# One to Many: Closing the Bandwidth Gap in AI Datacenters with Scalable Multicast

Sepehr Abdous, Jinqi Lu, Jiacheng Wan, Erfan Sharafzadeh
Ying Zhang, Soudeh Ghorbani

November 2025


# AI is **everywhere**

**Chatbots**

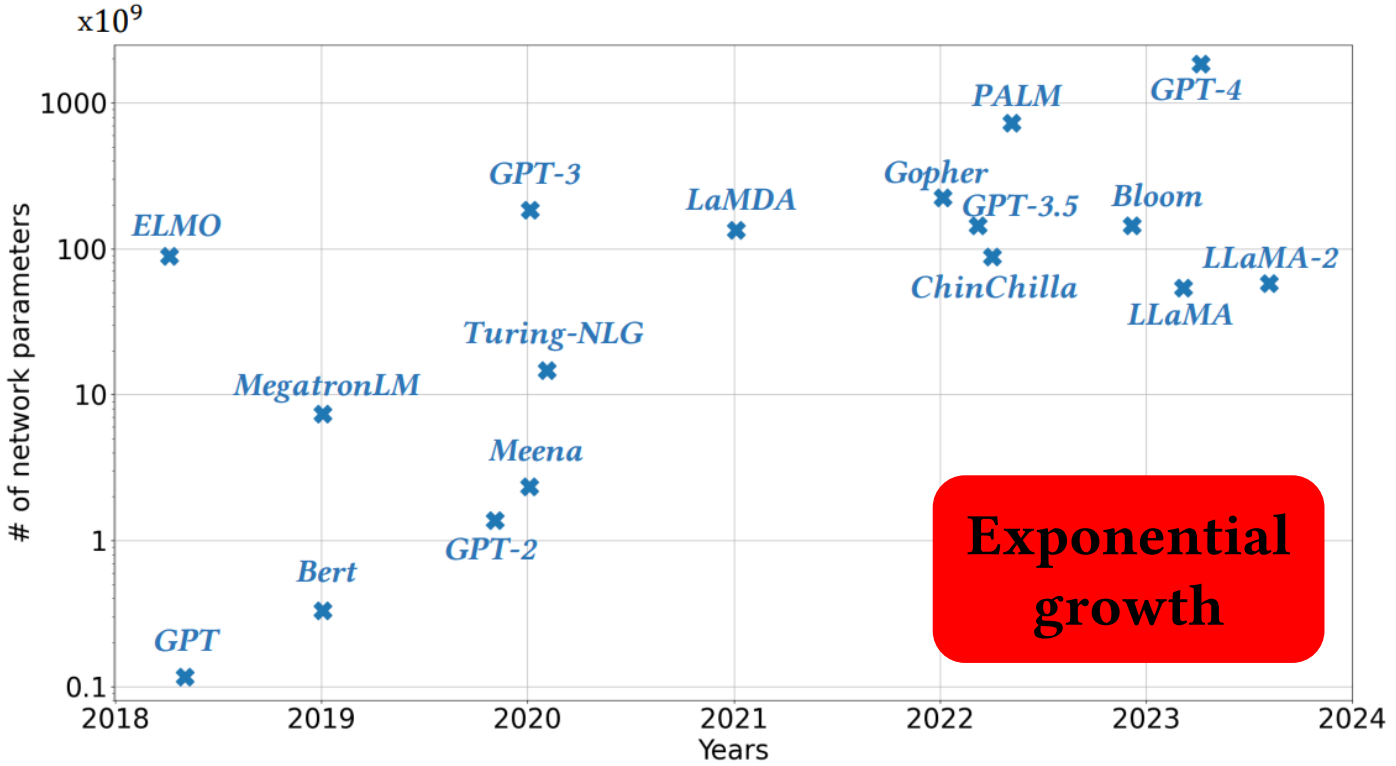
**Assisted driving**

**Healthcare**

zebra MEDICAL VISION

Insilico Medicine

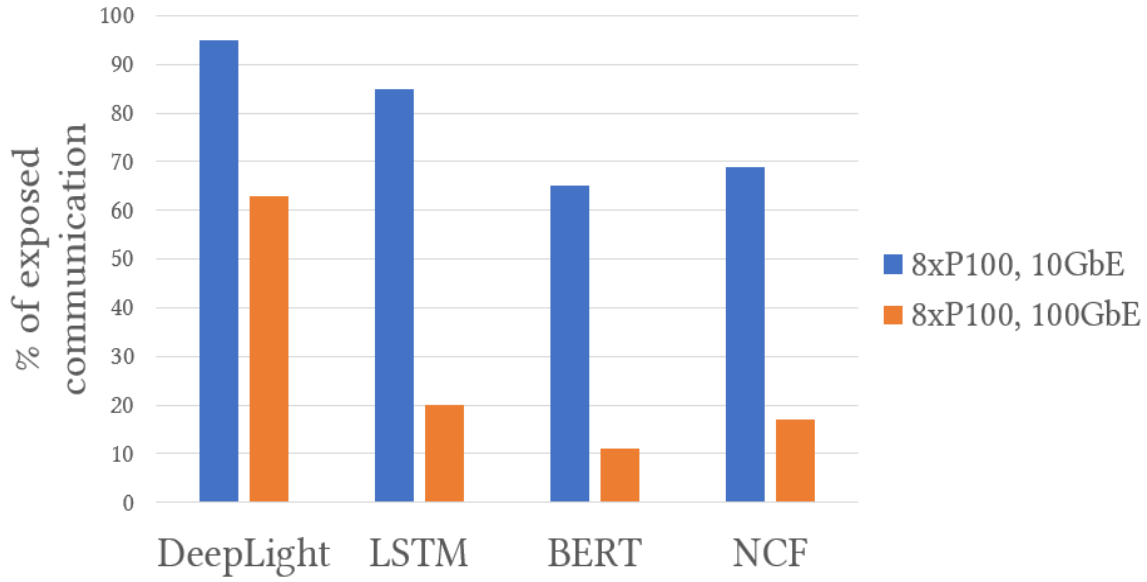
# Scale of AI training is growing fast




["Multibody Models Generated from Natural Language." Multibody System Dynamics '24]

