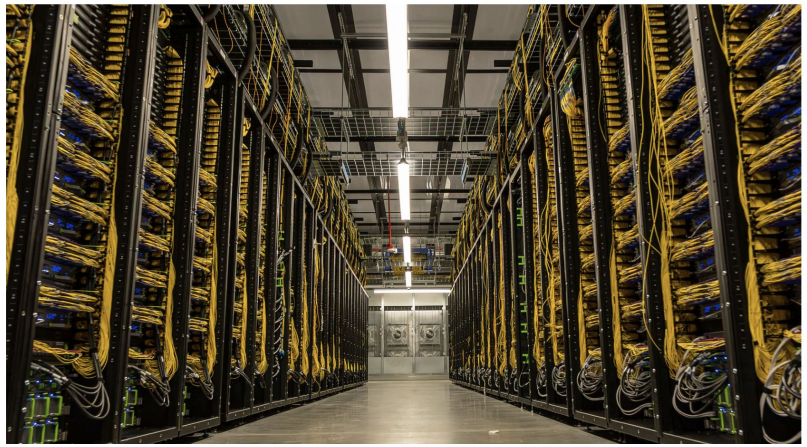
Arhan Jain & Ethan Shen

ML Networking

A recent hot topic

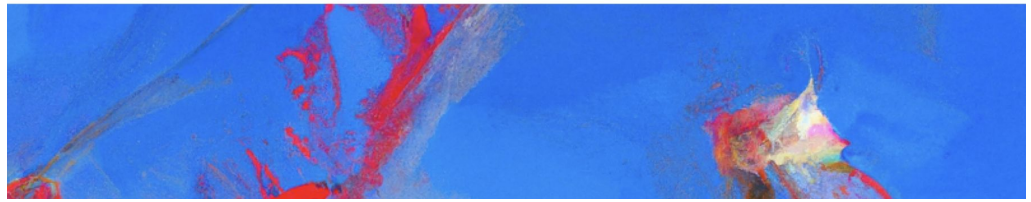
POSTED ON OCTOBER 19, 2026 TO CORE INFRA, DATA CENTER ENGINEERING, DATA INFRASTRUCTURE, ML APPLICATIONS, NETWORKING & TRAFFIC, PRODUCTION ENGINEERING

Inside Prometheus: Meta's Journey to a Gigawatt-Scale Supercluster



January 21, 2025 Company

Announcing The Stargate Project



The New York Times

Google Takes the A.I. Data Center Race to Outer Space

Next Thursday, Google is sending an experimental satellite into orbit that will have enough computing power to answer simple A.I. queries from space.

The AI datacenter: the new factory of the AI era

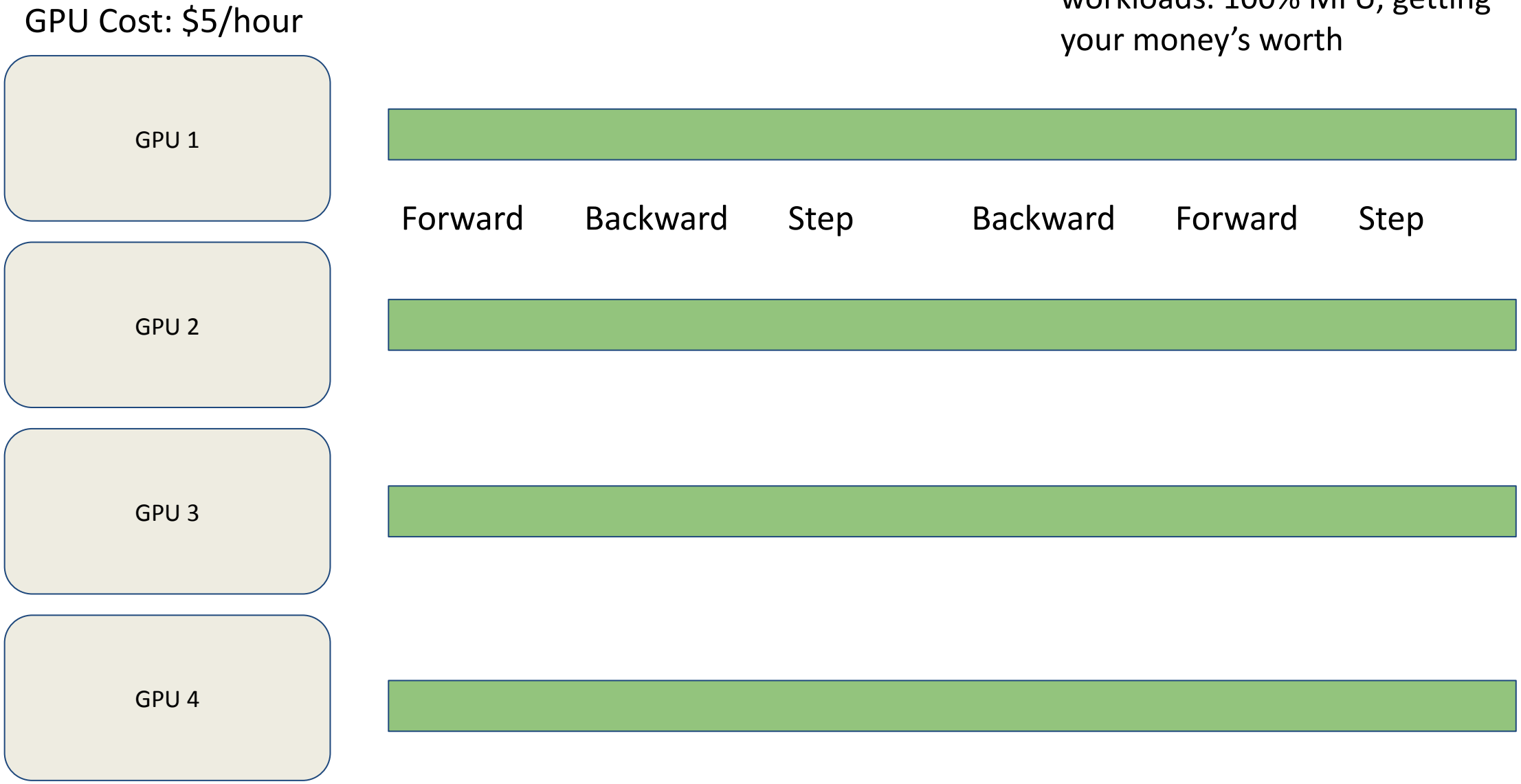


Aerial view of Microsoft's new AI datacenter campus in Mt Pleasant, Wisconsin.

