Jellyfish

networking data centers randomly

Brighten Godfrey • UIUC

Cisco Systems, September 12, 2013

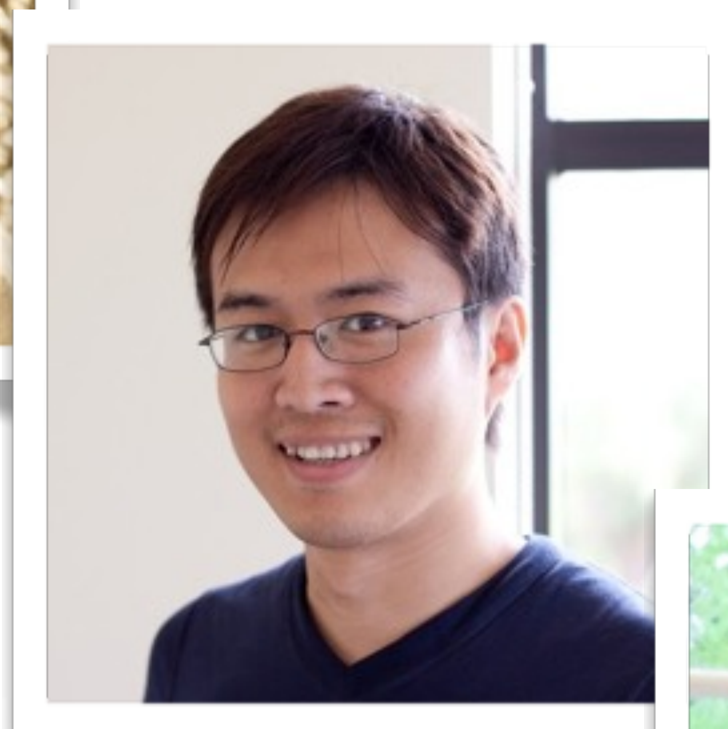
[Photo: Kevin Raskoff]

Ask me about...

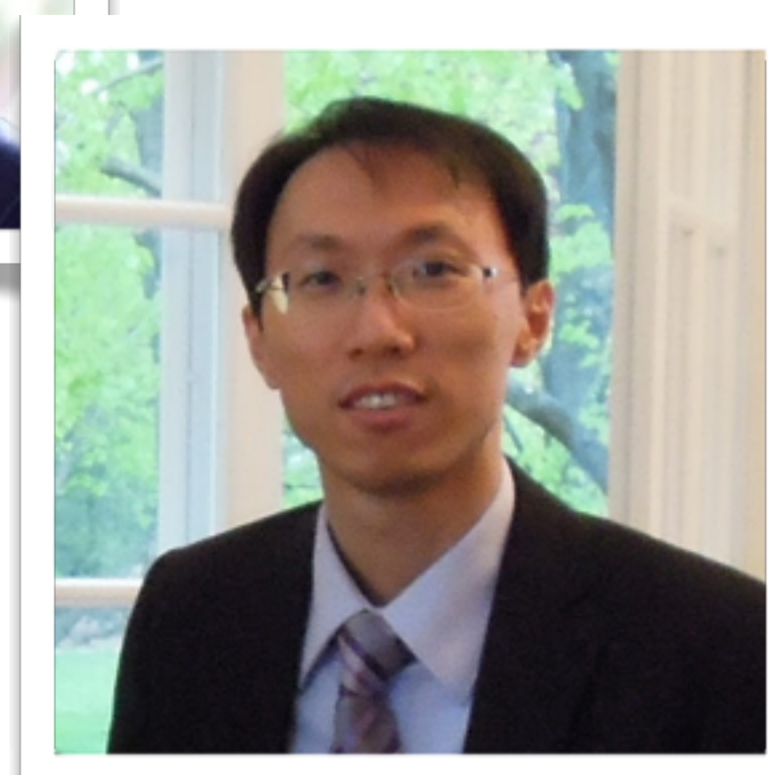
Low latency networked systems

Data plane verification (Veriflow)

Ankit Singla
UIUC



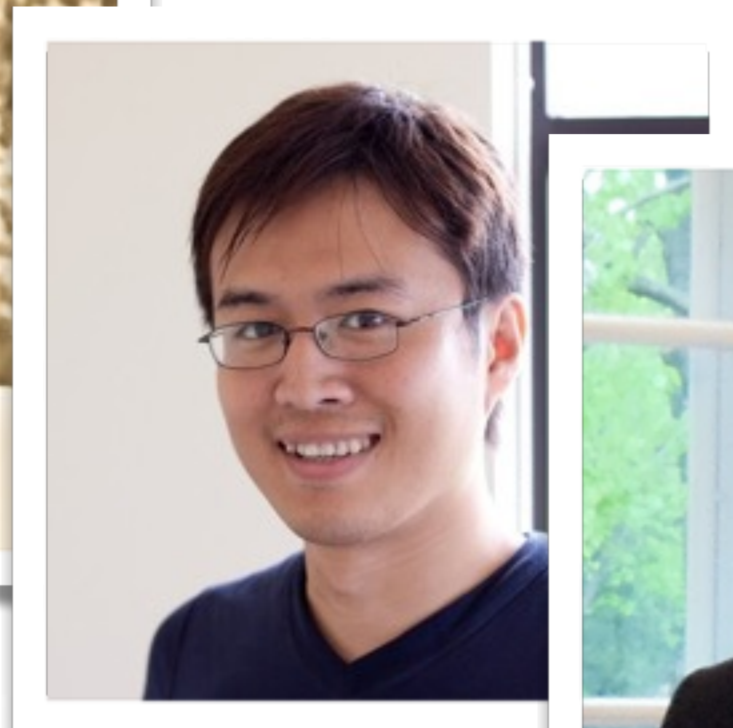
Chi-Yao
Hong
UIUC



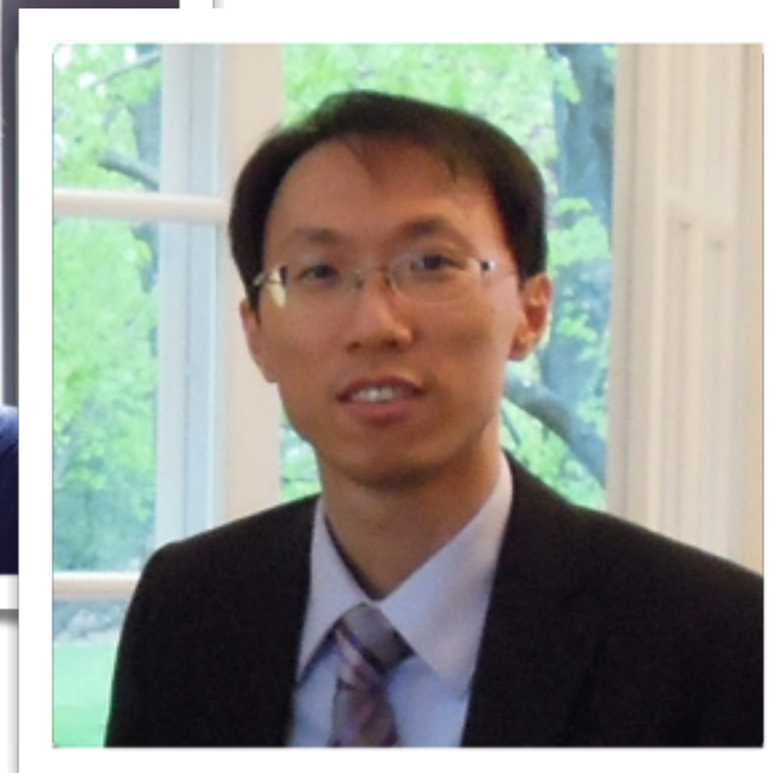
Kyle Jao
UIUC

Sangeetha Abdu Jyothi
UIUC

Ankit Singla

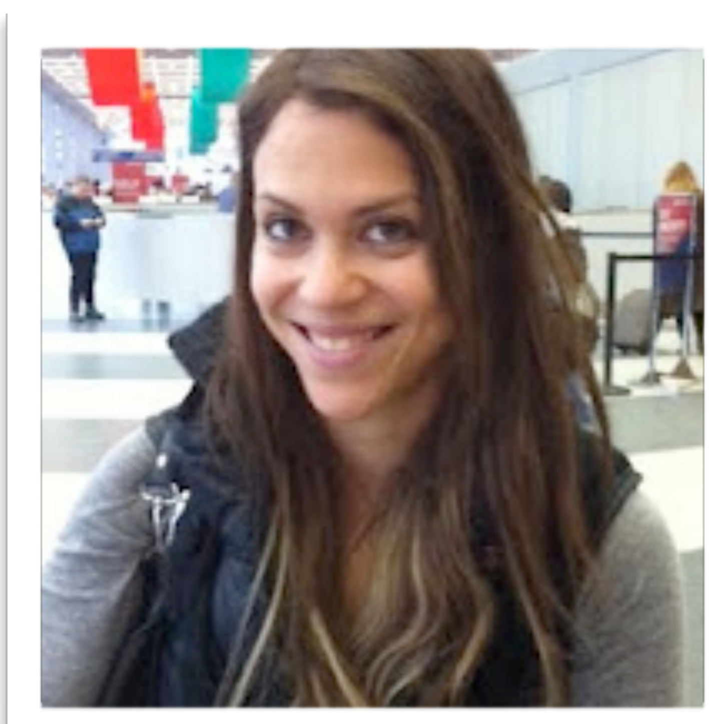


Chi-Yao



Kyle Jao
UIUC

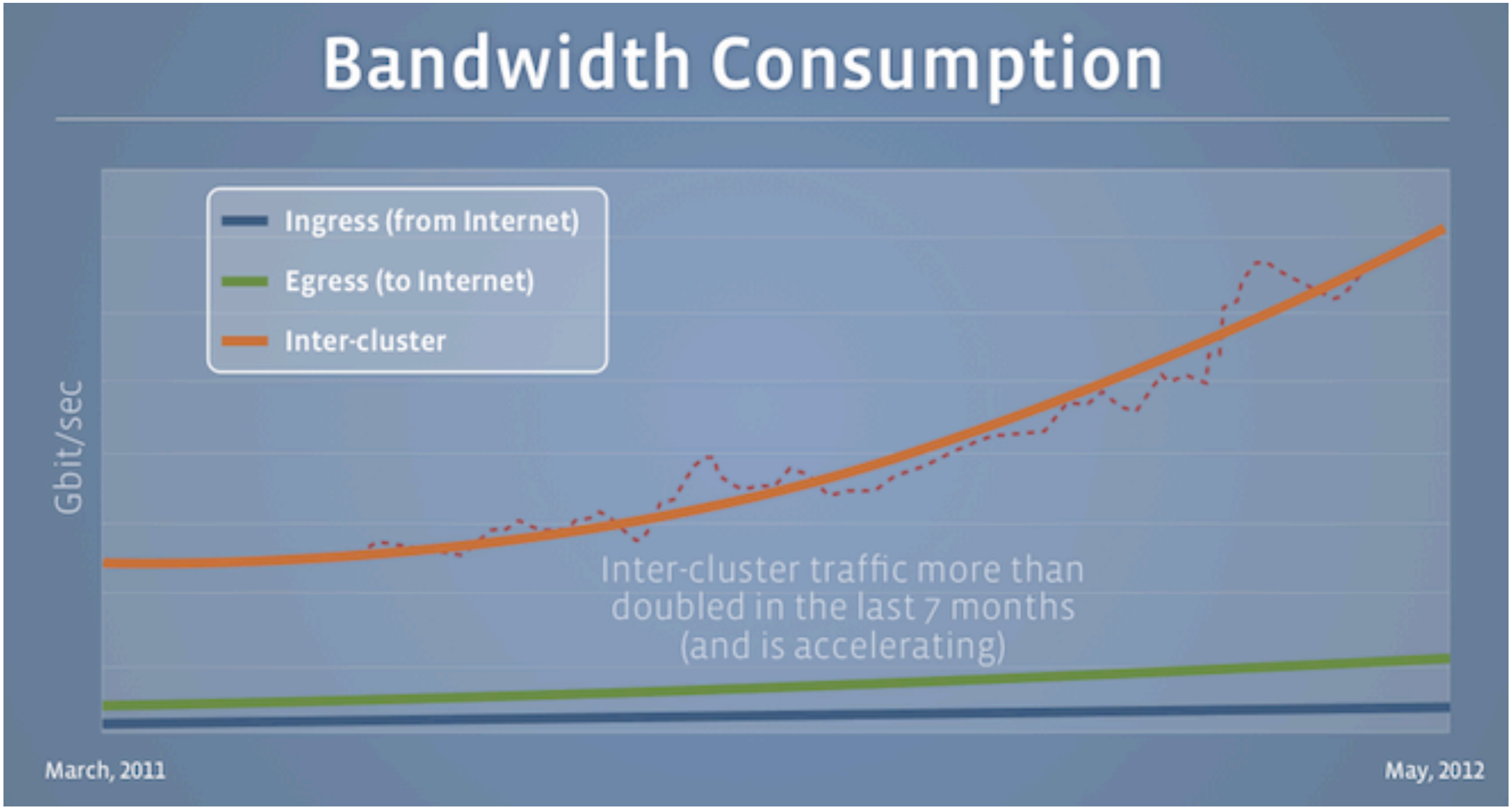
Sangeetha Abdu Jyothi
UIUC



Alexandra Kolla
UIUC

Lucian Popa
HP Labs

The need for throughput



March 2011

May 2012

[Facebook, via Wired]

Difficult goals

High throughput with minimal cost

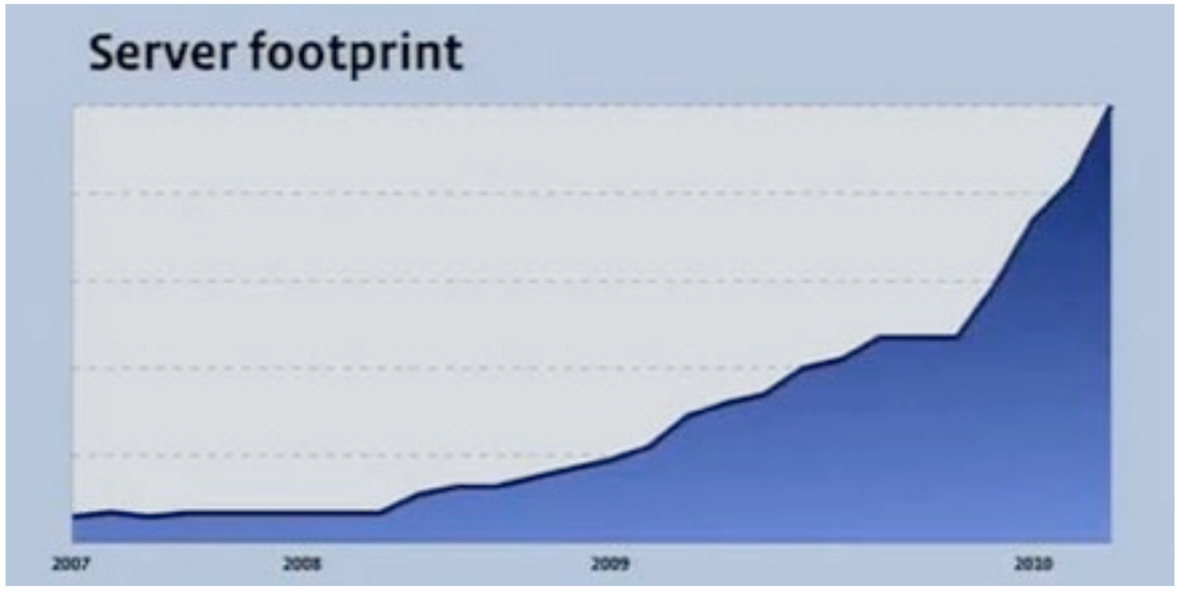
Support big data analytics
Agile placement of VMs

Flexible incremental expandability

Easily add/replace servers & switches

Incremental expansion

Facebook "adding capacity on a daily basis"