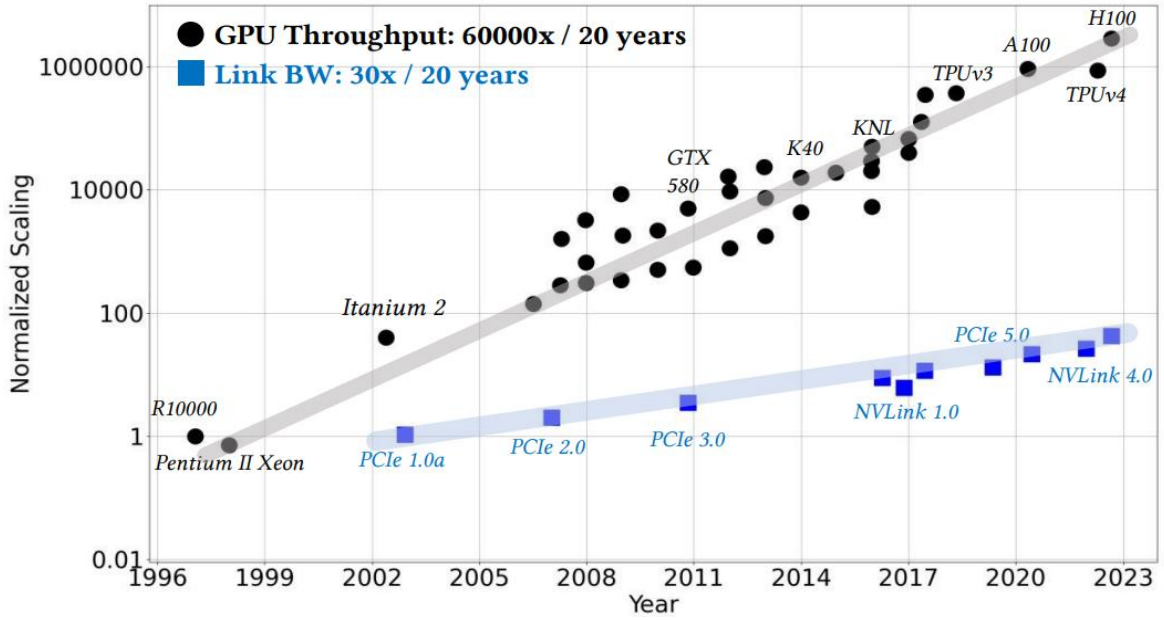
Large-scale AI training tasks are typically distributed across ***multiple machines***

## ***Efficient network communication*** is critical




["SwitchML Scaling Distributed Machine Learning with In-Network Aggregation" NSDI '21]

## ***Network bandwidth*** *is a major bottleneck*



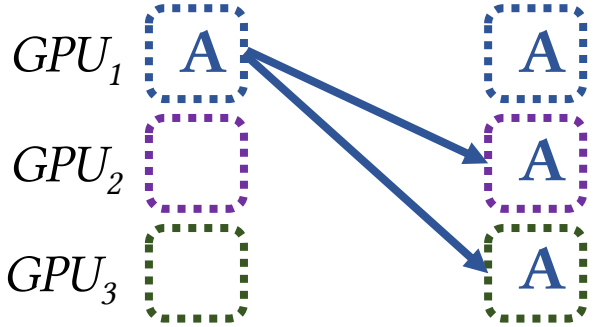

["AI and Memory Wall" IEEE Micro '24]

