Multipass Random Leaf-Spine Networks for Extreme Scale Computing Systems

A. Cano, C. Camarero, C. Martínez, R. Beivide
Computer Architecture Research Group
University of Cantabria (Spain)

August 19, 2026

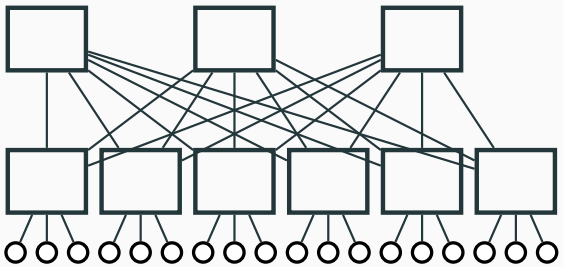
Outline

Introduction

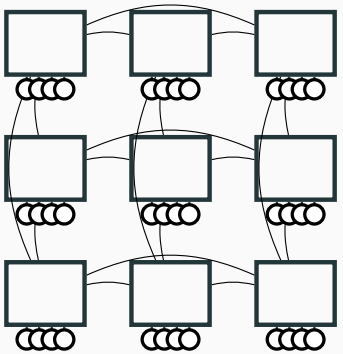
Network topologies for DC and HPC

Computational demands are growing (AI, HPC, etc.), massive **increase in computing endpoints**.

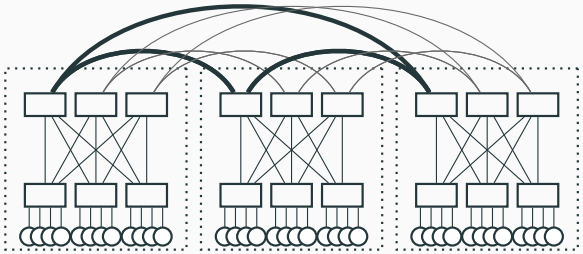
- Number of ports per switch (**radix**).
- Low diameter and low average distance.



(a) Fat-Tree



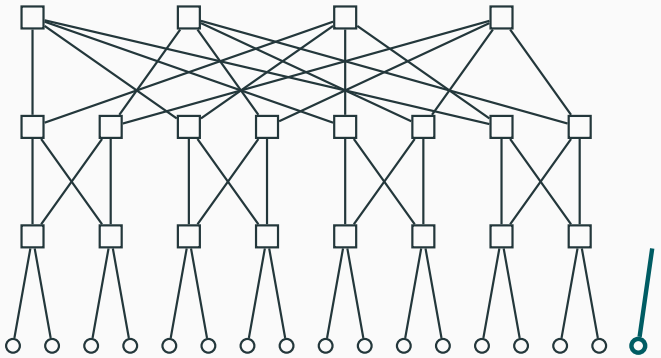
(b) HyperX



(c) Dragonflies

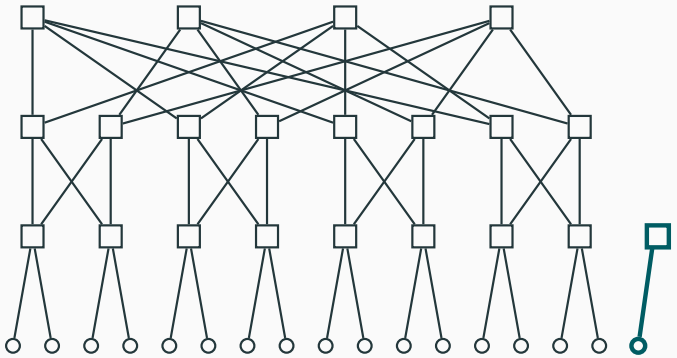
The problem of network topologies

Can new endpoints be added to a deployed network?



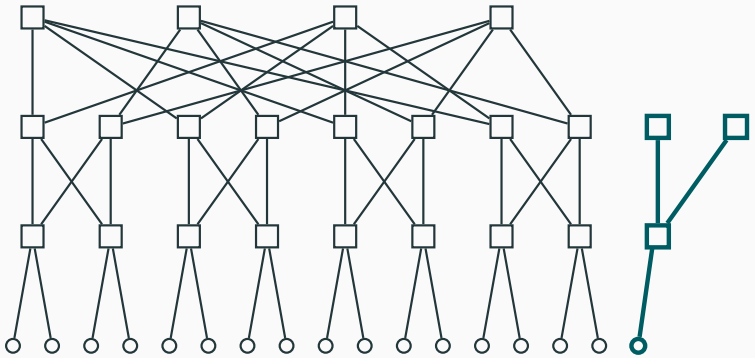
The problem of network topologies

Can new endpoints be added to a deployed network?



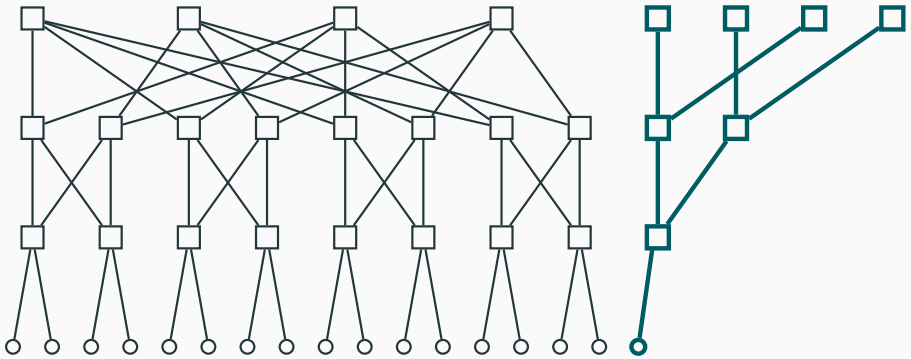
The problem of network topologies

Can new endpoints be added to a deployed network?



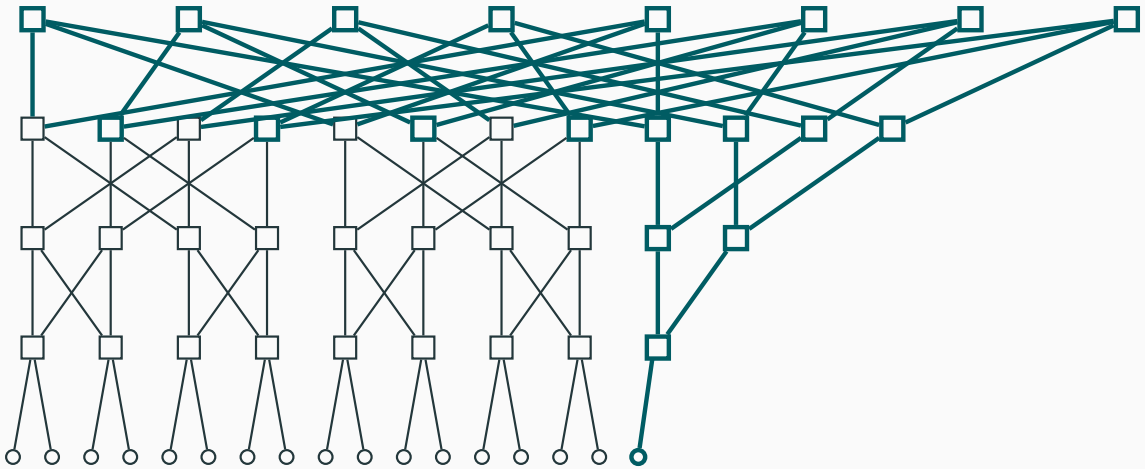
The problem of network topologies

Can new endpoints be added to a deployed network?



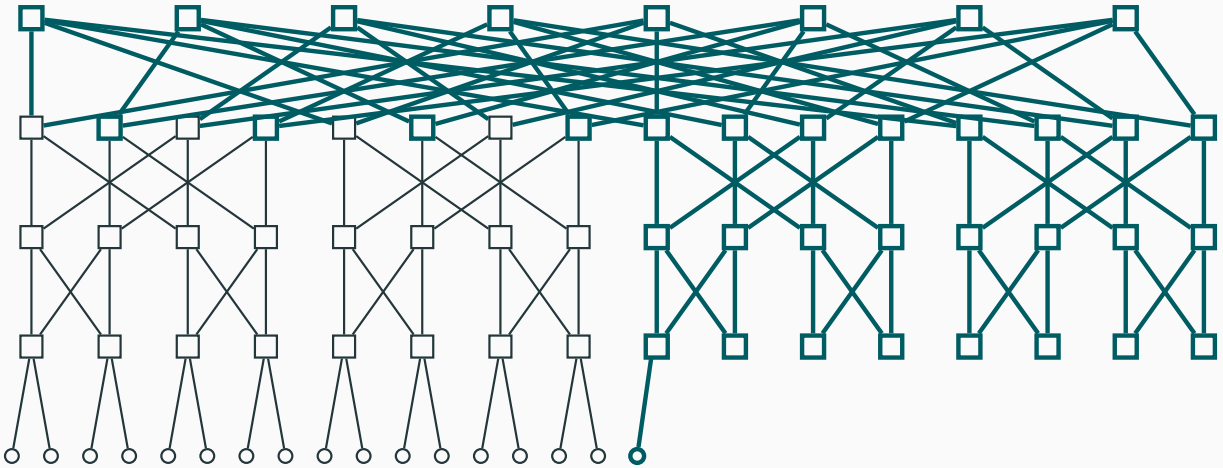
The problem of network topologies

Can new endpoints be added to a deployed network?



The problem of network topologies

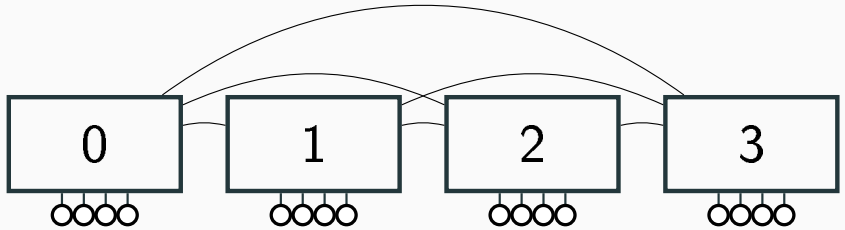
Can new endpoints be added to a deployed network?



- Additional level of switches even for a small increment.
- Increment in the diameter of the network.

The problem of network topologies

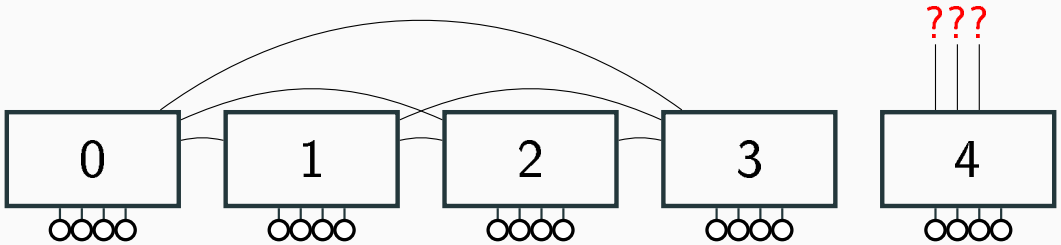
Can new endpoints be added to a deployed network?



Full-Mesh

The problem of network topologies

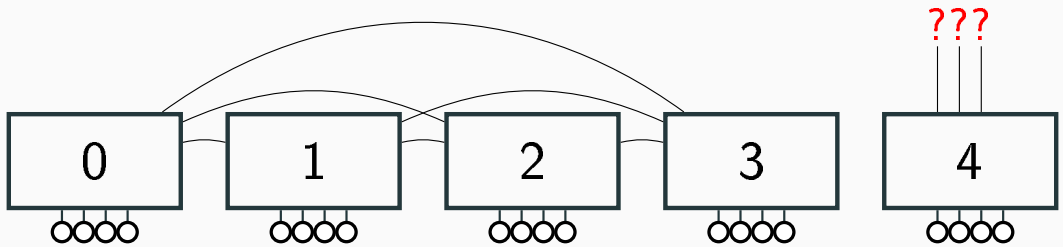
Can new endpoints be added to a deployed network?



Full-Mesh

The problem of network topologies

Can new endpoints be added to a deployed network?



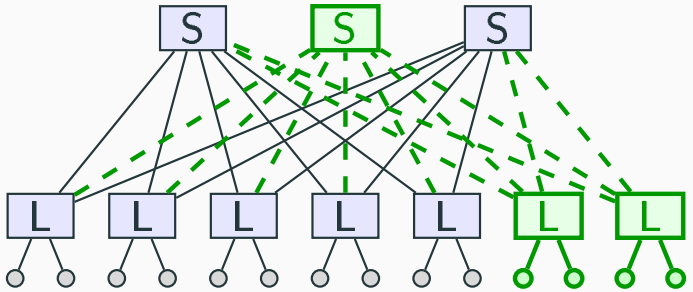
Full-Mesh

- Very rigid structure, not clear how to expand.

A desirable property for topologies

Expandability: The capacity to add new endpoints to a deployed network of a supercomputer without a big cost/effort.

- **Allows phased deployments**.
- **Zero Downtime** in integration of new racks or nodes.
- **Predictable Performance** preserving basic network properties.