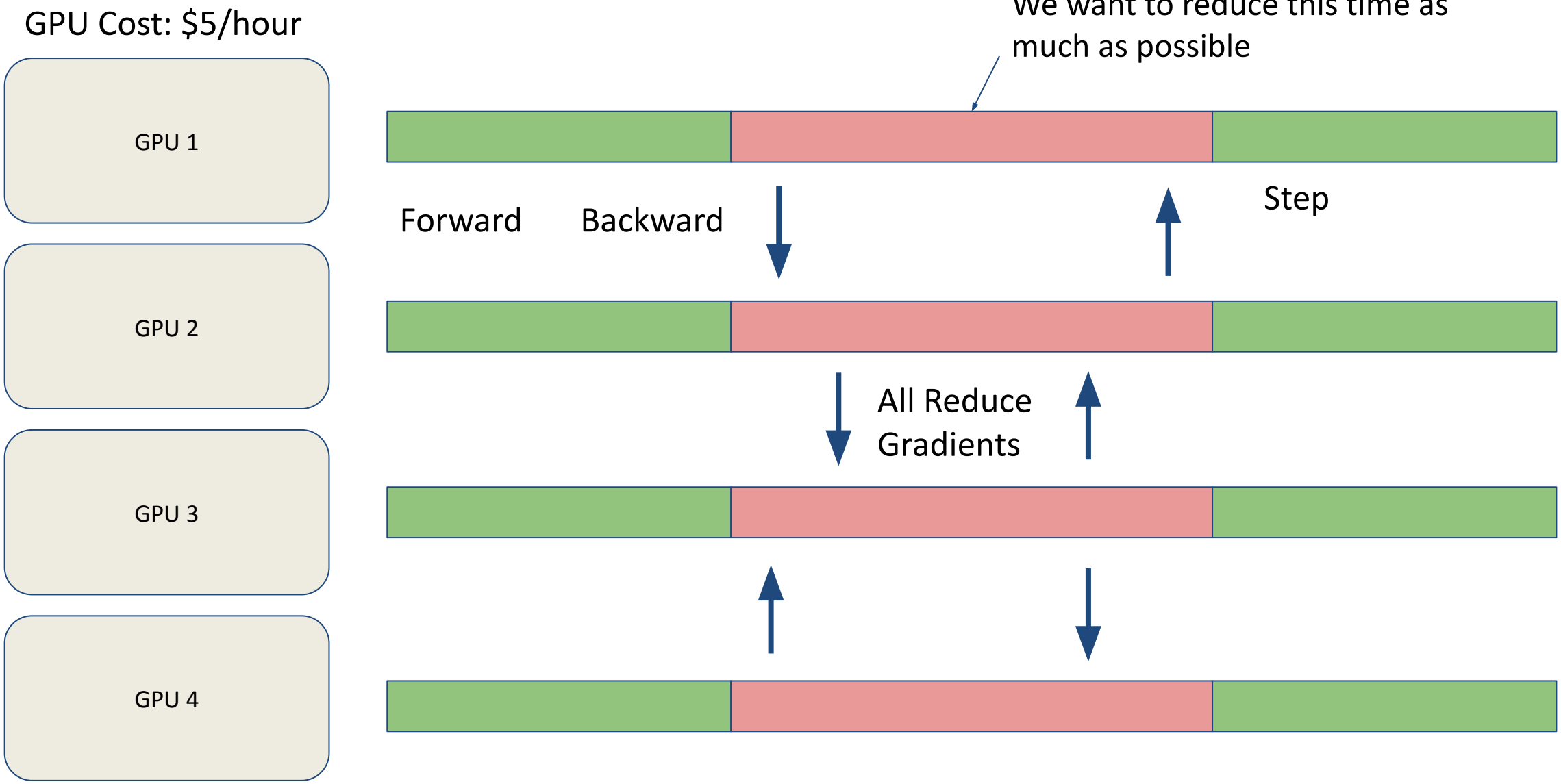
An AI datacenter is a unique, purpose-built facility designed specifically for AI training as well as running large-scale artificial intelligence models and applications. Microsoft's AI datacenters power OpenAI, Microsoft AI, our Copilot capabilities and many more leading AI workloads.

Why is networking so important to ML?

Case study: Independent workloads. 100% MFU, getting your money's worth

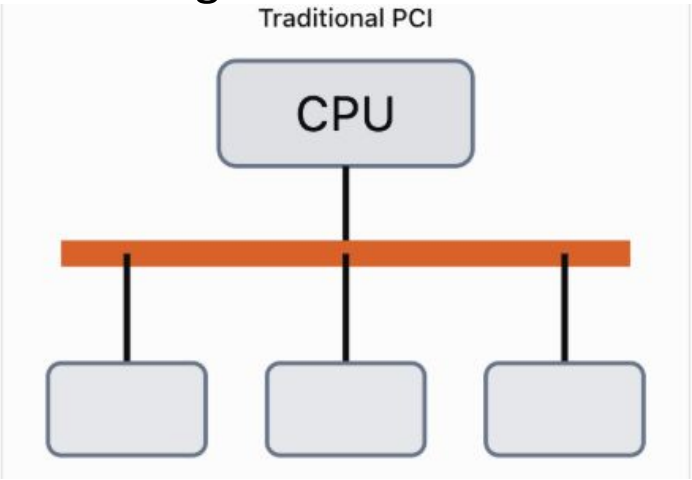


Why is networking so important to ML?



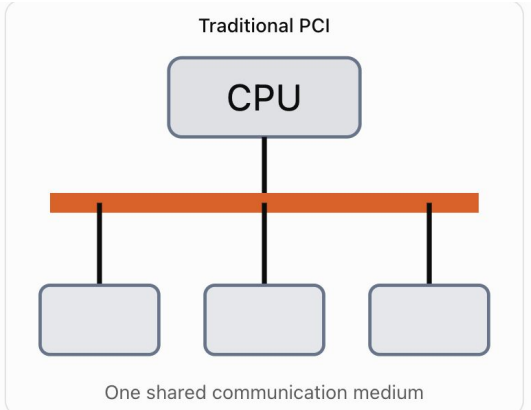
An Introduction to the InfiniBand™ Architecture

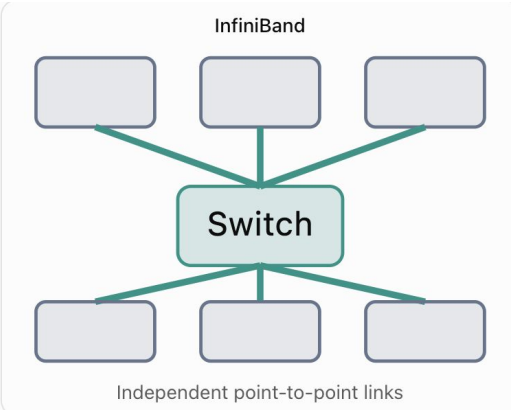
- What is InfiniBand?
  - The InfiniBand Architecture (IBA) is a “new” industry-standard architecture for server I/O and inter-server communication
  - Developed by a consortium of many companies you’ve probably heard of in tech
-
- Reasons for the development of InfiniBand
  - Traditional bus i/o is power inefficient
    - Shared bus systems are inefficient due to the orchestration deciding which bus can transmit at a time
    - Processor idling during i/o loads. on order of 30% based on benchmarking
    - Fault tolerance (if a device goes down it could be unusable)



An Introduction to the InfiniBand™ Architecture

- Key Characteristics
  - Point 2 Point: allowing simultaneous device communication leveraging networking switches





  - Channel semantics
    - Decouples processing from I/O - enabled async. More on this later

InfiniBand Subnet

- All links are bidirectional, with the options of increasing width for greater bandwidth

- Switches have routing tables built on initialization and networking modifications

- End nodes can be
  - Host systems
  - Storage systems
  - Legacy Shared Bus systems

- InfiniBand specifies standardization against link types, end node form factor, power dissipation, etc.

- Host Channel Adapters and Target Channel Adapters
  - Only host is required to expose software interface

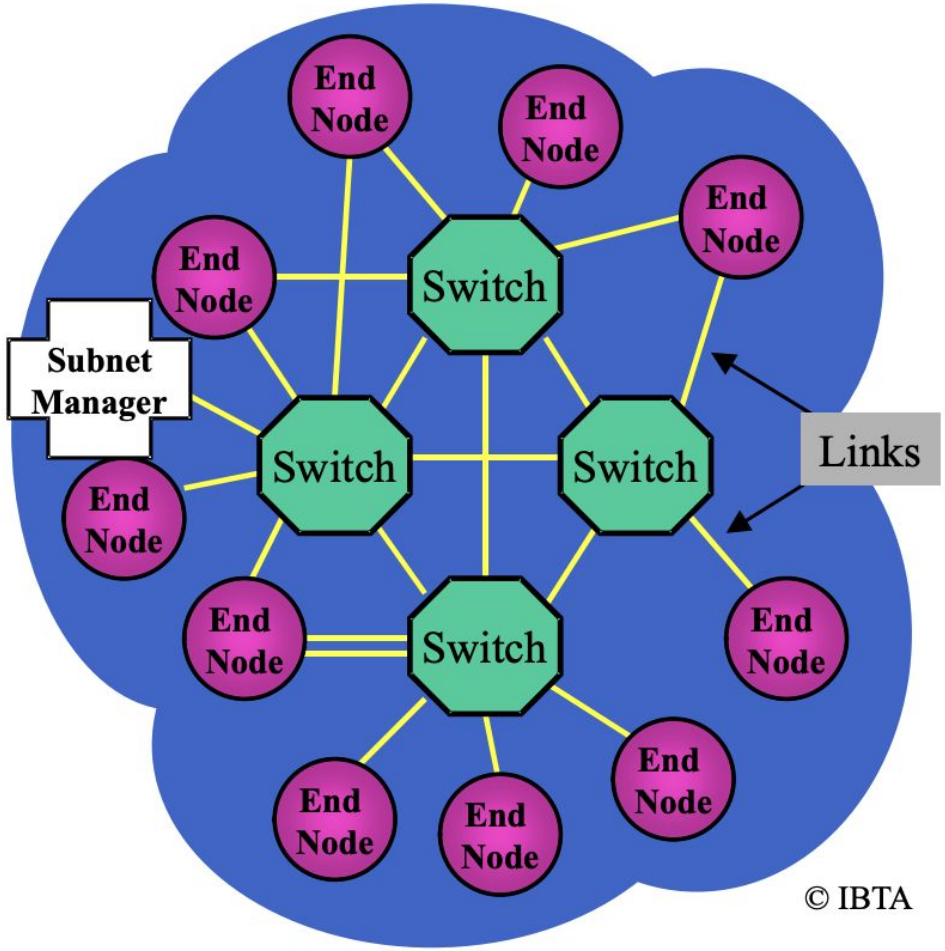
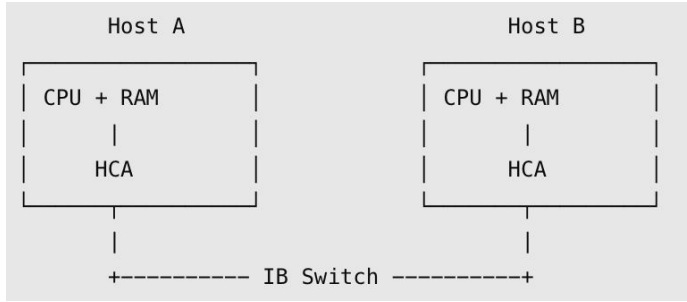