# Collective operations are **BW-inefficient**

Distributed AI training

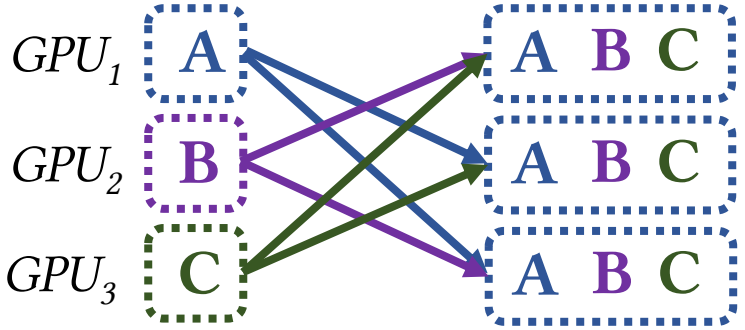
## Broadcast

Distributing gradients



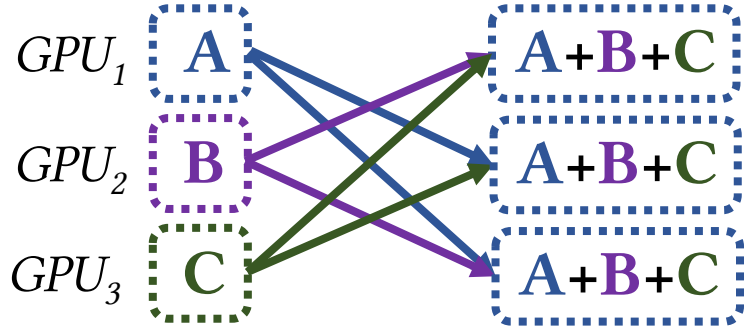

## All-Gather

Collecting model shards




## All-Reduce

Synchronizing gradients

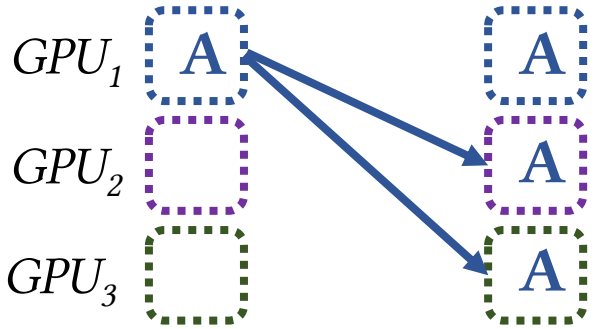


# Collective operations are **BW-inefficient**

Distributed AI training

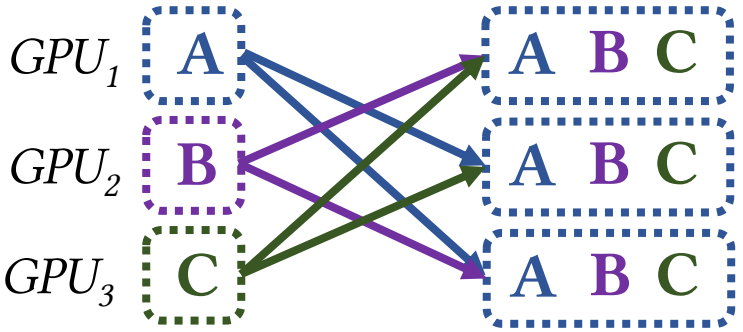
## Broadcast

Distributing gradients



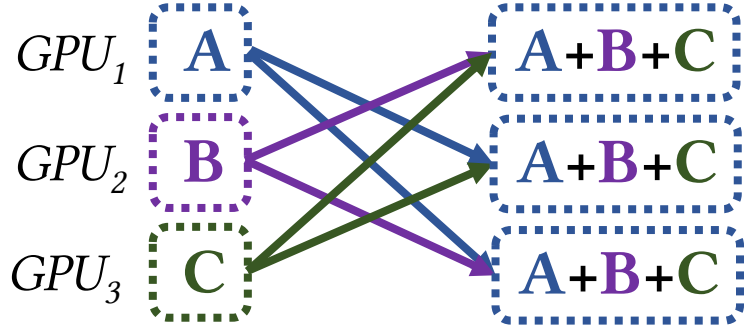

## All-Gather

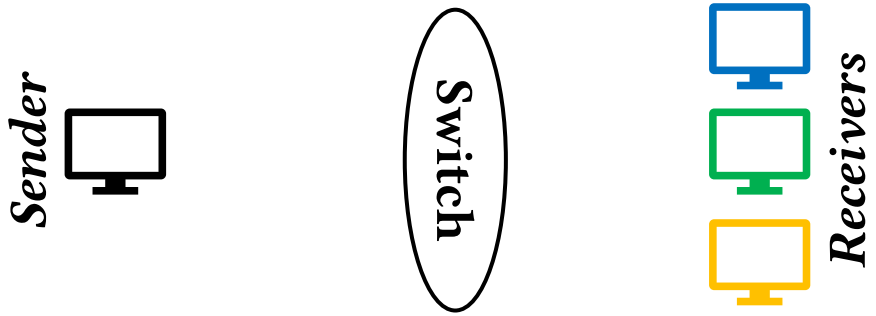
Collecting model shards




## All-Reduce

Synchronizing gradients



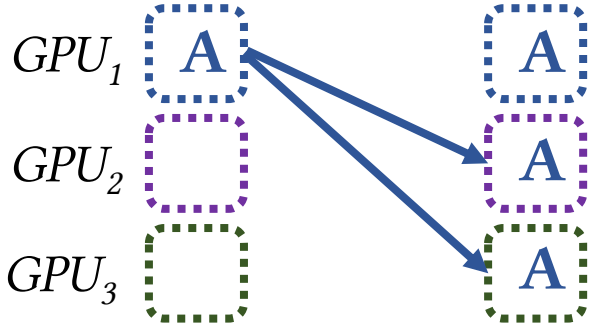



# Collective operations are **BW-inefficient**

Distributed AI training

