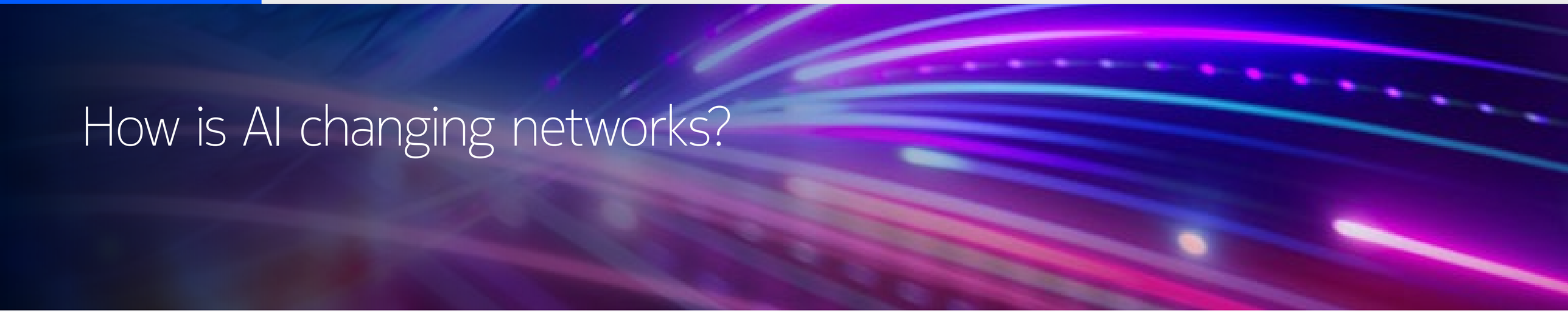

NOKIA
Reliable AI data center networking
From the latest Ethernet techniques to emerging Ultra Ethernet Consortium (UEC) approaches for AI networking

Contents

The impact of AI networking



How is AI changing networks?

Artificial intelligence (AI) is changing the world as we know it by transforming industries, enhancing daily life and reshaping societal structures. For evidence of this transformation, we need look no further than our own day-to-day lives. It is now becoming natural to use an AI application for text generation, content summarization, code generation, language tra nslation and sentiment analysis on a body of text.

This is just the beginning. AI applications are poised to impact virtually every industry—from finance to insurance, human resources, healthcare and beyond. Think autonomous transportation, personalized healthcare, smart shopping, hyper personalized smart homes and advanced personalized assistants to name a few.

But all this new AI technology comes at a price. This price is much more compute, much more power consumption, and a different approach to networking.

What is AI training and why is the network crucial for its success?

AI training builds models that learn patterns from data to make predictions, classify information or generate content. Starting from scratch, a model learns a specific task through an extensive computational process dictated from the theory of neural networking. The computing capabilities are provided by modern graphics processing units (GPUs), or accelerators.

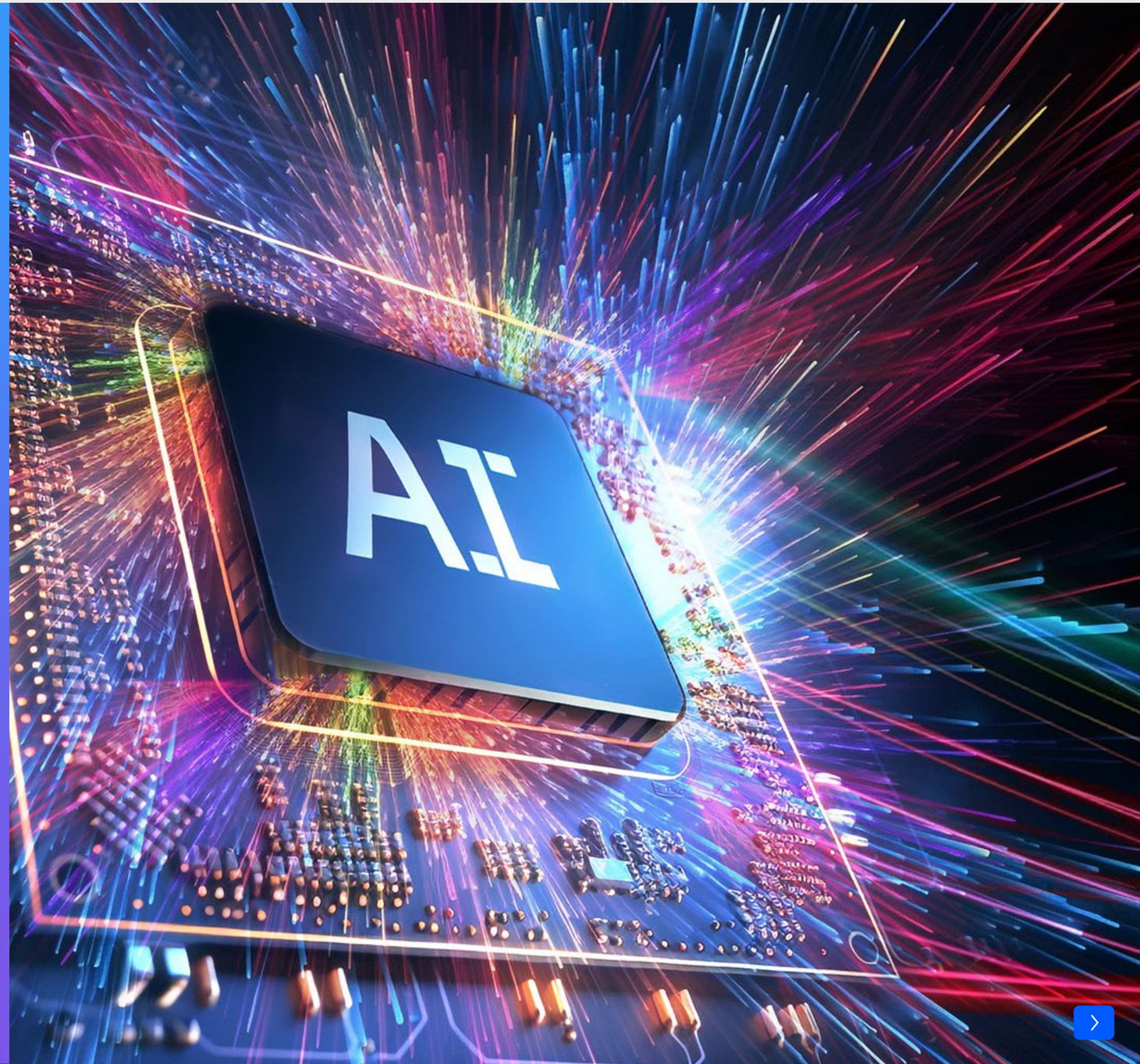
Training uses massive, structured datasets taken from many sources, including documents, the web, books, news, stock indexes, social media and synthetic generators. Because a single GPU's memory is often not large enough, data is split and loaded across many GPUs in a cluster, typically involving tens of thousands of GPUs for the largest models.

Each GPU processes its local data, then exchanges results with all others. These exchanges create "elephant flows" or traffic spikes, and without proper networking, can result in contention and data loss. Data loss results in costly restarts, rollbacks, wasted power and cooling..

