NILS BLACH*, **MACIEJ BESTA***, DANIELE DE SENSI, JENS DOMKE, HUSSEIN HARAKE, SHIGANG LI, PATRICK IFF, MAREK KONIECZNY, KARTIK LAKHOTIA, ALES KUBICEK, MARCEK FERRARI, FABRIZIO PETRINI, TORSTEN HOEFLER

A High-Performance Design, Implementation, Deployment, and Evaluation of The Slim Fly Network

@ NSDI'24

A High-Performance Design, Implementation, Deployment, and Evaluation of The Slim Fly Network

Nils Blach*, Maciej Besta*, Daniele De Sensi*,◇, Jens Domke†, Hussein Harake§, Shigang Li*, Patrick Iff*, Marek Konieczny¶, Kartik Lakhotia‖, Ales Kubicek*, Marcel Ferrari*, Fabrizio Petrini‖, Torsten Hoefler*

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

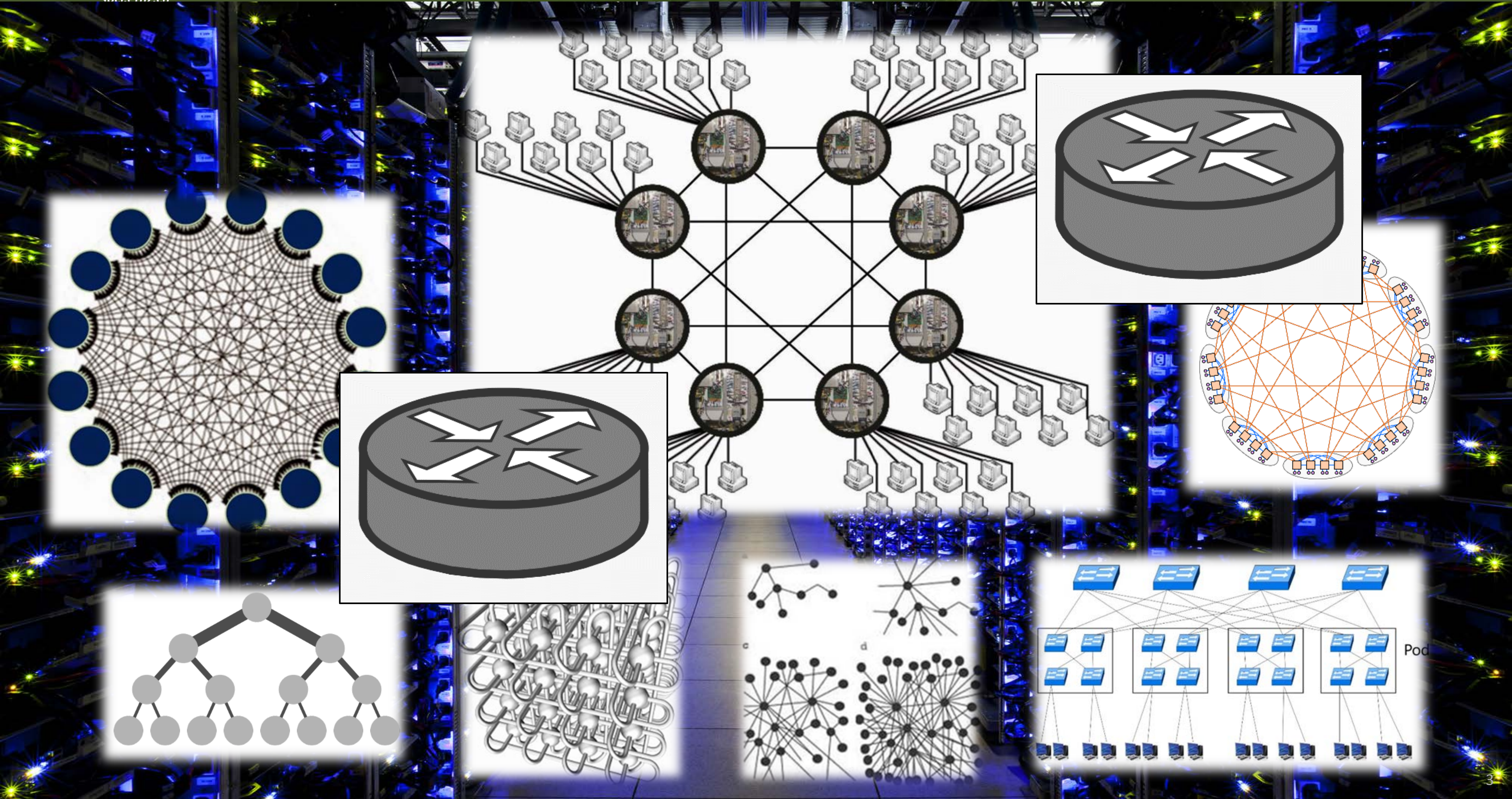
The interconnect: a key part of supercomputers and data centers, relevant both for high performance and low cost

50% [1]

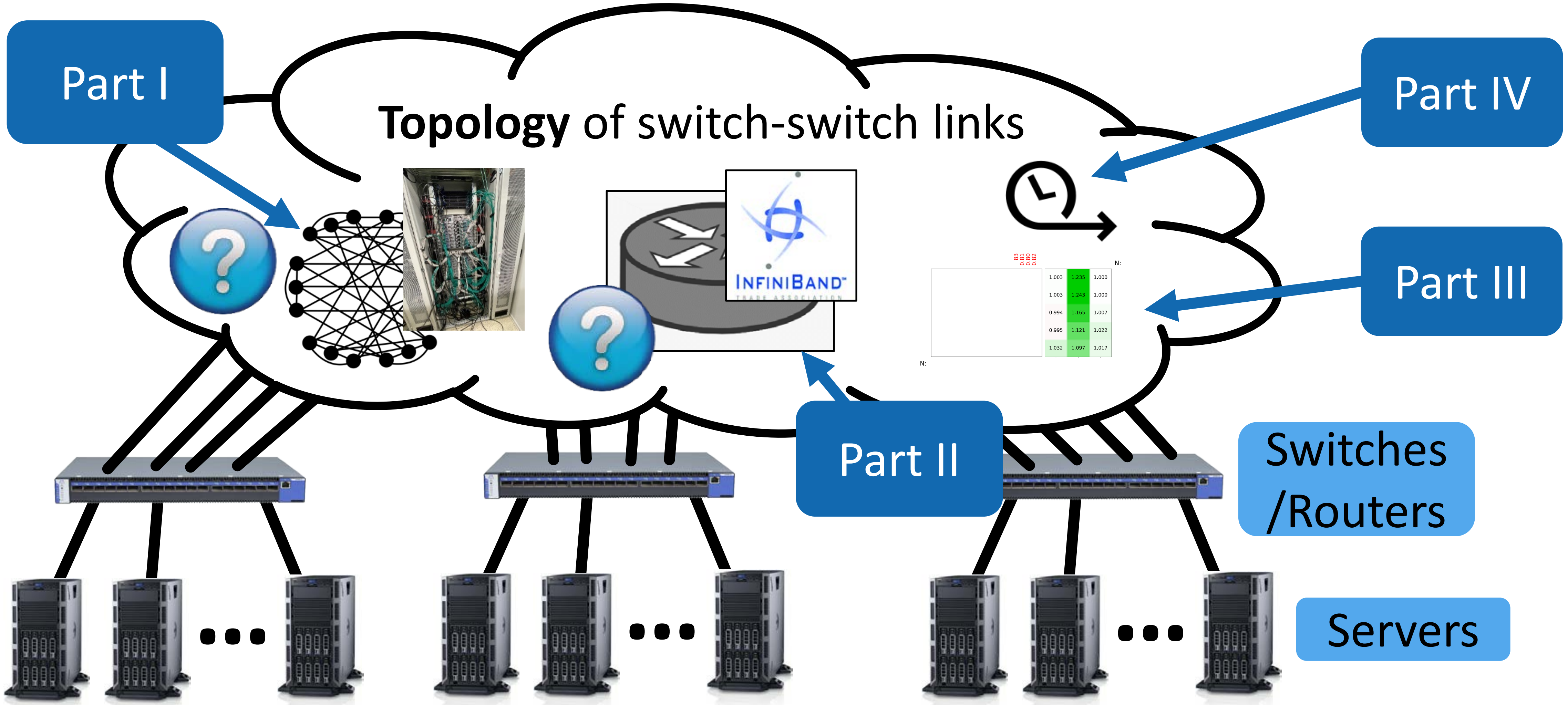
33% [2]

[1] D. Abts et al. (2010), *Energy Proportional Datacenter Networks*, ISCA’10

[2] J. Kim et al. (2007), *Flattened Butterfly: A Cost-Efficient Topology for High-Radix Networks*, ISCA’07


Pod

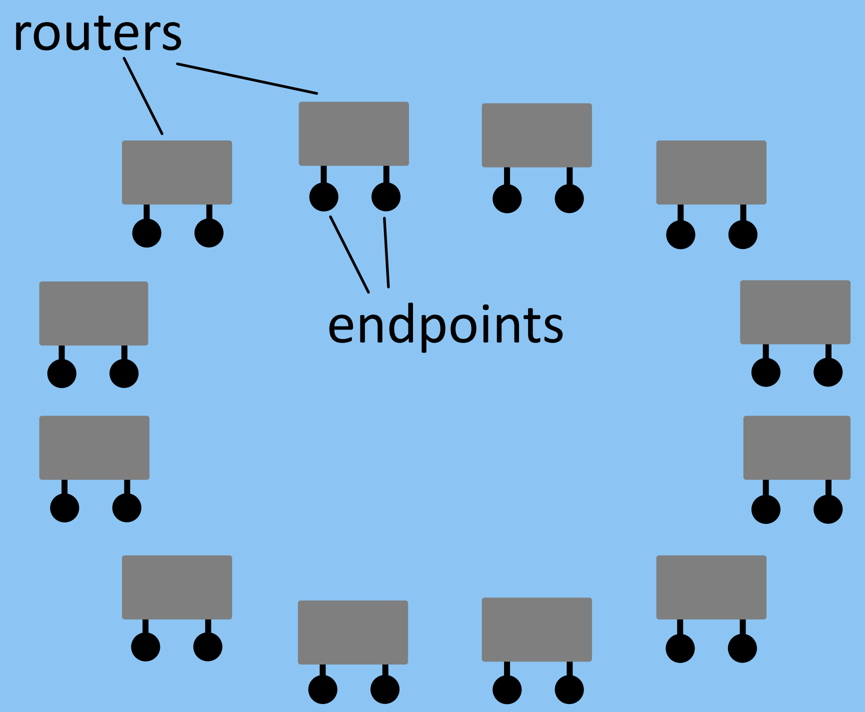
Network Topologies : setting & presentation plan



SLIM FLY: AN EFFICIENT LOW-DIAMETER NETWORK TOPOLOGY [1]

Key idea

Lower diameter and thus average path length: fewer cables and routers necessary.

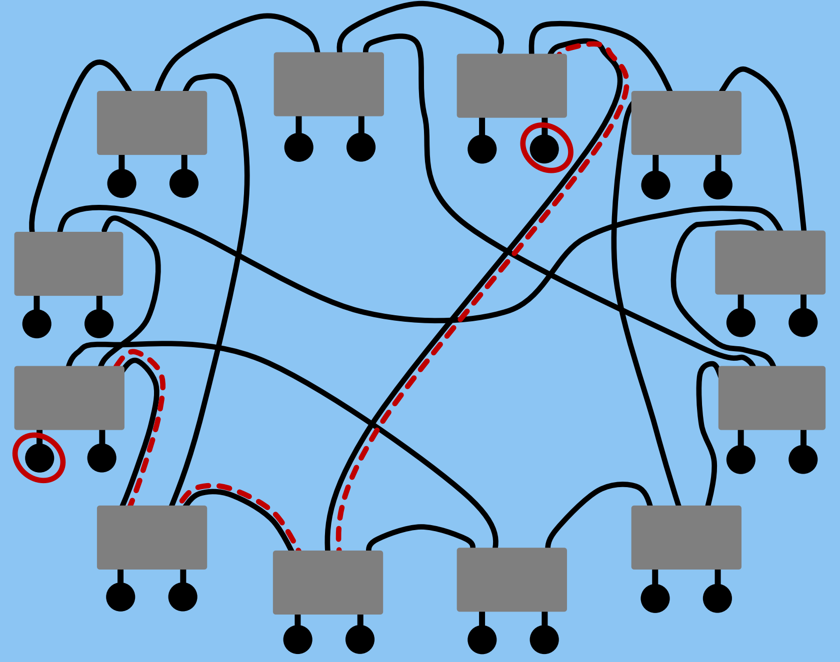


[1] M. Besta, T. Hoefler. *Slim Fly: A Cost-Effective Low-Diameter Network Topology*. ACM/IEEE Supercomputing, 2014. **Best Student Paper Award**

SLIM FLY: AN EFFICIENT LOW-DIAMETER NETWORK TOPOLOGY [1]

Key idea

Lower diameter and thus average path length: fewer cables and routers necessary.



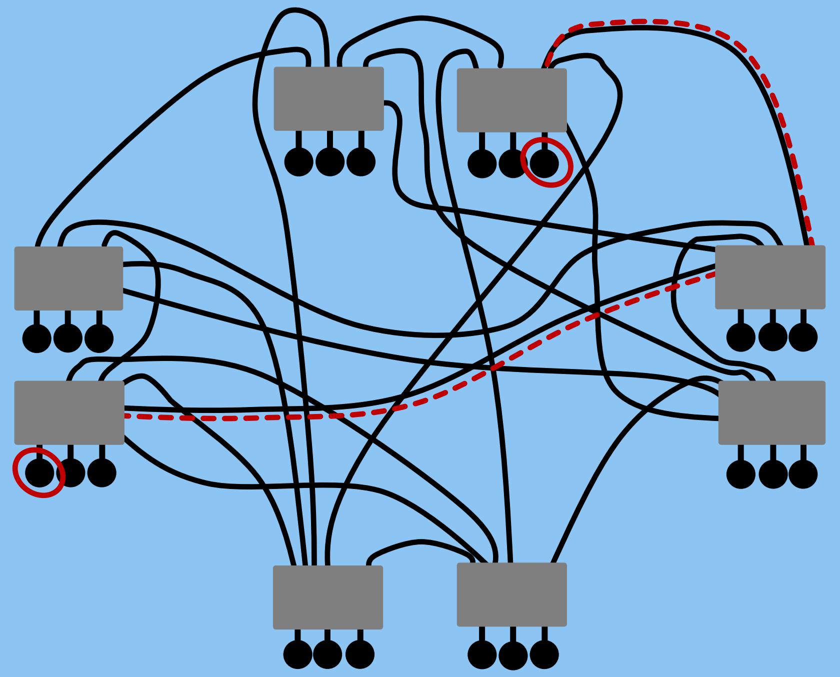
[1] M. Besta, T. Hoefler. *Slim Fly: A Cost-Effective Low-Diameter Network Topology*. ACM/IEEE Supercomputing, 2014. **Best Student Paper Award**

Slim Fly: An Efficient Low-Diameter Network Topology [1]

Lower diameter → more performance, smaller cost, less consumed power

Key idea

Lower diameter and thus average path length: fewer cables and routers necessary.



[1] M. Besta, T. Hoefler. *Slim Fly: A Cost-Effective Low-Diameter Network Topology*. ACM/IEEE Supercomputing, 2014. **Best Student Paper Award**

SLIM FLY: AN EFFICIENT LOW-DIAMETER NETWORK TOPOLOGY

Fix diameter (e.g., D = 2)

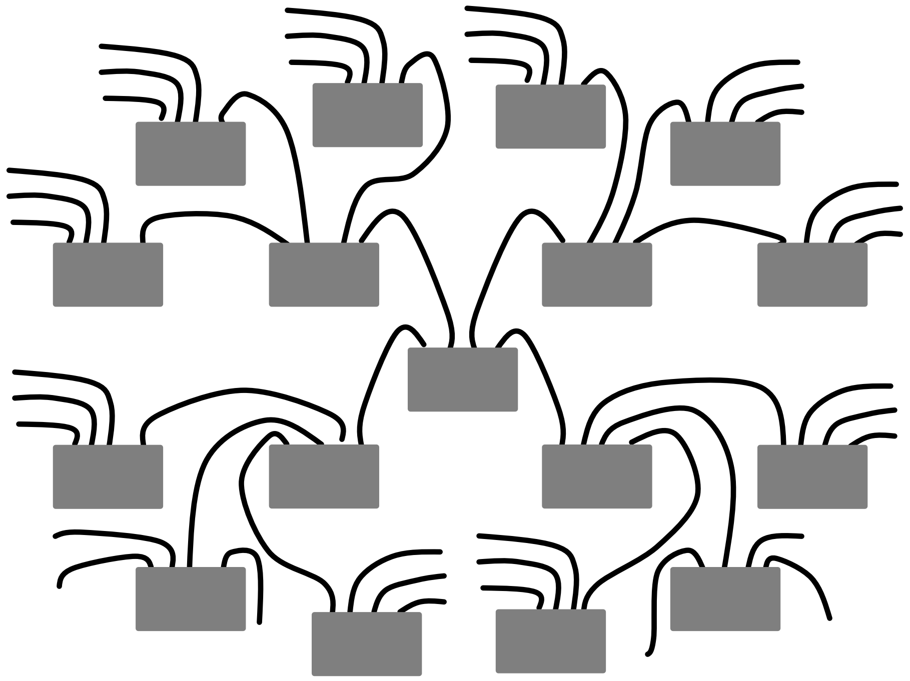
Fix radix k (router port count) as needed

With Moore Bound optimization, the network gets as many routers as possible (cost per router is minimized)

Key method

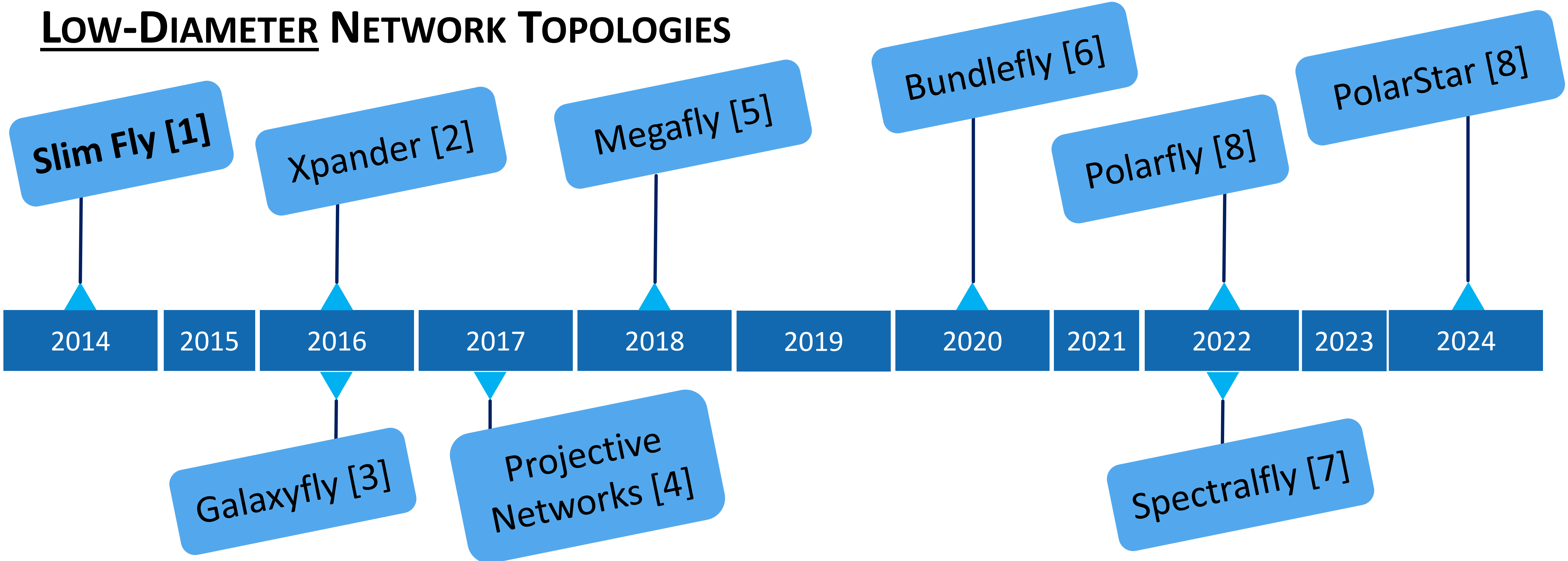
Optimize towards the Moore Bound [1]: the upper bound on the *number of vertices* in a graph with given *diameter D* and *radix k.*

$$MB(D,k) = 1 + k + k(k-1) + k(k-1)^2 + \cdots = 1 + k\sum_{i=0}^{D-1}(k-1)^i$$



[1] M. Miller, J. Siráň. Moore graphs and beyond: A survey of the degree/diameter problem, Electronic Journal of Combinatorics, 2005.