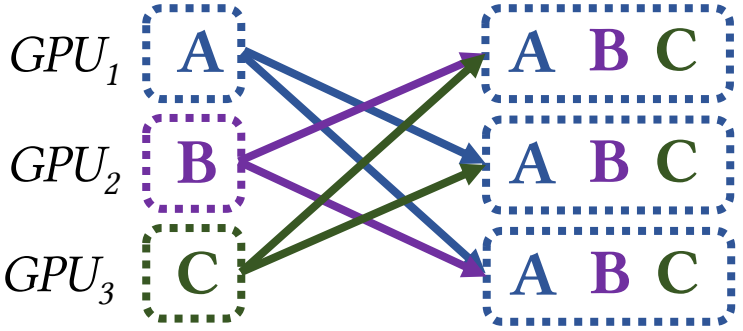
## Broadcast

Distributing gradients

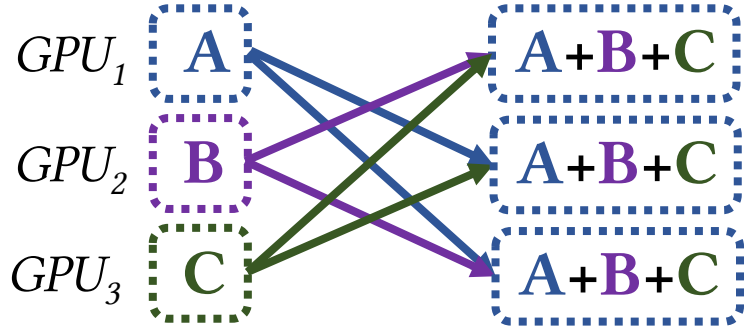


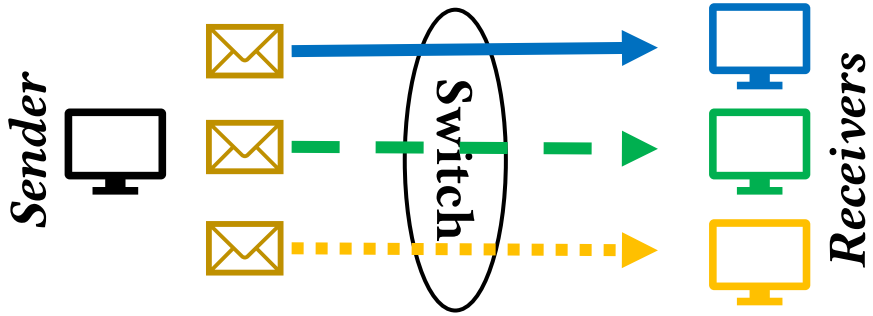

## All-Gather

Collecting model shards

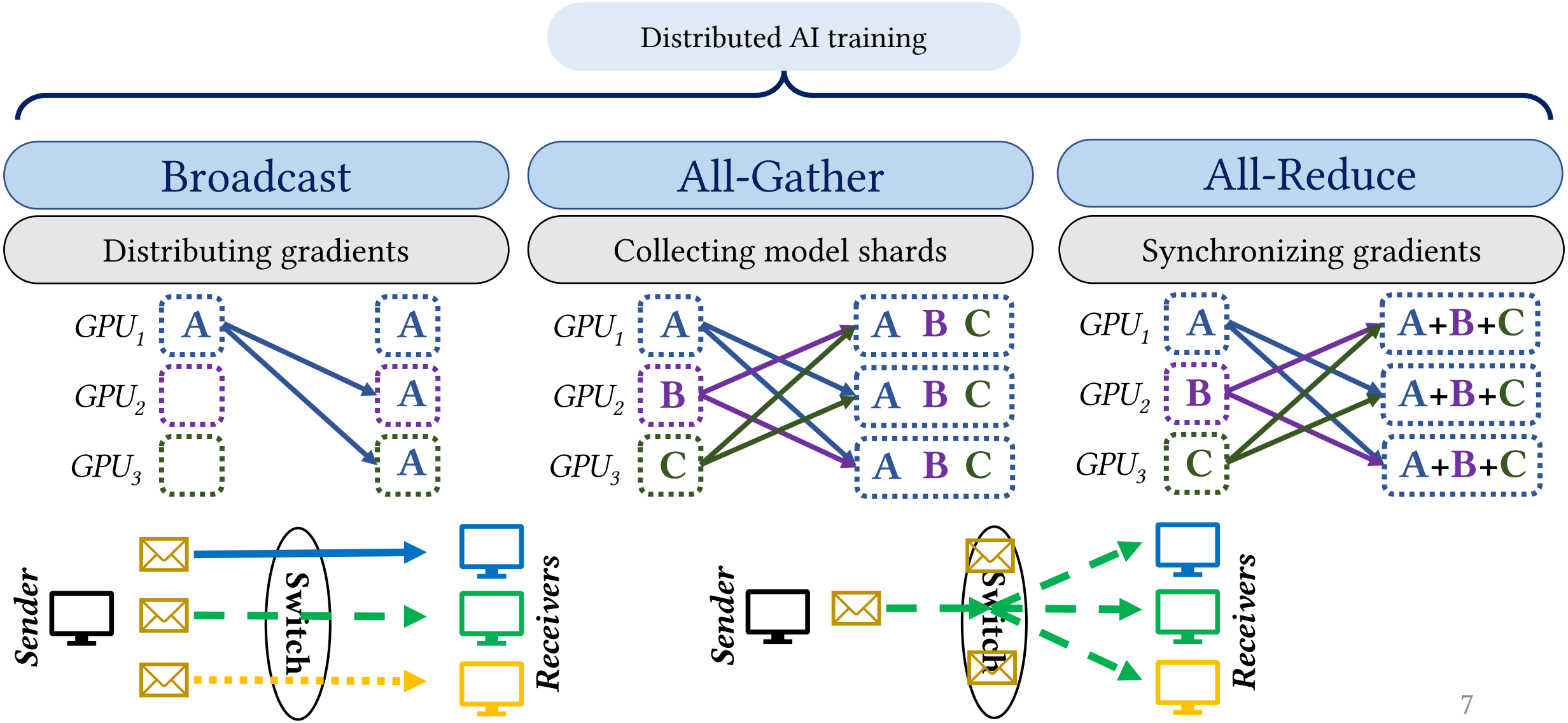



## All-Reduce



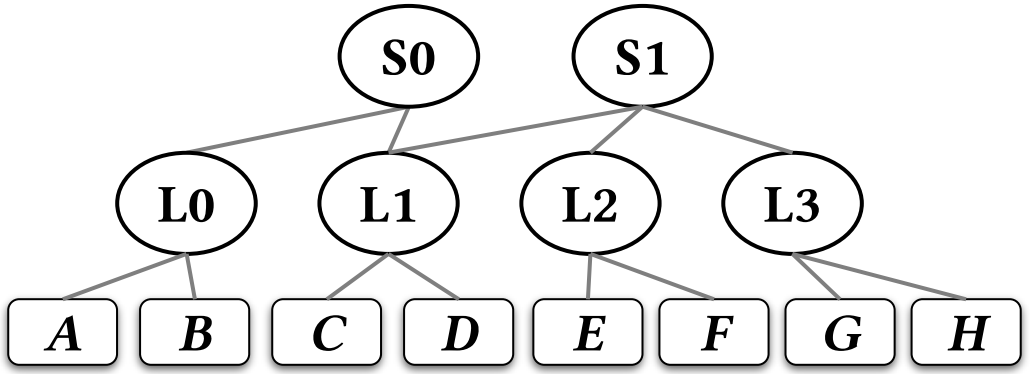
# Collective operations are **BW-inefficient**



# “Multicast is not scalable”

Computing the optimal multicast is **NP-hard**

[“ESM: Efficient and Scalable Data Center Multicast Routing” ToN ‘11]




Existing proposals cause **state explosion**

- *Naïve multicast* requires $2^{64}$ rules in a 64-ary fat-tree
- *Compression-based* mechanisms (*e.g.*, RSBF) create >100% packet header overhead in 32-ary fat-tree