Figure 42.1: An InfiniBand Architecture Subnet



Queues & Work Requests

© IBTA

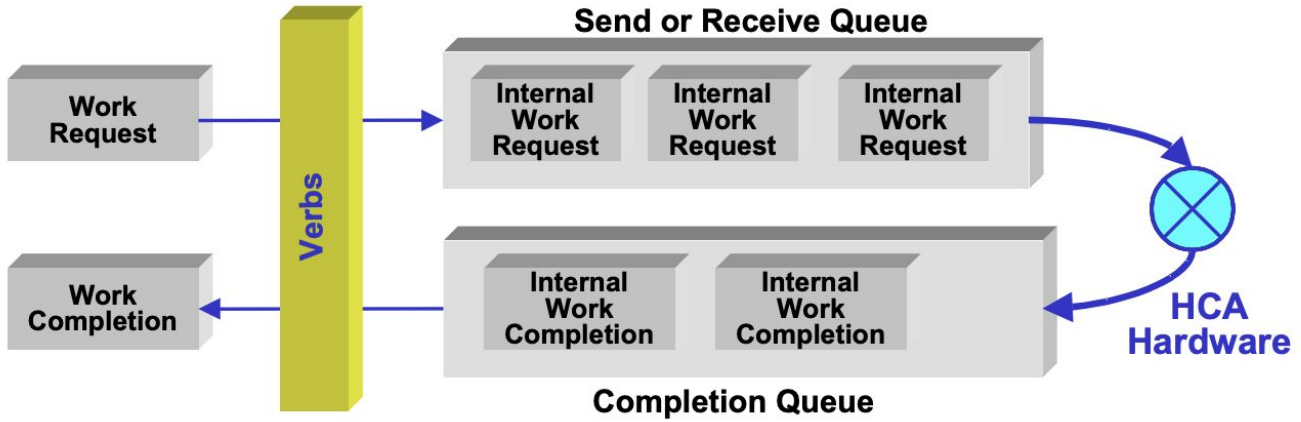


Figure 42.5: Work Request Processing

- When an application which interfaces with InfiniBand starts up, it creates Send and Receive Queue Pairs which are buffers describing work requests
- Receive requests point to place in memory for data to be written
- This allows for asynchronous transfer without resource idling

