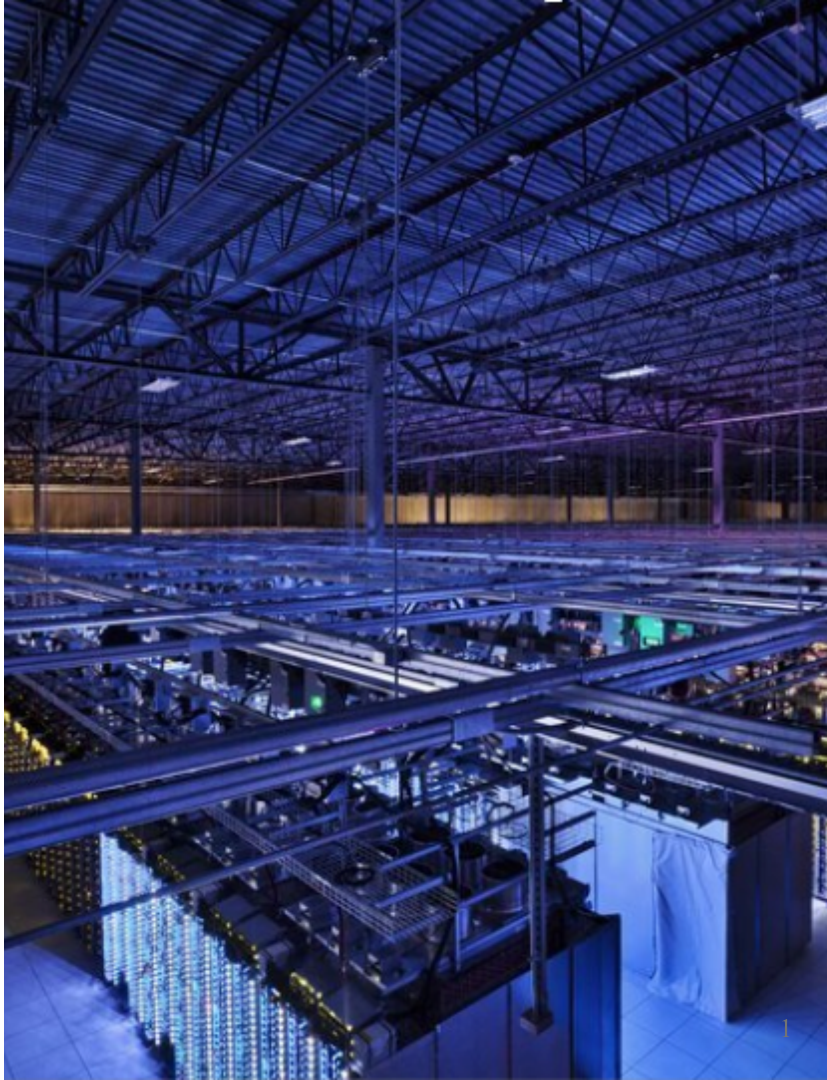
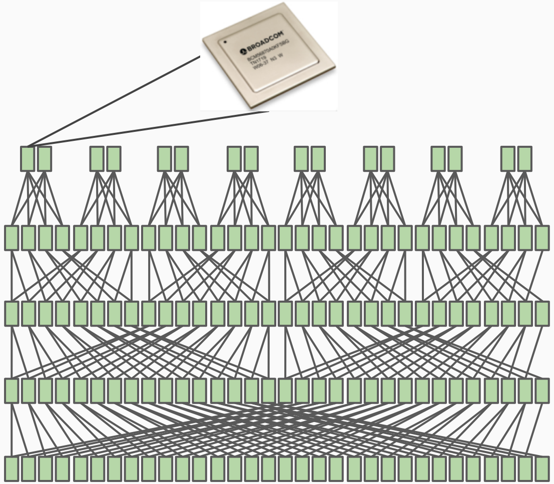
Understanding **Lifecycle Management** Complexity of Datacenter Topologies

Mingyang Zhang (USC)
Radhika Niranjan Mysore (VMware Research)
Sucha Supittayapornpong (USC)
Ramesh Govindan (USC)

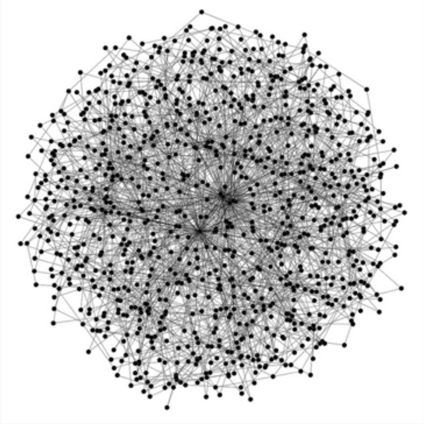
USC Viterbi School of Engineering
vmware



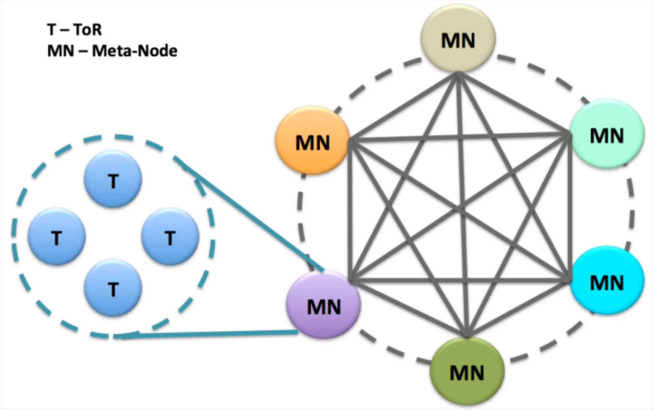
Datacenter topology designs



5-layer Clos



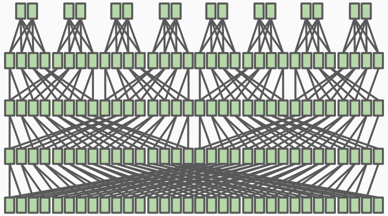
Jellyfish [NSDI12]



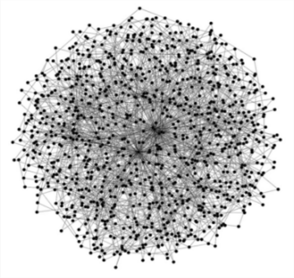
Xpander [CoNEXT16]

Previous focus

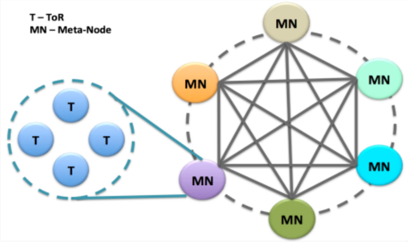
Clos

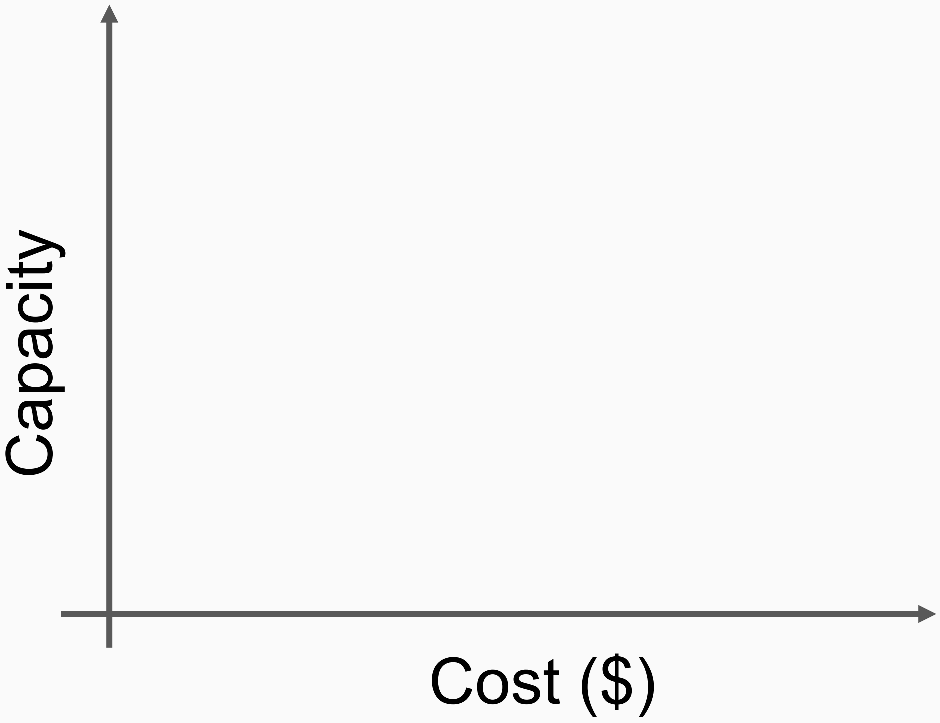


Jellyfish



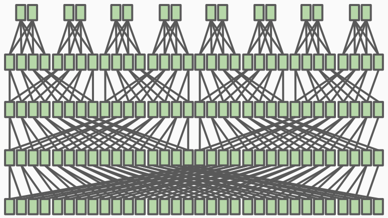
Xpander



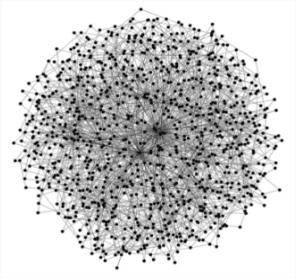


Manageability has received very little attention!

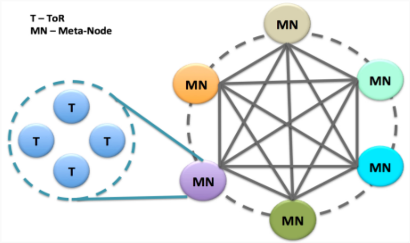
Clos

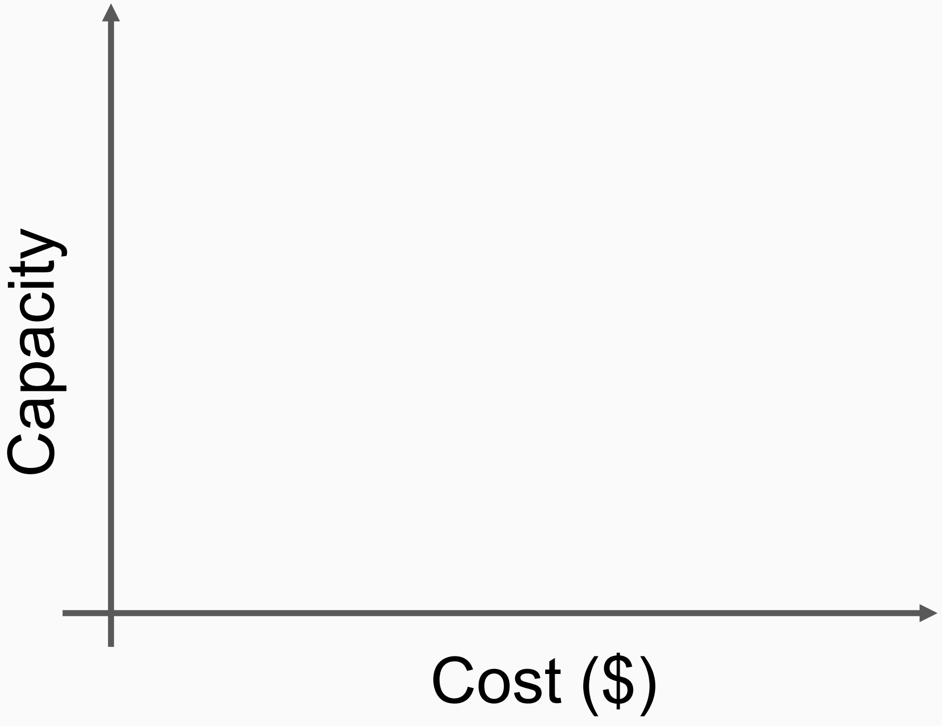


Jellyfish



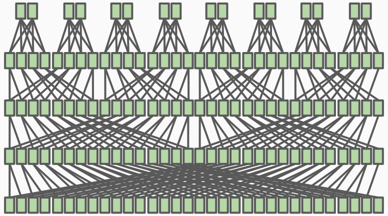
Xpander



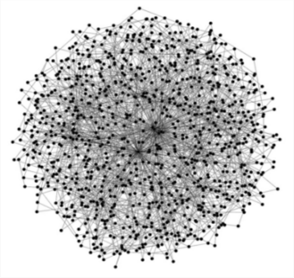


Manageability has received very little attention!

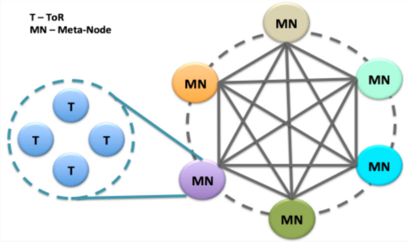
Clos

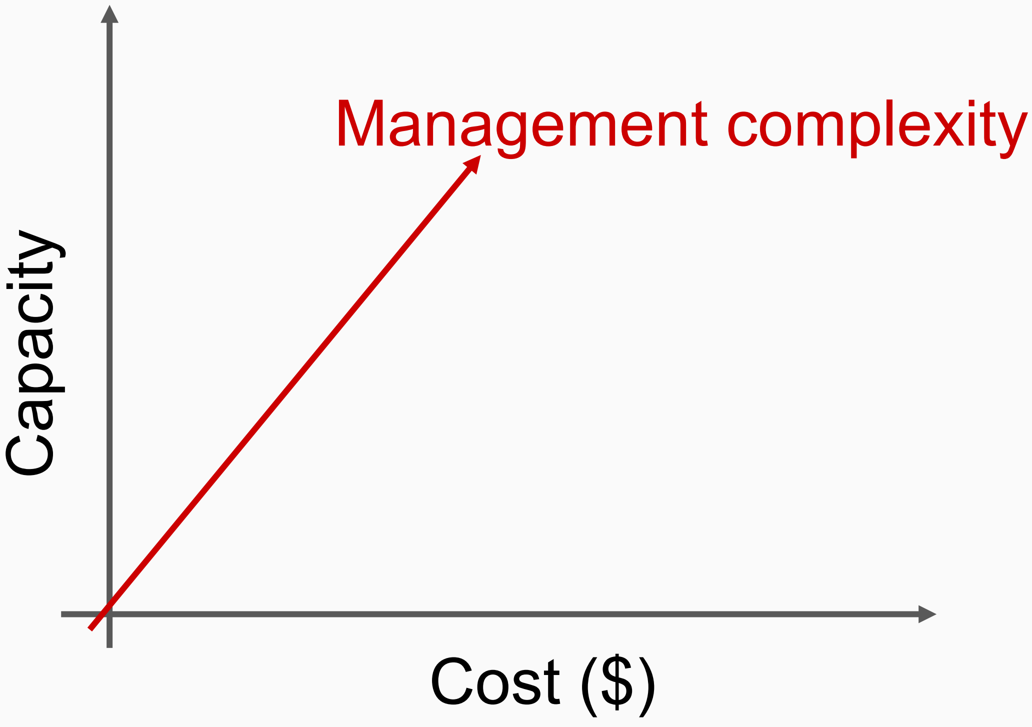


Jellyfish



Xpander

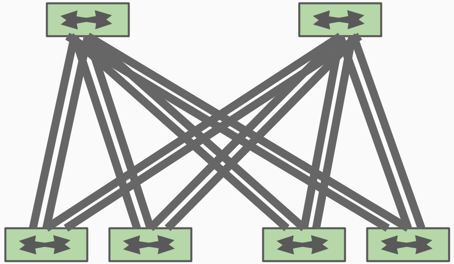




How does the complexity of managing data centers depend on the topology?

