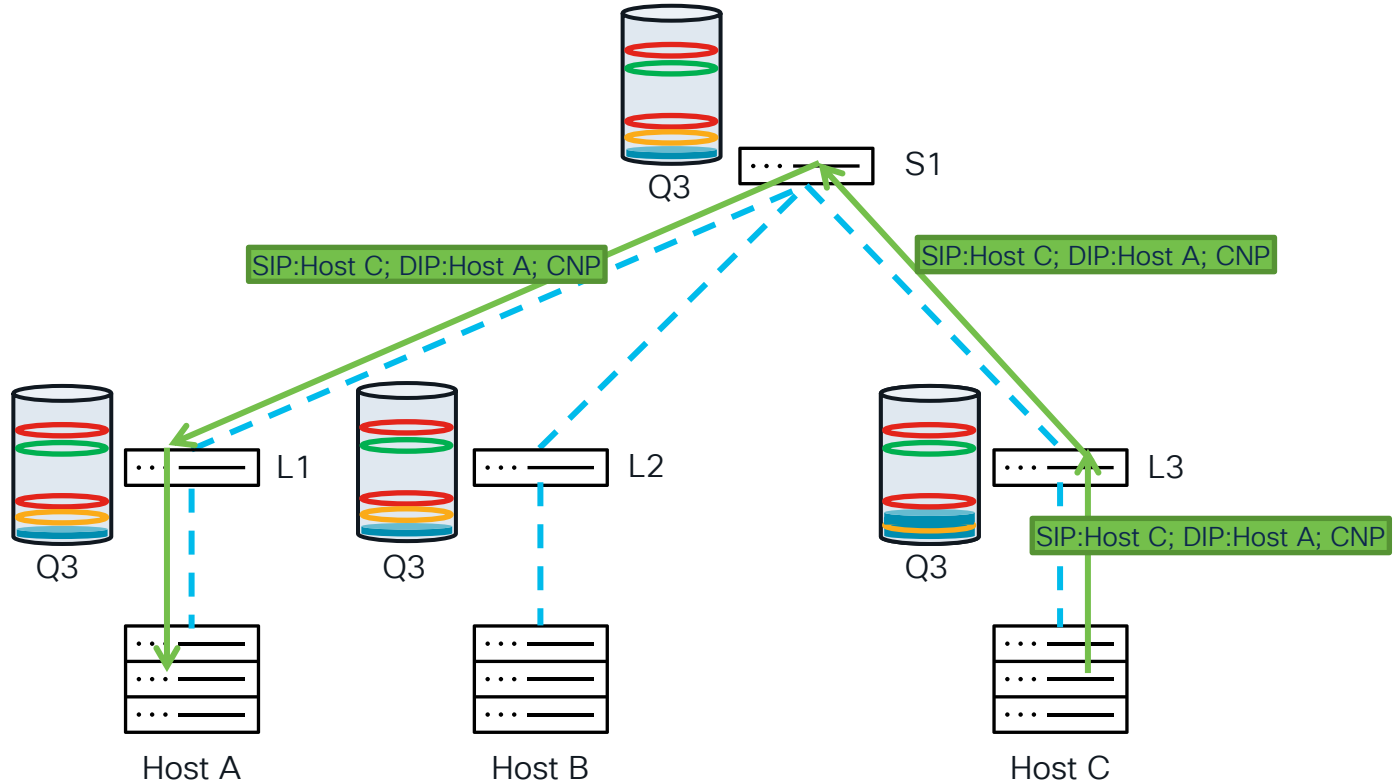
- WRED threshold are set low in no-drop queue
 - Signalize early for congestion, give enough time for end points to react
- PFC threshold are set higher than ECN
 - In case oversubscription buffers can be filled quickly without giving time to ECN to react
 - PFC will react and mitigate congestion



