# Hostmesh: Monitor and Diagnose Networks in Rail-optimized RoCE Clusters

**Kefei Liu, Jiao Zhang, Zhuo Jiang, Xuan Zhang, Shixian Guo, Yangyang Bai, Yongbin Dong, Zhang Zhang, Xiang Shi, Lei Wang, Haoran Wei, Zicheng Wang, Yongchen Pan, Tian Pan, and Tao Huang**

# Background

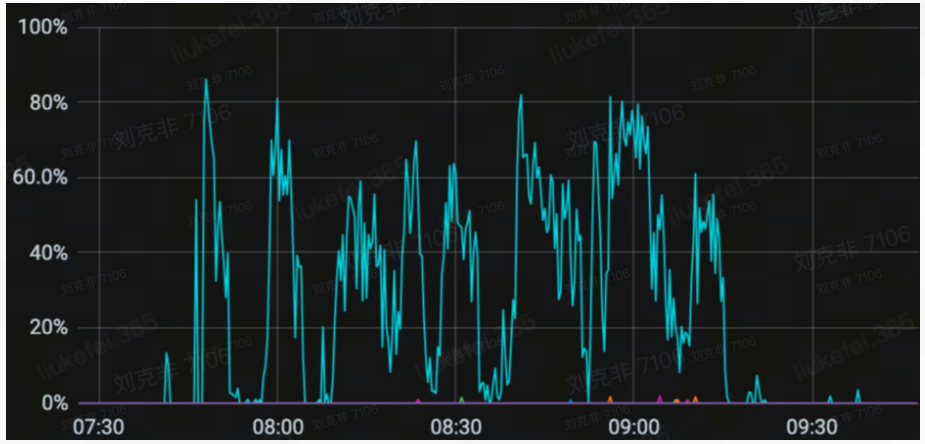
## ■ Characteristics of RoCE Network Problems

- **RDMA throughput is sensitive to packet drops**
  - **Commodity RDMA NICs (RNICs) use go-back-n as the retransmission mechanism.**
- **Frequent network misconfigurations**
  - **RoCE (RDMA over Converged Ethernet) networks typically require a variety of configurations, such as enabling PFC and configuring PFC headroom, leading to a high probability of misconfigurations, which can cause packet drops and throughput degradation.**
- **Frequent RNIC problems**
  - **Such as RNIC flapping, which can lead to packet drops and throughput degradation.**

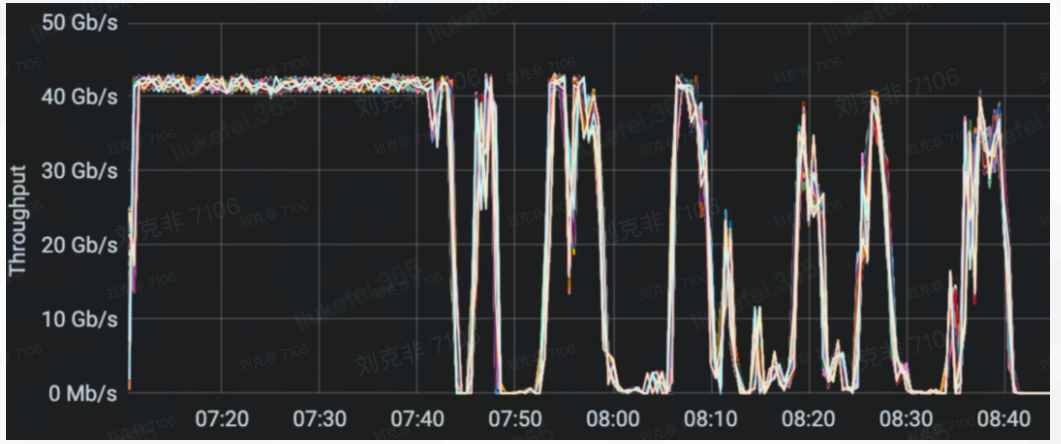
# Background

## ■ Characteristics of RoCE Network Problems

- **Distributed Machine Learning (DML) has a barrel effect.**
  - **In DML, all participating GPUs periodically synchronize their local gradients over the network. The completion time of this process is determined by the slowest GPU.**
  - **Either RNIC or in-network packet drops can cause throughput degradation of multiple flows, further degrading the average throughput of the *ENTIRE* DML cluster.**