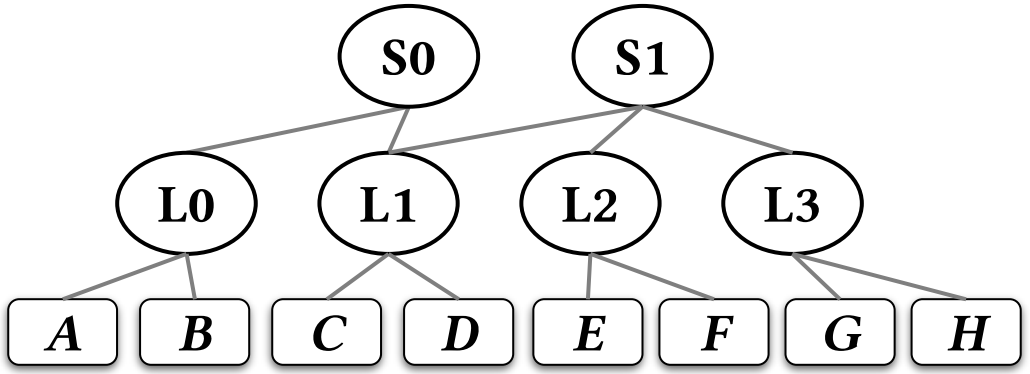
  [“Achieving High Efficiency for Datacenter Multicast using Skewed Bloom Filter.” ICPP ’24]
- *On-demand* approaches (*e.g.*, Orca) create multi-millisecond delay

  [“Orca: Server-assisted Multicast for Datacenter Networks.” NSDI ’22]

# "Multicast is ~~not~~ scalable"

Computing the optimal multicast is **NP-hard**

["ESM: Efficient and Scalable Data Center Multicast Routing" ToN '11]




Existing proposals cause **state explosion**

- *Naïve multicast* requires $2^{64}$ rules in a 64-ary fat-tree
- *Compression-based* mechanisms (*e.g.*, RSBF) create >100% packet header overhead in 32-ary fat-tree

["Achieving High Efficiency for Datacenter Multicast using Skewed Bloom Filter." ICPP '24]

- *On-demand* approaches (*e.g.*, Orca) create multi-millisecond delay

["Orca: Server-assisted Multicast for Datacenter Networks." SDI '22]