Our Focus: Lifecycle management

Lifecycle management of datacenter topologies

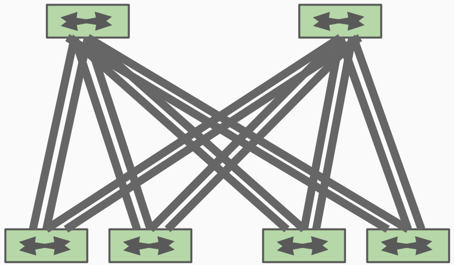


Logical topology

Deployment

Physical topology

Lifecycle management of datacenter topologies

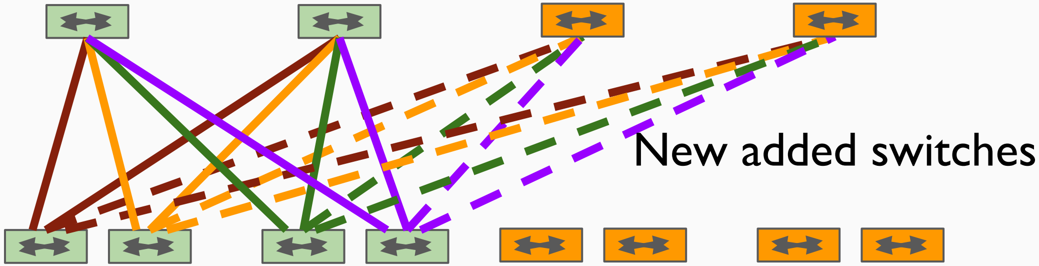


Logical topology

Deployment

Physical topology

Expansion

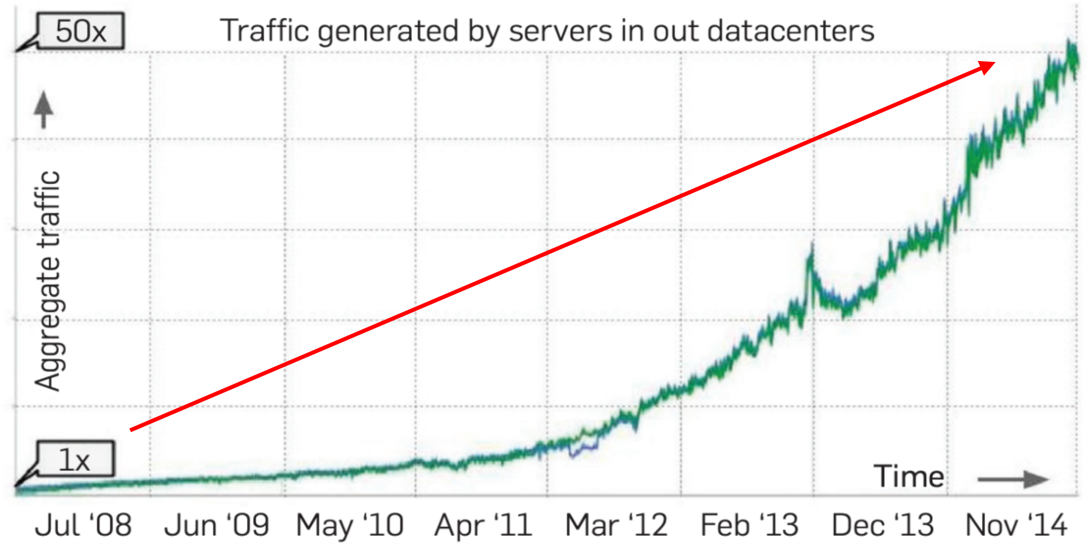


Management complexity is important

– Complex deployment stalls the rollout of services for a long time

Management complexity is important

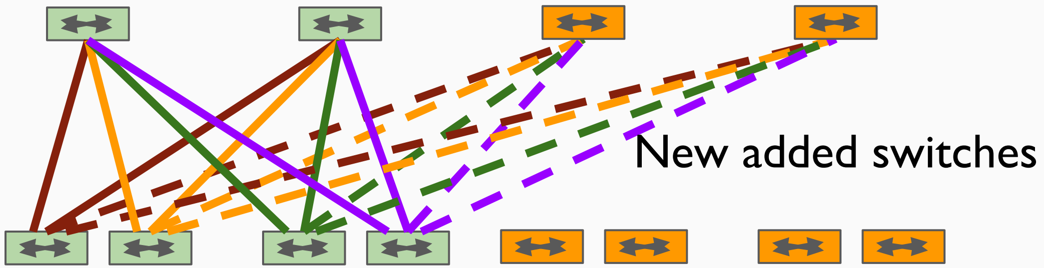
- Complex deployment stalls the rollout of services for a long time
- Expensive considering the increasing traffic demand



From Singh et al. Sigcomm15

Management complexity is important

- Topology expansion leads to capacity drop due to rewiring
- Complex expansion leads to degraded capacity for a long time



Challenges

Contributions

Challenges

How to characterize the management complexity?

Contributions

Metrics

- Deployment
- Expansion

Challenges

How to characterize the management complexity?

How does topology structure affect the management complexity?

Contributions

Metrics

- Deployment
- Expansion

Comparison of topologies

- No topology dominates
- Principles learned

Challenges

How to characterize the management complexity?

How does topology structure affect the management complexity?

Is there a topology family with lower management complexity, lower cost and high capacity?

Contributions

Metrics
- Deployment
- Expansion

Comparison of topologies
- No topology dominates
- Principles learned

New topology
- **FatClique**

Challenges

How to characterize the management complexity?

How does topology structure affect the management complexity?

Is there a topology family with lower management complexity, lower cost and high capacity?

Contributions

Metrics
- Deployment
- Expansion

Comparison of topologies
- No topology dominates
- Principles learned

New topology
- **FatClique**

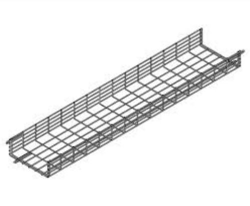
Lifecycle management overview

– Problems: packaging, wiring, placement, rewiring...
– Constraints: switch, rack, patch panel, cable tray...

Broadcom Trident 3

Rack

Optical patch panel



Cable tray

Methodology

From first principles

- Understand **in detail** how topologies are deployed and expanded
- Derive metrics that capture the complexity of these operations

Challenges

How to characterize the management complexity?

How does topology structure affect the management complexity?

Is there a topology family with lower management complexity, lower cost and high capacity?

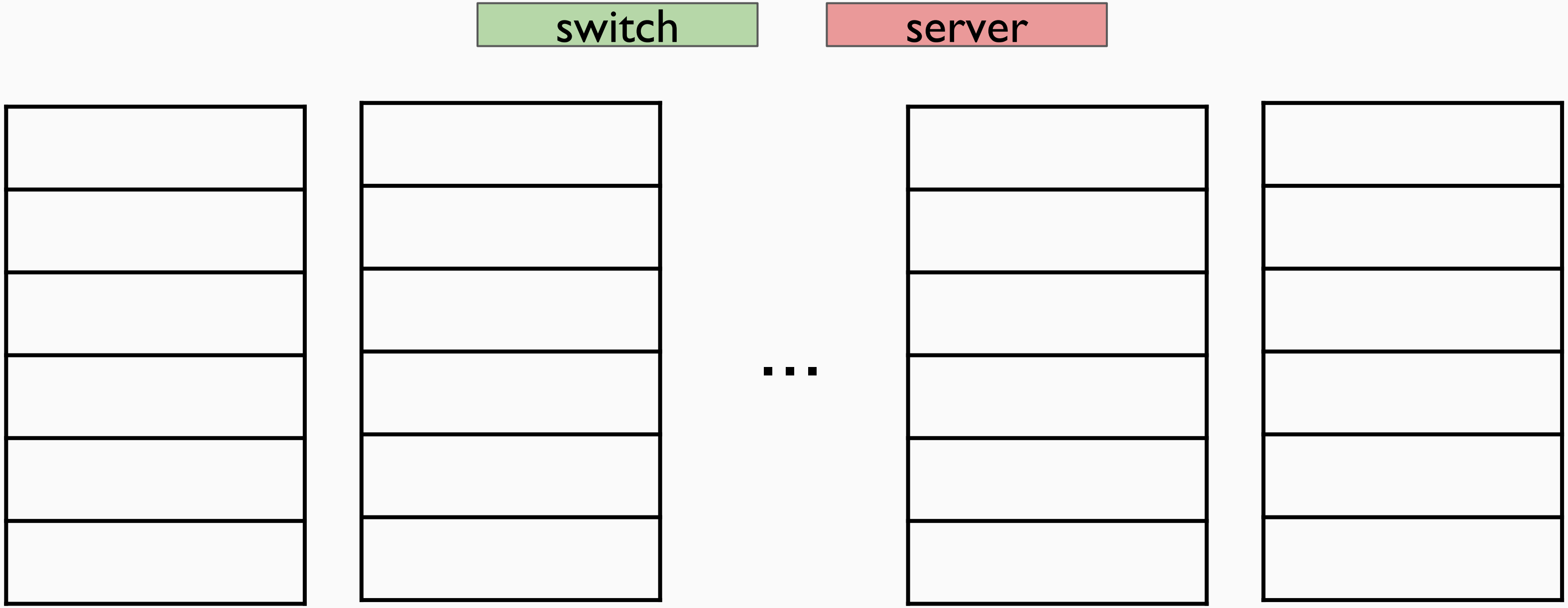
Contributions

Metrics
- **Deployment**
- Expansion

Comparison of topologies
- No topology dominates
- Principles learned

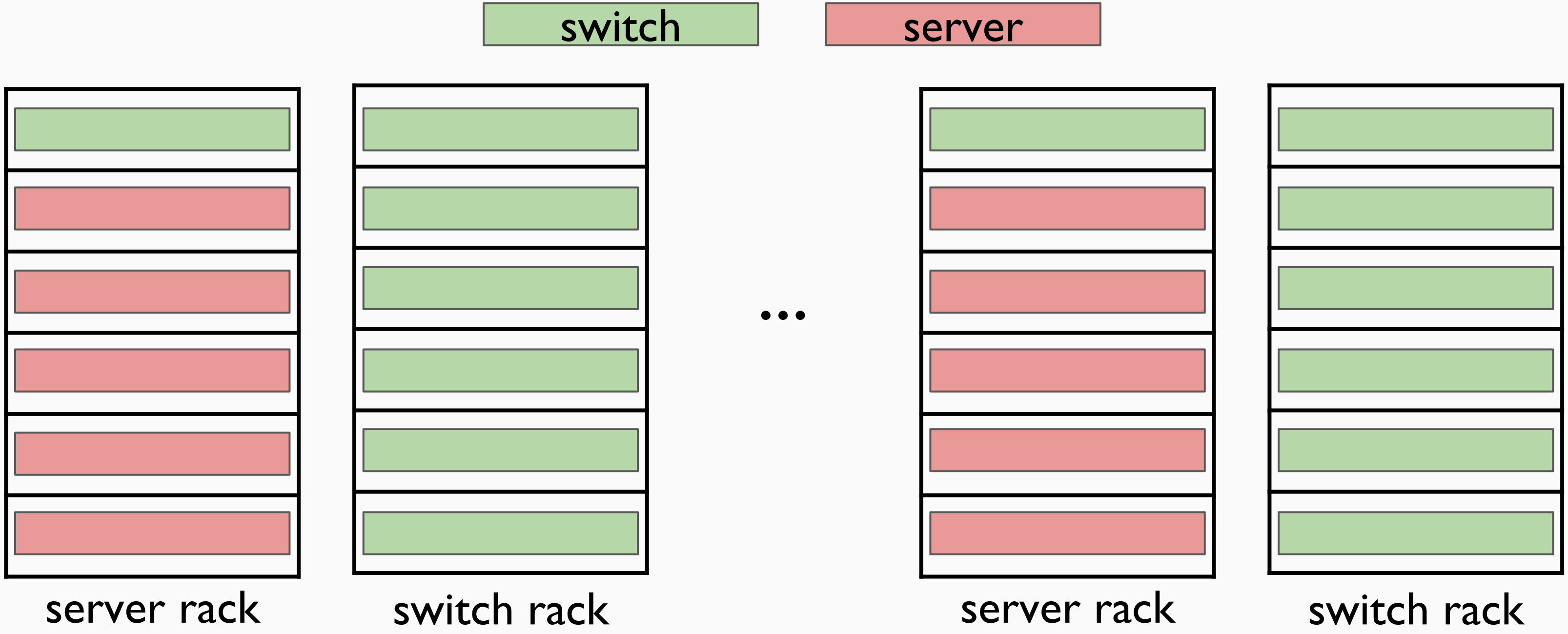
New topology
- **FatClique**

Packaging Deployment



data center racks

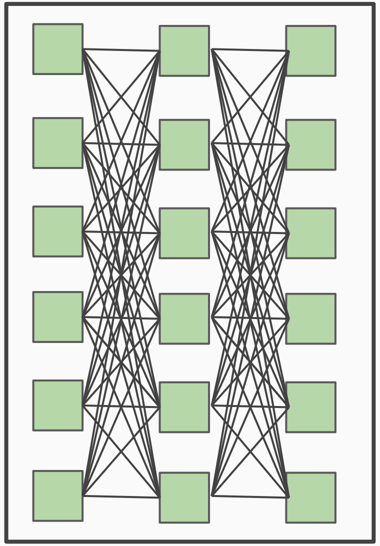
Packaging Deployment



Metric: number of switches

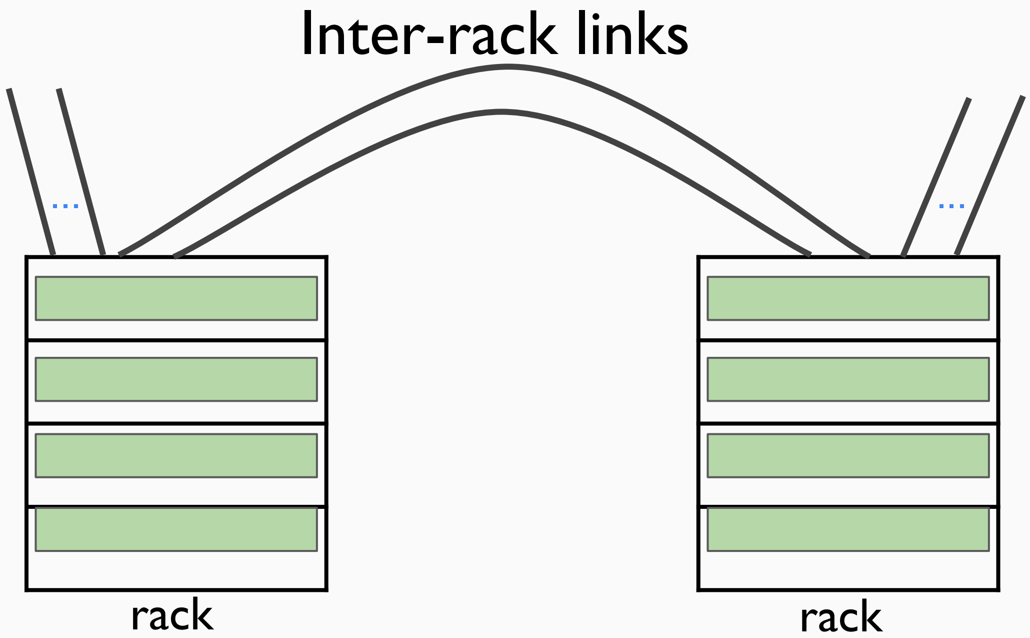
Wiring

Intra-rack links: short and cheap



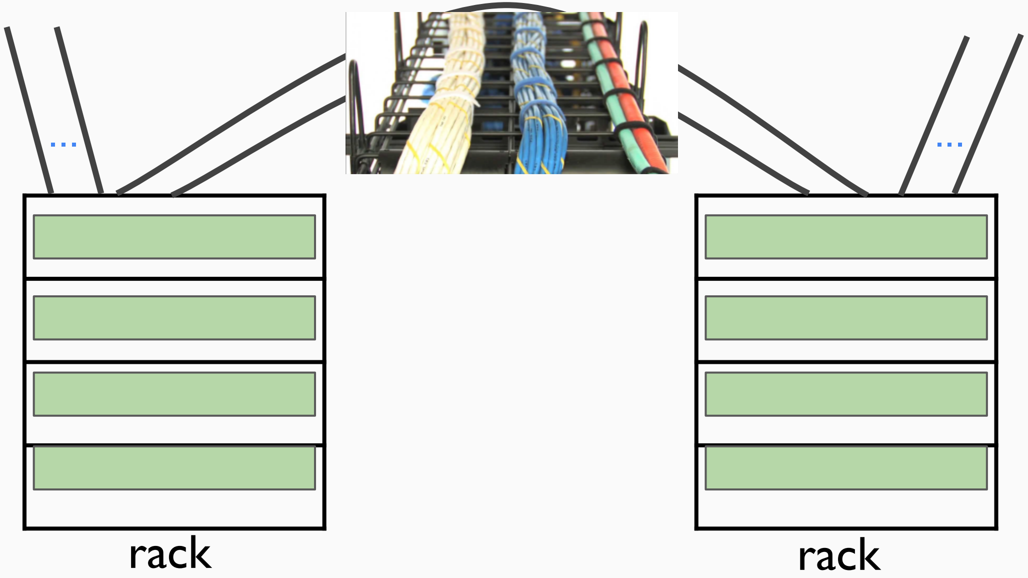
switch rack

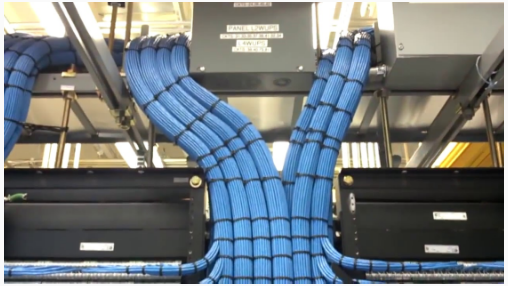
Wiring complexity



Wiring

Inter-rack links over cable trays (expensive)

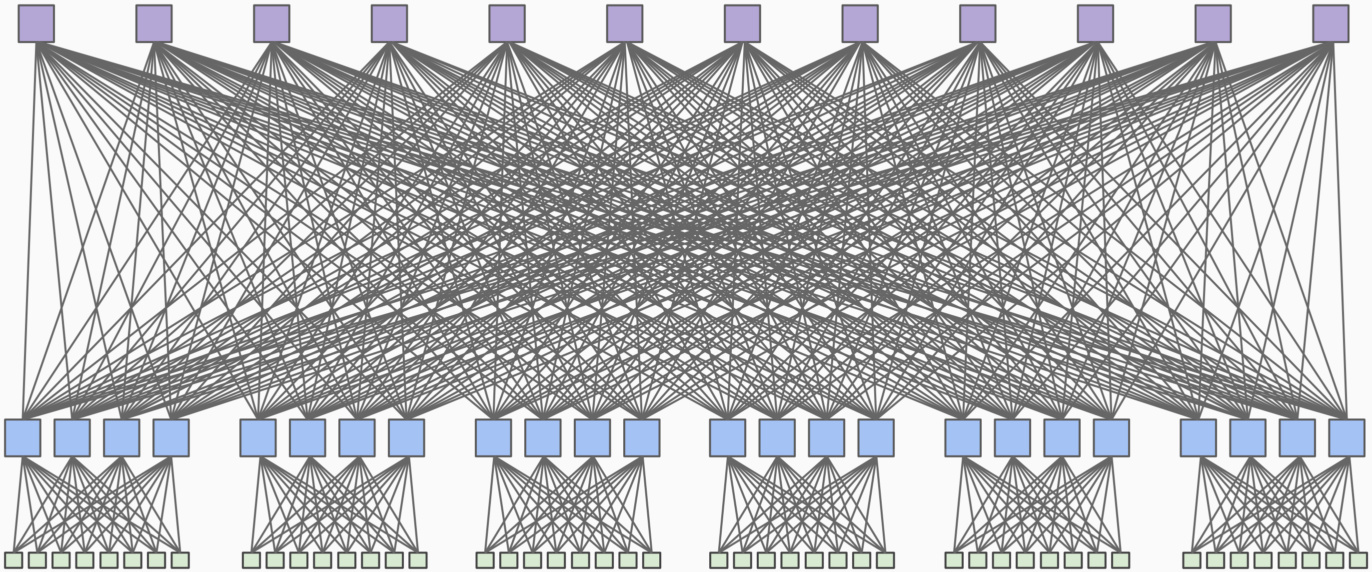




Main wiring complexity comes from inter-rack links!