Key Achievements

The **Multipass Random Leaf Spine (MRLS)** architecture delivers:

Cost-Efficient
& High-Performing

Optimized Routing
& Simplified Layout

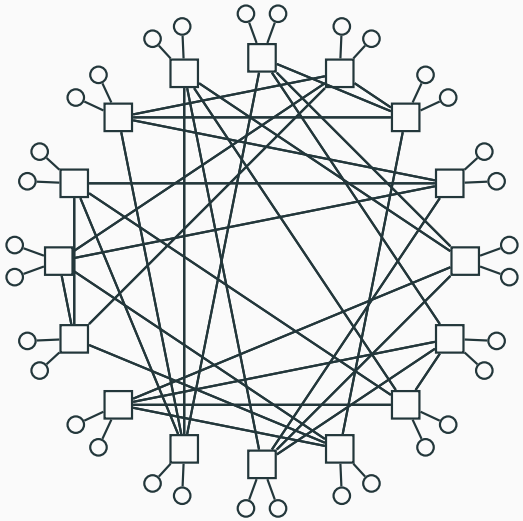
Unlimited
Expandability

Ideal for large-scale Data Centers and HPC environments

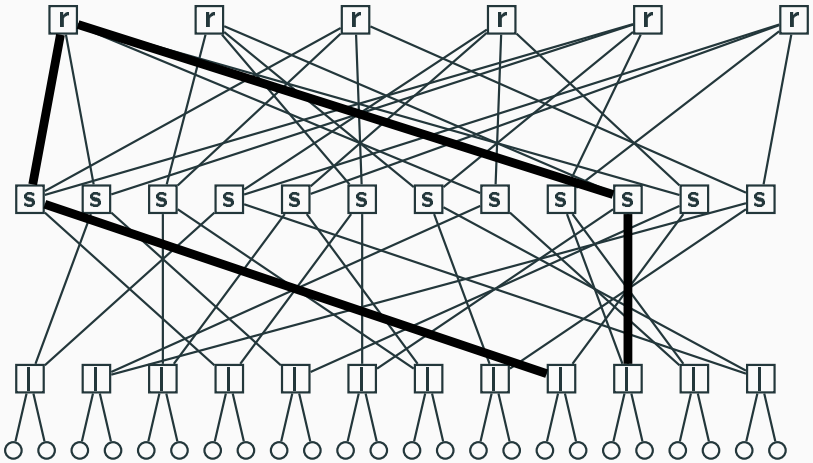
UC UNIVERSIDAD DE CANTABRIA
Computer Architecture and Technology

Random topologies

Types of random networks



(a) Jellyfish network (RRG)

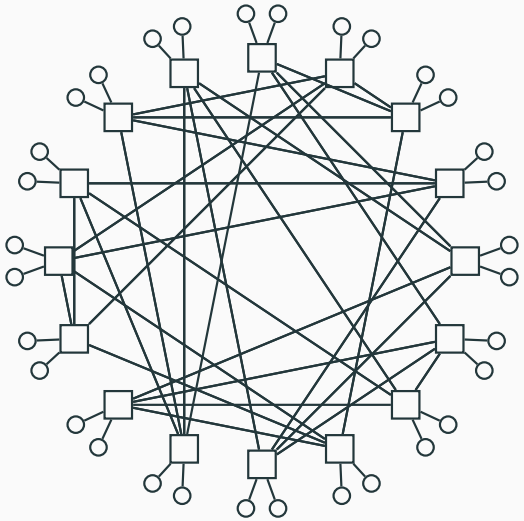


(b) Random Folded Clos (RFC)

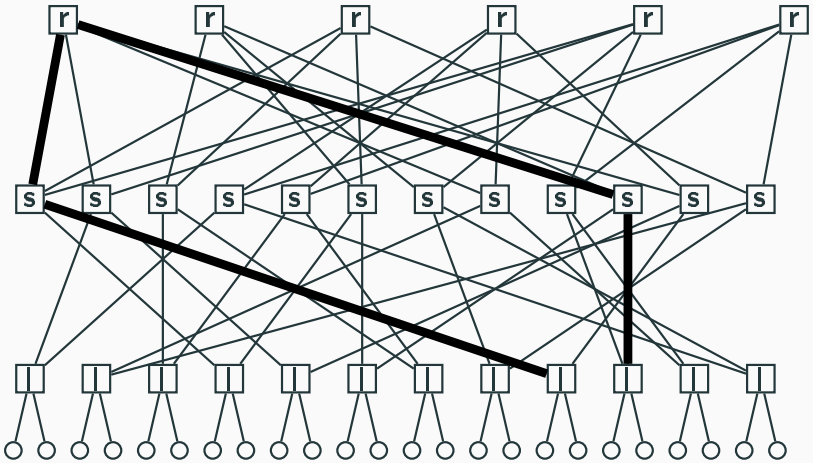
[1] Singla, A., et al. "Jellyfish: Networking data centers randomly." *NSDI*. 2012.
[2] Camarero, C., et al. "Random folded Clos topologies for datacenter networks." *HPCA*. 2017.

Types of random networks

Random networks are **cost-efficient** and **expandable**.



(a) Jellyfish network (RRG)

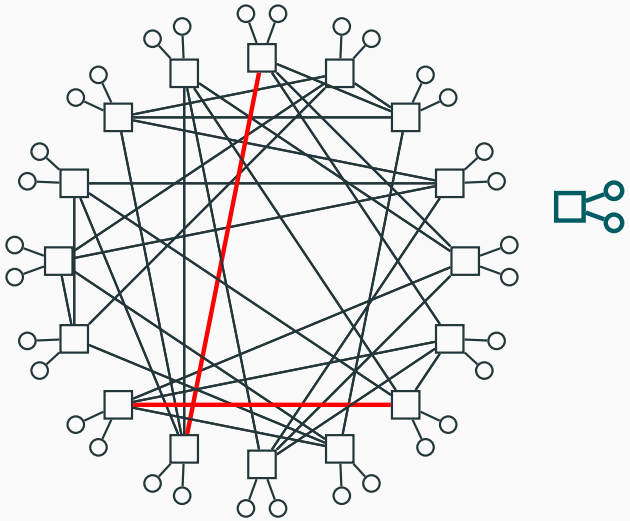


(b) Random Folded Clos (RFC)

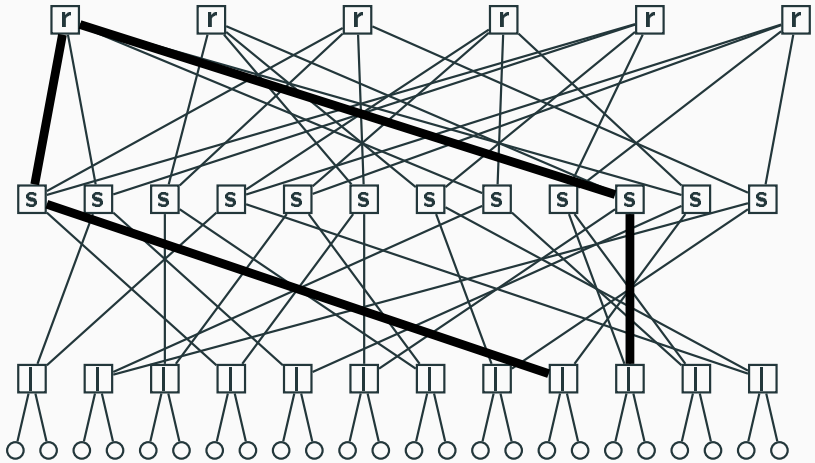
[1] Singla, A., et al. "Jellyfish: Networking data centers randomly." *NSDI*. 2012.
[2] Camarero, C., et al. "Random folded Clos topologies for datacenter networks." *HPCA*. 2017.

Types of random networks

Random networks are **cost-efficient** and **expandable**.



(a) Jellyfish network (RRG)

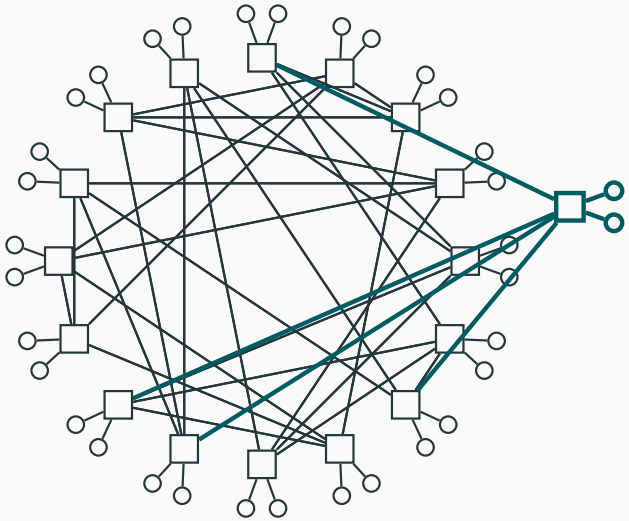


(b) Random Folded Clos (RFC)

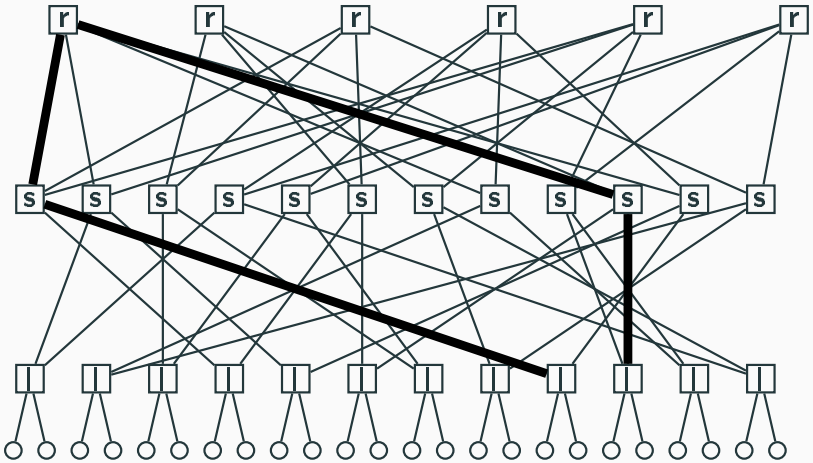
[1] Singla, A., et al. "Jellyfish: Networking data centers randomly." *NSDI*. 2012.
[2] Camarero, C., et al. "Random folded Clos topologies for datacenter networks." *HPCA*. 2017.

Types of random networks

Random networks are **cost-efficient** and **expandable**.



(a) Jellyfish network (RRG)

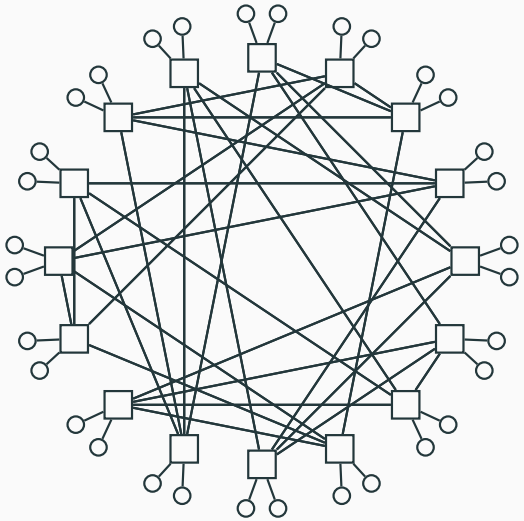


(b) Random Folded Clos (RFC)

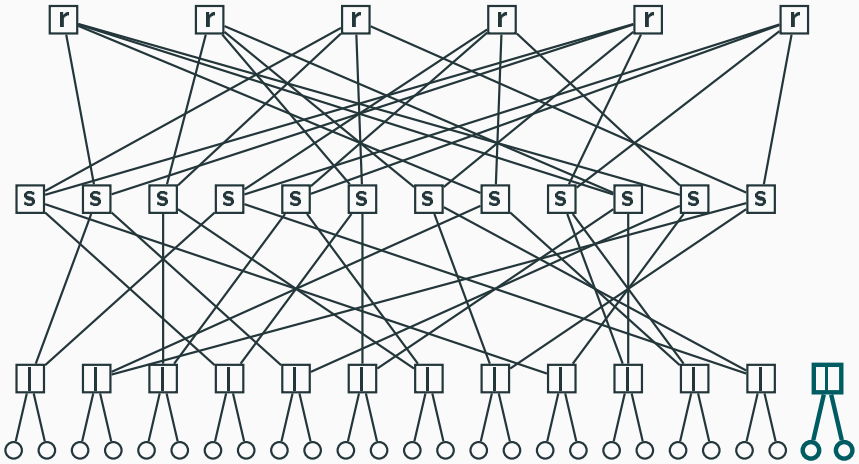
[1] Singla, A., et al. "Jellyfish: Networking data centers randomly." *NSDI*. 2012.
[2] Camarero, C., et al. "Random folded Clos topologies for datacenter networks." *HPCA*. 2017.

Types of random networks

Random networks are **cost-efficient** and **expandable**.