***PEEL***

**Layer-peeling greedy algorithm**

***Bounded** approximation*

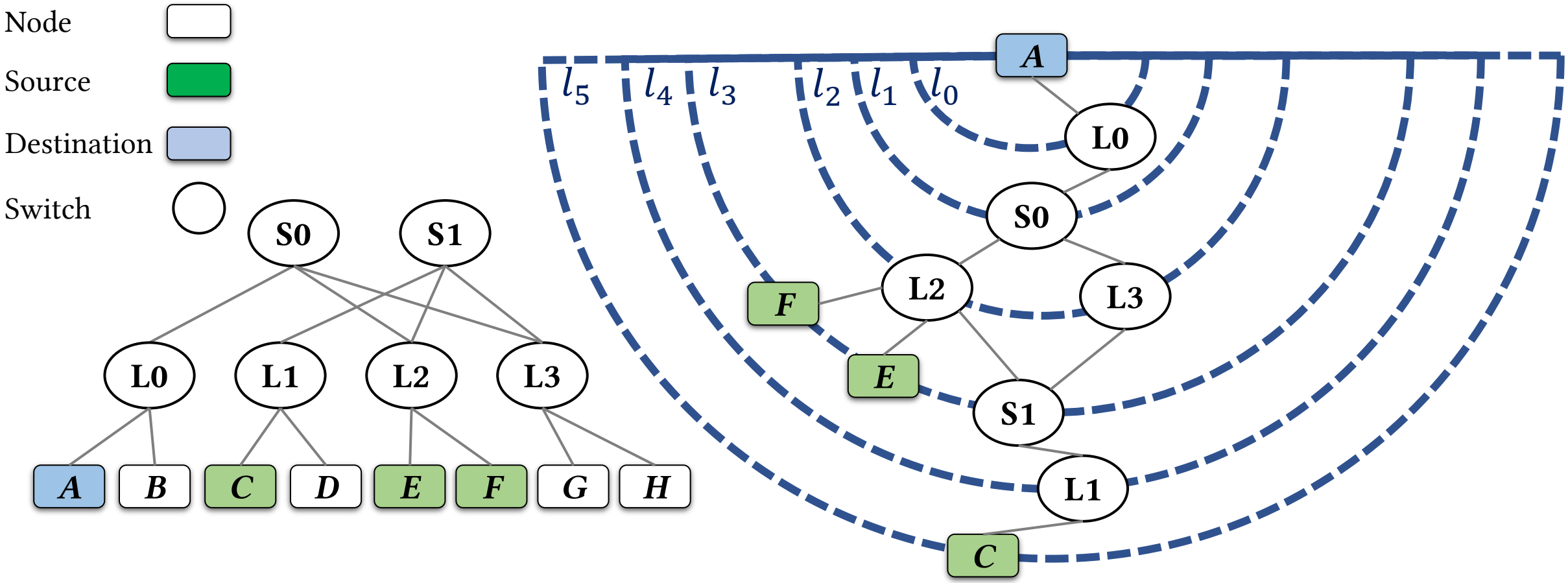
***Polynomial** execution time*

**Power-of-two prefix aggregation**

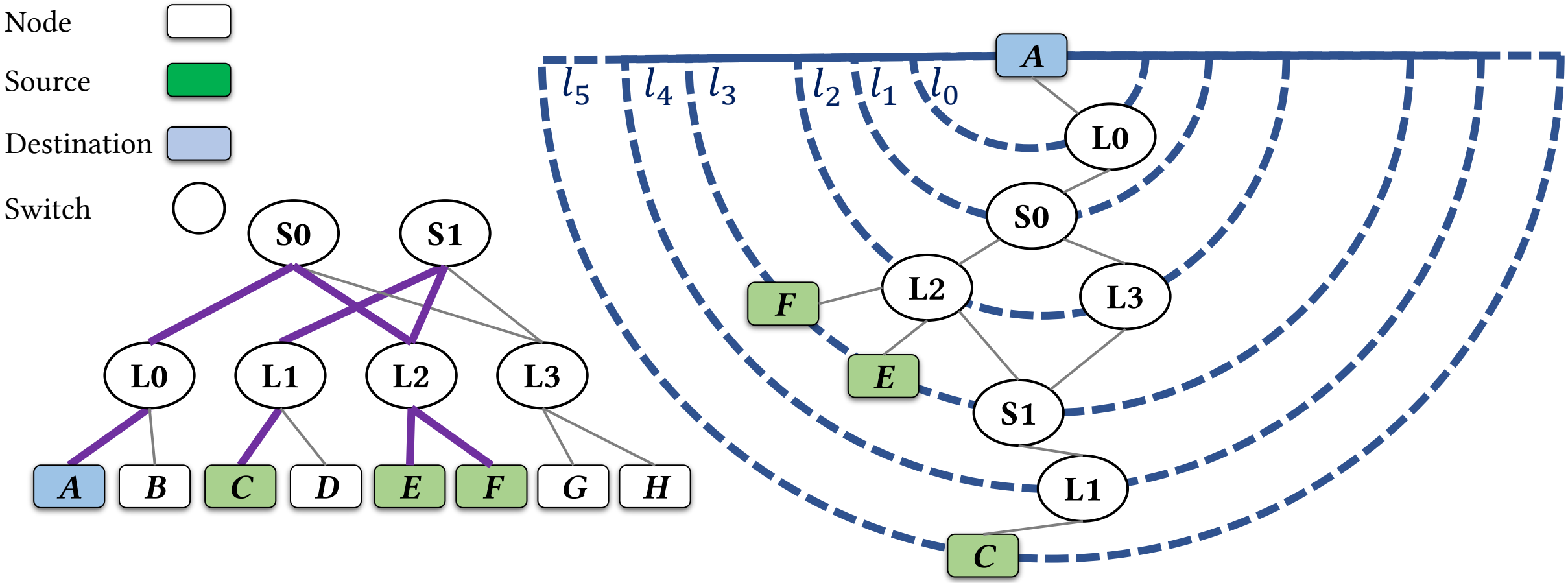
*From **4 billion** to only **31** rules*

*< **8 B** per packet overhead*

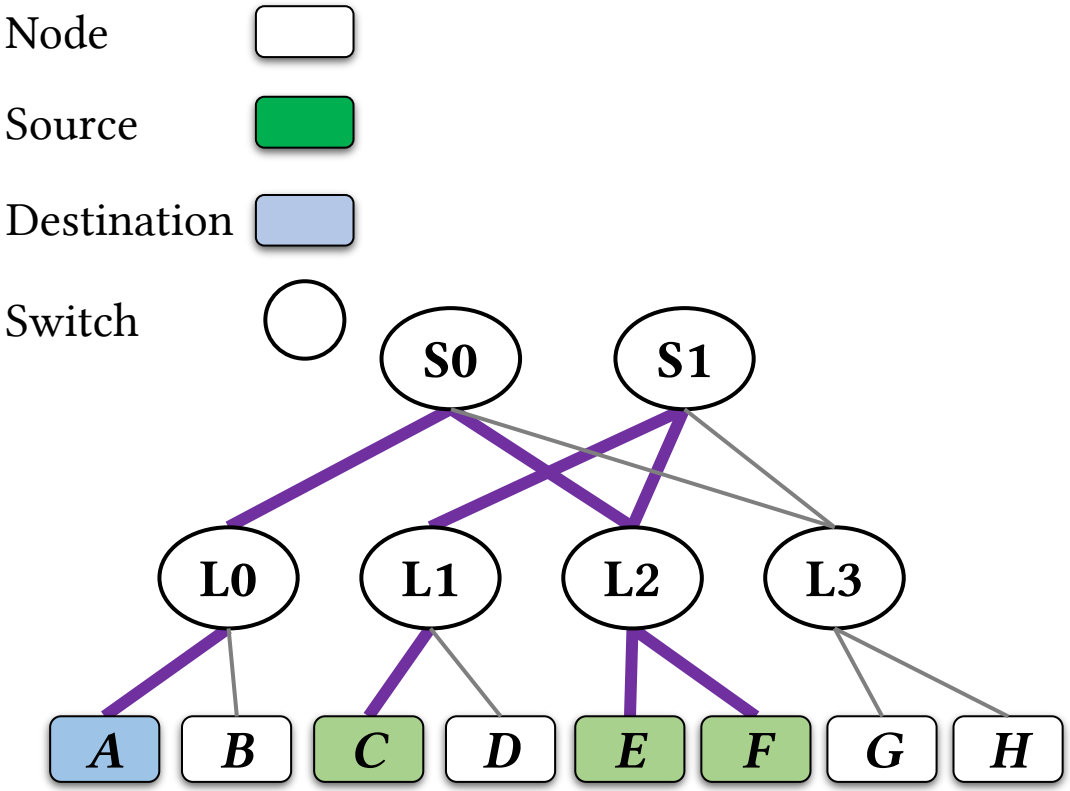
# **Peeling** complexity one **layer** at a time