Cable bundling

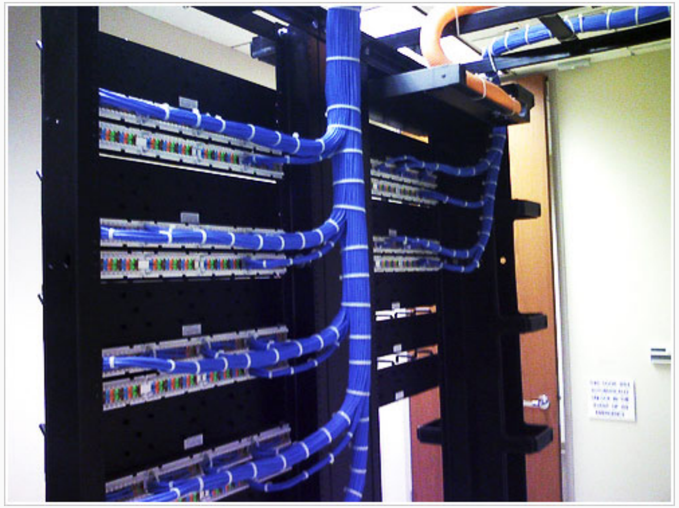
Too many fibers to be handled individually!



Cable bundling

Cable bundle

- a **fixed** number of **identical-length** fibers between two clusters of network devices.



Bundle type

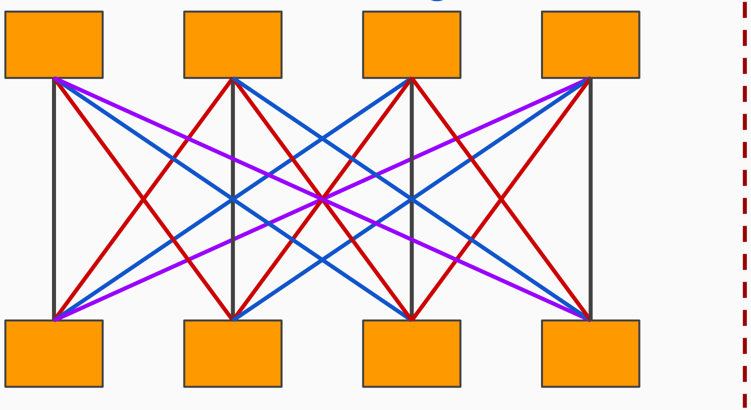
- capacity (# fibers in a bundle)
- length

Cable bundling

Deployment

Bundle type: (bundle capacity, bundle length)

w/o bundling



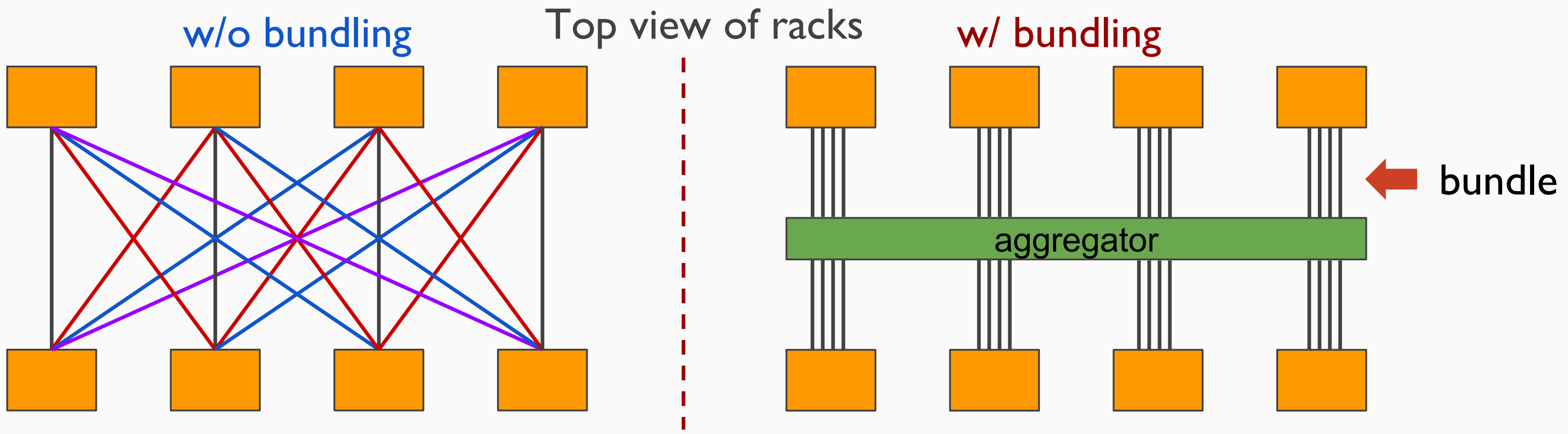
Top view of racks

16 individual fibers, 4 types of length

Cable bundling

Deployment

Bundle type: (bundle capacity, bundle length)



16 individual fibers, 4 types of length

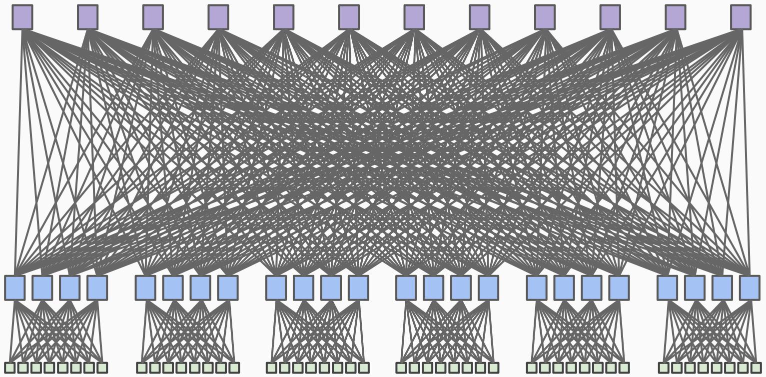
8 equal-length bundles, 1 bundle type

Metric: the number of bundle types

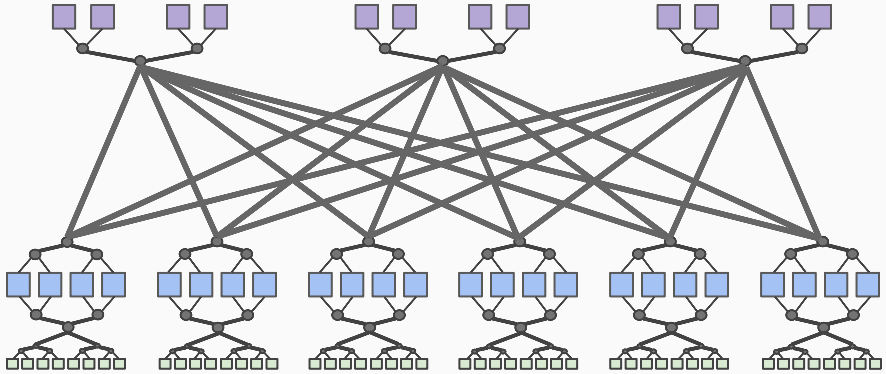
Cable bundling

Deployment

It is hard to handle individual fibers with various length!



w/o bundling

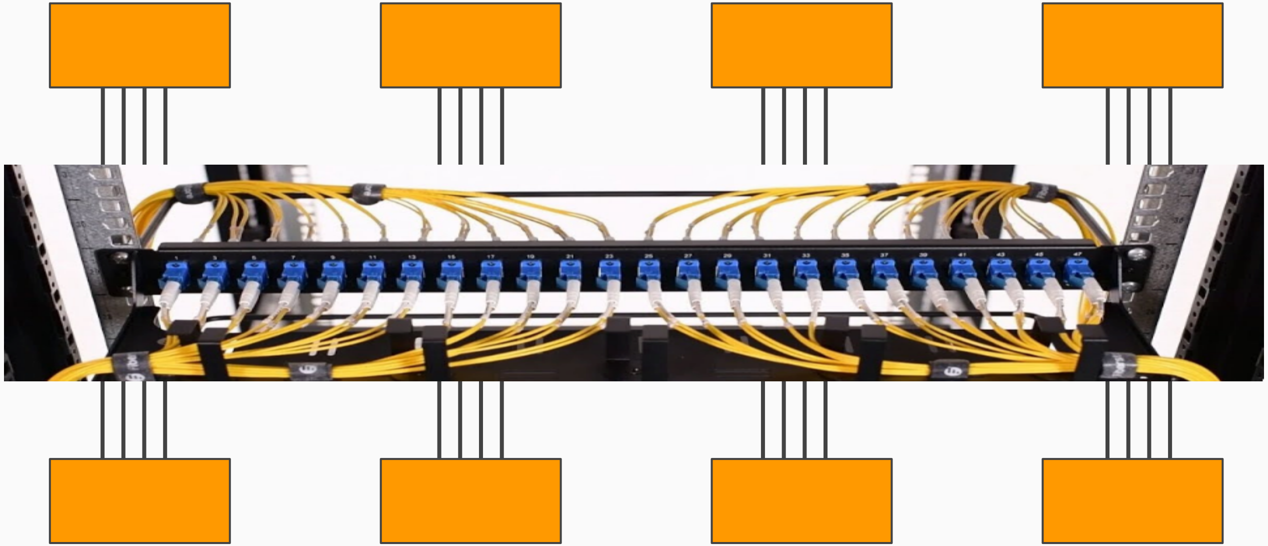


w/ bundling

[Singh, et al. Sigcomm15]