One RNIC's drop rate

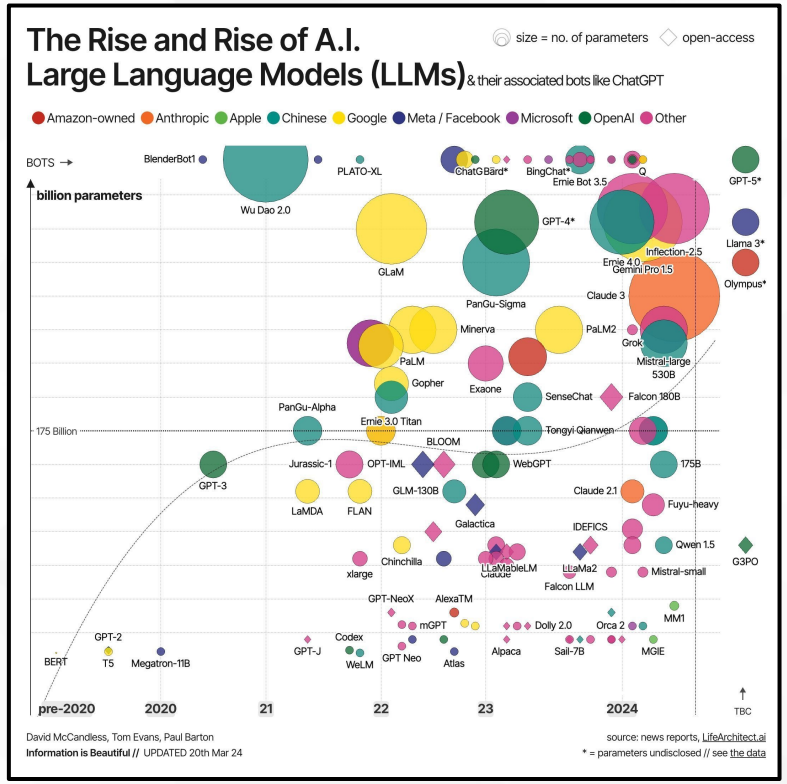


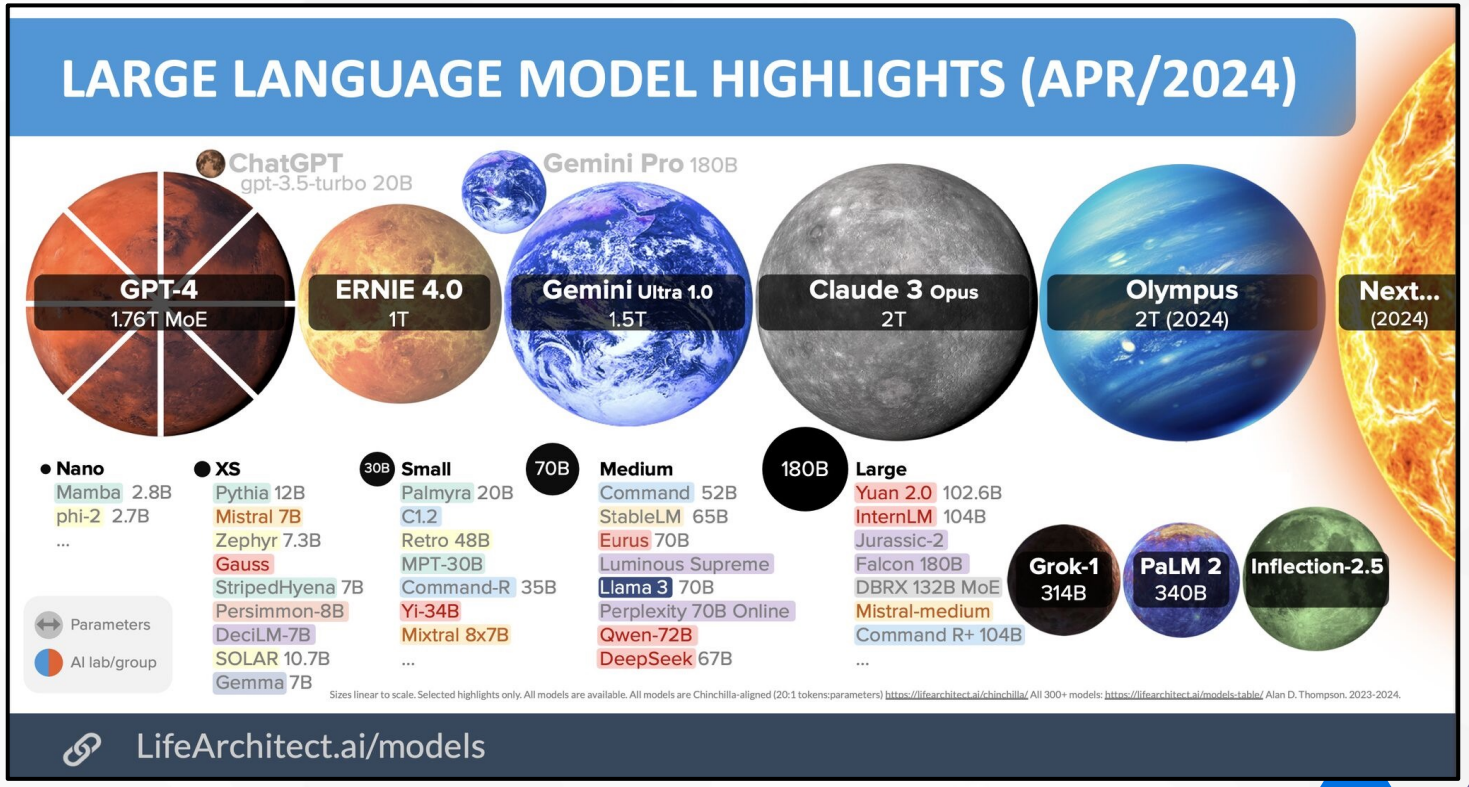

Average throughput of the DML cluster

# Background

## ■ RoCE Clusters are Growing in Size

- **The size of large language models (LLMs) is increasing, and typical LLM model parameters have now reached 100B or even >1000B.**

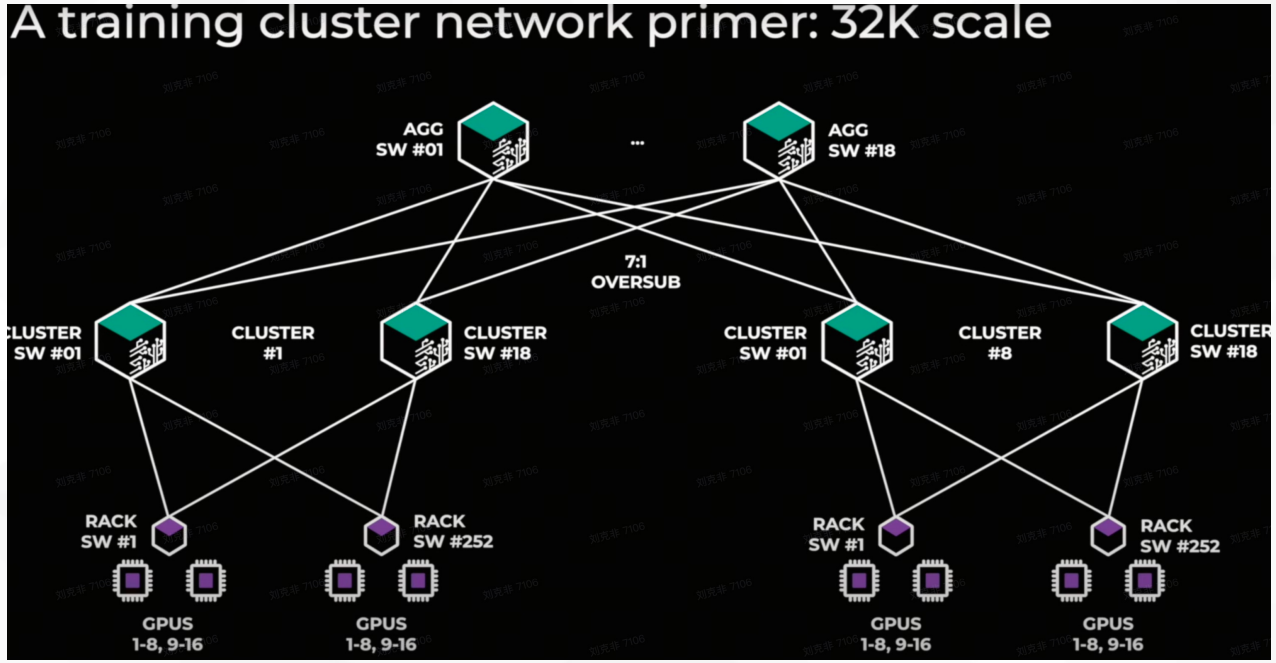


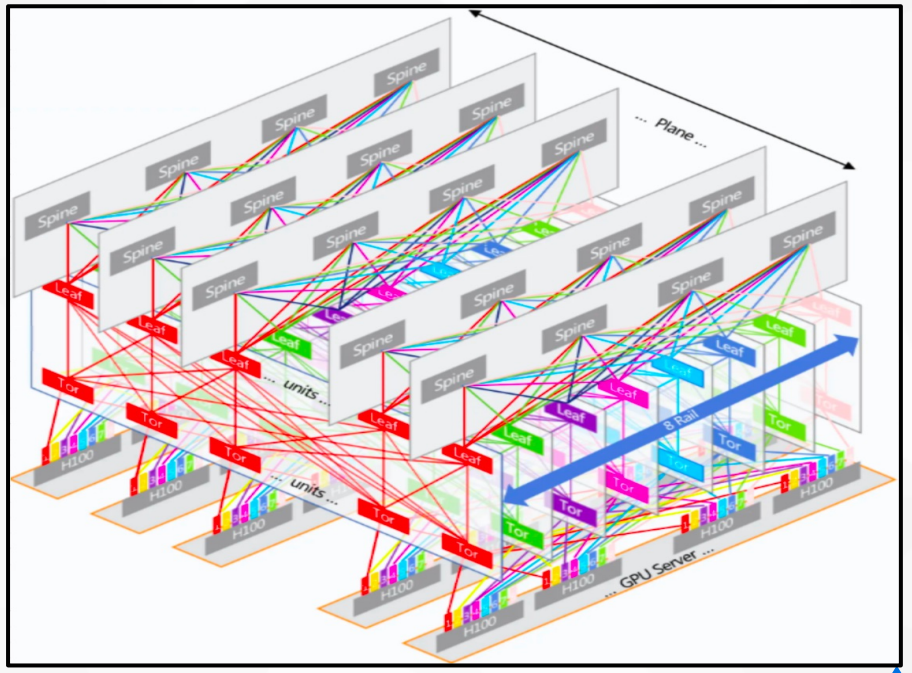
# Background

## ■ RoCE Clusters are Growing in Size

- To increase the training rate, the DML cluster uses more GPUs for training as well as more RNICs and switches to connect these GPUs.
- As a result, both the frequency and impact of network problems are increasing.