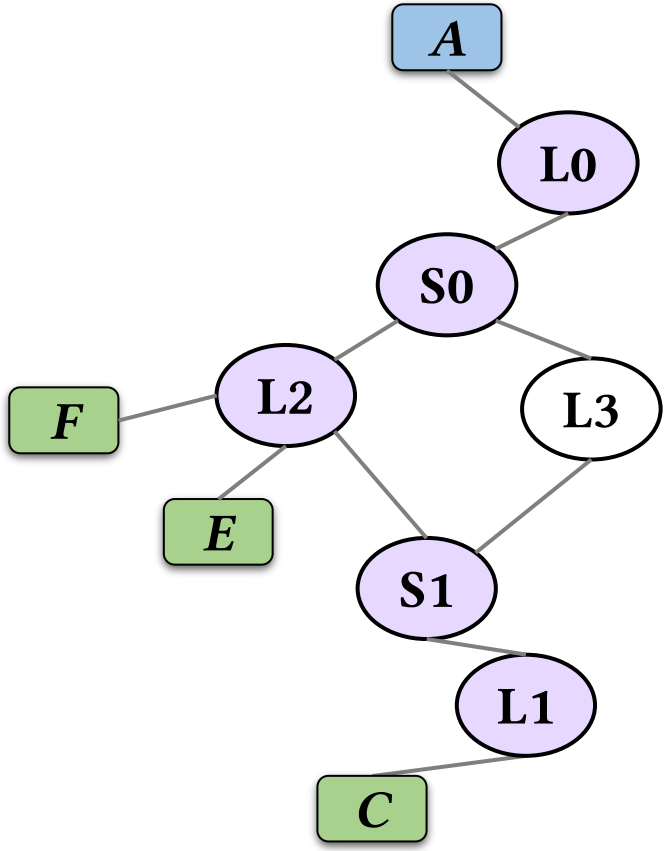


# **Peeling** complexity one **layer** at a time



# **Peeling** complexity one **layer** at a time



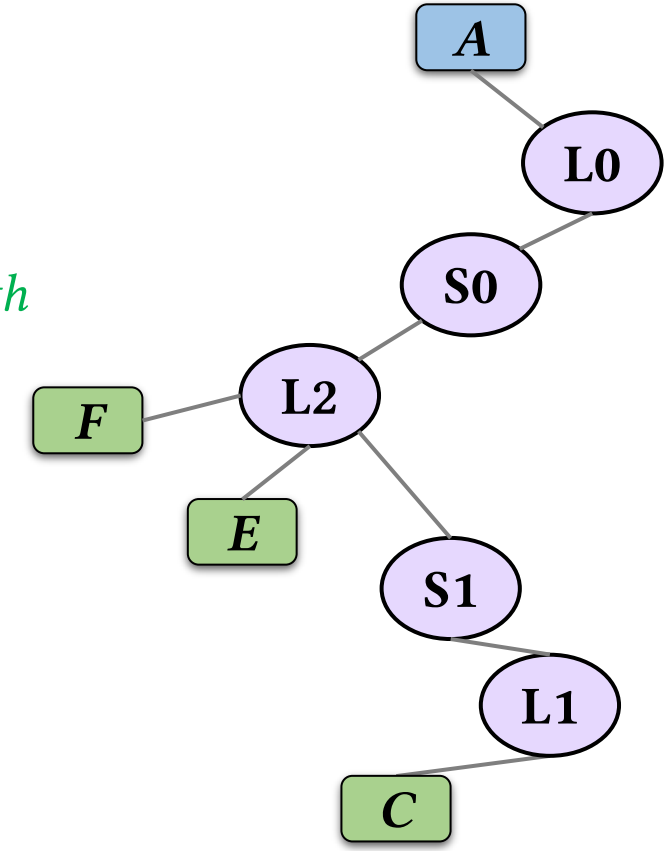



# **Peeling** complexity one **layer** at a time

*D: number of destinations*
*F: number of hop layers -- length of longest shortest path*

$$O(\min(D, F)) - \textit{approximation}$$

F is typically small in Clos networks
*The obtained multicast tree is* ***near-optimal***