Role of patch panel in bundling

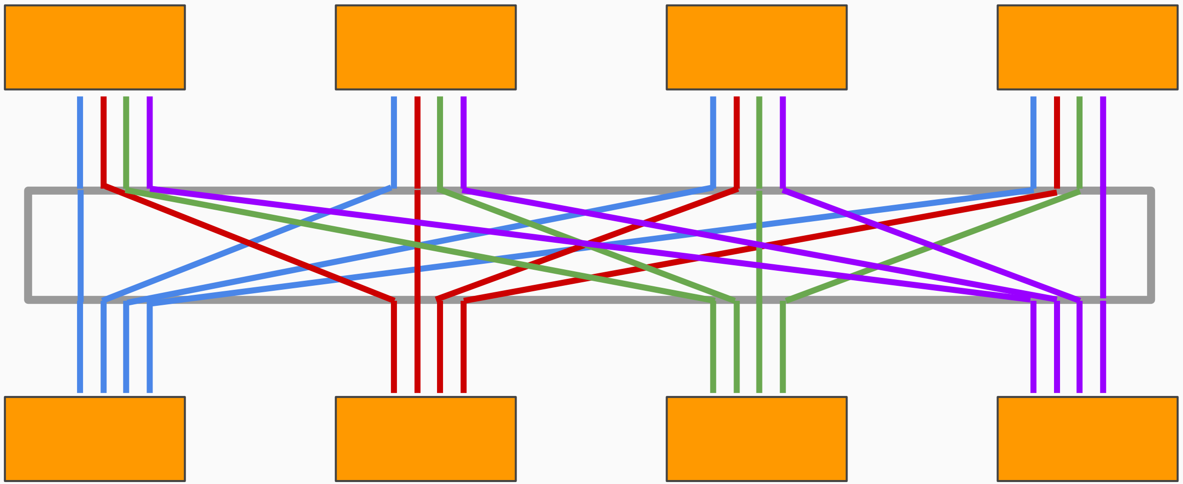
Aggregator: Patch panel



Role of patch panel in bundling

Deployment

Aggregator: Patch panel



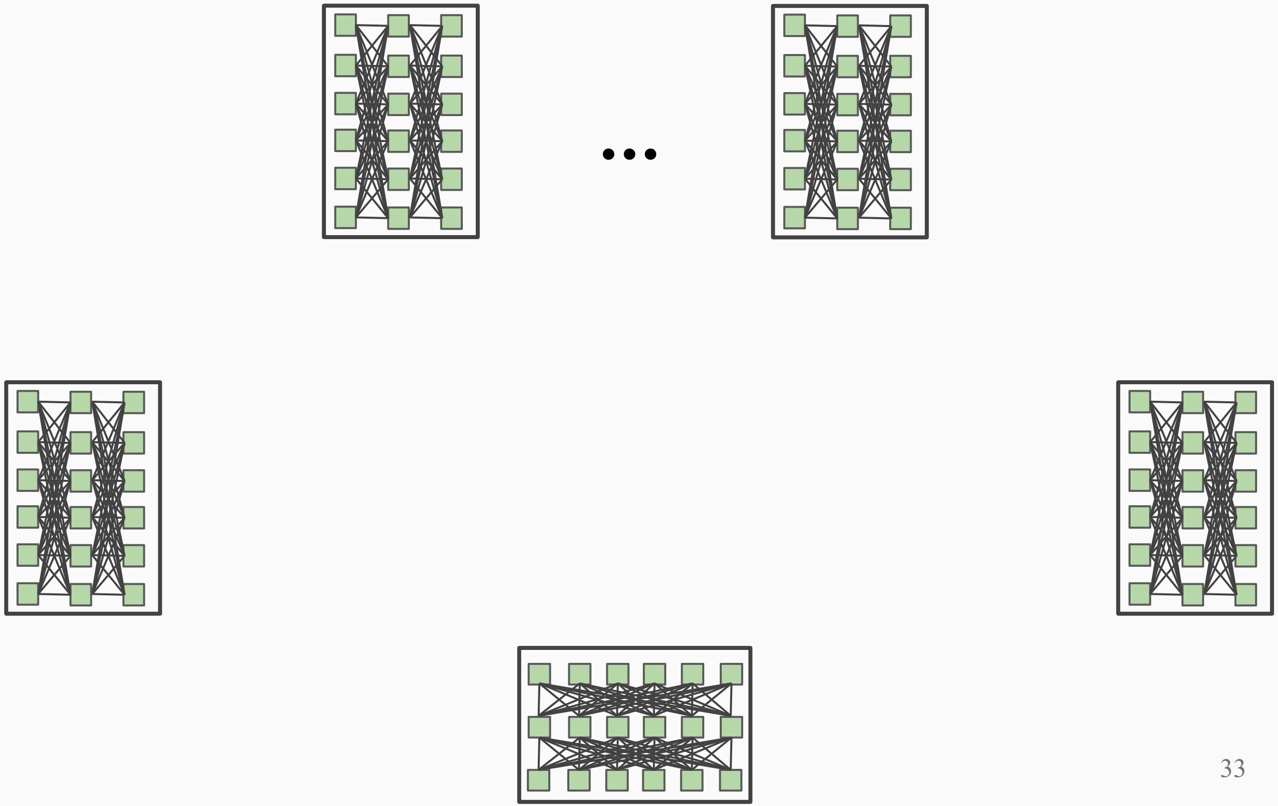
Manual process

Metric: the number of patch panels

Deployment complexity metrics

Deployment complexity metrics

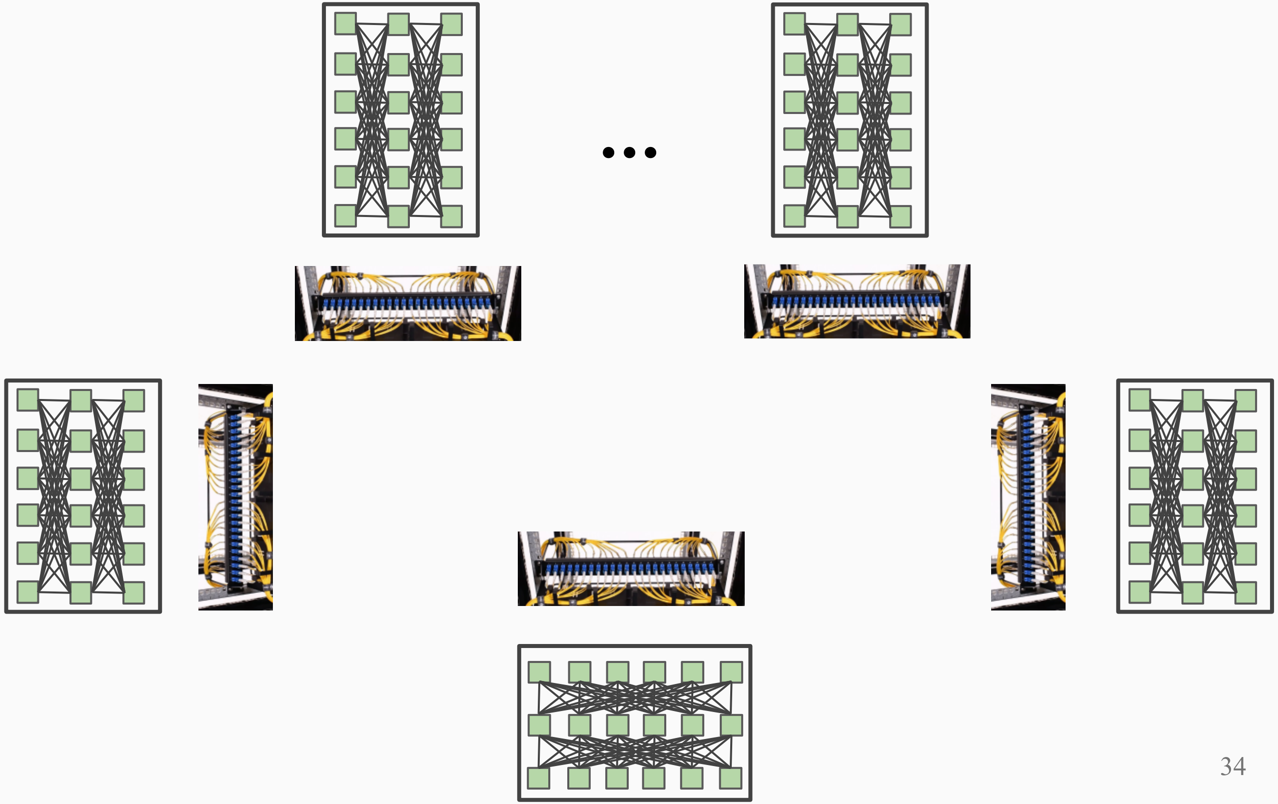
switches



Deployment complexity metrics

switches

patch panels

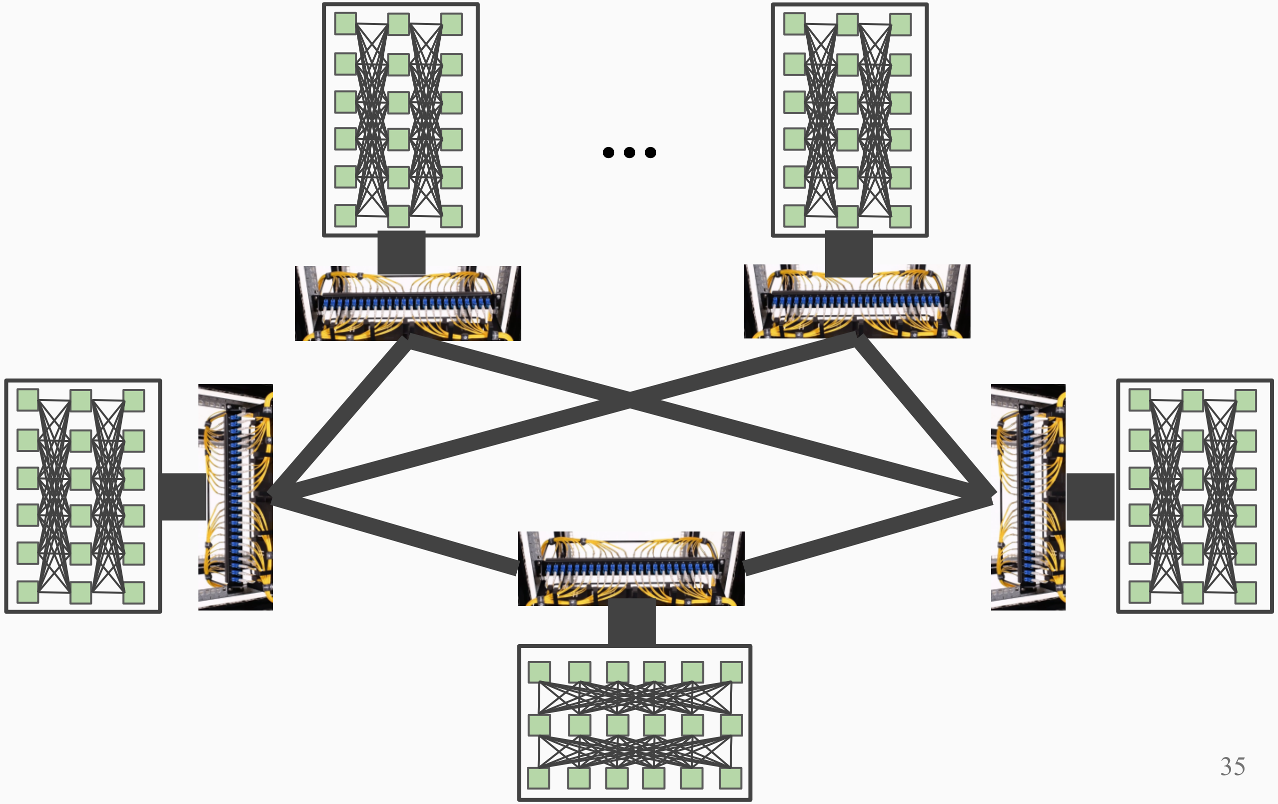


Deployment complexity metrics

switches

patch panels

bundle types



Challenges

How to characterize the management complexity?

How does topology structure affect the management complexity?

Is there a topology family with lower management complexity, lower cost and high capacity?