LLM training cluster in Meta (32K GPUs and RNICs)

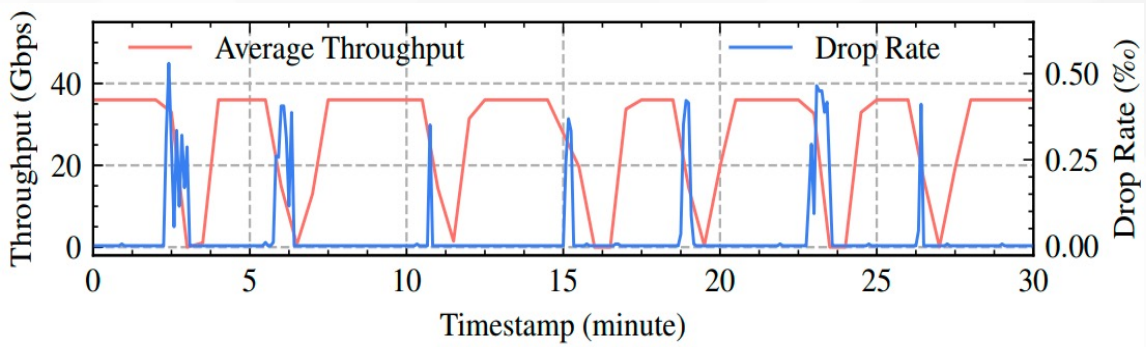


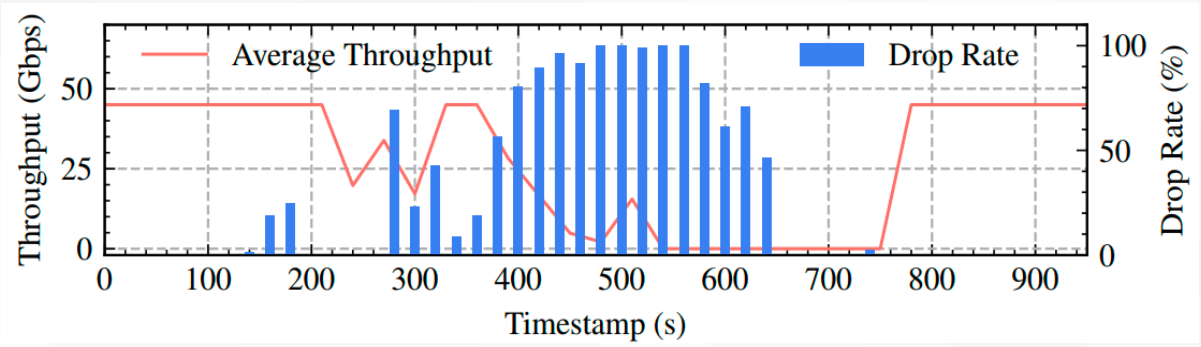

A rail-optimized cluster for 32K GPUs

# Background

## ■ Efficient Troubleshooting is Critical for Optimal Training Performance

- When training **fails** or training **performance degrades**, the problem needs to be quickly located to **restart the training task** or **restore optimal training performance**.
- Large clusters that are **idle for troubleshooting** or running in **a suboptimal state** can result in **significant financial losses**.



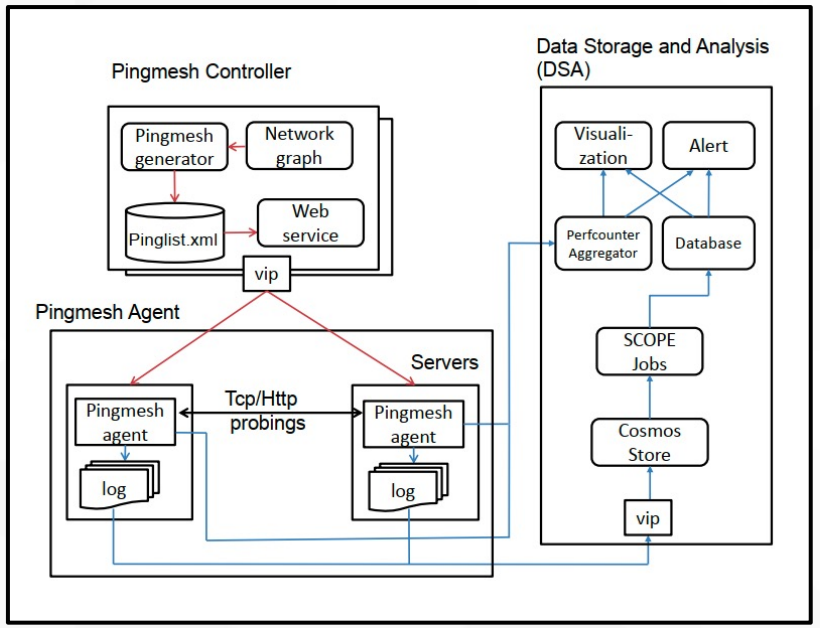




One **single** flapping switch link (left) or RNIC (right) results in **severe cluster throughput degradation**

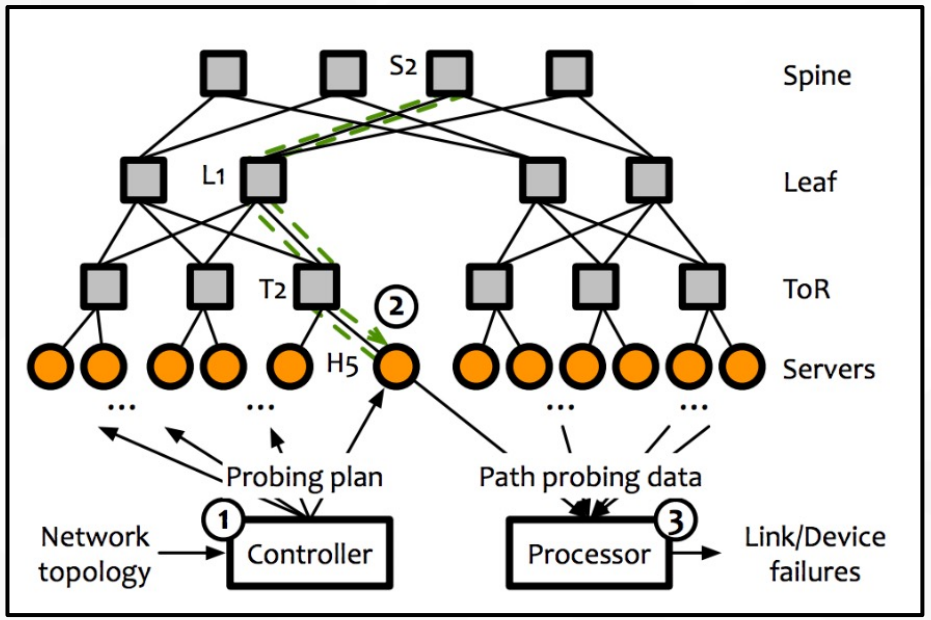
# Motivation

## ■ Need an Efficient Network Monitoring and Diagnostic System

- Detect and locate RoCE network **failures** and **performance bottlenecks**.
- The system should be based on **active probing**.
  - Ensure that network problems can be detected **independent of service traffic**.