Low-Diameter Network Topologies



[1] M. Besta, T. Hoefler. Slim Fly: A Cost-Effective Low-Diameter Network Topology. ACM/IEEE Supercomputing, 2014. Best Student Paper Award
[2] A. Valadarsky et al. Xpander: Towards optimal-performance datacenters. ACM CoNEXT, 2016
[3] Fei Lei et al. Galaxyfly: A novel family of flexible-radix low-diameter topologies for large-scales interconnection networks. ACM/IEEE Supercomputing, 2016
[4] Cristóbal Camarero et al. Projective Networks: Topologies for Large Parallel Computer Systems. ACM/IEEE TPDS, 2017
[5] M. Flajslik et al. Megafly: A topology for exascale systems. ICHPC 2018
[6] F. Lei et al. Bundlefly: a low-diameter topology for multicore fiber. ACM/IEEE Supercomputing, 2020
[7] S. Aksoy et al. Spectralfly: Ramanujan graphs as flexible and efficient interconnection networks. arXiv preprint, 2022
[8] K. Lakhotia et al. PolarFly: A Cost-Effective and Flexible LowDiameter Topology. ACM/IEEE Supercomputing, 2022
[9] K. Lakhotia et al. PolarStar: Expanding the Scalability Horizon of Diameter-3 Networks. ACM SPAA 2024

These networks look complicated...

Moore graph

Article Talk

From Wikipedia, the free encyclopedia

In graph theory, a **Moore graph** is a regular graph whose girth (the shortest cycle length) is more than twice its diameter (the distance between the farthest two vertices). If the degree of such a graph is d and its diameter is k, its girth must equal $2k+1$. This is true, for a graph of degree d and diameter k, if and only if its number of vertices equals

$$1 + d\sum_{i=0}^{k-1}(d-1)^i,$$

a) Step 1: Constructing Base Ring $\mathbb{Z}_q$: Let $\mathbb{Z}_q = \{0, 1, ..., q-1\}$ be a commutative ring with modulo addition and multiplication. We have to find a *primitive element* ξ of $\mathbb{Z}_q$. ξ is an element of $\mathbb{Z}_q$ that *generates* $\mathbb{Z}_q$: all non-zero elements of $\mathbb{Z}_q$ can be written as ξ^i ($i \in \mathbb{N}$). In general, there exists no universal scheme for finding ξ [45], however an exhaustive search is viable for smaller rings; all the SF MMS networks that we tested were constructed using this approach.

b) Step 2: Constructing Generator Sets X and X': In the next step we utilize ξ to construct two sets X and X' called *generators* [35]. For $\delta = 1$ we have $X = \{1, \xi^2, ..., \xi^{q-3}\}$ and $X' = \{\xi, \xi^3, ..., \xi^{q-2}\}$ (consult [35] for other formulae). We will use both X and X' while connecting routers.

Are they really so complex?
Can we route/deploy them?

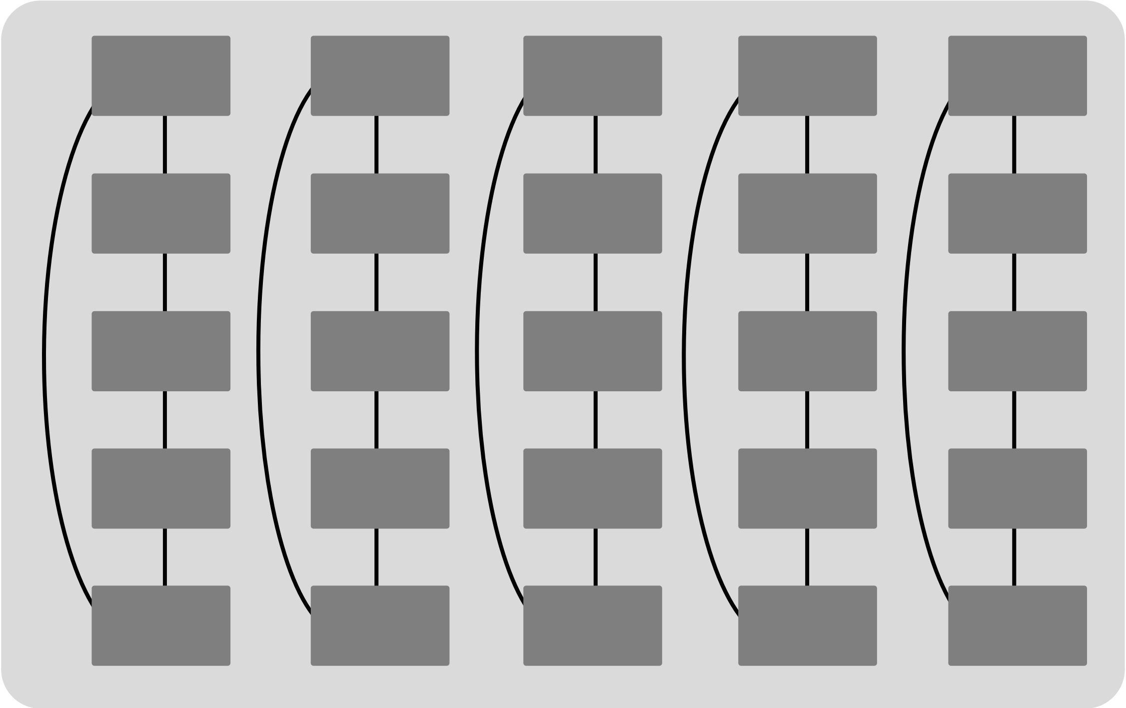
Let's see an example
Slim Fly

[1] M. Besta, T. Hoefler. Slim Fly: A Cost-Effective Low-Diameter Network Topology. ACM/IEEE Supercomputing 2014. Best Student Paper Award
[2] A. V... rs. ACM CoNEXT,
[3] Fei ... pologies for large
[4] Cris ... omputer S
[5] M. ...
[6] F. Le ... upercom
[7] S. Aksoy et al. Spectralfly: Ramanujan graphs as flexible and efficient ... nnection ne
[8] K. Lakhotia et al. PolarFly: A Cost-Effective and Flexible LowDiameter Topology. ACM/IE...
[9] K. Lakhotia et al. PolarStar: Expanding the Scalability Horizon of Diameter-3 Networks. ACM SPAA 2024

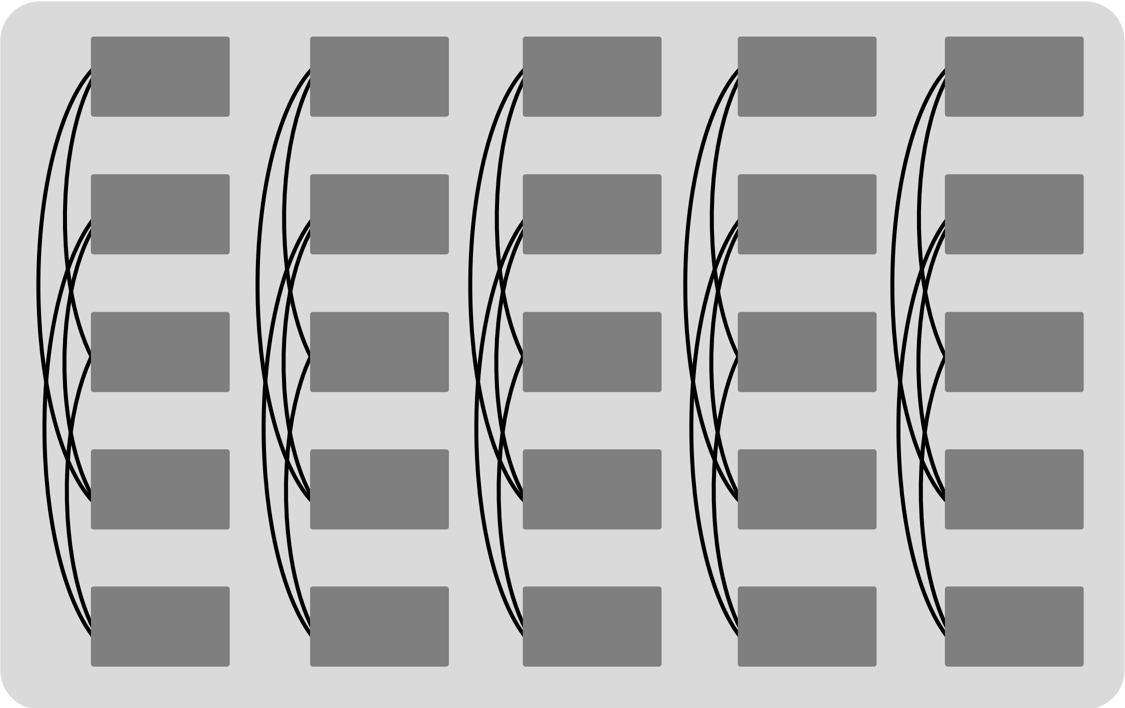
DEPLOYING SLIM FLY: STRUCTURE INTUITION

50 routers, 200 servers (omitted)

A subgraph with identical groups of routers

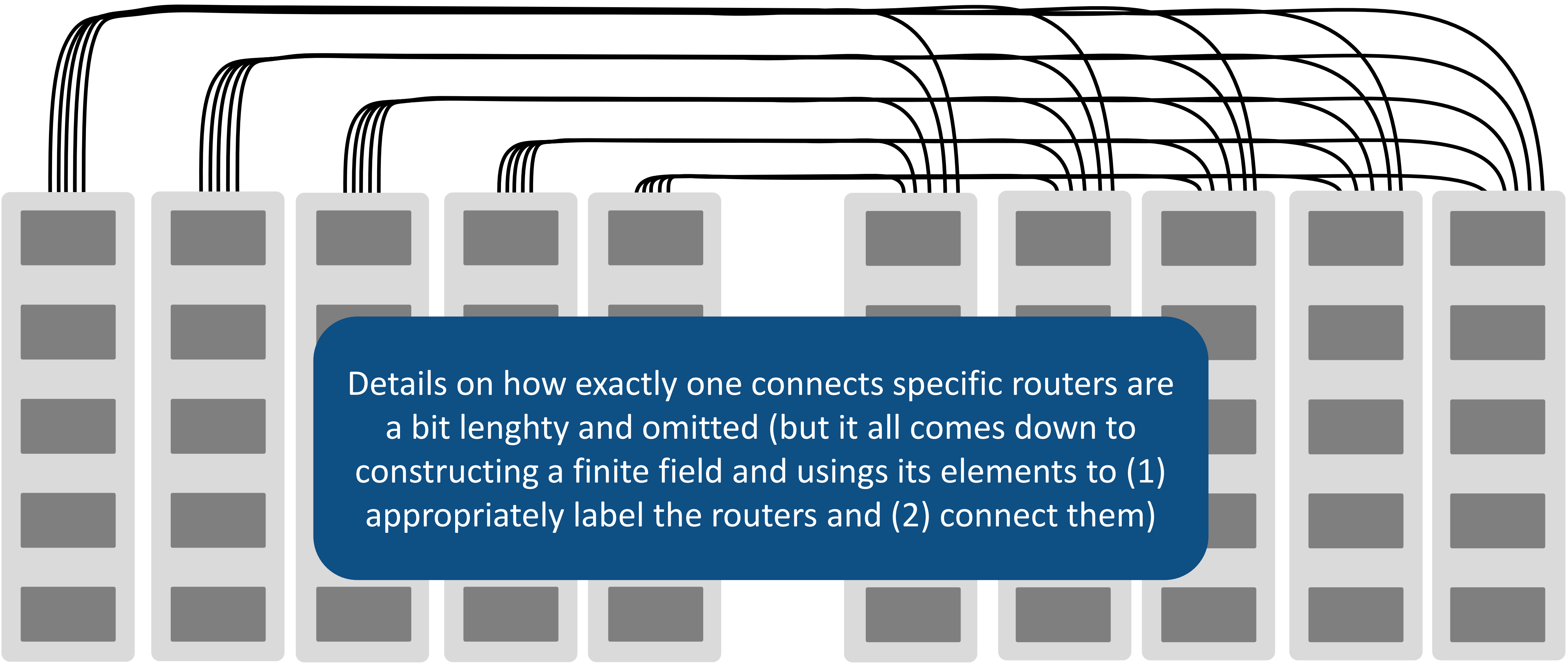


A subgraph with identical groups of routers



Deploying Slim Fly: Structure Intuition

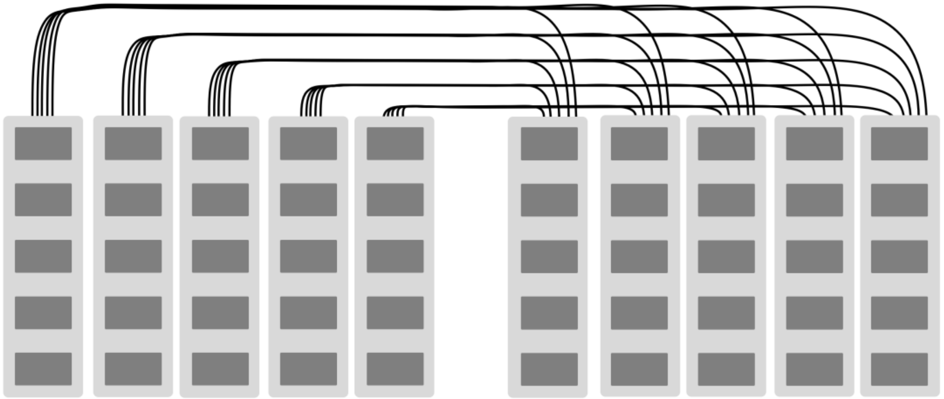
50 routers, 200 servers (omitted)

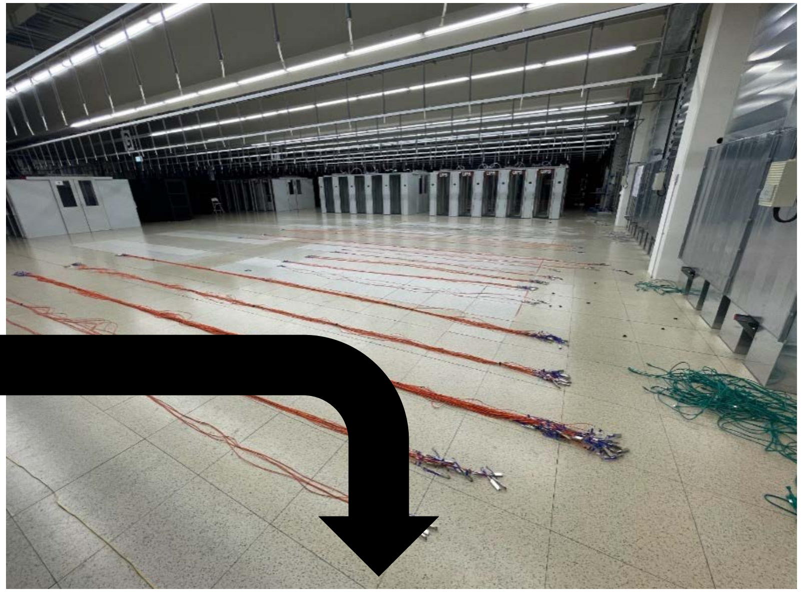


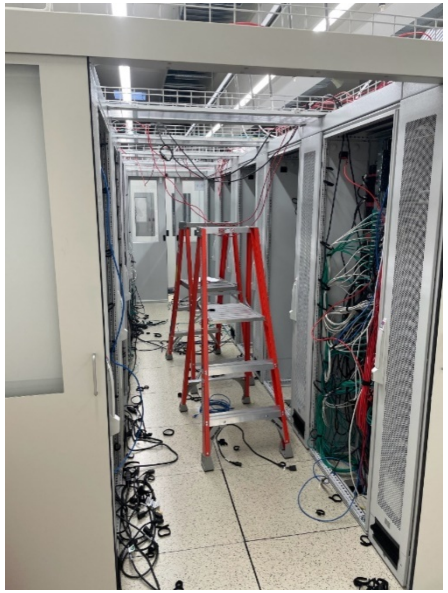
Details on how exactly one connects specific routers are a bit lenghty and omitted (but it all comes down to constructing a finite field and usings its elements to (1) appropriately label the routers and (2) connect them)

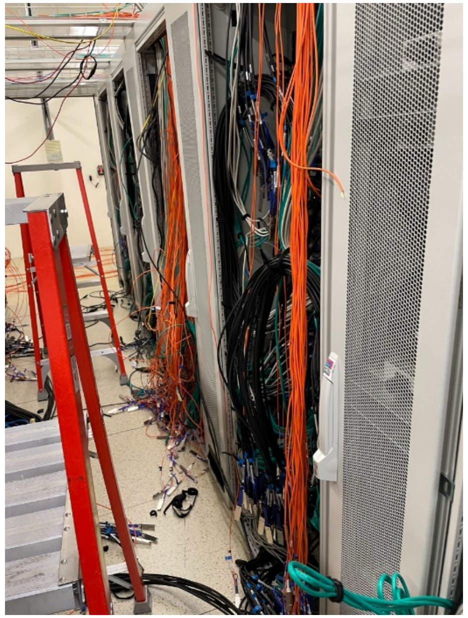
Groups form a fully-connected bipartite graph