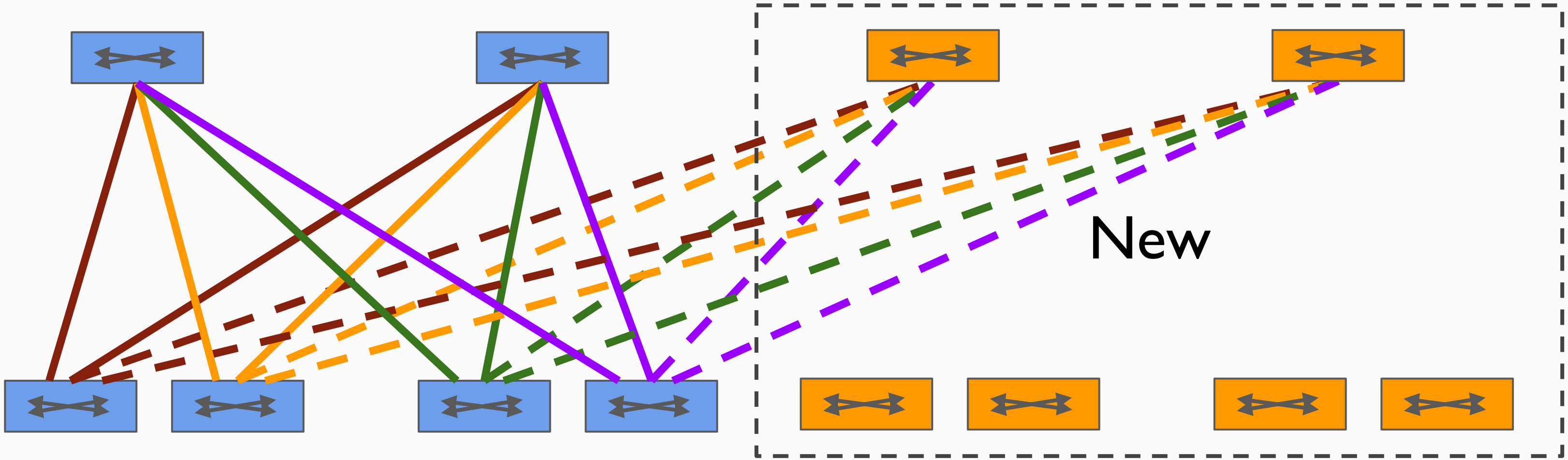
Contributions

Metrics
- Deployment
- **Expansion**

Comparison of topologies
- No topology dominates
- Principles learned

New topology
- **FatClique**

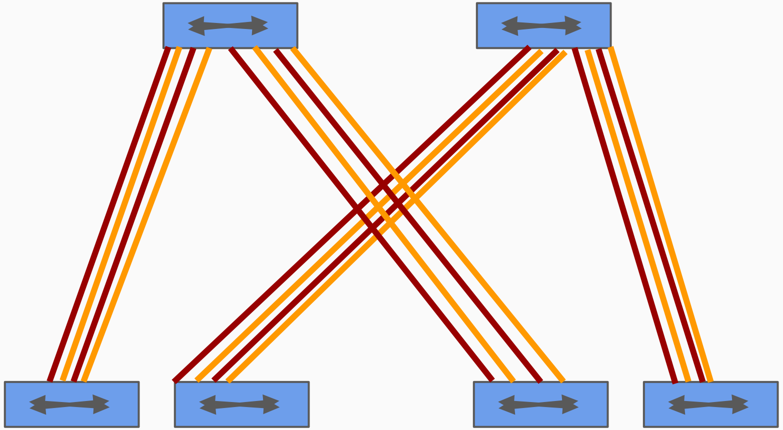
Expansion complexity

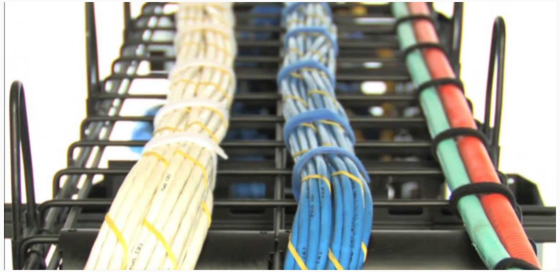


Metric: # Expansion steps

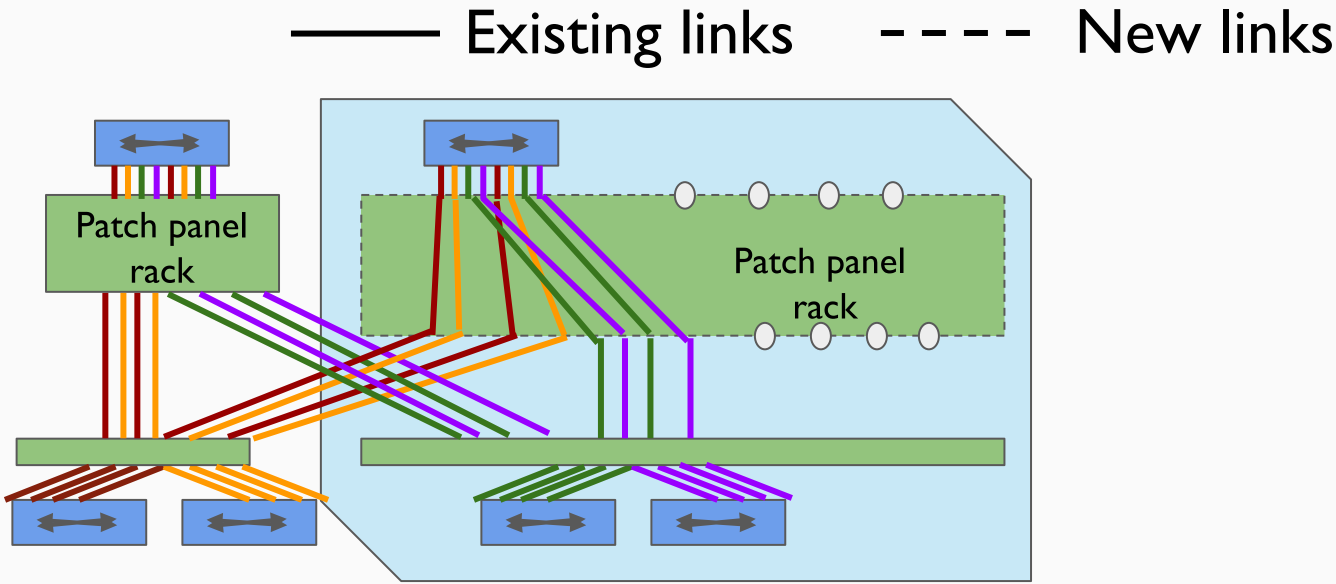
A single expansion step complexity

It is hard to move existing links in cable trays

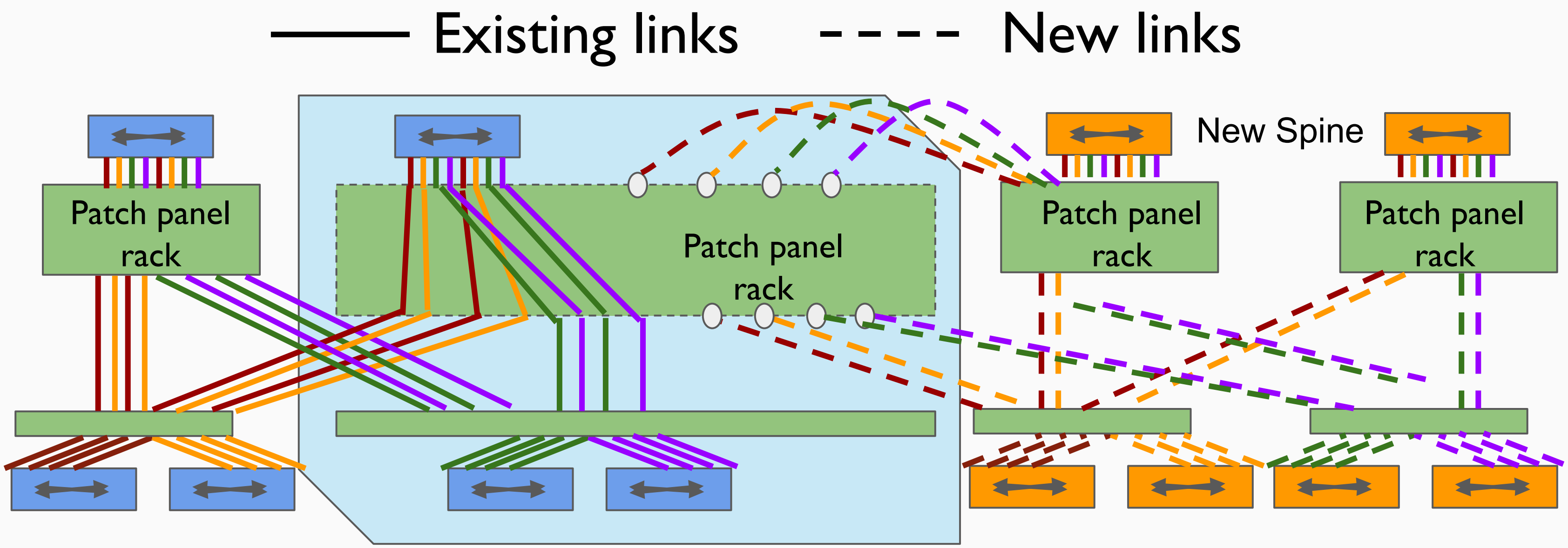




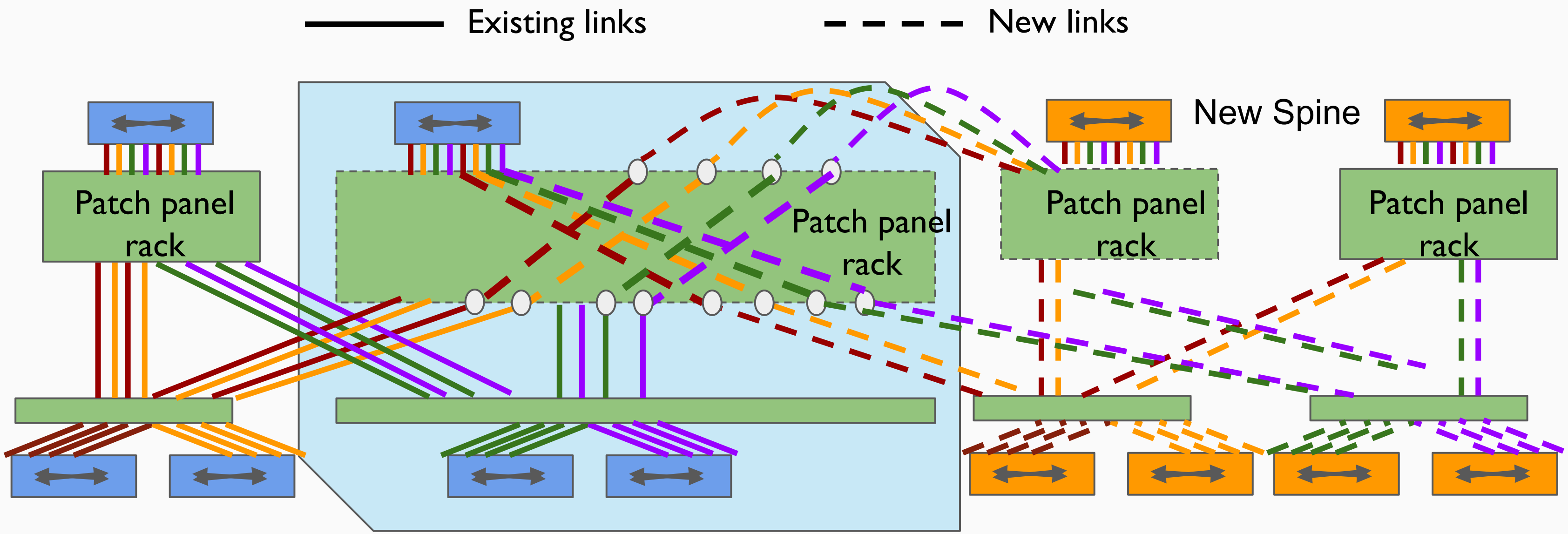
A single expansion step complexity



A single expansion step complexity



A single expansion step complexity



Metric: # Rewired links per patch panel rack

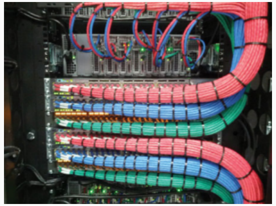
Metrics

Deployment

Switches

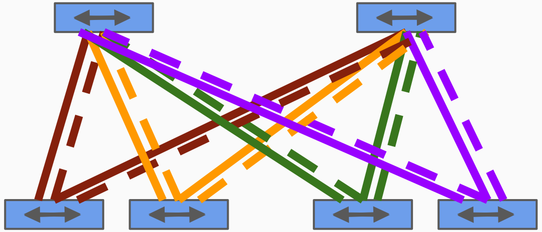
Patch panels

Bundle types

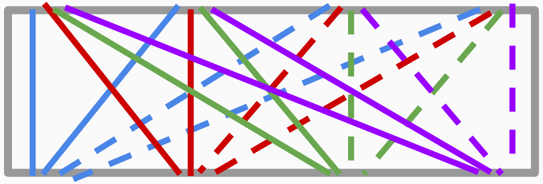


Expansion

Expansion step



Rewired links per patch panel rack



Challenges	Contributions
How to characterize the management complexity?	Metrics – Deployment – Expansion
How does topology structure affect the management complexity?	Comparison of topologies – No topology dominates – Principles learned
Is there a topology family with lower management complexity, lower cost and high capacity?	New topology – **FatClique**

Topology comparison case study

We equalize capacities of topologies

	4-layer Clos (Medium)	Jellyfish
Patch panels		
Bundle types		
Switches		
Re-wired links per patch panel rack		
Expansion steps		

Topology comparison case study

We equalize capacities of topologies

	4-layer Clos (Medium)	Jellyfish
Patch panels	✓	
Bundle types	✓	
Switches		
Re-wired links per patch panel rack		
Expansion steps		

Topology comparison case study

We equalize capacities of topologies

	4-layer Clos (Medium)	Jellyfish
Patch panels	✓	
Bundle types	✓	
Switches		✓
Re-wired links per patch panel rack		✓
Expansion steps		✓

Topology comparison case study

We equalize capacities of topologies

	4-layer Clos (Medium)	Jellyfish
Patch panels	✓	
Bundle types	✓	
Switches		✓
Re-wired links per patch panel rack		✓
Expansion steps		✓

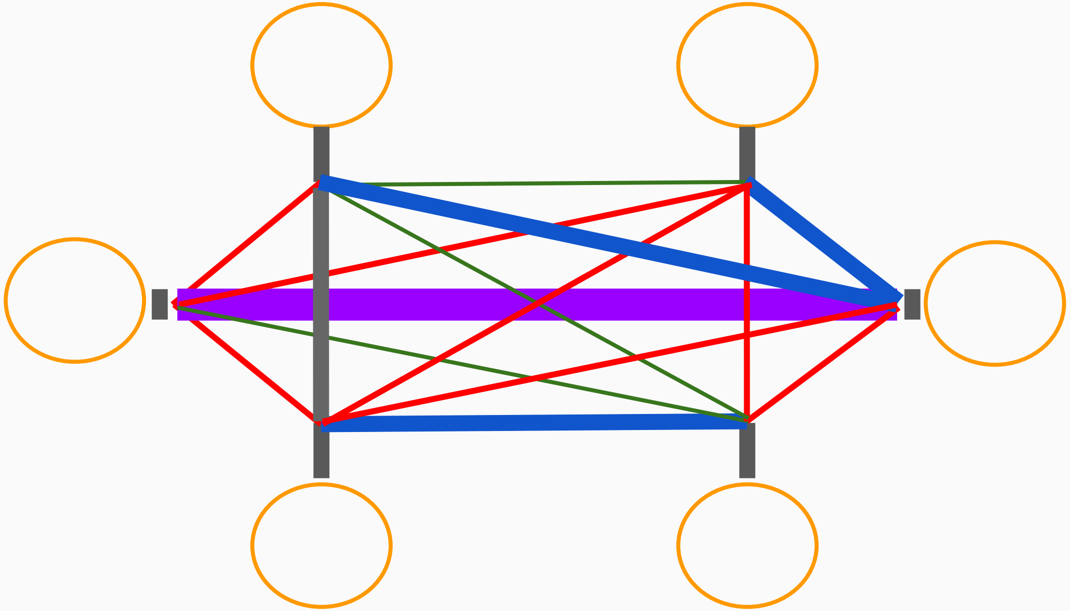
No topology dominates by all metrics!

Principles learned

- Importance of regularity
- Importance of maximizing intra-rack links
- Importance of fat edge

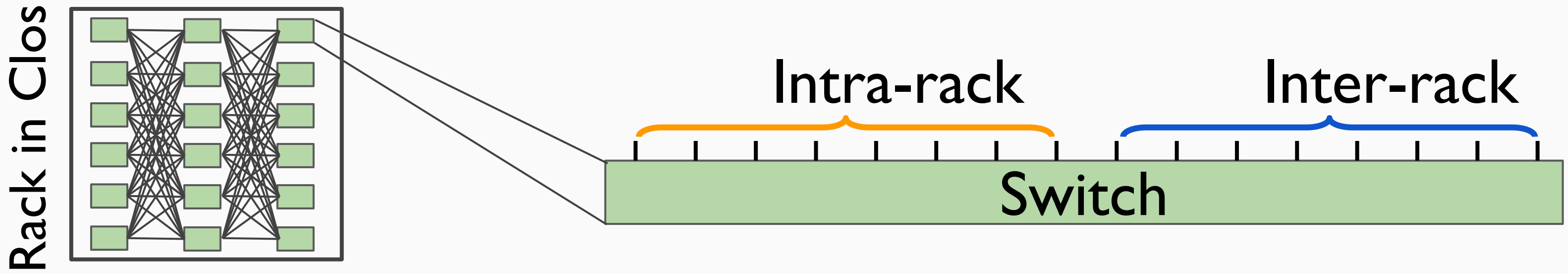
Principle 1: Importance of regularity

Jellyfish is a random graph which leads to non-uniform bundles between switch clusters.

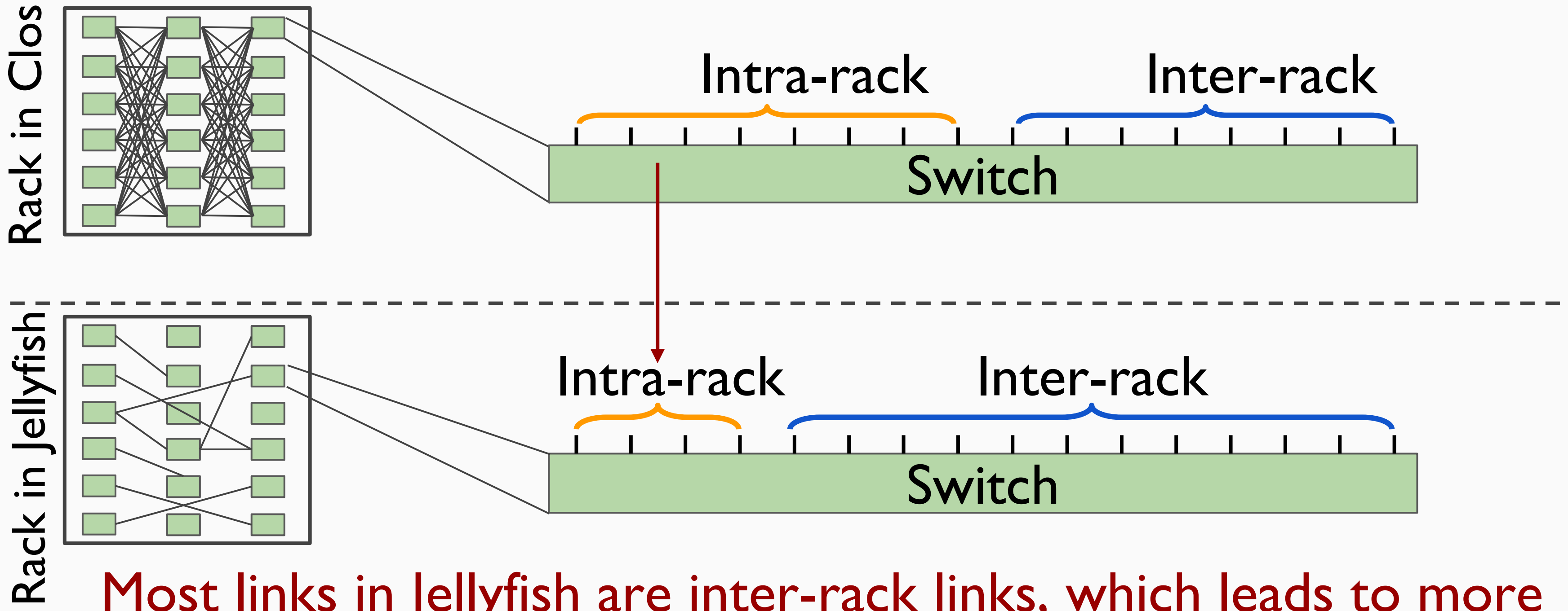


In large scale, Jellyfish has one order of magnitude more bundle types than Clos!

Principle 2: Importance of maximizing intra-rack links

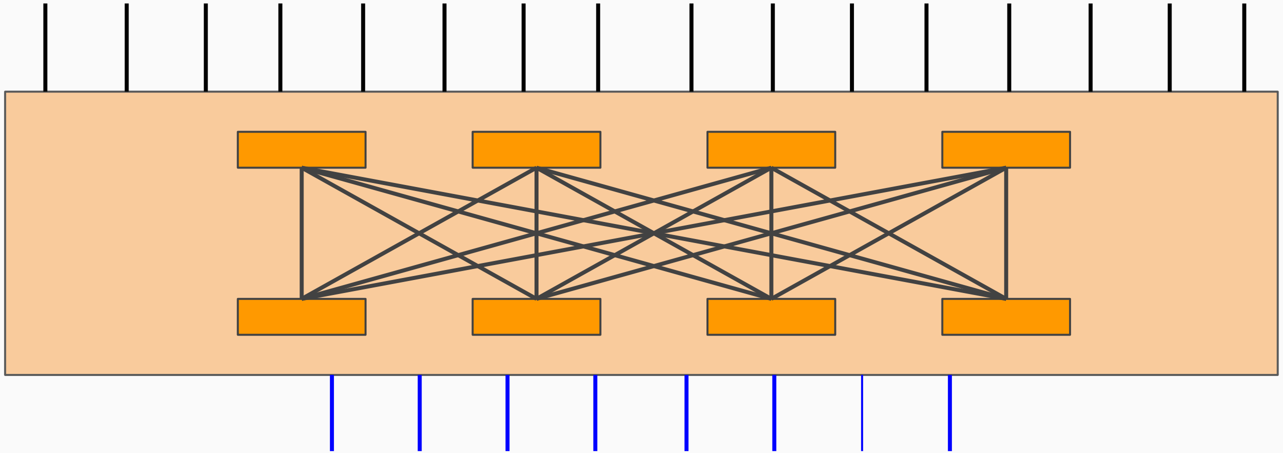


Principle 2: Importance of maximizing intra-rack links



Most links in Jellyfish are inter-rack links, which leads to more patch panel usage and high wiring complexity!