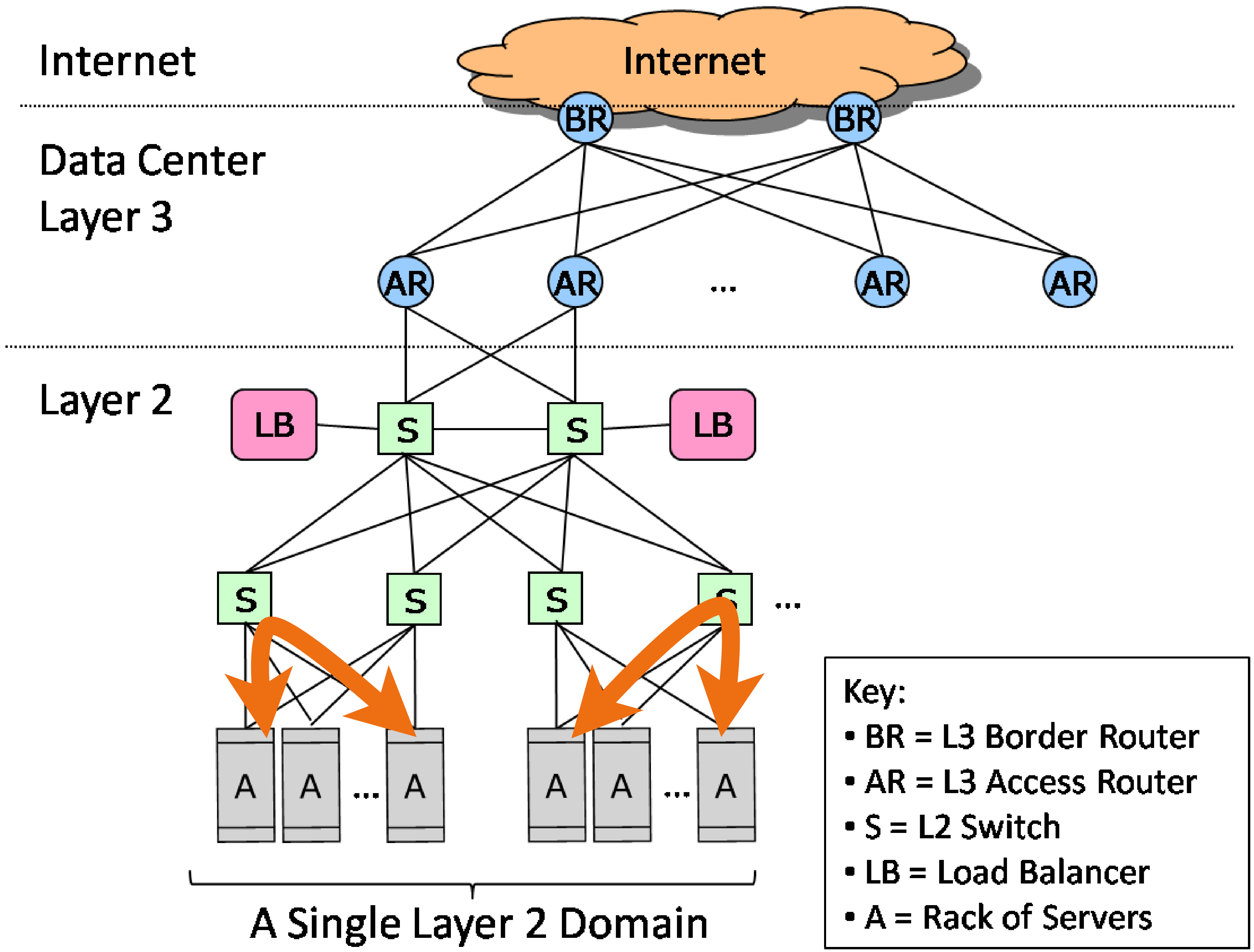
2007 08 09 10

Reduces up-front capital expenditure

Commercial products expand servers but not the net

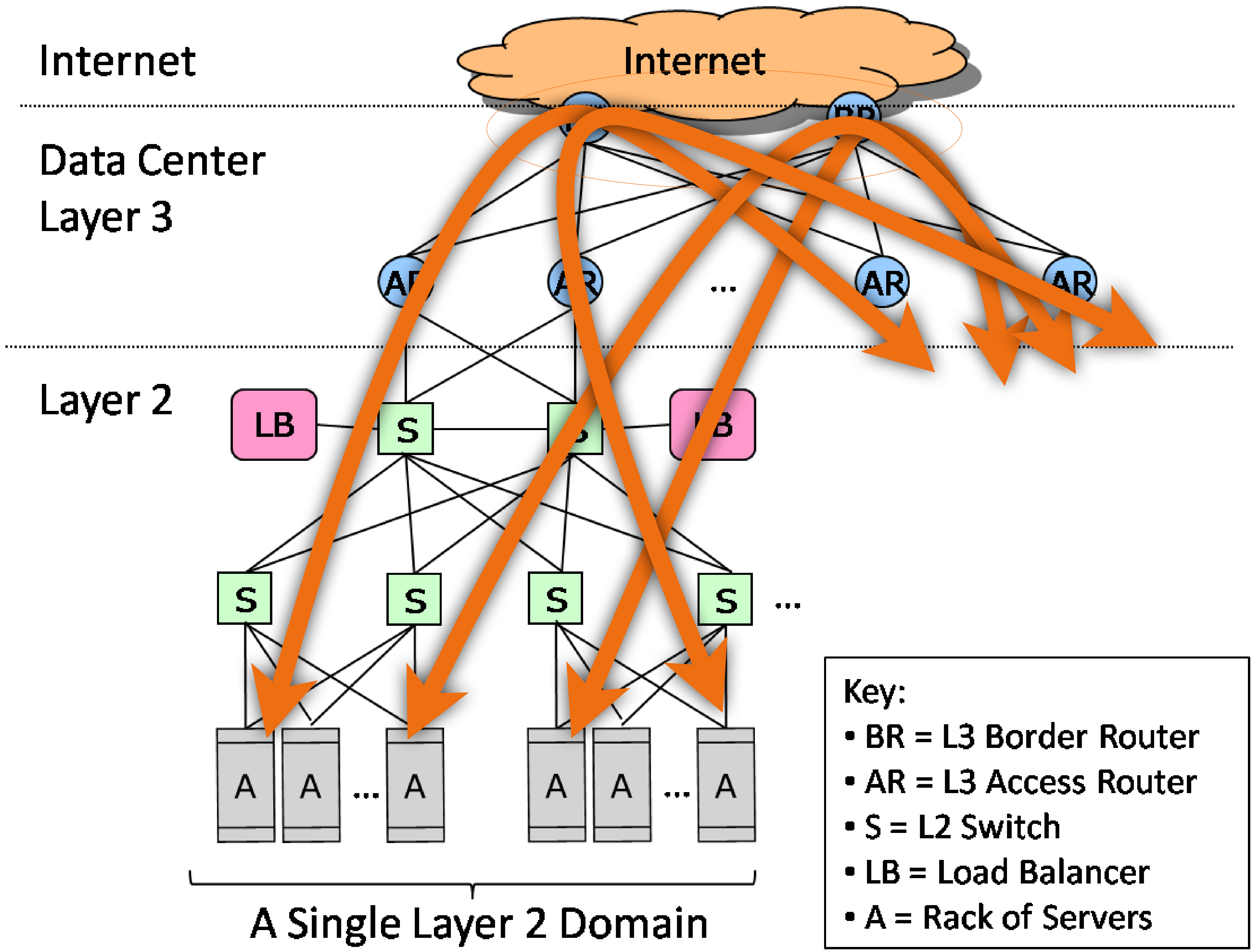
- SGI Ice Cube ("Expandable Modular Data Center")
- HP EcoPod ("Pay-as-you-grow")

Today's structured networks



[Greenberg et al, CCR Jan. 2009]

Today's structured networks



[Greenberg et al, CCR Jan. 2009]

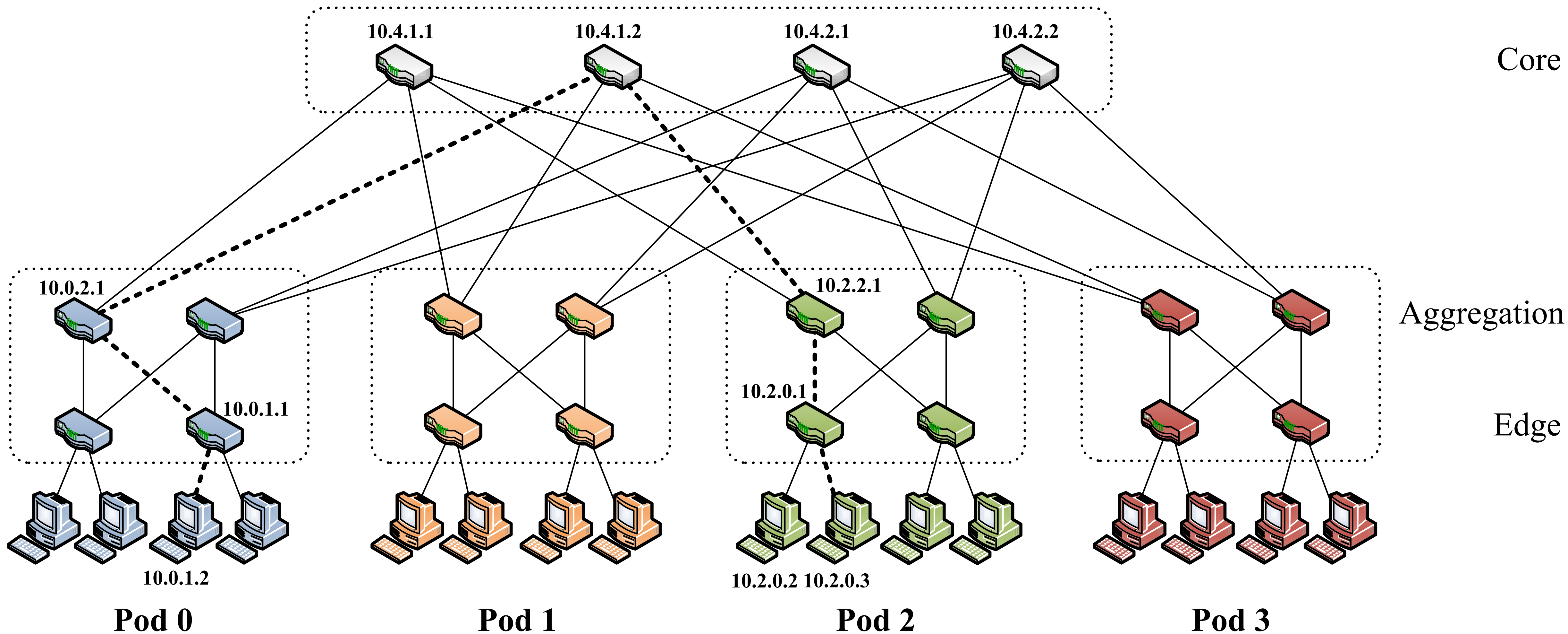
Today's structured networks

Fat tree

[Al-Fares, Loukissas, Vahdat, SIGCOMM '08]

Today's structured networks

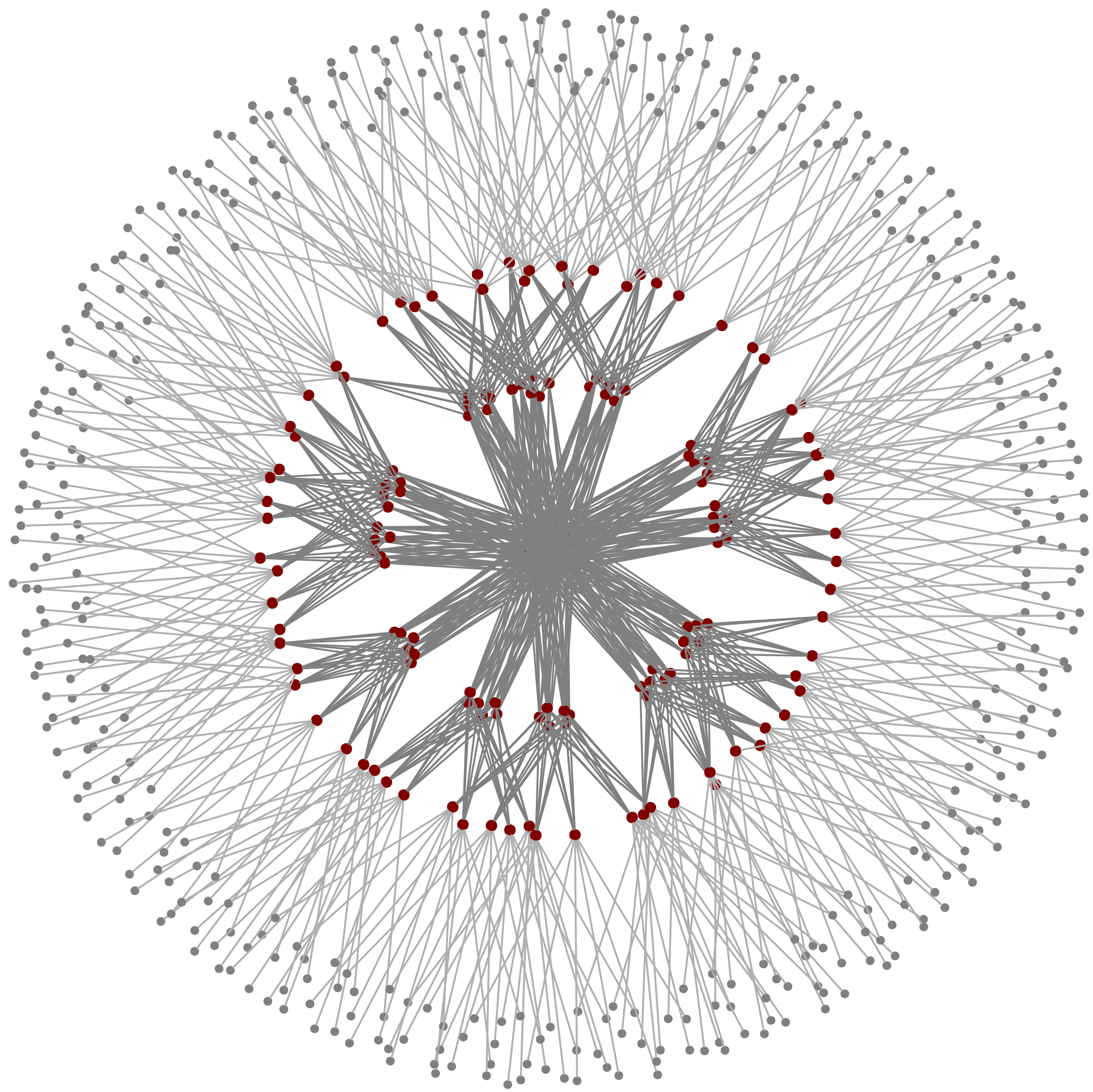
Fat tree



[Al-Fares, Loukissas, Vahdat, SIGCOMM '08]

Today's structured networks

Fat tree



Structure constrains expansion

Coarse design points

- Hypercube: 2^k switches
- de Bruijn-like: 3^k switches
- 3-level fat tree: $5k^2/4$ switches

Fat trees by the numbers:

- (3-level, with commodity 24, 32, 48, ... port switches)
- 3456 servers, 8192 servers, 27648 servers, ...

Unclear how to maintain structure incrementally

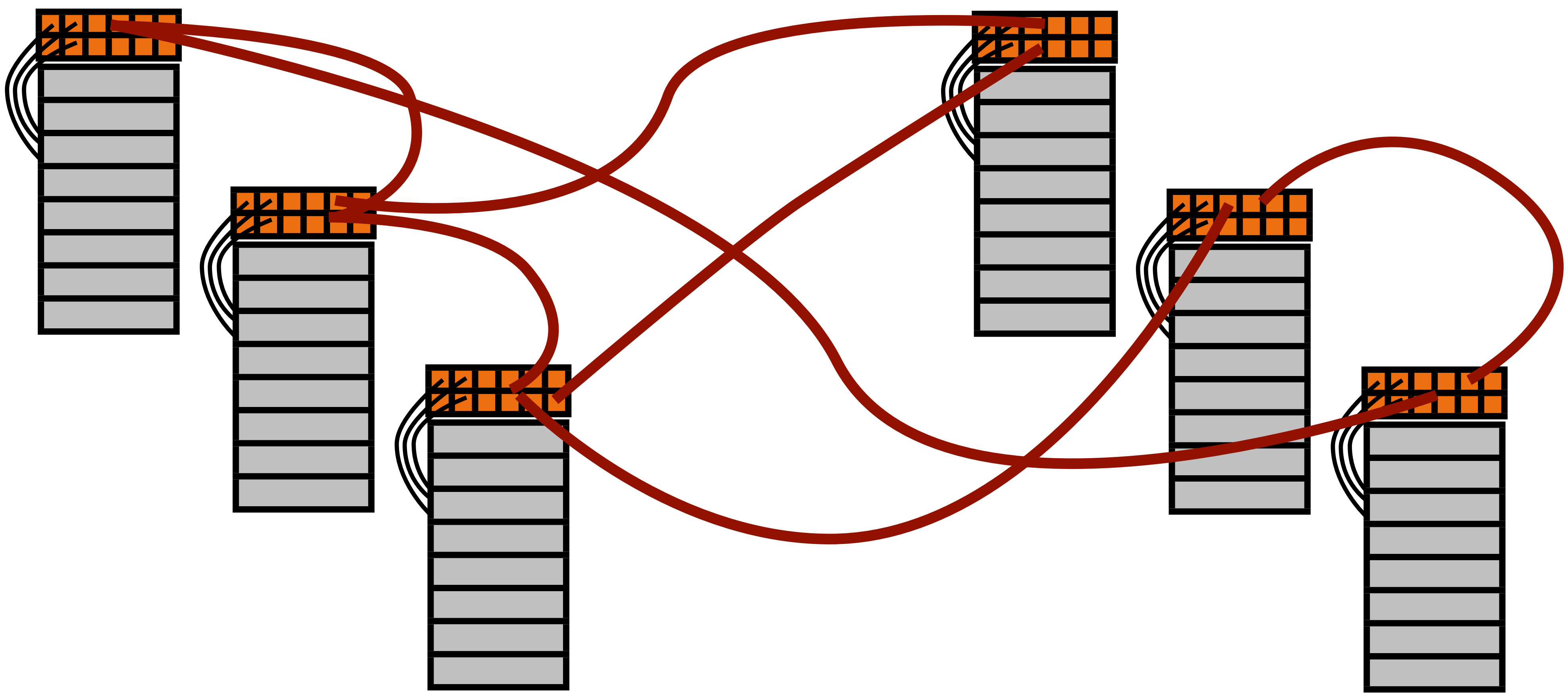
- Overutilize switches? Uneven / constrained bandwidth
- Leave ports free for later? Wasted investment

Our Solution

Forget about structure –
let's have no structure at all!