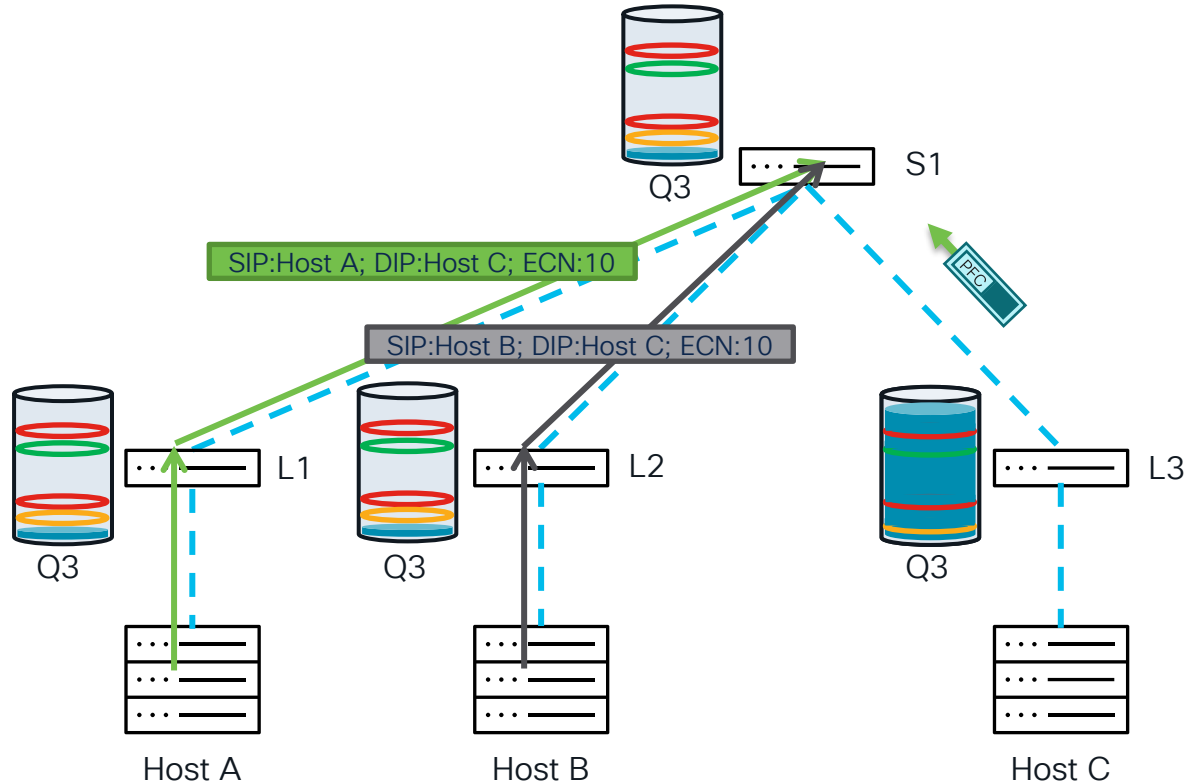
ECN and PFC make DCQCN



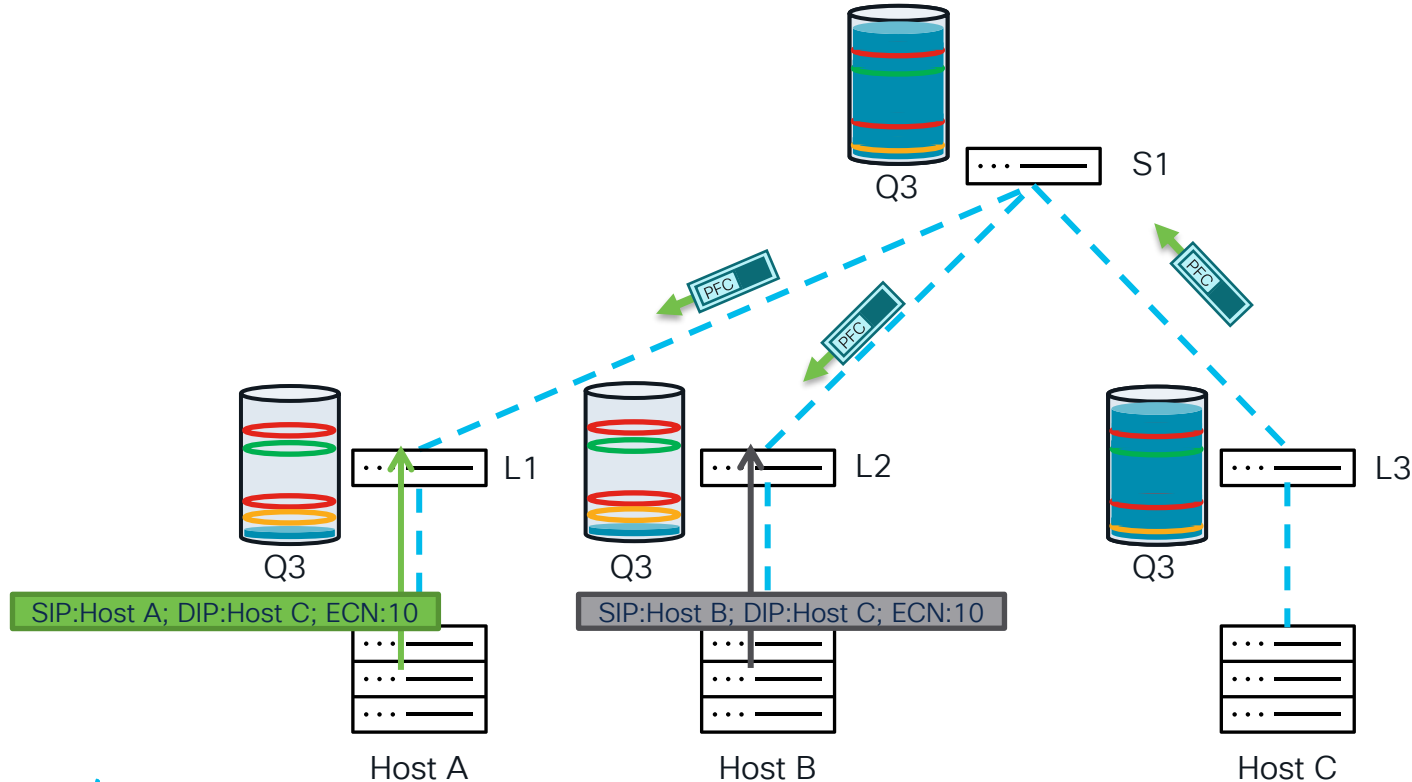
ECN and PFC make DCQCN



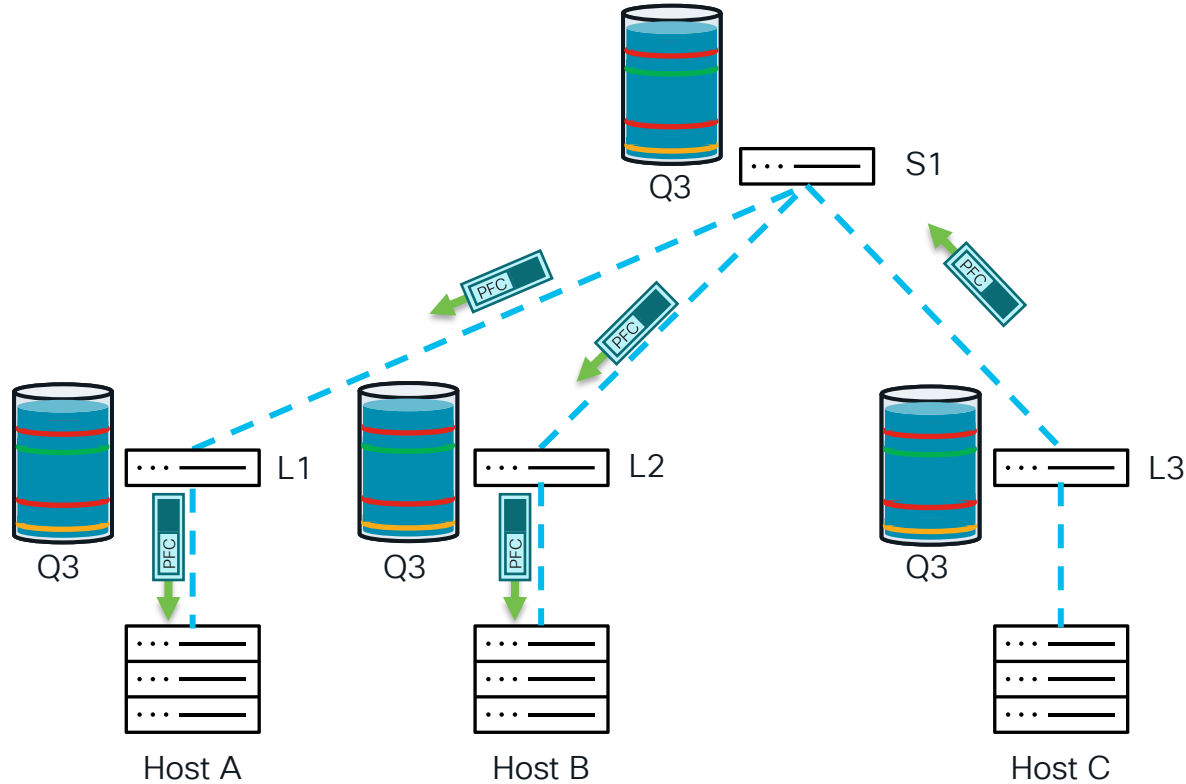
ECN and PFC make DCQCN



ECN and PFC make DCQCN



ECN and PFC make DCQCN



Quality of Service

- Required separate queue for RoCEv2 traffic
 - Distinguished from other traffic in the port
 - Provide dedicated scheduling resources, to reduce latency
 - No contention for buffer resources with other traffic
 - RoCEv2 is not be part of strict priority queue, high volume of it might affect control plane
- RoCEv2 traffic requires ECN, PFC on the queue, while other traffic does not
- CNP traffic is part of strict priority queue, to deliver congestion signaling in time



Quality of Service- Configuration

```
class-map type qos match-all class-roce  
  match dscp 24
```

Classification
for RoCE

```
class-map type qos match-all class-cnp  
  match dscp 48
```

Classification
for CNP

```
policy-map type qos QOS_classification_policy  
  class class-roce  
    set qos-group 3  
  class class-cnp  
    set qos-group 7  
  class class-default  
    set qos-group 0
```

Quality of Service- Configuration

```
policy-map type queuing custom-8q-out-policy
  class type queuing c-out-8q-q7
    priority level 1
```

```
<snip>
```

```
  class type queuing c-out-8q-q3
    bandwidth remaining percent 99
```

```
    random-detect minimum-threshold 150 kbytes maximum-threshold 3000 kbytes drop-probability 7  
weight 0 ecn
```

```
<snip>
```

```
  class type queuing c-out-8q-q-default
    bandwidth remaining percent 1
```

```
policy-map type network-qos custom-8q-nq-policy
```

```
<snip>
```

```
  class type network-qos c-8q-nq3
    mtu 9216
```

```
    pause pfc-cos 3
```

```
<snip>
```

WRED min

WRED max

WRED random
marked packets

Enable PFC

Quality of Service- Configuration

```
system qos
```

```
service-policy type network-qos custom-8q-nq-policy
```

```
service-policy type queuing output custom-8q-out-policy
```

Enable WRED ECN

```
interface Ethernet1/1
```

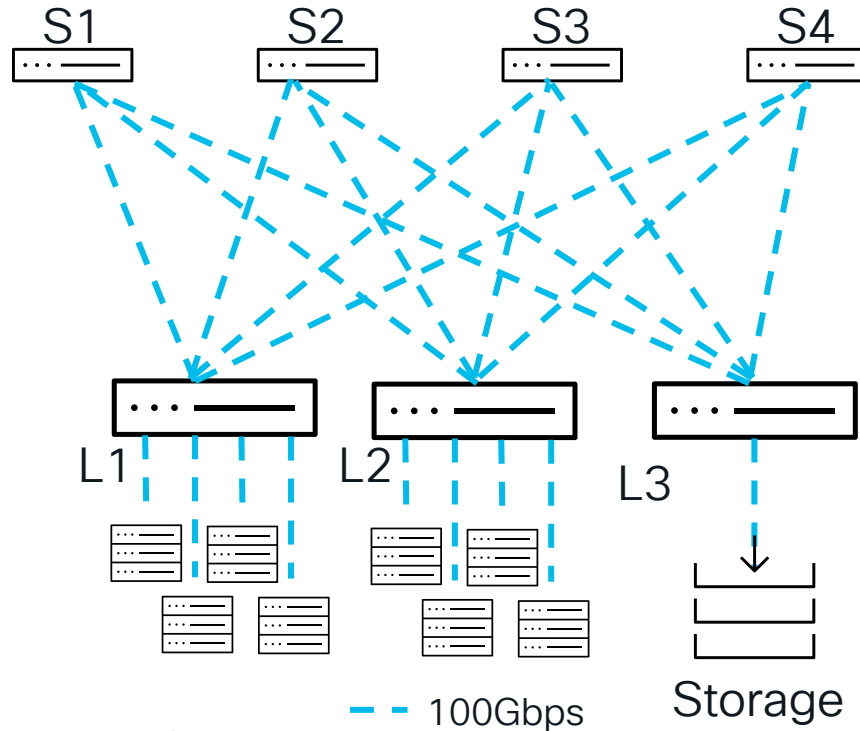
```
service-policy type qos input QOS_classification_policy
```

```
priority-flow-control mode on
```

```
priority-flow-control watch-dog-interval on
```

Enable PFC
per interface

Non-blocking Network



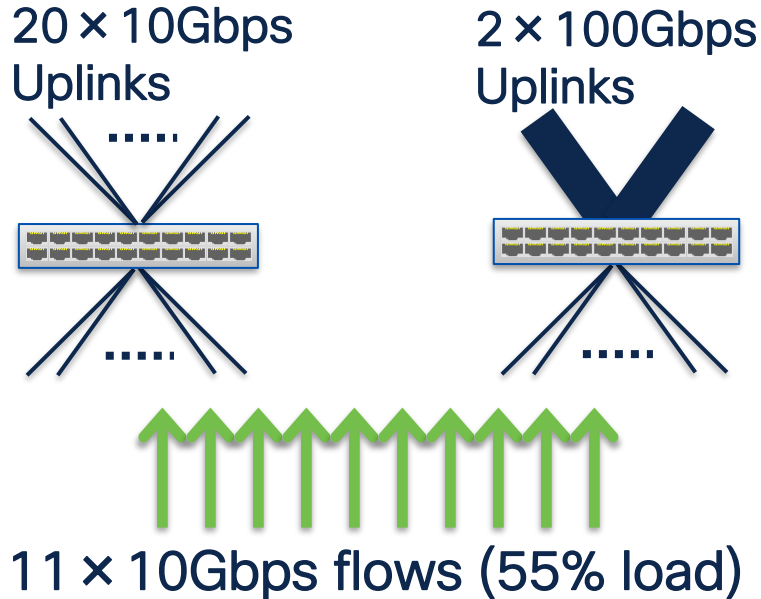
- Non-Blocking Network, allow host to talk to other hosts at full bandwidth
- Leaf: Same bandwidth to the host as to the spine
- Reduces need for congestion management to increase performance

Traffic Load-balancing



Hashing – Where traffic goes?