Job Completion Time (JCT) measures the interval from job start to final checkpoint write. JCT depends on compute efficiency, storage throughput, cluster orchestration, and especially network performance—latency and packet loss can dominate the overall training time. Designing low-latency, loss-free networks is therefore critical for efficient AI training.

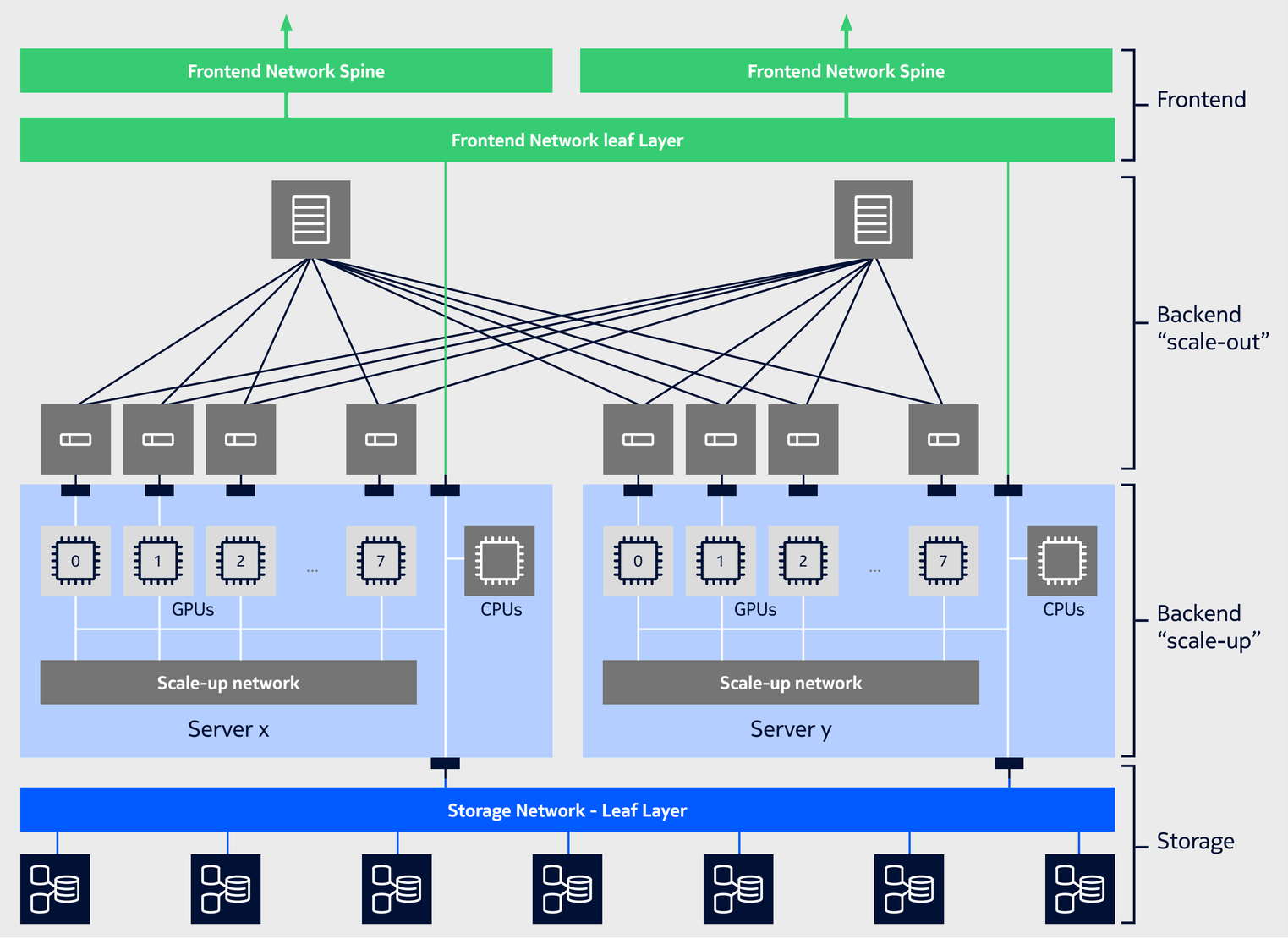
What is AI inferencing and how does it differ from training?

Inferencing is the process by which an AI model generates responses for an end user or machine via an AI agent—a central processing unit (CPU)-driven proxy for the trained model. For inferencing, the fully trained model is loaded onto multiple GPUs because it rarely fits on a single GPU. When the AI agent receives a request, the GPUs collaborate, exchanging intermediate results to produce the final answer. As the model generates output, it creates "tokens" in real-time, represented to the user as a part of the response. As more tokens are generated a larger part of the response is delivered. This process requires high performance GPU connectivity and a highly performing network between the GPUs and the end used. Inference performance is measured in tokens per second (TPS).



Key elements of the AI network

Figure 1. Key elements of the AI network



Key parts of the AI network

A network designed for AI can look a lot different to the more traditional data center network and includes various parts that operate together to deliver AI services. Four main parts of an AI network are shown in figure 1:

Frontend network

The part of the infrastructure that manages how external systems and users interact with the AI workloads. It acts as the gateway between the outside world and the internal compute resources.

Backend "scale-out" network

High-performance network infrastructure designed to connect multiple GPU within a large GPU cluster across servers, racks, and rows enabling distributed AI workloads.

Backend "scale-up" network

High-bandwidth, low-latency infrastructures that connect GPUs within a single server or tightly coupled cluster to accelerate AI training and inference.

Storage network

High-performance, scalable systems that deliver fast, reliable access to massive datasets required for training and inference.

Current AI networking techniques and protocols

Let's explore a few techniques being used today to optimize network performance so that AI training and inferencing can deliver the desired results.

The importance of RDMA and backend networks

Remote direct memory access (RDMA) provides the most efficient way to transfer data between GPUs for AI training and inference. Using a "zero copy" method, RDMA lets GPUs read/write each other's memory directly, bypassing the CPU and operating system (OS), which boosts throughput, scalability and security and dramatically cuts latency.

In AI GPU clusters, RDMA is used to transfer memory between GPUs within and across servers. This approach is essential for fast data exchange and synchronization between GPUs.

InfiniBand, Ethernet and RoCEv2

InfiniBand was built for RDMA traffic and has dominated AI and high-performance computing (HPC) networking. It uses large windows for data transfer and guarantees end-to-end delivery, eliminating packet loss.

Ethernet is widely adopted for its cost effectiveness, openness and scalability, making it a strong alternative for hyperscalers, cloud providers and enterprises.

As shown in Figure 2, modern Ethernet back-end solutions retain InfiniBand's transport layer but encapsulate it in User Datagram Protocol (UDP), IP and Ethernet via the RoCEv2 protocol. RoCEv2's routable IP encapsulation supports AI workloads and high-performance computing in cloud and enterprise settings.