(a) Jellyfish network (RRG)

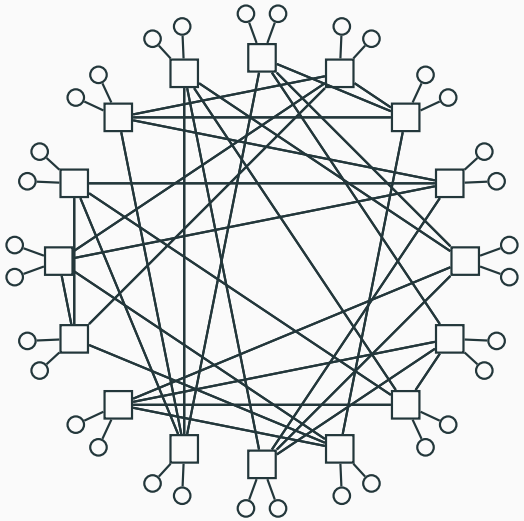


(b) Random Folded Clos (RFC)

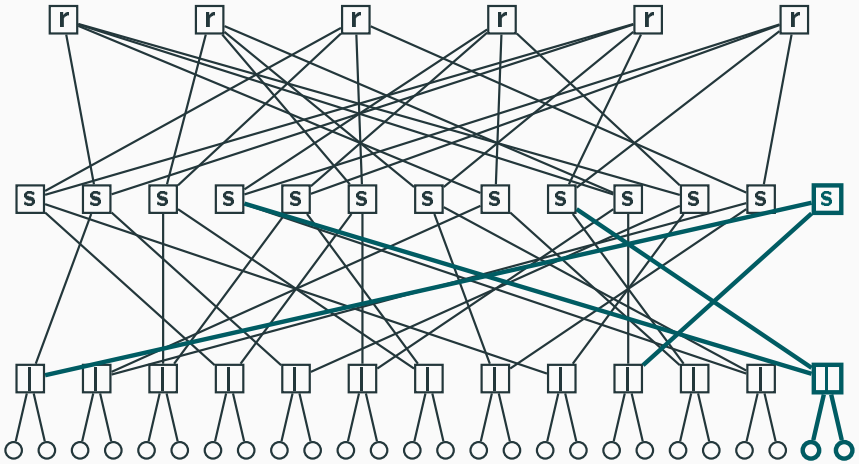
[1] Singla, A., et al. "Jellyfish: Networking data centers randomly." *NSDI*. 2012.
[2] Camarero, C., et al. "Random folded Clos topologies for datacenter networks." *HPCA*. 2017.

Types of random networks

Random networks are **cost-efficient** and **expandable**.



(a) Jellyfish network (RRG)

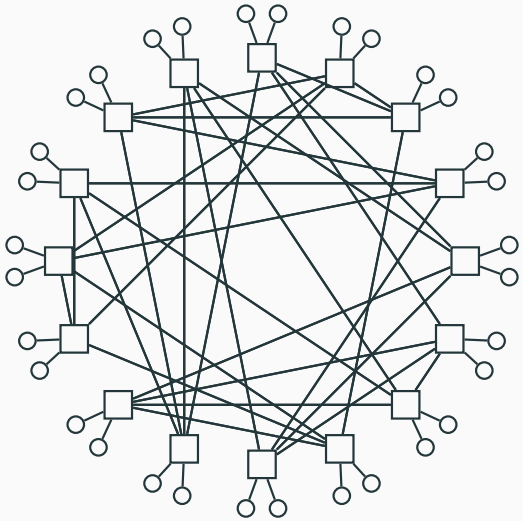


(b) Random Folded Clos (RFC)

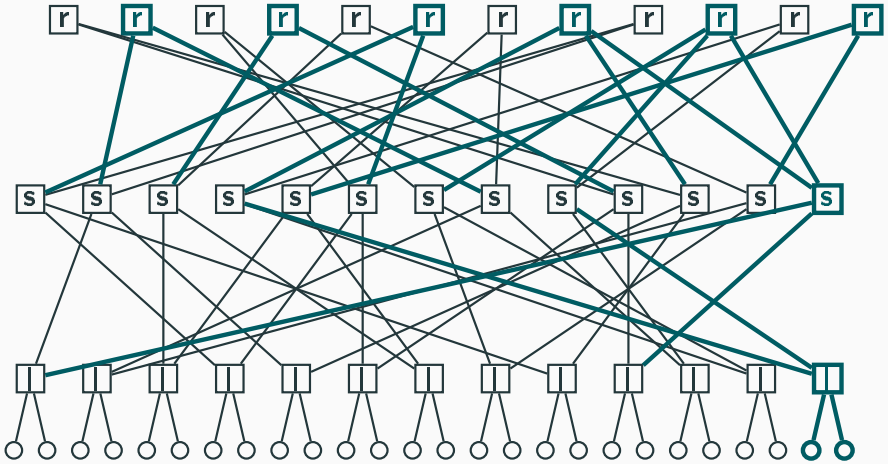
[1] Singla, A., et al. "Jellyfish: Networking data centers randomly." *NSDI*. 2012.
[2] Camarero, C., et al. "Random folded Clos topologies for datacenter networks." *HPCA*. 2017.

Types of random networks

Random networks are **cost-efficient** and **expandable**.



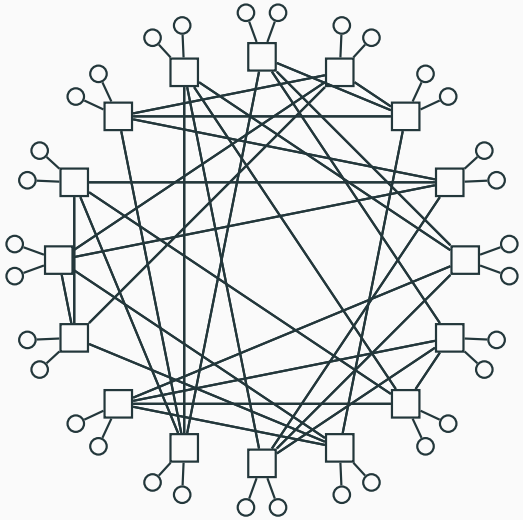
(a) Jellyfish network (RRG)



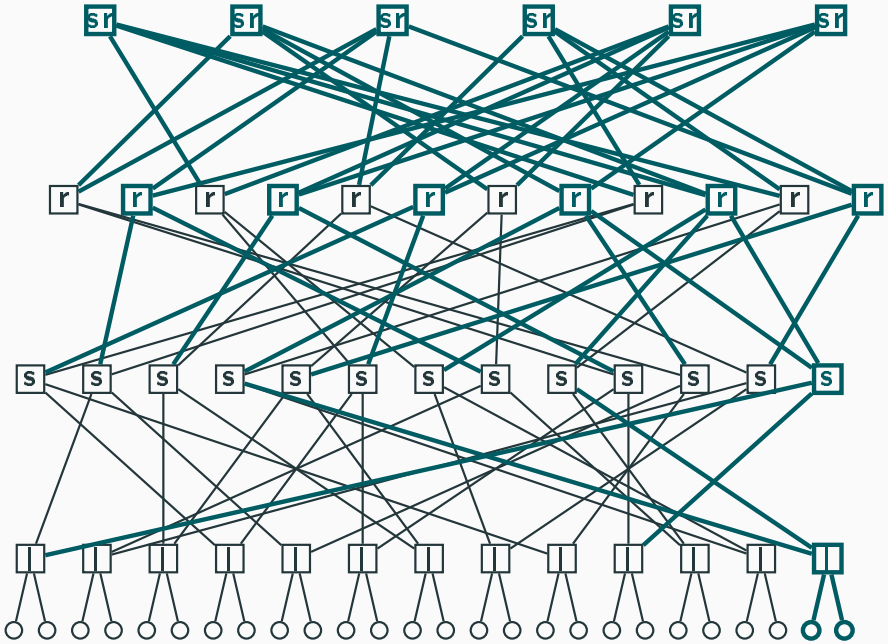
(b) Random Folded Clos (RFC)

[1] Singla, A., et al. "Jellyfish: Networking data centers randomly." *NSDI*. 2012.
[2] Camarero, C., et al. "Random folded Clos topologies for datacenter networks." *HPCA*. 2017.

Types of random networks



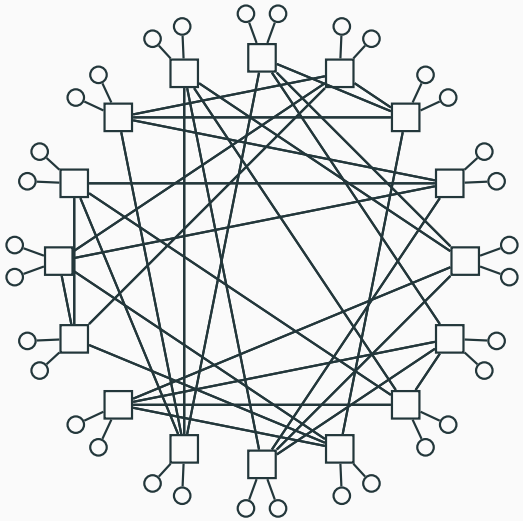
(a) Jellyfish network (RRG)



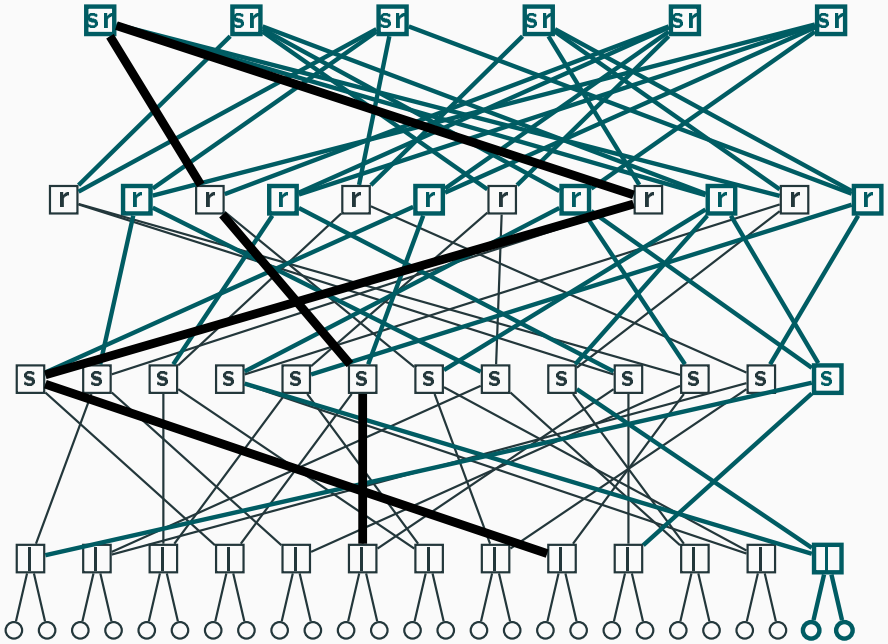
(b) Random Folded Clos (RFC)

[1] Singla, A., et al. "Jellyfish: Networking data centers randomly." *NSDI*. 2012.
[2] Camarero, C., et al. "Random folded Clos topologies for datacenter networks." *HPCA*. 2017.

Types of random networks



(a) Jellyfish network (RRG)

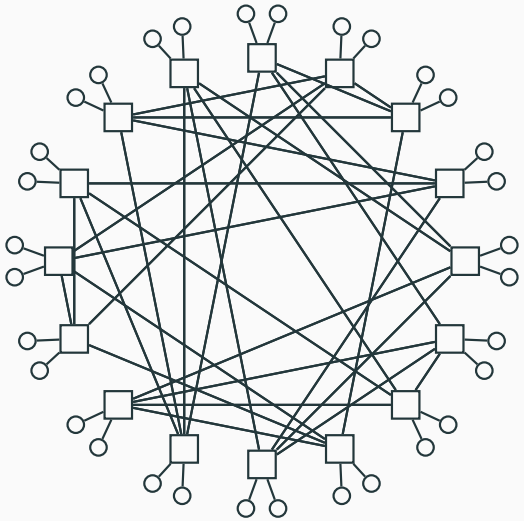


(b) Random Folded Clos (RFC)

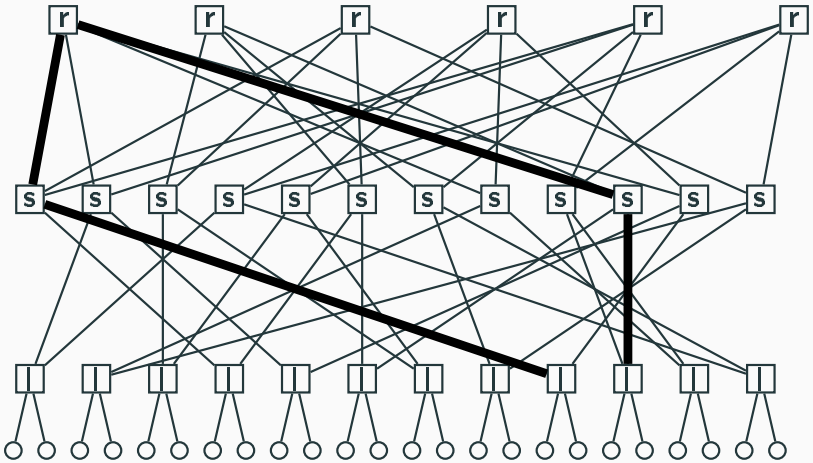
[1] Singla, A., et al. "Jellyfish: Networking data centers randomly." *NSDI*. 2012.
[2] Camarero, C., et al. "Random folded Clos topologies for datacenter networks." *HPCA*. 2017.

Types of random networks

Routing and **layout** are challenging.



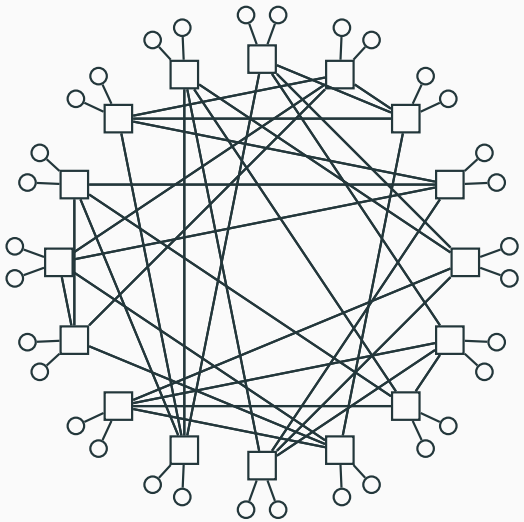
(a) Jellyfish network (RRG)



(b) Random Folded Clos (RFC)

[1] Singla, A., et al. "Jellyfish: Networking data centers randomly." *NSDI*. 2012.
[2] Camarero, C., et al. "Random folded Clos topologies for datacenter networks." *HPCA*. 2017.