Subnet Manager State Machine

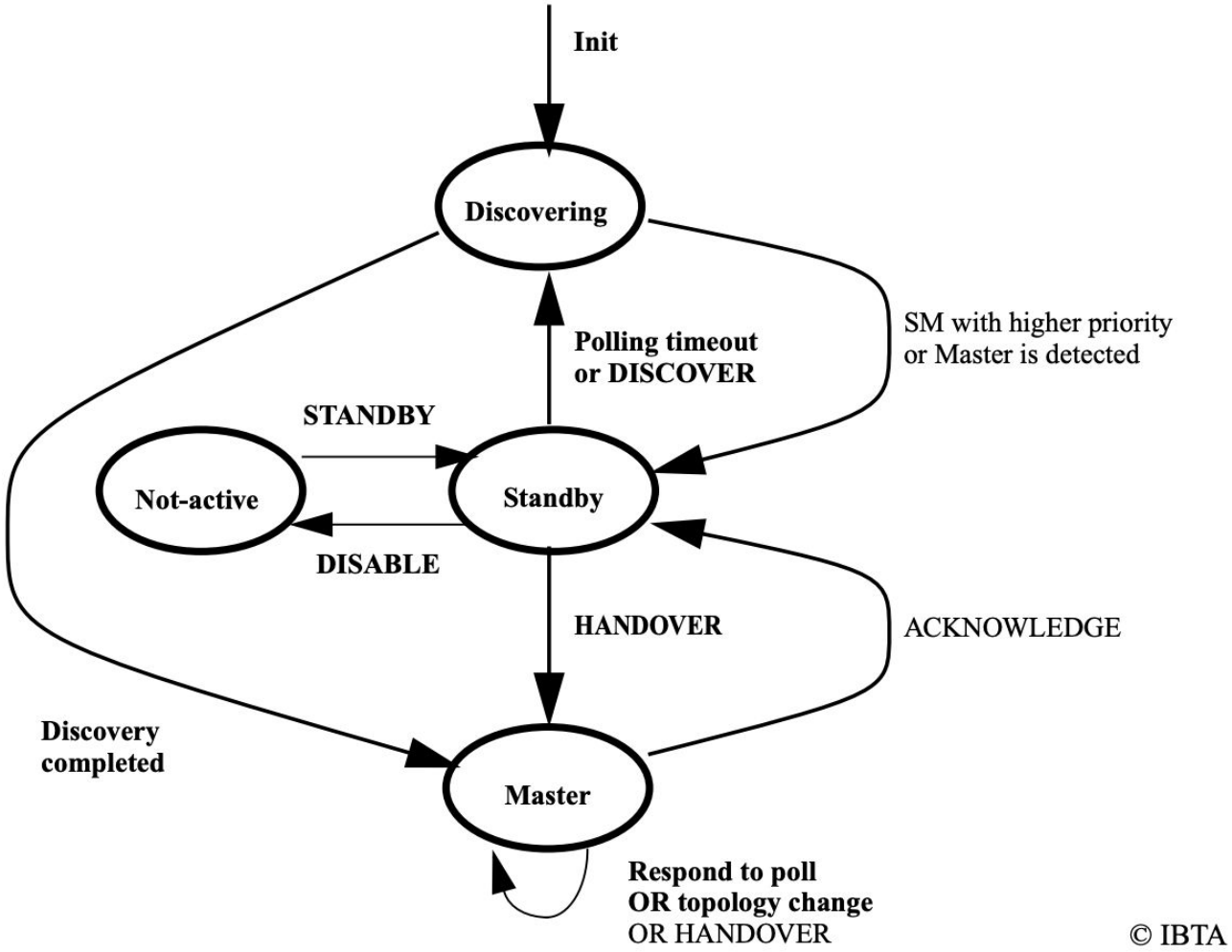
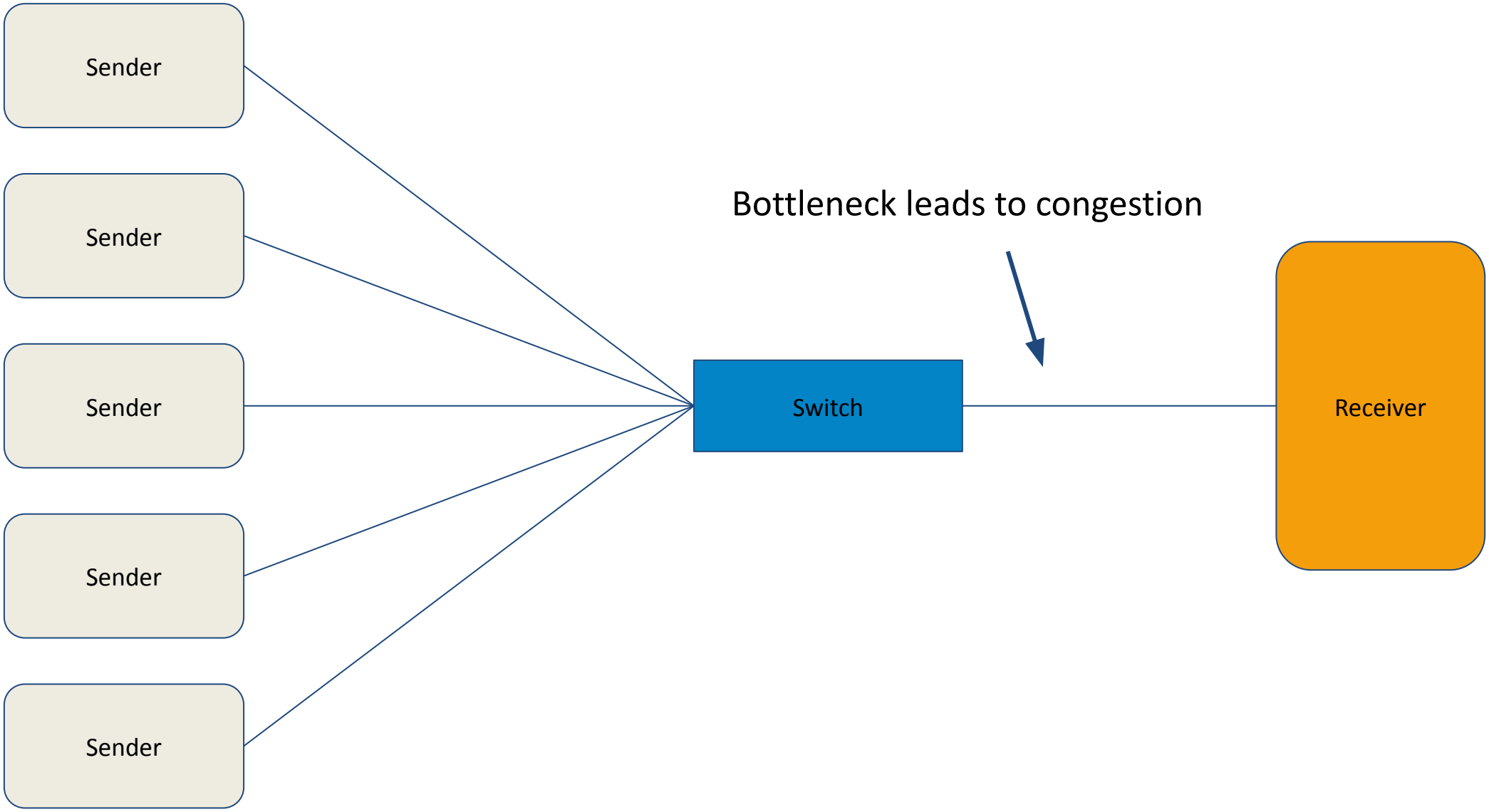
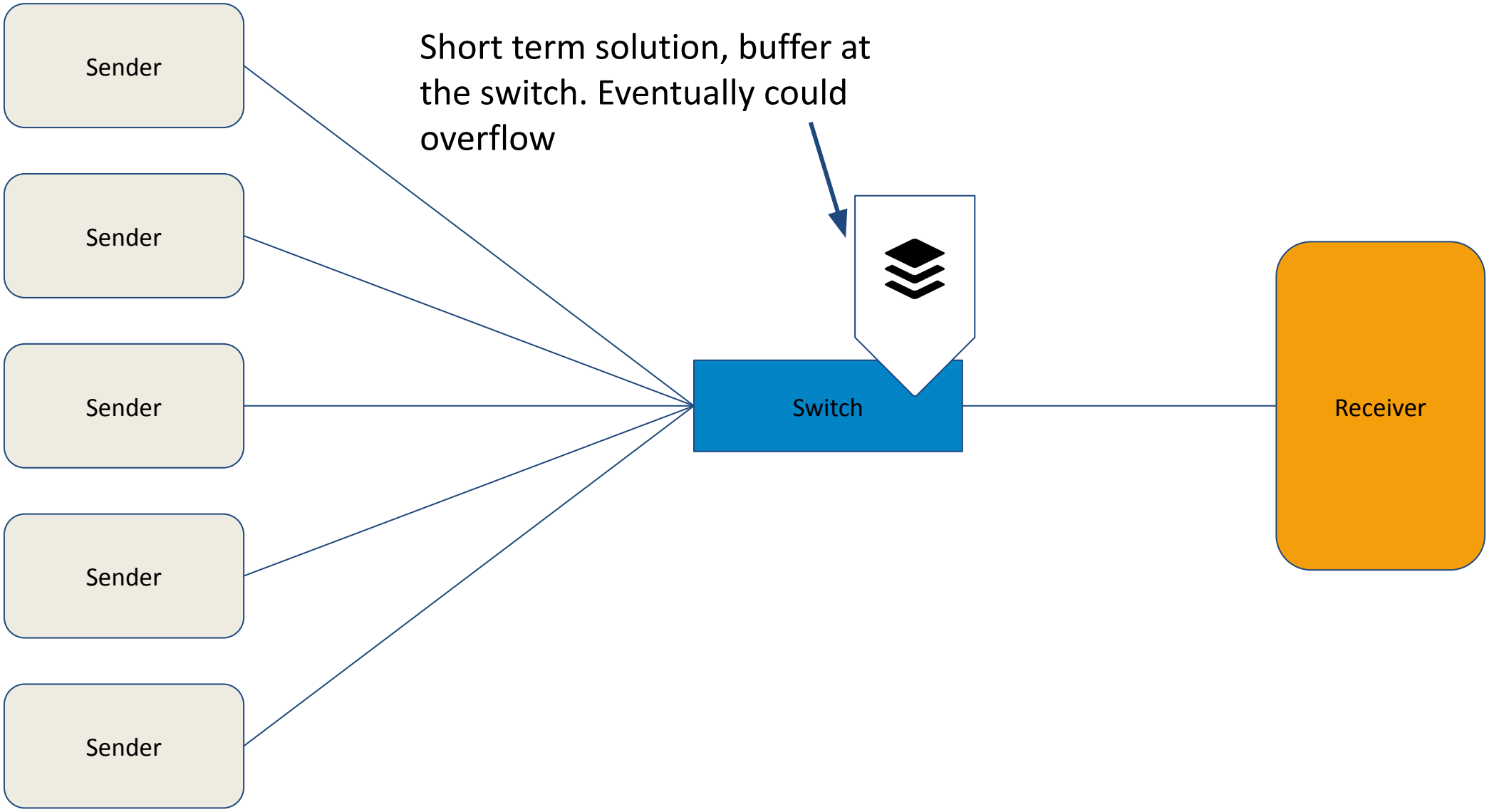


Figure 42.6: Subnet Management Initialization State Machine

Credit-Based Flow Control for ATM Networks



Credit-Based Flow Control for ATM Networks



Credit-Based Flow Control for ATM Networks

- We want to manage network traffic which is **bursty**, making it hard to model or predict
  - Bursts are often the cause of switch buffer exhaustion, leading to retransmissions and potentially further congestion

- Solution is to have receivers allocate *credit* to senders which guarantees good network performance, as long as senders will abide to maintaining nonnegative balance

- Importantly this credit based system will be adaptive to the data being requested to send

Sender Credit Calculation

Credit Balance Remaining =
Allocated Creds - (**Total Units Sent** - **Total Units Forward by Receiver**)

Receiver Allocation Calculation

$$Buf_Alloc_i = \underbrace{(M/2 - TQ - N)}_{\text{Available budget}} \cdot \underbrace{\frac{VU_i}{TU}}_{\text{Connection's share}}$$

M = Buffer Capacity
TQ = Buffer Units Filled
N = Active Circuits

VU_i = Receiver packets forwarded for circuit *i*
TU = Total receiver packets forwarded

Case Study

Our switch buffer has 1000 slot capacity, with 100 units filled and 2 active connections. Both connections are transmitting 50 units, with receiver capturing and transmitting all of them

$$Buf_Alloc_i = \underbrace{(M/2 - TQ - N)}_{\text{Available budget}} \cdot \underbrace{\frac{VU_i}{TU}}_{\text{Connection's share}}$$

M = Buffer Capacity
TQ = Buffer Units Filled
N = Active Circuits

VU_i = Receiver units forwarded for circuit *i*
TU = Total receiver units forwarded

InfiniBand Credit-Based Link-Layer Flow Control

Achieving deterministic, lossless link behavior via receiver commitments and absolute credit accounting.