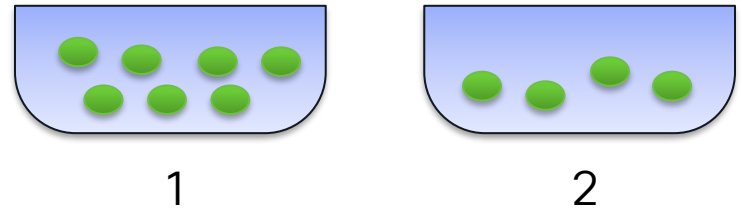
- Avoid oversubscribing link, allow multiple paths to ECMP



Prob of 100% throughput = 3.27%



Prob of 100% throughput = 99.95%



Default ECMP algorithm

- By default, ECMP looks at source and destination IP and Layer 4 ports
- Hosts in AI fabric may belong to uniform subnets
- Layer 4 ports, destination port is 4791 for RoCEv2
- Entropy comes from Layer 4 Source Port

```
N9K-switch# show ip load-sharing
IPv4/IPv6 ECMP load sharing:
Universal-id (Random Seed): 2467474893
Load-share mode : address source-destination port source-destination
Rotate: 32
```

User Defined Field (UDF) ECMP algorithm

- UDF ECMP looks at source and destination IP and User Defined Field in a packet
- User can choose what field to look at to enhance entropy
- Every RoCE conversation is identified by Destination Queue Pair, in IB header
- Destination Queue Pair is 3-byte field

```
N9K-switch# show ip load-sharing
IPv4/IPv6 ECMP load sharing:
Universal-id (Random Seed): 908907021
Load-share mode : address source-destination udf offset 33 length 24
Rotate: 32
```

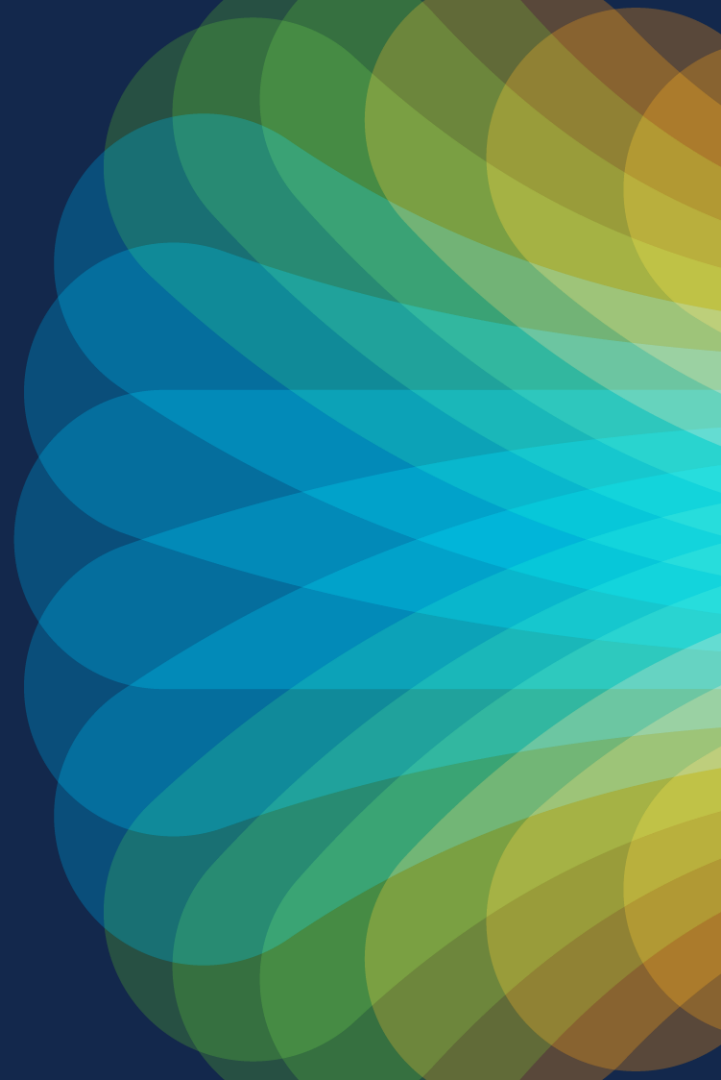

User Defined Field (UDF) – Destination Q Pair

- UDF offset of Nexus 9000 switches starts from first byte of IP header
- Destination Q Pair is 33 bytes from beginning of IP field (IP (20B) + UDP (8B) + IB (5B)) or 6th byte in InfiniBand header

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.16.103.11	172.16.101.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000a19
2	0.000020260	172.16.103.11	172.16.101.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000a19
5	0.116549167	172.16.103.11	172.16.104.11	RRoCE	78	RC RDMA Read Request QP=0x000a00
6	0.116561041	172.16.103.11	172.16.104.11	RRoCE	78	RC RDMA Read Request QP=0x000a00
7	0.122176963	172.16.103.11	172.16.112.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000903
8	0.122185013	172.16.103.11	172.16.112.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000903
9	0.236039751	172.16.103.11	172.16.101.11	RRoCE	78	RC RDMA Read Request QP=0x000a19
10	0.236050373	172.16.103.11	172.16.101.11	RRoCE	78	RC RDMA Read Request QP=0x000a19

> Frame 7: 1086 bytes on wire (8688 bits), 1086 bytes captured (8688 bits) on interface 0/23/0	0030	ff ff 00 00 09 03 00 14	72 81 f5 6e
> Ethernet II, Src: MellanoxTech_c4:7c:ab (b8:ce:f6:c4:7c:ab), Dst: Cisco_23:d3:95	0040	64 3d 50 c3 60 3d db 77	94 3d 05 6e
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 103	0050	1b 3d fd f2 da 3c c7 a1	87 3d 78 1e
> Internet Protocol Version 4, Src: 172.16.103.11, Dst: 172.16.112.11	0060	a2 3c 4f 94 47 3d 00 24	f4 3a bc 5d
> User Datagram Protocol, Src Port: 49152, Dst Port: 4791	0070	8c 3d 89 88 16 3d 9d c4	a0 3c b5 6e
> InfiniBand	0080	57 3d a3 14 9d 3d 39 76	85 3d a5 2e
Base Transport Header	0090	a2 3d 43 f0 96 3d a9 a5	96 3d f2 6e
Opcode: Reliable Connection (RC) – RDMA READ response Middle (14)	00a0	0c 3b 45 49 11 3d 61 df	82 3d 6b 7e
0... = Solicited Event: False	00b0	1a 3d b5 8e 38 3d 25 33	ed 3c bd 1e
1.. = MigReq: True	00c0	57 3d 6d 44 55 3d 80 12	da 3a 90 3e
... .. = Pad Count: 0	00d0	ad 3c 5d 92 34 3d 2b 2a	22 3c fd 3e
.... 0000 = Header Version: 0	00e0	31 3d 60 ca 70 3d 1f 68	6d 3d 6d 5e
Partition Key: 65535	00f0	10 3d 65 e8 aa 3c e4 4f	75 3d 4b 6e
Reserved: 00	0100	81 3d 5c 15 a5 3d b8 1e	82 3d 22 3e
Destination Queue Pair: 0x000903	0110	0f 3d 88 9e 1a 3d 65 97	7f 3c 29 c0
0... = Acknowledge Request: False	0120	8d 3d a7 8c 41 3d 58 48	ff 3c 07 5e
... .. = Reserved (7 bits): 0	0130	17 3d 5b 72 64 3c d5 b7	d7 3a 71 5e
Packet Sequence Number: 1340033	0140	a1 3d d9 f9 45 3d eb 6a	1c 3c 03 3e
Invariant CRC: 0xb5baee4d	0150	06 3d 73 c4 60 3d b1 8d	a3 3d 50 4e
Data (1024 bytes)	0160	7a 3c a0 6e eb 3b a0 16	6d 3c b9 4e
	0170	28 3d cd c9 5e 3d eb f6	28 3c a3 3e
	0180	13 3d 4b 93 86 3d 85 28	c4 3c 6b 6e
	0190	53 3a 53 c9 f4 3c 6c fa	51 3d 64 7e
	01a0	91 3d f4 44 69 3d f2 8e	90 3d 0b 1e
	01b0	5f 3d 91 cb 97 3d 43 51	ba 3c e5 6e

Automation and Visibility



Nexus Dashboard Fabric Controller

Create Fabric?—×

N9K Cloud Scale Platform Queuing Policy	
<div>Select an Option</div>	Queuing Policy for all 92xx, -EX, -FX, -FX2, -FX3, -GX series switches in the fabric
N9K R-Series Platform Queuing Policy	
<div>Select an Option</div>	Queuing Policy for all R-Series switches in the fabric
Other N9K Platform Queuing Policy	
<div>Select an Option</div>	Queuing Policy for all other switches in the fabric
Enable AI / ML QoS and Queuing Policies	
<input checked="" type="checkbox"/>	Configures QoS and Queuing Policies specific to N9K Cloud Scale switch fabric for AI / ML network loads
AI / ML QoS & Queuing Policy*	
<div>AI_Fabric_QOS_100G</div>	Queuing Policy based on predominant fabric link speed: 400G / 100G / 25G
<div>AI_Fabric_QOS_400G</div>	
<div>AI_Fabric_QOS_100G</div>	Enable MACsec in the fabric
<div>AI_Fabric_QOS_25G</div>	
	Cisco Type 7 Encrypted Octet String

Nexus Dashboard Fabric Controller

Edit interface(s) ? — ×

Additional CLI for the interface

Enable Interface* ☒ Uncheck to disable the interface

Enable Netflow ☐ Netflow is supported only if it is enabled on fabric

Netflow Monitor Provide the Layer 3 Monitor Name

Netflow Sampler Netflow sampler name, applicable to N7K only

Enable priority flow control ☒ Enable priority flow control

Enable QoS Configuration ☒ Enable to configure a QoS Policy for this interface. If AI/ML Queuing is enabled on the fabric, will use the QOS_CLASSIFICATION policy. Enter a custom policy below to override

Custom QoS Policy Custom QoS Policy must be defined previously

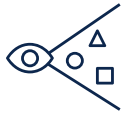
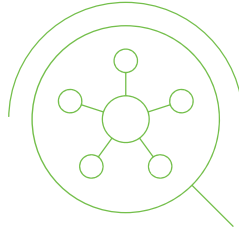
Custom Queuing Policy Queuing Policy must be defined previously

Visibility – Flow Table

- Collects full flow information plus metadata
 - 5-tuple flow info
 - Interface/queue info
 - Flow start/stop time
 - Packet disposition (drop indicators)
 - Burst measurement
- Export data to collector
- Leveraged by Nexus Dashboard Insights



Cisco Nexus Dashboard Insights



With the granular visibility provided by Cisco Nexus Dashboard Insights the network administrator can observe drops