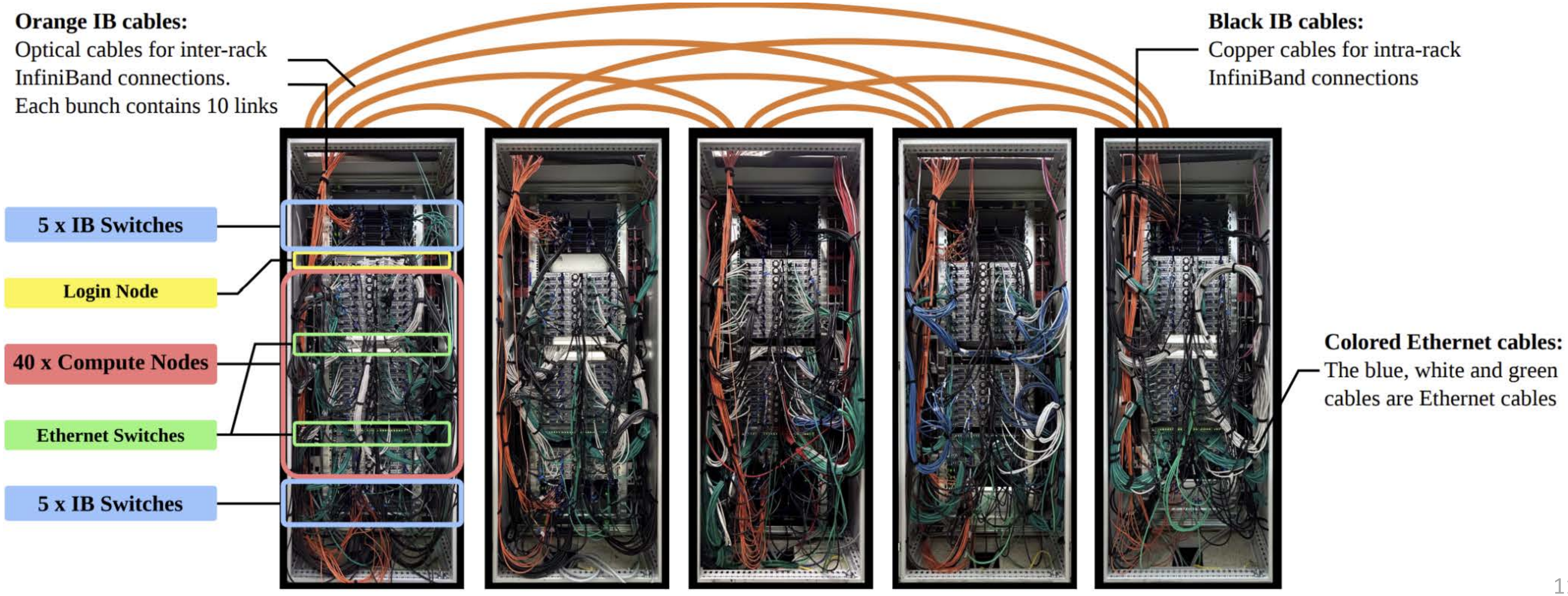
The First Slim Fly Construction





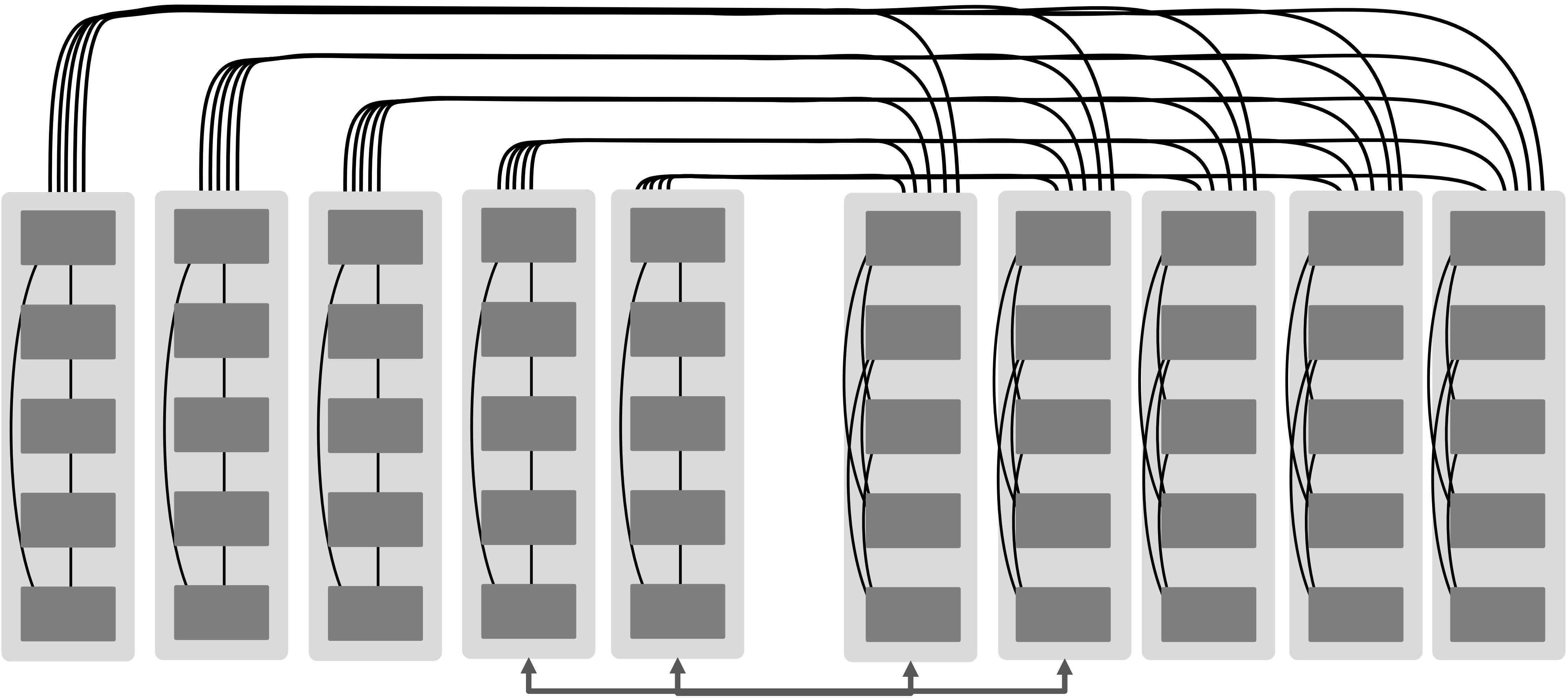






Deploying Slim Fly: Physical Layout

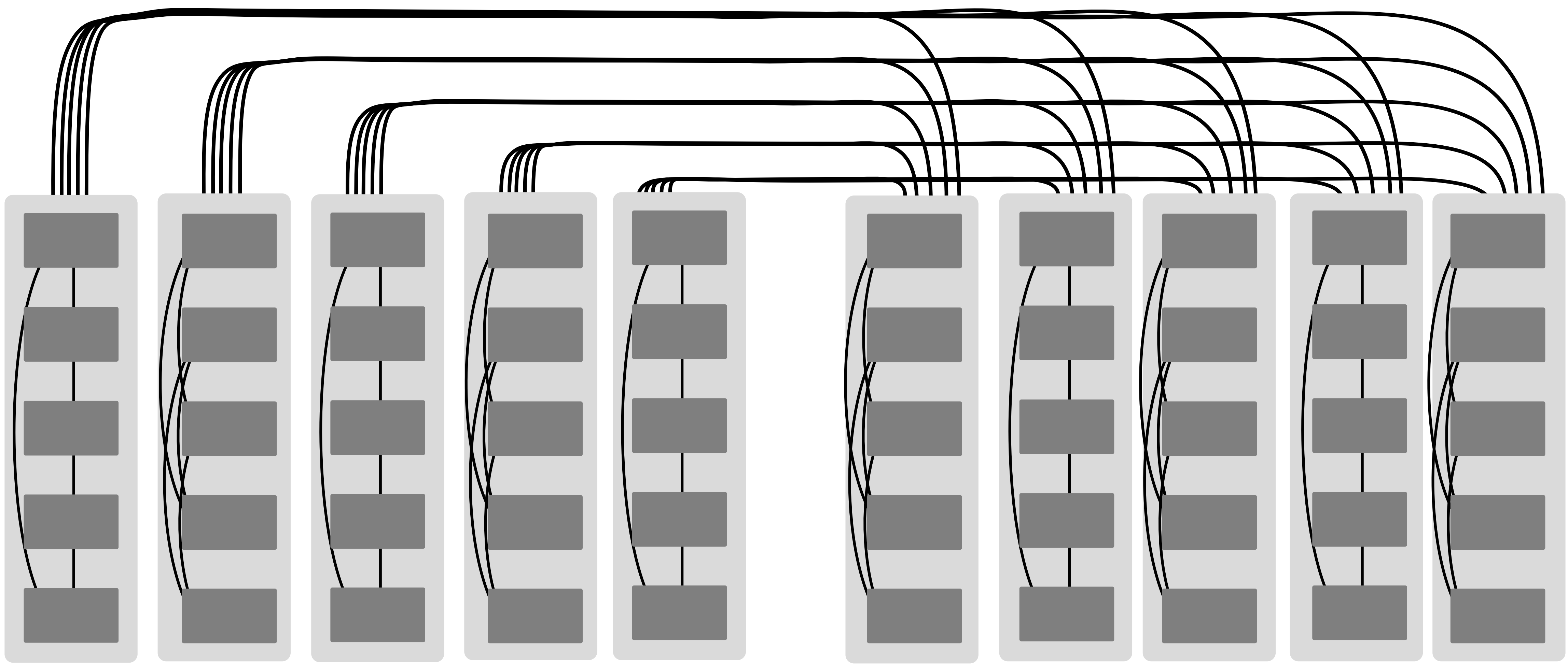
50 routers, 200 servers (omitted)



Mix (pairwise) groups
with different cabling patterns
to shorten inter-group cables

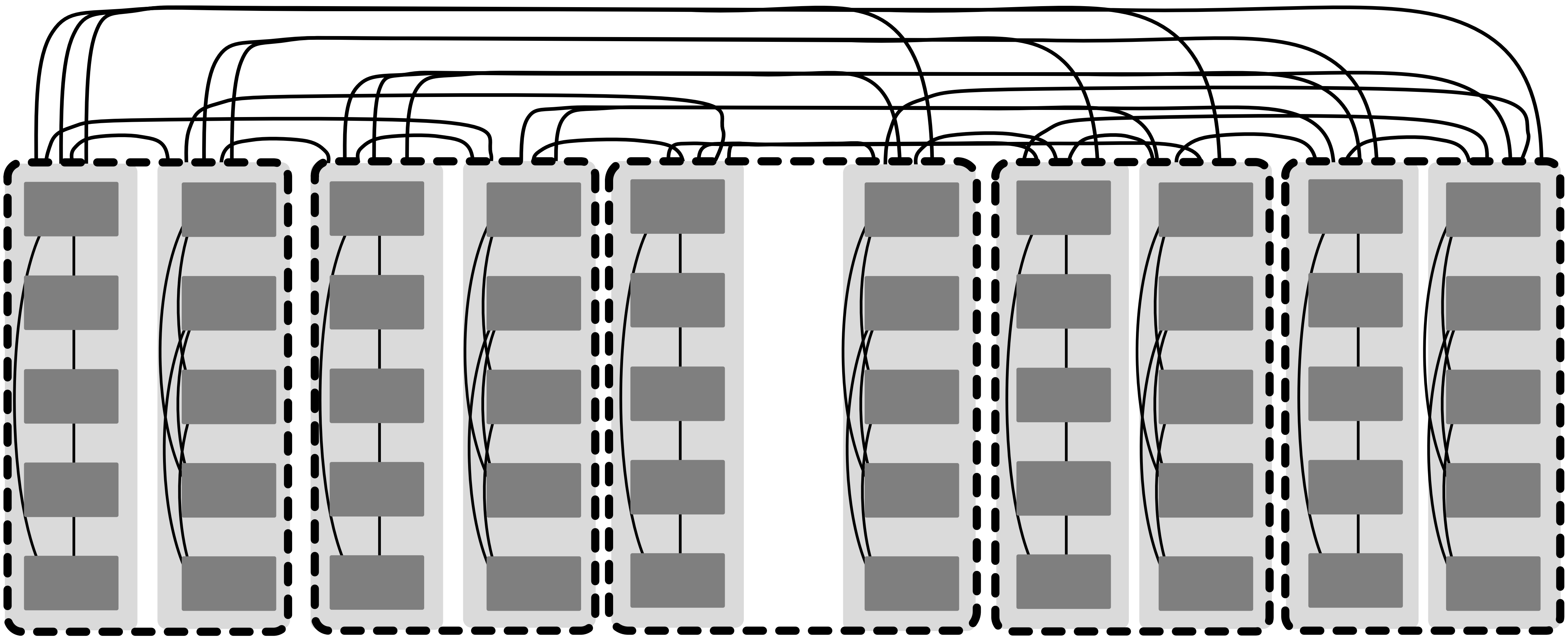
Deploying Slim Fly: Physical Layout

50 routers, 200 servers (omitted)



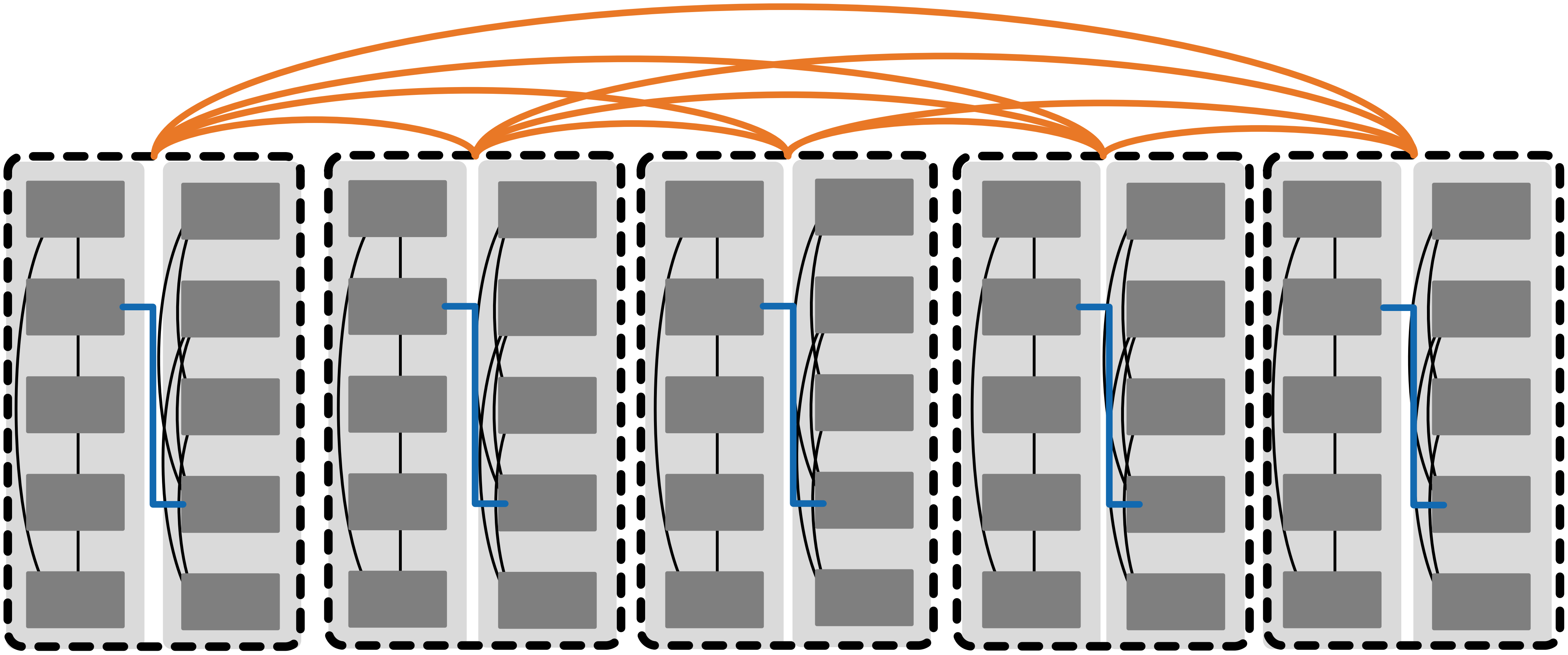
Deploying Slim Fly: Physical Layout

50 routers, 200 servers (omitted)



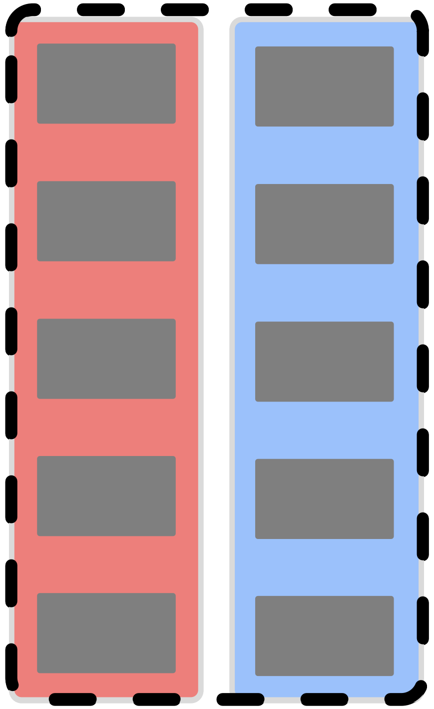
Deploying Slim Fly: Physical Layout

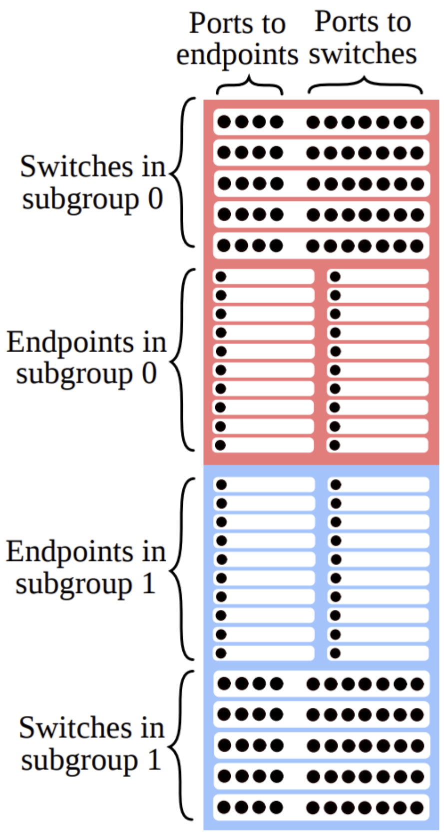
50 routers, 200 servers (omitted)