Routing and Layout in a Jellyfish



Polarized routing was proposed for Random Regular Graphs:

- Medium complexity of implementation.
- Bound of $4D - 3$ hops.

Amazon proposed a way to deploy **a subset** of Random Regular graphs based on shuffleboxes.

---

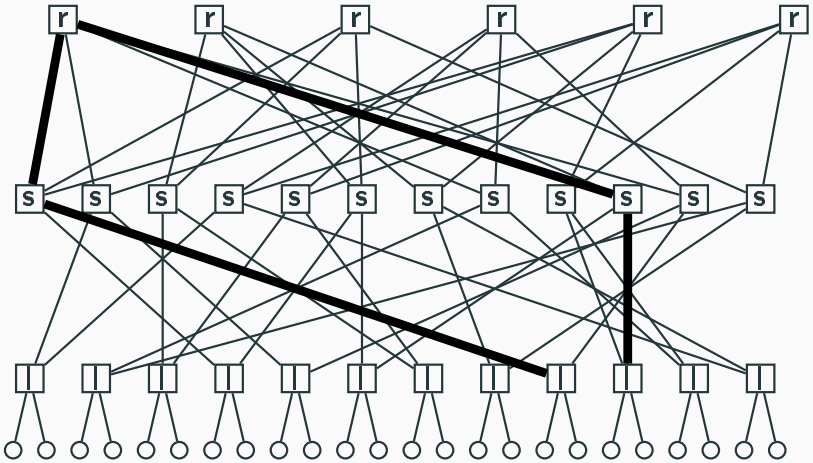
[1] C. Camarero, et al. "Polarized routing: an efficient and versatile algorithm for large direct networks" HOTI 2021
[2] Bernardi, Giacomo, et al. "RNG: Flat Datacenter Networks at Scale." *arXiv preprint 2026*.

UC UNIVERSIDAD DE CANTABRIA A+C Computer Architecture and Technology

Routing and Layout in a Random Folded Clos

Up/Down routing exists in these networks.

The layout is **easier** than in a completely unstructured network.



[1] Camarero, C., et al. "Random folded Clos topologies for datacenter networks." *HPCA*. 2017.

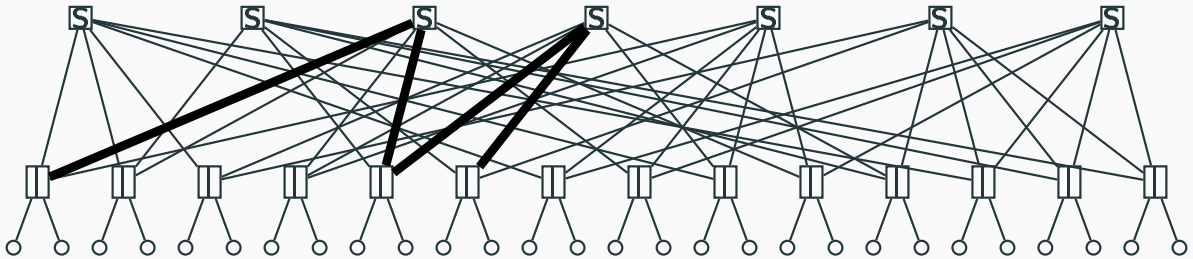
Summary

	Efficiency	Routing	Layout	Expandability
Jellyfish	✓	×	×	✓
RFC	✓	✓	✓	×

Our proposal will take **the best of both worlds**.

Multipass Random Leaf Spine networks

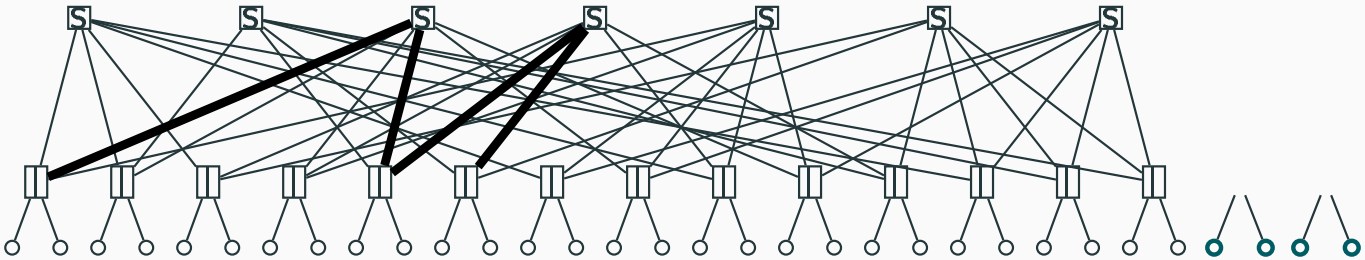
MRLS network



MRLS network

- Two levels of switches randomly connected.
- **Multipass connectivity**, no Up-Down.
- Unlimited expansion.

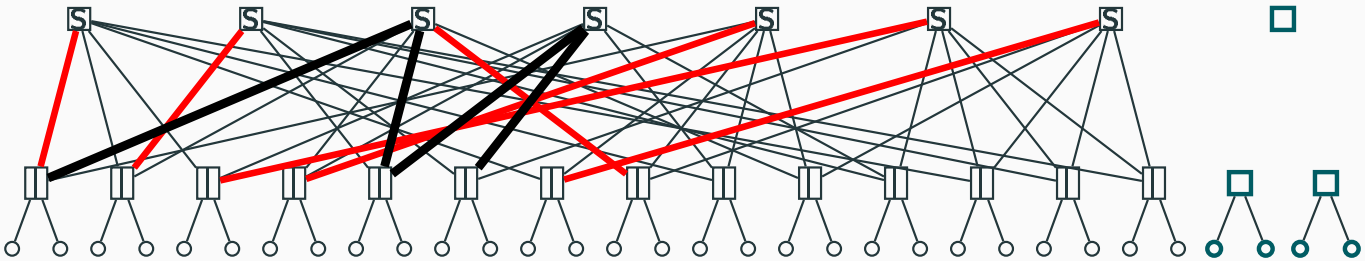
MRLS network



MRLS network

- Two levels of switches randomly connected.
- **Multipass connectivity**, no Up-Down.
- Unlimited expansion.

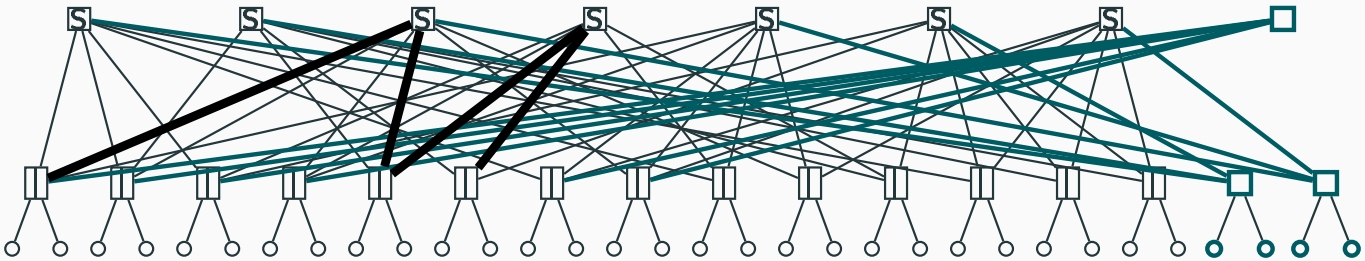
MRLS network



MRLS network

- Two levels of switches randomly connected.
- **Multipass connectivity**, no Up-Down.
- Unlimited expansion.

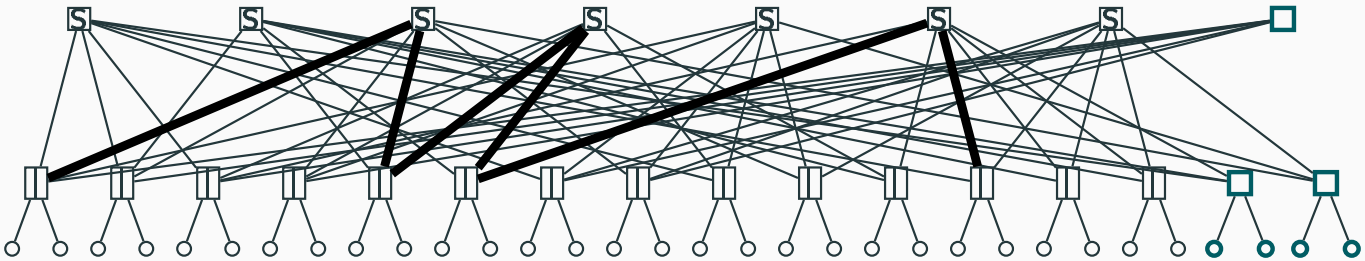
MRLS network



MRLS network

- Two levels of switches randomly connected.
- **Multipass connectivity**, no Up-Down.
- Unlimited expansion.

MRLS network

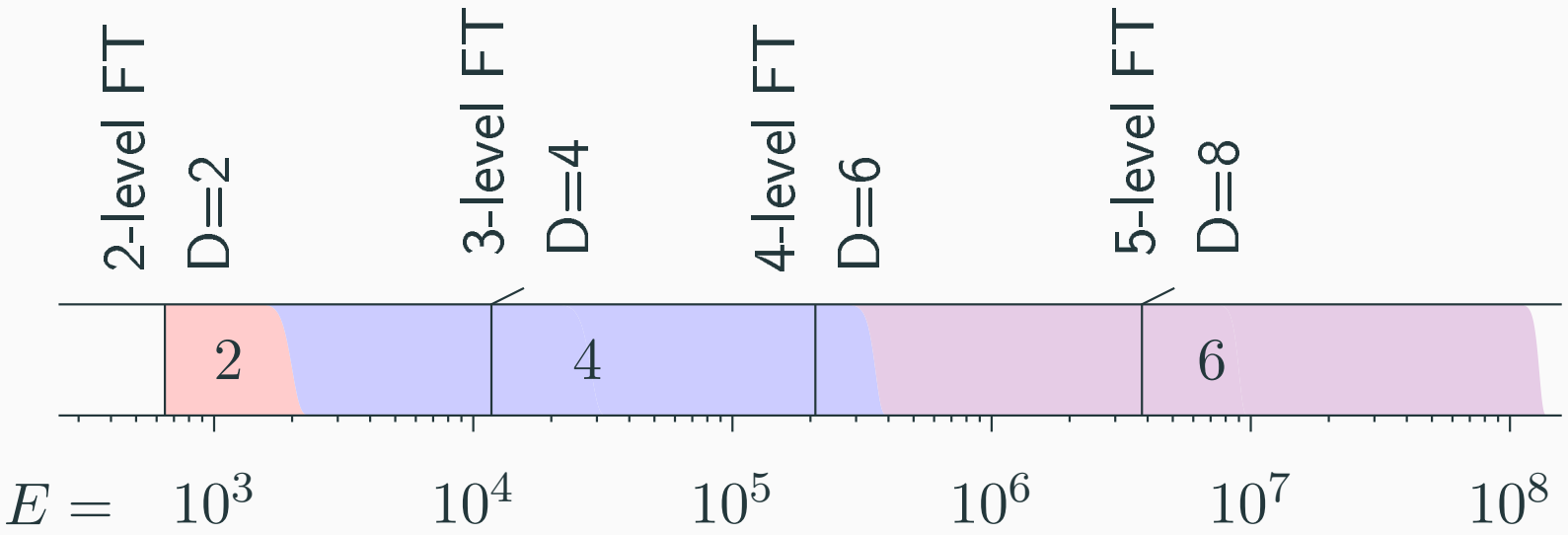


MRLS network

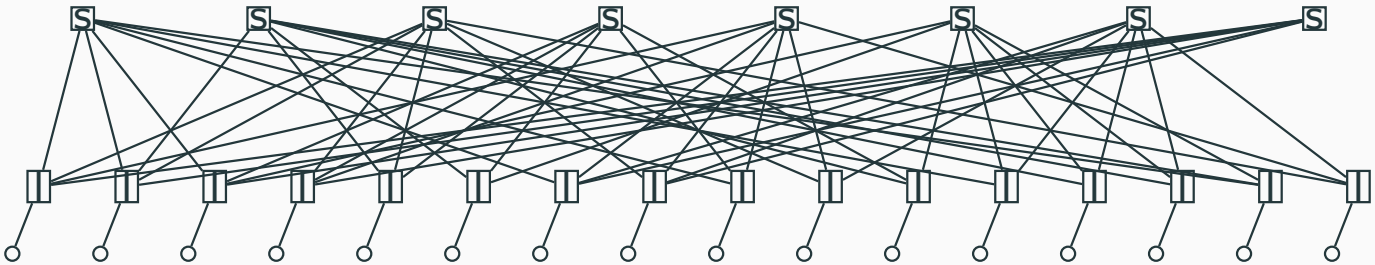
- Two levels of switches randomly connected.
- **Multipass connectivity**, no Up-Down.
- Unlimited expansion.