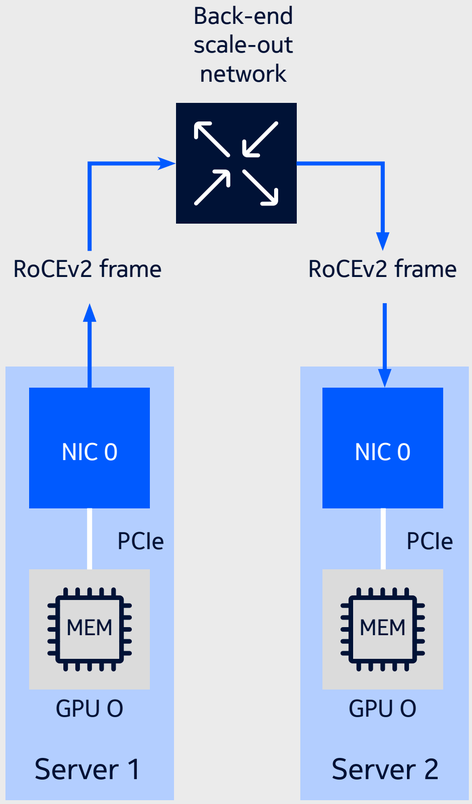
Figure 2. InfiniBand and RoCEv2 protocol data units

<table>
<tr><th>Infiband PDU</th><th>RoCEv2 PDU</th></tr>
<tr><td>Applications</td><td>Applications</td></tr>
<tr><td>RDMA</td><td>RDMA</td></tr>
<tr><td>InfiniBand Transport</td><td>InfiniBand Transport</td></tr>
<tr><td rowspan="3">Infiniband</td><td>UDP</td></tr>
<tr><td>IP</td></tr>
<tr><td>ETHERNET</td></tr>
</table>

Transmitting RDMA using RoCEv2

GPU memory is sent between servers over the back-end scale-out network using InfiniBand PDUs while inside the servers and RoCEv2 while on the scale-out network between the network interface cards (NICs). Figure 3 shows how this process works.

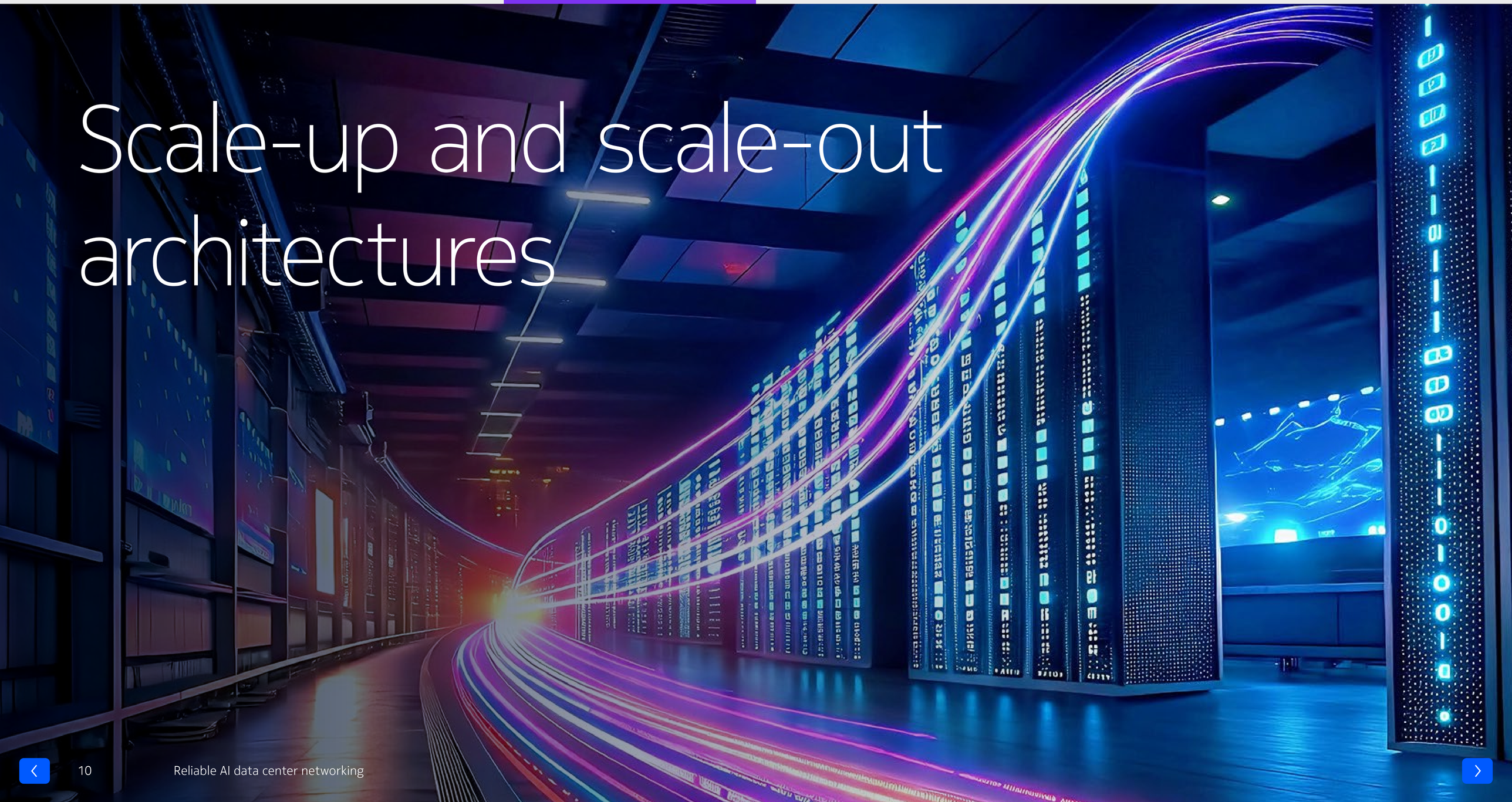
Figure 3. Transmitting GPU memory using RDMA and RoCEv2



RDMA data transmission

1. Data read directly from memory on server 1
2. RDMA data chunks sent from GPU 0 on server 1
3. RDMA chunks sent across PCIe on server 1
4. NIC 0 on server 1 encapsulates chunks in RoCEv2 frames
5. NIC 0 on server 1 sends RoCEv2 frames across scale-out network
6. RoCEv2 chunks received by NIC 0 on server 2
7. NIC 0 on server 2 de-encapsulates RoCEv2 frames into RDMA data chunks
8. RDMA chunks sent across PCIe on server 2
9. Data written directly to memory on server 2

Scale-up and scale-out architectures



Direct interconnect or "scale-up" architectures

Direct GPU interconnects—also called the scale-up network—provide the fastest memory transfer between GPUs or accelerators. Using RDMA, technologies such as NVIDIA NVLink, AMD Infinity Fabric, and UALink can move data at up to 1.8 Tb/s with latency of only a few microseconds. This architecture is ideal for AI training and inference, but it requires the GPUs to reside in the same server (or, in some cases, the same rack or row). Figure 4 depicts an example of a scale-up network contained within one server. The dotted lines in figure 4 represent memory transfer between GPUs within the same server. Scale-up networks are expanding and can also extend to entire racks and even rows.