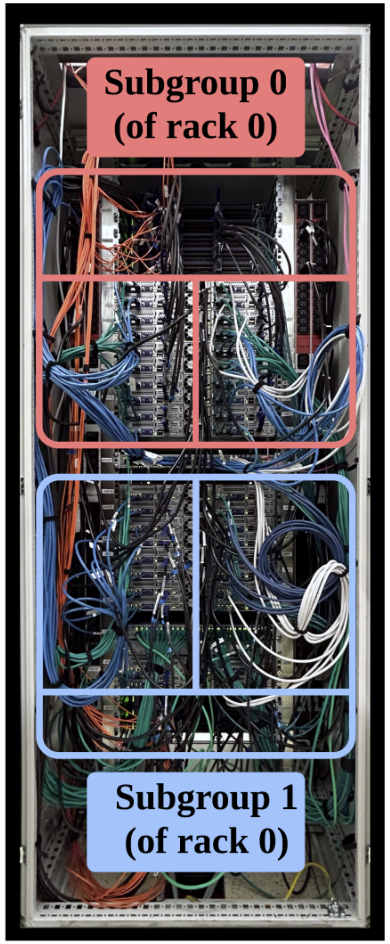


Racks form a fully-connected graph

DEPLOYING SLIM FLY: PHYSICAL LAYOUT

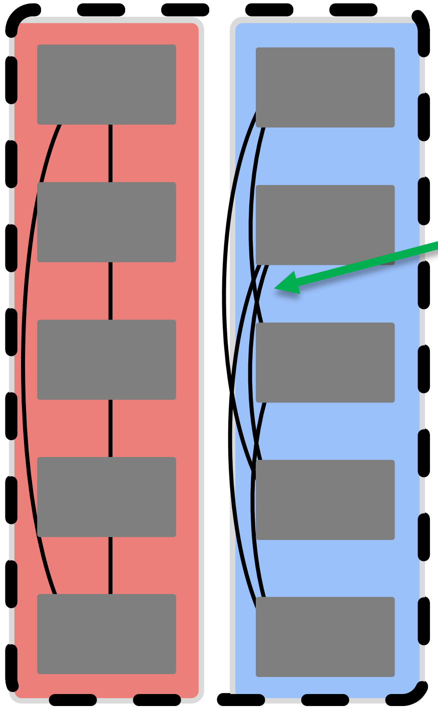


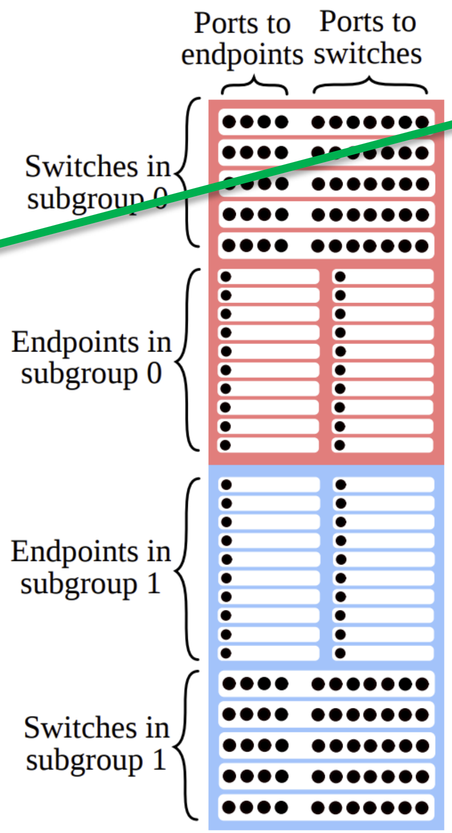


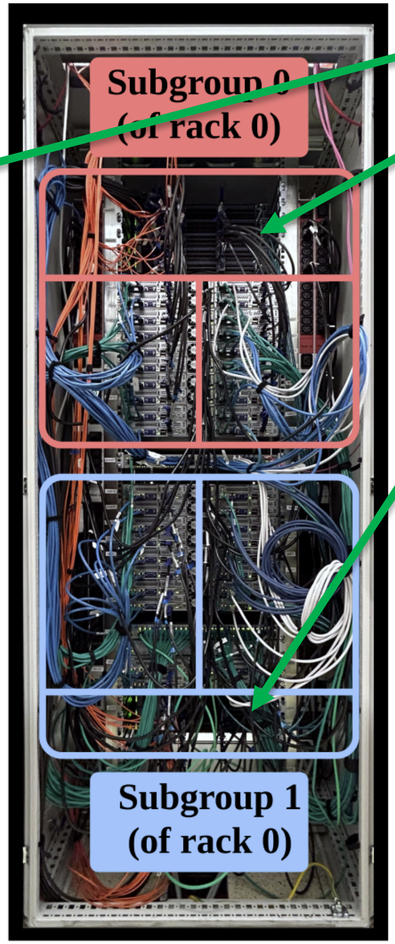


Deploying Slim Fly – Step 1: Intra-Subgroup Connections

Step 1

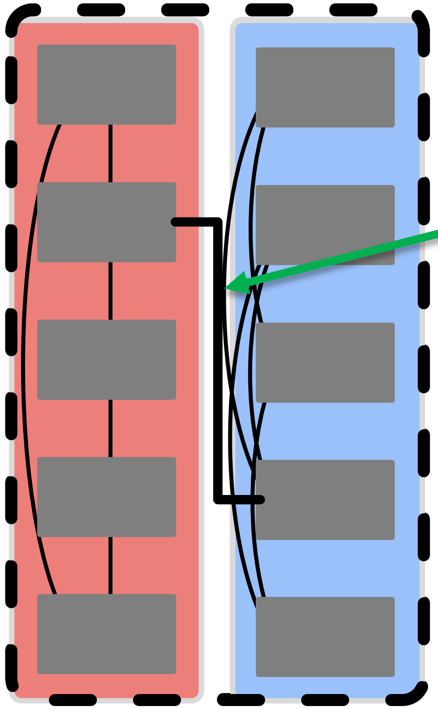


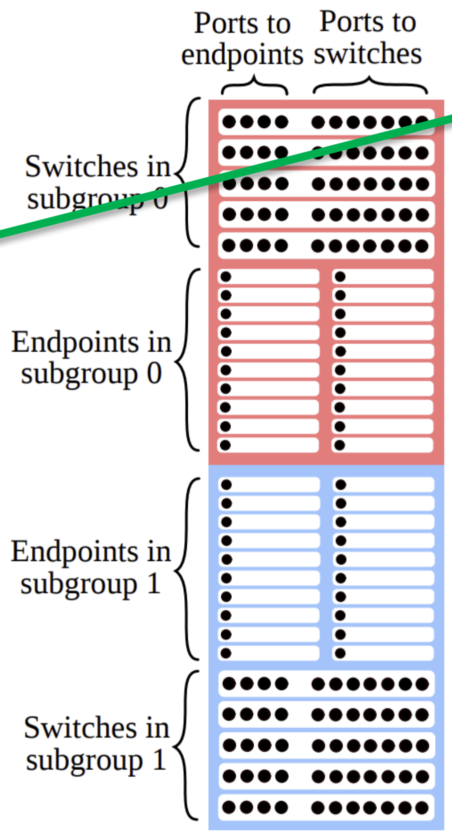


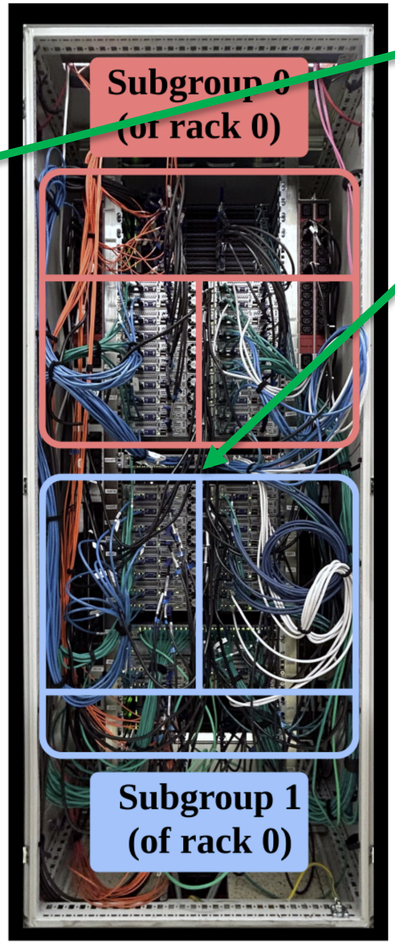


Deploying Slim Fly – Step 2: Inter-Subgroup Connections

Step 2

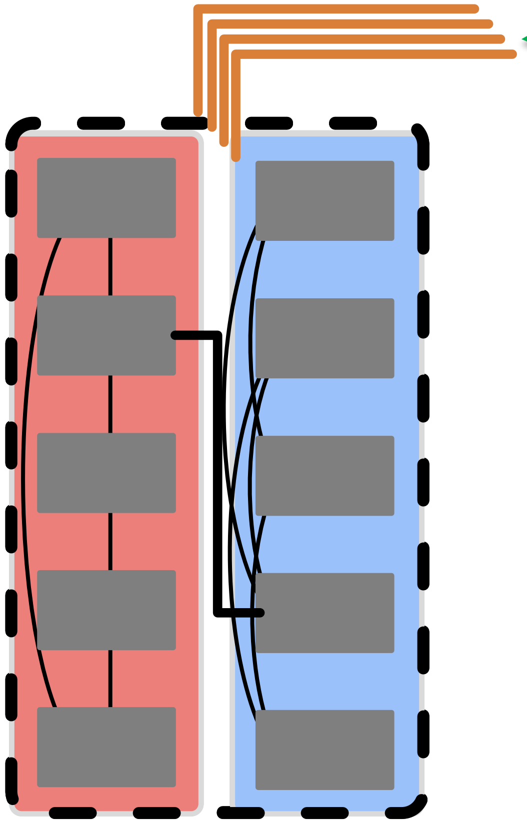


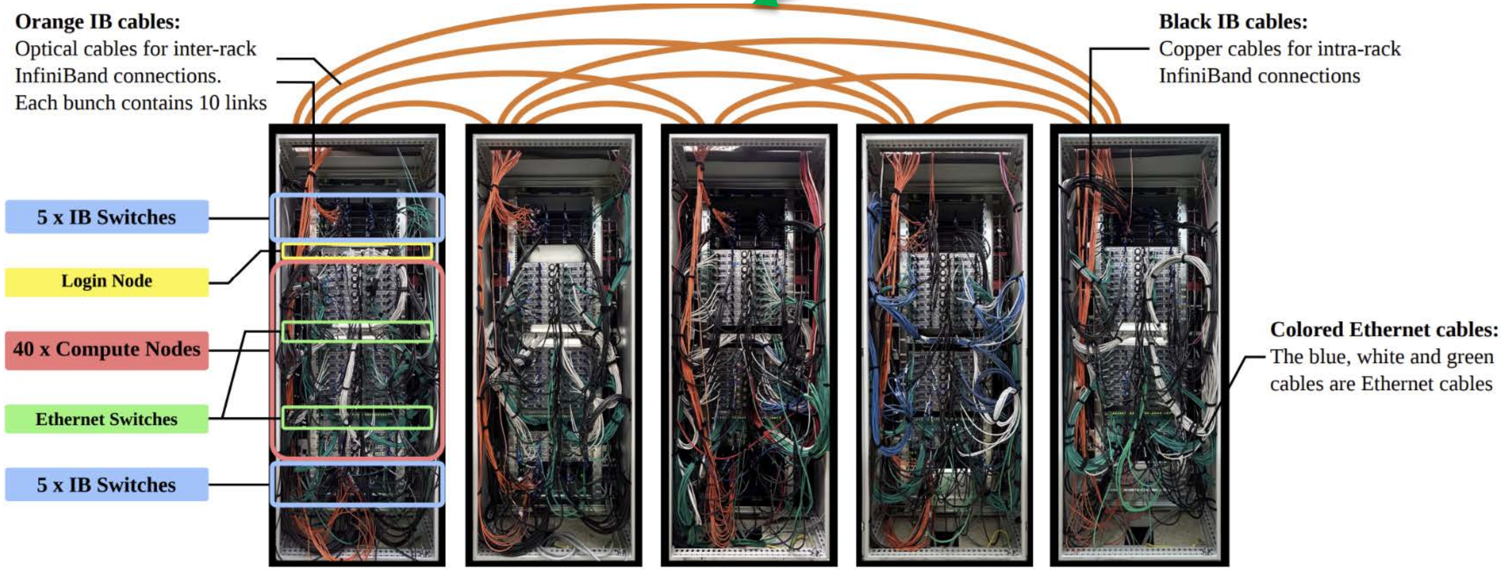




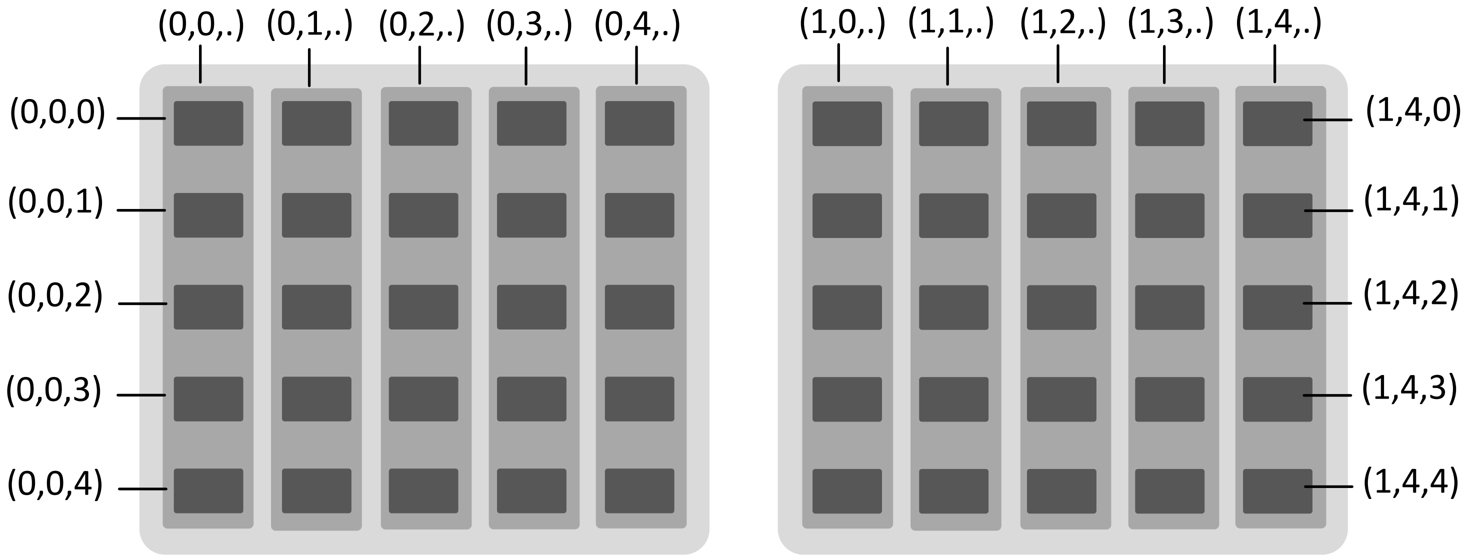
Deploying Slim Fly – Step 3: Inter-Rack Connections

Step 3



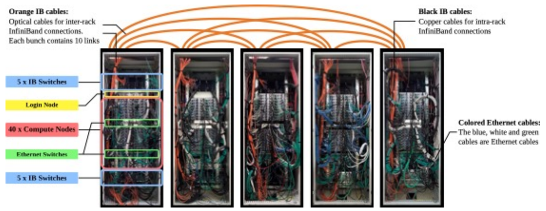


DEPLOYING SLIM FLY – VERIFICATION



Connectivity determined by the following algebraic equations

router $(0, x, y)$ is connected to $(0, x, y')$ iff $y - y' \in X$;
router $(1, m, c)$ is connected to $(1, m, c')$ iff $c - c' \in X'$;
router $(0, x, y)$ is connected to $(1, m, c)$ iff $y = mx + c$;

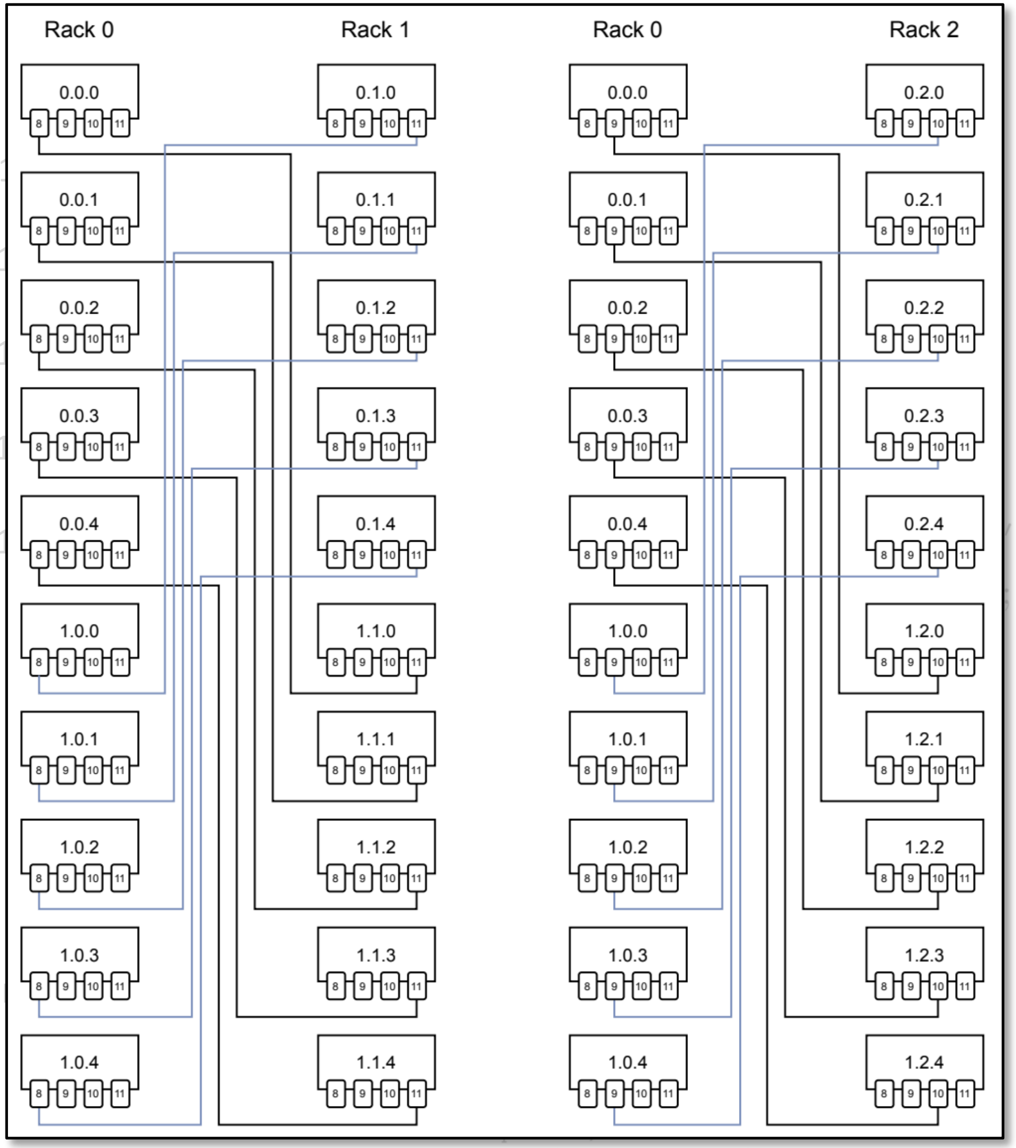


→ INFINIBAND TRADE ASSOCIATION → (Python) →

```
Problems with switch 90e200:
Rack: 5 Slot: 10
----------------------------------------------------
Missing or Extra Connections:
Switch 90e200 (Rack: 5 Slot: 10) should have been
    connected to the following other switches but
    isn't: 90db80, e46880
----------------------------------------------------
Incorrectly wired ports:
Switch 90e200 (Rack: 5 Slot: 10) does not have a
    connection on port 5, but should be connected
    to 90db80 (Rack: 5 Slot: 8)
Switch 90e200 (Rack: 5 Slot: 10) does not have
    a connection on port 10, but should be
    connected to e46880 (Rack: 2 Slot: 3)
----------------------------------------------------
```