Jellyfish:
The Topology

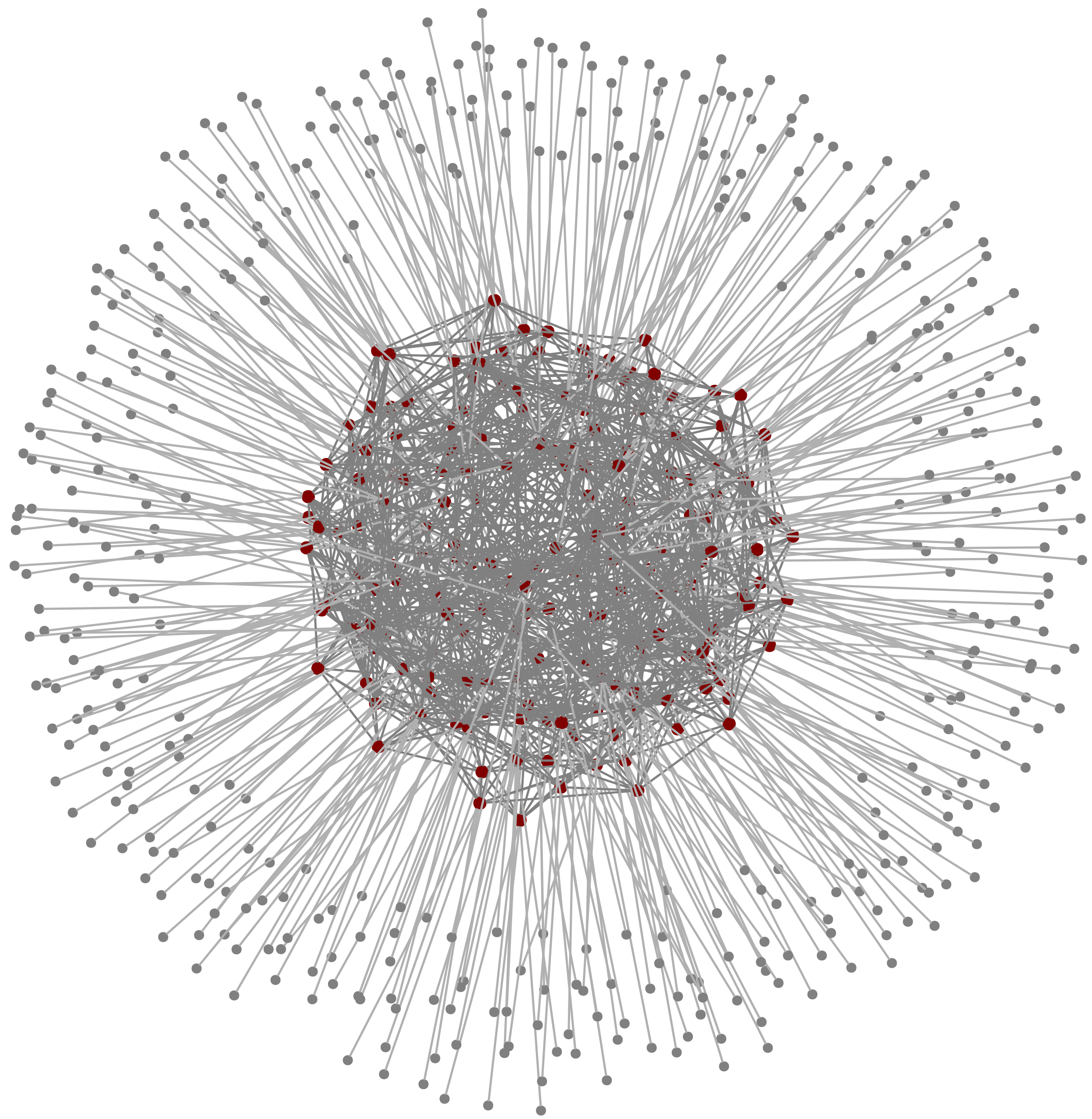
Jellyfish: The Topology



Servers connected to top-of-rack switch

Switches form uniform-random interconnections

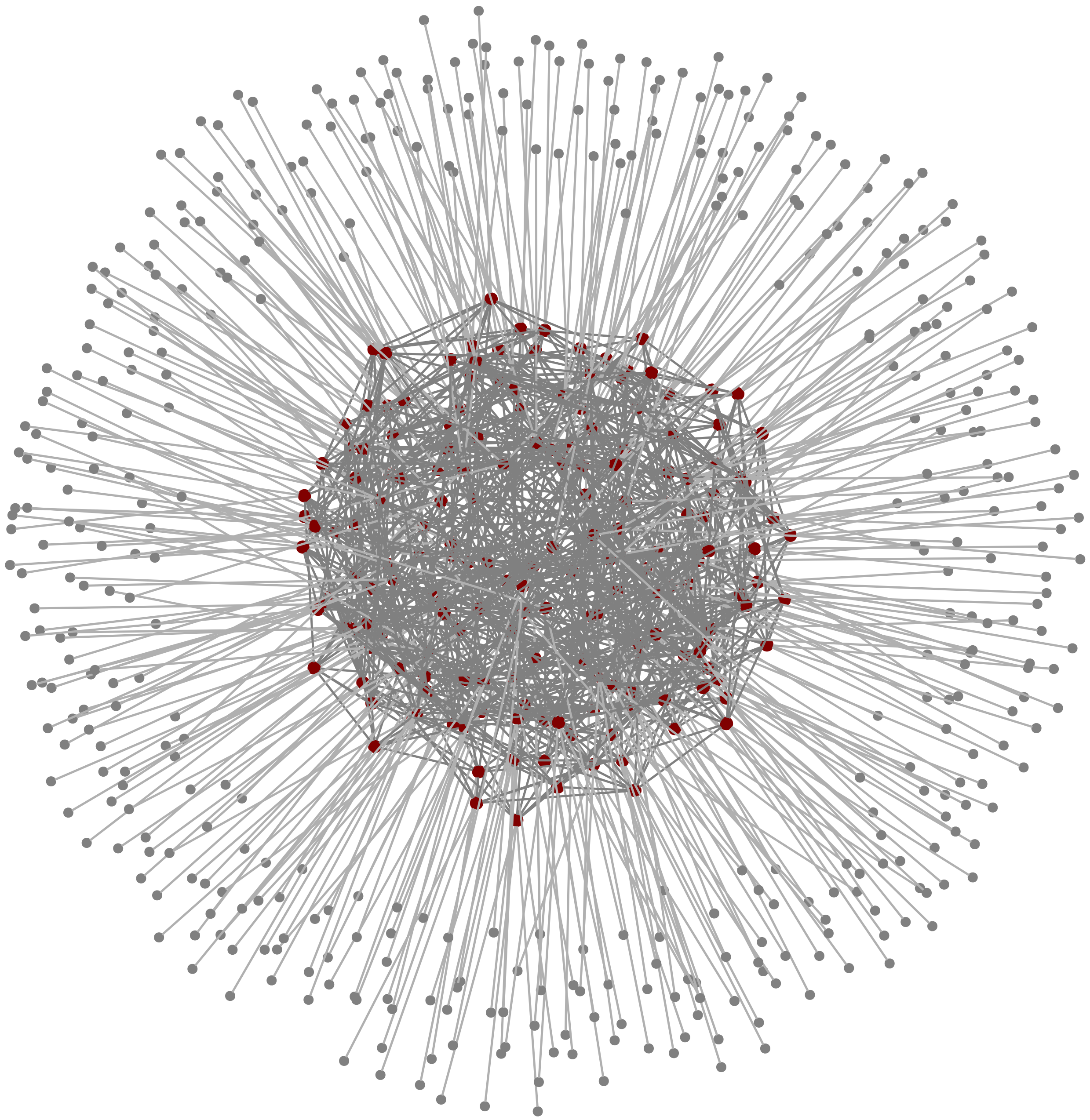
Capacity as a fluid



Jellyfish random graph

432 servers, 180 switches, degree 12

Capacity as a fluid



Jellyfish random graph

432 servers, 180 switches, degree 12

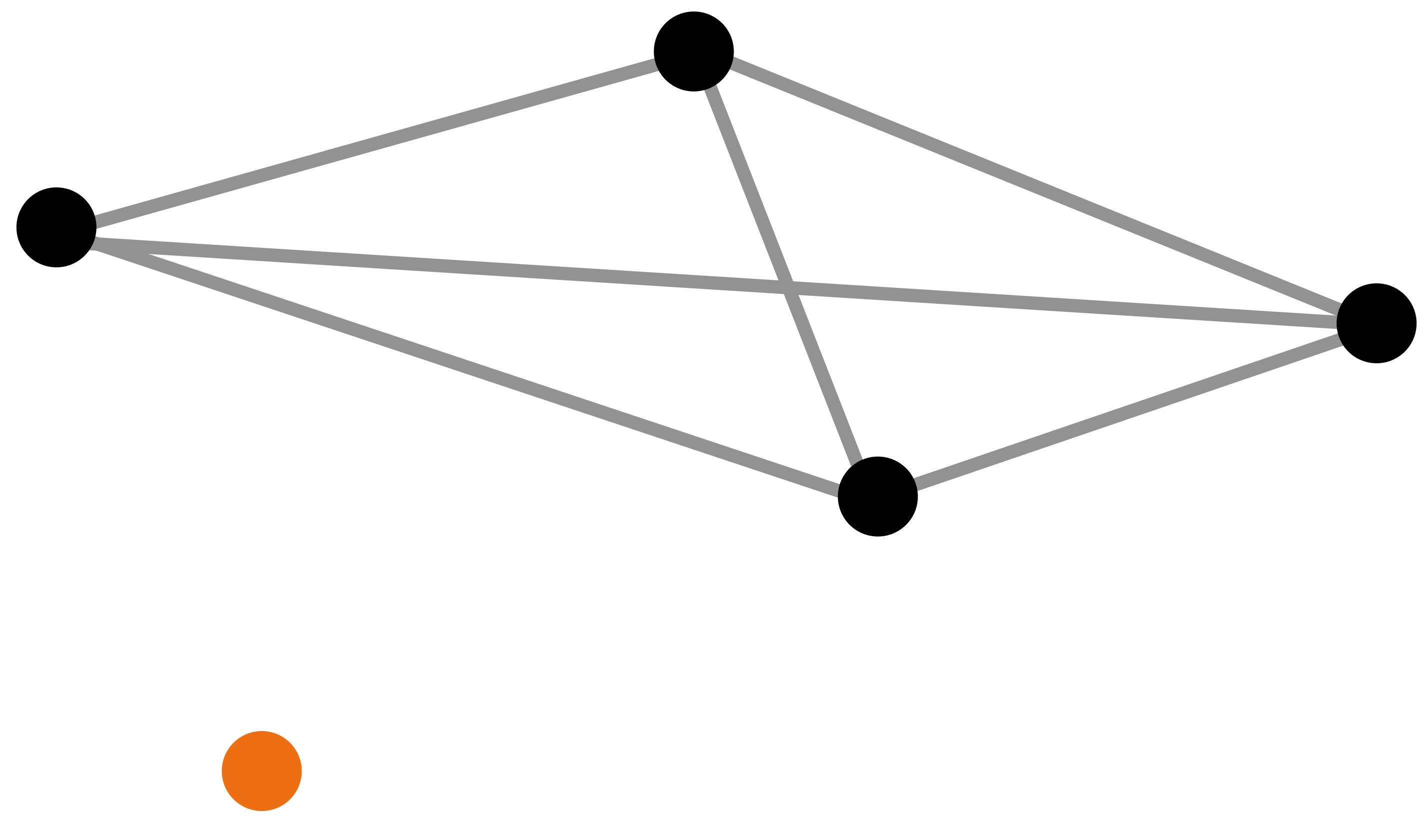
Jellyfish

Crossota norvegica

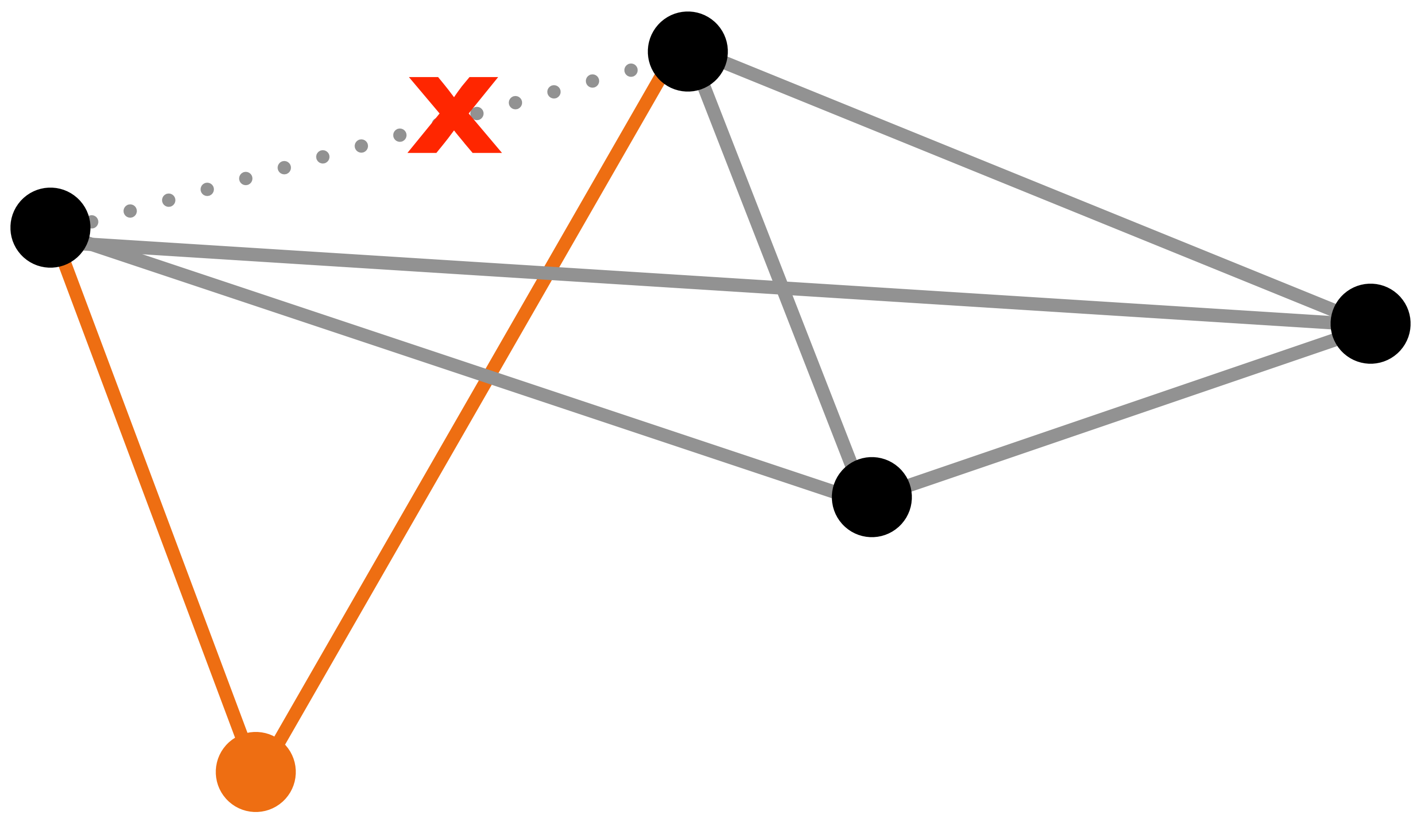
Photo: Kevin Raskoff

Construction & Expansion

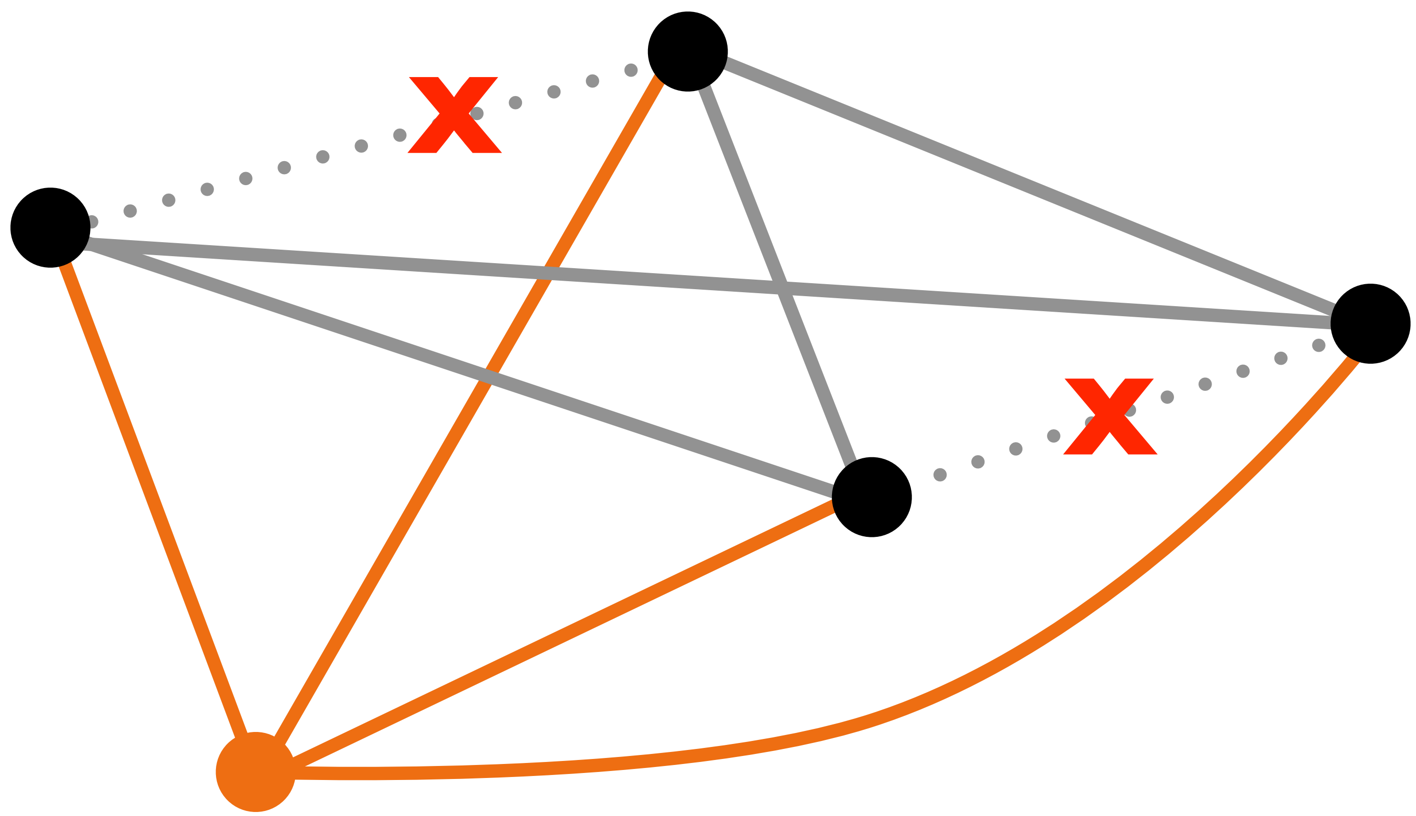
Building Jellyfish



Building Jellyfish



Building Jellyfish



Same procedure for initial construction
and incremental expansion