Figure 4. Scale-up design for AI back-end networking

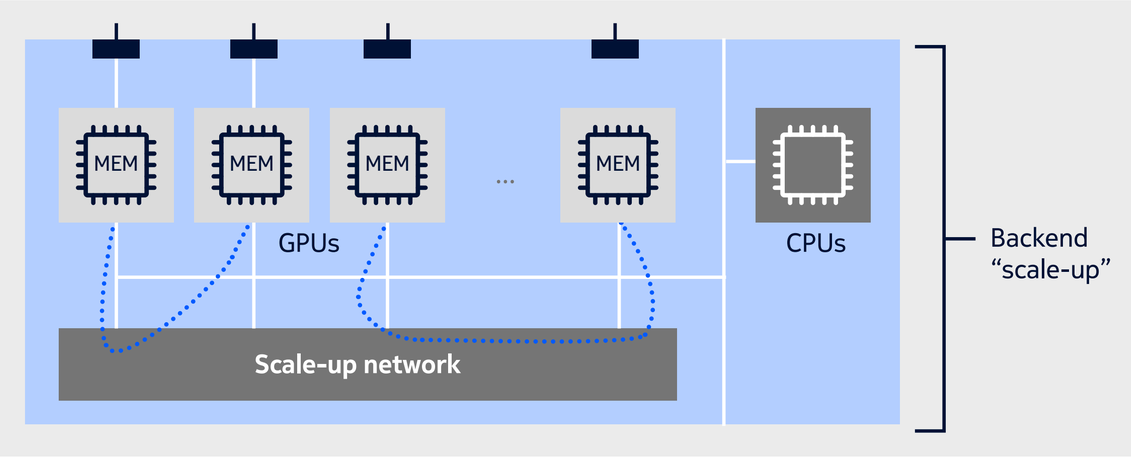
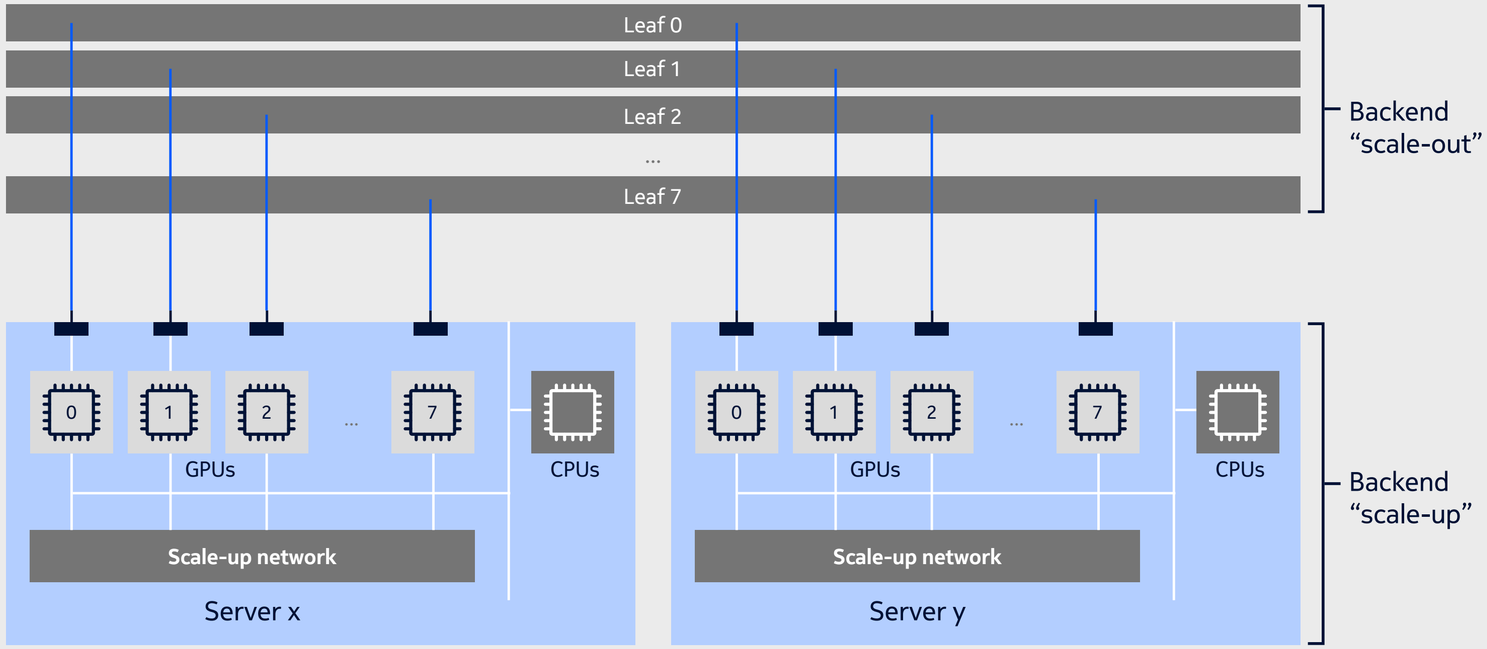


Figure 5. Rail design for scale-out AI back-end networking



Scale-out networks

Scale-out networks are the backend interconnect fabrics that links many data center servers (and the GPUs inside them) across a data center rack or GPU cluster. Their primary role is to move large volumes of GPU memory and training data between nodes quickly and efficiently to enable distributed AI training and inference.

Rail design for scale-out backend networks

The rail design is a common architecture used for backend scale-out networking. The AI back-end networking rail design links GPUs across servers and racks while minimizing latency and bandwidth contention. The rail design distributes traffic across multiple independent rails to achieve high bandwidth, low latency for large scale training and inference. With the rail design there is never more than a single hop across the "scale-out" network.

In the example shown in Figure 5, each server has eight GPUs and eight NICs (ranks 0–7), and there are eight leaf switches (rails) with matching ranks (0–7). Each GPU connects to the leaf of the same rank. Three communication paths exist:

1. **In-server GPU-to-GPU traffic** uses the direct interconnect (scale-up network).
2. **Cross-server same-rank traffic** goes through the rail of that rank (one hop).
3. **Cross-server different-rank traffic** first uses the scale-up network to switch to the correct rank, then travels via the corresponding rail.

Congestion avoidance in scale-out architectures

Congestion avoidance for scale-out networks

AI backend networks often face congestion from bursty traffic and elephant flows—e.g., many GPUs sending updates to a single GPU. Because RoCEv2 runs over Ethernet/IP, it doesn't guarantee lossless delivery, so a couple of key congestion avoidance techniques are used to prevent packet loss.