MRLS network expansion evolution

Distance evolution with switches of 32 ports:



MRLS adjusting cost and performance



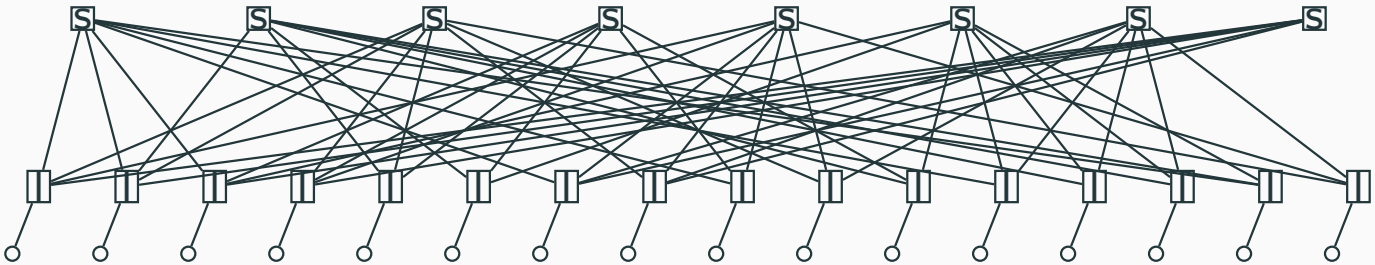
Cost Definition

$$\text{Cost} = \frac{\text{Total Links}}{\text{Total Servers}} = \frac{\text{Up}}{\text{Down}}$$

Balanced performance

$$\frac{\text{Avg}_{dist}}{2} \leq \text{Cost}$$

MRLS adjusting cost and performance



Cost Definition

$$\text{Cost} = \frac{\text{Total Links}}{\text{Total Servers}} = \frac{\text{Up}}{\text{Down}}$$

Balanced performance

$$\frac{\text{Avg}_{dist}}{2} \leq \text{Cost}$$

To compare with **other topologies**, a fixed cost (or fitness ratio) can be selected and then the topology is built.

Multipass routing in MRLS Networks

Polarized routing is adapted to multipass leaf-spine networks.

	Bound	Implementation	M & NM Paths
Jellyfish	× $4D-3$	× Complex	✓ Yes
MRLS (Ours)	✓ $2D$	✓ Simple	✓ Yes

Polarized Routing in MRLS Networks

Each packet in a switch has a state represented by the pair:

$$(\text{Distance to source}, \text{Distance to destination})$$

- **Routing Sequence:** From src to dst, the state evolves as:

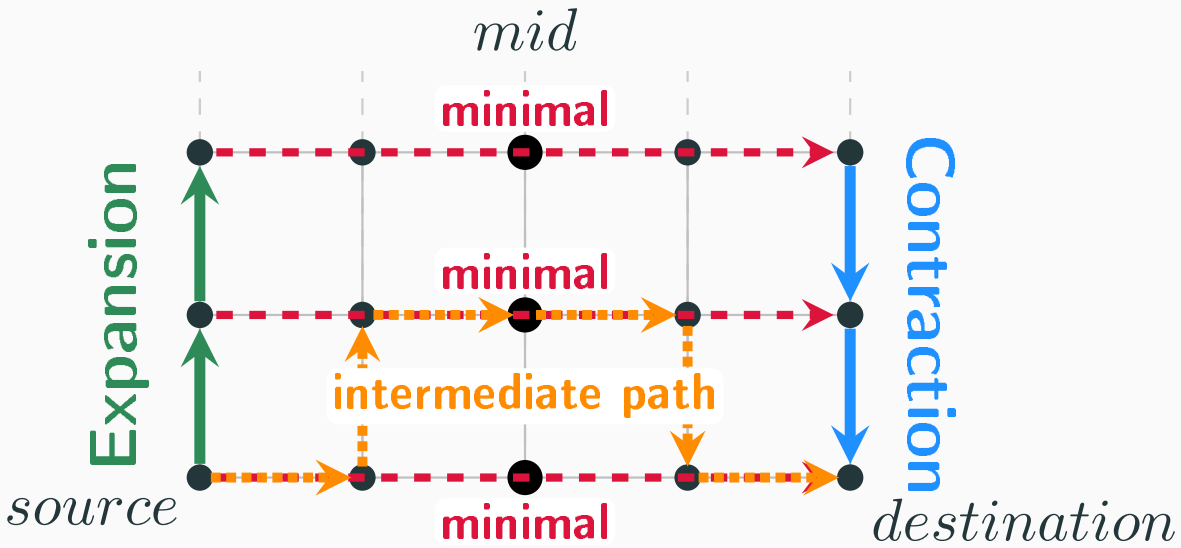
$$(0, x) \xrightarrow{\cdots} (a, b) \xrightarrow{\cdots} (x, 0)$$

- **Hop Transitions:** At each hop, the state updates by:

$$(+1, +1), \quad (-1, -1), \quad (+1, -1), \quad \text{or} \quad (-1, +1)$$

Polarized routing in MRLS Networks

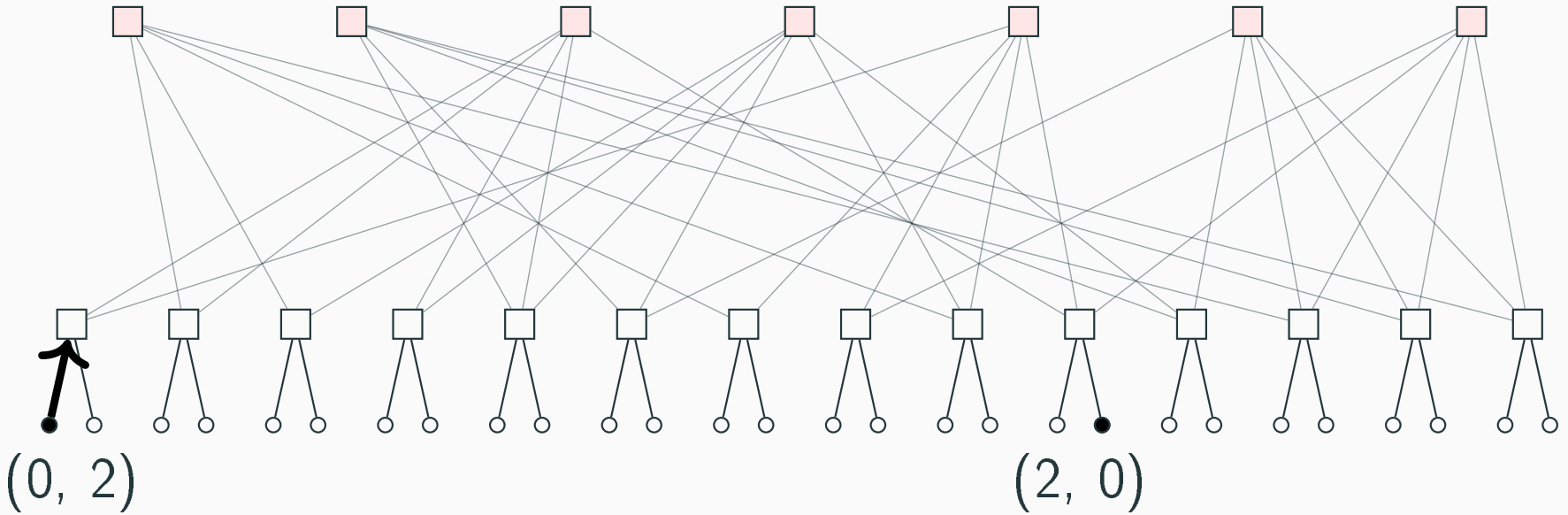
- Minimal (+1, -1): moves away from src and closer to dst.
- Expansion (+1, +1): moves away from **both** src and dst.
- Contraction (-1, -1): moves closer to **both** src and dst.
- Backtrack (-1, +1): moves closer to src and away from dst.



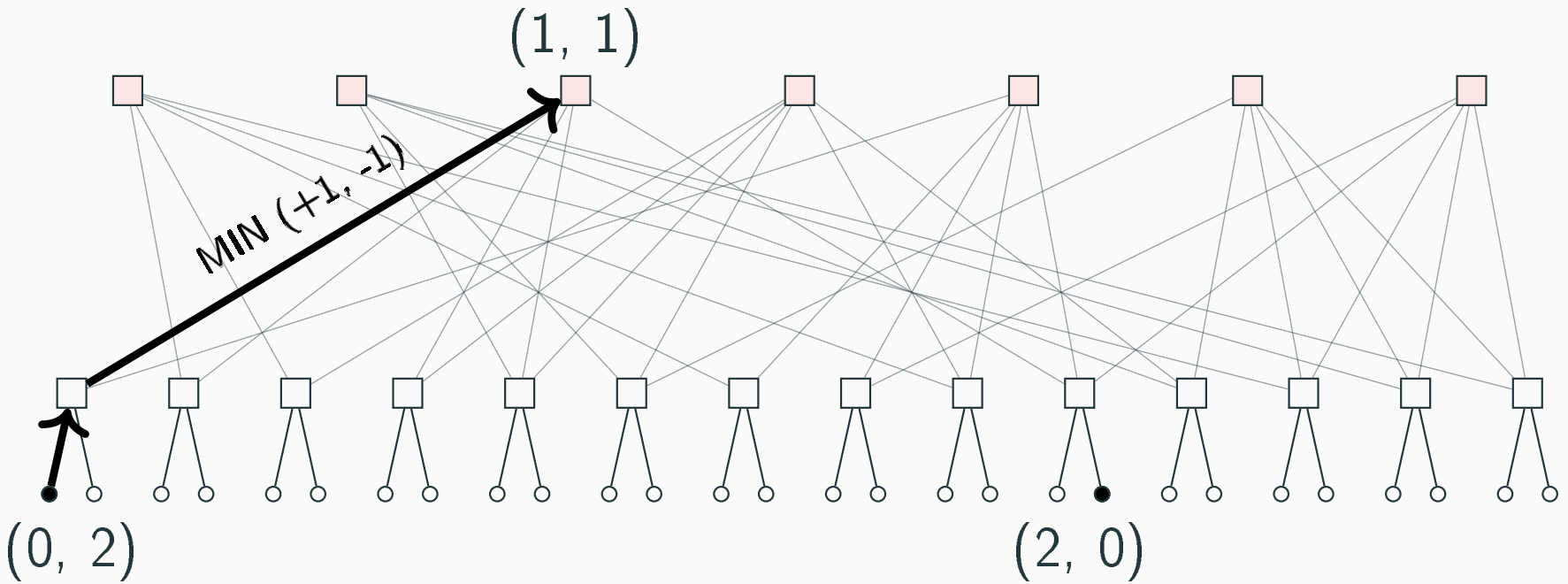
Polarized Routing

Step Type	Change	Condition
Minimal	(+1, -1)	**Always allowed**: moves away from src and closer to dst.
Expansion	(+1, +1)	**Allowed** only when closer to the **source** than the destination.
Contraction	(-1, -1)	**Allowed** only when closer or equal to the **destination** than the source.
Backtrack	(-1, +1)	**Never allowed**: moves closer to src and away from dst.

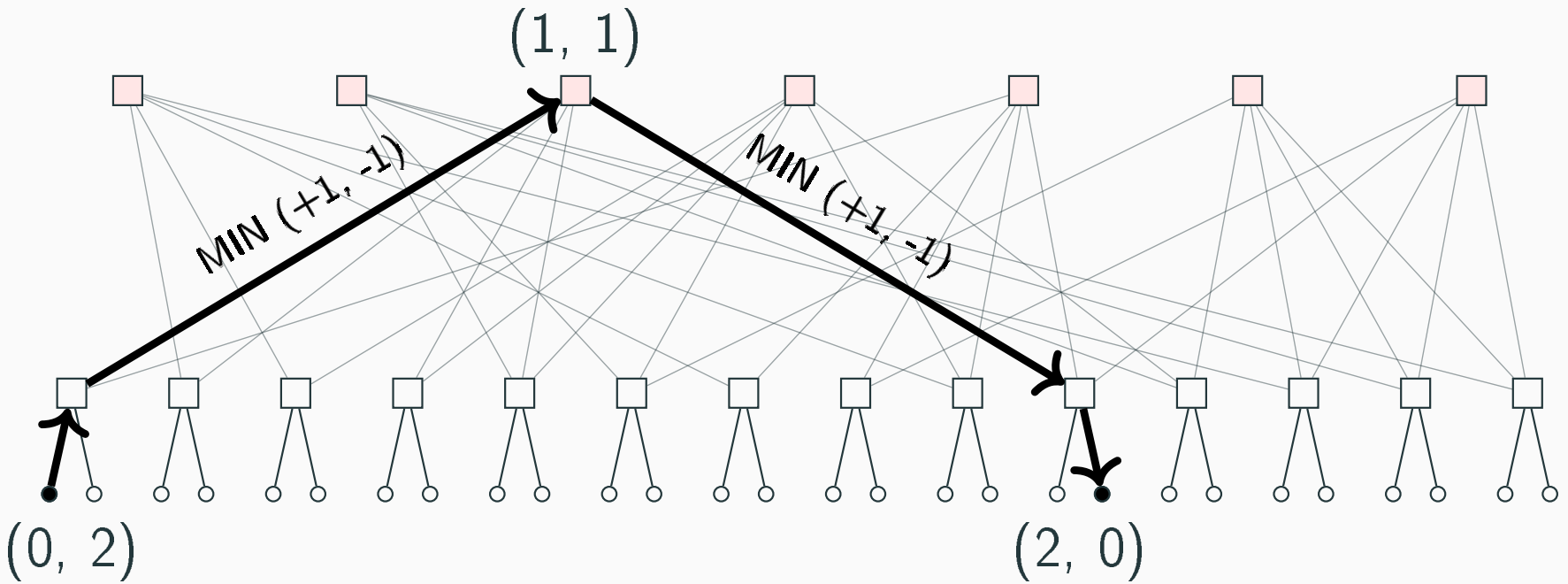
Polarized Routing (Minimal Path)