Can flexibly incorporate any type of equipment

Building Jellyfish

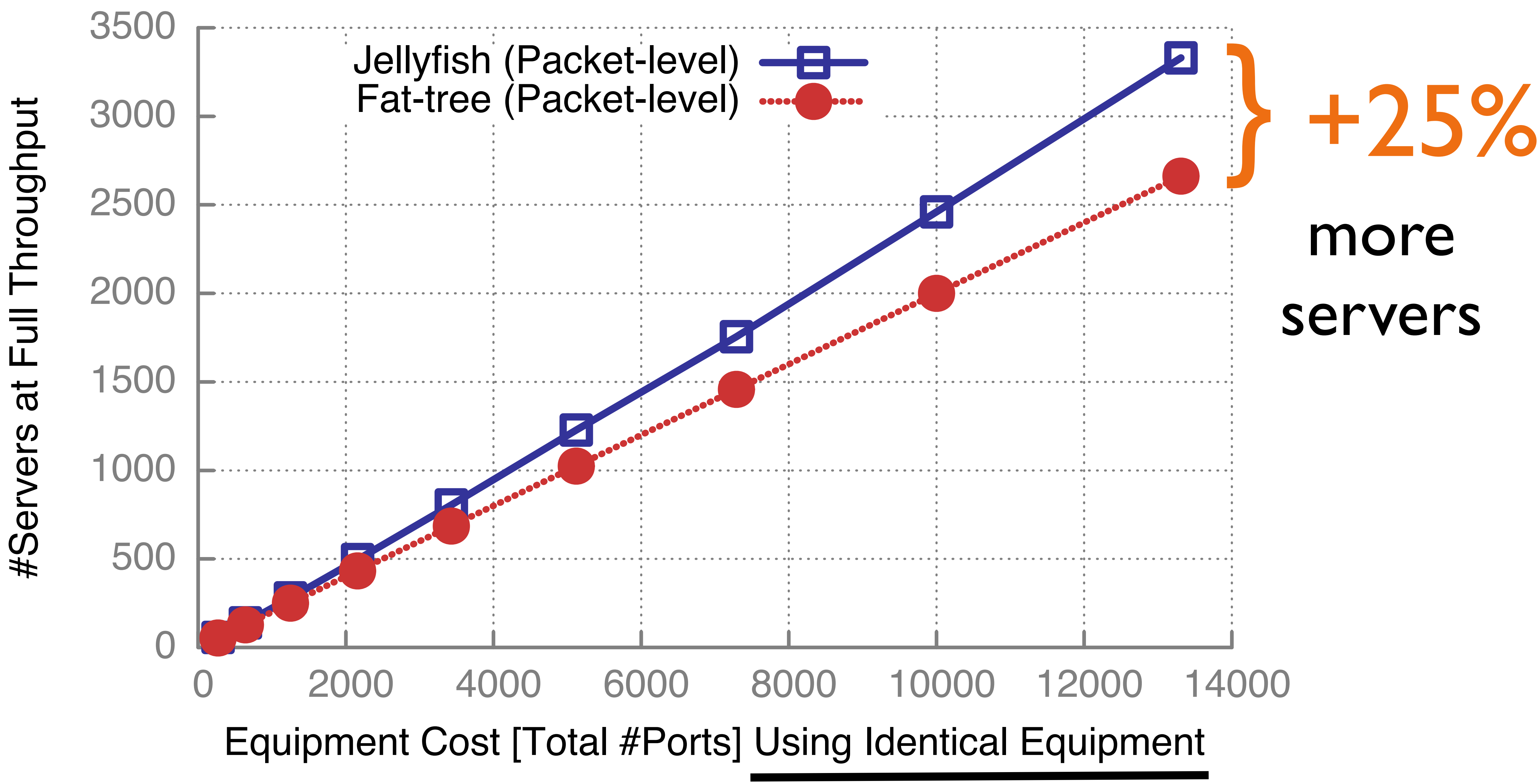
60% cheaper incremental expansion compared with past technique for traditional networks

LEGUP: [Curtis, Keshav, Lopez-Ortiz, CoNEXT'10]

Throughput

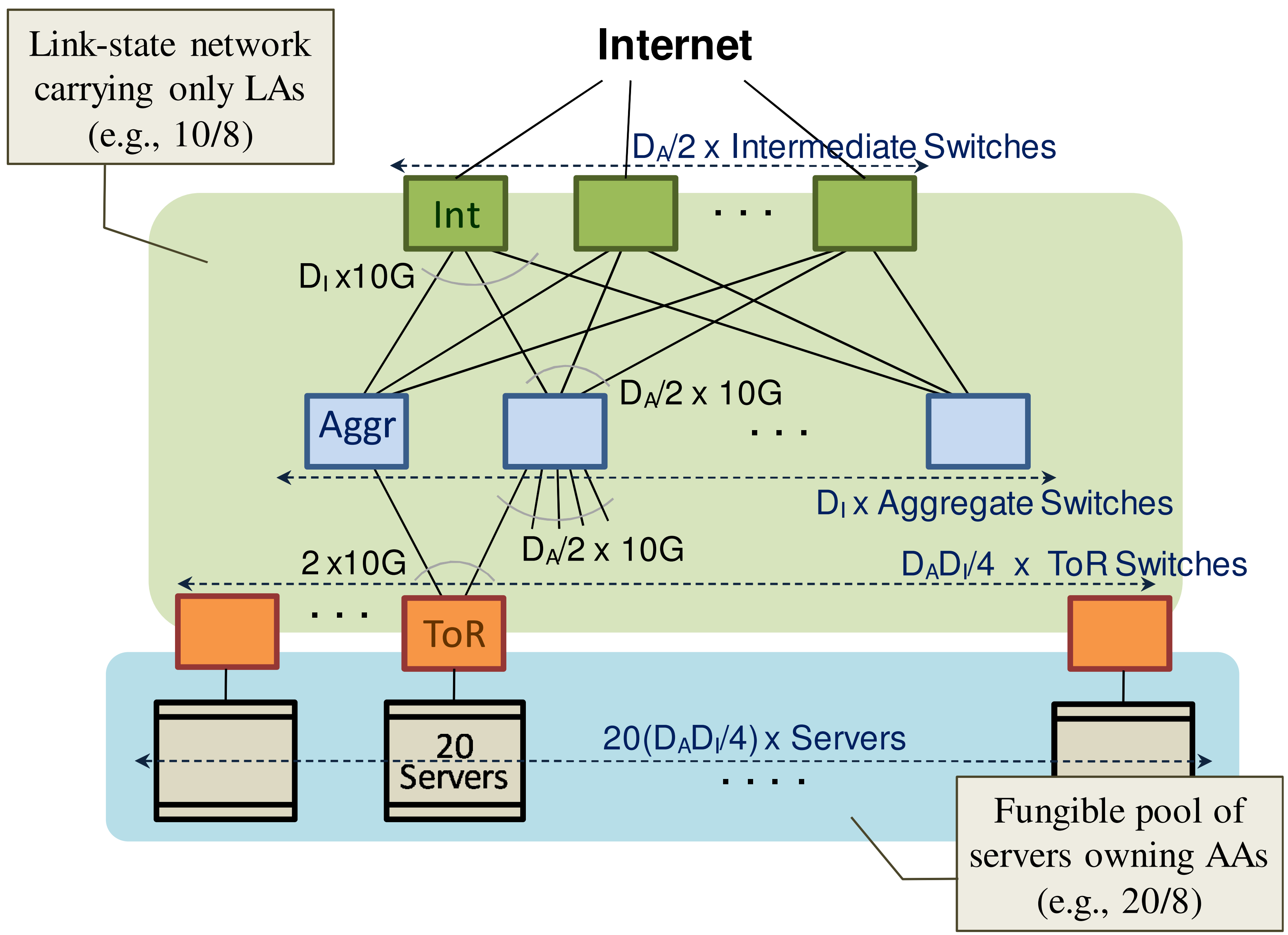
By giving up on structure,
do we take a hit on throughput?

Throughput: Jellyfish vs. fat tree

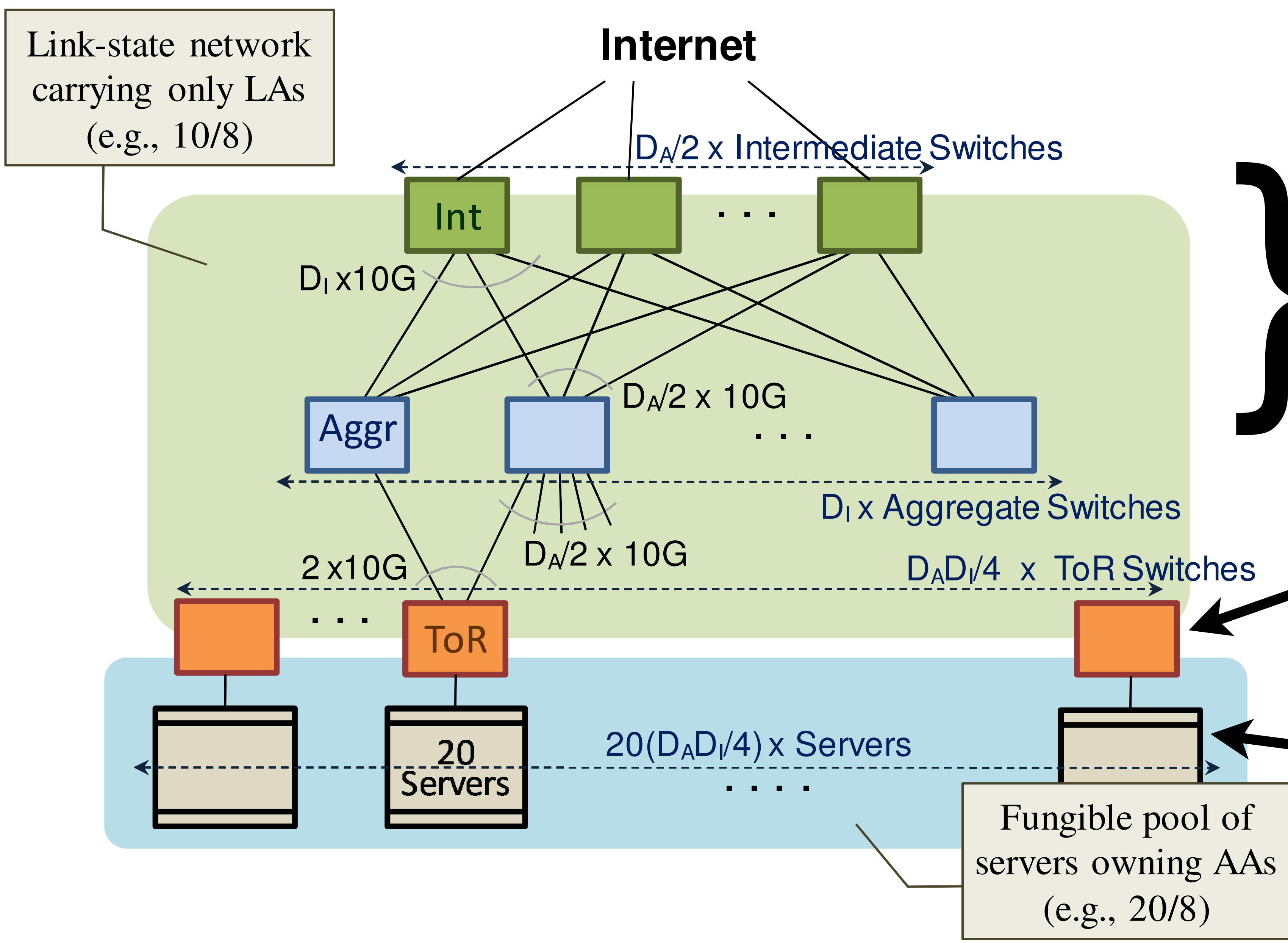


The VL2 topology

[Greenburg, Hamilton, Jain, Kandula, Kim, Lahiri, Maltz, Patel, Sengupta, SIGCOMM'09]



Rewiring VL2

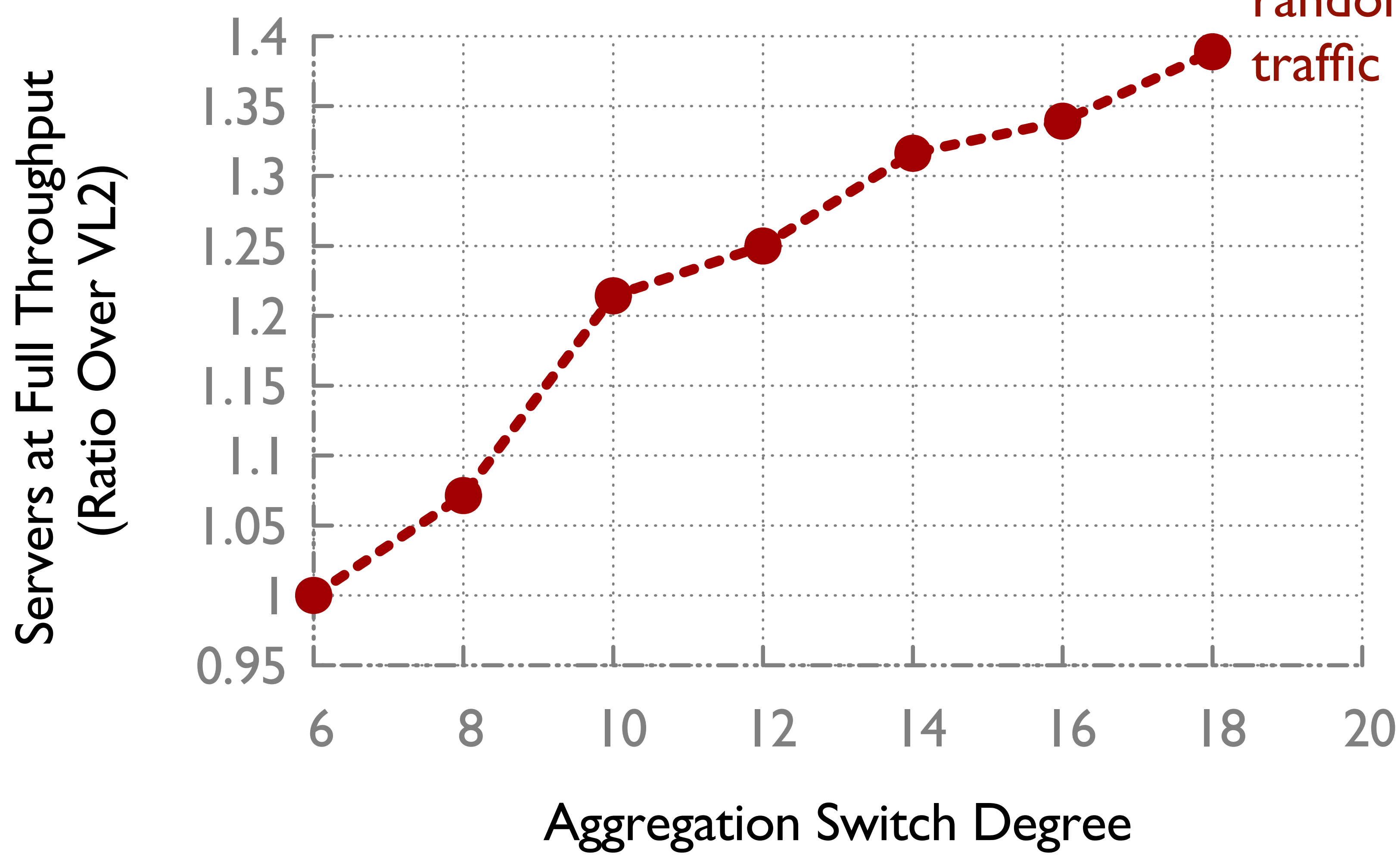


Uniform-random interconnection

Connect ToRs proportional to Intermediate/Agg degree

Servers unchanged (only ToRs have 1 Gbps ports)