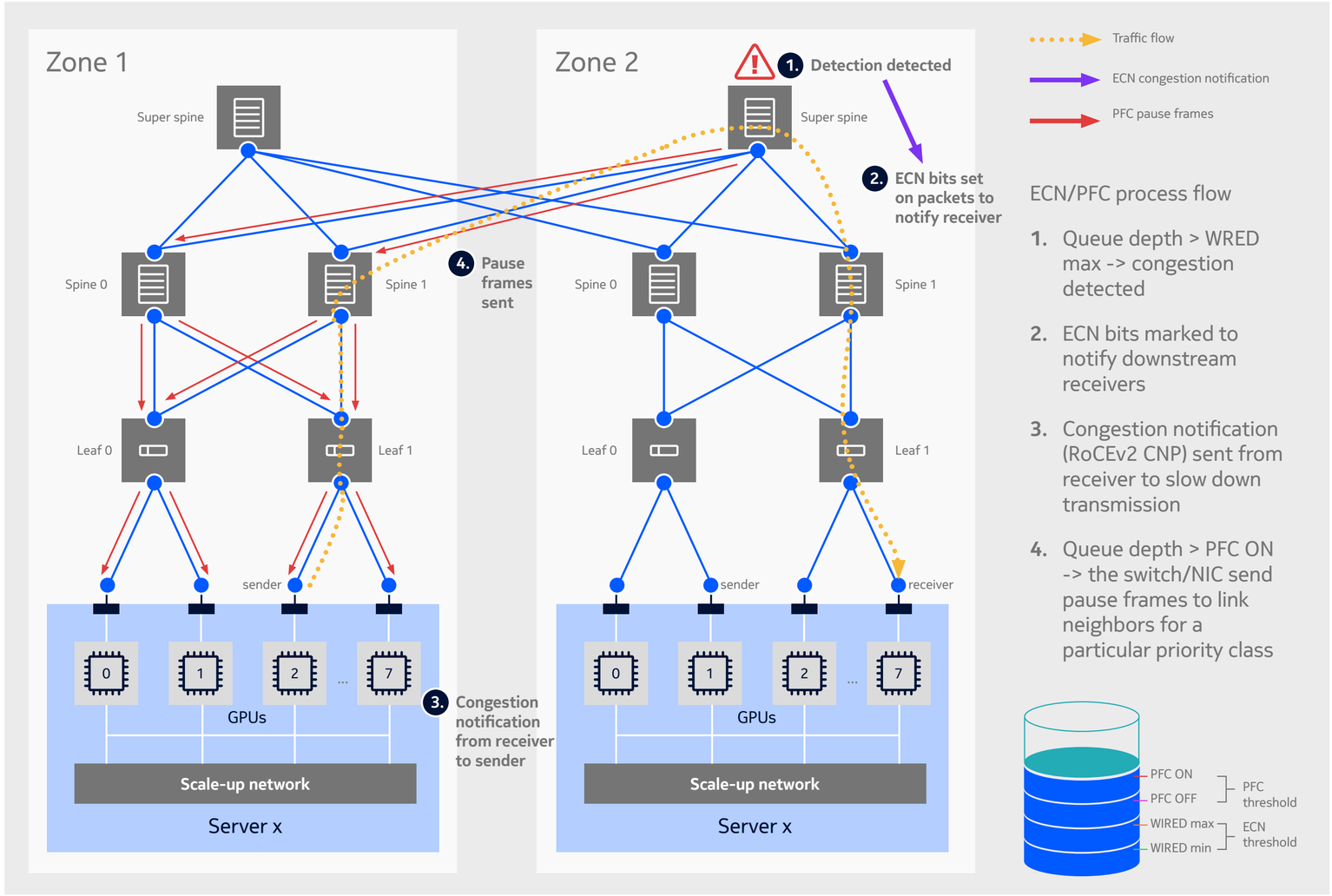
These two techniques work hand in hand and are triggered based on certain queue thresholds:

1. **Explicit Congestion Notification (ECN)**, is triggered first when switch queues exceed a weighted random early detection (WRED) threshold. Switches mark ECN bits on a linear proportion of packets as the queue depth increases between WRED (min) and WRED (max). Receivers notify senders, which then throttle their data transmission rate until congestion clears.

2. **Priority Flow Control (PFC)**, is activated only if ECN can't relieve congestion. When a queue a switch reaches the PFC ON threshold, it sends pause frames to connected downstream neighbors, within the impacted priority class, halting traffic until the queue drains or when the queue depth is below PFC OFF.

When the two are used together, ECN handles minor, end-to-end congestion, while PFC provides hop-by-hop per priority class back pressure for more severe cases, keeping AI training traffic lossless. Figure 6 depicts ECN and PFC in action. Please note that the architecture in figure 6 is strictly used to show the behavior of ECN and PFC.

Figure 6. DCQCN with ECN and PFC



Data Center Bridging Capability Exchange protocol

The Data Center Bridging Exchange Protocol (DCBX) extends Link Layer Discovery Protocol (LLDP) so networking devices (i.e., NICs, leaf–spine switches) can discover and share Data Center Bridging (DCB) settings, ensuring loss less Ethernet for AI back-end networks. DCBX exchanges parameters such as Priority-based Flow Control (PFC) to spot and fix configuration mismatches. NICs can be set from leaf switches when in "willing" mode, but Nokia SR Linux leaf switches won't alter their PFC settings based on DCBX. They display the information only for troubleshooting.

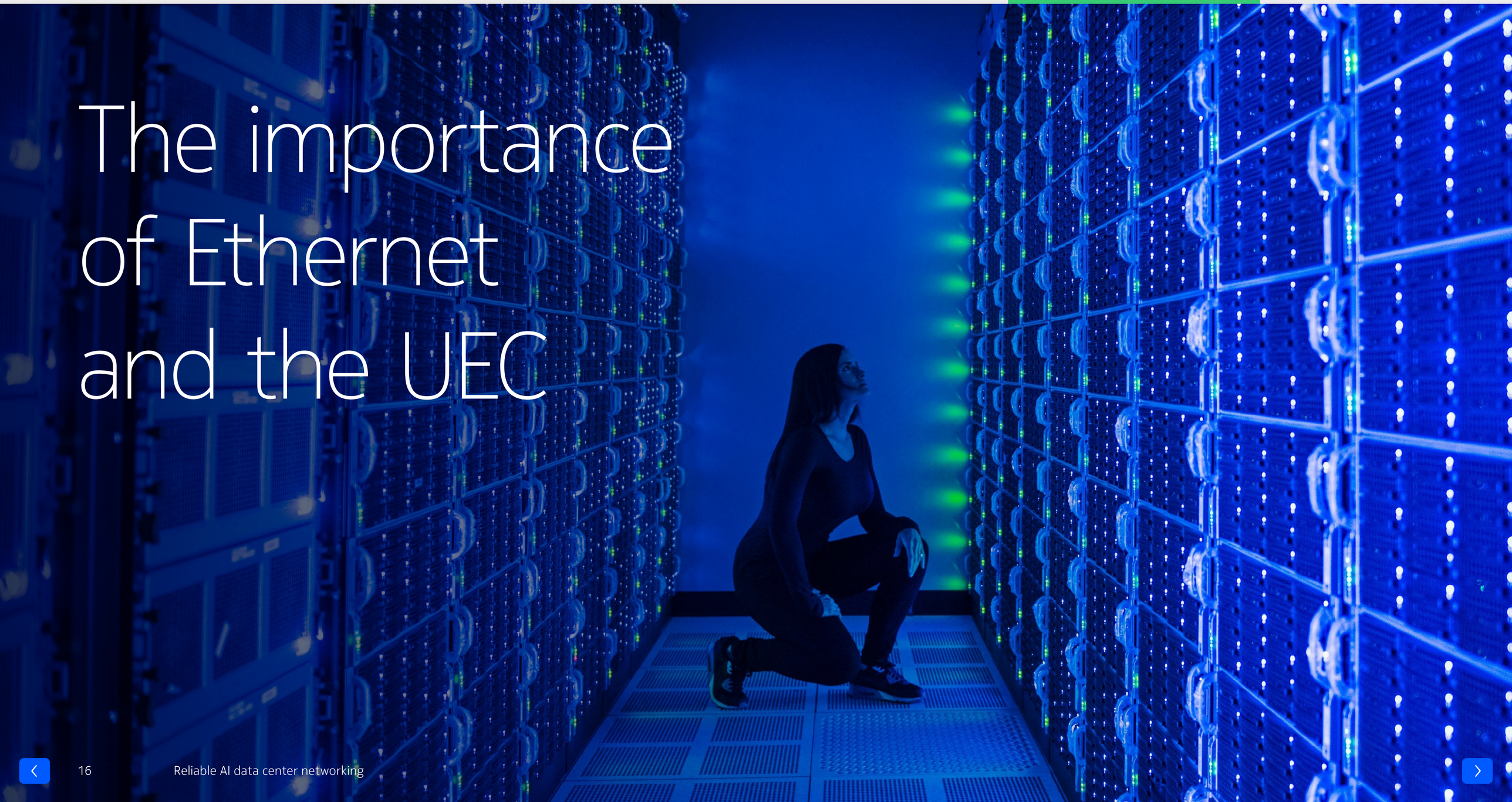
Load balancing techniques in back-end networks