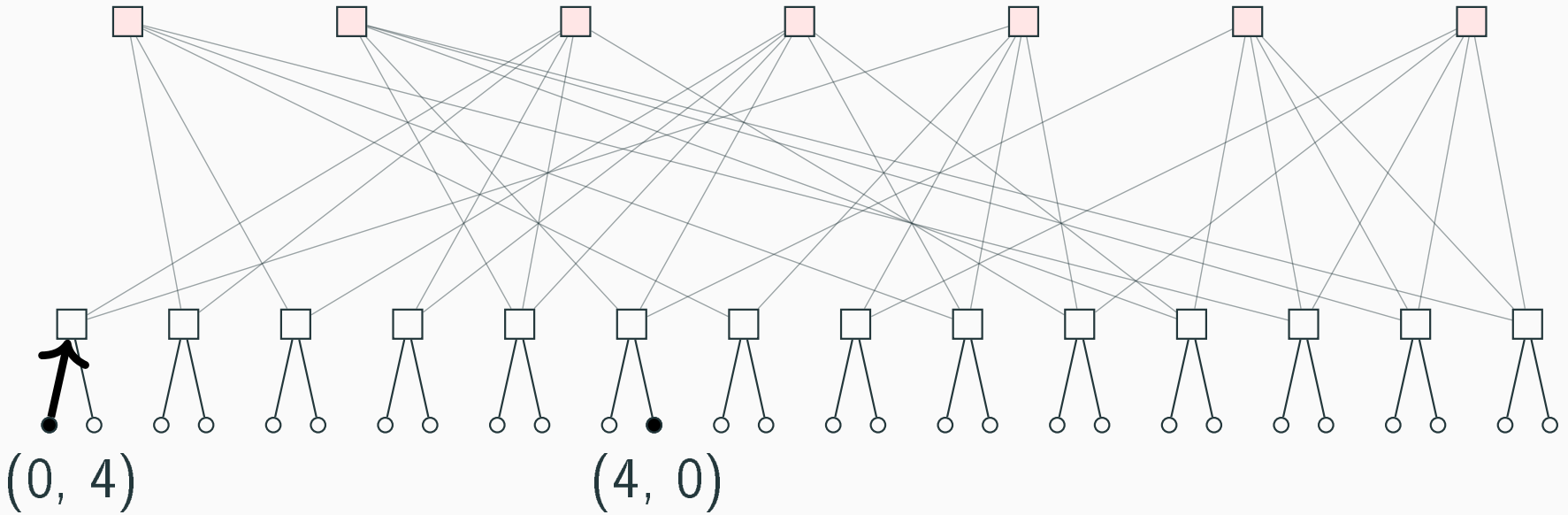
Polarized Routing (Minimal Path)



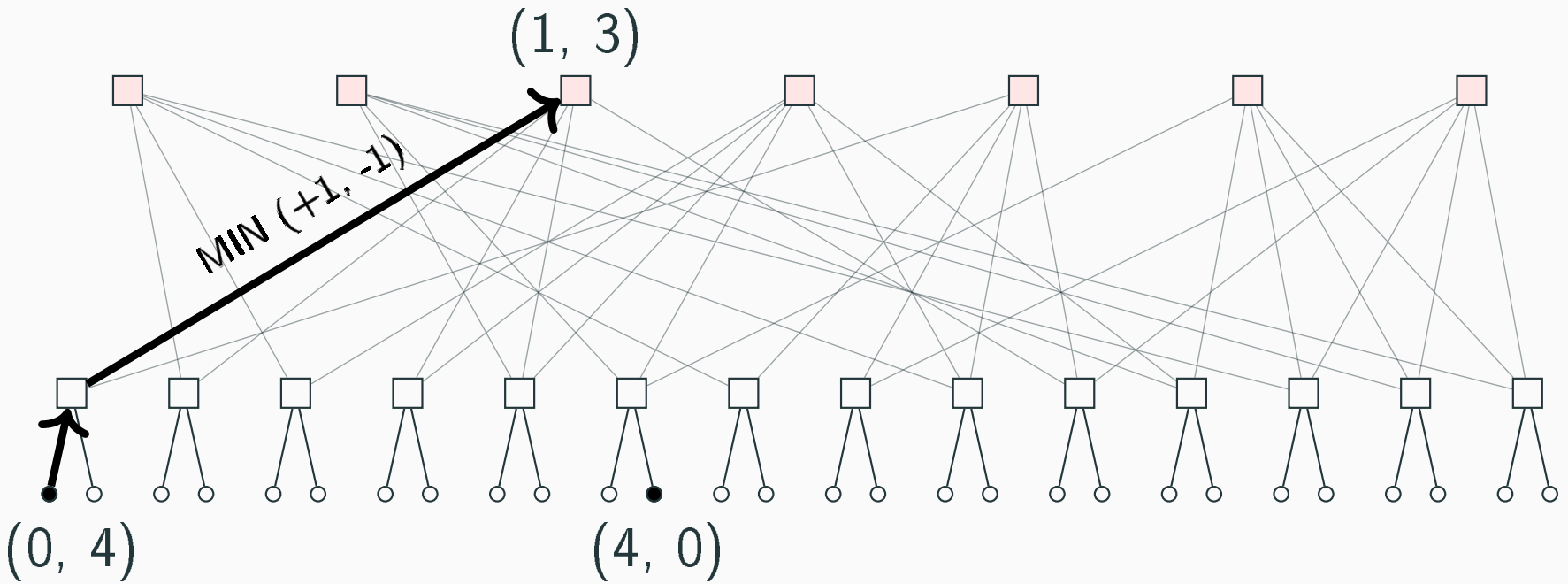
Polarized Routing (Minimal Path)



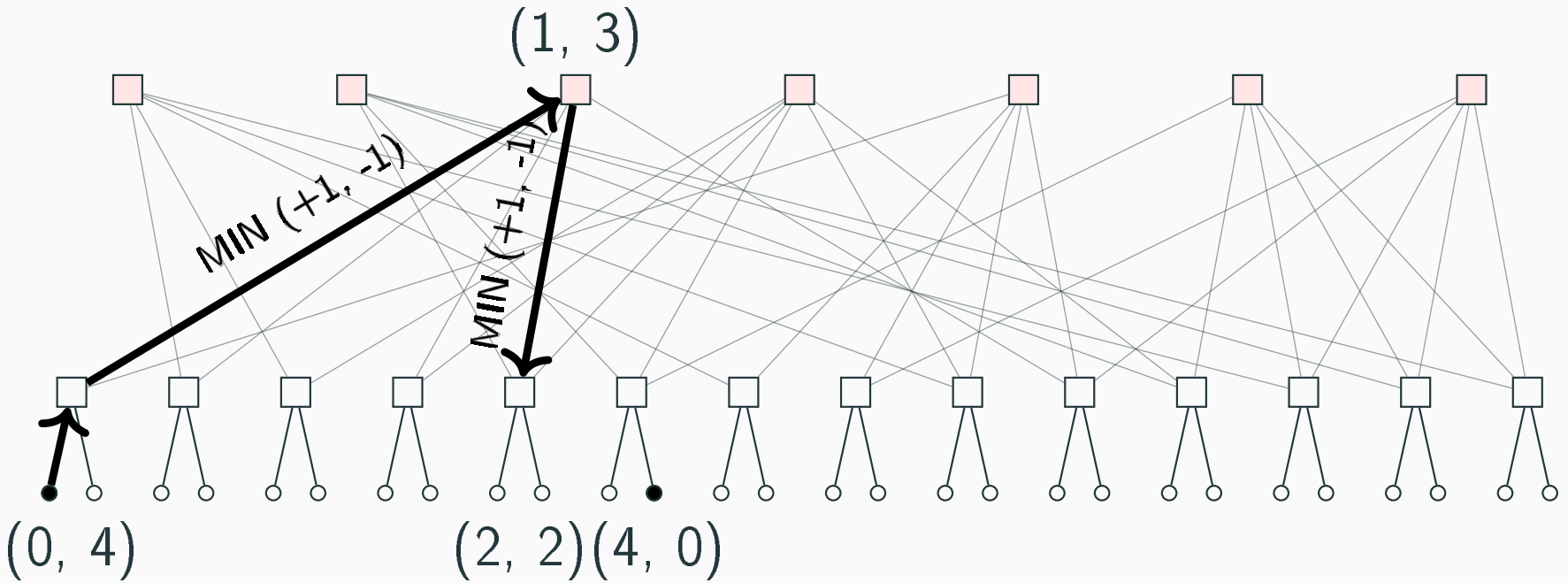
Polarized routing (Minimal Path)



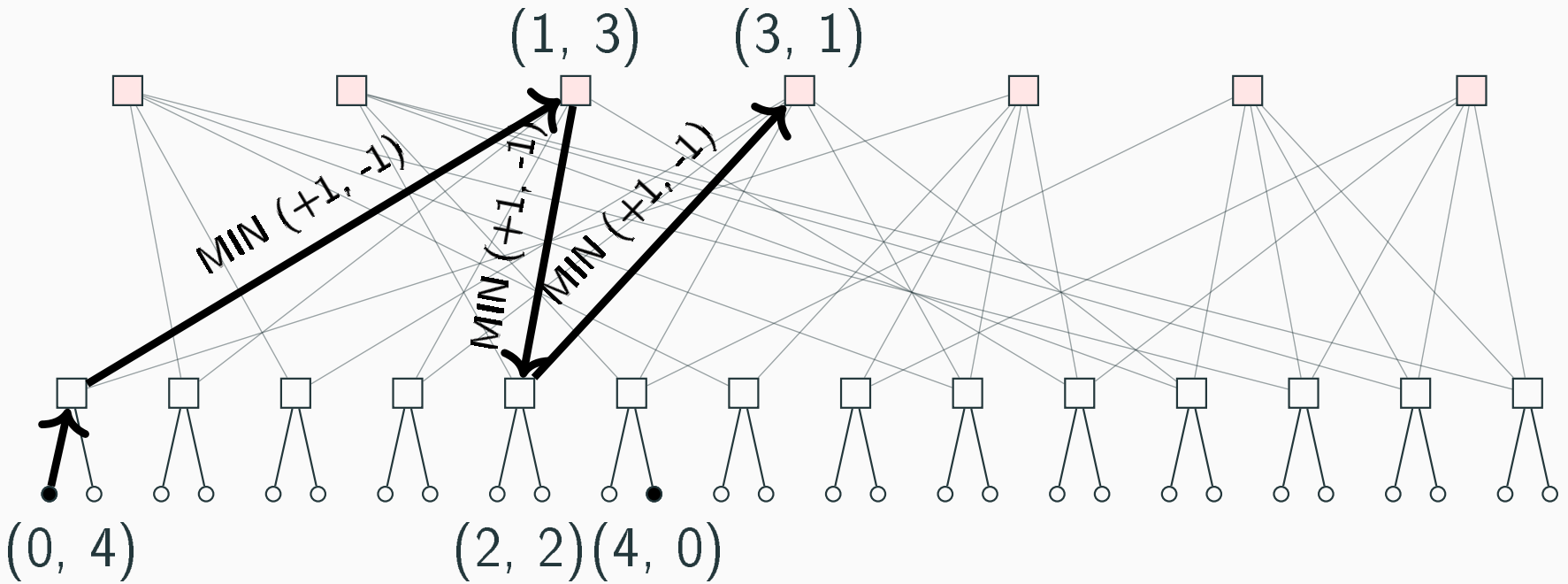
Polarized routing (Minimal Path)



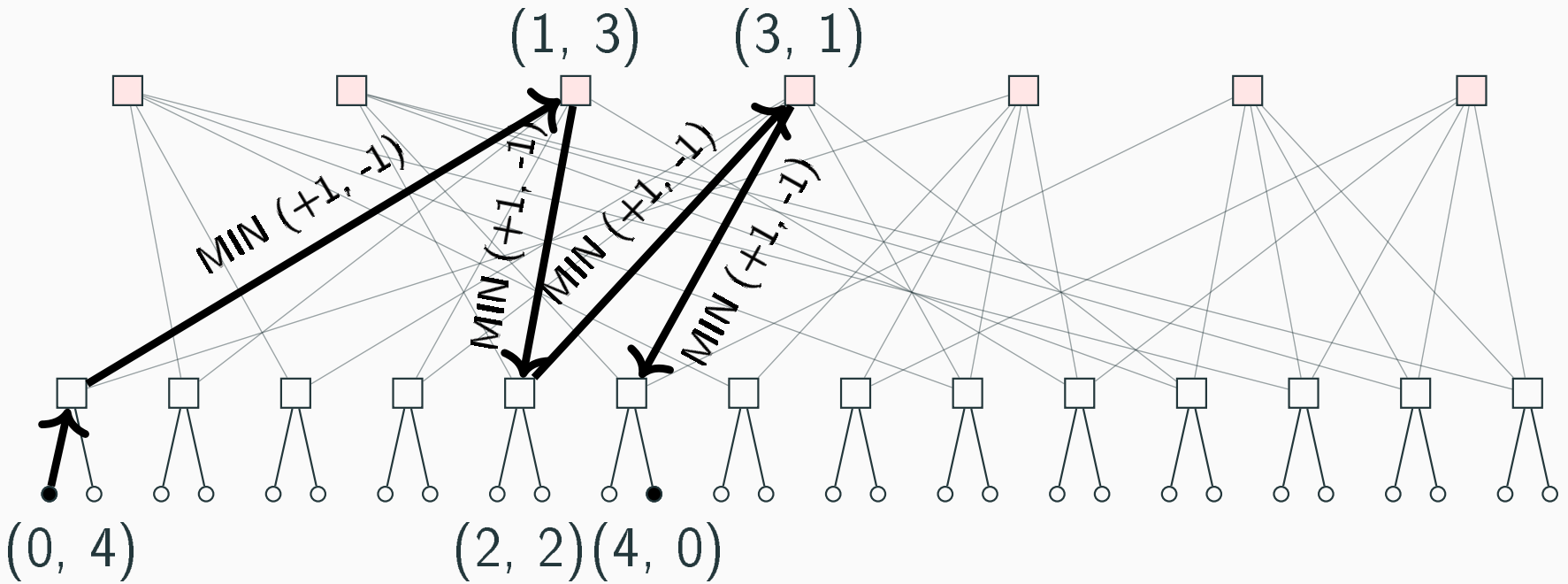
Polarized routing (Minimal Path)