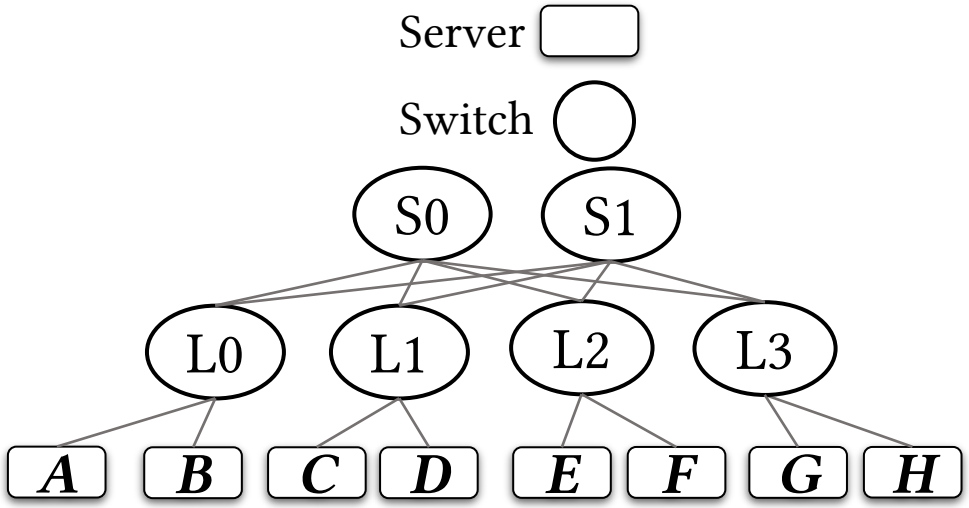


# Collapsing state space via **prefix aggregation**

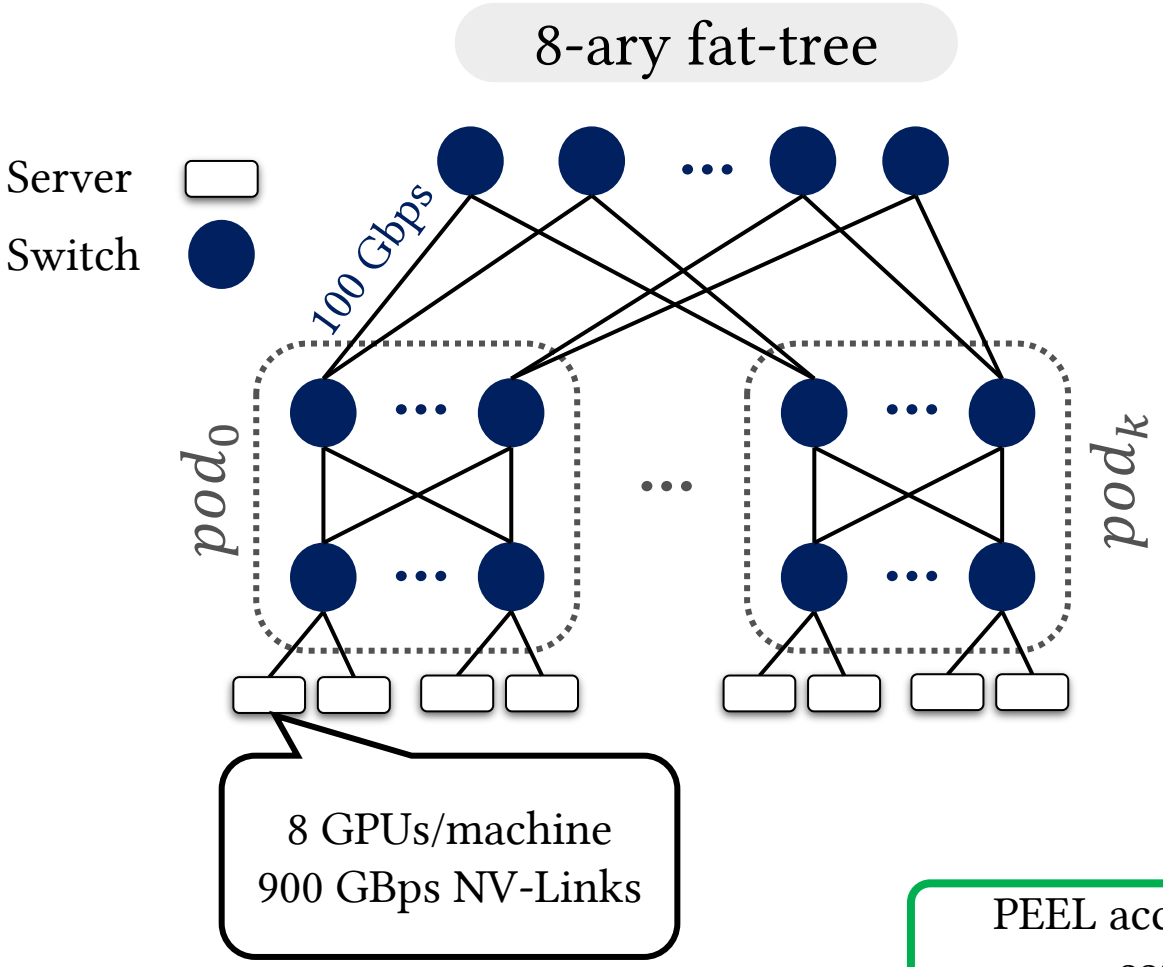
- AI job placement is **bin-packed**
- Switch state overhead can be **significantly** reduced using **power-of-two-prefix rules**
  - Changing switch state requirement from ***exponential*** to ***linear***
  - 32-ary fat-tree: number of states reduces from **4 billion** to **31**

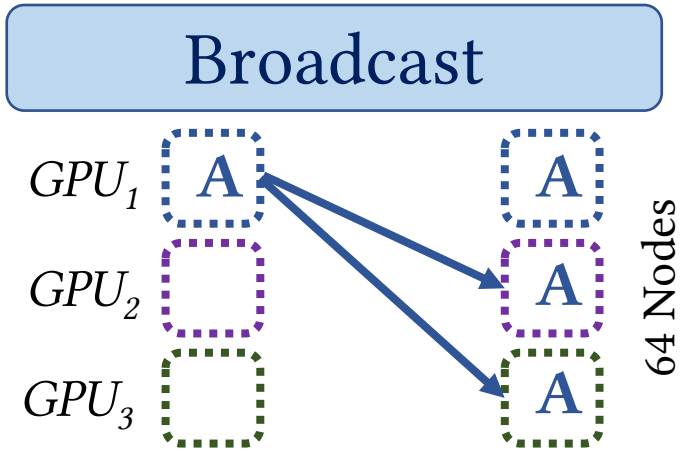
*Spine Match-Action Table*

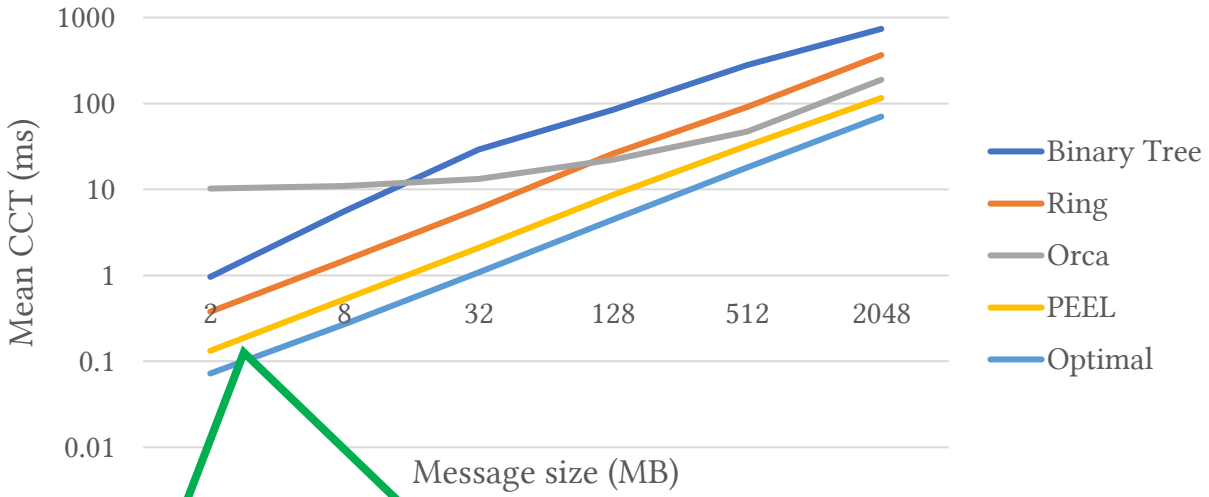
| **Match** (prefix/mask) | **Action** (output ports) |
|---|---|
| **/0 | forward(0,1,2,3) |
| 0*/1 | forward(0,1) |
| 1*/1 | forward(2,3) |
| 00/2 | forward(0) |
| 01/2 | forward(1) |
| 10/2 | forward(2) |
| 11/2 | forward(3) |



# PEEL outperforms state-of-the-art techniques




PEEL accelerates Broadcast by up to **77x** compared to other baselines

# Takeaway: **Multicast is scalable**

**PEEL achieves:**

***Linear*** switch state requirements

***Small*** packet header overhead

***Up to 77x*** performance benefit

**Open research avenues:**

- Multicast x **Transport**
- Multicast x **Heterogeneity**