ARCHITECTURAL COMPARISON

Virtual Lanes

- Virtual Lanes represent separate IB buffer queues to handle different types of data
  - E.g. we want to ensure that small ACKs dont get stuck behind big messages & small ACKs will always have room to be processed
  - And a missing packet on msg 0 doesnt stall msg 1 if msg 1 is on a separate lane

CORE ACCOUNTING STATE

The Absolute Credit Algorithm

ABR Counter

Adjusted Blocks Received

Receiver counts total 64-byte blocks received since link initialization.

FCCL Limit

Flow Control Credit Limit

Receiver grants absolute limit calculated as ABR plus current free buffer blocks.

FCTBS Counter

Total Blocks Sent

Transmitter sends data only when FCTBS plus packet size stays within FCCL.

- ABR can fall behind so its periodically updated to match FCTBS

CORE ACCOUNTING STATE

Resiliency

Loss of flow control updates

- FCCL is total open slots ever seen so it monotonically increases and guarantees packets arent sent until there is buffer space

Loss of packets

- Use of ACKs from receiver to sender about highest packet ID received consecutively

META ENGINEERING • ACM SIGCOMM 2024

RDMA over Ethernet for Distributed AI Training at Meta Scale

Architecting, operating, and co-designing lossy and lossless RoCEv2 fabrics across tens of thousands of GPUs.

TRAFFIC PATTERNS

AI Training



Why ROCEv2?

- You need inter-node connects for LLM training
- Infiniband (and other interconnects) are not commodity
  - Depends on NVIDIA to set up
- So let's use ethernet!
  - Works well with existing datacenters @ start of LLM boom



CONTEXT & SCOPE

RDMA Idea

Let’s bypass traditional NIC sockets (that write to CPU buffers) and instead let NIC read directly from GPU HBM and write to another GPU’s HBM.



PHYSICAL ARCHITECTURE

Hardware

Fat Tree-like!

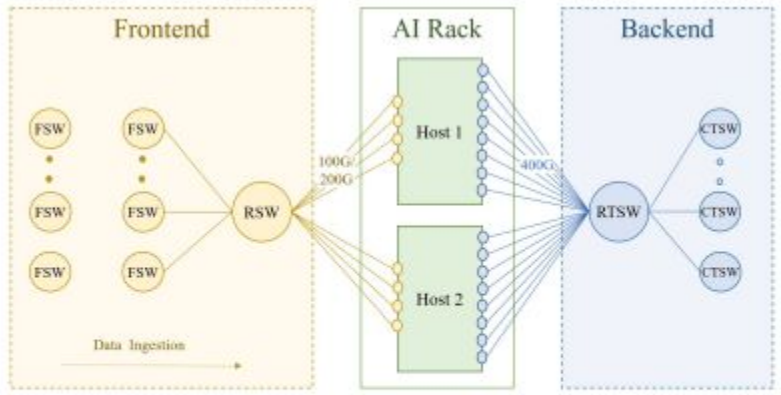


Figure 5: Frontend and Backend networks

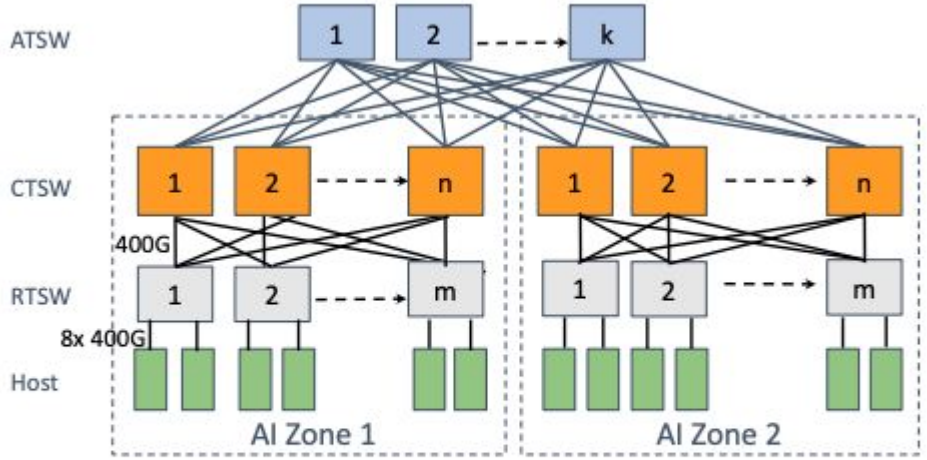


Figure 6: Backend Network Topology

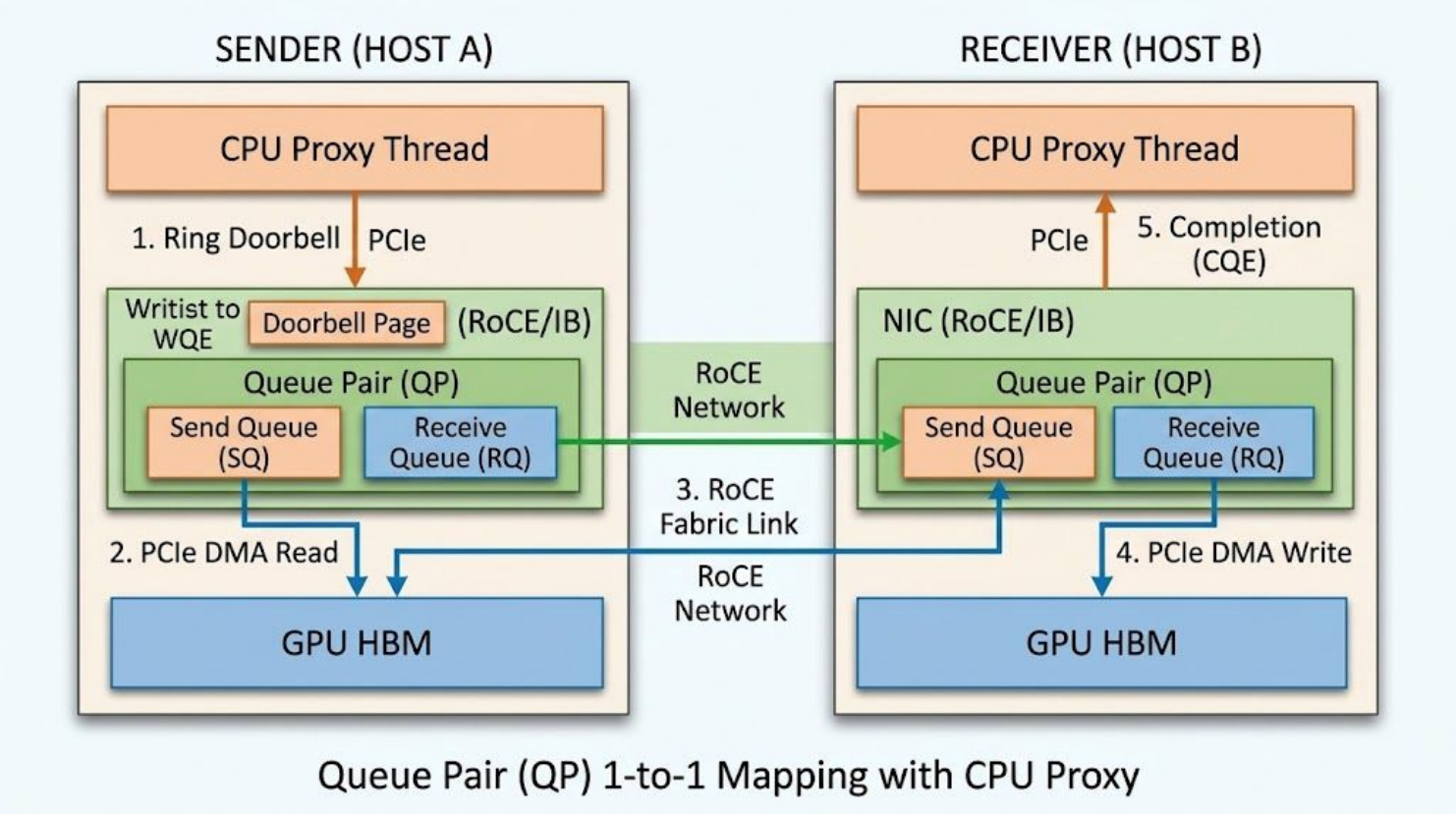
Frontend ingests data to CPUs!