Pingmesh Framework

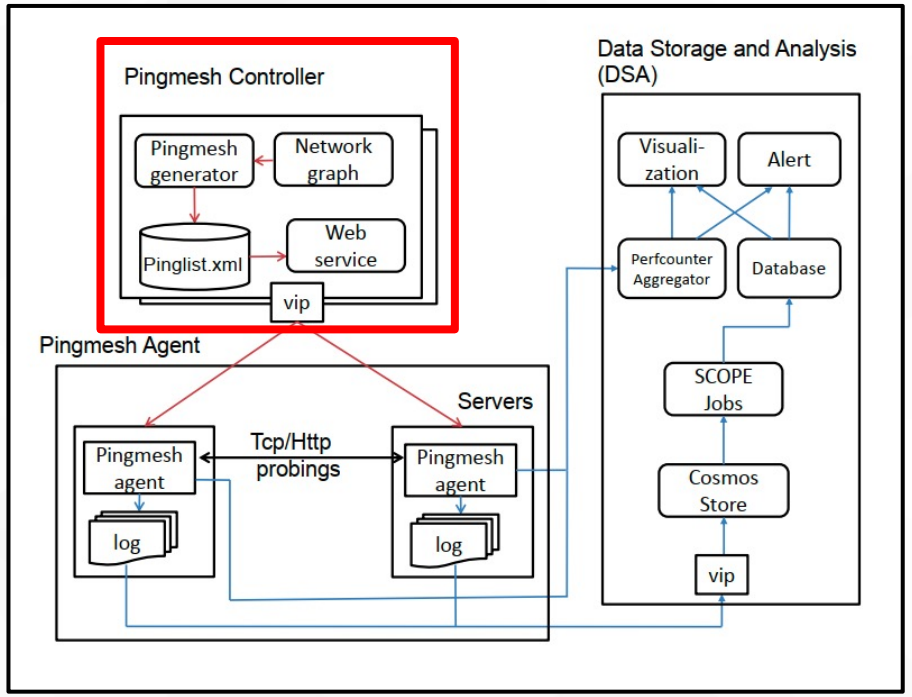


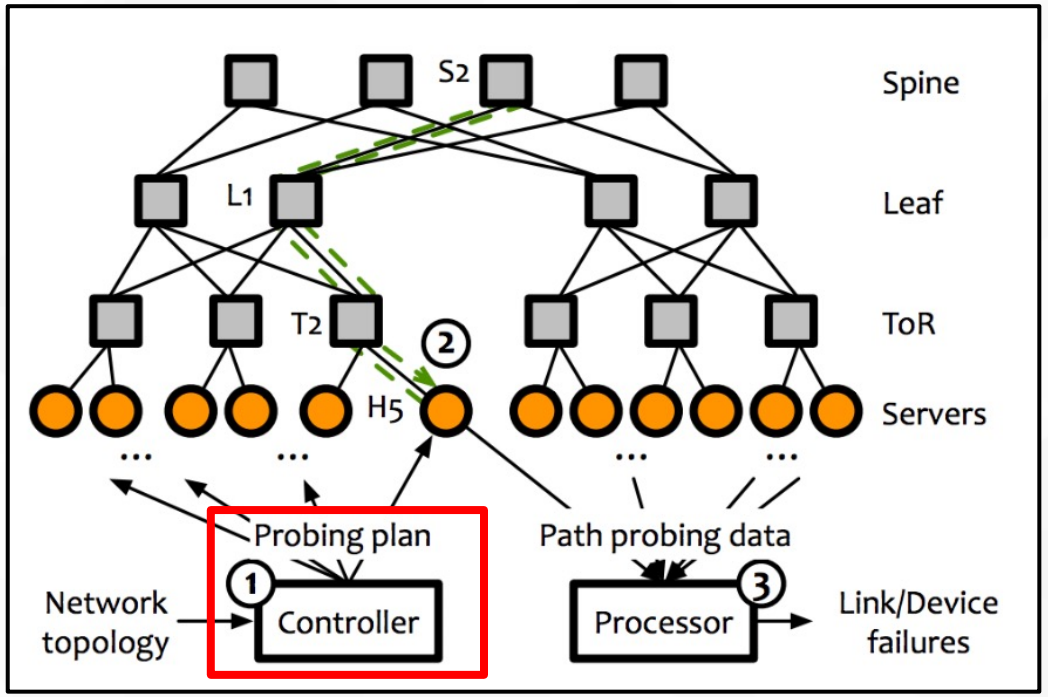

Netbouncer Framework

# Motivation

## ■ Limitations of Existing Active Probing Mechanisms

- Rely on a **centralized controller** to generate pinglists (probing plan).
- The need for a **controller** has limitations in **multi-tenant clusters**.




**Pingmesh Framework**

**Netbouncer Framework**

# Motivation

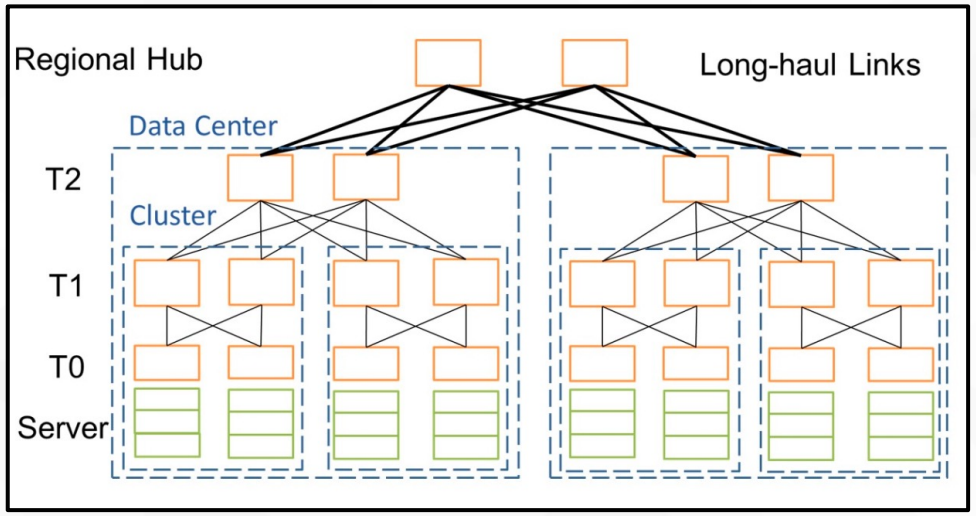
## ■ Limitations of Existing Active Probing Mechanisms

- Deploying a controller in multi-tenant clusters is difficult.
  - A centralized controller may not be accepted due to security concerns.
  - A server within each tenant cluster as the controller consumes additional tenant resources and is difficult to ensure stable operation of the controller.
- Controllers in multi-tenant clusters have high synchronization overhead.
  - Controllers must maintain the latest server information of each tenant cluster and frequently update the pinglist for each server to avoid probing noise.

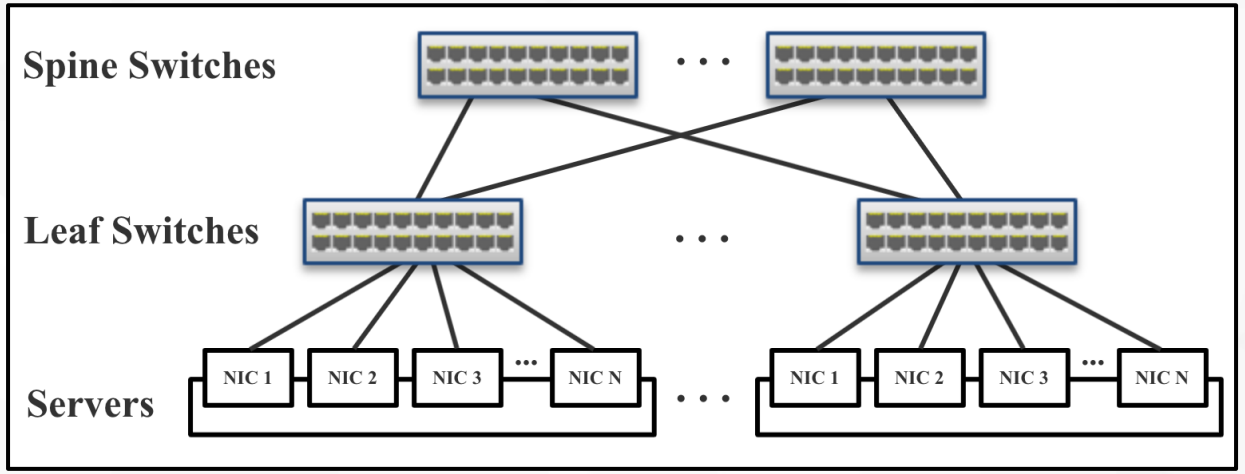
# Motivation

## ■ Our System Design Targets

- An active probing system for RoCE clusters without relying on a controller.
- However, in some topologies, inter-server probing is necessary, and a controller is essential to provide the communication address of the target servers.