Principle 3: Importance of fat edge

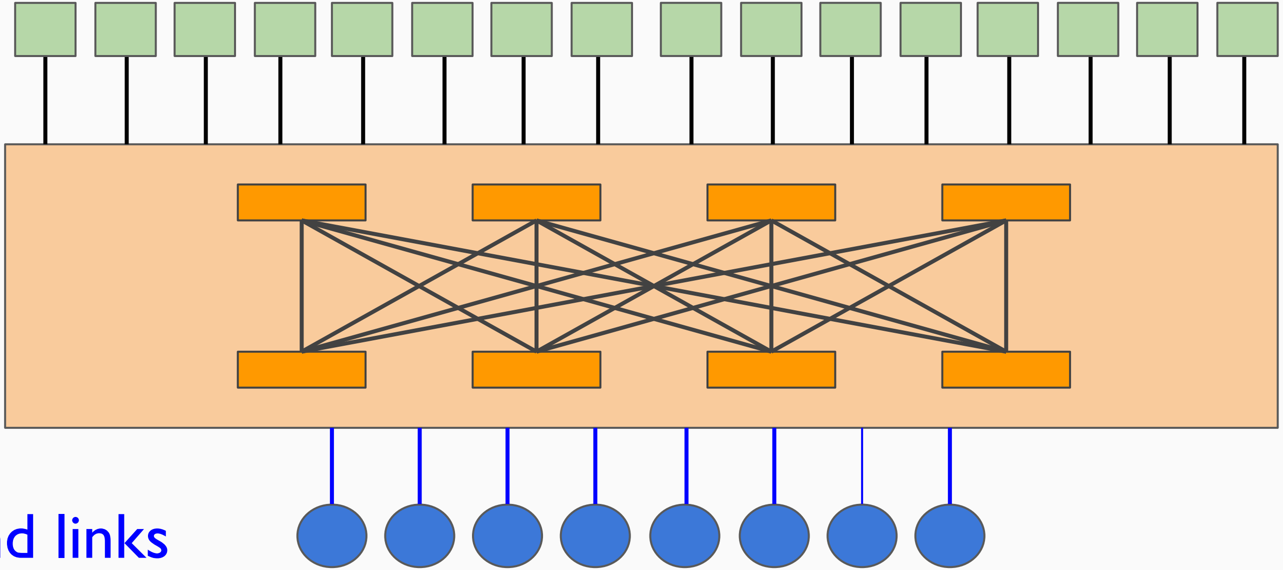


Network edge

Principle 3: Importance of fat edge

Northbound links

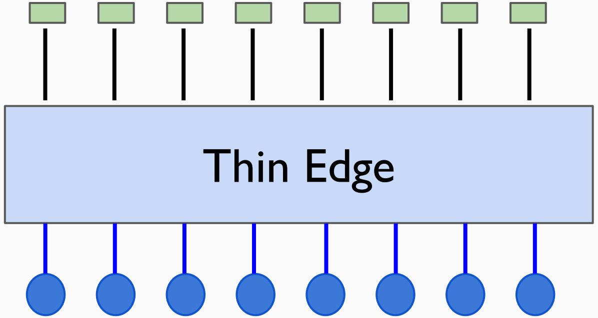
Switches



Southbound links

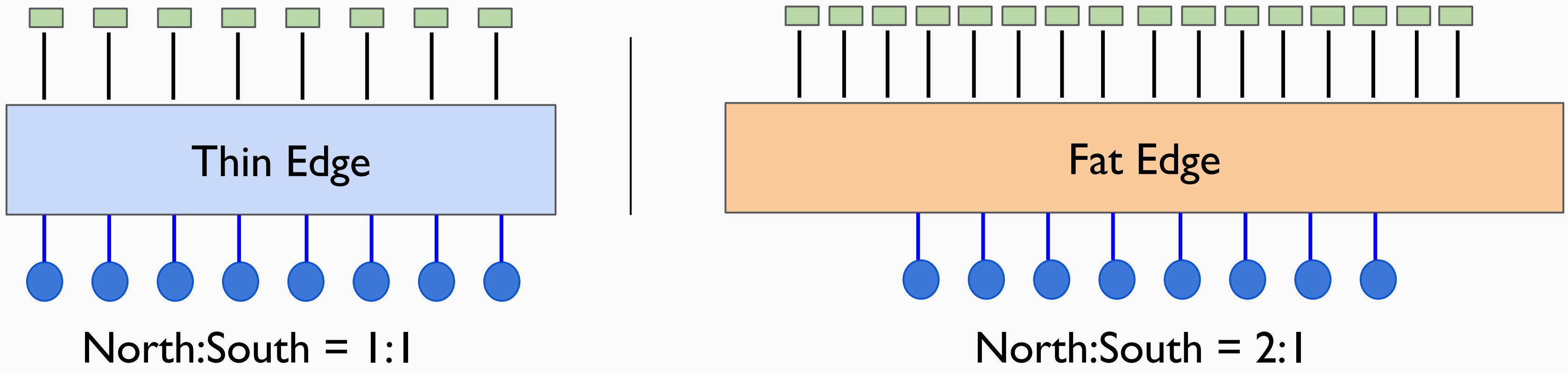
Servers

Principle 3: Importance of fat edge



North:South = 1:1

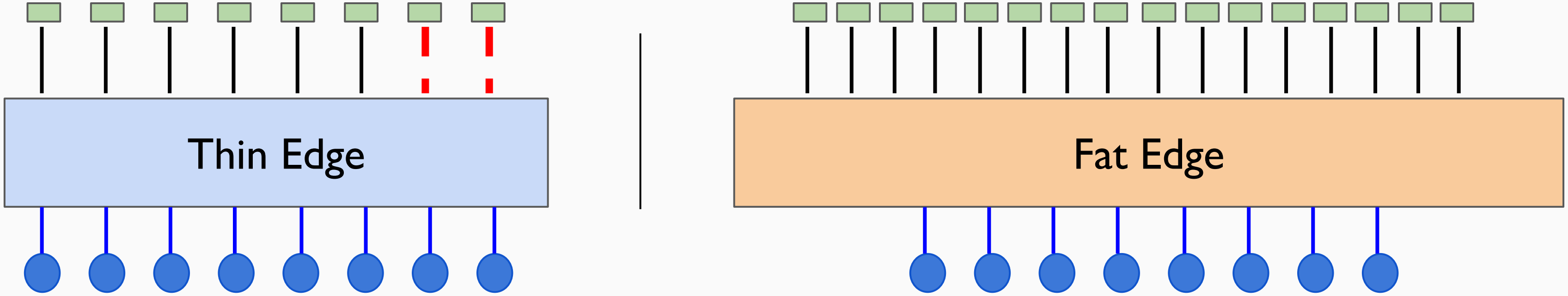
Principle 3: Importance of fat edge



Principle 3: Importance of fat edge

Residual capacity requirement during expansion: 75%
Rewiring leads to capacity drop; Drain traffic before rewiring

Draining 25% links --> 25% lose



North:South = 1:1

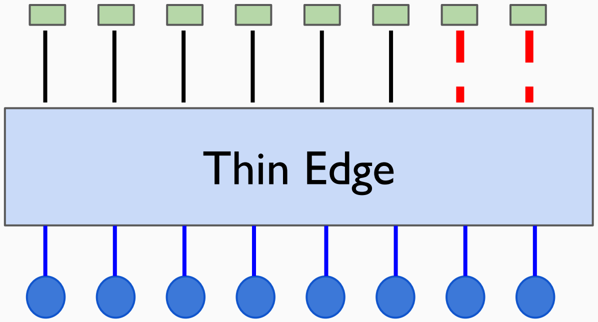
North:South = 2:1

Principle 3: Importance of fat edge

Residual capacity requirement during expansion: 75%

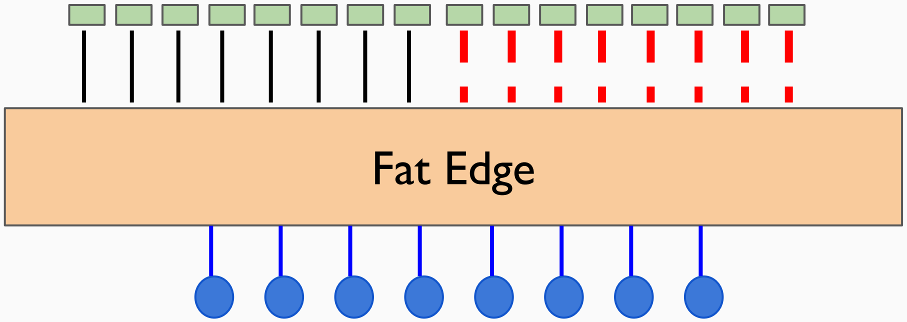
Rewiring leads to capacity drop; Drain traffic before rewiring

Draining 25% links --> 25% lose



North:South = 1:1

Draining 50% links --> 0% lose



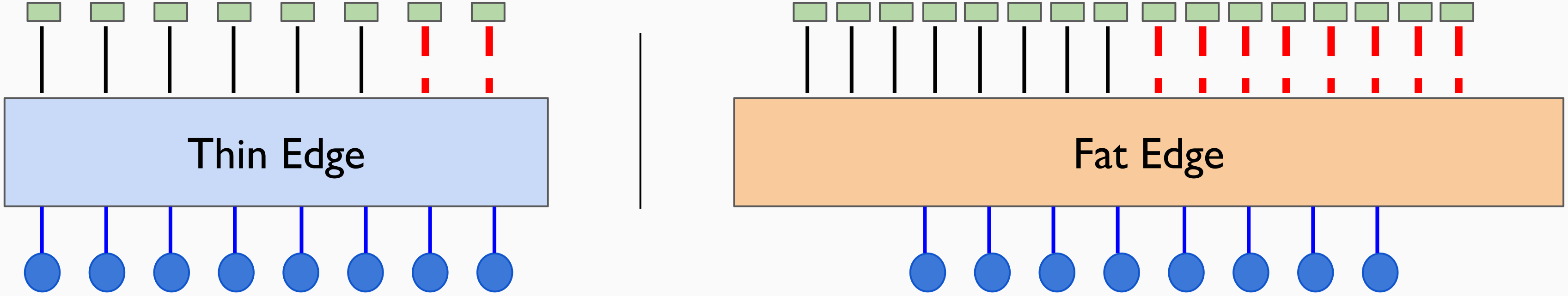
North:South = 2:1

Principle 3: Importance of fat edge

Residual capacity requirement during expansion: 75%

Draining 25% links --> 25% lose

Draining 50% links --> 0% lose



North:South = 1:1

North:South = 2:1

- At fat edge, more links can be rewired in a single expansion step.
- Jellyfish has fat edge = fewer expansion steps
- Clos has thin edge = more expansion steps

Summary of case study

	4-layer Clos (Medium)	Jellyfish
Regularity	✓	
Maximizing intra-rack links	✓	
Fat edge		✓

Challenges

How to characterize the management complexity?

How does topology structure affect the management complexity?

Is there a topology family with lower management complexity, lower cost and high capacity?

Contributions

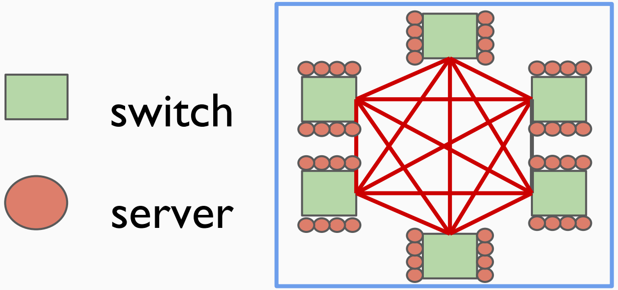
Metrics
- Deployment
- Expansion

Comparison of topologies
- No topology dominates
- Principles learned

New topology
- **FatClique**

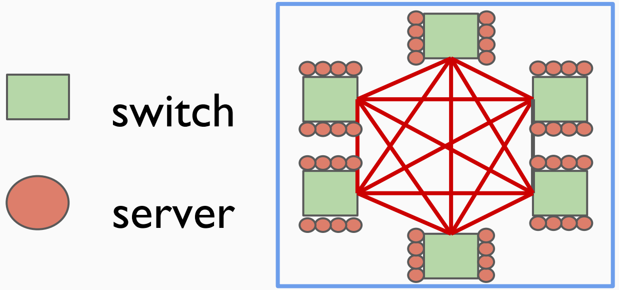
FatClique

Sub-block (Clique of Switches)



FatClique

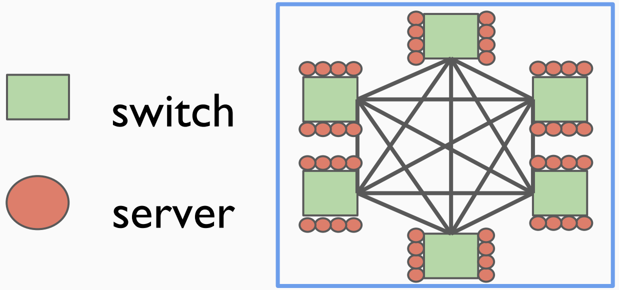
Sub-block (Clique of Switches)



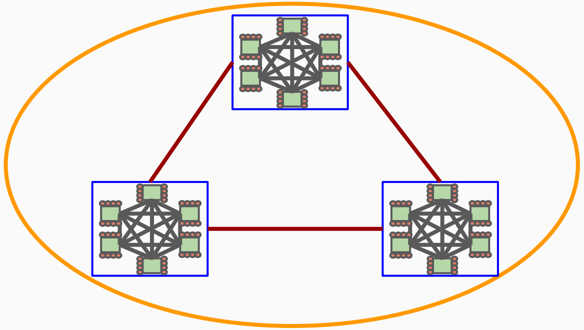
Goal: one or multiple sub-blocks should be packed into a single rack to maximize intra-rack links

FatClique

Sub-block (Clique of Switches)

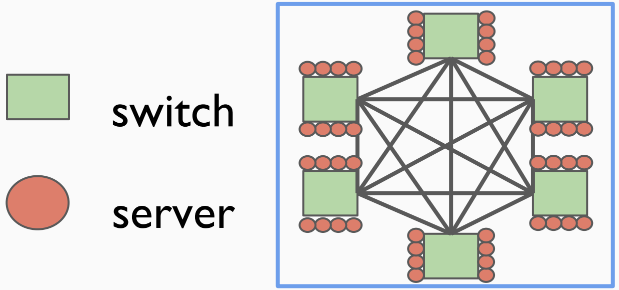


Block (Clique of Sub-blocks)

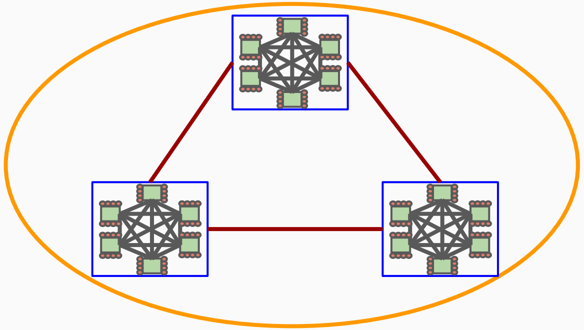


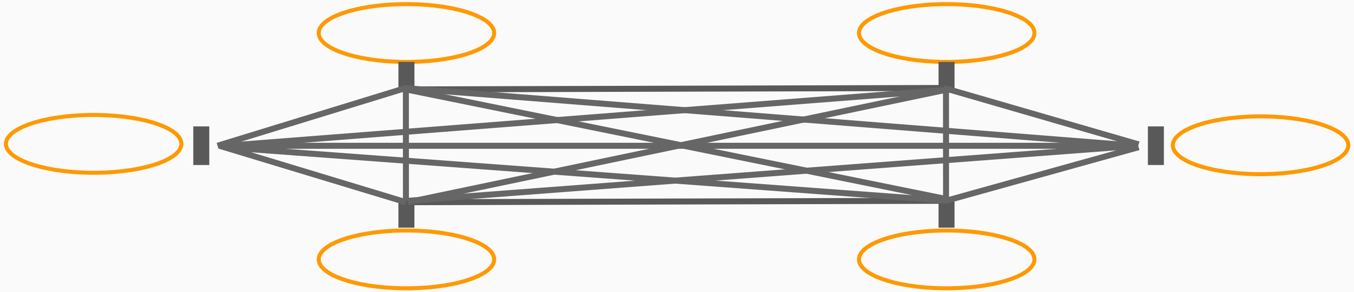
FatClique

Sub-block (Clique of Switches)



Block (Clique of Sub-blocks)





The Whole Network (Clique of Blocks)

Goal: blocks should be large enough to form uniform bundles.

Does FatClique satisfy principles learned?

- Regularity ✓
- Maximizing intra-rack links
- Fat edge

Does FatClique satisfy principles learned?

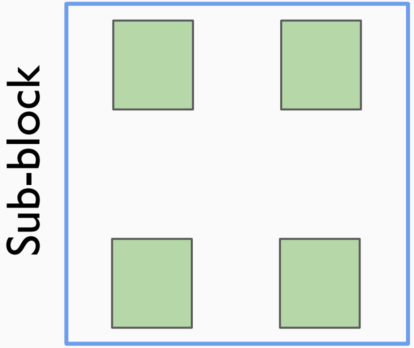
– Regularity ✓

– Maximizing intra-rack links

– Fat edge

Challenges

Conflicts: Fat edge *vs* maximizing intra-rack links

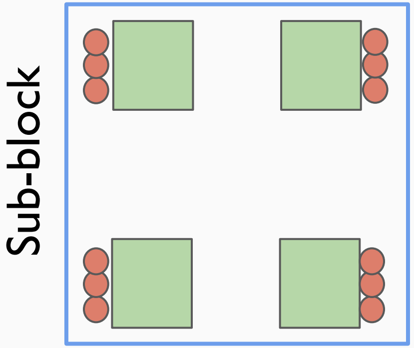


For each switch,

- 3 servers
- 3 intra-rack links
- 3 inter-rack links

Challenges

Conflicts: Fat edge *vs* maximizing intra-rack links



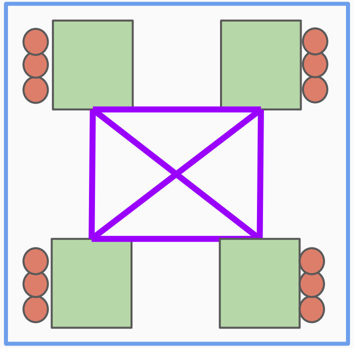
For each switch,

- 3 servers
- 3 intra-rack links
- 3 inter-rack links

Challenges

Conflicts: Fat edge *vs* maximizing intra-rack links

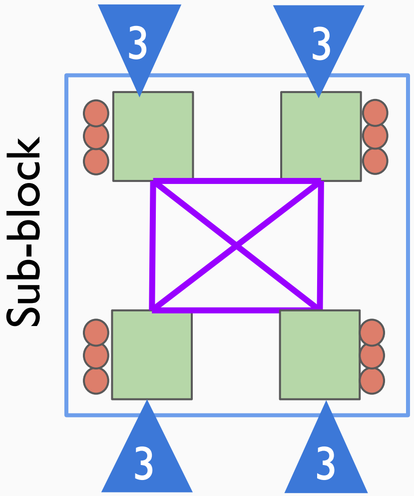
Sub-block



For each switch,
- 3 servers
- 3 intra-rack sw
- 3 inter-rack sw

Challenges

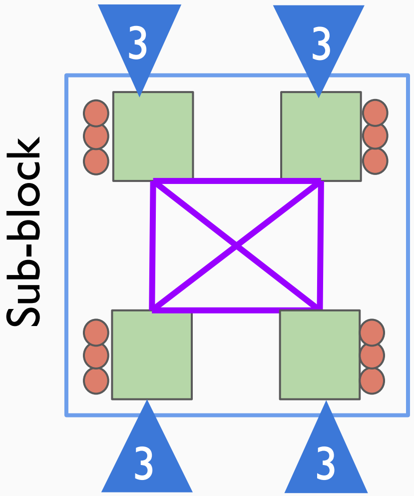
Conflicts: Fat edge *vs* maximizing intra-rack links