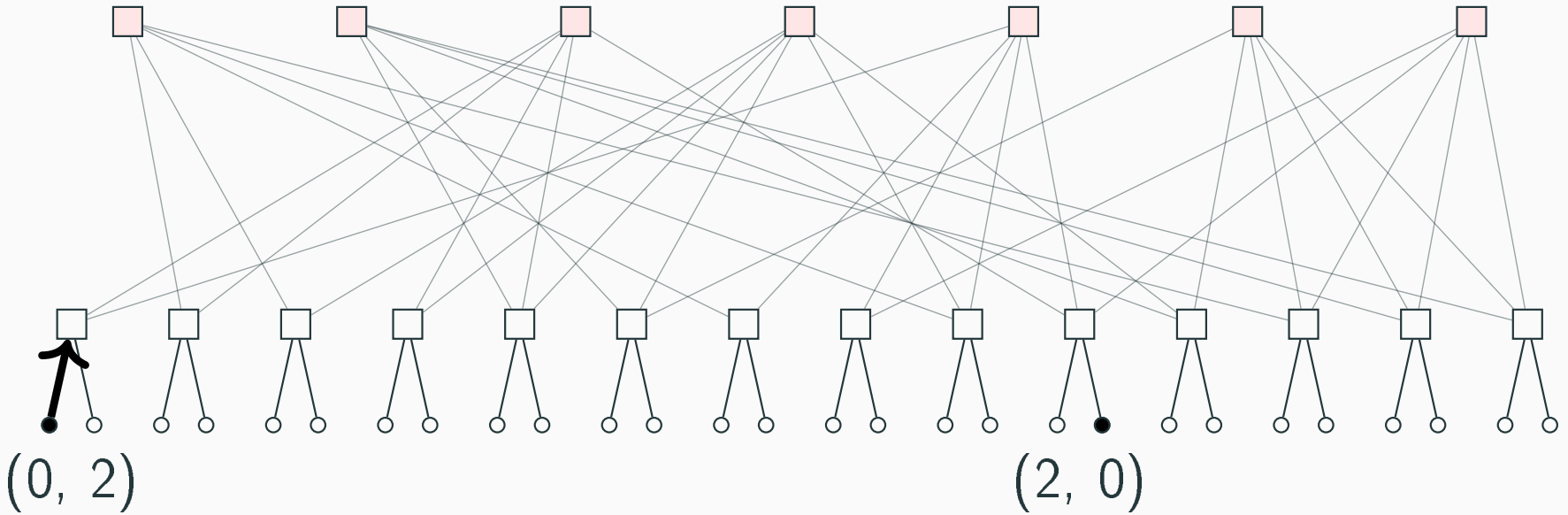
Polarized routing (Minimal Path)



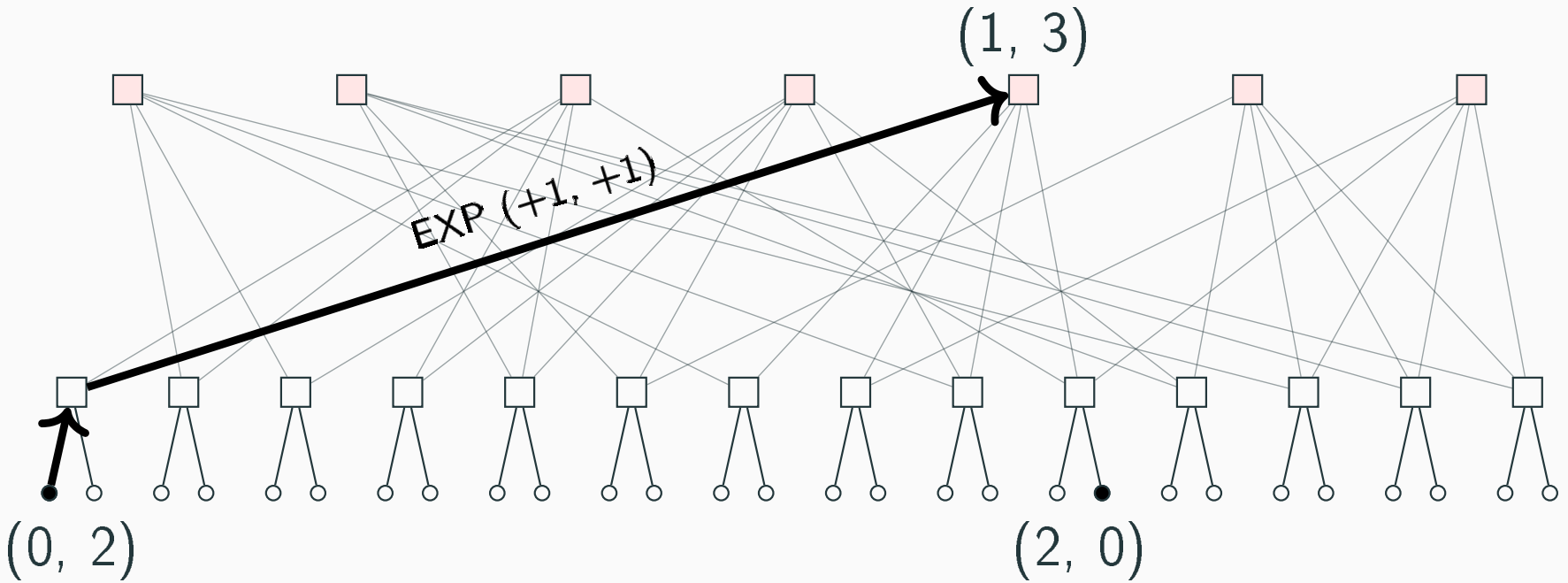
Polarized routing (Minimal Path)



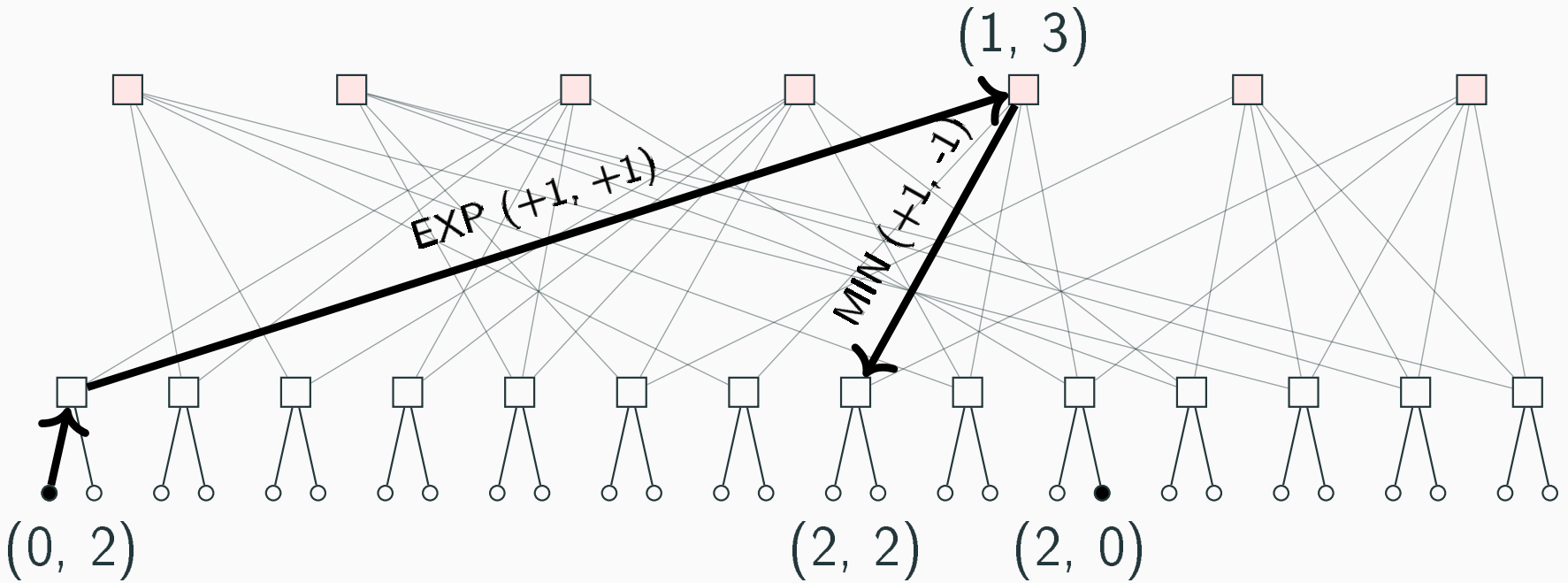
Polarized Routing (Non-Minimal Path)



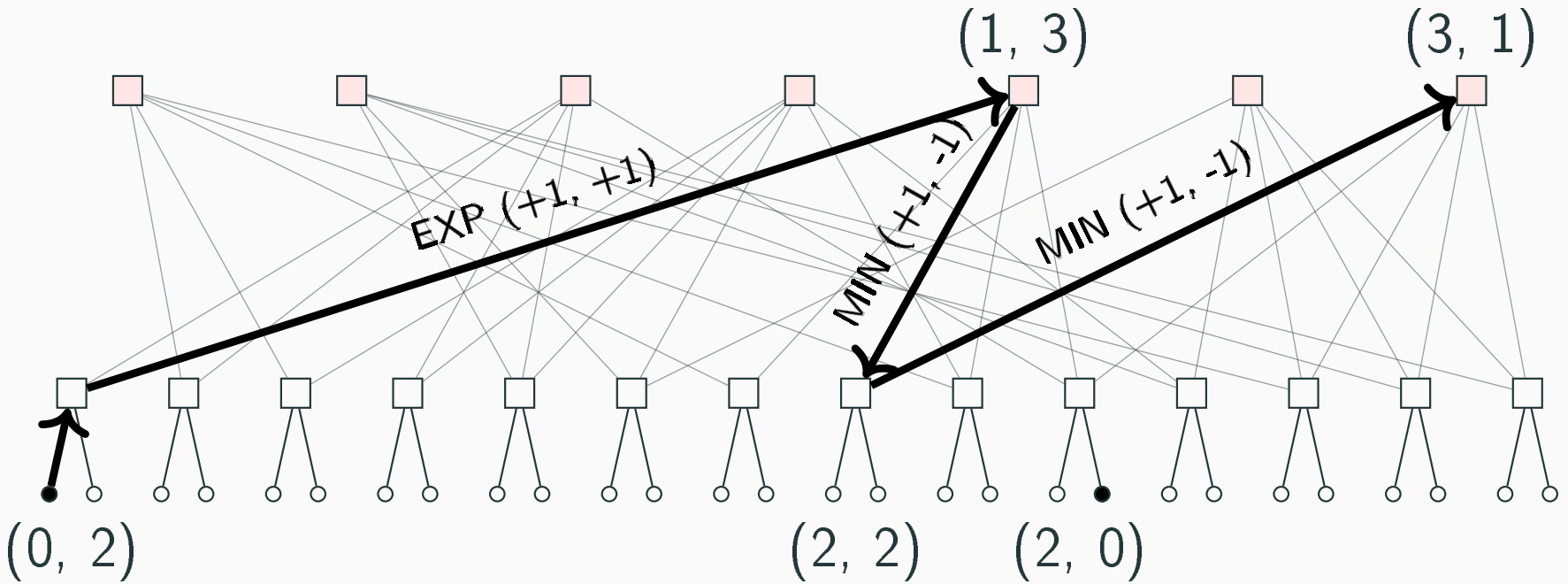
Polarized Routing (Non-Minimal Path)



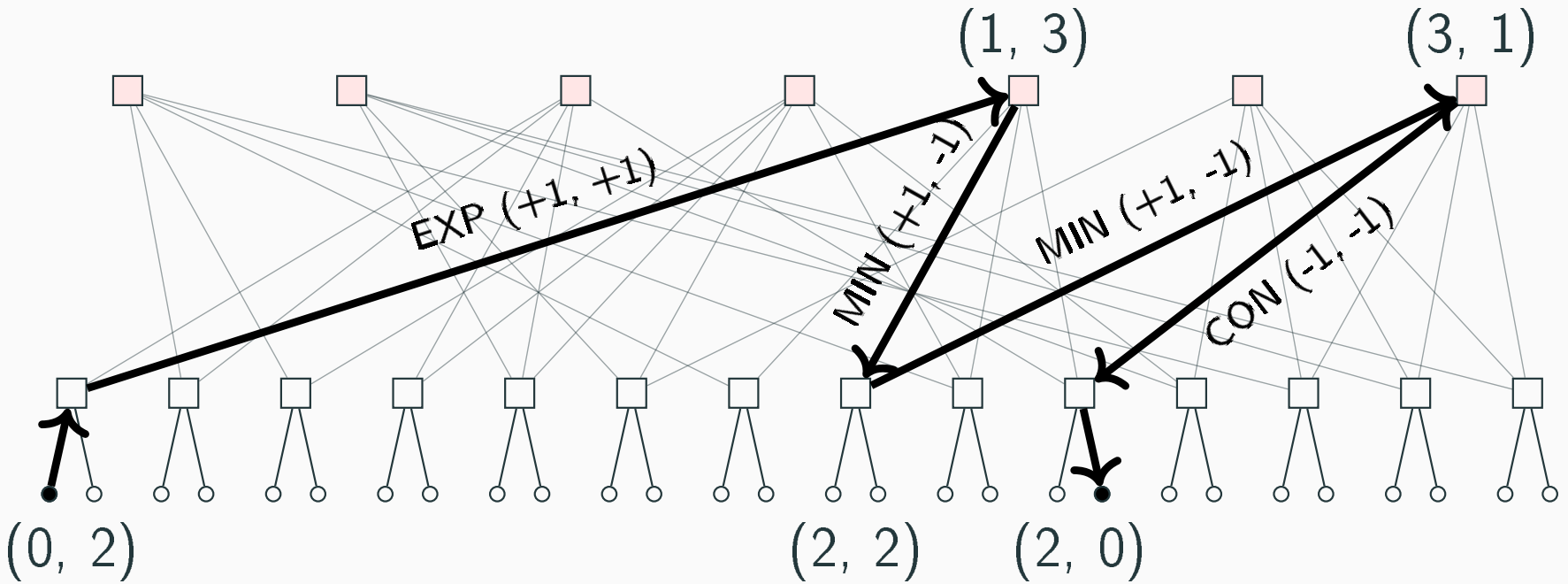
Polarized Routing (Non-Minimal Path)



Polarized Routing (Non-Minimal Path)



Polarized Routing (Non-Minimal Path)



Polarized in Hardware

1. MIN route table is queried **twice**: min ports to src and dst.
2. Assign a 2-bit label (MIN, EXP, CON, BT) to each port.
3. Discard options: packet has a field with (src_d, dst_d).
4. The next port is selected: occupancy and penalties.