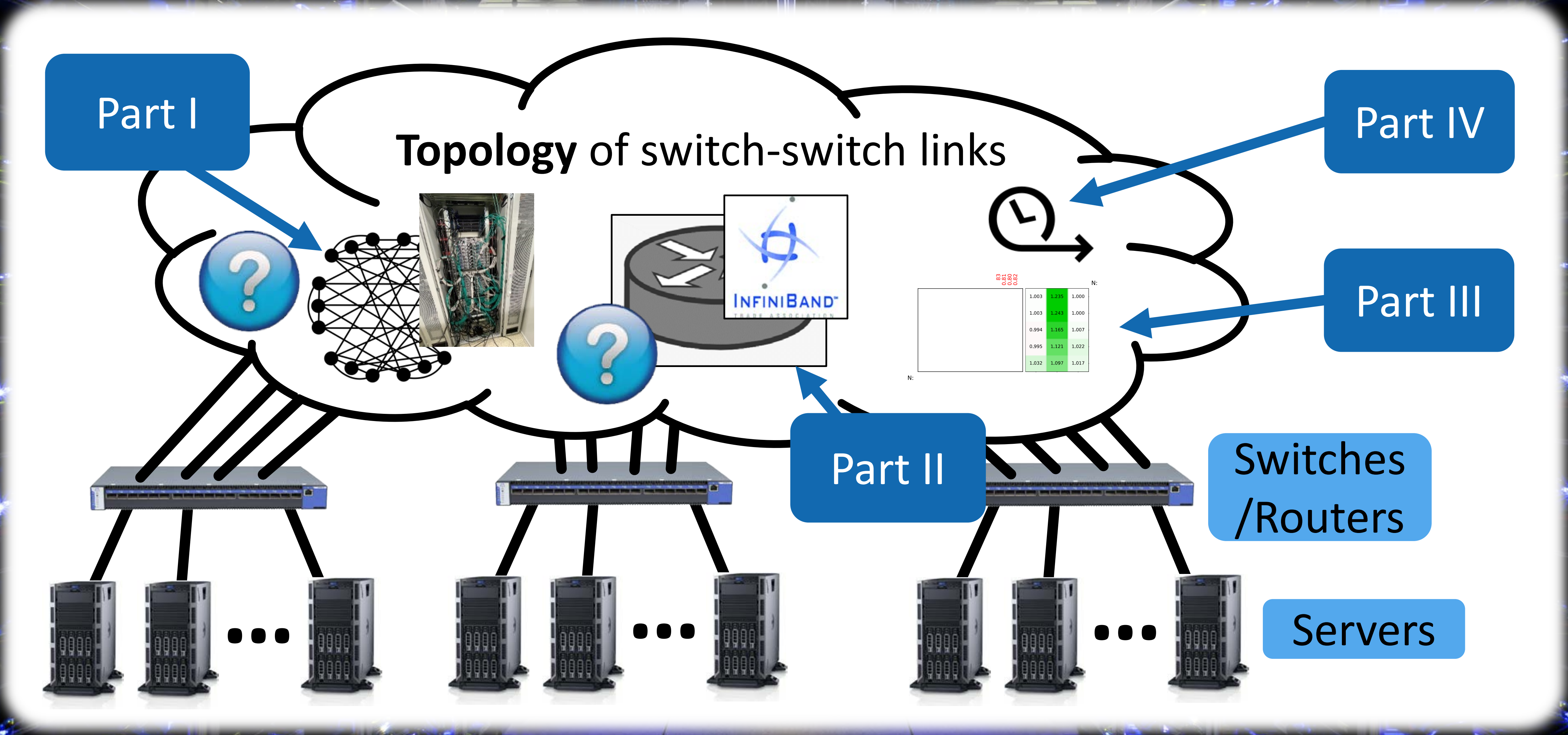
Deploying Slim Fly – Verification

Verification is straightforward



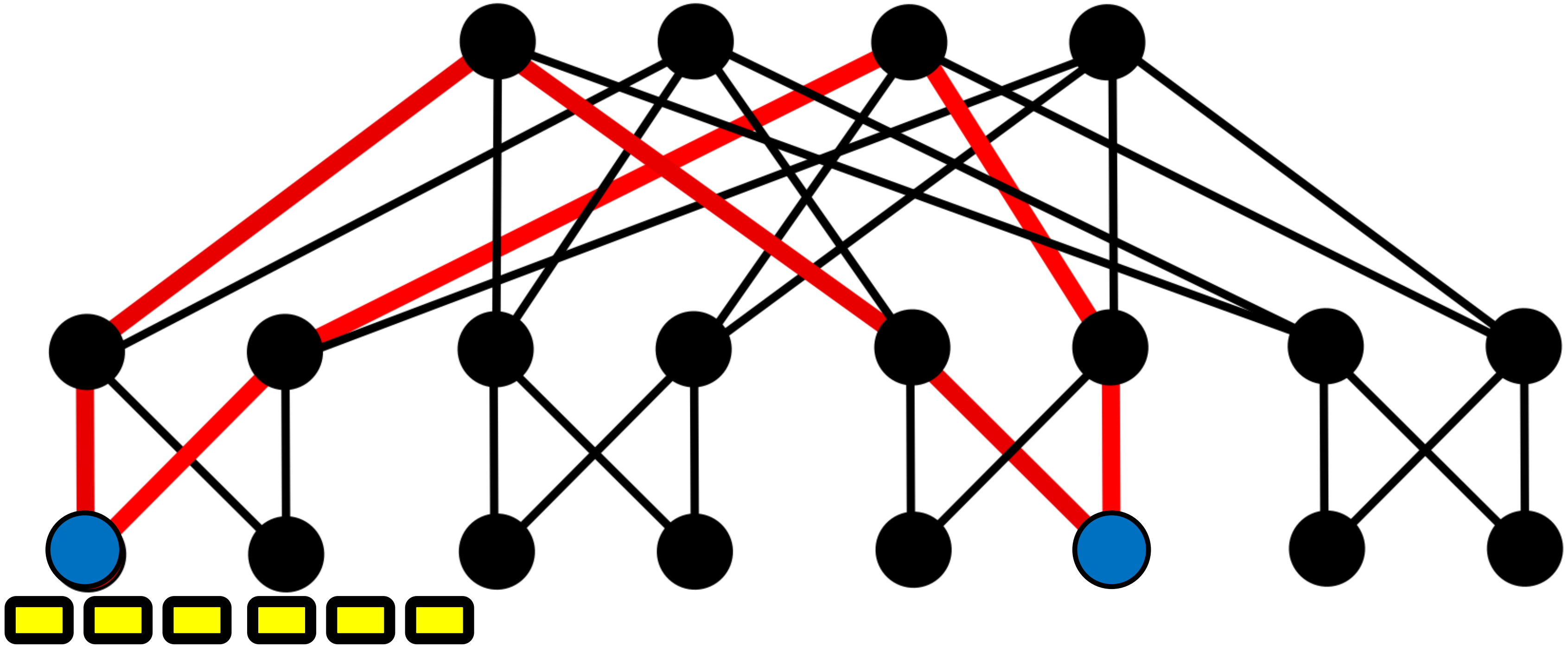
connected to e46880 (Rack: 2 Slot: 3)

NETWORK TOPOLOGIES : SETTING & PRESENTATION PLAN



Routing in Fat Trees

High-performance routing is facilitated by **numerous multiple shortest paths of equal lengths** between any endpoints



Routing in Fat Trees

High-performance routing is facilitated by **numerous multiple shortest paths of equal lengths** between any endpoints

INFORMATIONAL

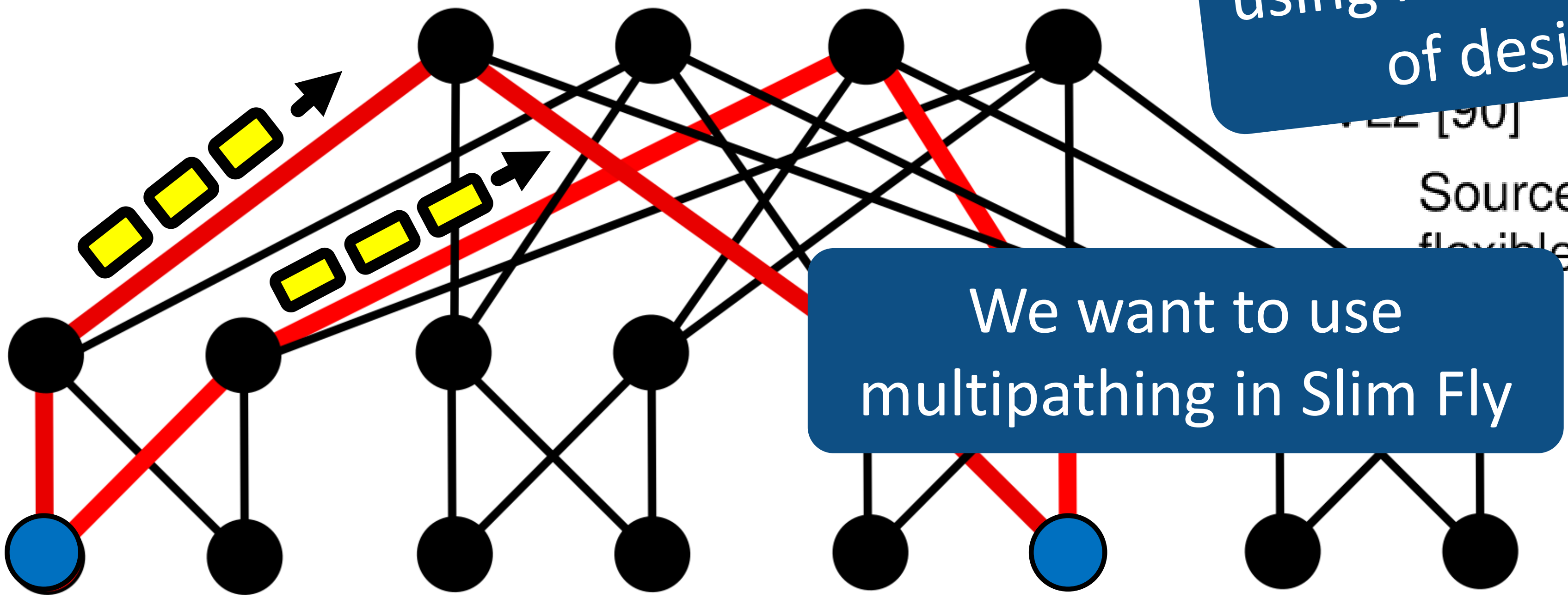
Network Working Group
Request for Comments: 2992
Category: Informational

C. Hopps
NextHop Technologies
November 2000

Analysis of an Equal-Cost Multi-Path Algorithm

ECMP

Established techniques for using multipathing & plethora of designs available



We want to use multipathing in Slim Fly

Al-Fares et al. [7]

[90]

WCMP for DC [233]

Source routing for flexible DC fabric [117]

Monsoon [91]

PortLand [160]

SPAIN [158]

ECMP-VLB [123]

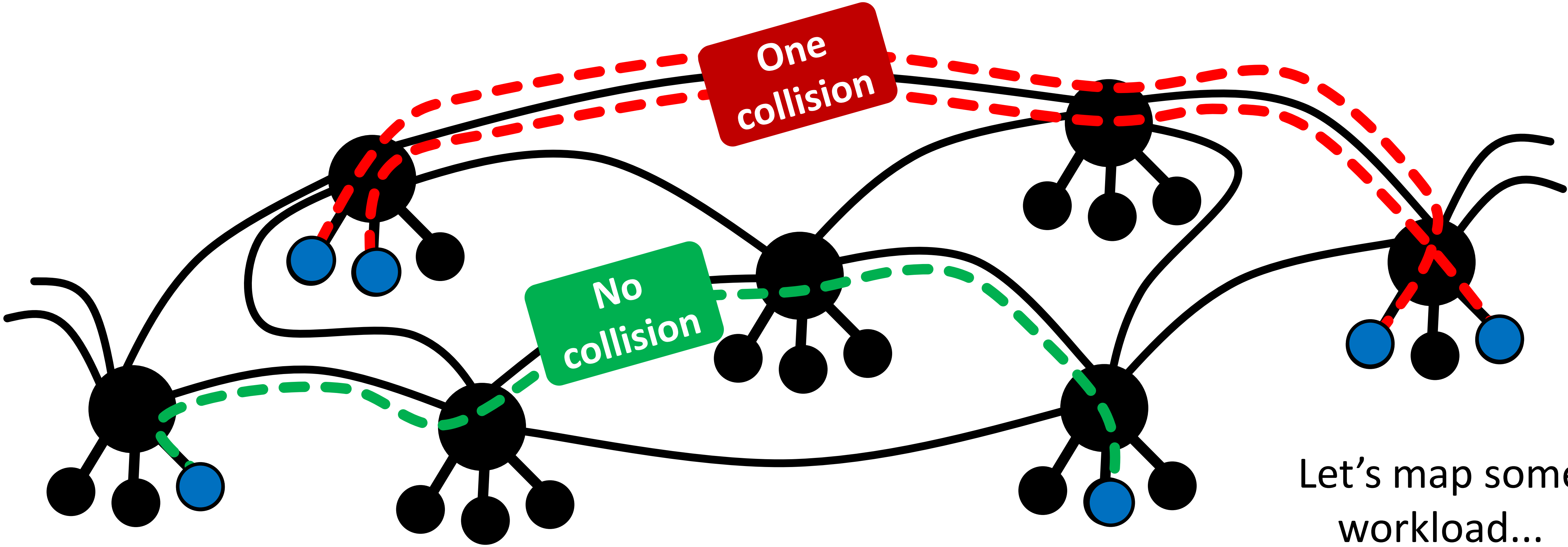
Work by Linden et al. [215]

Work by Suchara et al. [204]

Multipath Routing: Motivation

What are the problems that we want to tackle with multipathing?

Flows collide!

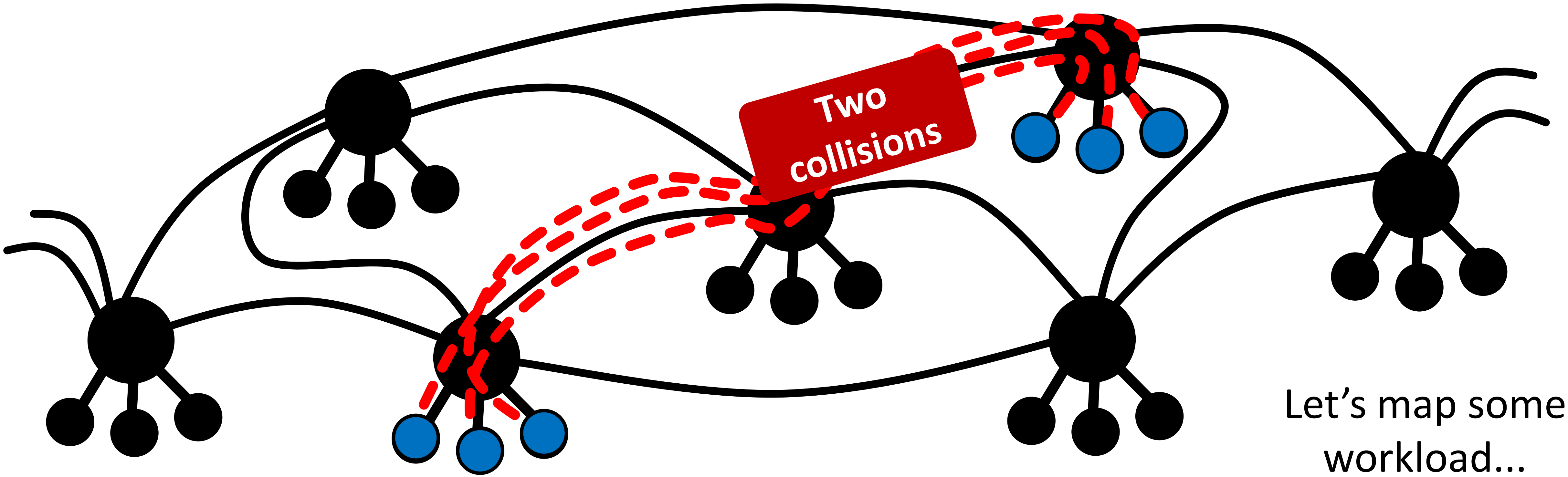


Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

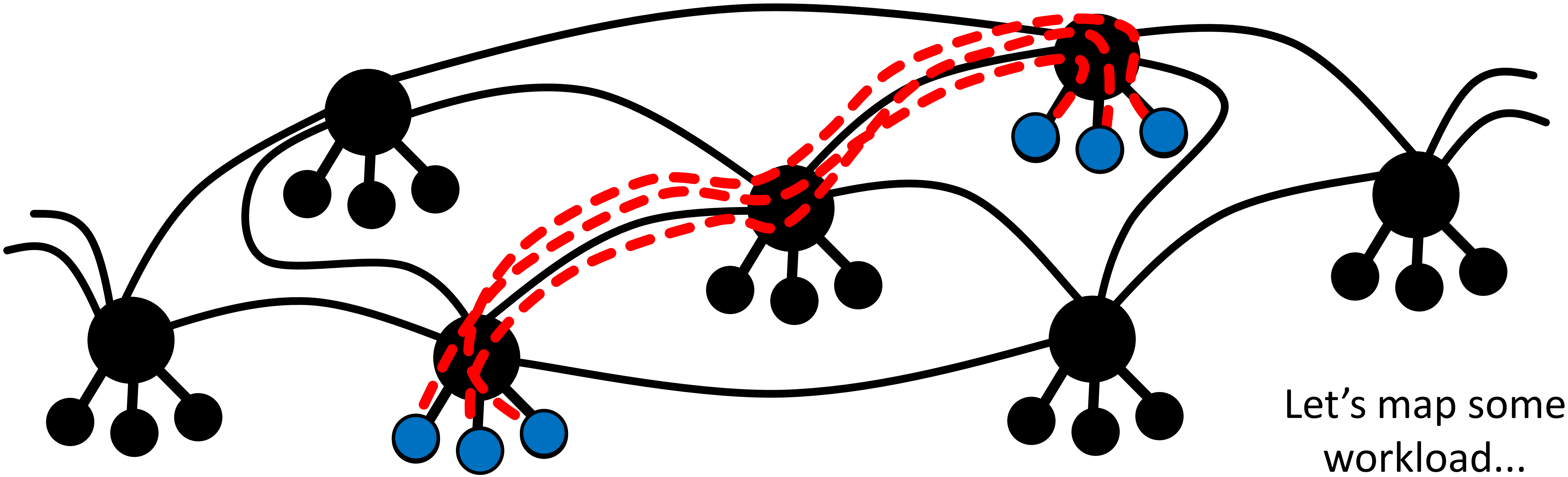


Let's map some workload...

Multipath Routing: Motivation

What are the problems that we want to tackle with multipathing?

Flows collide!



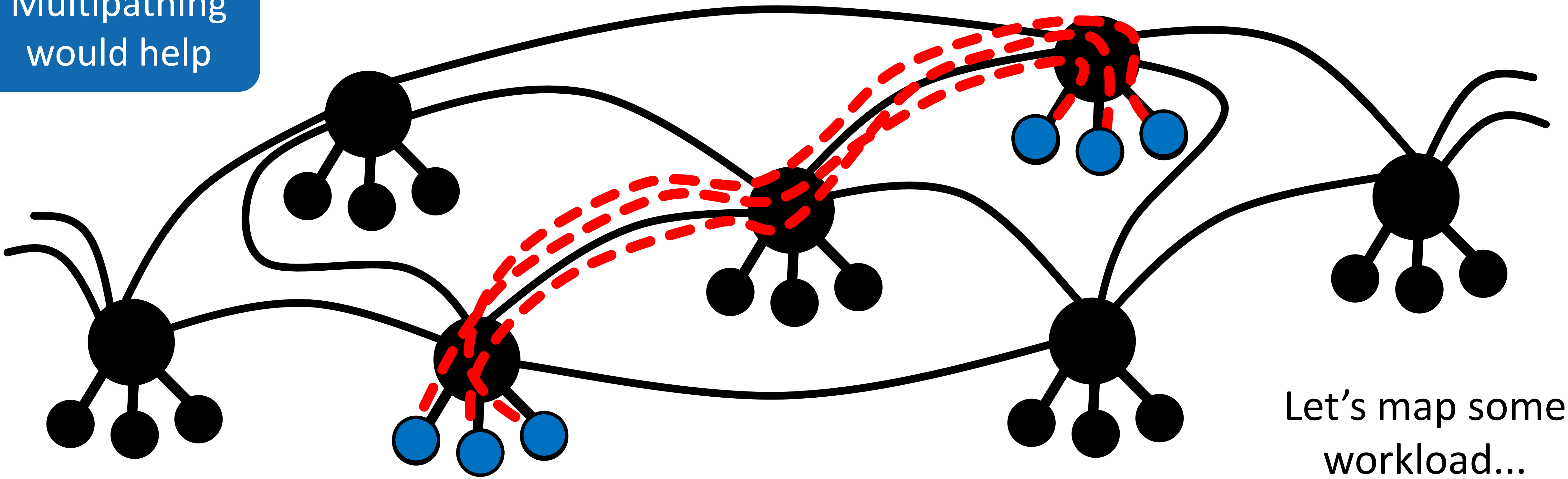
Let's map some workload...