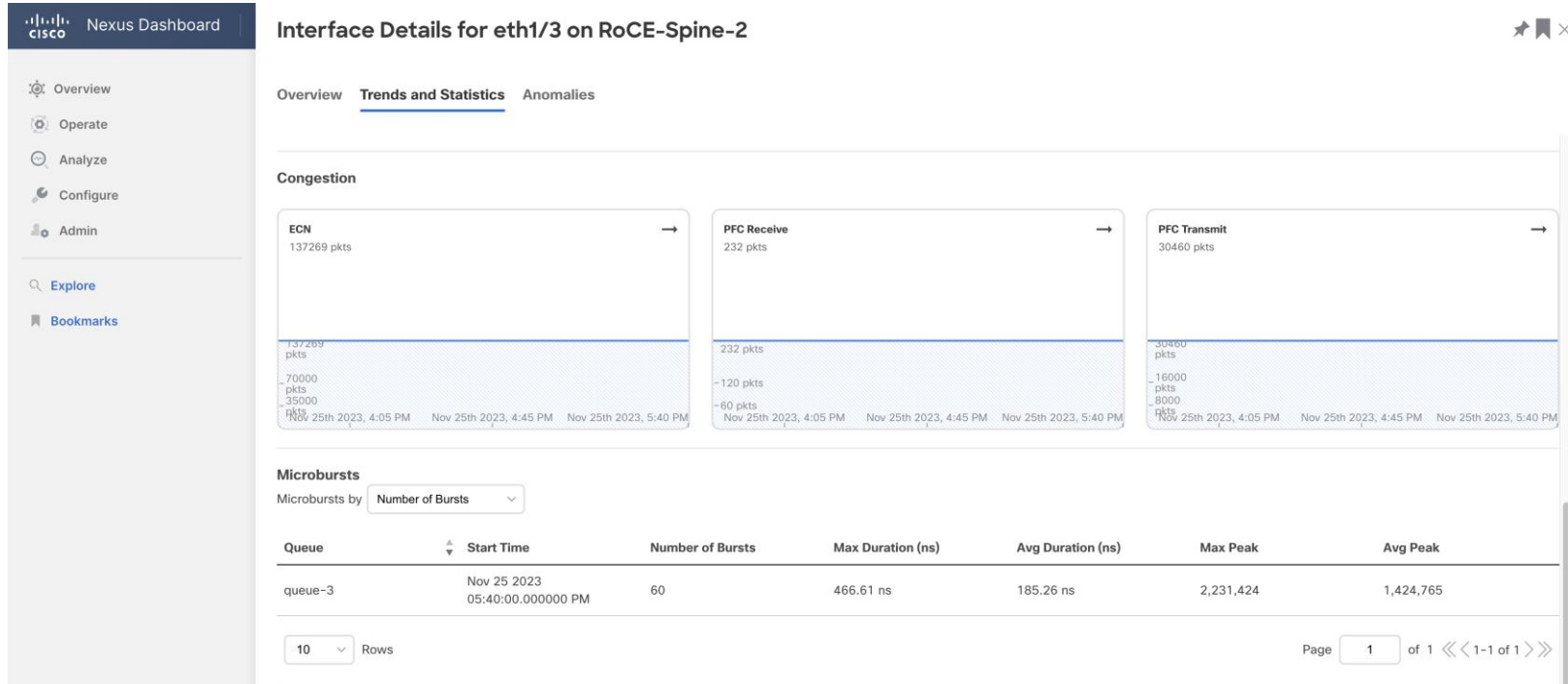


Tune thresholds until congestion hot spots clear and packet drops stop in normal traffic conditions

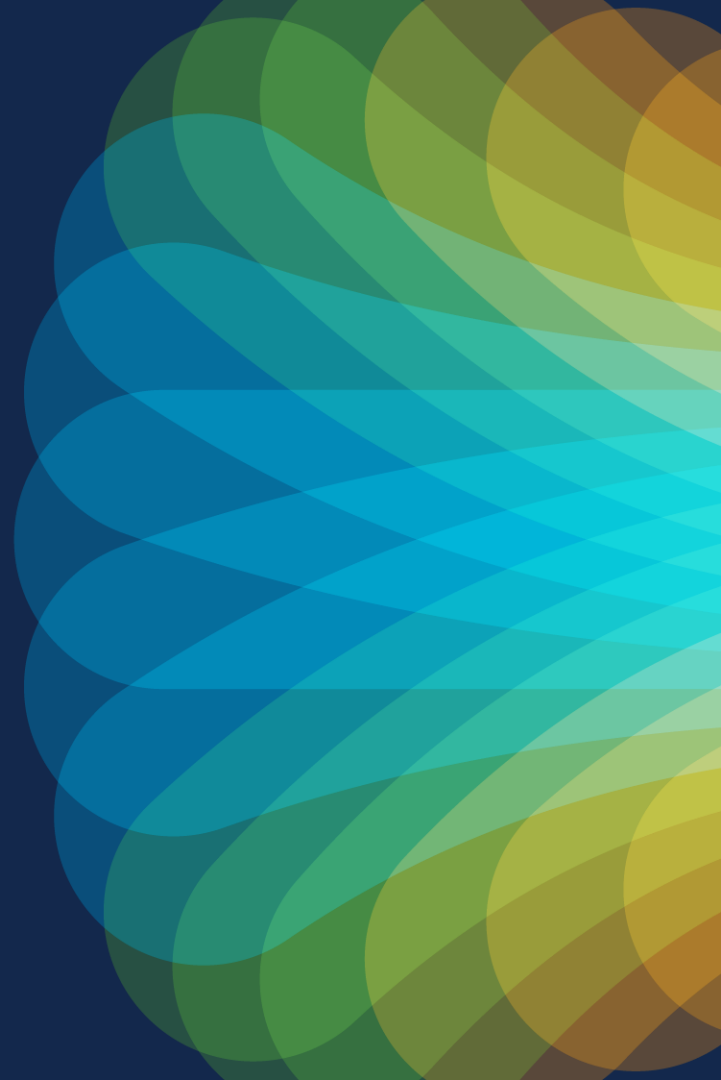


This is the first and most important step to ensure that the AI/ML network will cope with regular traffic congestion occurrences effectively

Nexus Dashboard Insights – Congestion Visibility

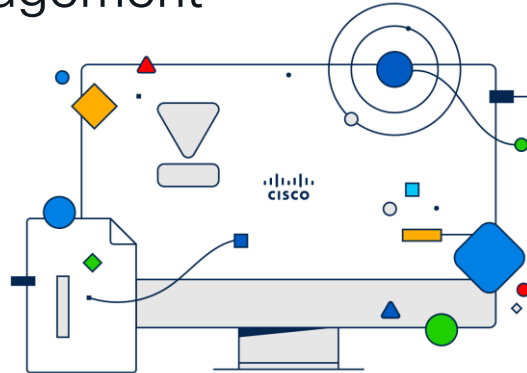


Blueprint of Today



Choosing the right infrastructure

- Build a Clos Fabric / Spine-Leaf
- Fixed Switches
 - Lower Latency, single AISC
 - Power Efficient
- Right Congestion management
- Routed Fabric
 - Use BGP for control plane
- Scalable design
 - Two tier design
 - Three tier design



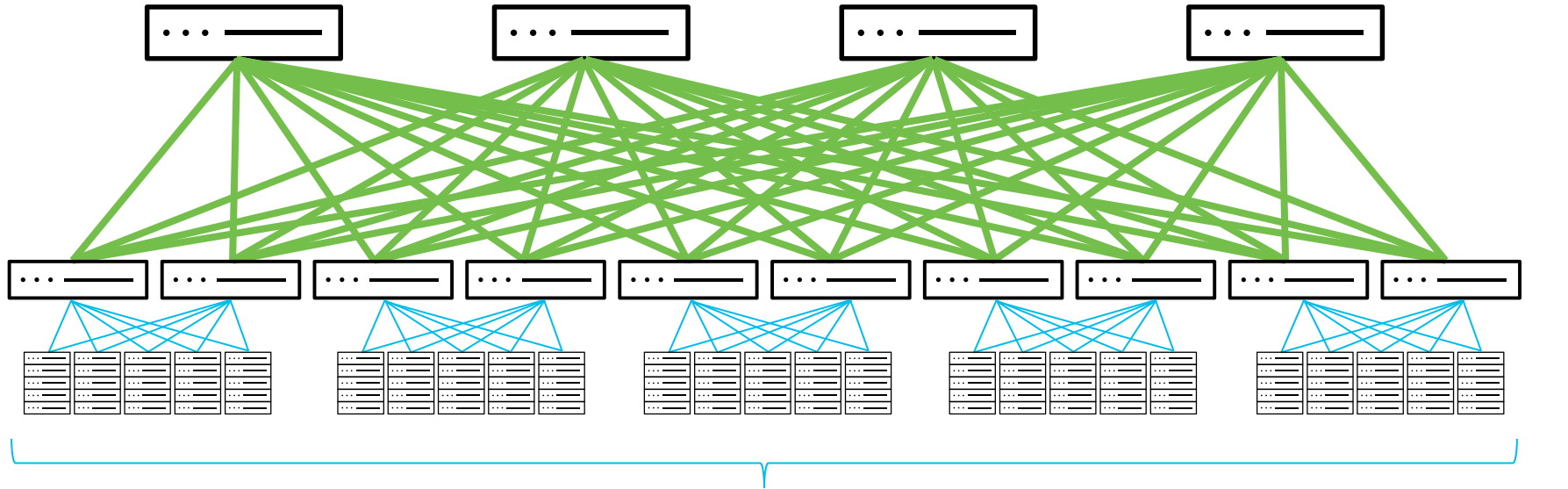
Customer Request #1

- Build Cluster of 260 GPUs, for small training model use case
 - Cluster is built with stand alone server (e.g. Cisco UCS-C240-M7)
- Build non-blocking network, for GPU communication
- Predictable and low latency for efficient training
- Host connectivity at dual 100Gbps
- Fabric at 400Gbps for efficient load-balancing

Customer Request #1 – Proposal

- Standalone server can have up to 2 GPUs
 - As required is 260 GPUs, 130 Stand alone servers are needed
 - Each standalone server is equipped with dual 100G port NIC
- 260 x 100G ports required for host connectivity in leaf layer
 - 26 x 100G host interfaces per leaf switch, for 10 leaf switches
 - For seeped up have 8 x 400G uplinks per switch for Spine connectivity
- Total of 4 spines 20 x 400G ports used per Spine
- Leave room for future expansion