For each switch,
- 3 servers
- 3 intra-rack sw
- 3 inter-rack sw

Challenges

Conflicts: Fat edge *vs* maximizing intra-rack links



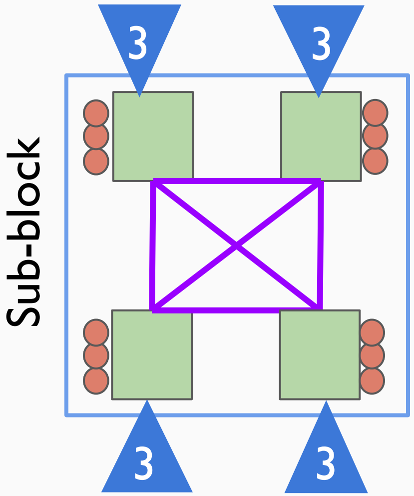
For each switch,

- 3 servers
- 3 intra-rack sw
- 3 inter-rack sw

Thin edge

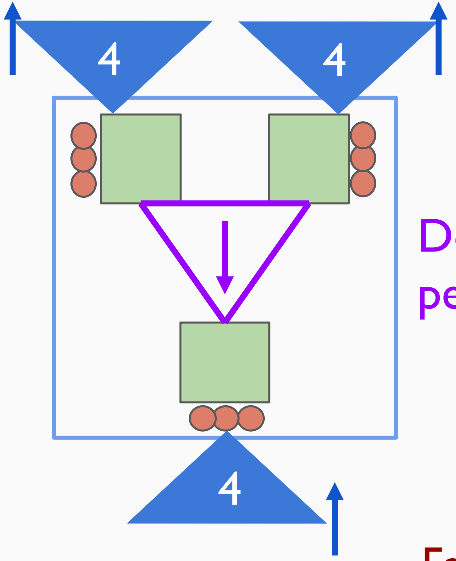
Challenges

Conflicts: Fat edge *vs* maximizing intra-rack links



For each switch,
- 3 servers
- 3 intra-rack sw
- 3 inter-rack sw

Thin edge



Decrease intra-rack links per switch from **3** to **2**

Fat edge

Challenges

Conflicts

- Fat edge *vs* maximizing intra-rack links
- Fat edge vs minimizing switches

Challenges

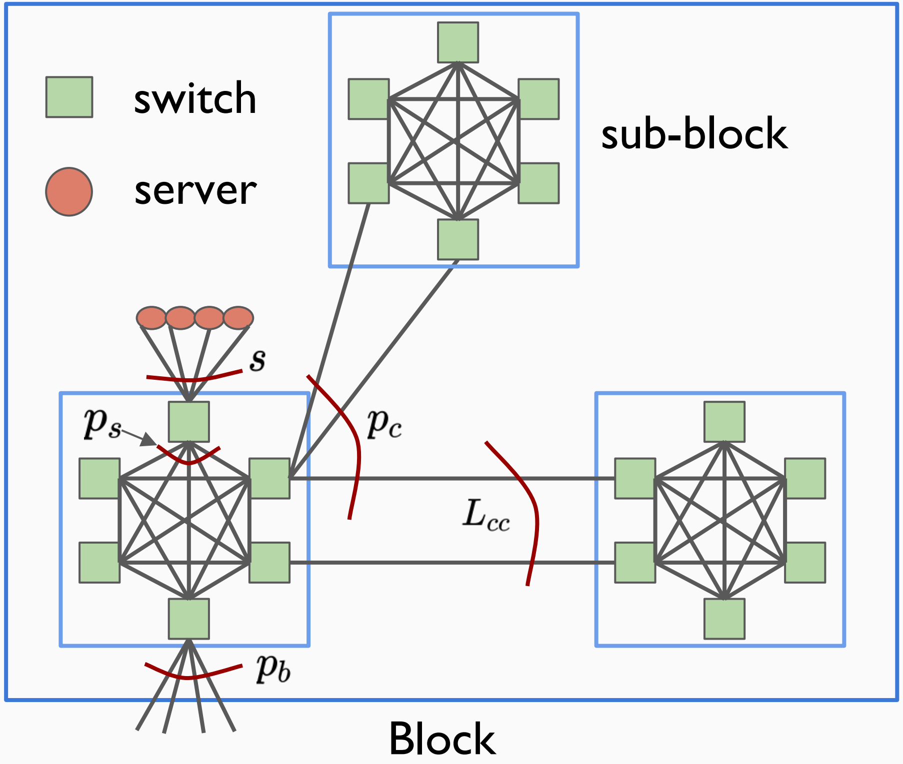
Conflicts

- Fat edge *vs* maximizing intra-rack links
- Fat edge vs minimizing switches

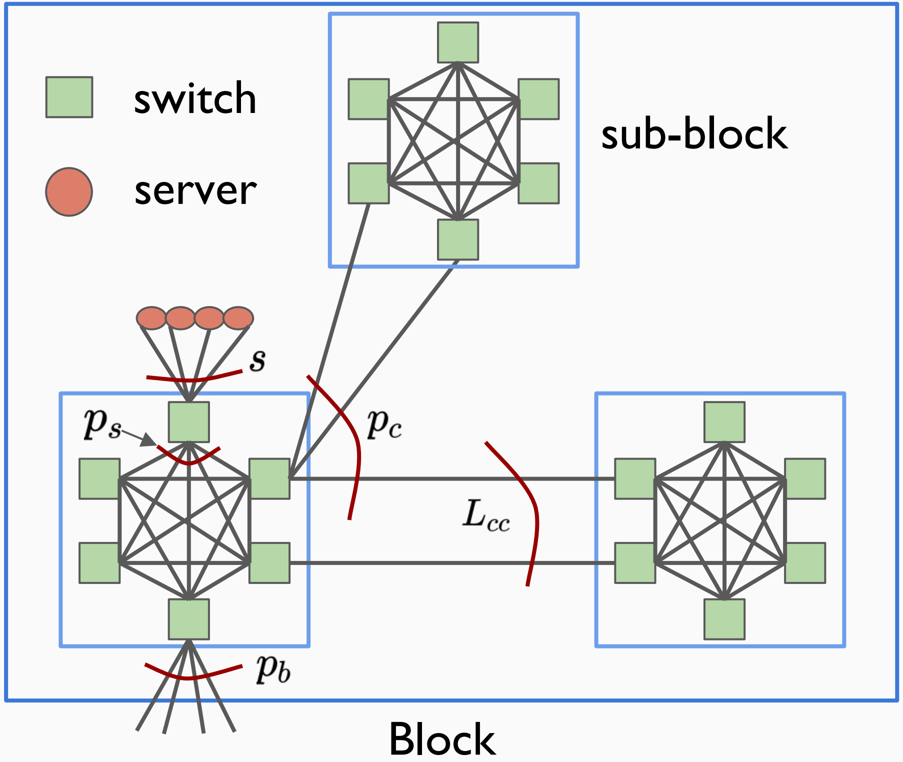
Constraints

- provide right amount of capacity
- minimize rack fragmentation
- minimize overall cable length
- …

Constraint-based search



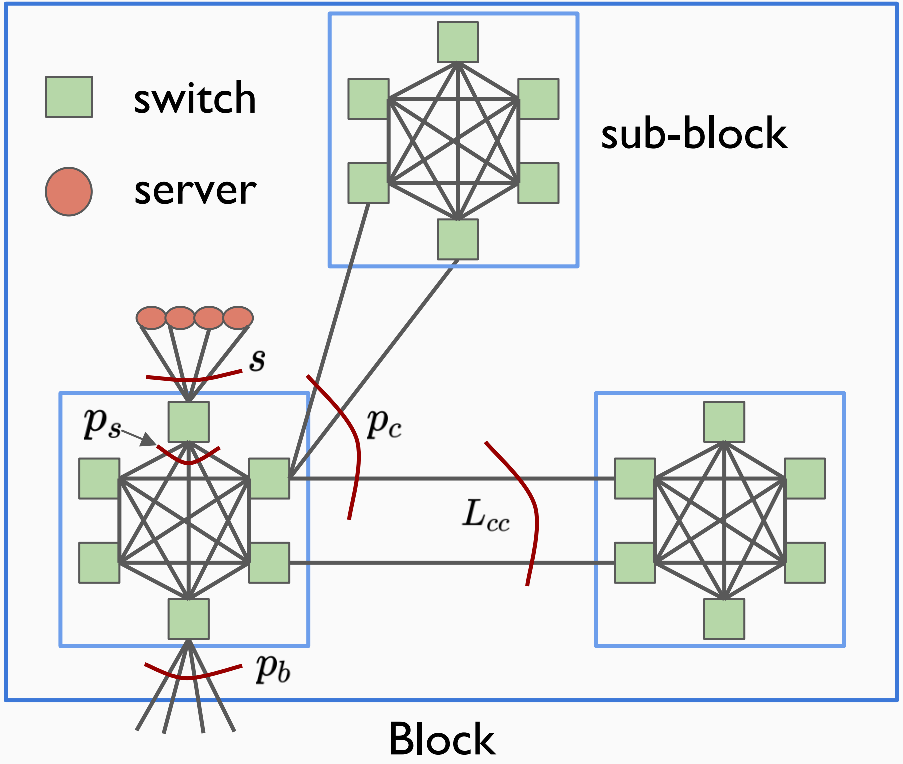
Constraint-based search



Constraints

- Fat edge at a switch
  - # Northbound > # Southbound

Constraint-based search

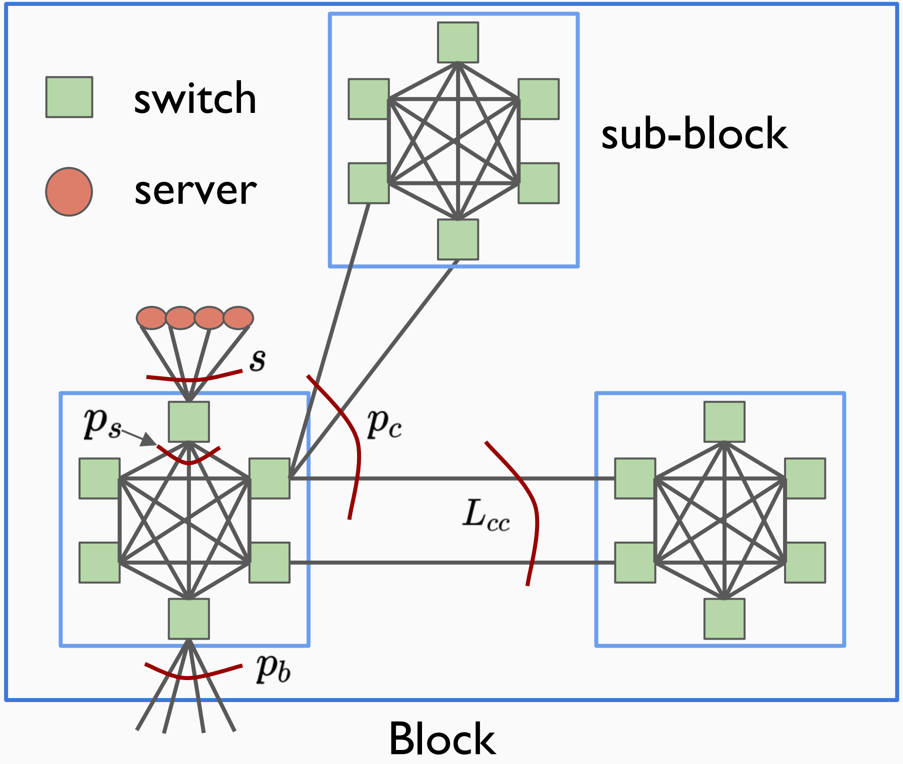


Constraints

- Fat edge at a switch
  - # Northbound > # Southbound

    $p_s + p_b + p_c > s$

Constraint-based search



Constraints

- Fat edge at a switch
  - # Northbound > # Southbound

    $p_s + p_b + p_c > s$

- Fat edge at a block
- Block size
- …

Evaluation

– Does FatClique have lower deployment complexity?

– Does FatClique have lower expansion complexity?

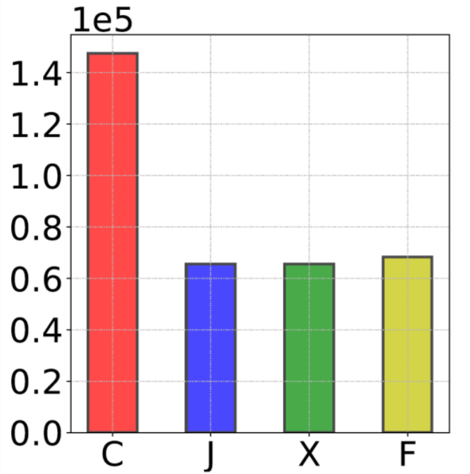
Evaluation Methodology

– Equalize capacities for topologies
– Compare topologies at different scale
– Highly optimized placement algorithms for different topologies
– Optimal expansion algorithm for symmetric Clos
– Search-based near-optimal expansion algorithm for FatClique
– Patch panel usage in different topologies
– ...