Multipath Routing: Motivation

What are the problems that we want to tackle with multipathing?

Flows collide!

Multipathing would help



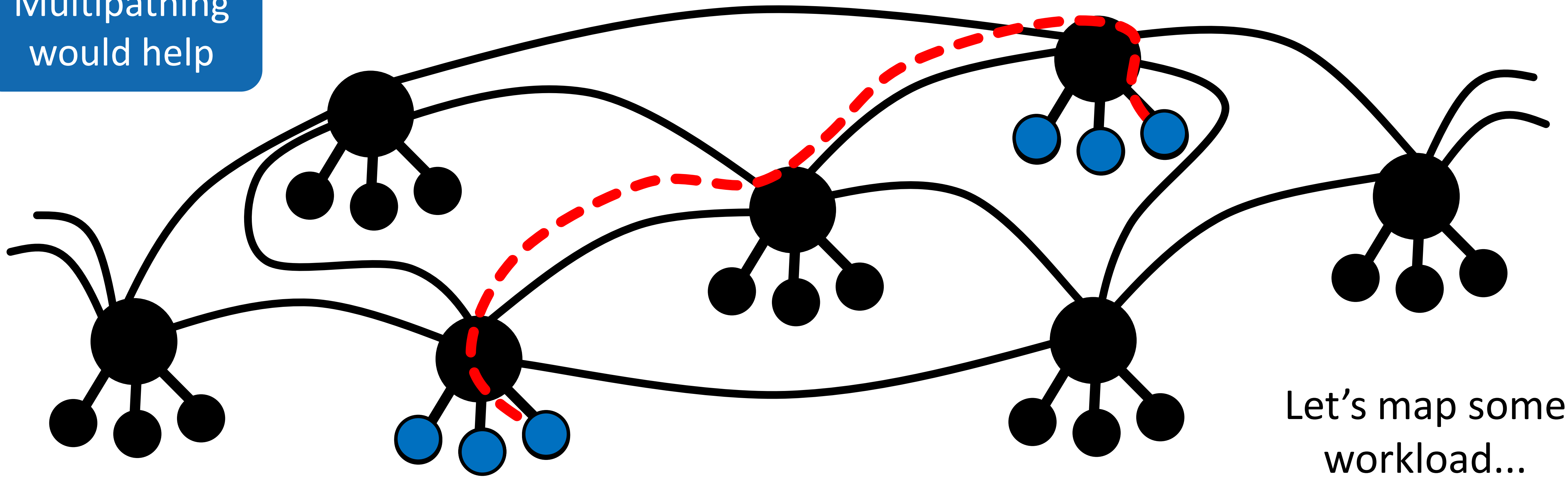
Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Flows collide!

Multipathing would help



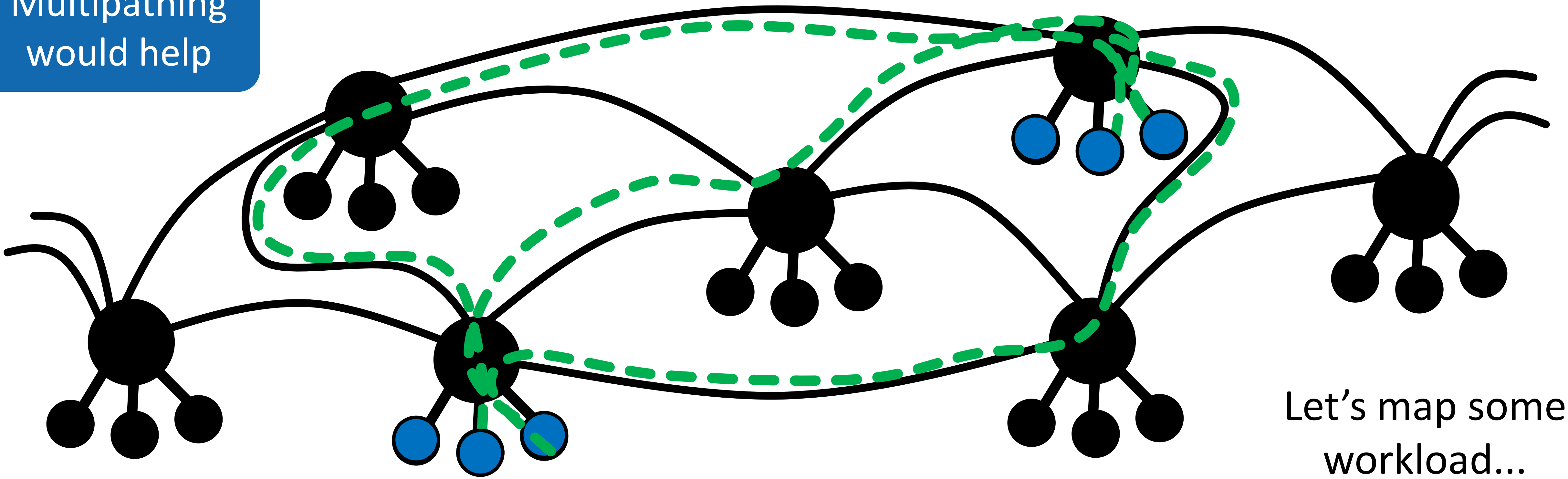
Let's map some workload...

MULTIPATH ROUTING: MOTIVATION

What are the problems that we want to tackle with multipathing?

Multipathing would help

Flows collide!



Let’s map some workload...

How many multiple paths do we need to tackle flow collisions?

Are there enough multiple paths in Slim Fly?

Key Insight 1: We need **three** disjoint paths per router pair to handle [almost all] colliding flows [1]

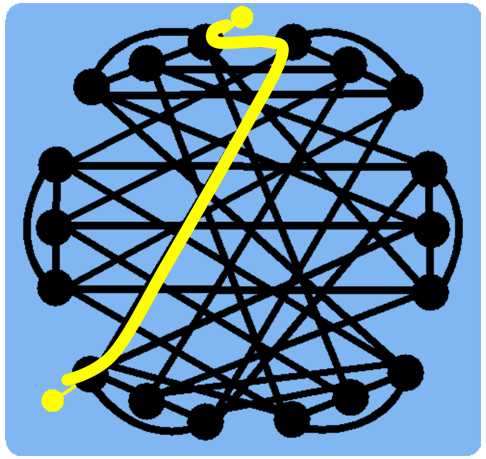
Key Insight 2: In most cases, there is only enough path diversity when considering “almost” minimal paths [1]

[1] M. Besta et al. FatPaths: Routing in Supercomputers and Data Centers when Shortest Paths Fall Short. SC’20.

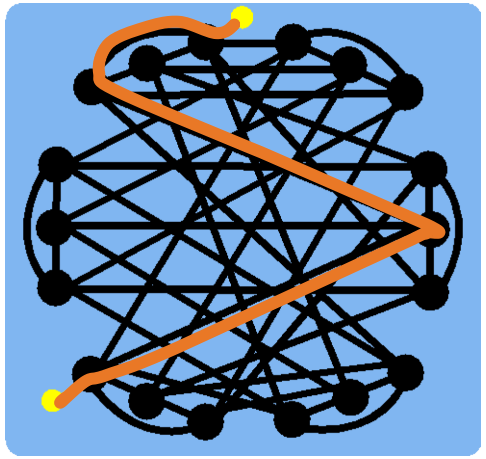
Novel Layered Routing Protocol

Key idea: distribute & encode different paths across „routing layers”

Layer 0:



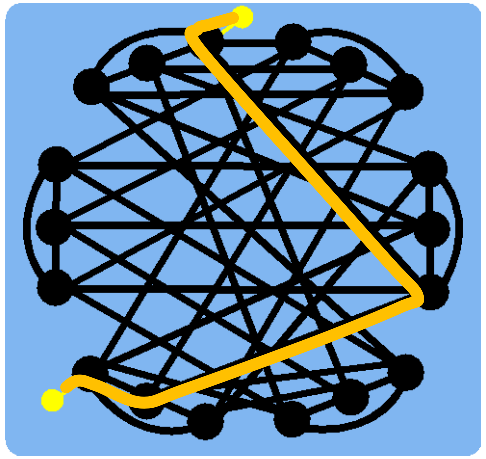
Layer 1:

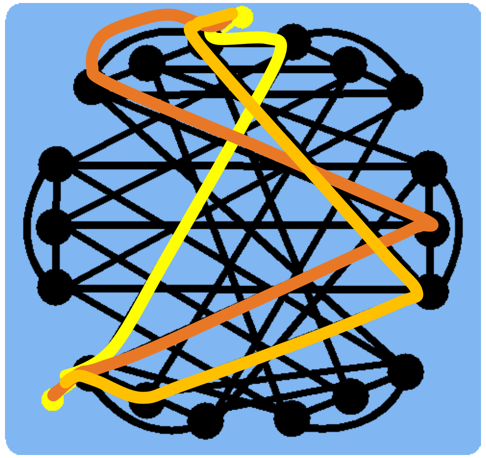


Layer 0: minimal paths

Layers 1-...: non-minimal paths

Layer 2:





We minimize the overlap of paths between layers

Novel Layered Routing Protocol

Key idea: distribute & encode different paths across „routing layers"

Layer 0:

Layer 1:

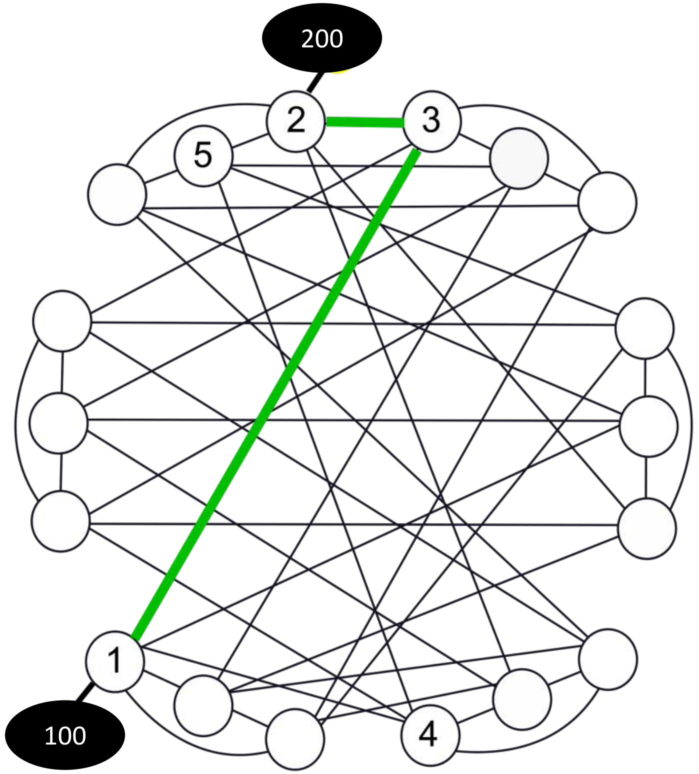
Layer 0: minimal paths

Layers 1-...: non-minimal paths

We minimize the overlap of paths between layers

What are example problems that we need to tackle when implementing this with IB?

InfiniBand - Addressing

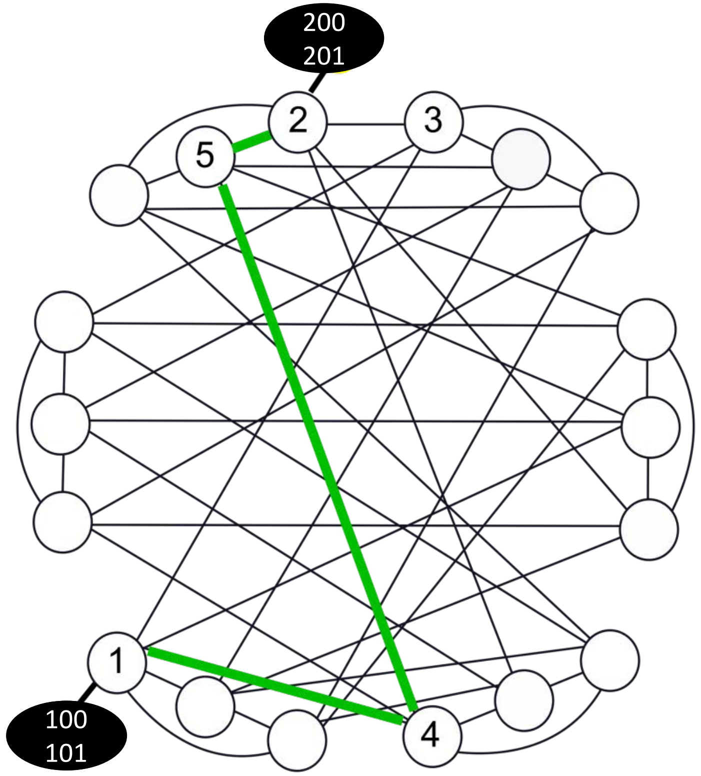


Key idea: use multiple LIDs (LID = Local Identifier) for the same endpoint to encode multiple paths

Problem 1: How do we introduce layers in InfiniBand?

Switch 1	
Destination LID	**Next Hop**
200	3

InfiniBand - Addressing



Key idea: use multiple LIDs (LID = Local Identifier) for the same endpoint to encode multiple paths

Problem 1: How do we introduce layers in InfiniBand?

Switch 1	
Destination LID	**Next Hop**
200	3
201	4

IB Address Space Limitations

40 Ports

#Layers	Max #Switches	Max #Endnodes	#Endnodes per Switch	Network radix
1	578	7514	13	25
2	587	7514	13	25
4	578	7514	13	25
8	450	5400	12	23
16	288	2592	9	18
32	162	1134	7	13
64	98	588	6	11
128	72	360	5	9

Problem 2: How many layers can we support?

IB supports at most 49’151 unicast addresses

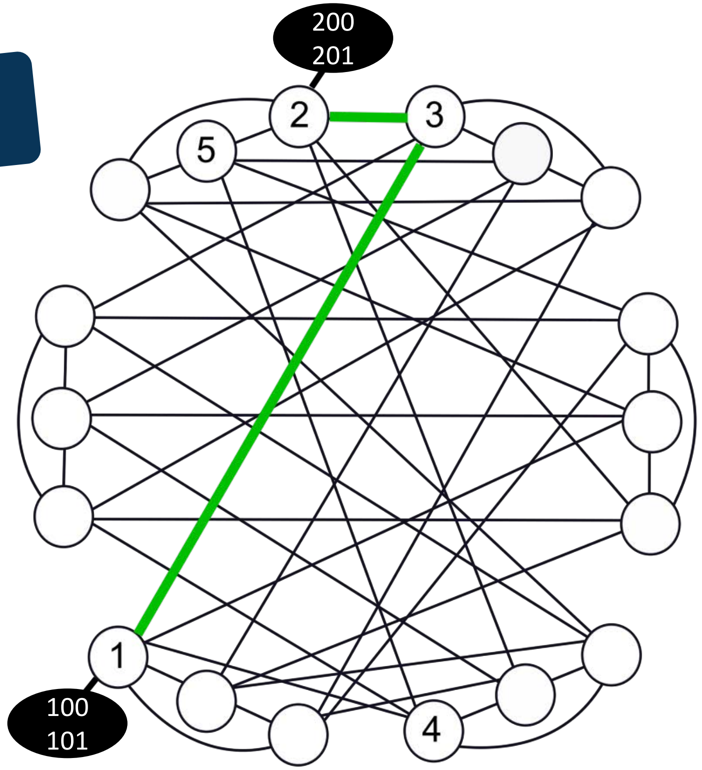
For a given number of layers, what is the largest Slimfly network that IB can support, while maintaining full global bandwidth?

InfiniBand - Layer Generation Algorithm

Problem 3: Given that we want 3 disjoint paths, how many layers do we need?

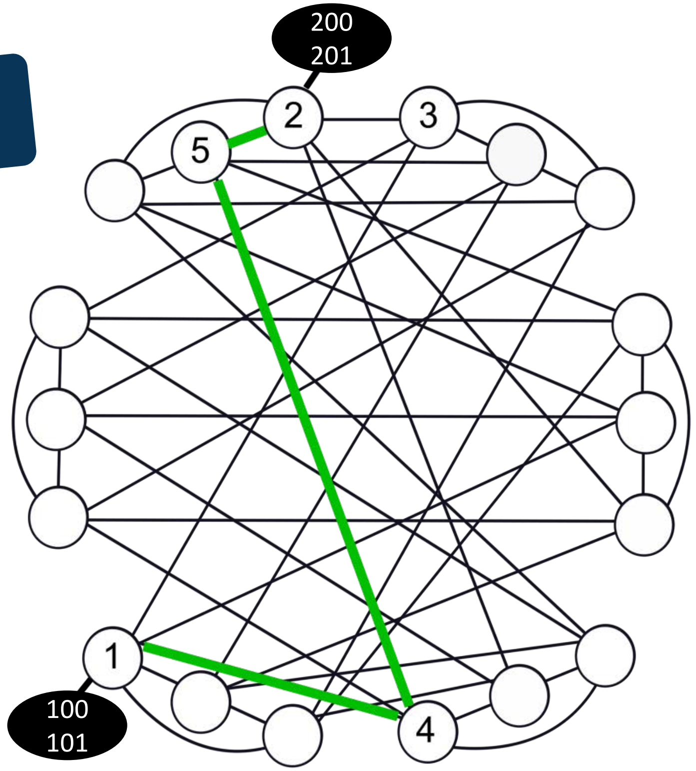
For each switch pair find and add an almost-minimal path in every layer (use minimal paths in Layer 0)

L 0



Switch 1	
Destination LID	**Next Hop**
200	3
201	4

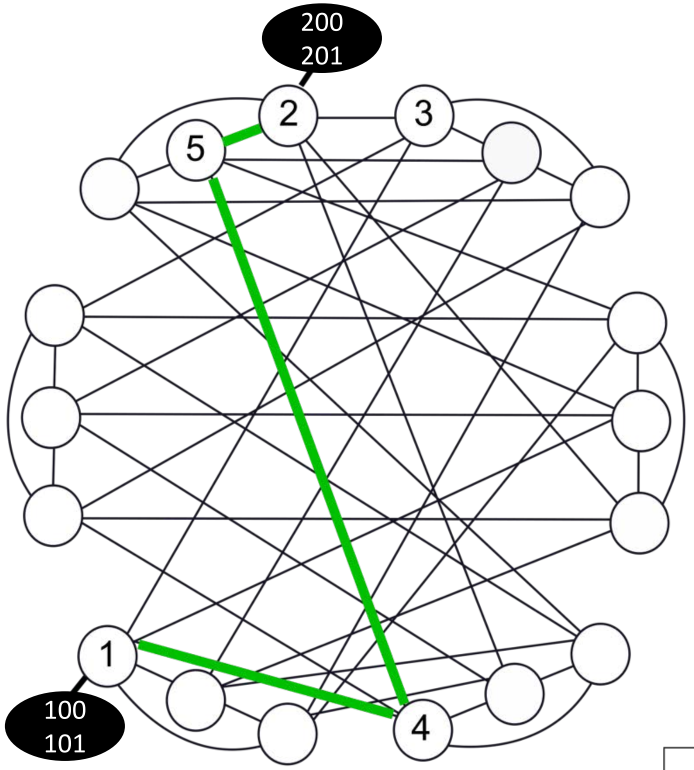
L 1



InfiniBand - Layer Generation Algorithm

Problem 4: Can we fail to add an almost-minimal disjoint path?