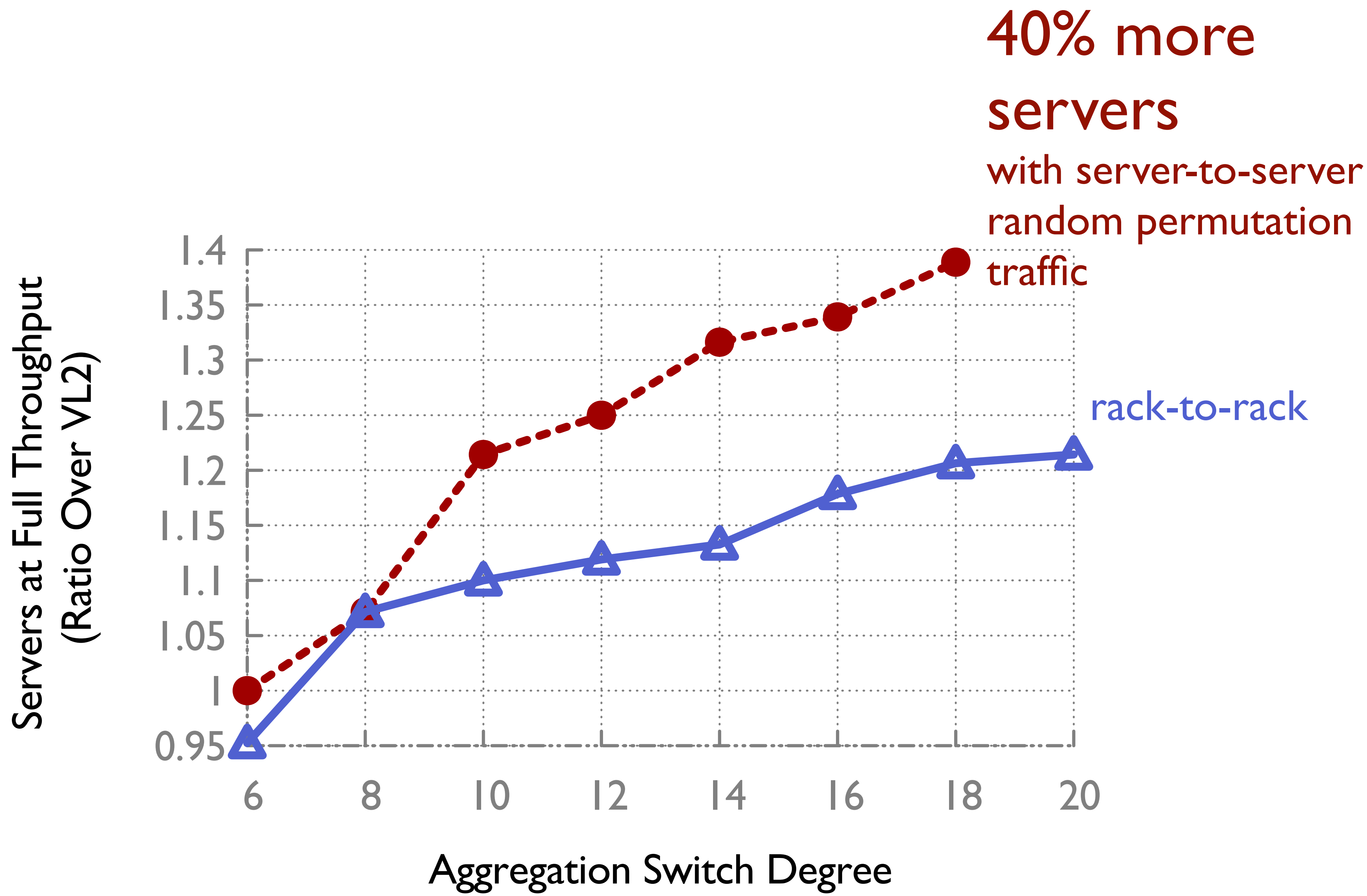
Rewiring VL2



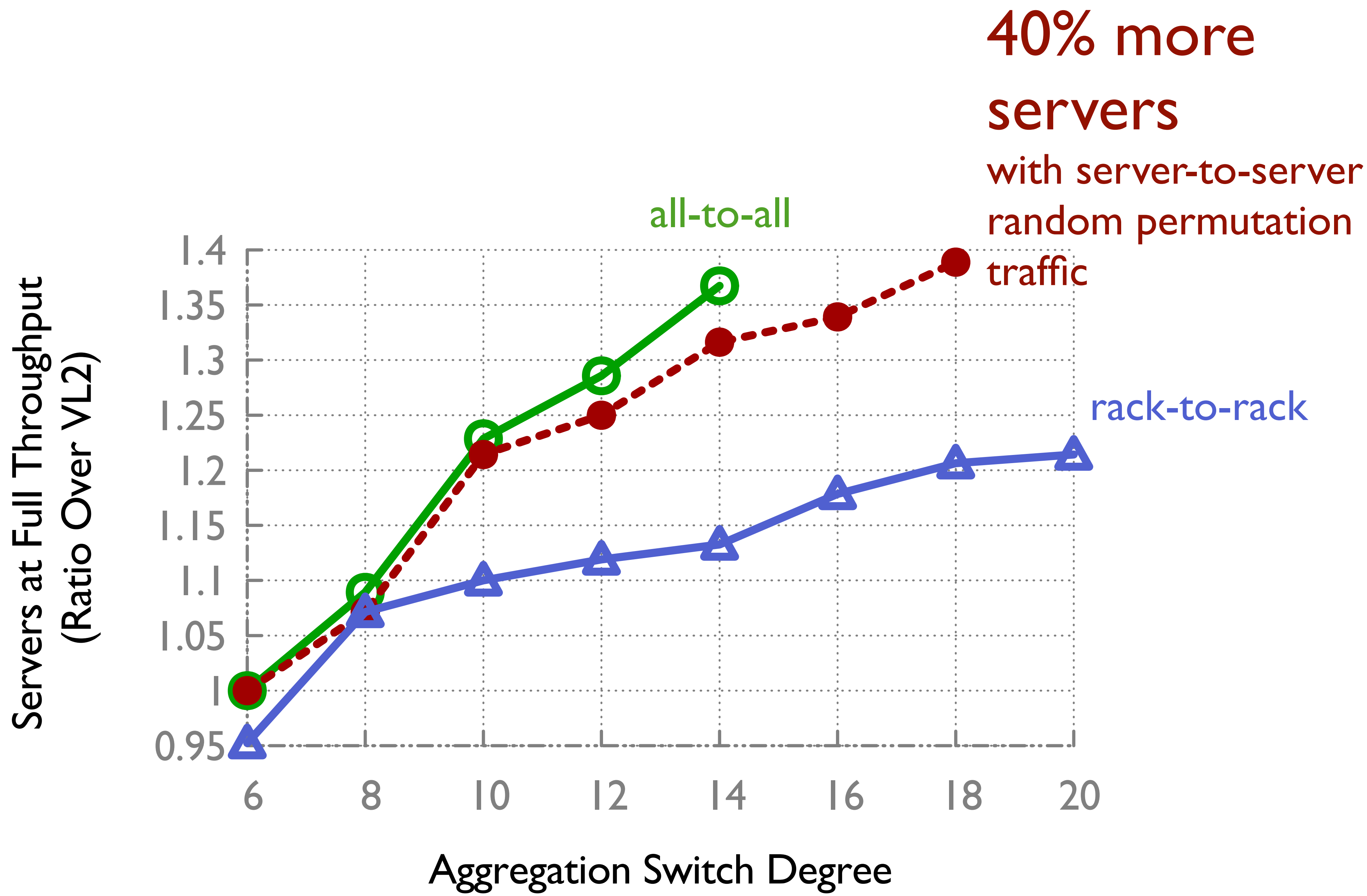
40% more servers

with server-to-server random permutation traffic

Rewiring VL2



Rewiring VL2



Just the beginning

Just the beginning

Topology design

- How close are random graphs to optimal?
- What if switches are heterogeneous?

System design (or: *"But what about..."*)

- Performance consistency?
- Cabling spaghetti?
- Routing and congestion control without structure?

Just the beginning

Topology design

- How close are random graphs to optimal?
- What if switches are heterogeneous?

System design (or: *"But what about..."*)

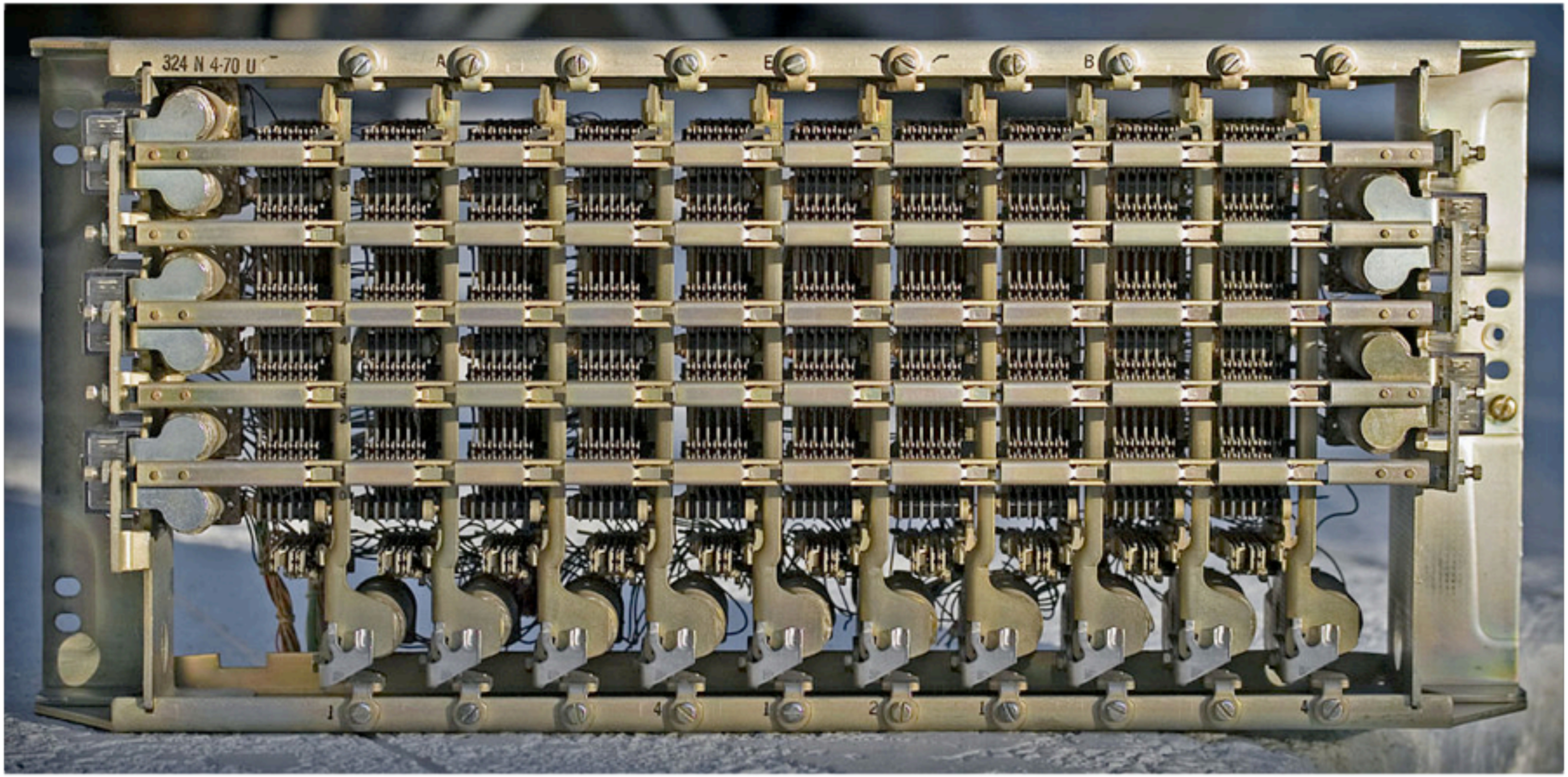
- Performance consistency?
- Cabling spaghetti?
- Routing and congestion control without structure?

Topology Design in Context

“It is anticipated that the whole of the populous parts of the United States will, within two or three years, be covered with net-work like a spider's web.”

“ *It is anticipated that the whole of the populous parts of the United States will, within two or three years, be covered with net-work like a spider's web.* ”

— *The London Anecdotes*, 1848



Western Electric crossbar switch

[Photo: Wikipedia user Yeatesh]

A Study of Non-Blocking Switching Networks

By CHARLES CLOS

(Manuscript received October 30, 1952)

This paper describes a method of designing arrays of crosspoints for use in telephone switching systems in which it will always be possible to establish a connection from an idle inlet to an idle outlet regardless of the number of calls served by the system.

INTRODUCTION

The impact of recent discoveries and developments in the electronic art is being felt in the telephone switching field. This is evidenced by

A Study of Non-Blocking Switching Networks

(Ma

This paper describ
use in telephone swit
establish a connectio
number of calls serve

INTRODUCTION

The impact of r

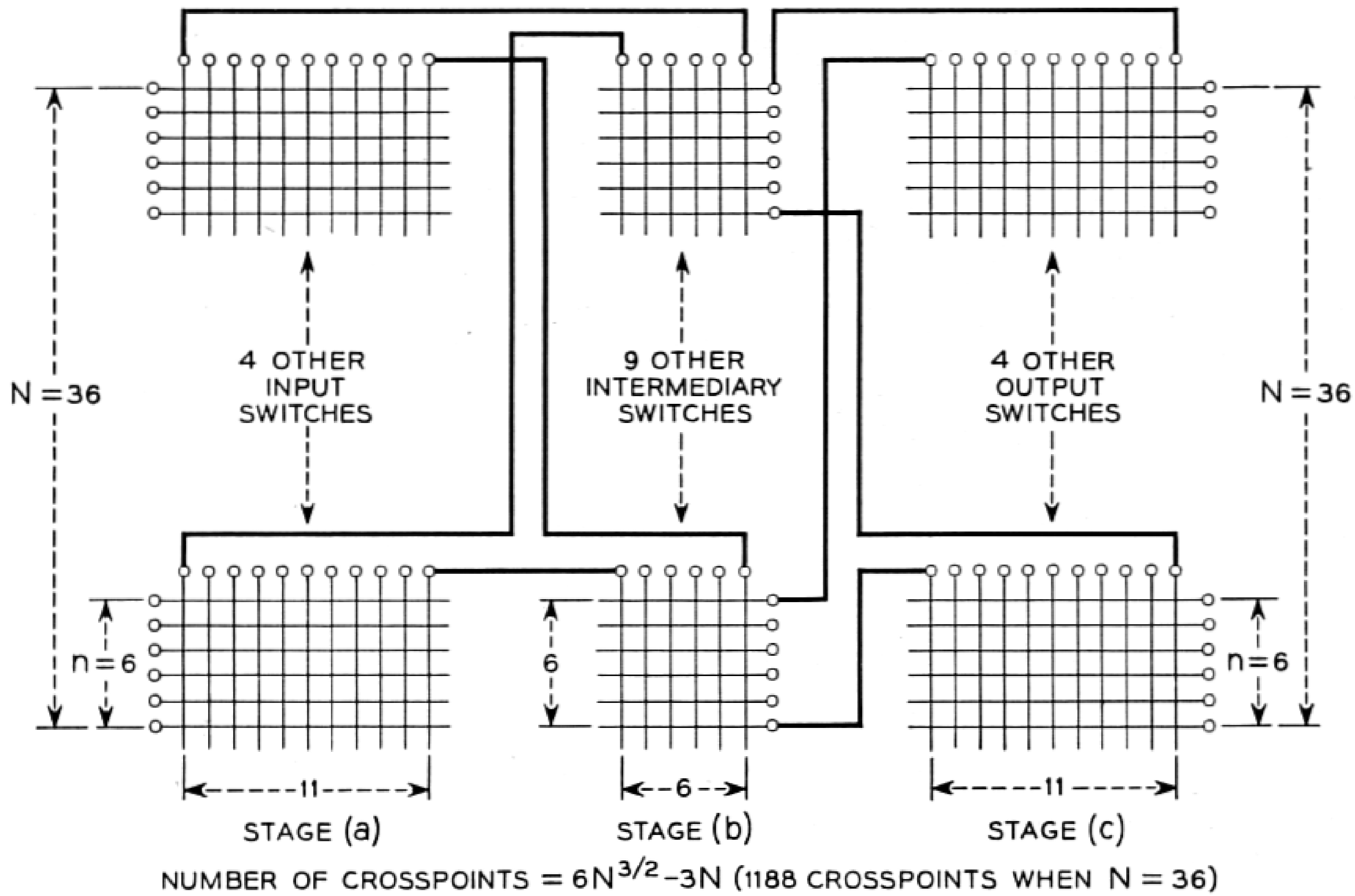
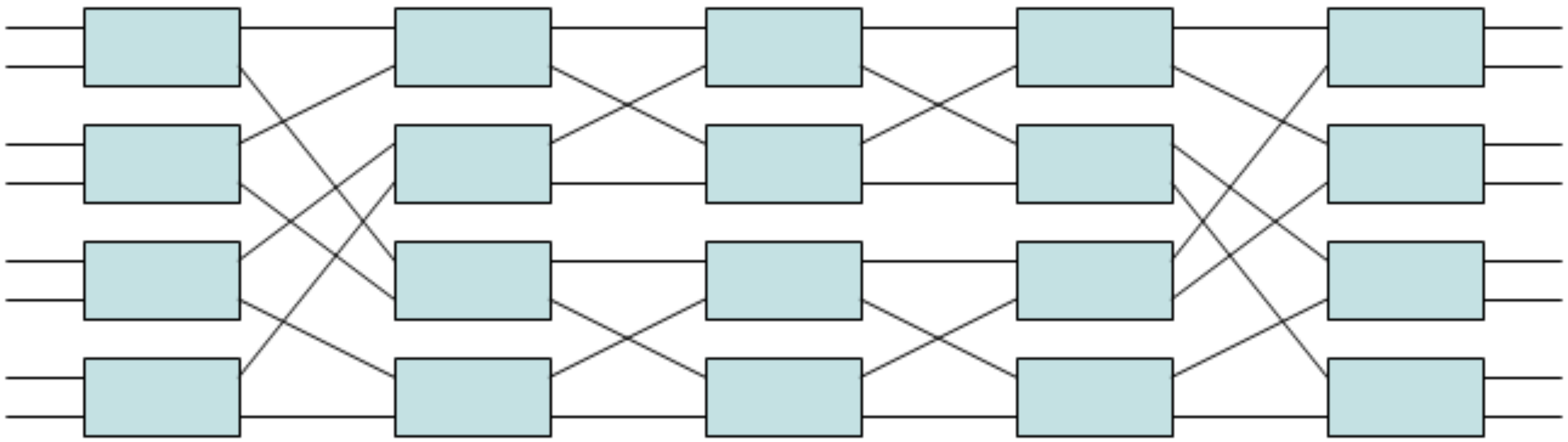
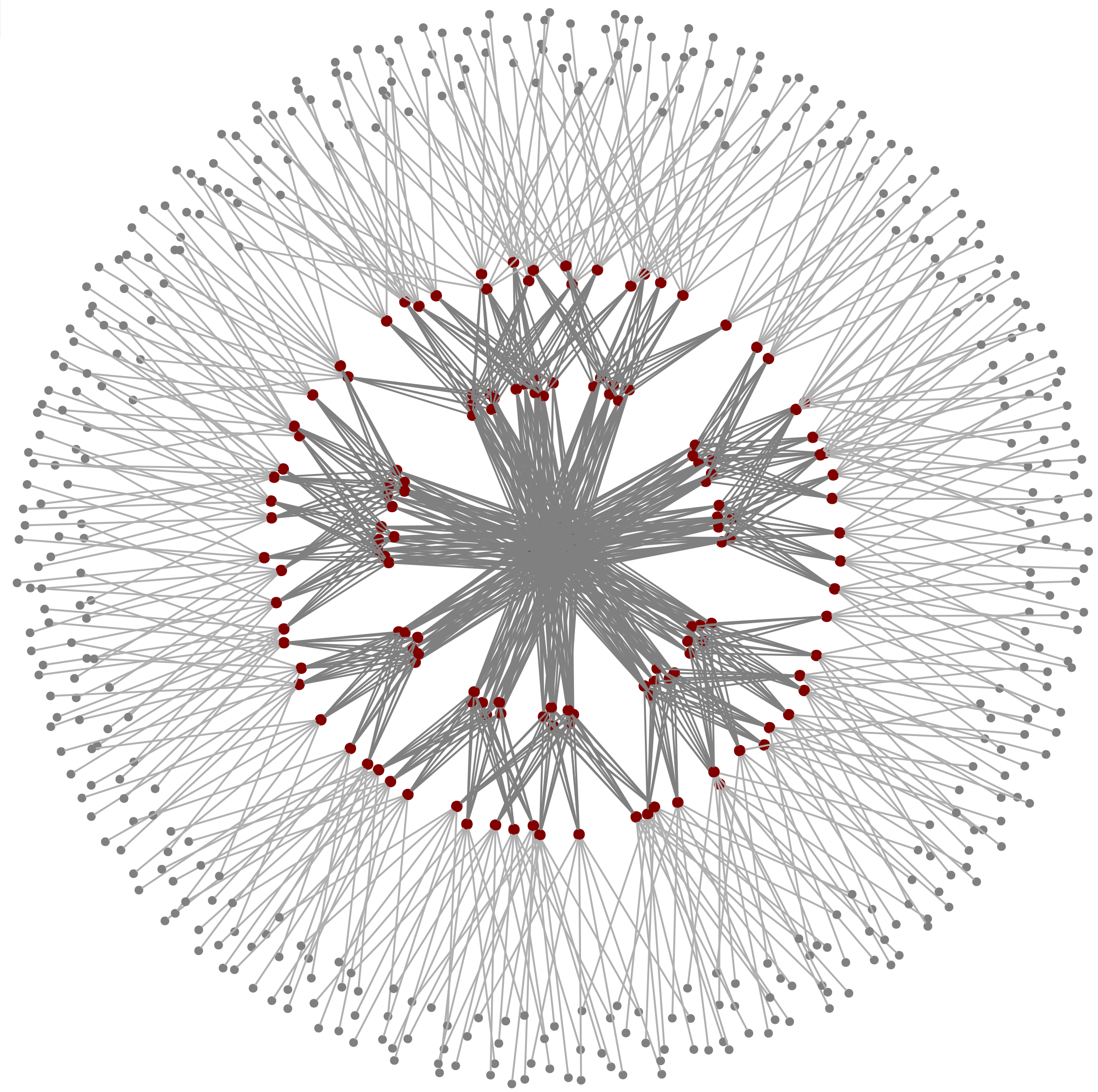