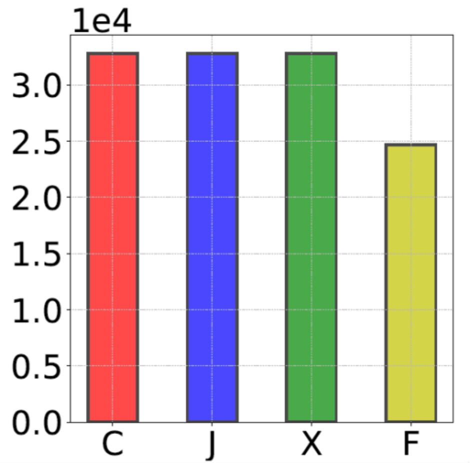
FatClique has low deployment complexity

C: Clos, J: Jellyfish, X: Xpander, F: FatClique

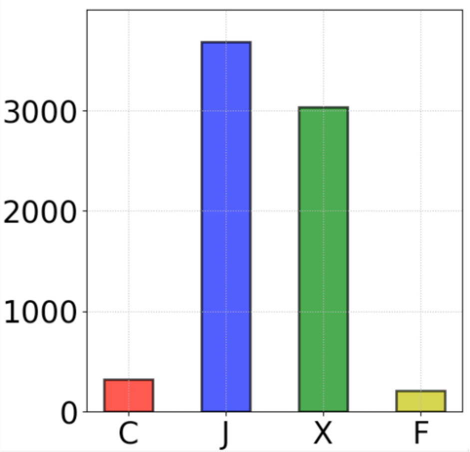
switches



patch panels



bundle types

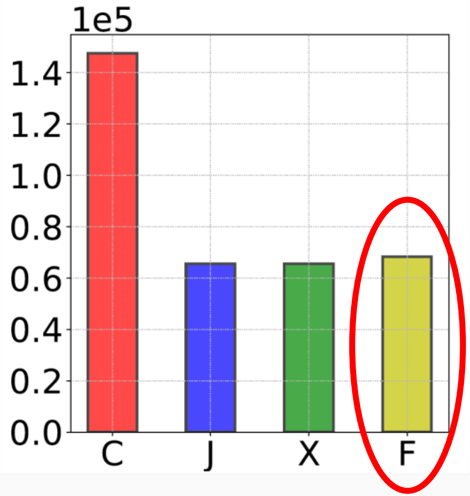


FatClique performs best by all deployment metrics

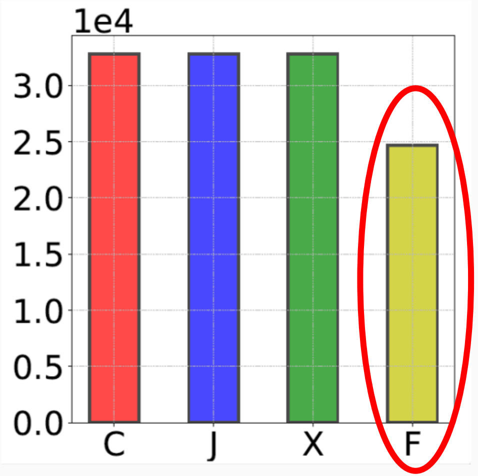
FatClique has low deployment complexity

C: Clos, J: Jellyfish, X: Xpander, F: FatClique

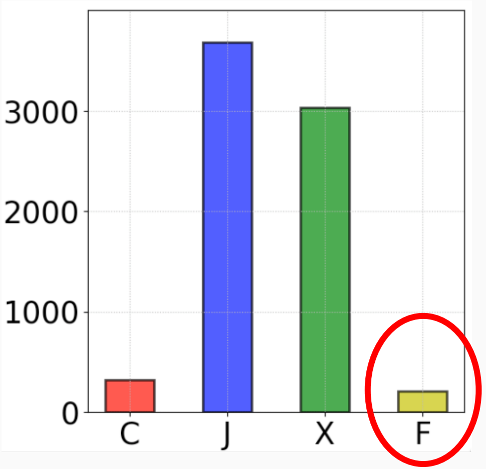
switches



patch panels

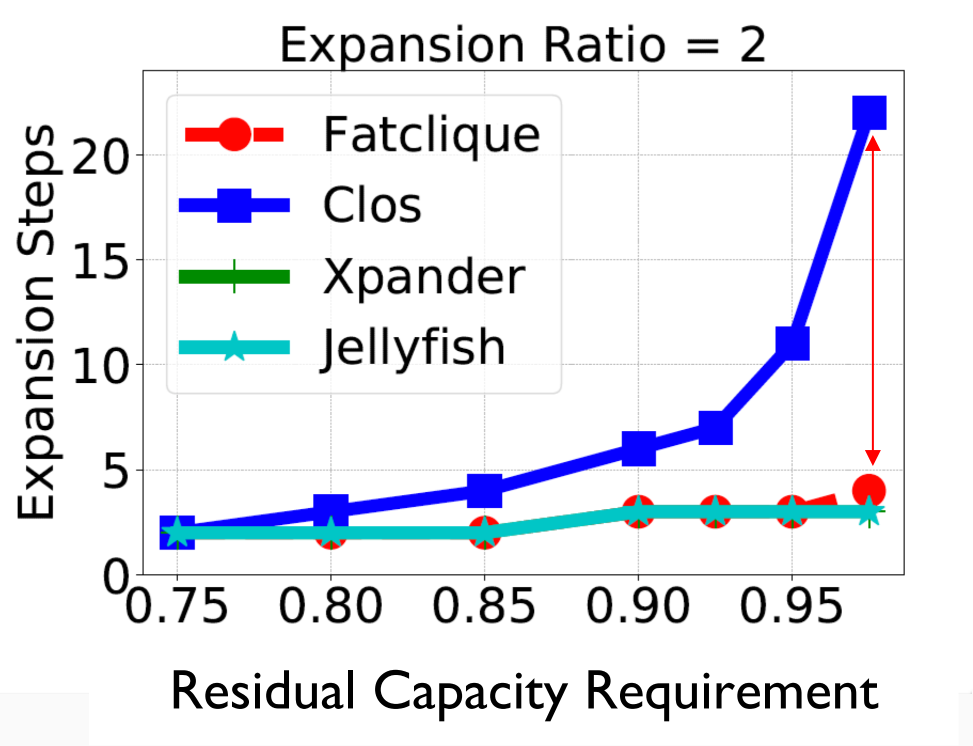


bundle types



FatClique performs best by all deployment metrics

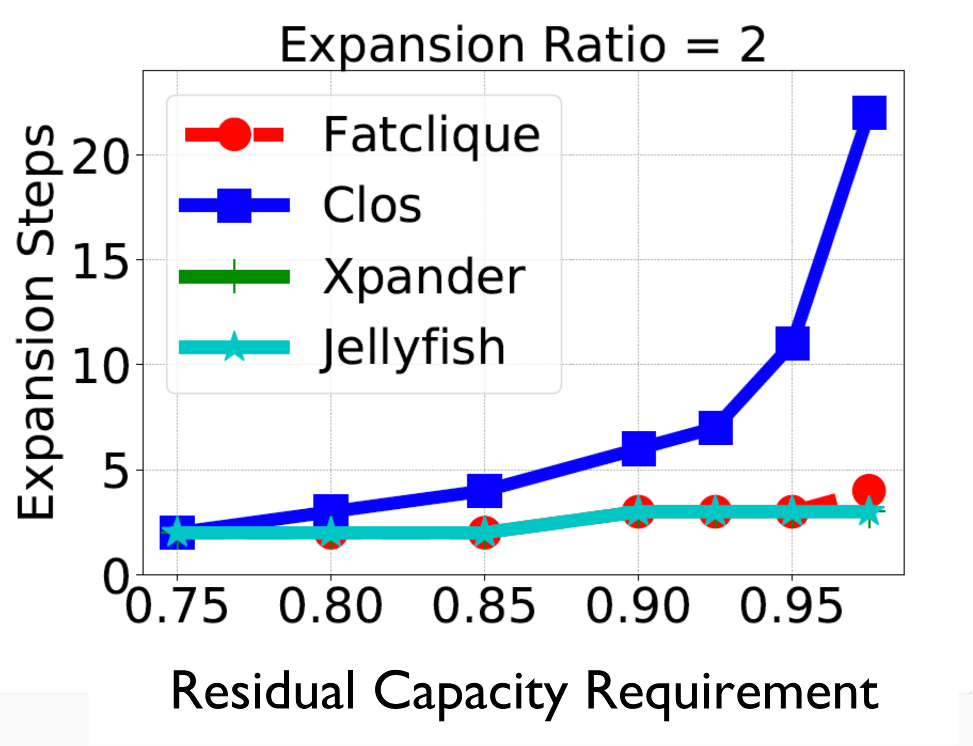
FatClique has low expansion complexity



FatClique is as good as expanders

- 3 or 4 expansion steps even when the residual capacity requirement is tight

FatClique has low expansion complexity



FatClique is as good as expanders

- 3 or 4 expansion steps even when the residual capacity requirement is tight
- FatClique enables higher availability

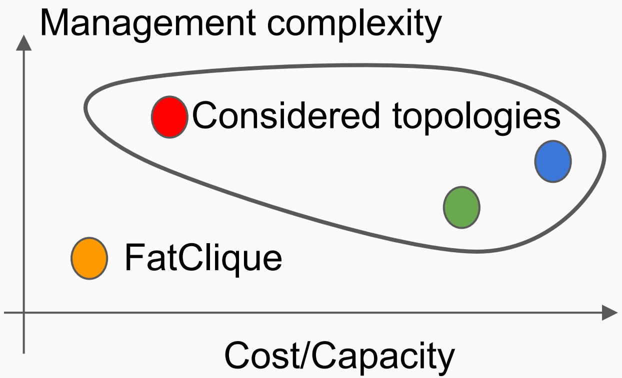
Conclusions and Future work

Management complexity is an important dimension for topology design

- Our work is a first step towards this direction
- Metric design

FatClique achieves lower management complexity

- with same capacity
- with lower cost



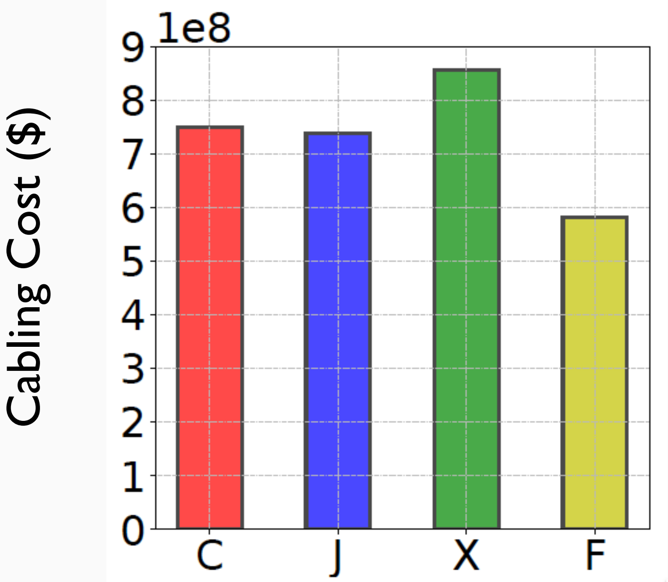
Future work

- control plane complexity
- network debuggability
- practical routing for FatClique

Thanks!

Backup

FatClique has low cabling cost



FatClique is 23% cheaper than Clos

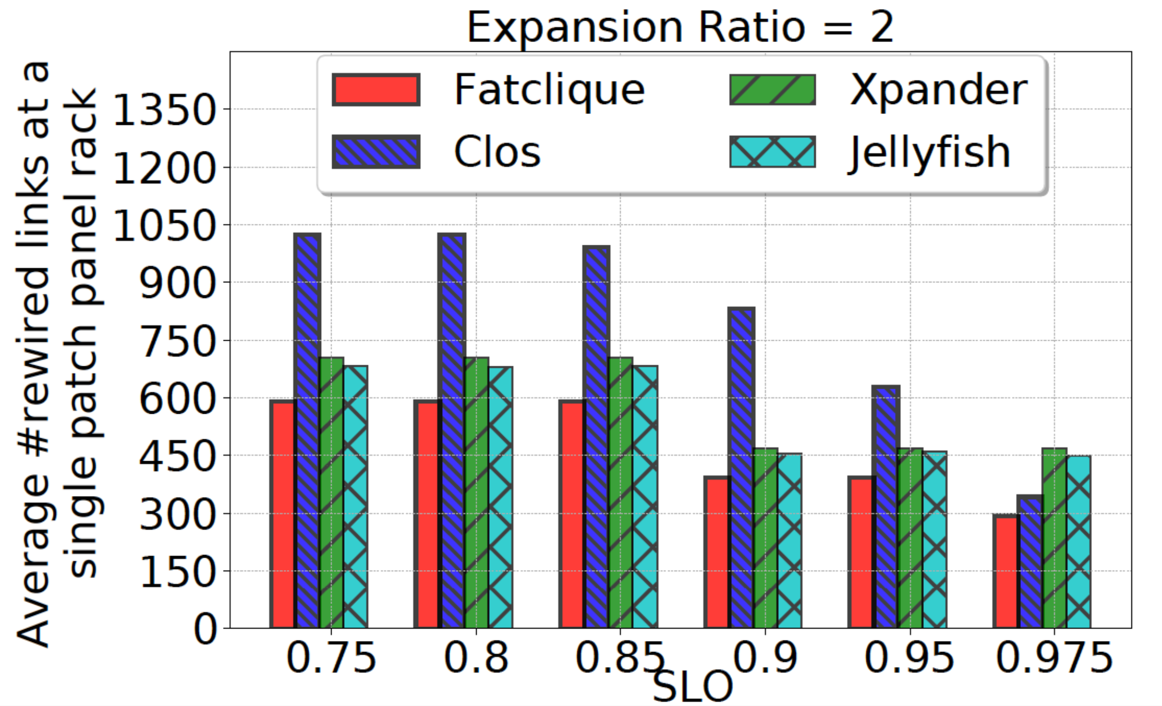
- Smaller number of links

FatClique is cheaper than Expanders

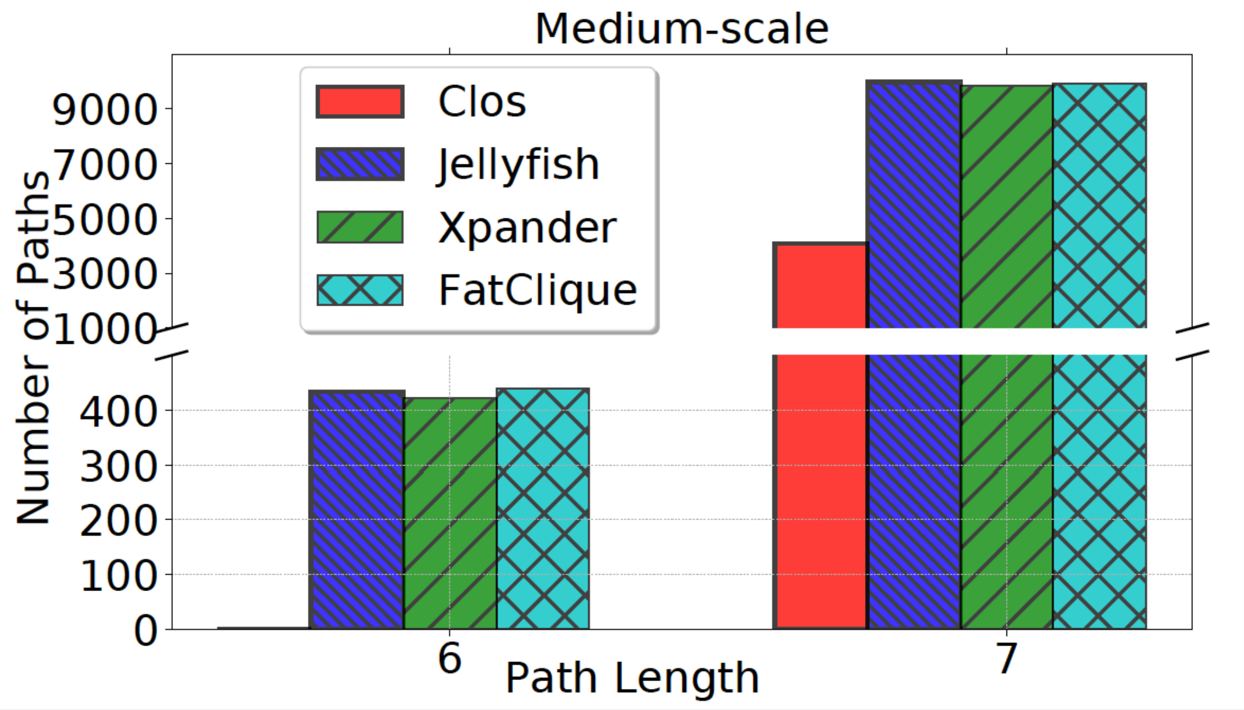
- Maximizing intra-rack links, which saves expensive optical transceivers.

C: Clos, J: Jellyfish, X: Xpander, F: FatClique

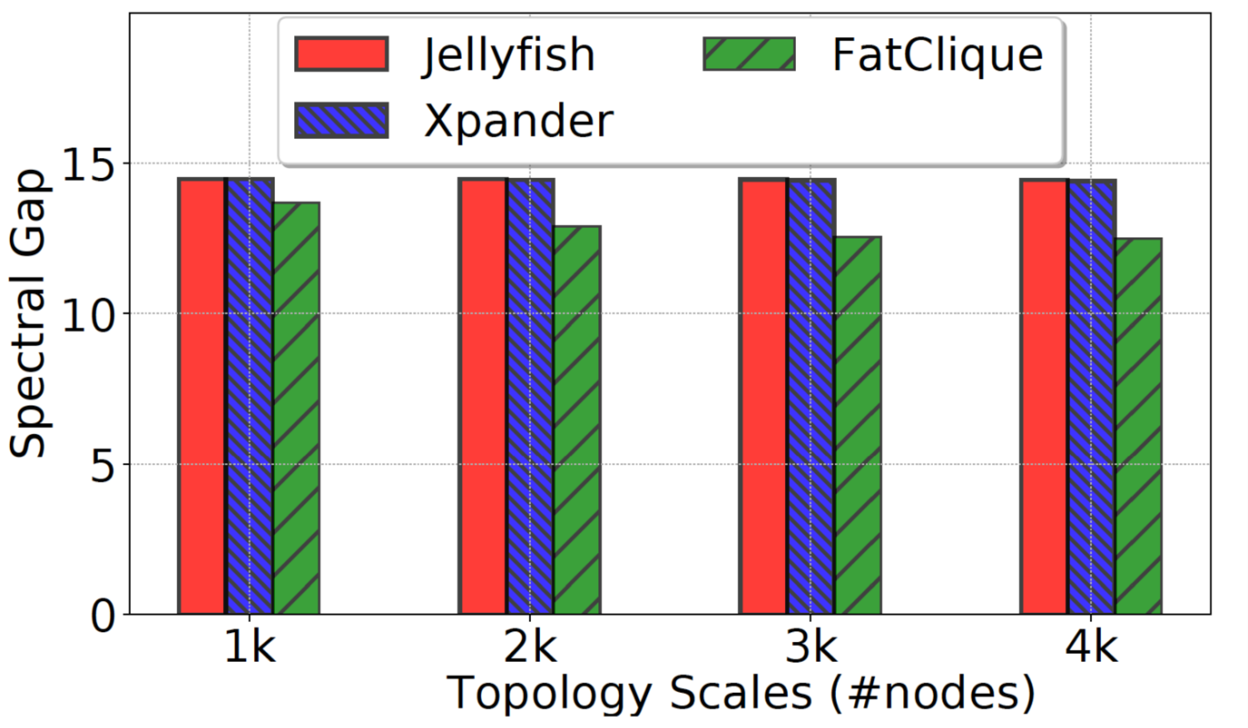
Single step complexity



Path diversity



Spectral gap

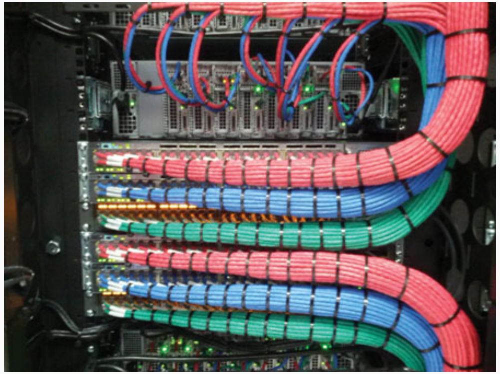


Deployment complexity metrics

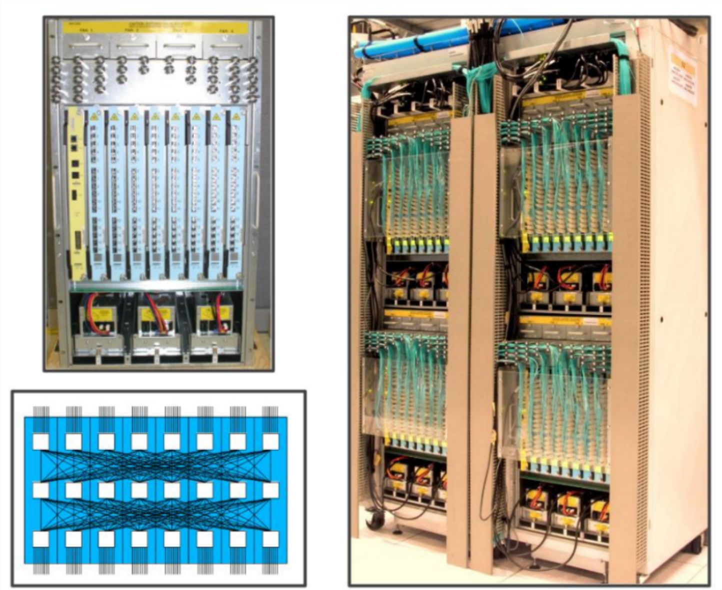
switches

patch panels

bundle types



Deployment-wiring



Google's Watchtower Chassis