Backend is underlying fabric providing faster connections for GPU-GPU communication

Fat Tree-like!

Queue Pairs

- Bypass normal sockets because do not want to write to CPU, then have kernel copy to HBM
- Queue Pairs instead!
  - Send buffer
  - Receive buffer



Queue Pair (QP) 1-to-1 Mapping with CPU Proxy

ROUTING ANALYSIS

Why Standard ECMP Fails for AI

Standard Web Traffic

High Flow Count: Millions of concurrent small connections.

Uniform Hash Spread: Statistical 5-tuple entropy balances links evenly.

Low Collisions: Microbursts absorb cleanly without persistent packet drops.

AI Distributed Training

Sparse Flows: Handful of 400Gbps monolithic elephant streams.

Hash Collisions: Multiple heavy streams hash to the identical path.

Severe Degradation: Saturated bottleneck links while adjacent links idle.

1. Five Tuple: Source IP, Dest IP, Source Port, Dest Port, Protocol all shared for different QP packets => also hash by QP packet as well

2. And if we are hashing w/QP number, then we want to equally spread among the QPs -> round robin QP assignment for messages

LOAD BALANCING SOLUTIONS

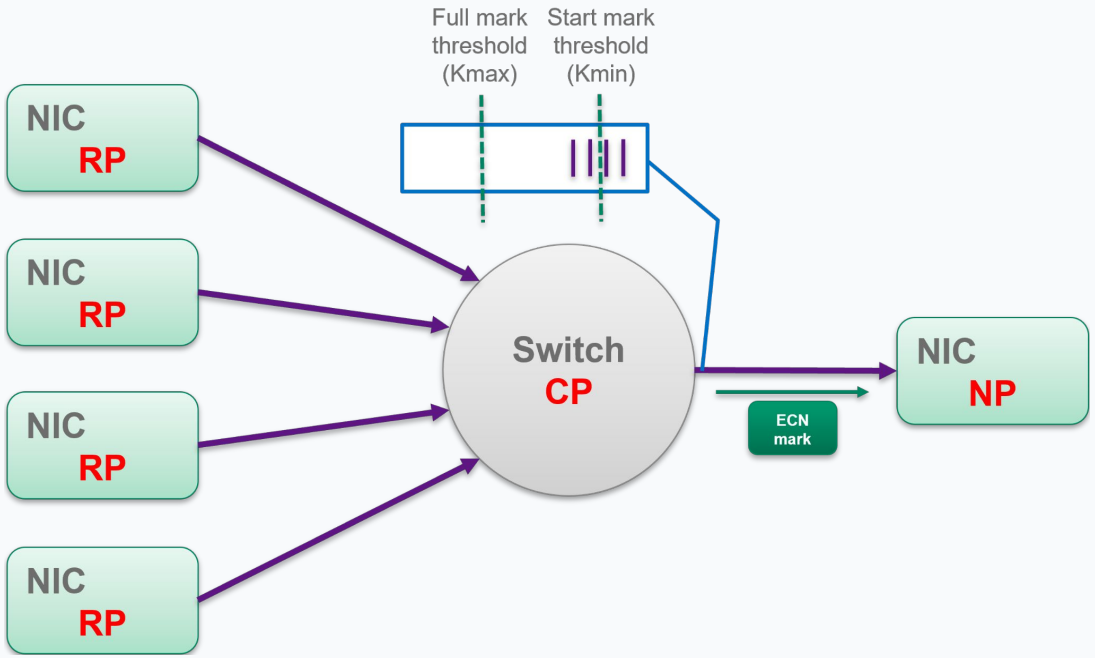
Traffic Controller

- Collector: Real time topology of the training cluster
- Flow Matrix: Actual bits per second across each connection
- Switch Programmer: Translates into actual routing decisions on network switches

Use of Constrained Shortest Path First to identify best path between A and B in the flow matrix!



Congestion Control & Flow Discipline



Normal DCQCN:

- When the CP (congestion point/switch) sees buffer filling up, it marks packets with ECN which basically tells the destination to inform the source to slow down
- Then, PFC is a threshold when the switch issues a STOP command on the previous switch becomes the switch is now a nearly full buffer

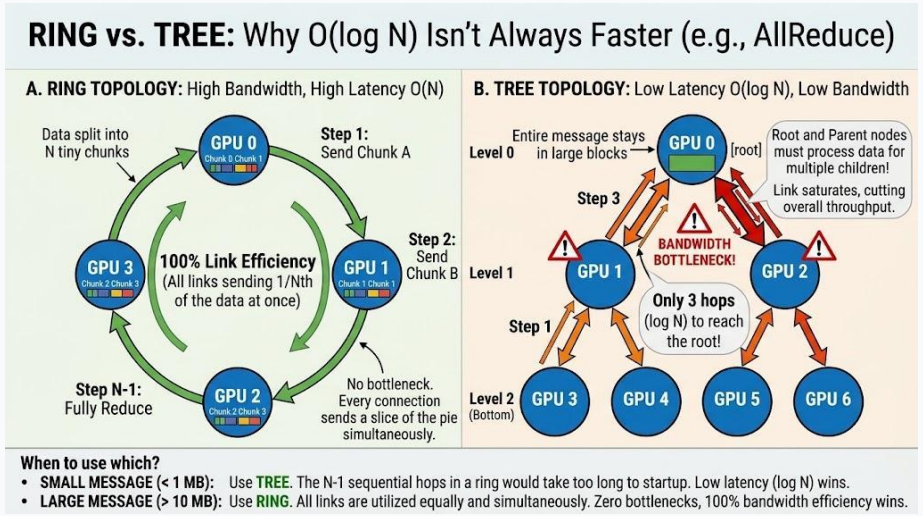
Dont want to freeze! Or else preceding switch also begins to accumulate and freeze!

But they just do PFC :), ECN threshold too hard to tune and too buggy at the time of the paper ⇒ report that it worked well still

Co-Tuning

1. 22 microseconds for RTT
   - receiver sends CTS (clear to send) and then sender sends packet
   - packet-wise ACKs

   So lets make bigger message sizes, so you can overlap sending for rec buffer 1 when the destination is emptying rec buffer 0
2. Prioritize ACKs
3. Use tree instead of ring topology because rings have O(N) sends while tree is O(logN) to circulate among GPUs in a collective.