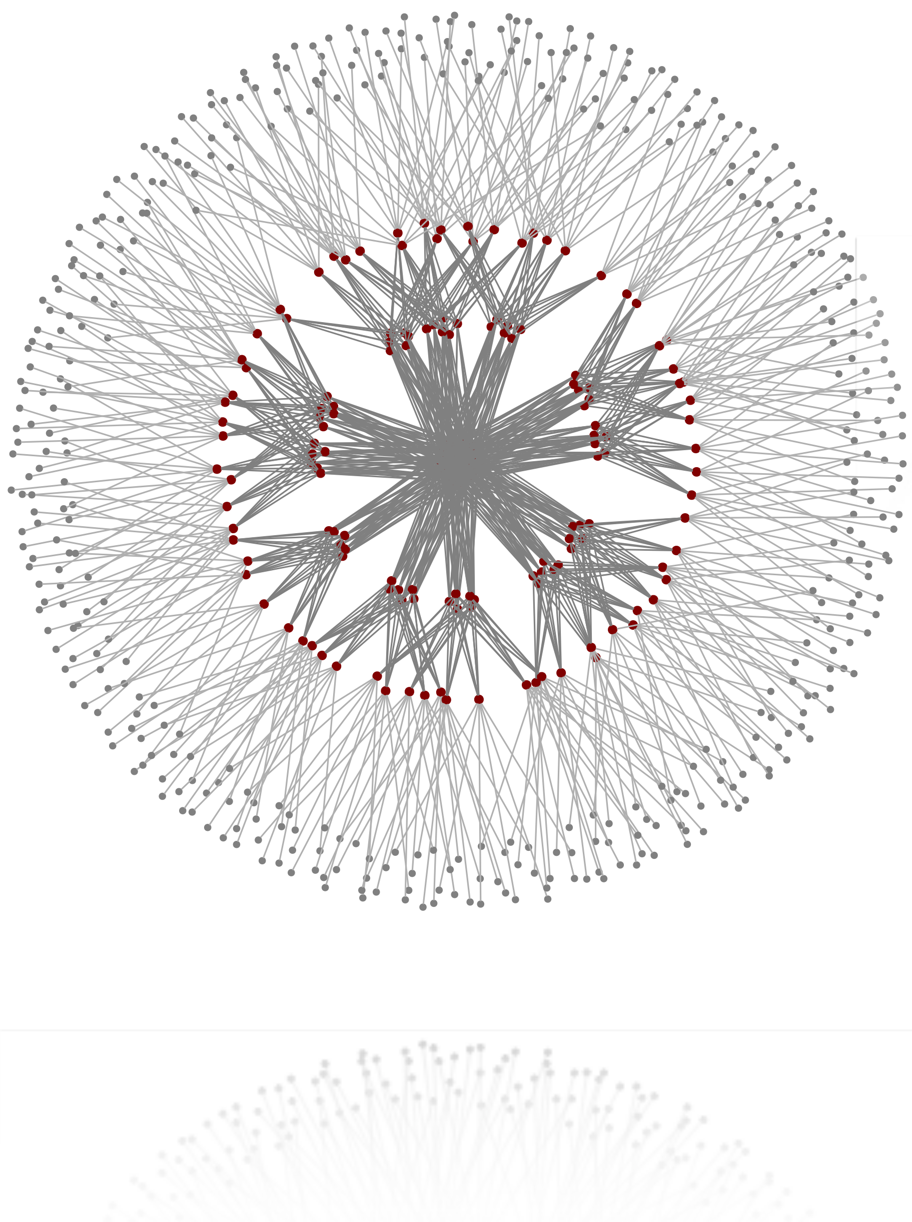
Fig. 2 — Three-stage switching array.

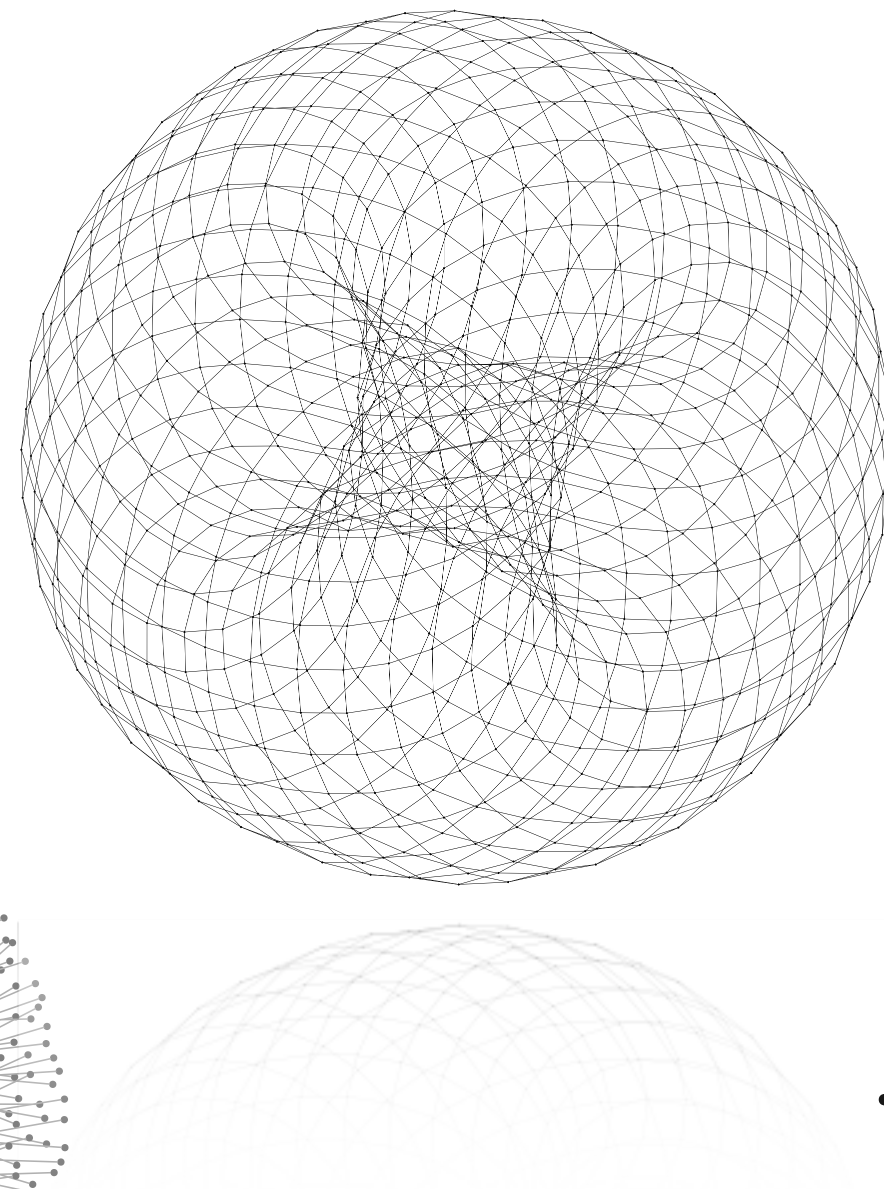


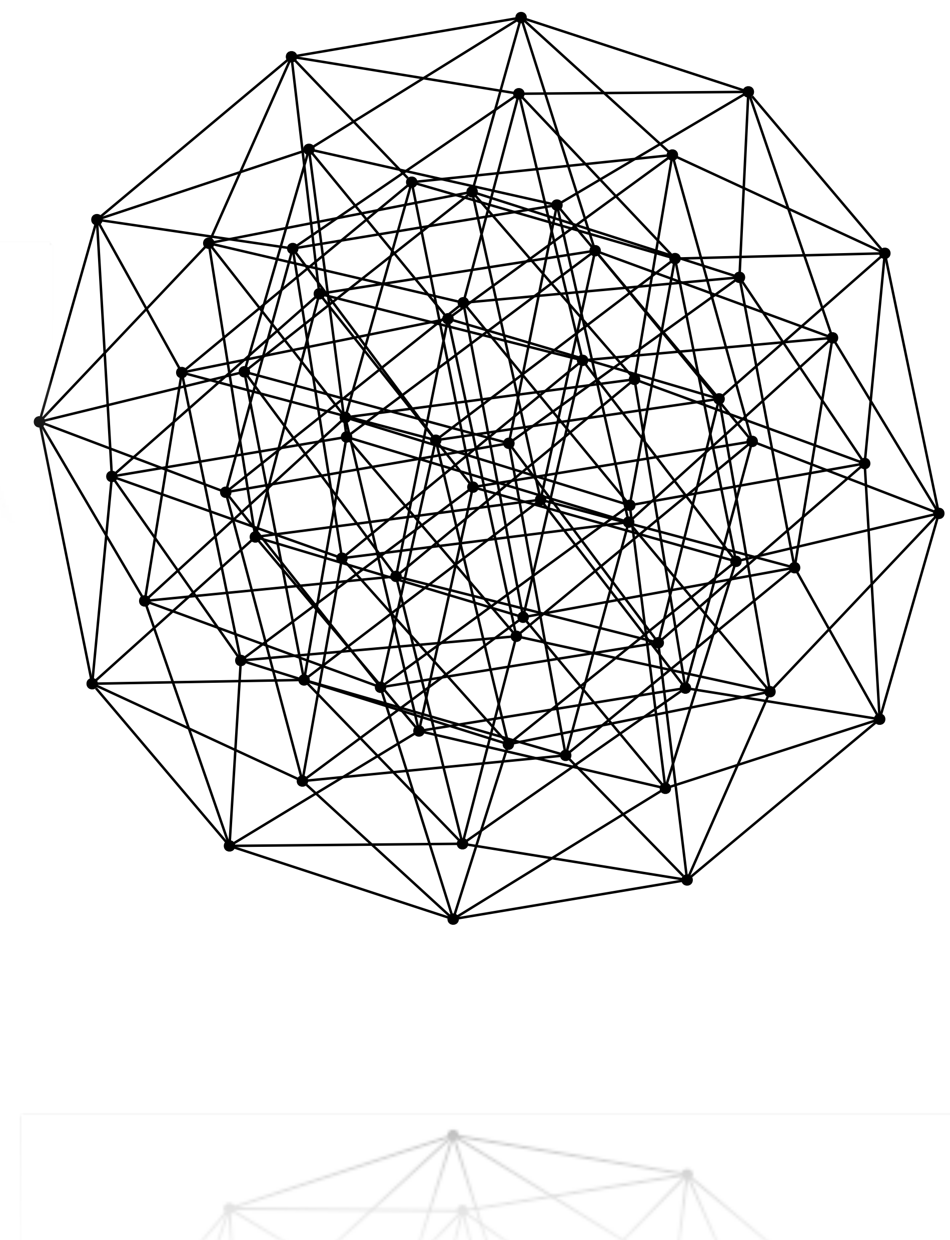
[Benes network: Wikipedia user Piggly]




CRAY
T3D







What's different about data centers

Flexible forwarding

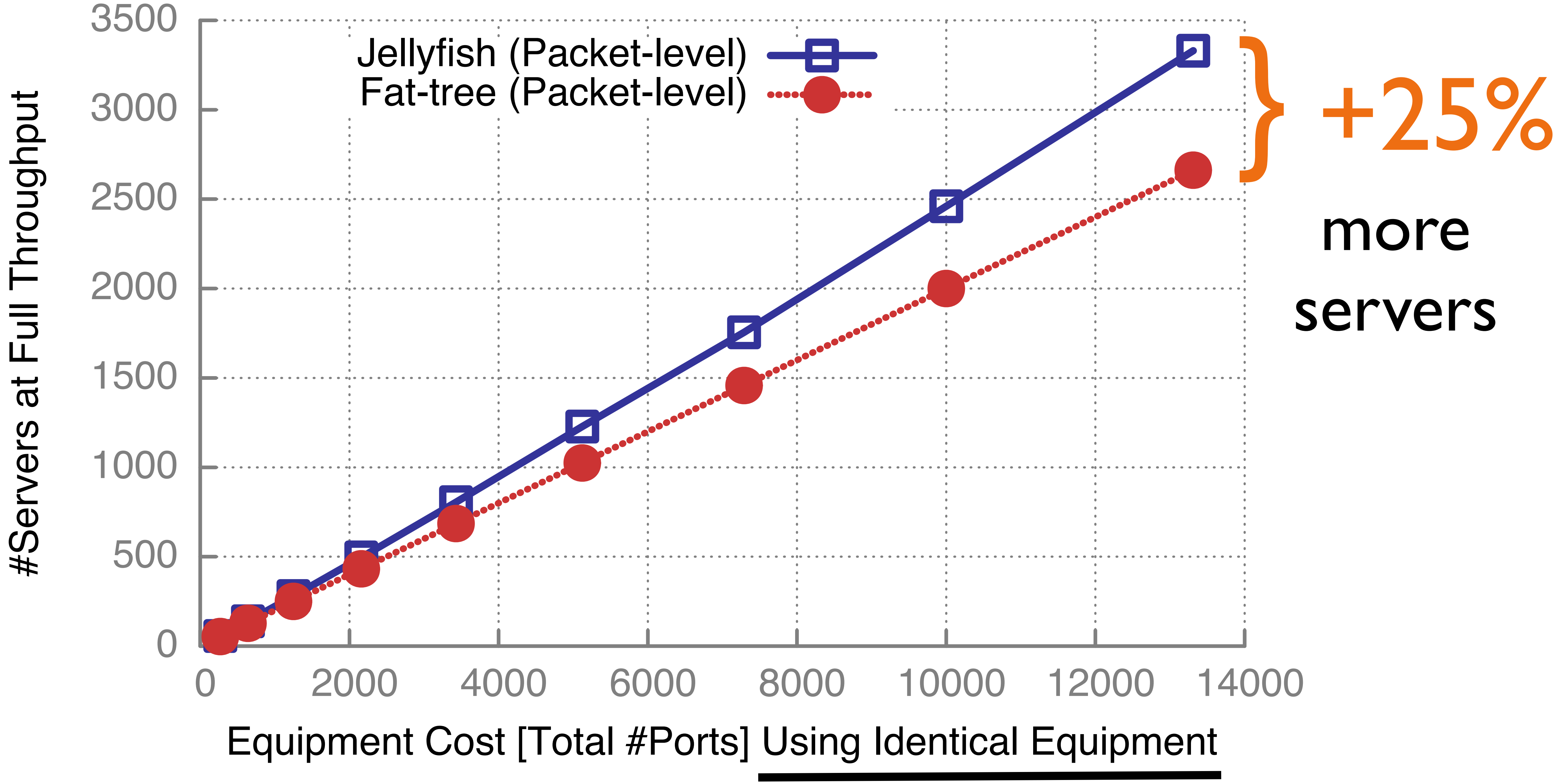
(compared with supercomputers)