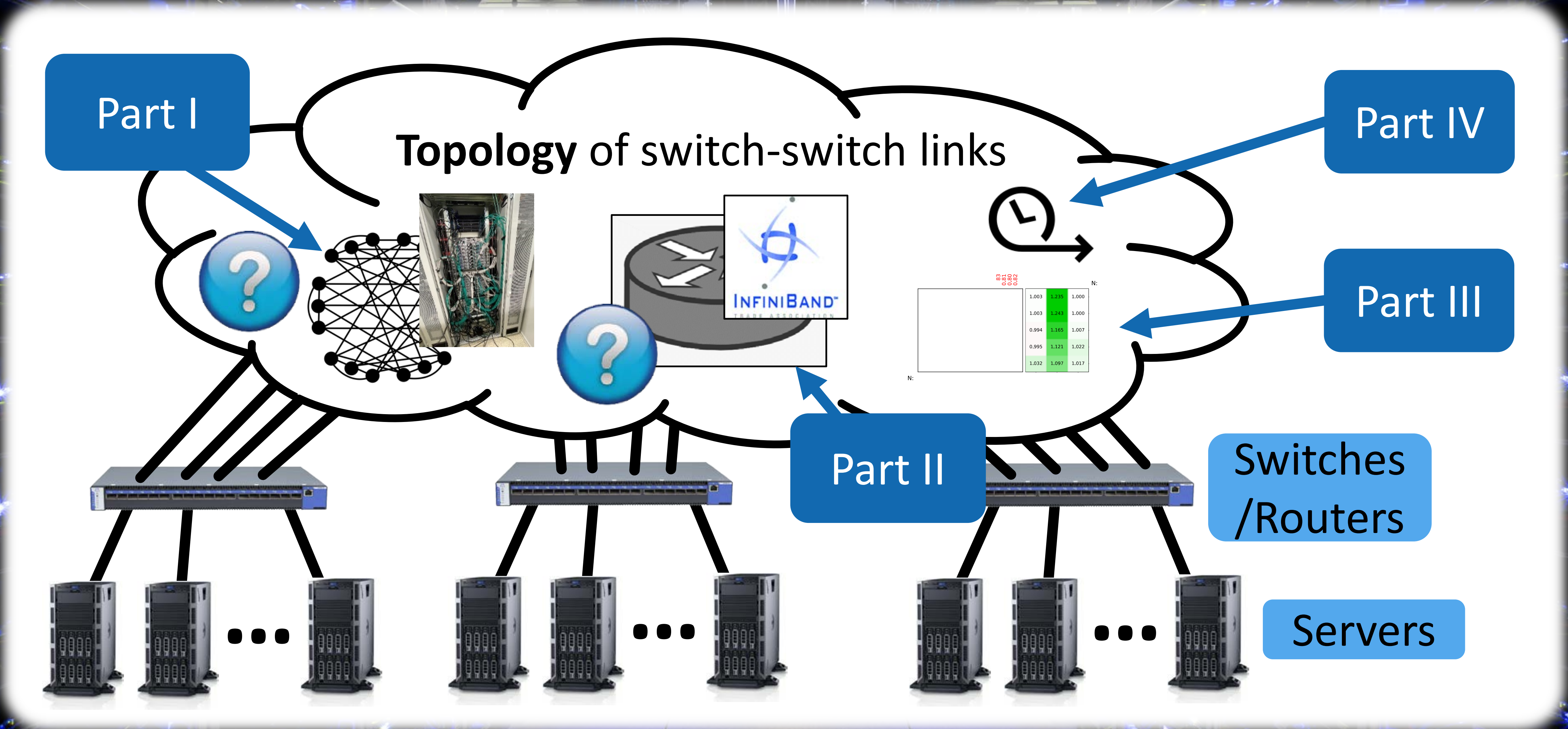
All packets that are in switch 5 that want to reach switch 2 have to take the direct link in this layer (due to IB's destination based routing)



Setting: We are trying to add a non-minimal path to this layer for the switch pair (5, 2), after a path for the pair (1, 2) has been inserted

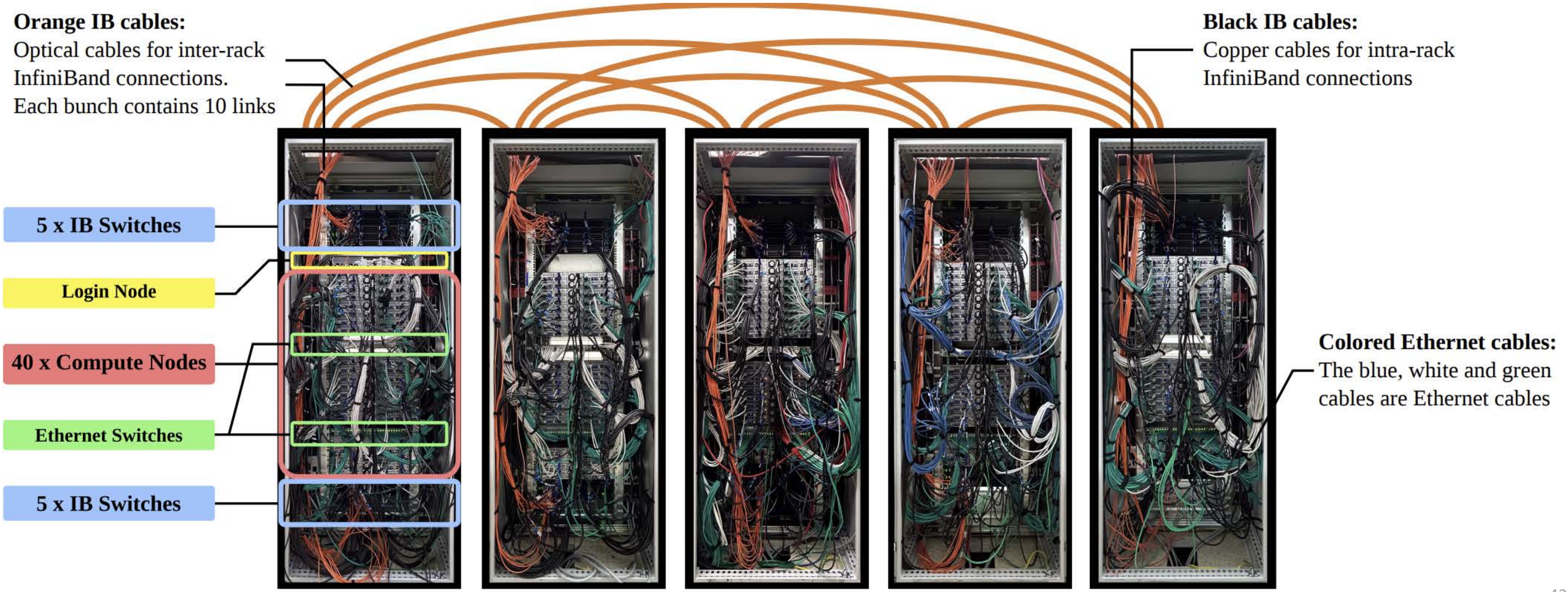
Switch 5	
Destination LID	Next Hop
201	2
...	...

NETWORK TOPOLOGIES : SETTING & PRESENTATION PLAN



Evaluation


GRAPH
500


Orange IB cables:
Optical cables for inter-rack InfiniBand connections. Each bunch contains 10 links
Black IB cables:
Copper cables for intra-rack InfiniBand connections
5 x IB Switches
Login Node
40 x Compute Nodes
Ethernet Switches
5 x IB Switches
Colored Ethernet cables:
The blue, white and green cables are Ethernet cables

Comparison Baselines & Setup

Theoretical analysis

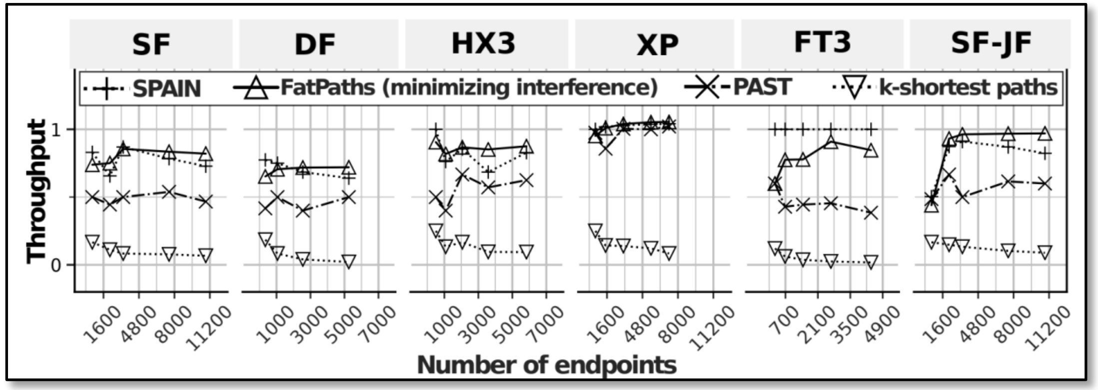
$$-\sum_{v\in V}\sum_{l=1,\dots,n} f_{is_ivl}\cdot\delta_{v,\sigma_l(s_i,t_i)} + T(s_i,t_i)\cdot\mathcal{T} \le 0,\quad i=1,2,\dots,k \qquad (5)$$

$$\sum_{i=1,\dots k}\sum_{l=1,\dots,n} f_{iuvl}\cdot\delta_{v,\sigma_l(u,t_i)} \le c(u,v),\quad \forall(u,v)\in E \qquad (6)$$

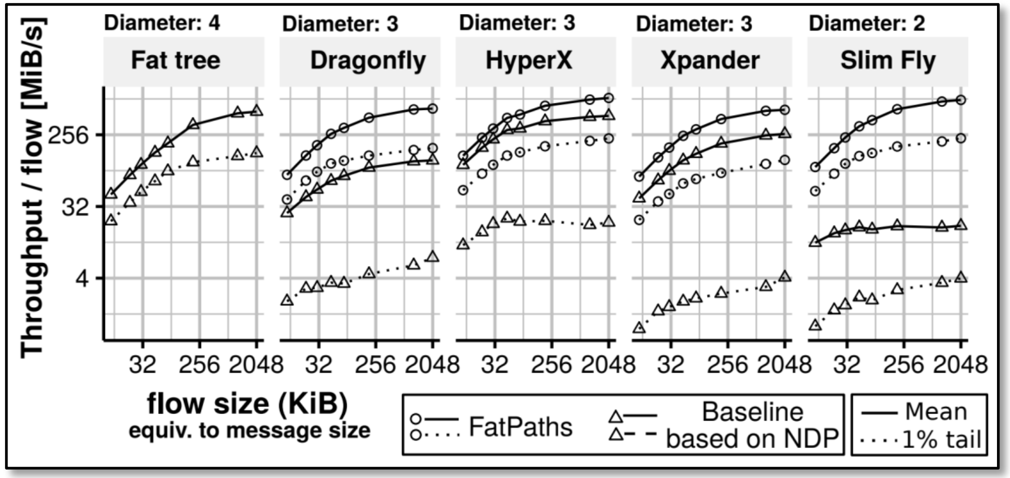
$$\sum_{v\in V} f_{iuvl}\cdot\delta_{v,\sigma_l(u,t_i)} - \sum_{v\in V} f_{ivul}\cdot\delta_{u,\sigma_l(v,t_i)} = 0,\quad i=1,\dots,k,\ \ l=1,\dots,n, \forall u\in V\setminus\{s_i,t_i\} \qquad (7)$$

$$\sum_{v\in V}\sum_{l=1,\dots,n} f_{is_ivl}\cdot\delta_{v,\sigma_l(s_i,t_i)} \le \mathcal{T}_{upperbound}\cdot T(s_i,t_i),\quad i=1,\dots,k \qquad (8)$$

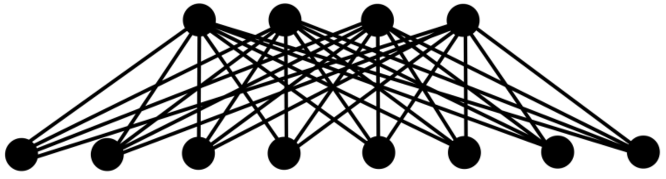
$$\sum_{v\in V}\sum_{l=1,\dots,n} f_{ivs_il}\cdot\delta_{s_i,\sigma_l(v,t_i)} = 0,\quad i=1,\dots,k \qquad (9)$$

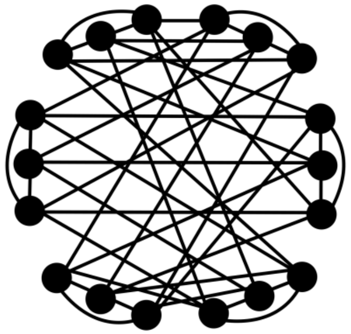


Simulations



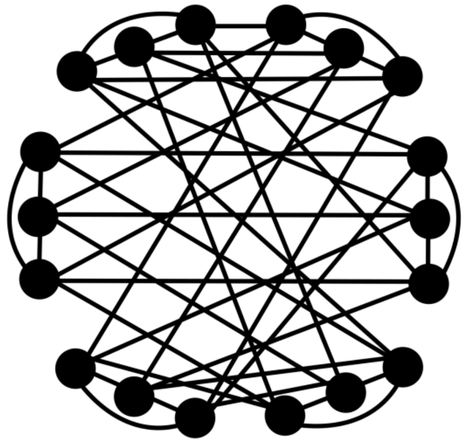
Real testbed comparisons

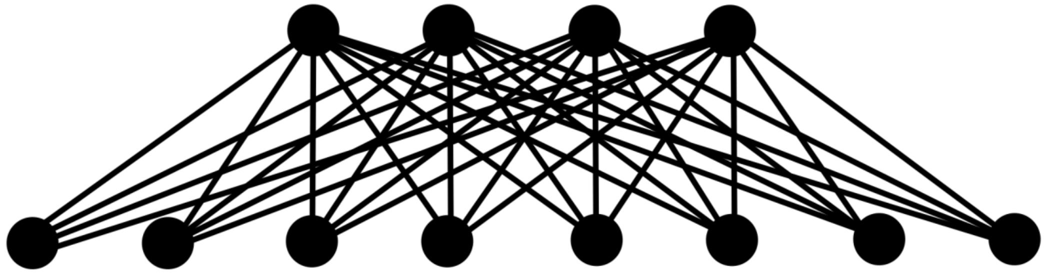




Comparison Baselines & Setup

Networks





Baseline: 2-Level Non-Blocking Fat-Tree constructed using the same hardware

Rank Placement Strategy: Linear for both Slim Fly (SF) and Fat-Tree (FT)

Routing

Baseline: Deadlock-Free SSSP (DFSSSP) [1] routing, a standard IB protocol

[1] J. Domke et al. Deadlock-Free Oblivious Routing for Arbitrary Topologies. IPDPS, 2011.

Microbenchmarks

In most scenarios, SF is comparable to FT

For 8 and 16-node configurations – especially with smaller message sizes – the FT displays marginal advantages.

Reason: FT has 16 nodes per switch (SF: 4), leading to more localized, zero inter-switch hop traffic

MPI Bcast – SF vs. FT

1.01	1.00	0.97	0.96	1.00	0.52	1.01	1.00
1.01	1.00	0.96	0.97	1.00	1.05	1.07	0.95
1.03	1.00	0.98	0.94	1.01	0.98	1.21	1.24
1.02	1.00	0.99	0.84	1.02	0.95	0.81	1.32
1.02	1.01	0.81	0.97	1.07	0.87	0.88	0.94
1.02	1.01	0.82	1.01	1.00	1.05	0.90	0.97
1.03	1.02	0.81	1.02	1.01	0.95	1.07	0.98
1.02	1.00	0.87	1.16	0.95	0.95	0.95	1.00
1.02	1.01	1.00	1.00	0.99	0.96	0.90	1.41
1.04	1.03	0.97	1.00	1.09	1.01	1.03	1.09
1.03	1.00	0.99	0.94	0.99	0.96	0.95	0.99
1.05	1.06	1.03	0.95	0.96	1.02	0.97	1.00
1.01	1.01	0.98	0.93	1.01	1.00	1.01	1.00
1.00	1.01	1.00	0.92	0.97	1.00	0.98	0.99
0.98	1.01	0.98	0.90	0.97	0.97	1.37	1.00
1.03	1.01	0.95	0.91	0.97	1.01	0.99	1.02
1.01	1.03	0.96	0.90	0.96	1.01	0.97	1.03
1.06	1.02	0.96	0.89	0.96	1.01	0.98	1.06
1.02	1.00	0.99	0.89	0.97	1.03	0.98	1.03
0.97	1.03	0.94	0.85	0.96	1.00	1.02	1.07
1.01	1.02	0.94	0.89	0.96	1.03	1.01	1.02
1.00	1.02	0.98	0.89	0.98	1.02	1.02	1.06
1.00	1.04	0.96	0.87	0.97	1.00	1.00	1.06
0.99	1.05	0.95	0.86	1.00	1.03	0.98	1.05
0.98	1.02	0.93	0.85	0.98	1.04	1.02	1.02
1.00	0.98	0.96	0.86	0.98	1.03	1.01	1.02

N:

Nodes

MPI Allreduce – SF vs. FT

1.05	1.11	1.07	1.05	1.03	0.67	0.71	2.95
1.03	1.14	1.10	1.09	1.07	0.71	0.79	2.64
0.89	1.05	1.02	1.02	1.08	0.72	0.78	2.73
0.90	1.02	1.03	1.01	1.06	0.82	0.87	3.10
1.01	1.12	1.03	1.03	1.06	0.90	0.97	0.95
1.00	1.02	1.01	0.90	1.07	0.97	1.06	0.98
1.01	0.98	1.02	0.94	1.02	1.08	1.07	1.04
1.04	1.01	1.01	0.91	1.01	1.02	1.04	1.00
1.01	1.01	0.95	1.00	1.00	1.00	0.98	0.99
1.00	1.01	0.99	0.98	1.00	0.97	0.98	1.08
1.03	1.01	1.01	1.01	0.98	0.96	0.98	0.99
0.99	1.00	0.98	1.00	1.00	0.97	0.97	0.97
1.01	1.05	0.93	0.84	0.84	0.89	0.86	1.29
1.09	1.06	0.95	0.95	0.93	0.97	0.98	1.00
0.95	1.01	0.99	0.94	0.97	1.02	1.03	1.00
0.95	1.03	0.97	0.93	1.06	1.04	1.01	1.01
0.93	1.01	0.99	0.93	1.05	0.98	1.05	1.01
0.94	0.87	0.97	0.90	1.03	1.01	1.02	1.07
1.19	1.12	1.09	0.89	1.02	1.07	1.00	1.02
1.08	1.01	1.06	0.94	1.00	1.03	1.02	1.04
1.11	0.99	1.01	0.90	0.99	1.02	1.02	1.00
1.34	0.95	1.02	0.99	0.94	0.98	1.03	1.02
1.28	1.02	0.92	0.93	0.97	1.10	1.04	1.01
1.07	1.10	0.91	0.89	1.02	1.04	0.97	1.02

N:

Nodes

Microbenchmarks

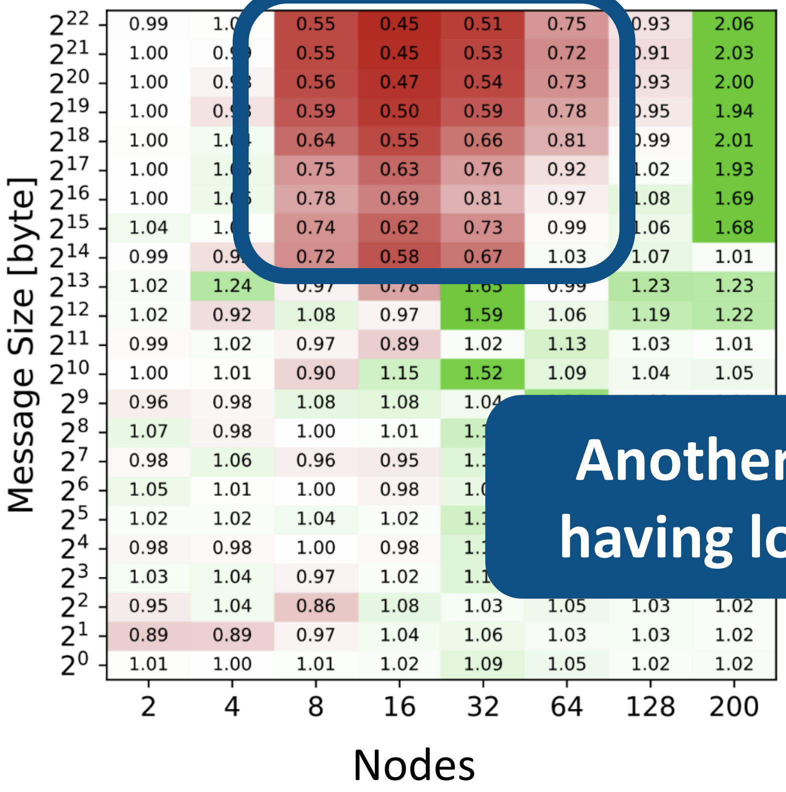
Traffic congestion on the (often) single shortest path between a few switches.