Destination \ Port	P0	P1	P2	P3
Node A	1	0	0	1
Node B	0	0	1	1

Table 1: Route table: 1 min ports, 0 non-min ports.

Summary

	Efficiency	Routing	Layout	Expandability
Jellyfish	✓	×	×	✓
RFC	✓	✓	✓	×
MRLS	✓	✓	✓	✓

Our proposal take **the best of both worlds**.

Methodology and Results

Methodology

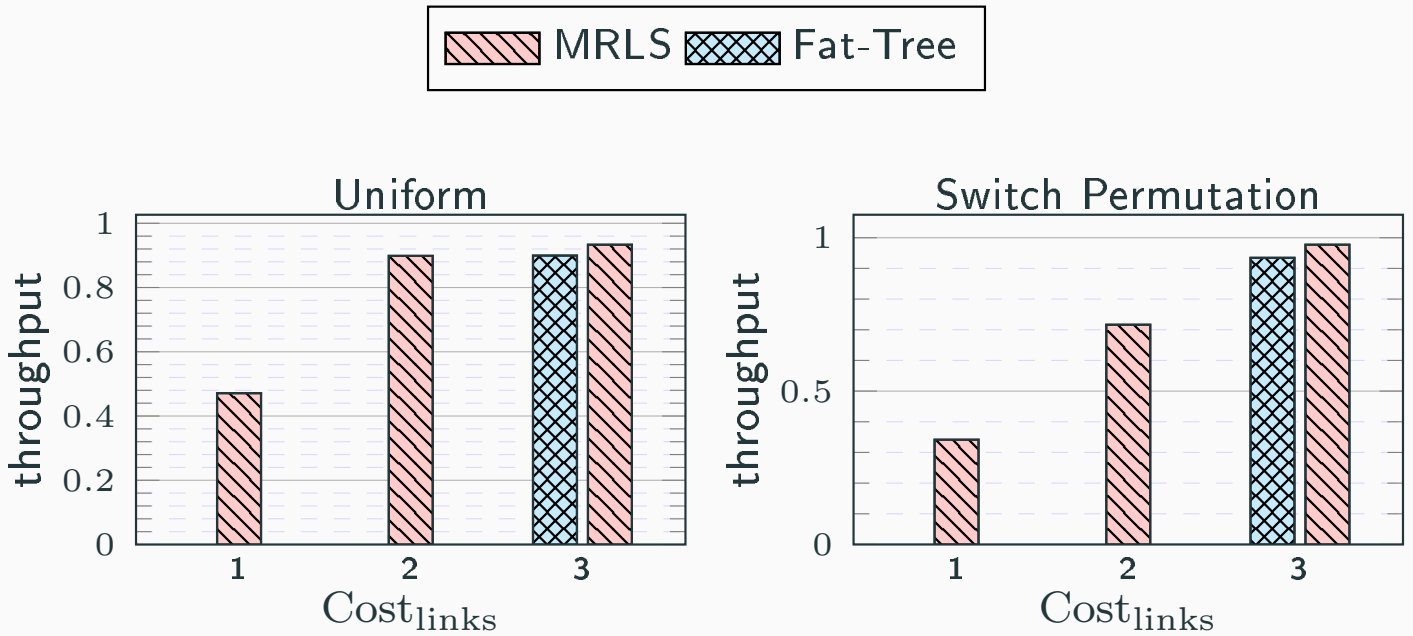
Evaluated in CAMINOS network simulator

- Cycle-accurate simulations with 16k and 100k endpoints.
- Compared MRLS with Fat-Tree and Dragonfly topologies.
- Evaluated with synthetic traffic (patterns and MPI collectives).

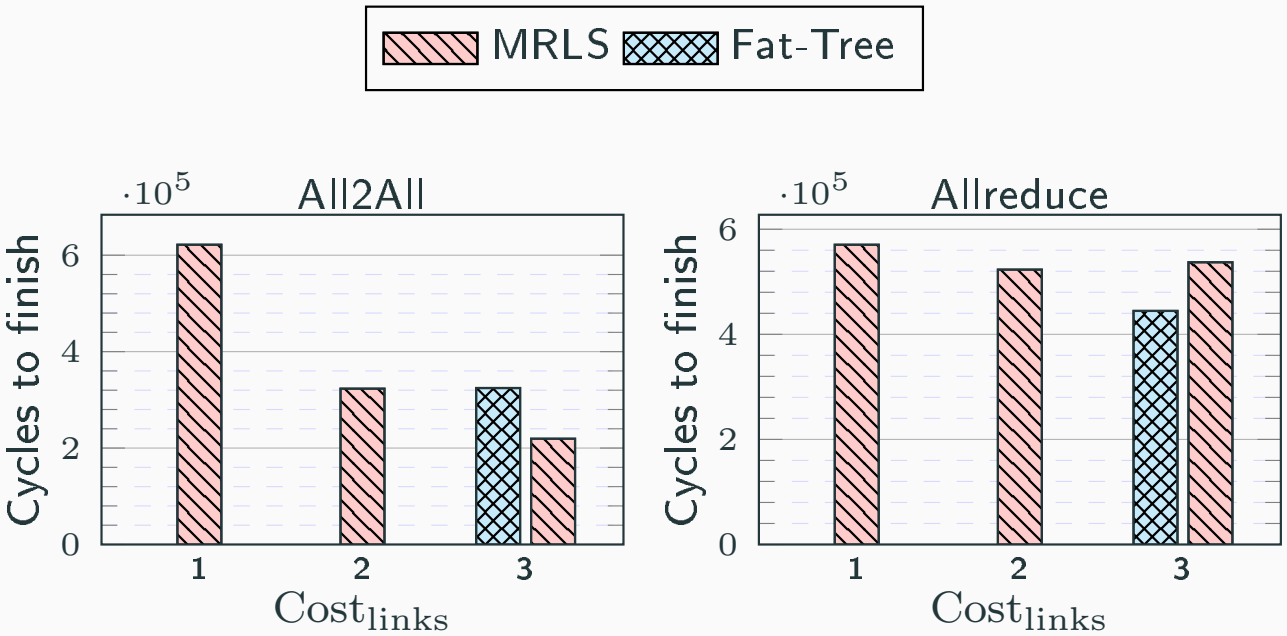
Code for reproducibility:

`https://github.com/alexcano98/MRLS-topology-reproducibility.git`

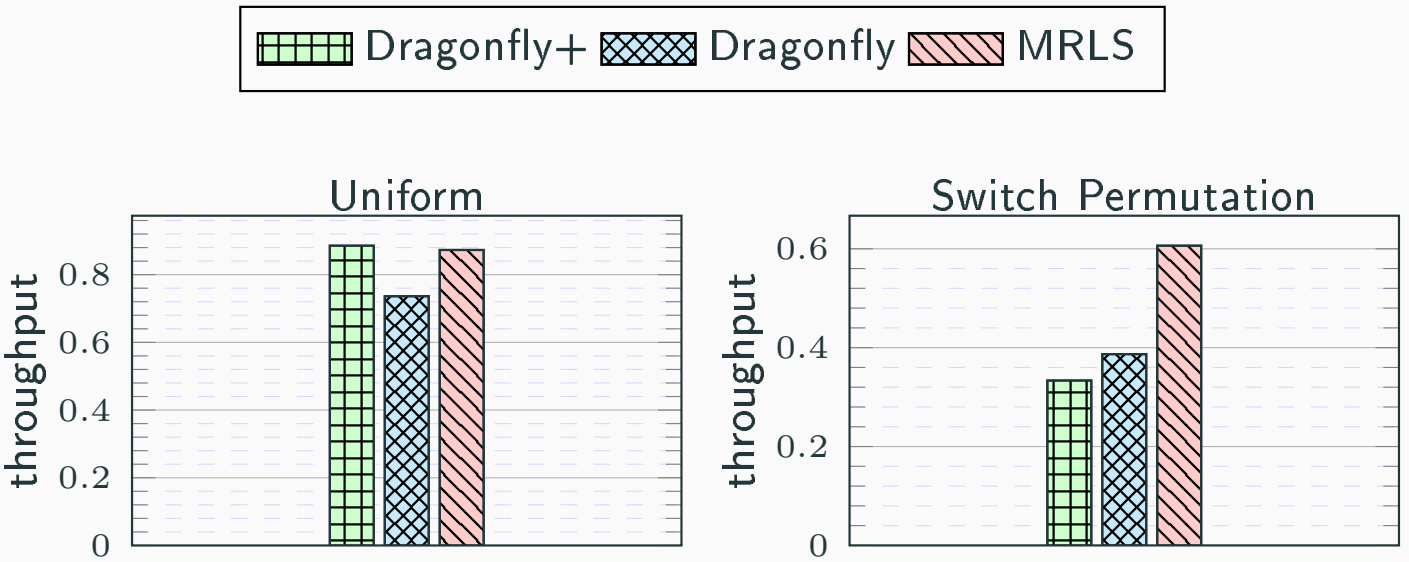
Comparison with 100k HPC networks



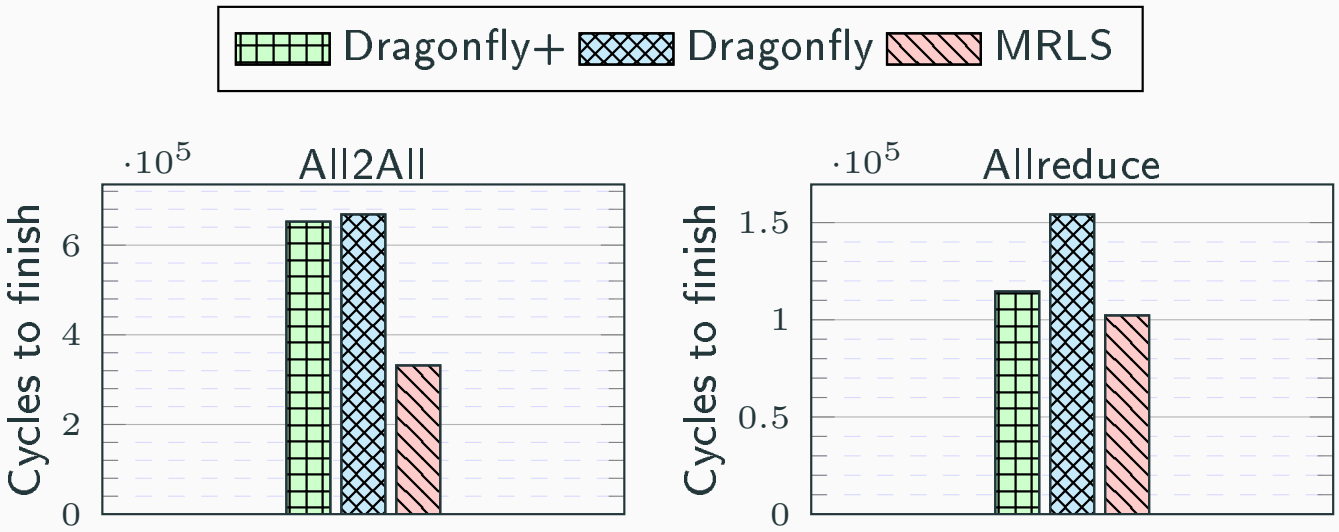
Comparison with 100k HPC networks



Comparison with 16k HPC networks



Comparison with 16k HPC networks



Conclusion

Key Achievements

The **Multipass Random Leaf Spine (MRLS)** architecture delivers:

Cost-Efficient
& High-Performing

Optimized Routing
& Simplified Layout

Unlimited
Expandability

Ideal for large-scale Data Centers and HPC environments

UC UNIVERSIDAD DE CANTABRIA
Computer Architecture and Technology

Thank you! Questions?

Alejandro Cano

alejandro.cano@unican.es

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES
Cofinanciado por la Unión Europea
AGENCIA ESTATAL DE INVESTIGACIÓN

UC UNIVERSIDAD DE CANTABRIA
ATC Computer Architecture and Technology