AI training and inference demand heavy computation and require many GPU accelerated servers to stay synchronized. Traffic patterns are uneven, with "elephant flows"—long lived, bandwidth-intensive transfers—that can overwhelm static load balancing schemes.

Dynamic load balancing (DLB) is therefore crucial for high-performance AI back-end networks, which handle unpredictable, resource-intensive, latency-sensitive workloads. Unlike static methods that rely on fixed rules, DLB continuously monitors network health and performance, redistributing traffic in real time to avoid bottlenecks, improve utilization and maintain availability.

DLB enhances traditional equal-cost multipath (ECMP) routing by using a state-aware approach: it evaluates the current load of aggregate members when assigning flows, allowing existing flows to adapt to changing conditions without causing packet reordering. A feedback loop detects imbalances and updates the load balancer's hashing algorithm based on metrics such as egress port queue depth, egress port utilization and ingress traffic manager (ITM) queue size.

The importance of Ethernet and the UEC



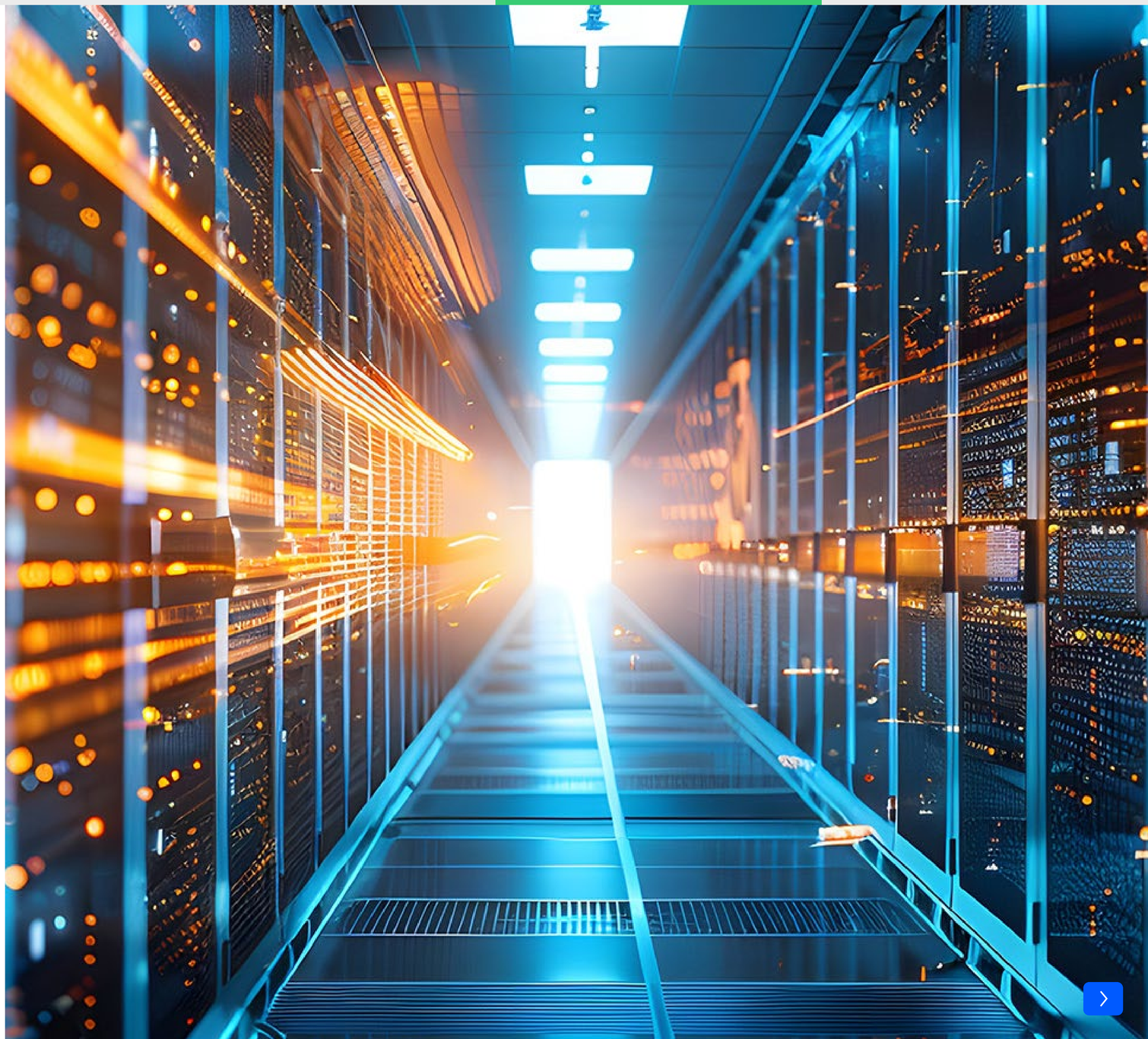
Future techniques with UEC Specification 1.0

Ethernet as the future of AI networking

InfiniBand has set the benchmark for AI back-end networking, but its closed, proprietary nature has opened the door for Ethernet to become the preferred, future-proof solution for GPU-based AI training and inference.

Why Ethernet?

- **Broad ecosystem:** Switches, NICs, test gear, SFPs, and open-source management tools.
- **Rapid innovation:** Continual upgrades in protocols, link speeds, optics, and cabling.
- **Universal familiarity:** Widely understood and easily adopted by engineering teams.
- **Scalable with IP:** Proven to expand across massive, super-scale networks.
- **Open, multivendor:** Flexible choice of vendors and components.



Why UEC Specification 1.0?