HW's lack of adaptive load balancing support (which we do enable in the protocols) limits its practical improvement

Random placement for SF, overcomes this bottleneck

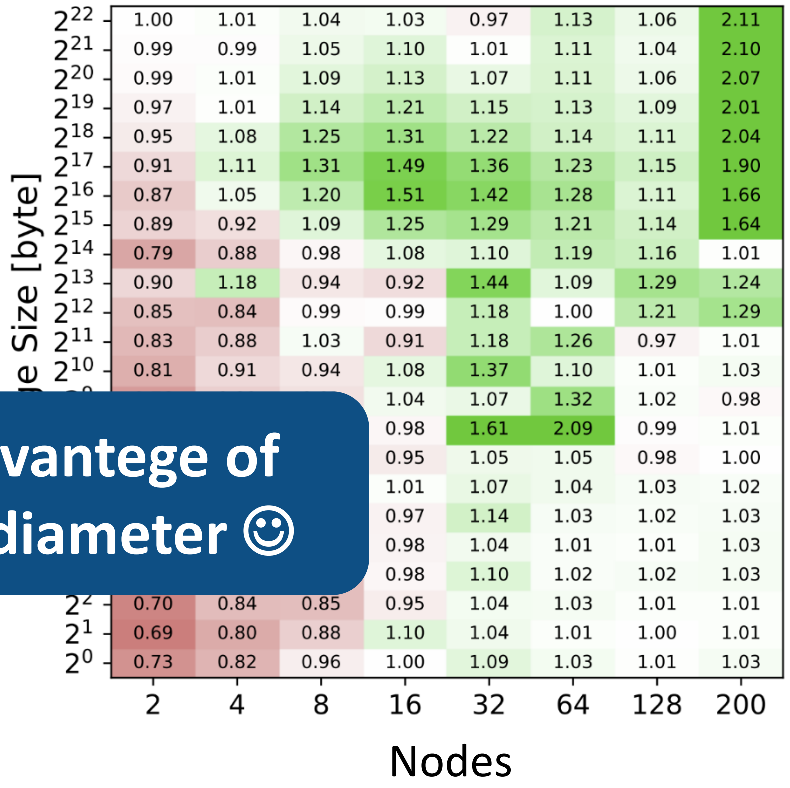
Linear placement in SF

MPI Alltoall – SF vs. FT



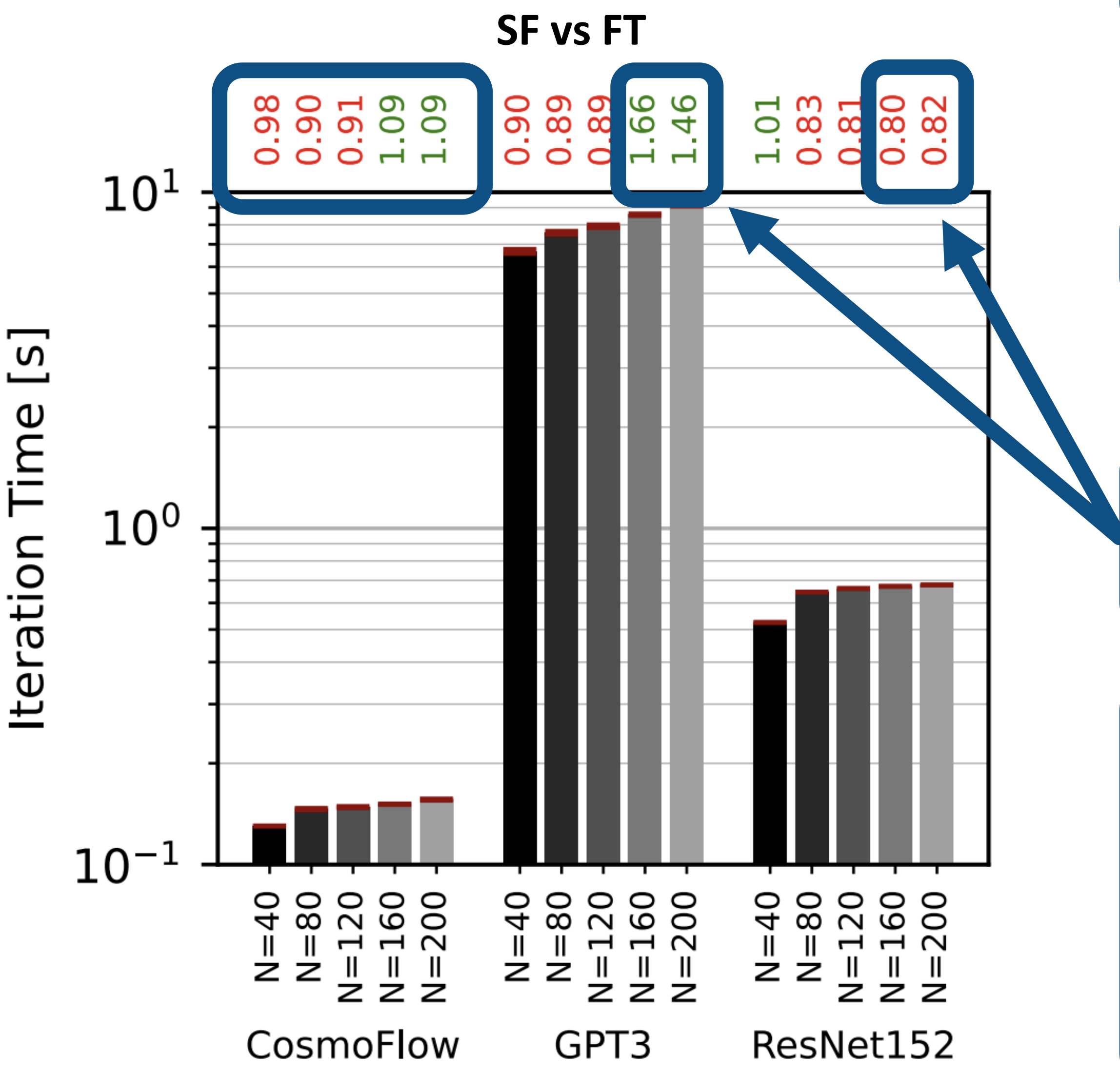
Random placement in SF

MPI Alltoall – SF vs. FT



Another advantege of having low diameter ☺

Deep Learning Proxy Workloads



CosmoFlow: Data + operator parallelism, requires allgather, reduce-scatter, allreduce, and point-to-point

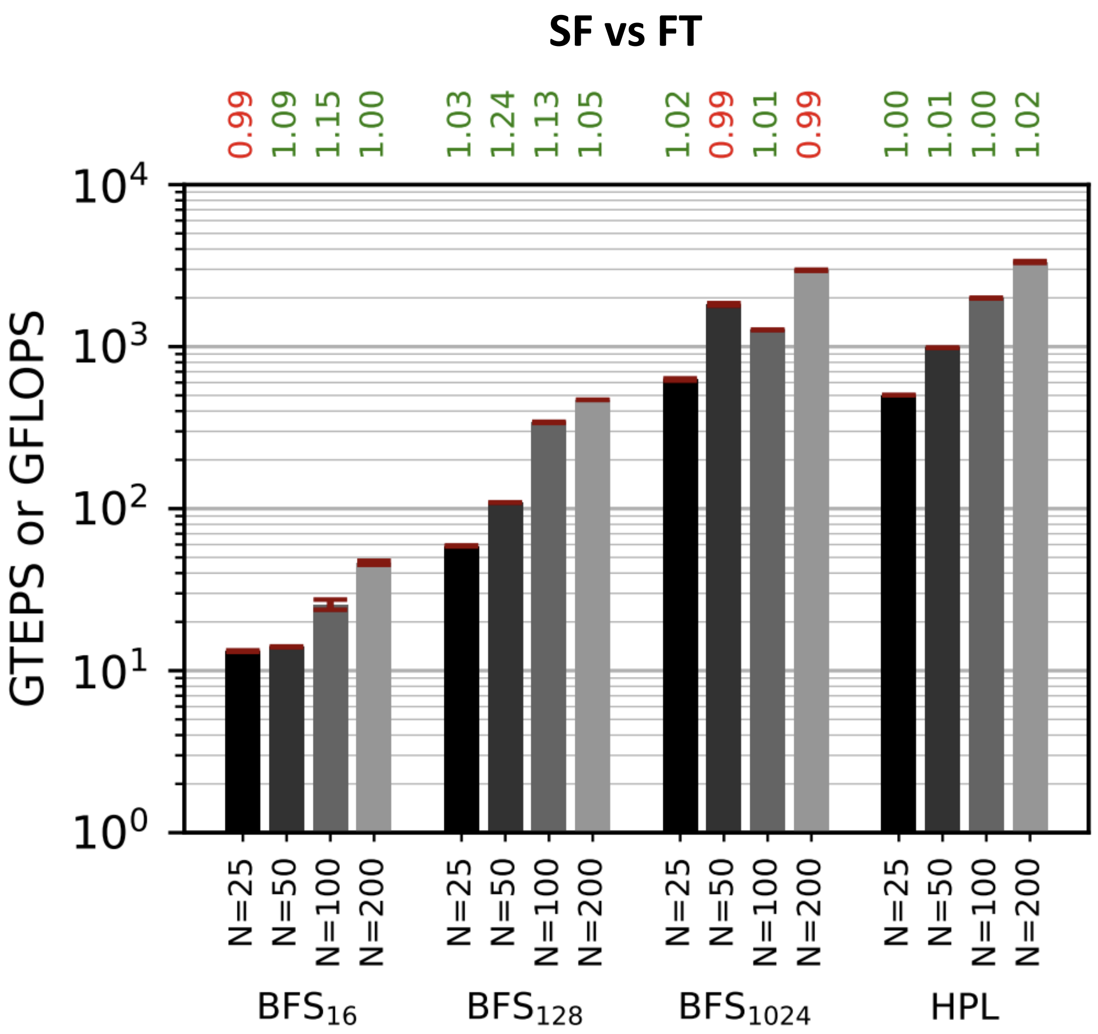
ResNet152: Pure data parallelism, only requires allreduce

GPT-3 : Data + operator + pipeline parallelism, requires allreduce and point-to-point

Both GPT-3 and ResNet152 predominantly rely on allreduces at higher node counts...

...but GPT-3 handles significantly larger messages than ResNet152. Expectedly, the performance trend of GPT-3 matches the trend of MPI Allreduce for the high node count configurations

HPC Benchmarks

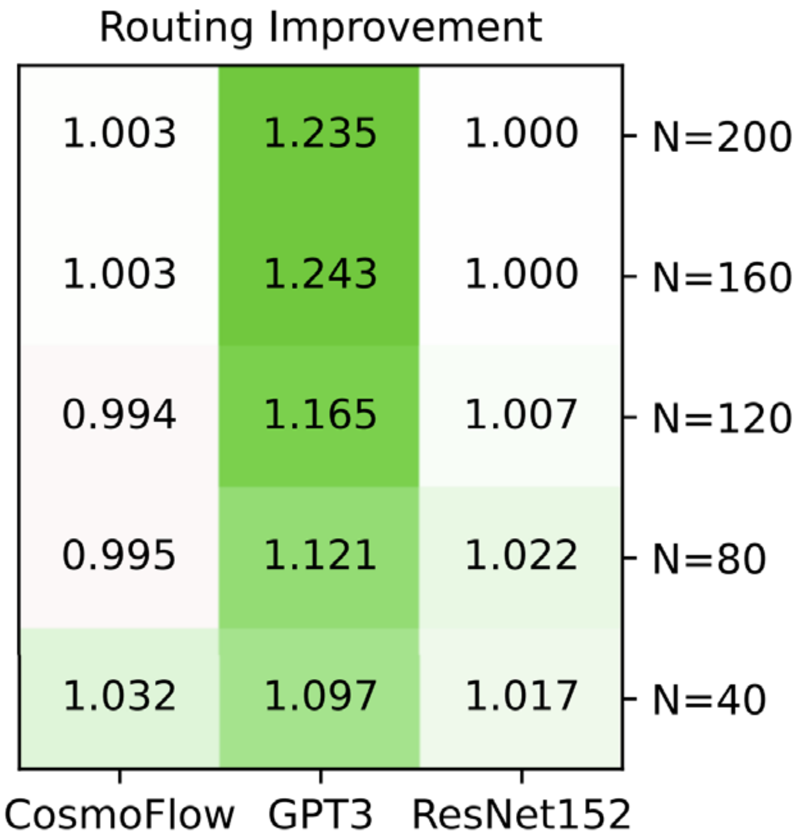


SF competes effectively with FT in terms of performance

SF is effective for scaling HPC benchmarks

Routing Improvements

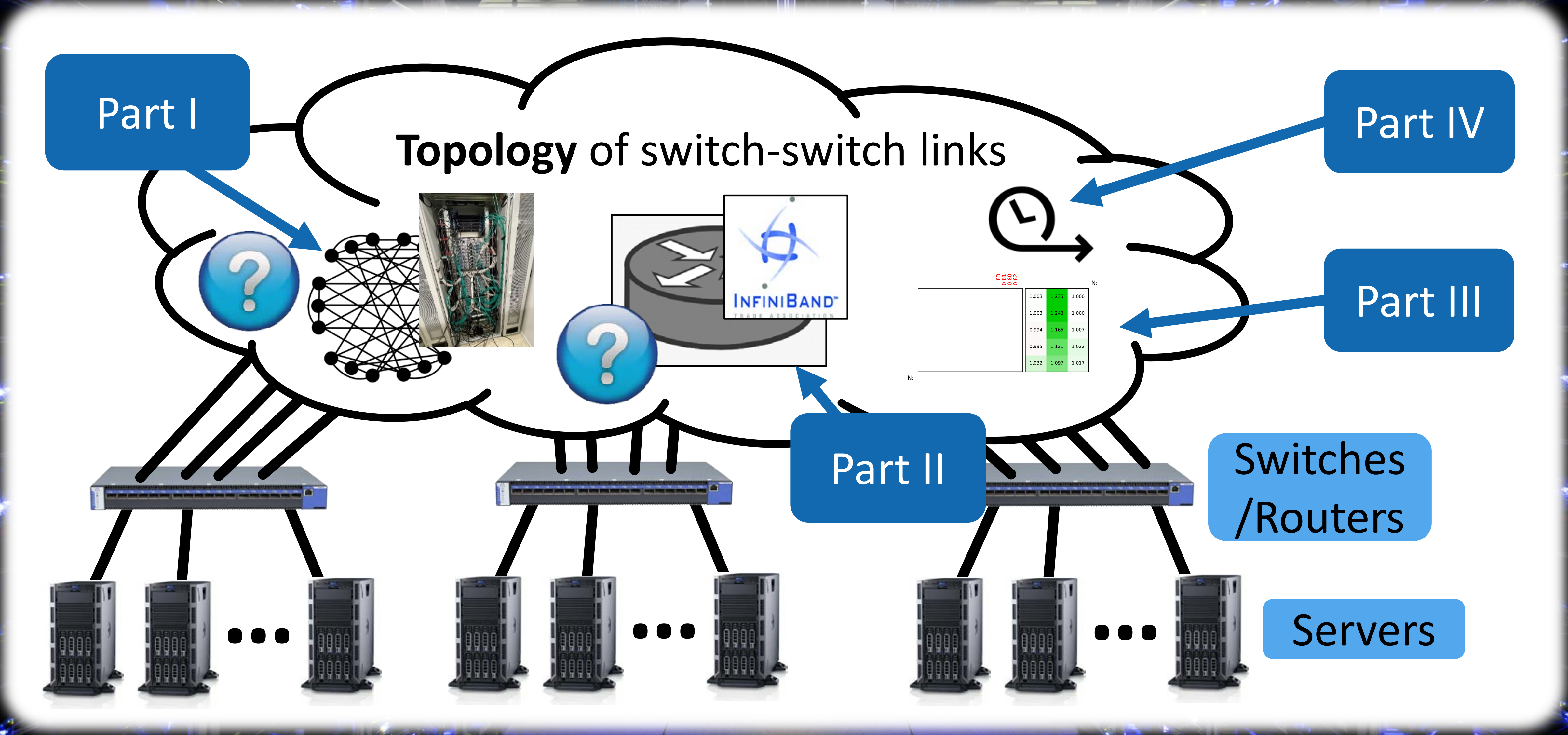
Layered Routing vs DFSSSP



Layered Routing outperforms DFSSSP

This is thanks to the multi-pathing support (despite not being able to leverage adaptivity)

NETWORK TOPOLOGIES : SETTING & PRESENTATION PLAN



Cluster Use Cases, Research Outcomes, & Future Potential

Past

Swing: Short-cutting Rings for Higher Bandwidth Allreduce

Daniele De Sensi, *Sapienza University of Rome*
Tommaso Bonato, *ETH Zurich*
David Saam, *RWTH Aachen University*
Torsten Hoefler, *ETH Zurich*

HammingMesh: A Network Topology for Large-Scale Deep Learning

Torsten Hoefler[*†], Tommaso Bonato[*], Daniele De Sensi[*], Salvatore Di Girolamo[*], Shigang Li[*], Marco Heddes[†], Jon Belk[†], Deepak Goel[†], Miguel Castro[†], and Steve Scott[†]

@ SC'22, Reproducibility Advancement Award, Invited as CACM Research Highlight

Congestion Benchmarking and Visualization of Large-Scale Interconnection Networks

@ NSDI'24

A High-Performance Design, Implementation, Deployment, and Evaluation of The Slim Fly Network

Nils Blach[*], Maciej Besta[*], Daniele De Sensi[*,◇], Jens Domke[†], Hussein Harake[§], Shigang Li[*], Patrick Iff[*], Marek Konieczny[¶], Kartik Lakhotia[‖], Ales Kubicek[*], Marcel Ferrari[*], Fabrizio Petrini[‖], Torsten Hoefler[*]

...

Present/Future

- Foundations of performance measures for interconnects
- Testing SOTA interconnects (HammingMesh, PolarFly, PolarStar)
- Enabling cheap computations by filling idleness gaps on HPC systems („ HPC for Free").
- Testing new batch scheduler policies, new paradigms, etc.
- ...

Bandwidth-Optimal, Fully-Offloaded Collectives

Conclusions

More of SPCL's research:

- youtube.com/@spcl — 180+ Talks
- twitter.com/spcl_eth — 1.4K+ Followers
- github.com/spcl — 3.8K+ Stars

… or spcl.ethz.ch