CLOS topology in Azure

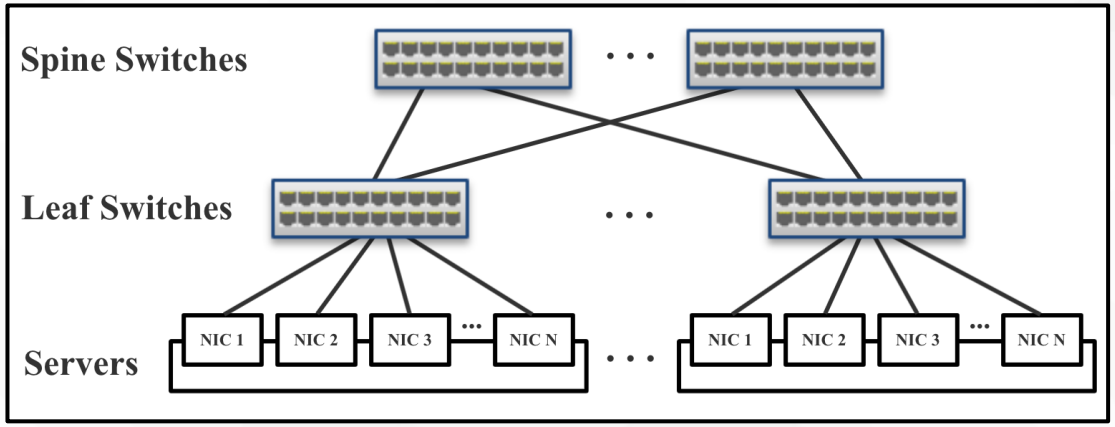



CLOS topology for hosts with mulitple NICs

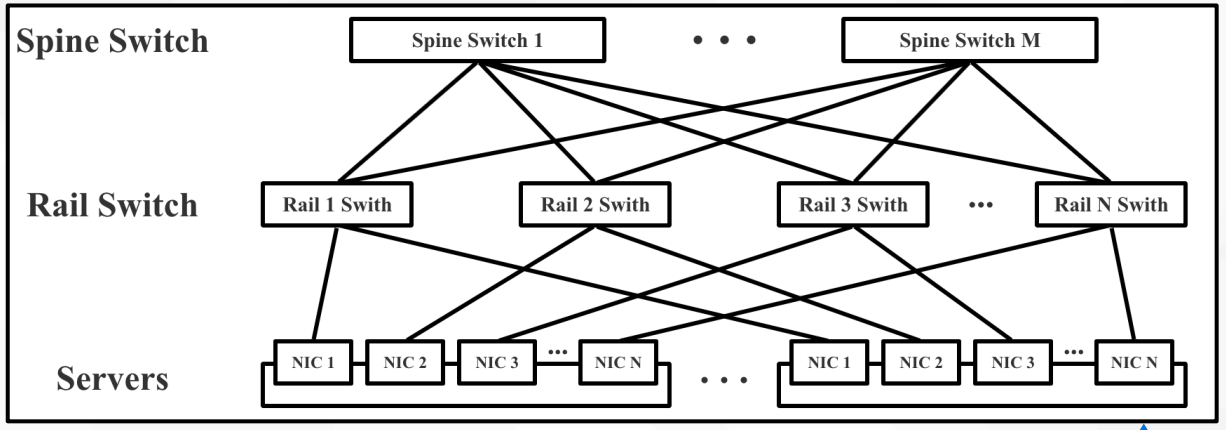
# Motivation

## ■ Opportunity: Rail-optimized Network

- For multi-NIC hosts, the rail-optimized topology improves network performance.
  - In Clos networks, all NICs on a host are connected to the same ToR switch.
  - In rail-optimized networks, NICs on a host connect to different ToR switches (rail switches).
- Rail-optimized topology allows more hosts to be under the same ToR switch.




CLOS topology for hosts with mulitple NICs

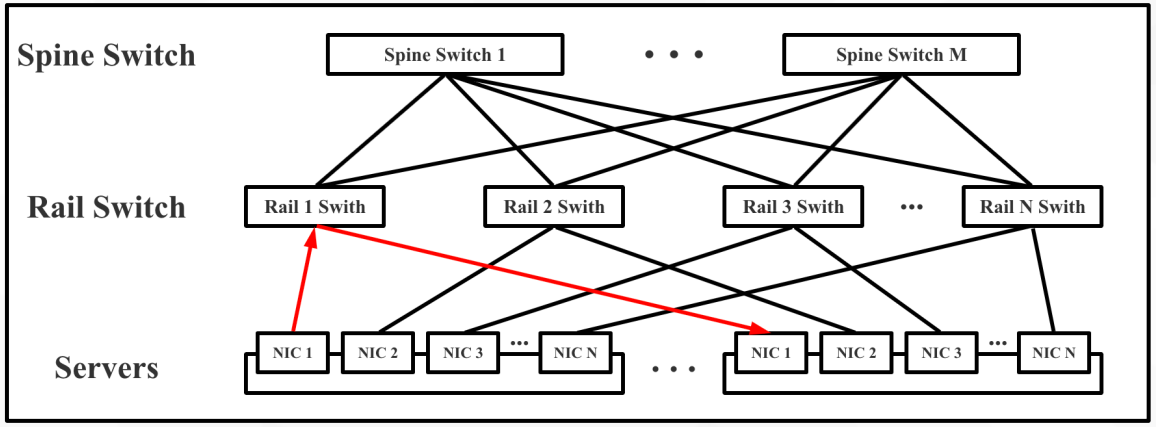



Rail-optimized topology for hosts with mulitple NICs

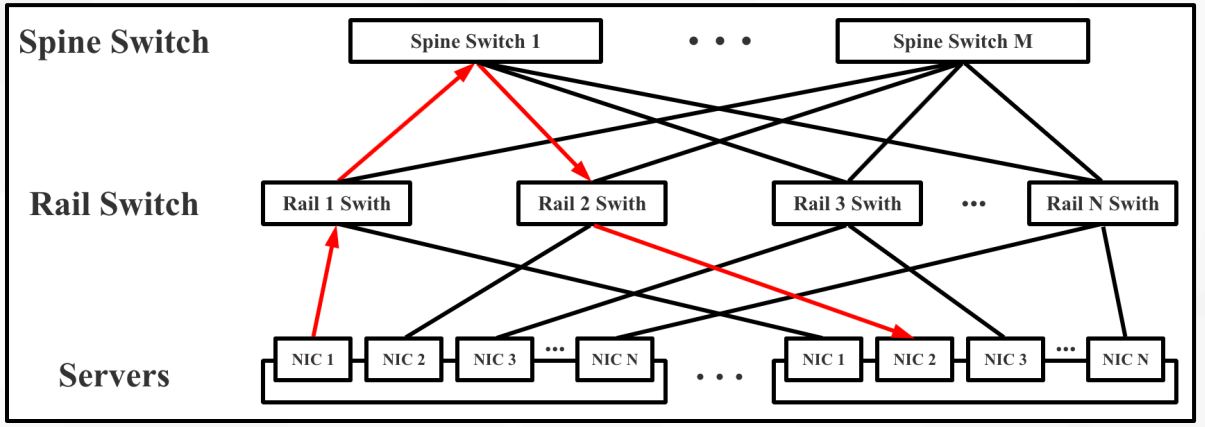
# Motivation

## ■ Opportunity: Rail-optimized Network

- Traffic paths in rail-optimized clusters
  - Intra-rail communication is **below the rail switch**.
  - Inter-rail communication must **pass through the top-tier switches** in the cluster.