The Ultra Ethernet Consortium (UEC) is a collaborative organization focused on advancing Ethernet technology to meet the demands of high-performance computing (HPC) and artificial intelligence (AI) applications.

In UEC specification 1.0, RoCEv2 is being replaced with an open, interoperable Ethernet stack tailored for AI and HPC workloads, focusing on scalable, back-end scale-out networks.

RoCEv2 represents the status quo for Ethernet-based scale-out networks but it has its drawbacks including:

- Scaling limitations for large scale deployments.
- Congestion prone and loss recovery inefficiency.
- Complex configuration requirements.
- Vendor-specific implementation of congestion avoidance t echniques.

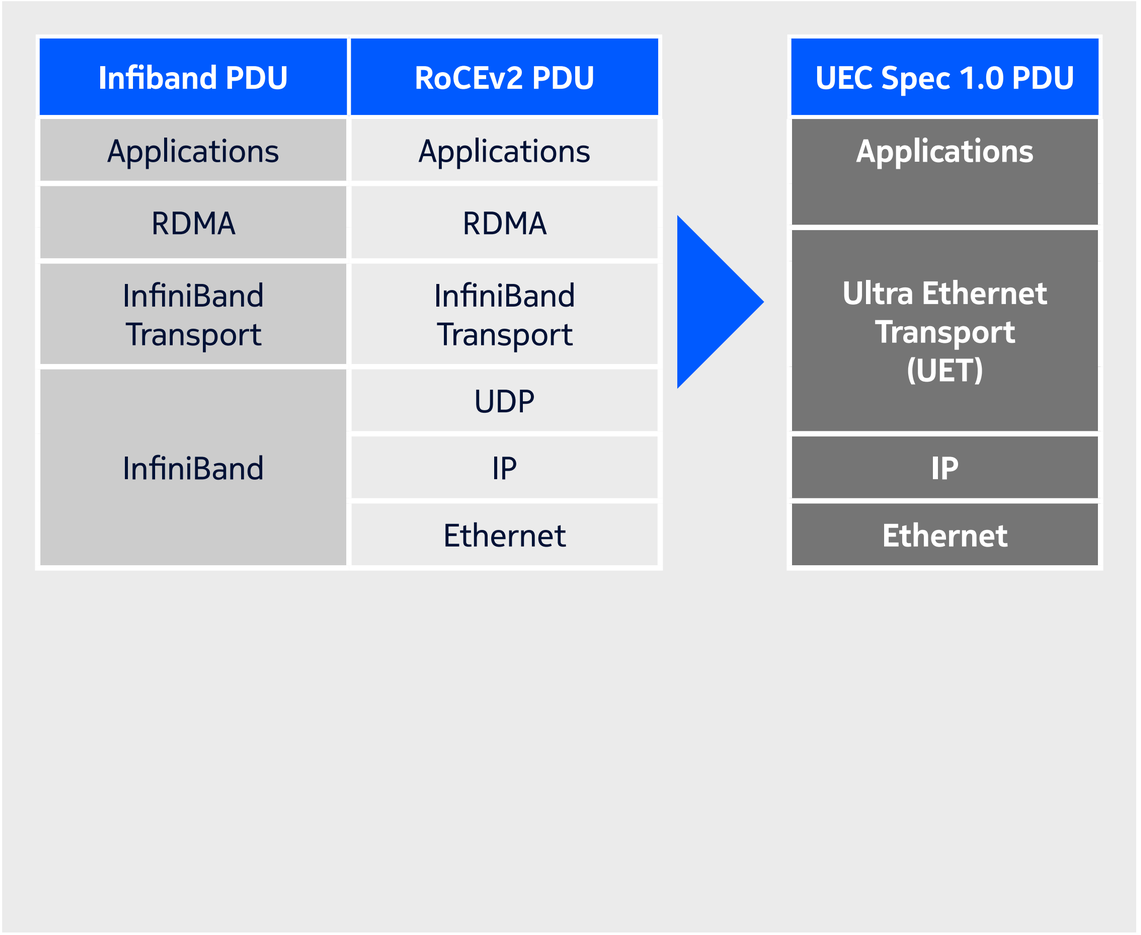
As shown in figure 7, UEC specification 1.0 includes four main layers:

- **Application layer:** Provides an API for remote memory access in HPC and AI networks.
- **Ultra Ethernet Transport (UET) layer:** An RDMA protocol that replaces RoCEv2, adding better multipath out-of-order delivery and security.
- **IP layer:** Mostly unchanged, with an optional packet trimming feature to ease congestion and boost efficiency.
- **Ethernet layer:** Largely unchanged, offering an optional link layer retransmission extension.

Multipath out-of-order packet transmission

UEC's "packet spraying" sends packets simultaneously over all viable paths, boosting network utilization and easing congestion. Tagged packets can be reassembled in memory regardless of arrival order, preventing hot spots caused by large, unevenly balanced flows.

Figure 6. Layers of UEC specification 1.0



Summary of key UEC Specification 1.0 features

UEC Specification 1.0 is designed to scale efficiently across very large AI clusters. It reflects a rethinking of RoCEv2 and related congestion avoidance techniques. Below are some of the key features introduced in UEC Specification 1.0.

Packet trimming (IP layer)

When a packet hits a congested queue, it is truncated to a much smaller size and moved to a high-priority queue.

Packet trimming provides the following benefits:

- Reduces congestion by sending much smaller packets.
- Due to transmission in high-priority queues, it gets to the receiver earlier and retransmission can start sooner.

Congestion control

UEC's algorithms quickly ramp packet flow to wire rate, even with packet spraying, while keeping traffic stable under congestion.