Flexible routing & congestion control

(especially with software-defined networking)

Understanding Throughput

Throughput: Jellyfish vs. fat tree



Intuition

if we fully utilize all available capacity ...

$$\#\ 1\ \text{Gbps flows} = \frac{\text{total capacity}}{\text{used capacity per flow}}$$

Intuition

if we fully utilize all available capacity ...

$$\#\ 1\ \text{Gbps flows} = \frac{\sum_{\text{links}} \text{capacity}(\mathit{link})}{\text{used capacity per flow}}$$

Intuition

if we fully utilize all available capacity ...

$$\#\ 1\ \text{Gbps flows} = \frac{\sum_{\text{links}} \text{capacity}(\textit{link})}{1\ \text{Gbps} \cdot \text{mean path length}}$$

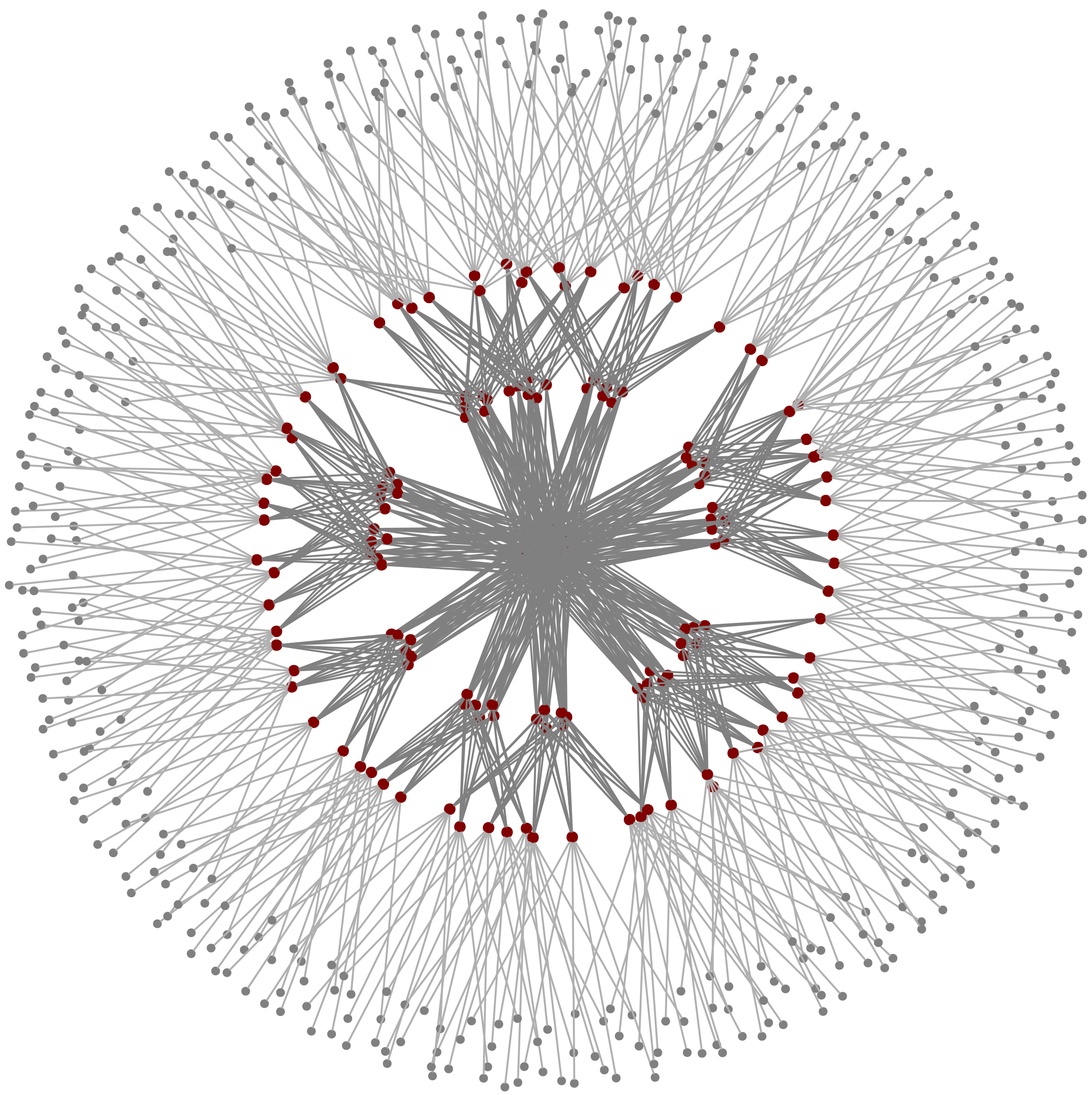
Intuition

if we fully utilize all available capacity ...

$$\#\ 1\ \text{Gbps flows} = \frac{\sum_{\text{links}} \text{capacity}(\textit{link})}{1\ \text{Gbps} \bullet \text{mean path length}}$$