Intra-rail traffic



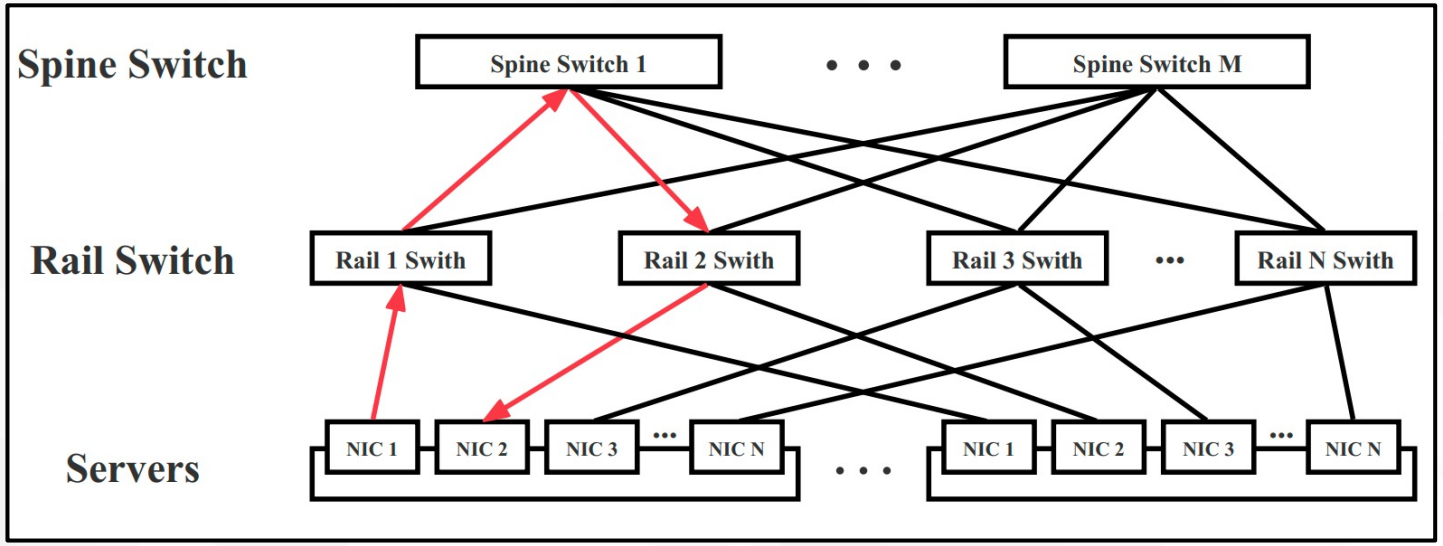

Inter-rail traffic

# Motivation

## ■ Opportunity: Rail-optimized Network

- **Key insights:** Each host has **multiple NICs**, and traffic between different NICs on the same host must **go through top-tier switches** in the cluster.
- We can use **different NICs on each host to probe each other**, and keep changing the probe's 5-tuple. Then these probe packets can cover all the links in the cluster.

**Need No Controller!**



# Motivation

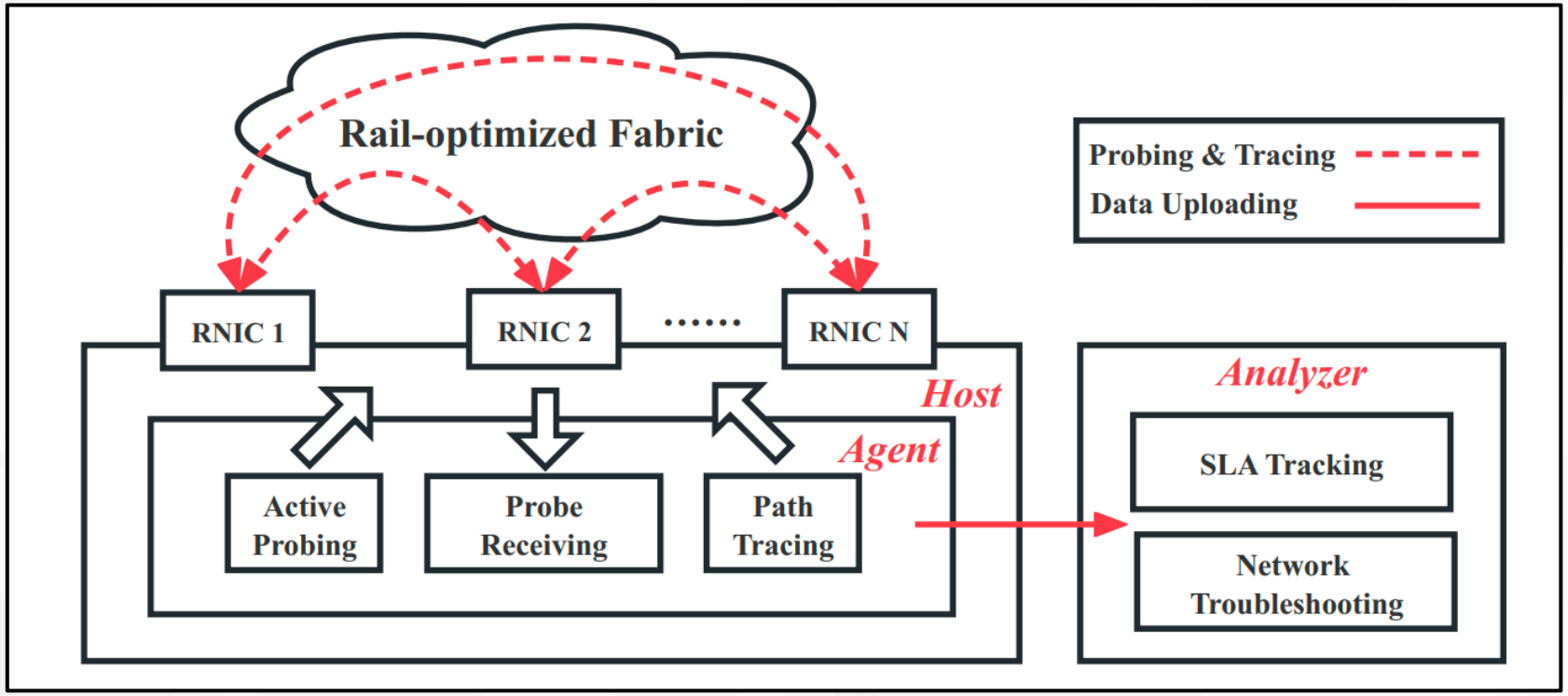
## ■ Benefits of Probing Between RNICs on the Same Host

- Get one-way latency and detect one-way packet drops with low overhead.
  - Probing between different servers typically involves both the probe and ACK paths. The problem with anomalous probes can be in either the probe path or the ACK path.
  - One-way probe data helps to locate network problems more accurately.
- No probe noise caused by high processing delay in the responder server.
  - In inter-server probing, if the responder has a high CPU load, it will spend more processing time responding with ACKs. In some severe cases, this will result in a probe timeout.