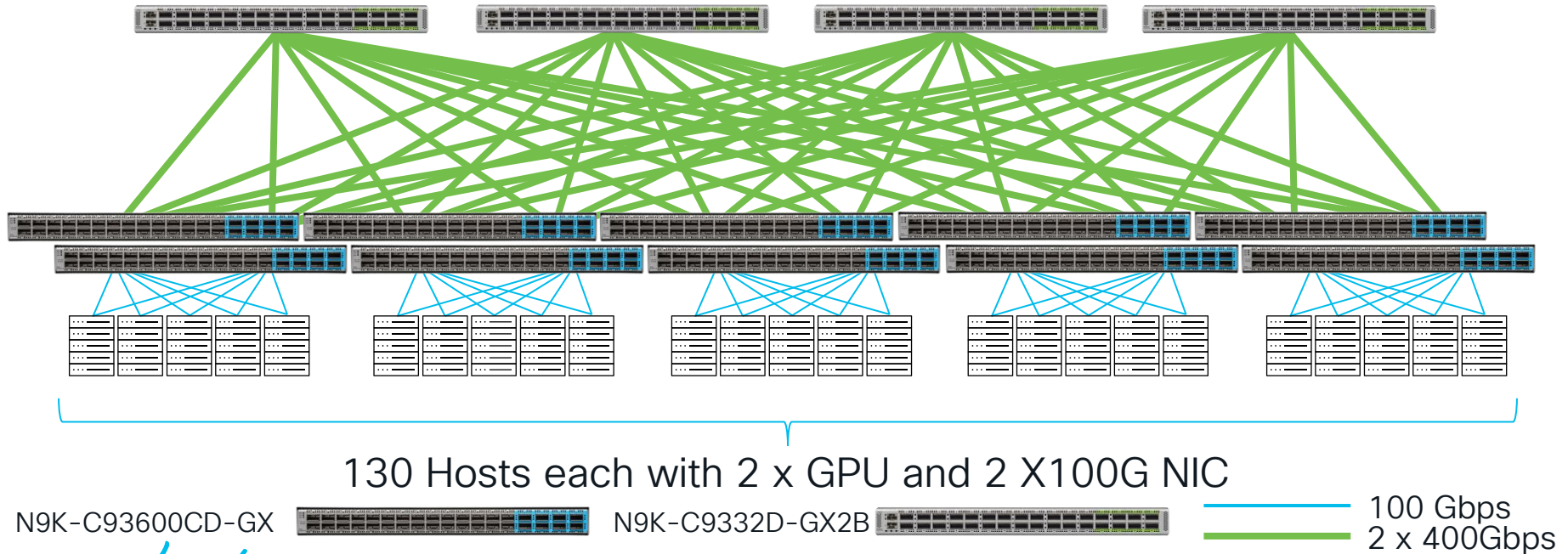
Customer Request #1 - Design



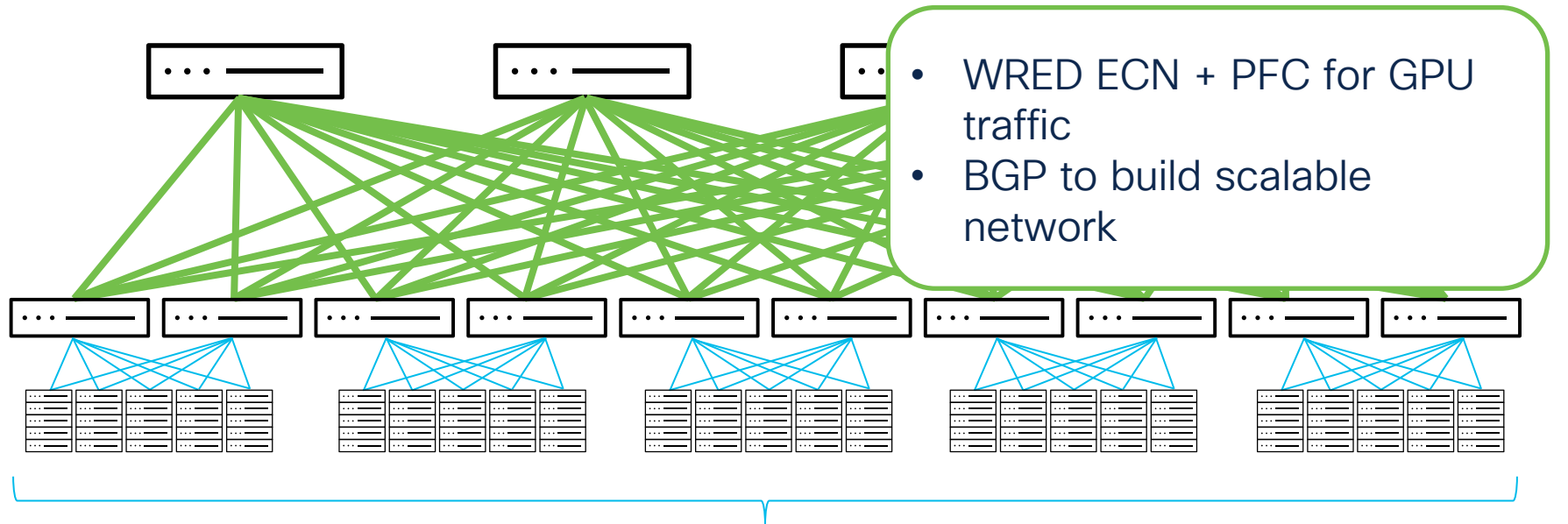
130 Hosts each with 2 x GPU and 2 X100G NIC

— 100 Gbps
— 2 x 400Gbps

Customer Request #1 - Design



Customer Request #1 - Design



130 Hosts each with 2 x GPU and 2 X100G NIC

— 100 Gbps
— 2 x 400Gbps

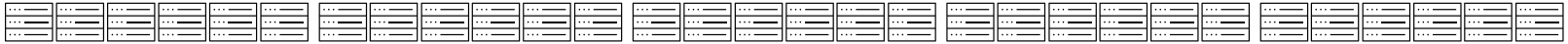
Customer Request #2

- Build Powerful Cluster of 320 GPUs, for large training model use case
 - Cluster is built with powerful accelerators (e.g. NVIDIA DGX, Intel Gaudi)
- Build non-blocking back-end network, for GPU communication
 - Predictable and low latency for efficient training
 - Host connectivity at 8 ports of 4 x 100Gbps (Logical 400Gbps)
 - Fabric at 400Gbps for efficient load-balancing
- Build front-end network, for server-to-server interaction, and storage connectivity
 - 2 x 100Gbps ports for connectivity, non-blocking and future expansion

Customer Request #2 – Proposal for Back-End

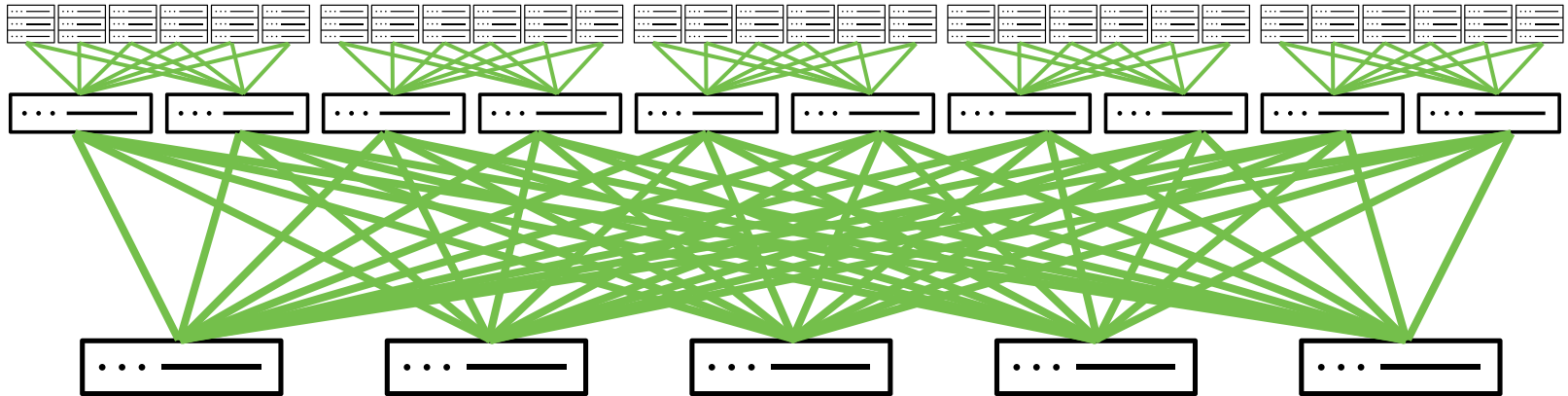
- Powerful AI training accelerator, has 8 GPUs per server
 - As required is 320 GPUs, 40 host are sufficient
 - Each host is equipped with 4 NICs each with dual QSFP-DD ports
- 320 x 400G ports required for host connectivity in leaf layer
 - 32 x 400G host interfaces per leaf switch, for 10 leaf switches
 - Non-blocking network 32 x 400G uplinks per leaf for Spine connectivity
- Total of 5 spines 64 x 400G ports used per Spine