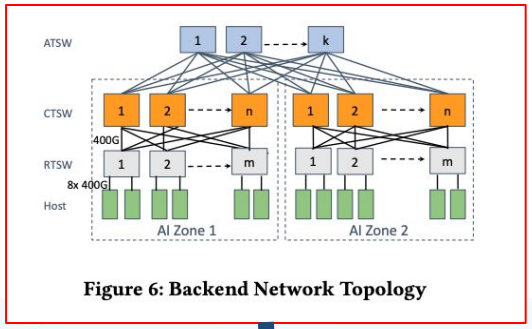
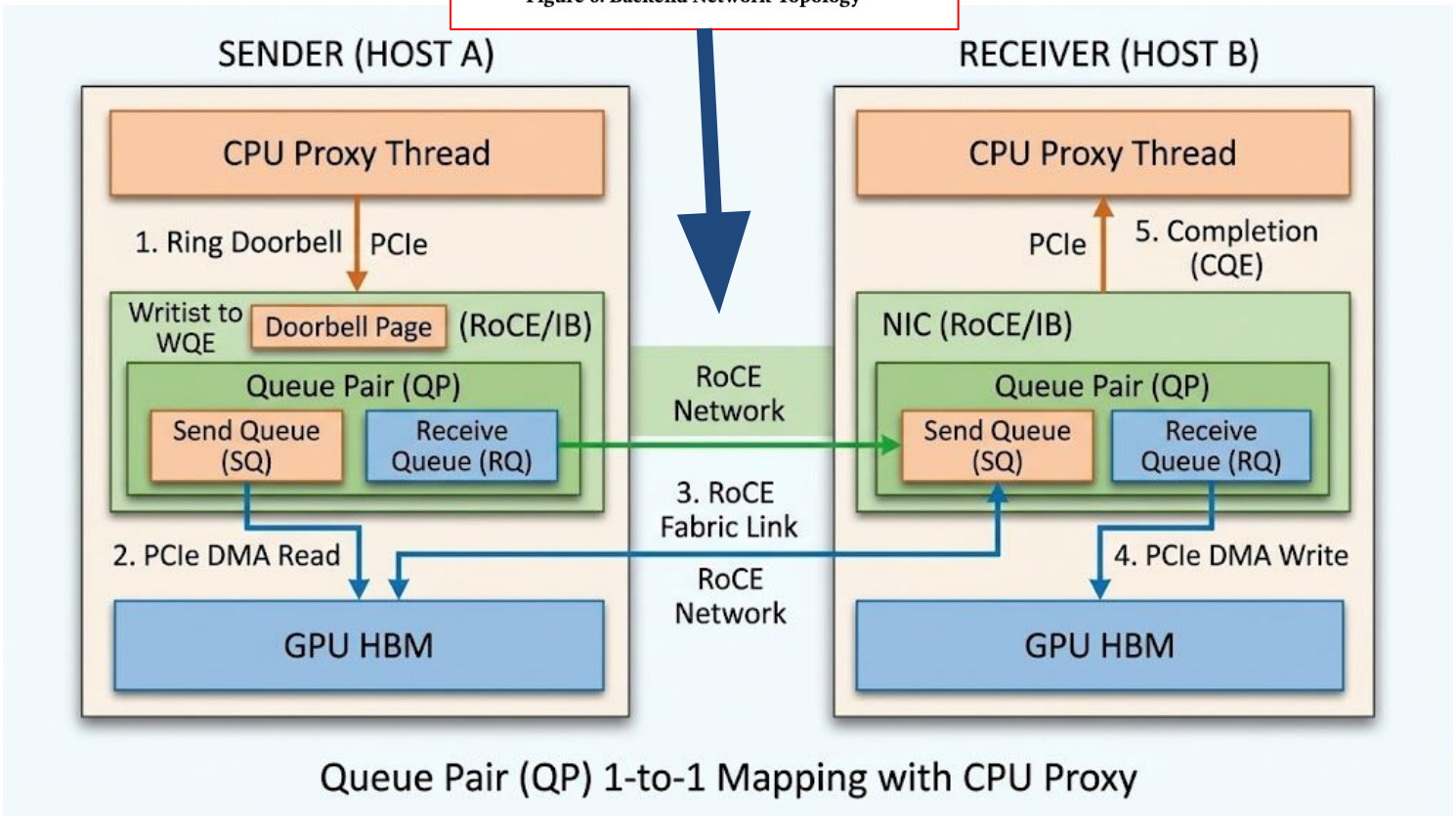


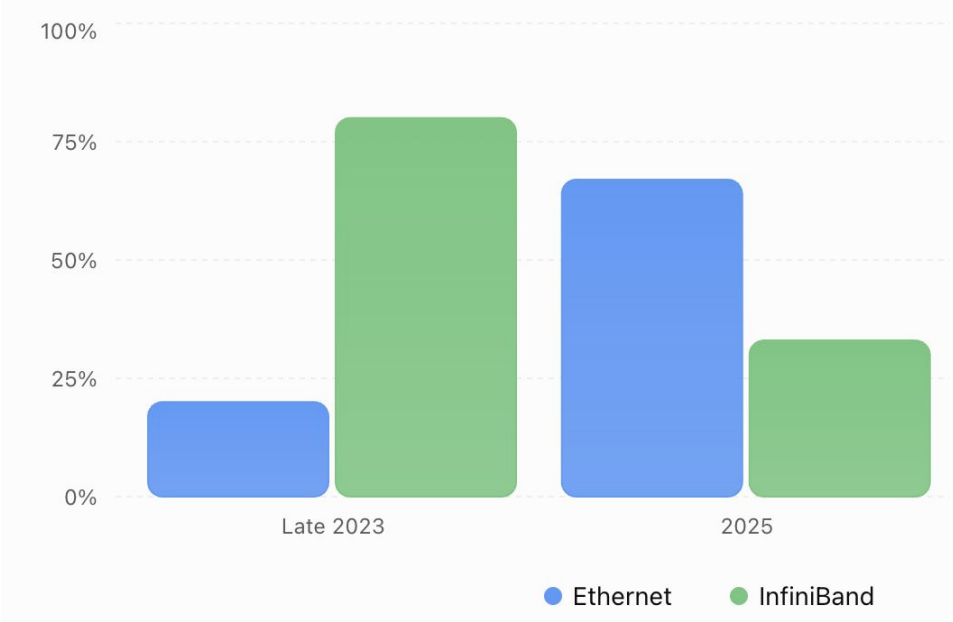
Figure 6: Backend Network Topology



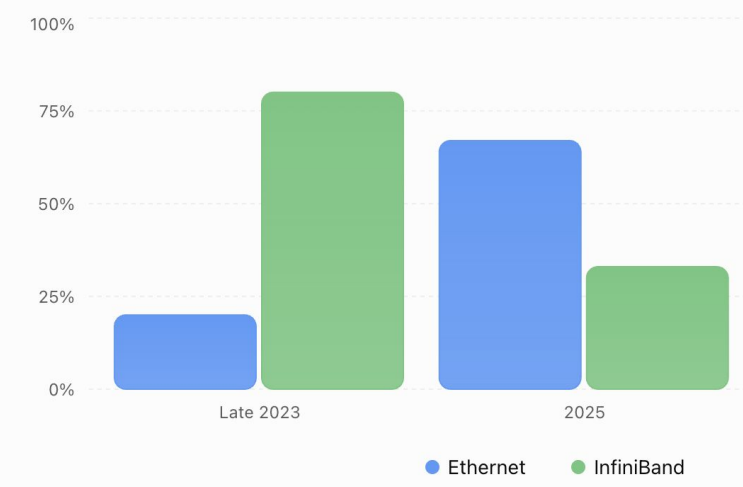
Beyond the Papers: Landscape of ML Networking

- There is not yet a clear winner between InfiniBand vs Ethernet for ML networking

- Based on independent party market research, Ethernet is coming back… but why?