# Hostmesh Design

## ■ Hostmesh Framework

- **Agent**: Active probing and probe path tracing.
- **Analyzer**: SLA tracking and network troubleshooting.

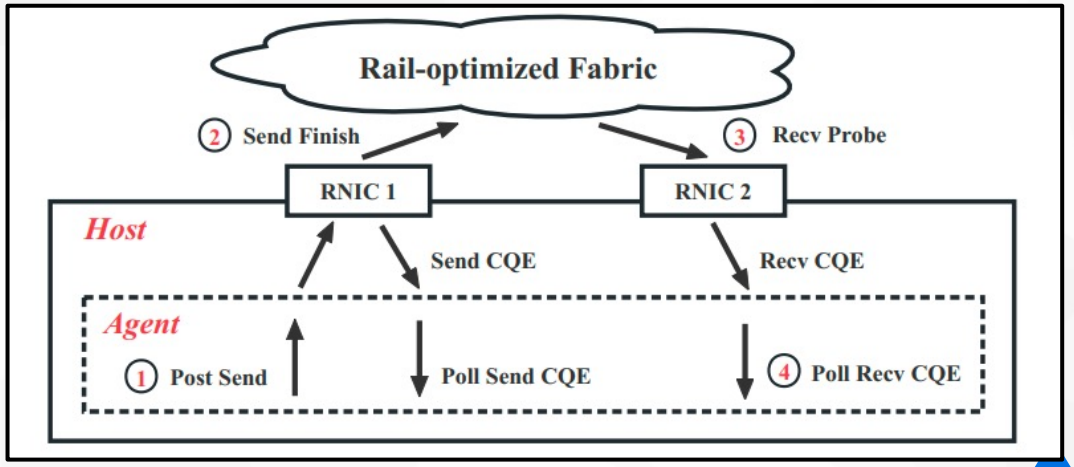


# Hostmesh Design

## ■ Hostemesh Agent - (1) Measure Processing Delay and One-way Latency

- **Choice of QP type: RC, UC, or UD?**
- **For accurate measurement:**
  - **Network latency: (③-②)**
  - **Processing delay: (④-①)-(③-②)**
  - **Commodity RNICs provide an RNIC timestamp when generating CQE.**
  - **All three QP types can obtain timestamps ③ from receive CQE.**
  - **② can only be obtained in UC or UD.**

| Features | RC | UC | UD |
|---|---|---|---|
| Accurate LAT Measurement | ✗ | ✓ | ✓ |
| Connection Overhead | High | High | **Low** |



# Hostmesh Design

## ■ Hostemesh Agent - (1) Measure Processing Delay and One-way Latency

- **Choice of QP type: RC, UC, or UD?**
- **For connection overhead:**
  - **With RC or UC, the local RNIC must create a QP for each target RNIC.**
  - **With UD, Agent only needs to create one QP on each RNIC, which has little connection overhead.**

**UD!**

| Features | RC | UC | UD |
|---|---|---|---|
| Accurate LAT Measurement | ✗ | ✓ | ✓ |
| Connection Overhead | High | High | **Low** |

# Hostmesh Design

## ■ Hostemesh Agent - (2) Path Tracing

- Periodically Traceroute the latest paths of probe packets.
- Easy to deploy and does not rely on advanced switch features such as INT.
- Continuous path tracing instead of path tracing when detecting anomalies.
  - Continuous path tracing allows Analyzer to locate network problems immediately as they occur.
  - In the case of a persistent failure, such as a link failure, the replayed dropped packets will be rehashed to other normal links, leading to inaccurate fault inference.

# Hostmesh Design

## ■ Hostemesh Analyzer - (1) Detect Anomalous RNICs in Real Time

- Detected packet drops can be attributed to RNICs or switches.
- If we can detect anomalous RNICs, we can filter out anomalous probes caused by RNICs and use switch drops to accurately locate switch network problems.
- A simple but effective method:
  - If many probes to an RNIC show anomalies at the same time, that RNIC is probably abnormal.
  - Premise: For each RNIC, there should be enough probes from other RNICs over a fine-grained time period.

# Hostmesh Design

## ■ Hostemesh Analyzer - (2) Locate Switch Network Problems

- **After detecting anomalous RNICs and filtering out anomalous probes caused by them, we can get anomalous probes caused only by switches.**
- **Analyzer uses a simple voting algorithm to infer abnormal devices. It traverses the paths of these probes and counts the number of times each link/switch is traversed.**
- **The link/switch with the highest number of votes is the most suspicious.**

**Algorithm 1** Identify the Most Suspicious Switch Links

**Input:** *abnormal paths*
**Output:** *the most suspicious links*

```
1:  function DetectAbnormalLinks()
2:      InitLinkStatus()
3:      for path_j in abnormal paths do
4:          for link_i in path_j do
5:              link_i.status ← abnormal
6:              link_i.abnormal_cnt ++
7:      return abnormal links with the largest abnormal_cnt
8:  function InitLinkStatus()
9:      for link_i in all network links do
10:         link_i.status ← normal
11:         link_i.abnormal_cnt ← 0
```

# Evaluation & Problems Found

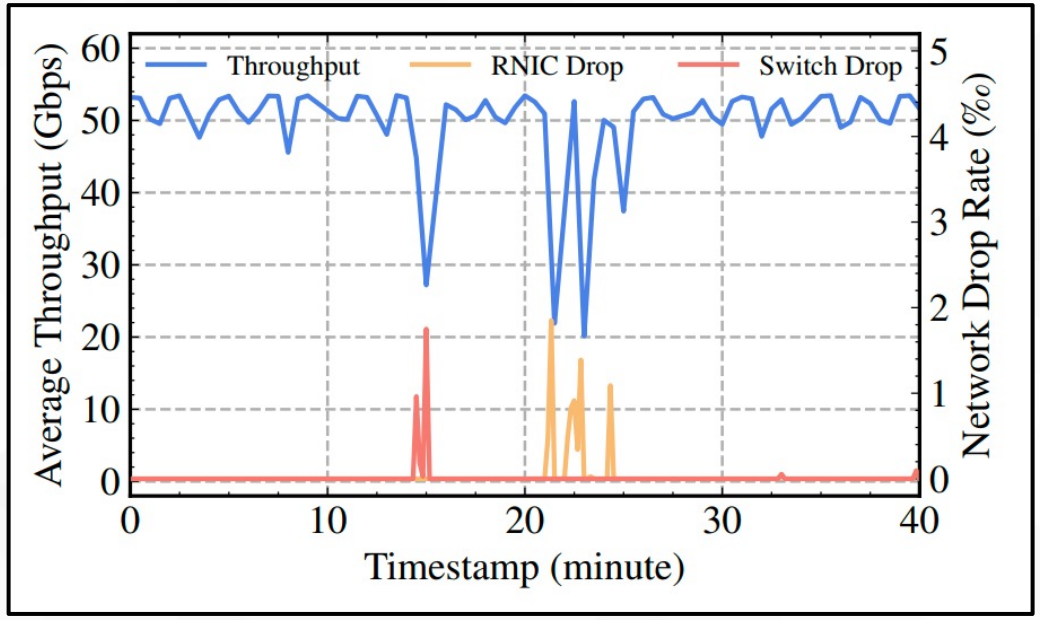
■ We Deployed Hostmesh for over 3 Months on a Multi-tenant DML RoCE Cluster