Front-End and Back-End Cluster Network

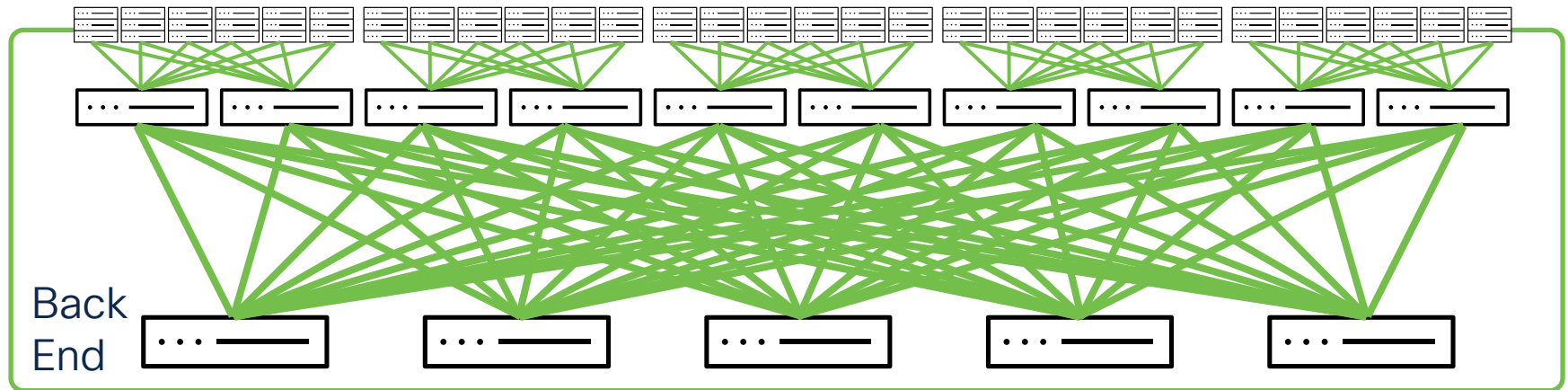


40 Hosts each with 8 x GPU

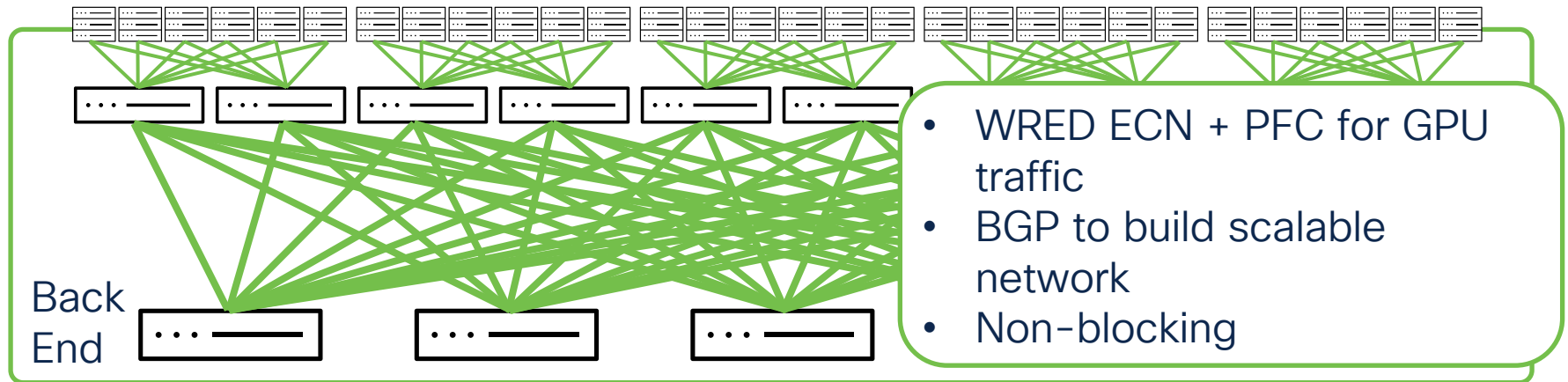
Back-End Cluster Network



Back-End Cluster Network



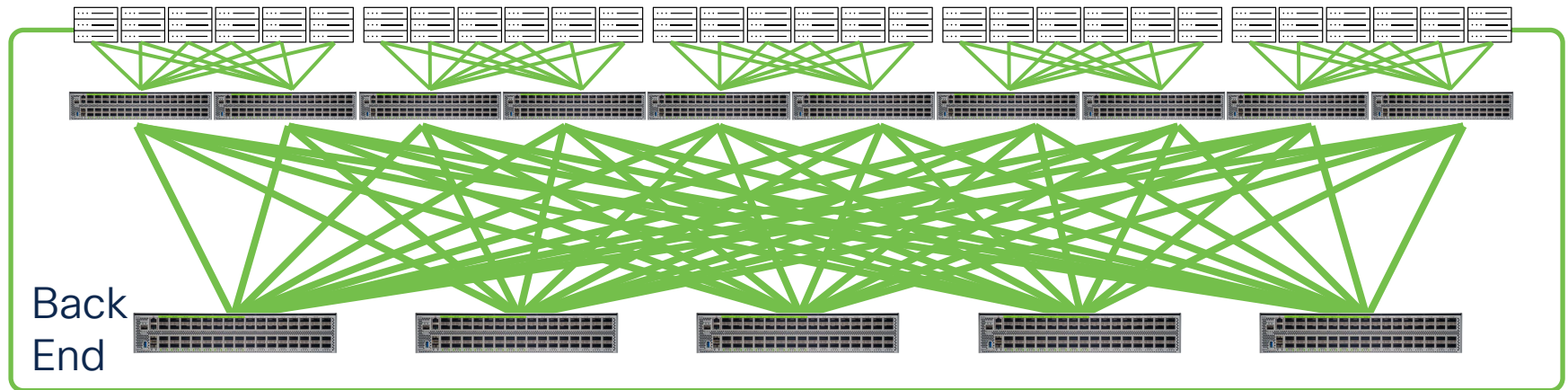
Back-End Cluster Network



Back-End Cluster Network



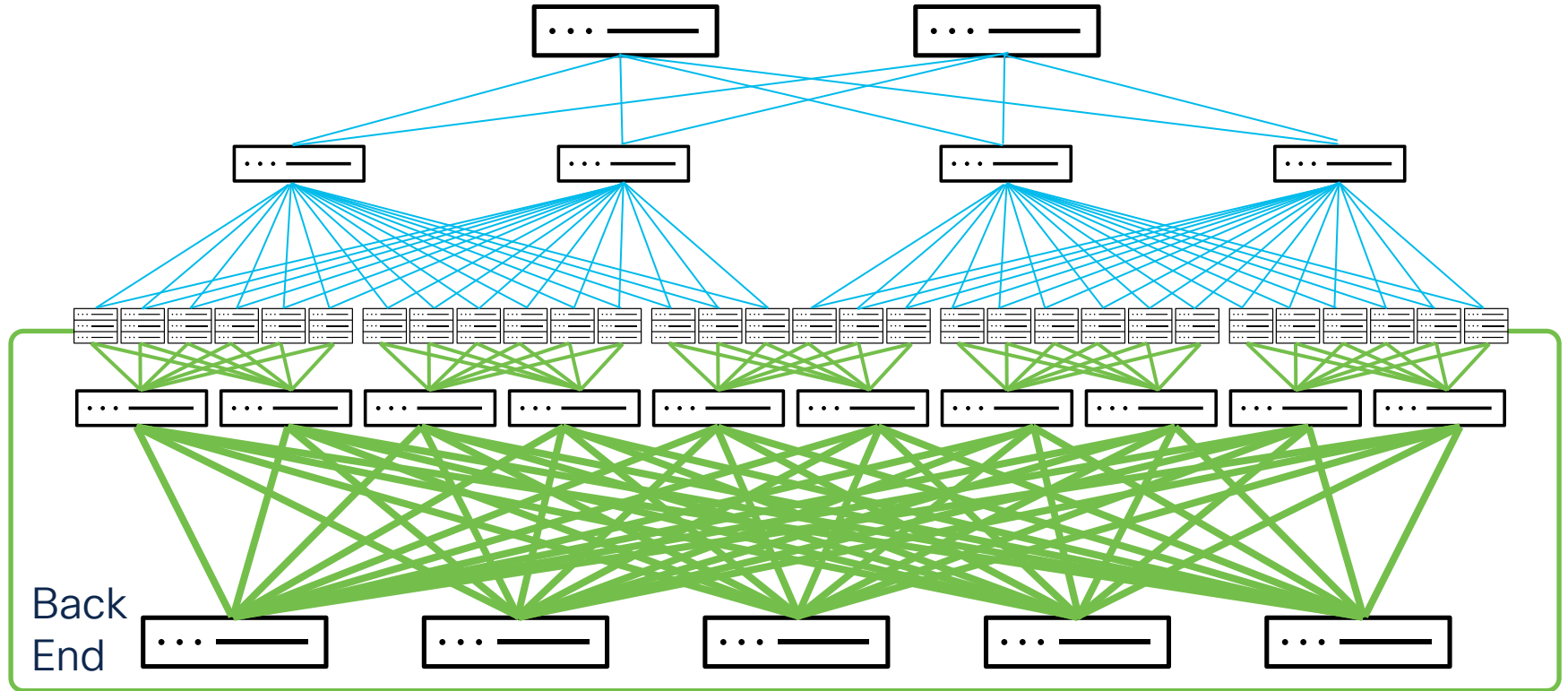
N9K-C9364D-GX2A



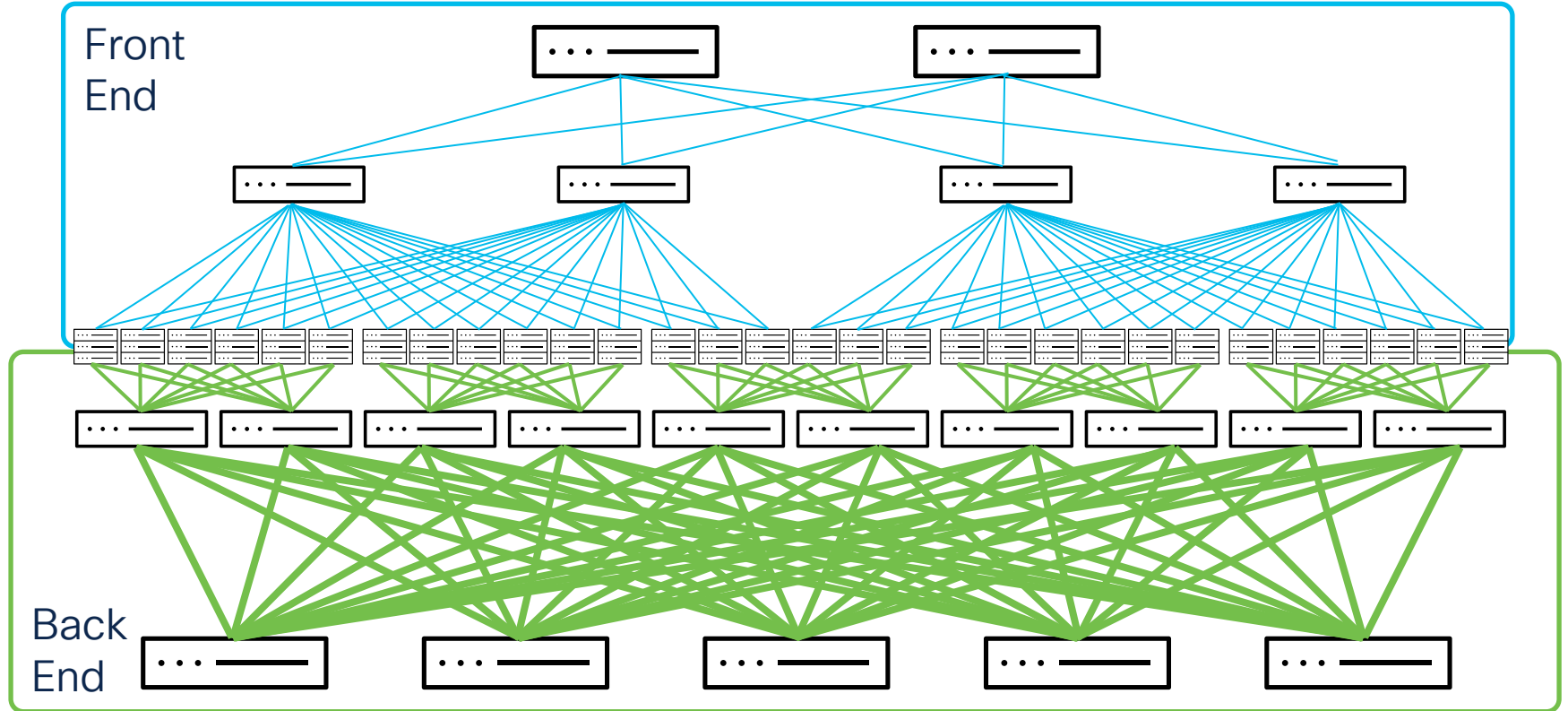
Customer Request #2 – Proposal for Front-End