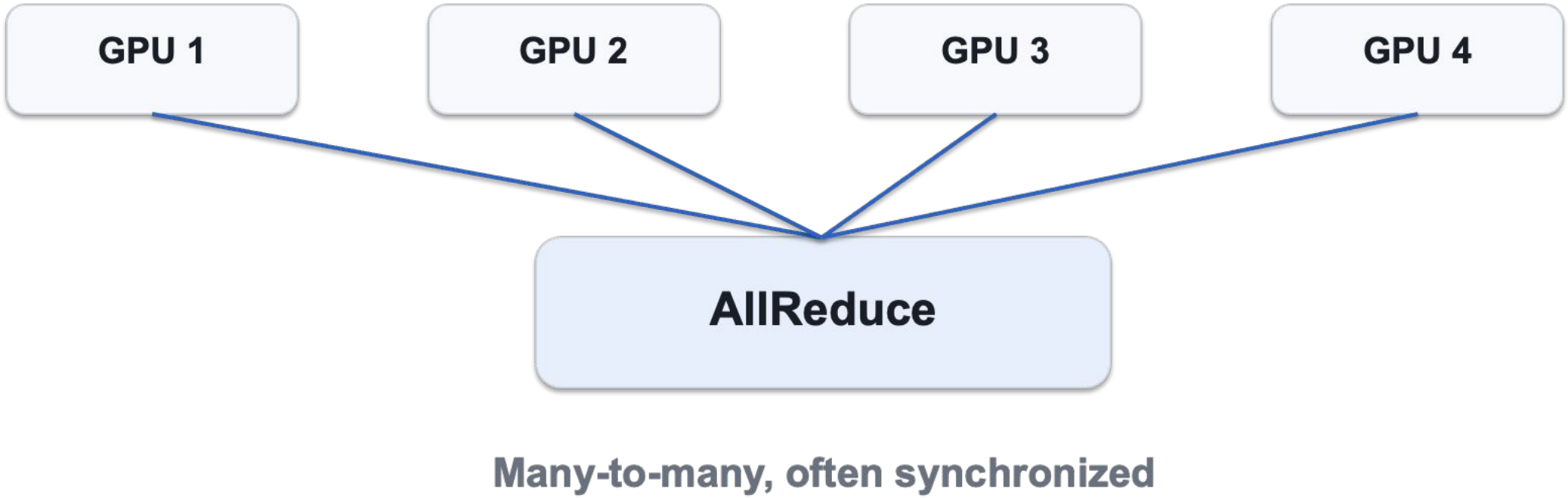


Beyond the Papers: Why the change?



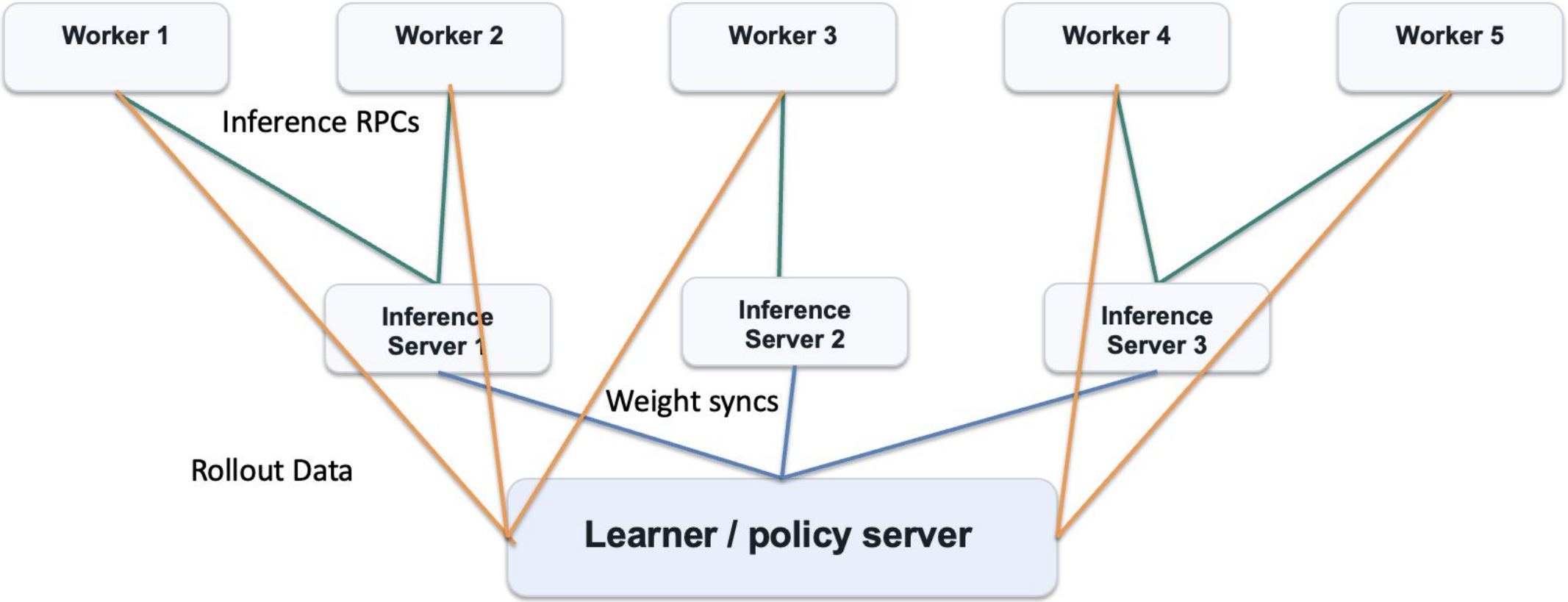
- NVIDIA is the dominant supplier of Infiniband
- InfiniBand requires special silicon engineering and cost which many vendors aren’t willing to do
- Ethernet is somewhat commoditized with a larger set of vendors

Beyond the Papers: Let's design a cluster for RL-focused efforts



What Meta more-or-less optimized

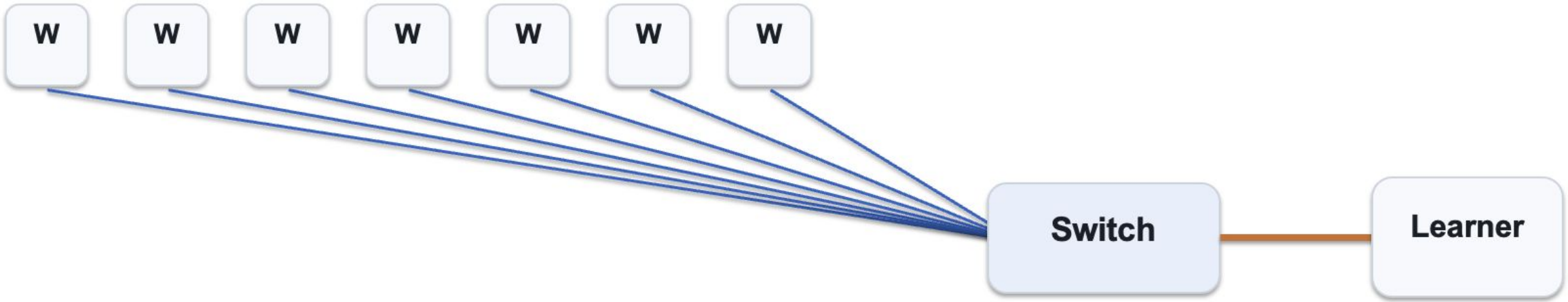
Beyond the Papers: Let's design a cluster for RL-focused efforts



Unlike AllReduce, messages aren't synchronized. Different types of data also looks very different. Workers finish at different times

Beyond the Papers: Let's design a cluster for RL-focused efforts

All workers finish a rollout and upload at once



100 × 100 MB = 10 GB

400 Gb/s link → ≥ 200 ms serialization

What should happen if the learner cannot ingest the burst?

Beyond the Papers: Let's design a cluster for RL-focused efforts

- What to prioritize for a cluster designed for RL?
  - Low latency inference?
  - High throughput trajectory uploads?
  - Fast and frequent weight sync?

- Do we need high reliability and lossless communication? Maybe only for some aspects?
  - Stale data is less useful in current standard RL algs!

- How to do congestion control with varying priorities: policy updates vs trajectory uploads vs inference

Beyond the Papers: How should we design our clusters to be future-proof?

- We just designed a cluster around a well-established RL algorithm, but what happens when new research proves other algorithms to work well?
  - For example, off-policy RL has different requirements than on-policy algorithms like PPO

- Should we optimize our clusters to adaptable to research rather than overfit to a paradigm? Bringup/Bringdown cost of these projects is incredible

Discussion: Ethan Li Shen @ gmail. com

InfiniBand relies on hardware-level credit flow control to prevent packet loss entirely, while RoCEv2 relies on Ethernet mechanisms like Priority Flow Control (PFC) and Explicit Congestion Notification (ECN). Which approach is easier to operate at scale?

When a link or switch degrades in a cluster running a massive distributed job, which fabric is easier to monitor, debug, and isolate without restarting the entire workload?

Is it worth investing in improved ethernet capabilities or is infiniband reliance an acceptable cost?