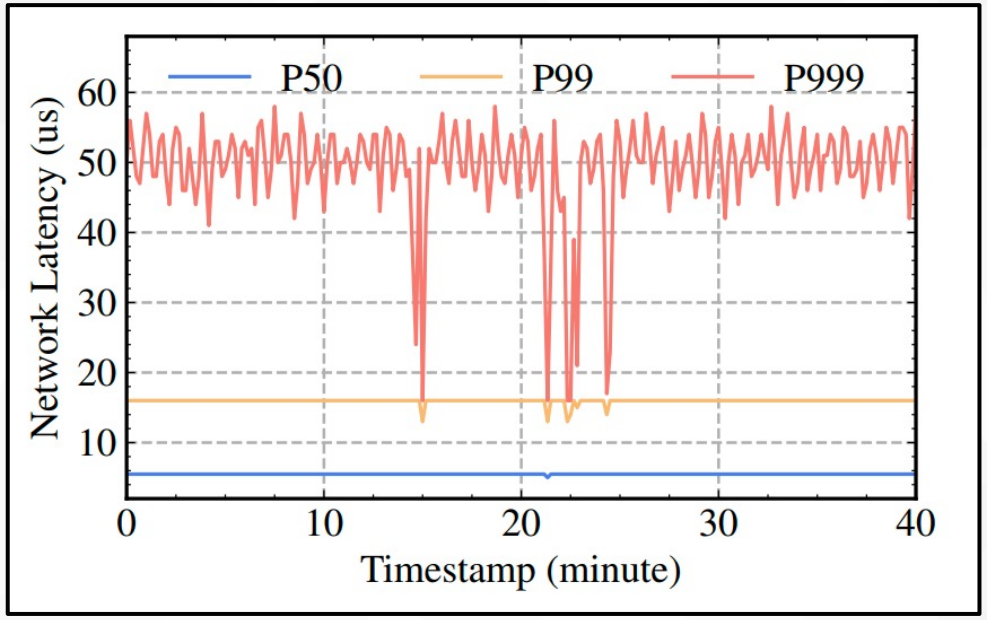
- Rail-optimized network topology.
- Hundreds of servers in total.
- Effectively detected and located 8 types of problems, including:
  - Hardware failures: (#1) RNIC/link flapping, (#2) packet corruption.
  - Misconfigurations: (#3) RNIC routing, (#4) switch ACL, and (5) PFC.
  - Network congestion: (#6) Uneven load balancing.
  - Intra-host bottlenecks: (#7) CPU overload, (#8) PCIe link downgrads.

# Evaluation & Problems Found

## ■ Detect and Categorize Packet Drops in Real Time

- Effectively determine the cause of degradations in training throughput.
- Accurately measure network latency and use it to reflect service states.



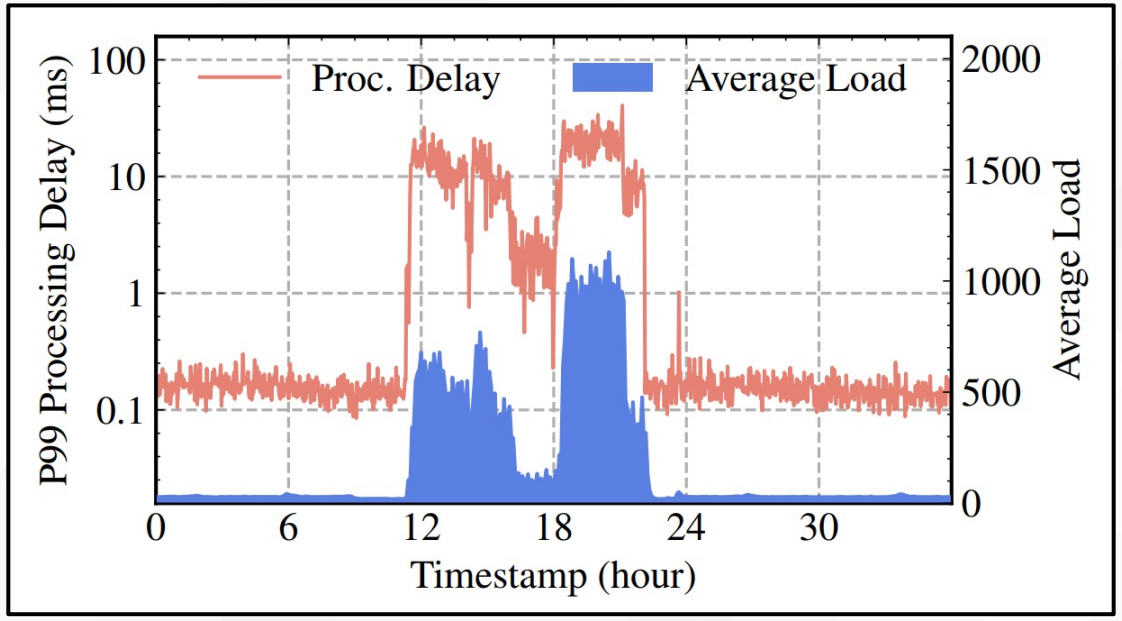

Packet drops lead to training throughput degradation

Measured network latency decreases at the same time

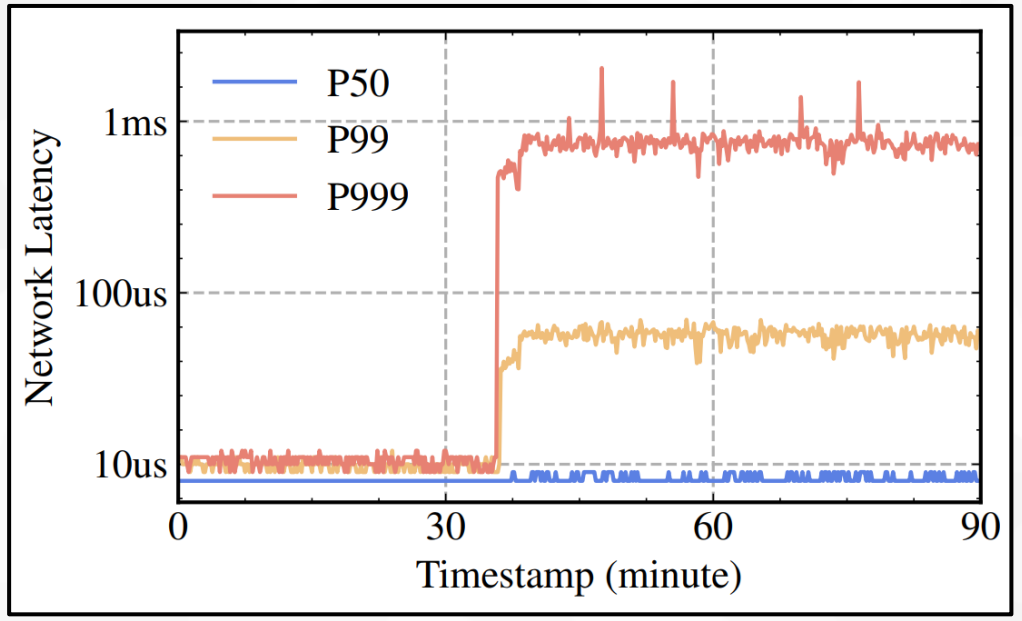
# Evaluation & Problems Found

## ■ Effectively Detect and Locate Performance Bottlenecks

- The measured end-host processing delay can **accurately reflect CPU utilization**.
- Use accurate network latency to **perceive network congestion**.




CPU overload results in high processing delay on a host




A PFC storm results in high tail network latency

# Conclusion

- We rethink the design of the network monitoring and diagnostic system.
- For rail-optimized RoCE clusters, Controller is not necessary.
  - Probing between RNICs on the same host can cover all links in the cluster.
- Easier to deploy in rail-optimized clusters, especially multi-tenant clusters.
- Finer-grained network measurements and less probing noise.

# THANK YOU