- Front End network for host communication, and storage
 - Each server has 2 x 100G ports
- 80 x 100G ports required for host connectivity in leaf layer
 - 20 x 100G host interfaces per leaf switch, for 4 leaf switches
 - Non-blocking network 20 x 100G uplinks per leaf for Spine connectivity
- Total of 2 spines 64 x 100G ports used per Spine
- Storage NFS network in Front End
 - 3 Storage array connected to leaf
 - RoCEv2 for storage networks

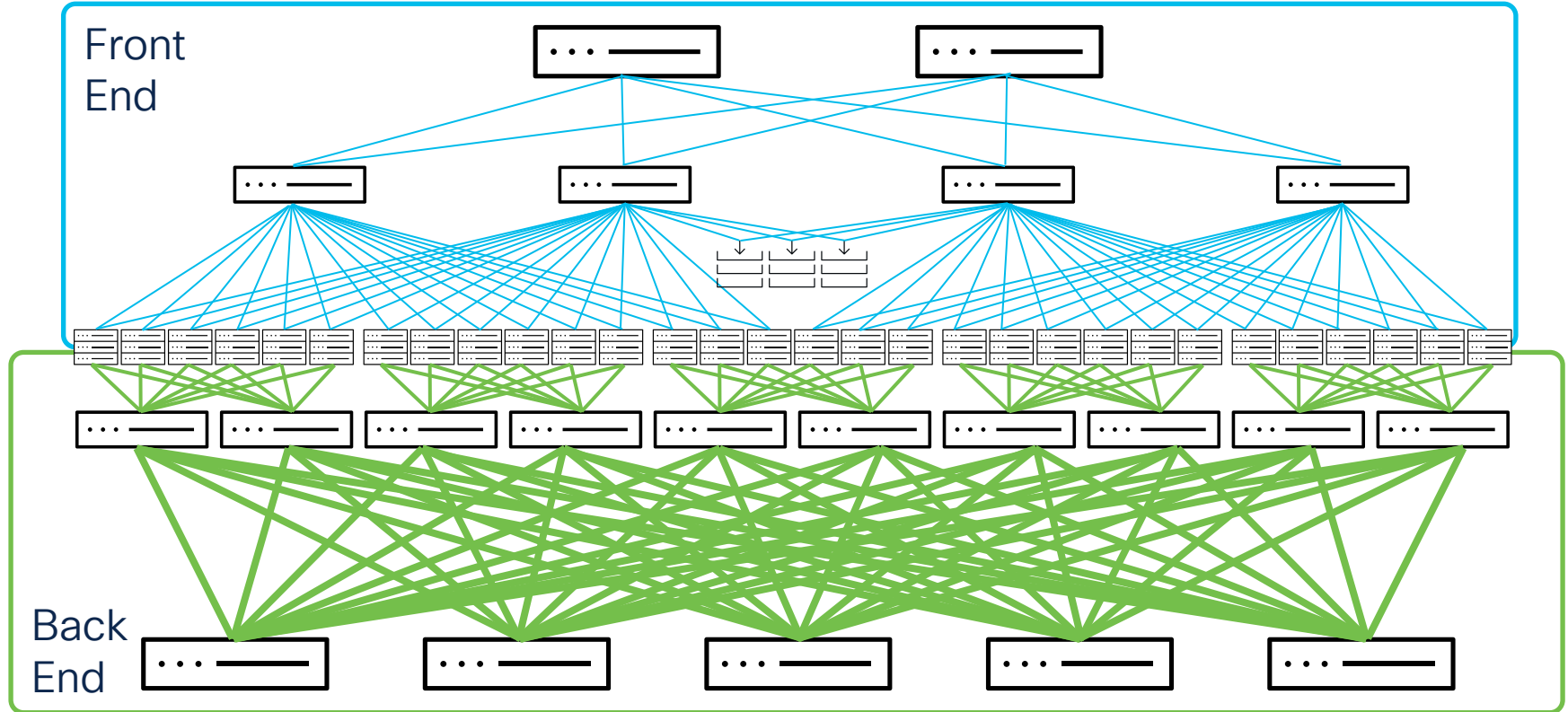
Front-End and Back-End Cluster Network



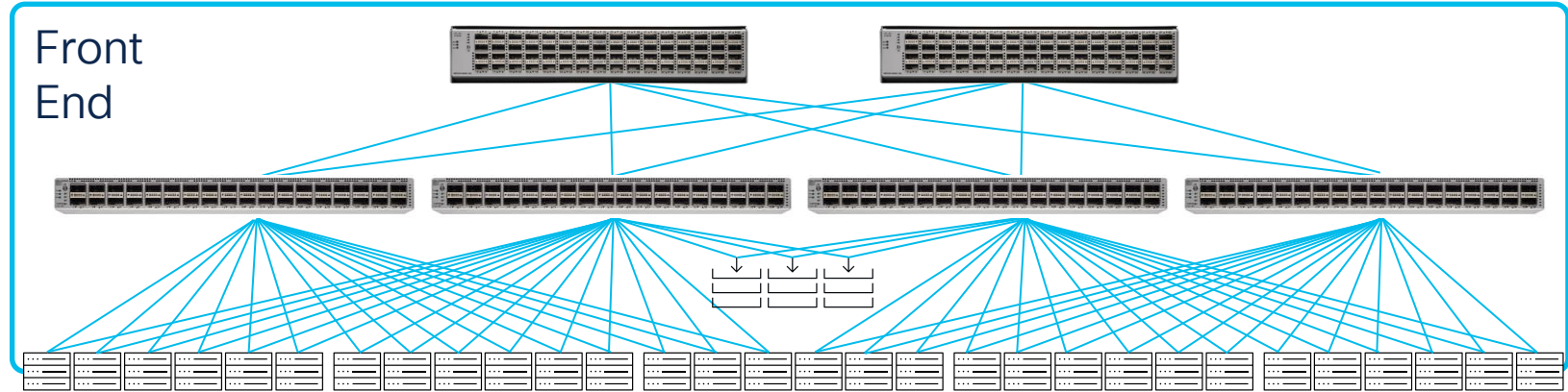
Front-End and Back-End Cluster Network



Front-End and Back-End Cluster Network



Front-End Cluster Network

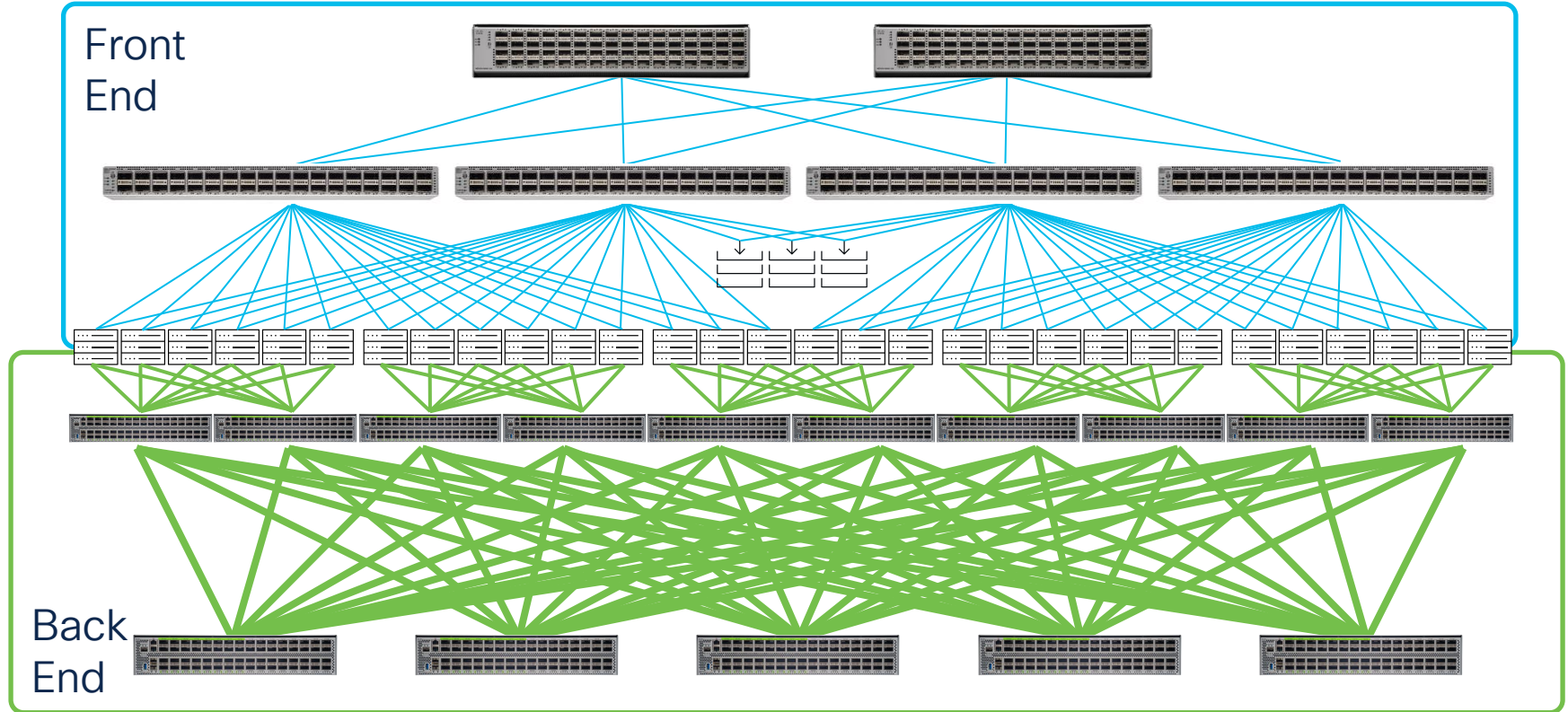


N9K-C9364C-GX

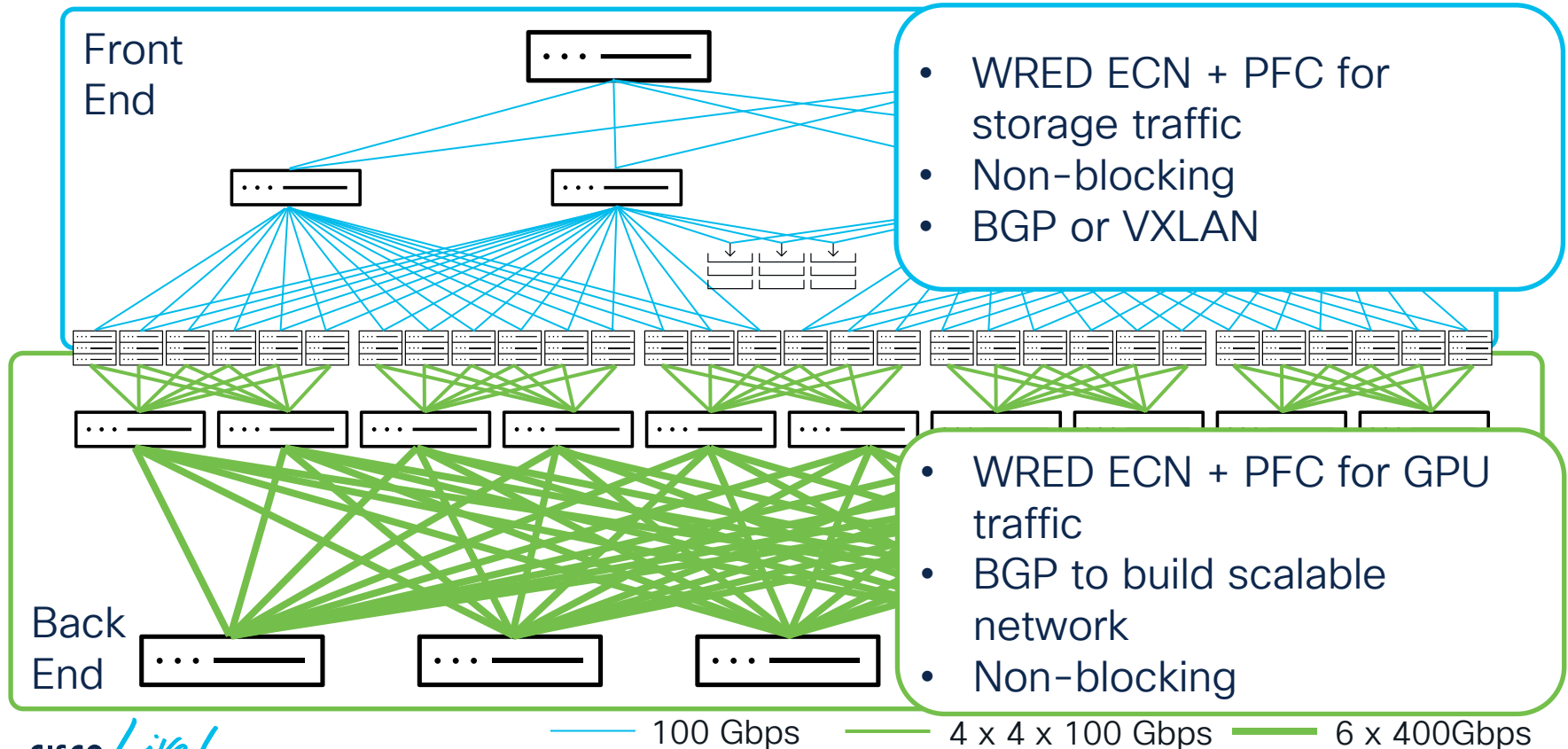


N9K-C9336C-FX2

Front-End and Back-End Cluster Network



Front-End and Back-End Cluster Network

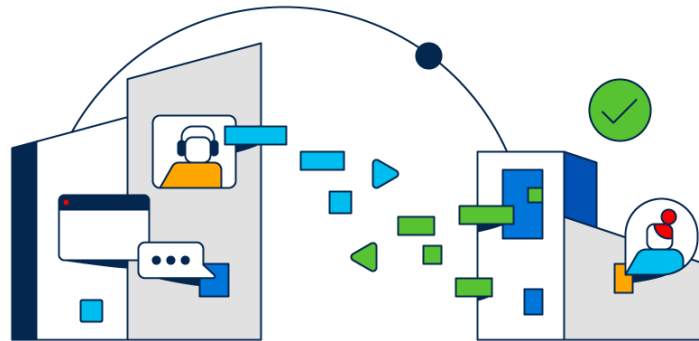


Conclusion



Takeaways

- Robust back-end network, and flexible front-end network
- Familiar data center fabric technologies, BGP or VXLAN
- Automate network for easier bring up and operation
- Visibility in network congestion, and bottleneck to troubleshoot and optimize



The Blueprint For Today



Products and Services Solutions Support Learn Partners

Explore Cisco Search

Products & Services / Cloud and Systems Management / Cisco Nexus Dashboard Fabric Controller / White Papers /

Cisco Data Center Networking Blueprint for AI/ML Applications

Updated: May 24, 2023

Bias-Free Language Contact Cisco

Introduction

Table of Contents

Introduction

RoCEv2 as Transport for AI Clu...

AI Clusters Require Lossless N... +

How to Manage Congestion Eff... +

How Visibility into Network Be... +

Network Design to Accommod... +

Conclusion

Related Materials

Introduction

RoCEv2 as Transport for AI Clusters

AI Clusters Require Lossless Networks

Explicit Congestion Notification (ECN)

Priority Flow Control (PFC)

How to Manage Congestion Efficiently in AI/ML Cluster Networks

How ECN Works

How PFC Works


Using ECN and PFC Together to Build Lossless Ethernet Networks

Using Approximate Fair Drop (AFD)

Save Download Print



The Blueprint For Today

 Product ▾ Solutions ▾ Open Source ▾ Pricing


Search / Sign in Sign up

allenrobel / NDFC-AIML-Fabric Public

Notifications Fork 0 Star 0 ▾

<> Code Issues Pull requests Actions Projects Security Insights

main ▾ 1 branch 0 tags

 allenrobel Ignore .graffle files 6101d38 47 minutes ago 10 commits