Security

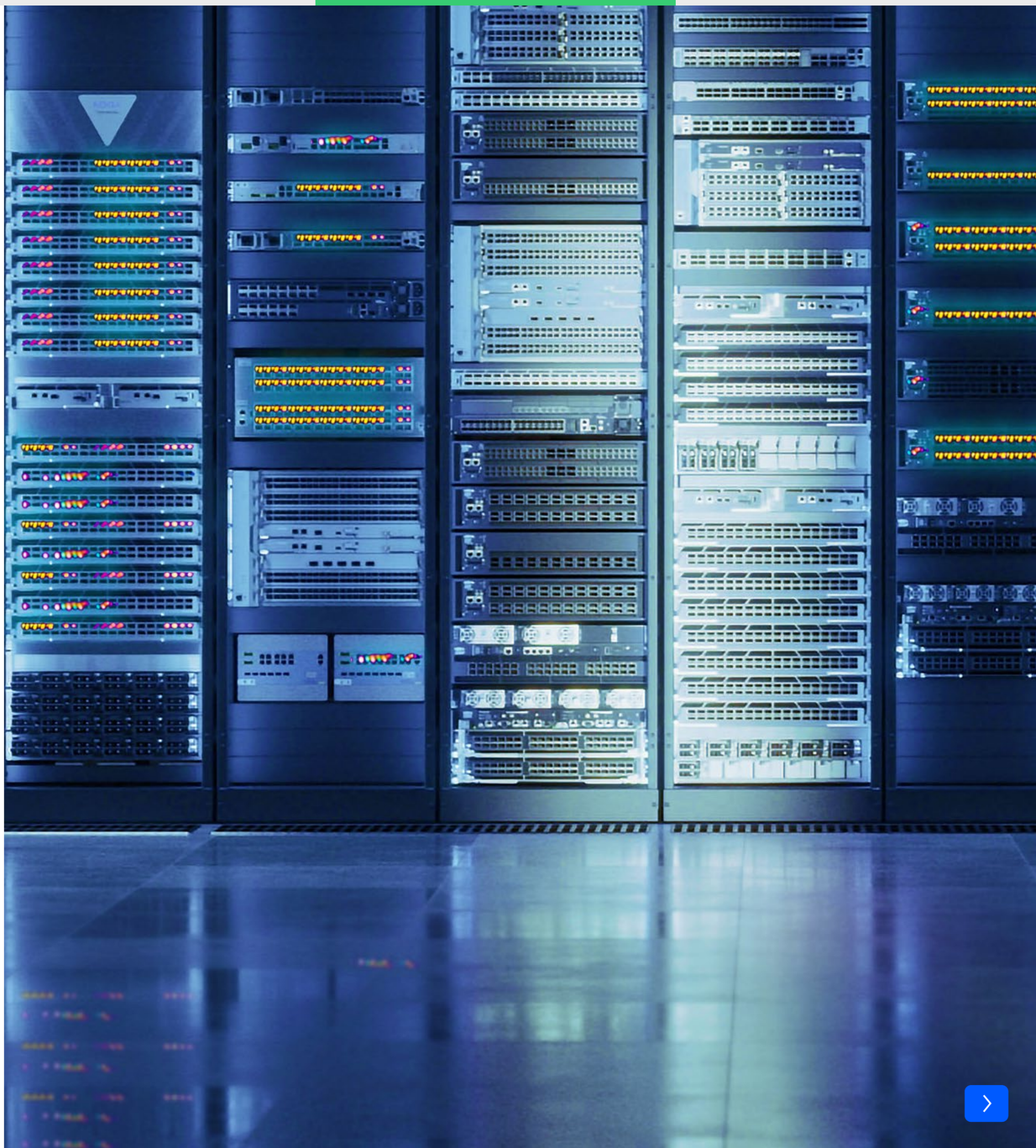
End-to-end encryption and authentication (AES-GCM, KDFs, anti-replay) plus a group-key scheme for AI and HPC workloads.

API

Adds Libfabric support for common HPC and AI operations (e.g., read-write, send/receive).

Link Layer Retry (LLR)

LLR is an important tool to improve the reliability of marginal links that may be a result of transient physical layer disruption, intermittent failing components, or faulty wiring. LLR is a hop-by-hop technology negotiated by the Link Layer Discovery Protocol (LLDP) that allows packets to be retransmitted in the case of loss between two link partners.



Nokia's commitment to Ethernet and the UEC

Nokia has been a leader in Ethernet and IP networking for decades. Our networking solutions have been proven in the largest and most demanding networks in the world. We offer the industry's most open, programmable and reliable networking capabilities and have built these characteristics into our AI back-end data center switching solution.

We play an active role in the UEC and are already building features based on UEC Specification 1.0. Capabilities such as out-of-order packet spraying, packet trimming and more are in development and we expect to implement them in an upcoming release. We recently tested the **transmission of UET traffic** on our high-performance 7220 IXR and 7250 IXR switches, showcasing our early commitment to the UEC specification.

Together with our customers and partners, we are working to ensure compatibility across any AI networking infrastructure. Our work with partners such as **Supermicro**, **Lenovo** and **Kyndryl** offers just a few examples of our commitment to ecosystem partnerships.

We have published both a **blog** on the importance of Ethernet for AI networking, as well as a **video** on Nokia's support of Ethernet and the UEC.

Partner with a committed leader in AI networking. For more information on our AI networking solution, visit our **AI Data Center Networking** page.

Nokia AI data center networking solution

Nokia AI data center networking solution

The Nokia AI data center networking solution provides the openness, programmability and extreme reliability you need to build and deploy network infrastructures that can meet the requirements of current and future AI and HPC workloads.

Modular and fixed-configuration platforms

Nokia's solution includes a comprehensive portfolio of modular and fixed-configuration hardware platforms for implementing high-performance leaf–spine designs for AI and HPC. You can use our platforms to build high-capacity, low-latency and lossless back-end networks that can efficiently handle demanding AI training workflows. We also offer platforms in many different form factors to support frontend network designs that will interconnect your AI inference compute, non-AI compute and shared storage resources.

The fixed-configuration 7220 IXR platforms (7220 IXR-D, 7220 IXR-H) provide high capacity and high density for data center leaf–spine deployments.

The modular 7250 IXR-6e/10e/18e platform and fixed 7250 IXR-X1b/X3b platform let you provide massive-scale interconnectivity for supporting the demanding needs of AI and HPC workloads.

7215 Interconnect System (IXS) provides reliable out-of-band management network solutions for data center servers, spine nodes and leaf nodes.

Nokia data center switches

A comprehensive portfolio of data center switches that can help you implement backend and frontend data center networks that meet the demands of artificial intelligence (AI) and traditional workloads.

A powerful and proven network operating system

Nokia's data center hardware platforms run on the Nokia SR Linux network operating system (NOS). SR Linux opens the NOS infrastructure with a unique architecture built around model-driven management and modern interfaces. Designed for openness, programmability and, most of all, extreme reliability and quality, ready for automation at scale, and easy to customize and extend.

SR Linux supports ECN and PFC congestion management techniques and traffic prioritization capabilities that let you deliver lossless Ethernet networking. It also supports high-performance AI infrastructures with superior telemetry, manageability, ease of automation and resiliency features.

SR Linux implements UEC techniques to help evolve Ethernet and establish it as the standard for AI networking.

Fabric management and automation toolkit

Nokia's Event-Driven Automation (EDA) is a modern data center network automation platform that combines speed, reliability and simplicity. It makes network automation more trustable and easier to use wherever you need it, from small edge clouds to the largest data centers.

With EDA, you can automate the entire data center network lifecycle from initial design to deployment to daily operations. This allows you to ensure reliable network operations, simplify network management and adapt to evolving demands. You can also use EDA to configure and automate scale-out networks.

EDA offers a range of flexible AIOps capabilities through its natural language interface. Operators simply use this interface to ask an operational question in the areas of troubleshooting, root cause analysis, remediation support, and more, and through an Agentic AI process, the system responds.

Nokia OYJ
Karakaari 7
02610 Espoo
Finland

Tel. +358 (0) 10 44 88 000

CID:215123

nokia.com

About Nokia

At Nokia, we create technology that helps the world act together.

As a B2B technology innovation leader, we are pioneering networks that sense, think and act by leveraging our work across mobile, fixed and cloud networks. In addition, we create value with intellectual property and long-term research, led by the award-winning Nokia Bell Labs, which is celebrating 100 years of innovation.

With truly open architectures that seamlessly integrate into any ecosystem, our high-performance networks create new opportunities for monetization and scale. Service providers, enterprises and partners worldwide trust Nokia to deliver secure, reliable and sustainable networks today – and work with us to create the digital services and applications of the future.