doc	Update with current 'show running-config ipqos'	1 hour ago
inventory	Initial commit	5 days ago
.gitignore	Ignore .graffle files	47 minutes ago
AIML_Fabric.yml	Template name will probably change, so let's not hardcode it	2 hours ago
AI_Cluster_QOS_template.template	Adding new QOS template	2 hours ago
README.md	Add topology diagram	5 days ago
ansible.cfg	Initial commit	5 days ago

☰ README.md

NDFC-AIML-Fabric

About

About

Ansible playbook to create an NDFC fabric which supports AI/ML workloads

📖 Readme

☆ 0 stars

👁 2 watching

🍴 0 forks

Report repository

Releases

No releases published

Packages

No packages published



The Blueprint For Today



Products and Services Solutions Support Learn Partners

Explore Cisco Search

Preferred Networks, Inc. Eliminates Overlapping Investment and Network Bottlenecks

< Back to URL

Updated: October 15, 2021

Bias-Free Language

Challenge



Table of Contents

Preferred Networks, Inc.

Challenge

Solution

Results and the future

Learn more

Save Download Print

Preferred Networks, Inc. chose Cisco due to high reliability, quick response to the latest protocol, and hardware-based streaming telemetry and used Integrated Interconnect Network for deep-learning computing infrastructure into Ethernet and eliminated overlapping investment and network bottlenecks.

Executive Summary

Customer Name: Preferred Networks, Inc.

Industry: Artificial Intelligence

Location: Chiyoda-ku, Tokyo

Number of Employees: Approximately 300



Related sessions

Session ID	Session Title	Day and Time
IBODCN-1010	An Interactive Conversation: AI/ML Networking Requirements and Blueprint	Wednesday, Feb 7, 11:30 AM Thursday, Feb 8, 5:15 PM
BRKDCN-2999	Multi-Tier Fabric-Networks Designs for the Modern Data Center	Thursday, Feb 8, 10:45 AM
BRKDCN-1619	Introduction to NDFC: Simplifying Management of Your Data Center	Tuesday, Feb 6, 3:30 PM
PSODCN-1732	Unlocking the Potential of AI/ML Workloads in Cisco Data Center Networks With Cisco Nexus 9000 Series Switches	Wednesday, Feb 7, 2:10 PM



The bridge to possible

Thank you

CISCO *Live!*

The background features a vibrant, multi-colored abstract design. On the left, there are horizontal, wavy bands of color in shades of red, orange, yellow, and green. On the right, a bright white light source emits a series of sharp, radiating lines in various colors, including blue, green, and yellow, creating a sunburst effect.

cisco *Live!*

Let's go