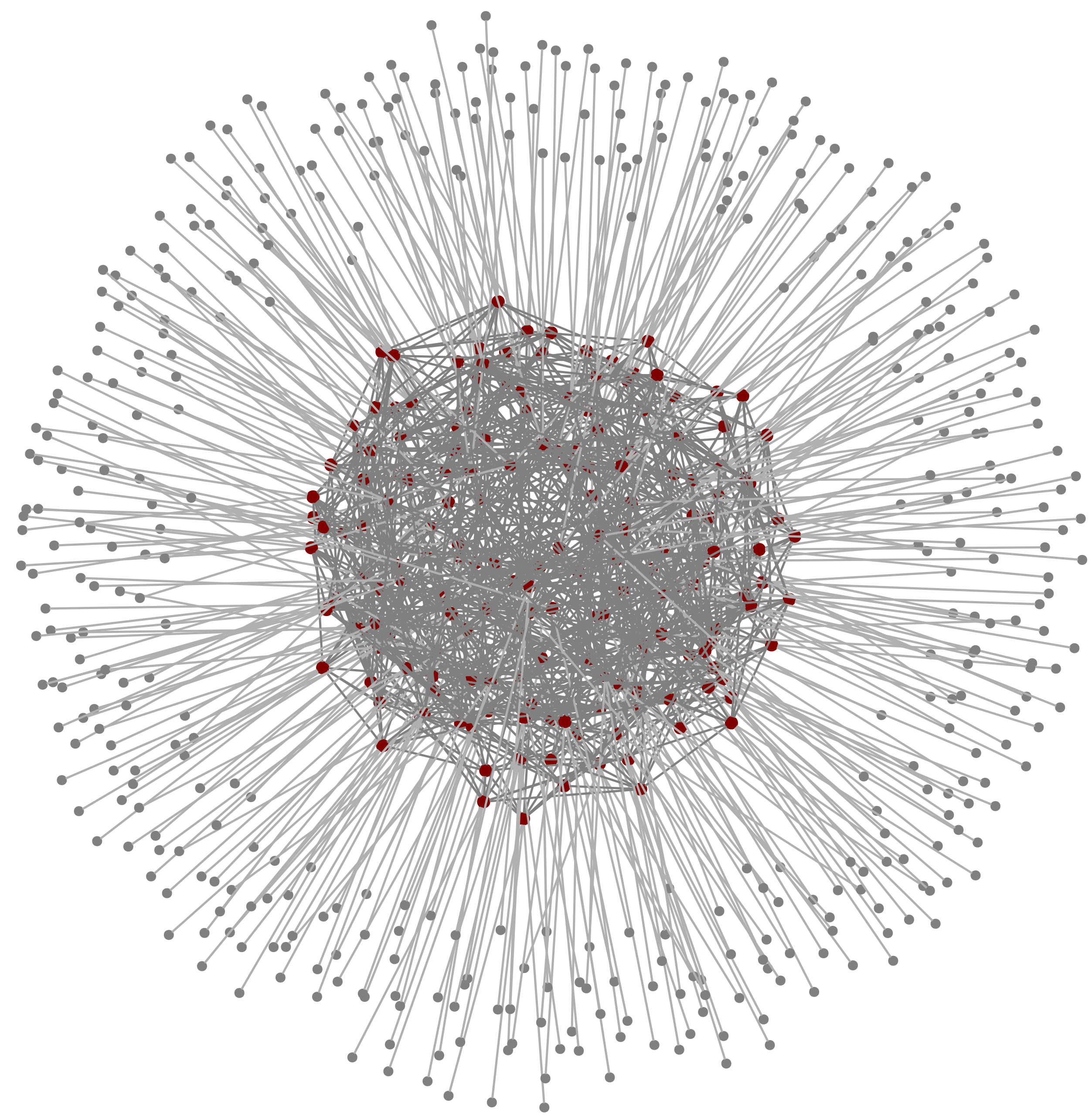
Mission:
minimize average path length

Example



Fat tree

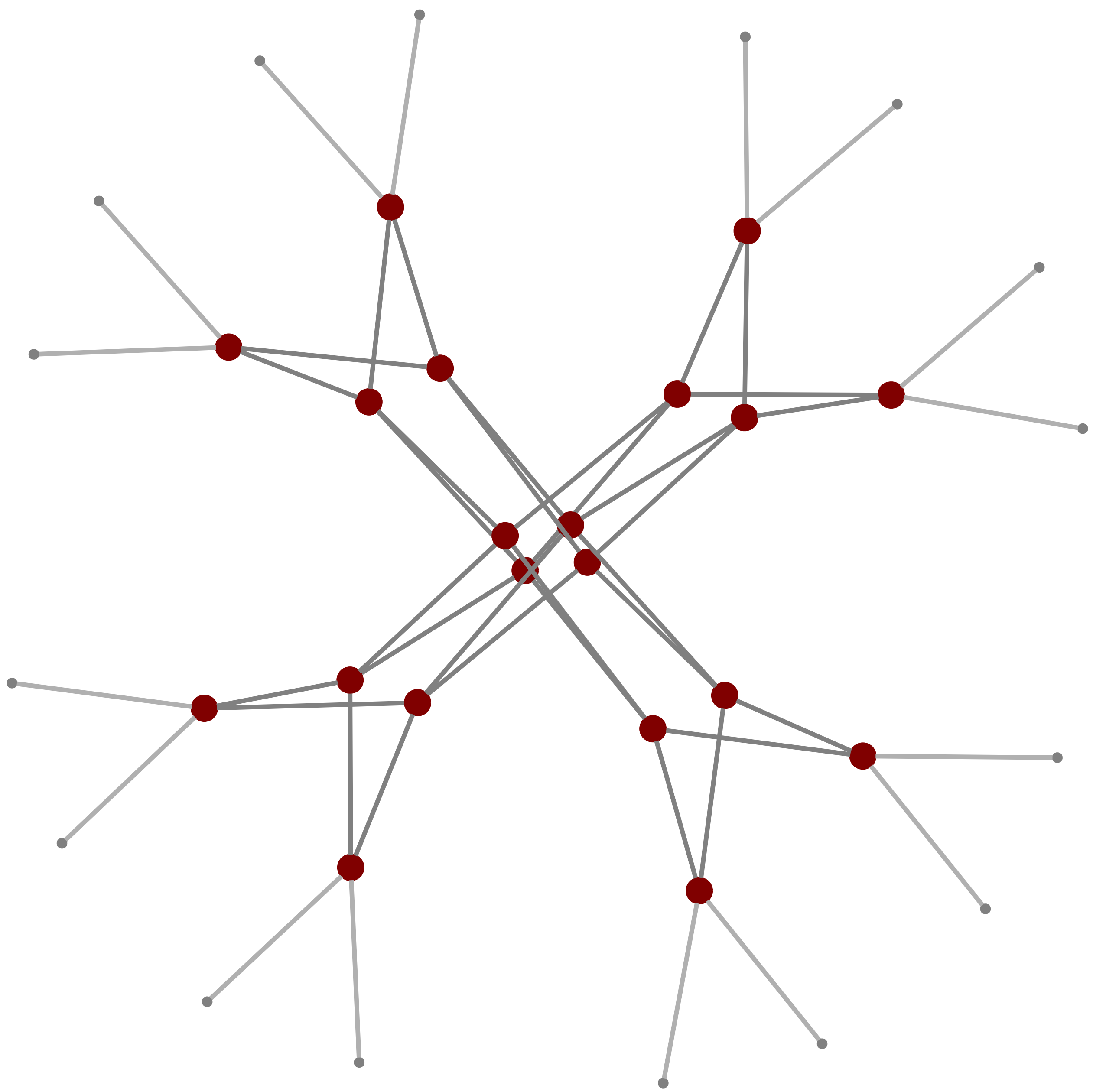
432 servers, 180 switches, degree 12



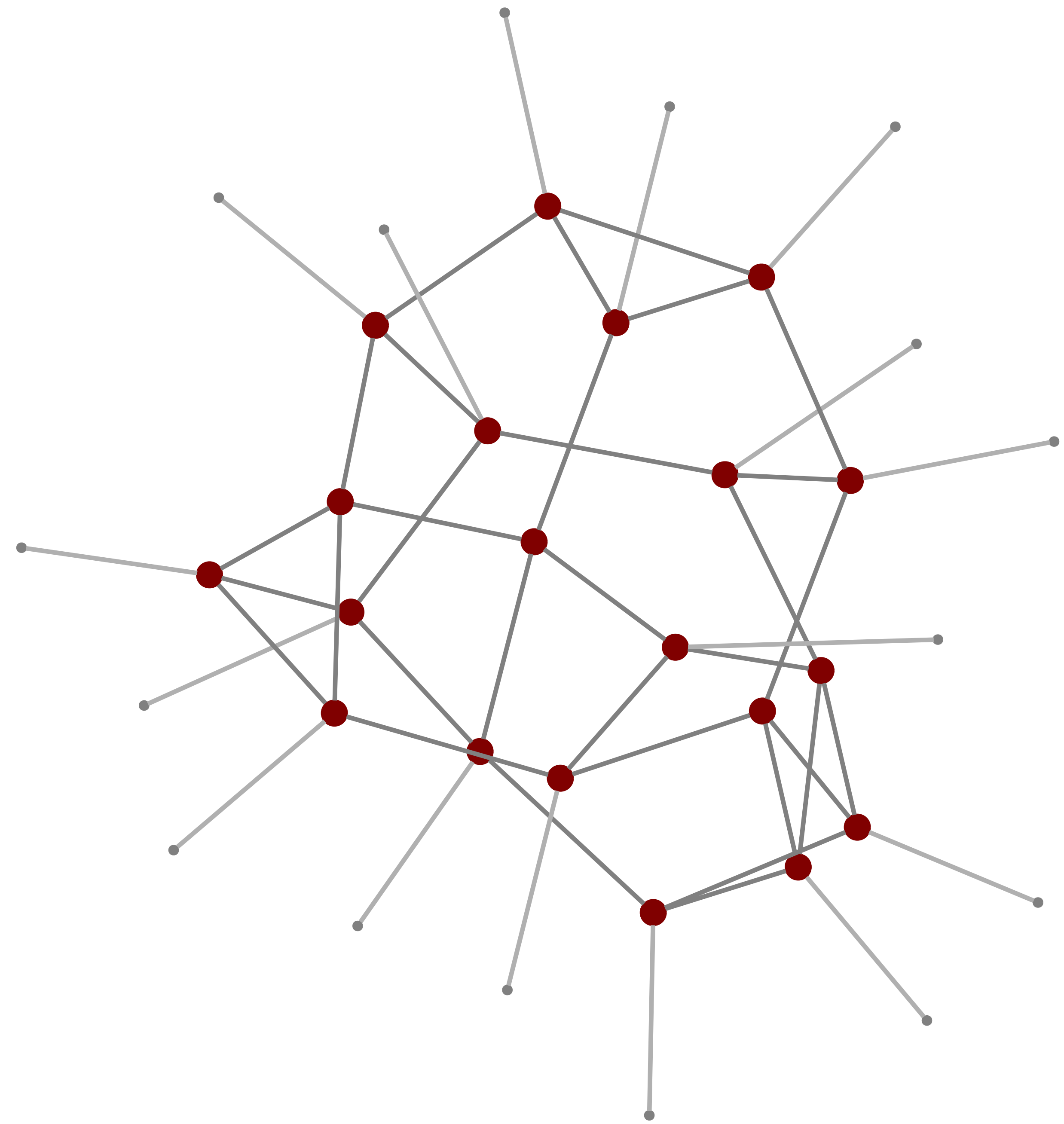
Jellyfish random graph

432 servers, 180 switches, degree 12

Example

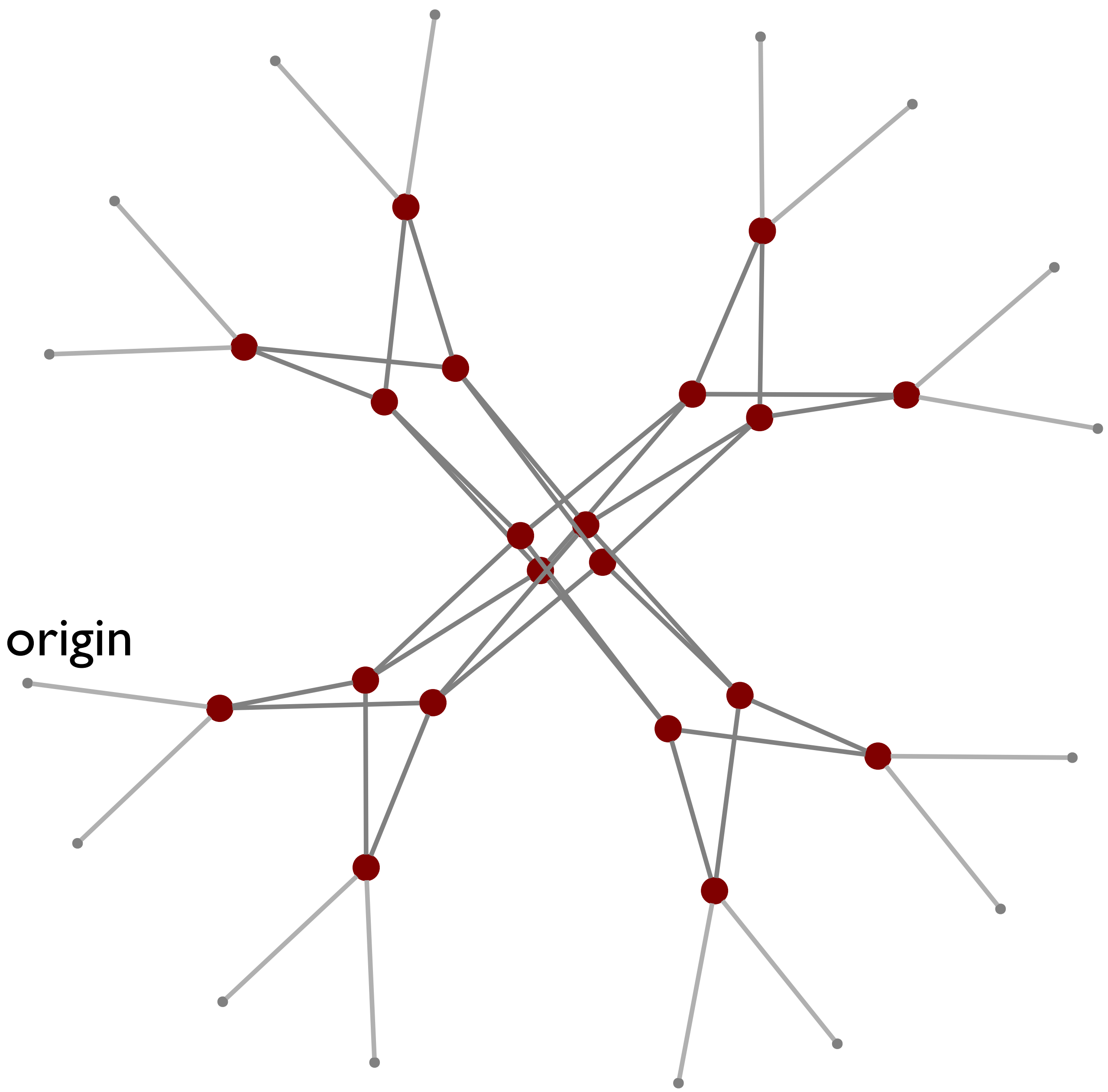


Fat tree
16 servers, 20 switches, degree 4



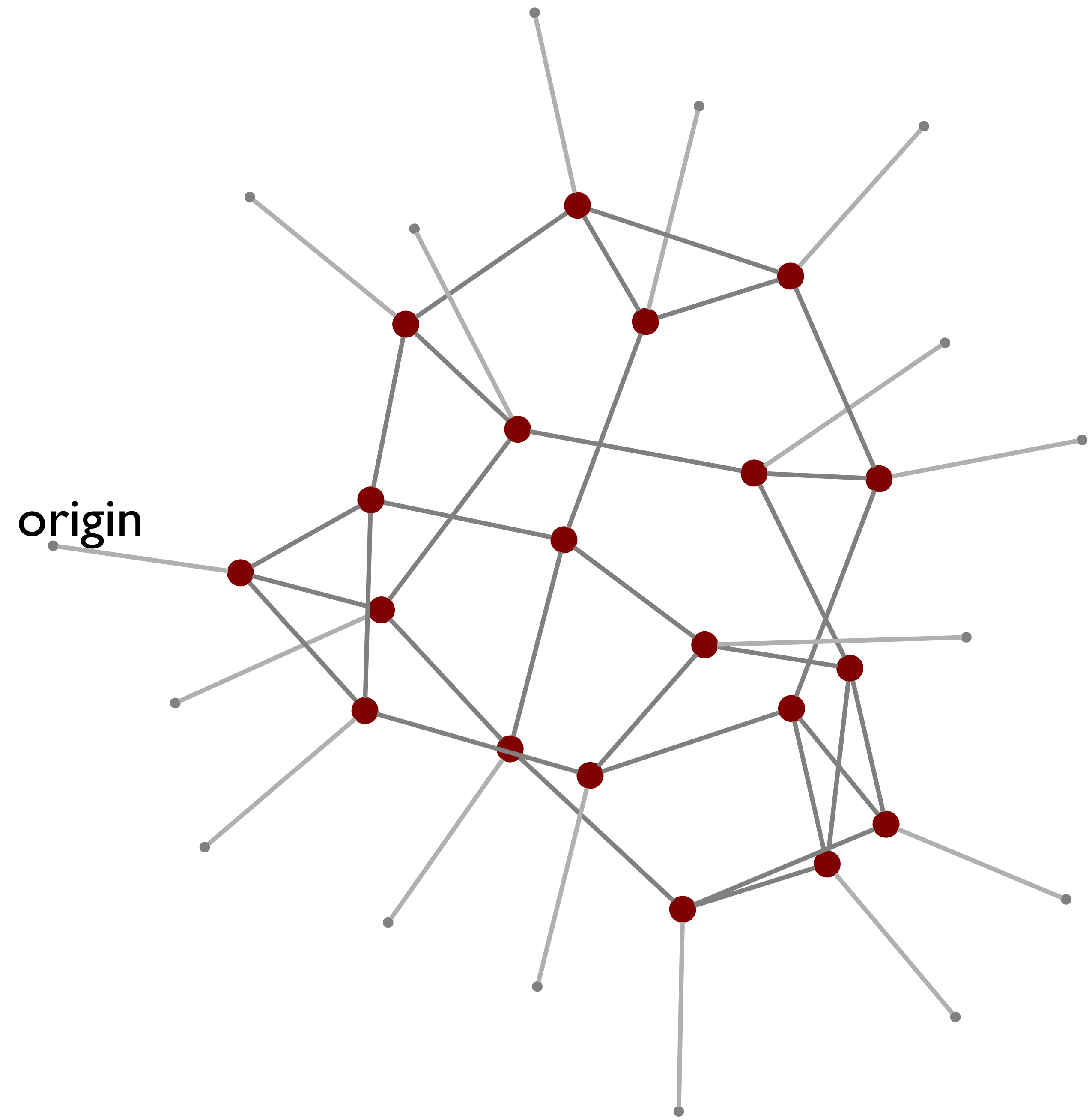
Jellyfish random graph
16 servers, 20 switches, degree 4

Example



Fat tree

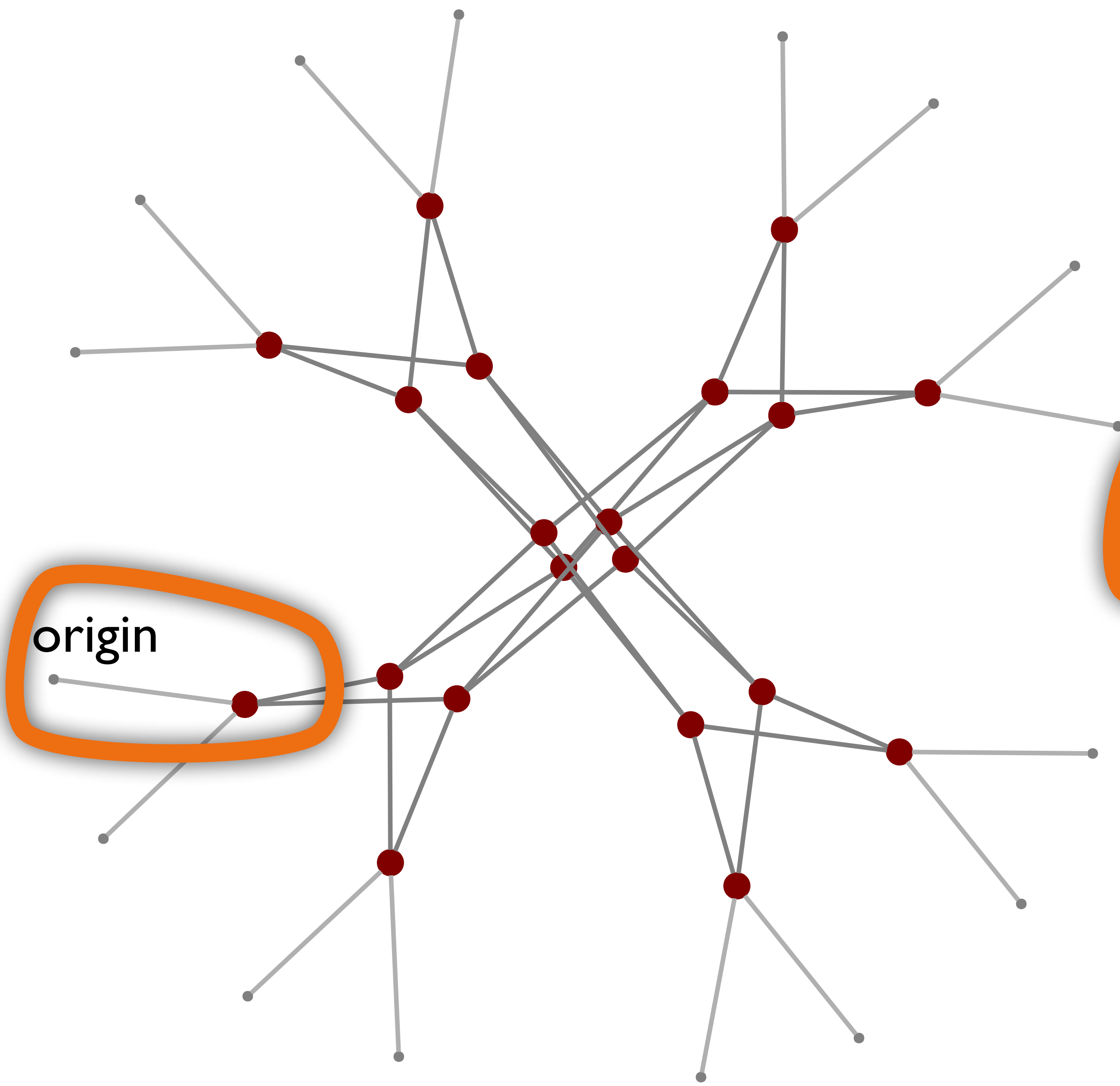
16 servers, 20 switches, degree 4



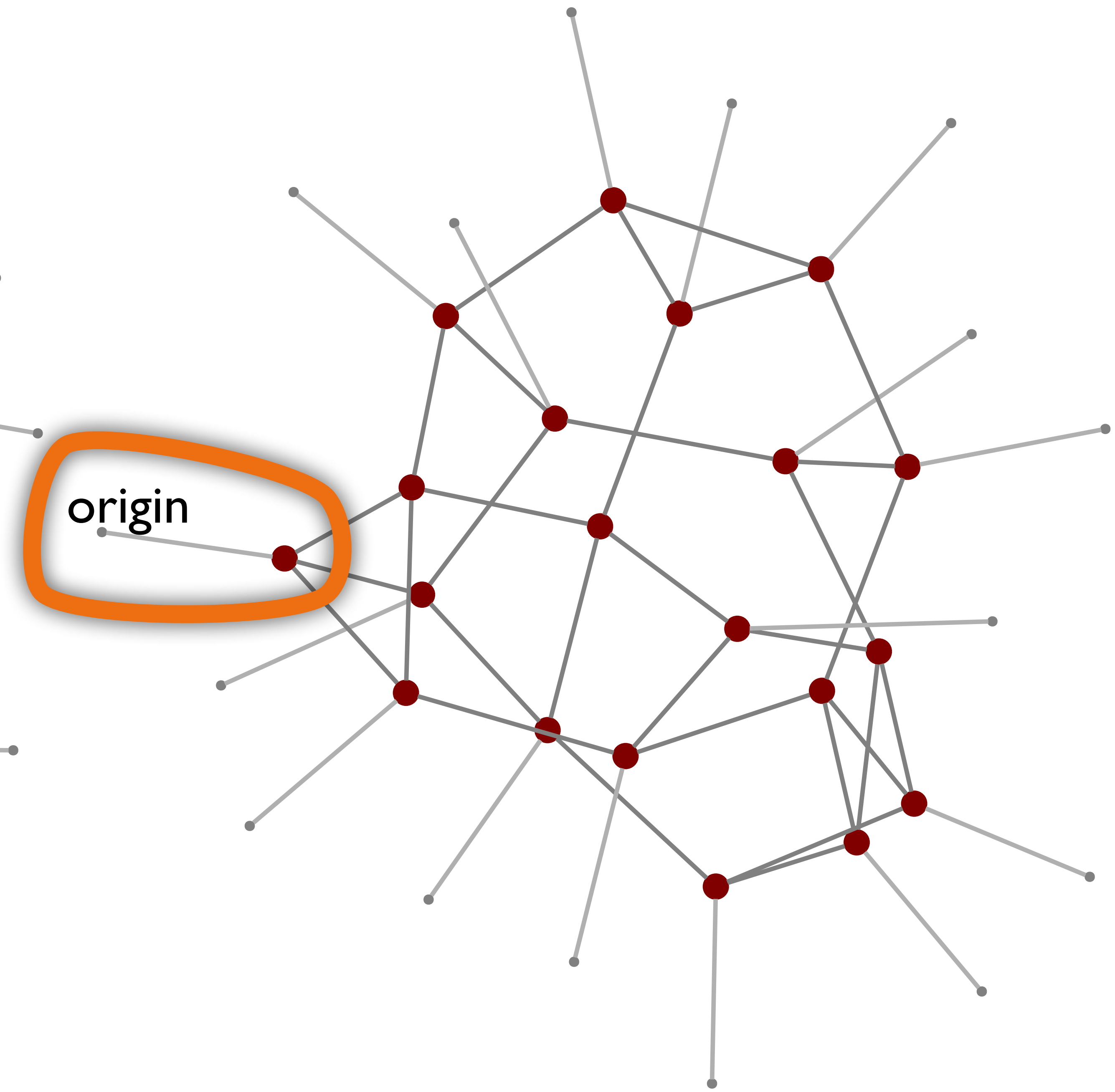
Jellyfish random graph

16 servers, 20 switches, degree 4

Example

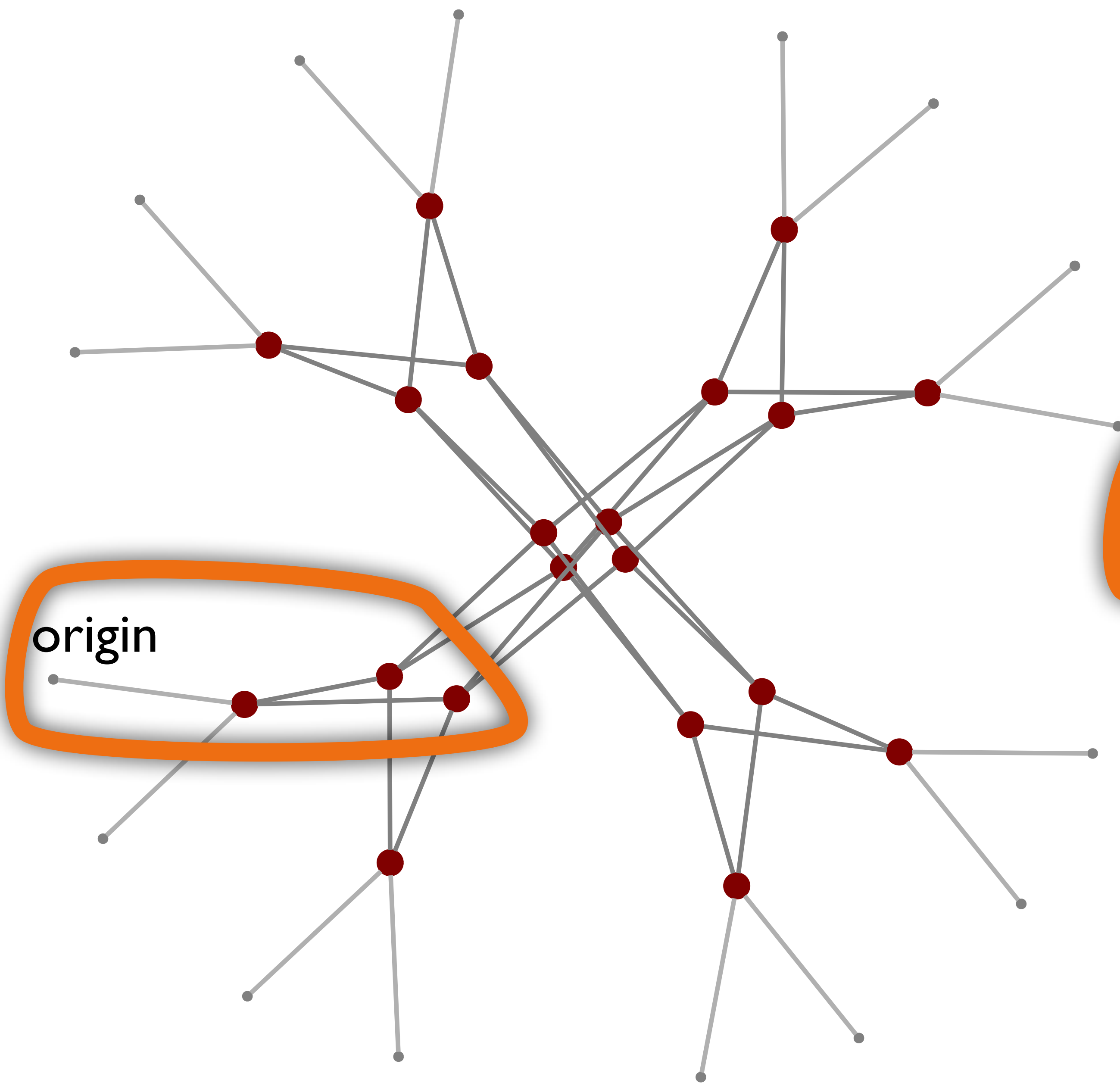


Fat tree
16 servers, 20 switches, degree 4



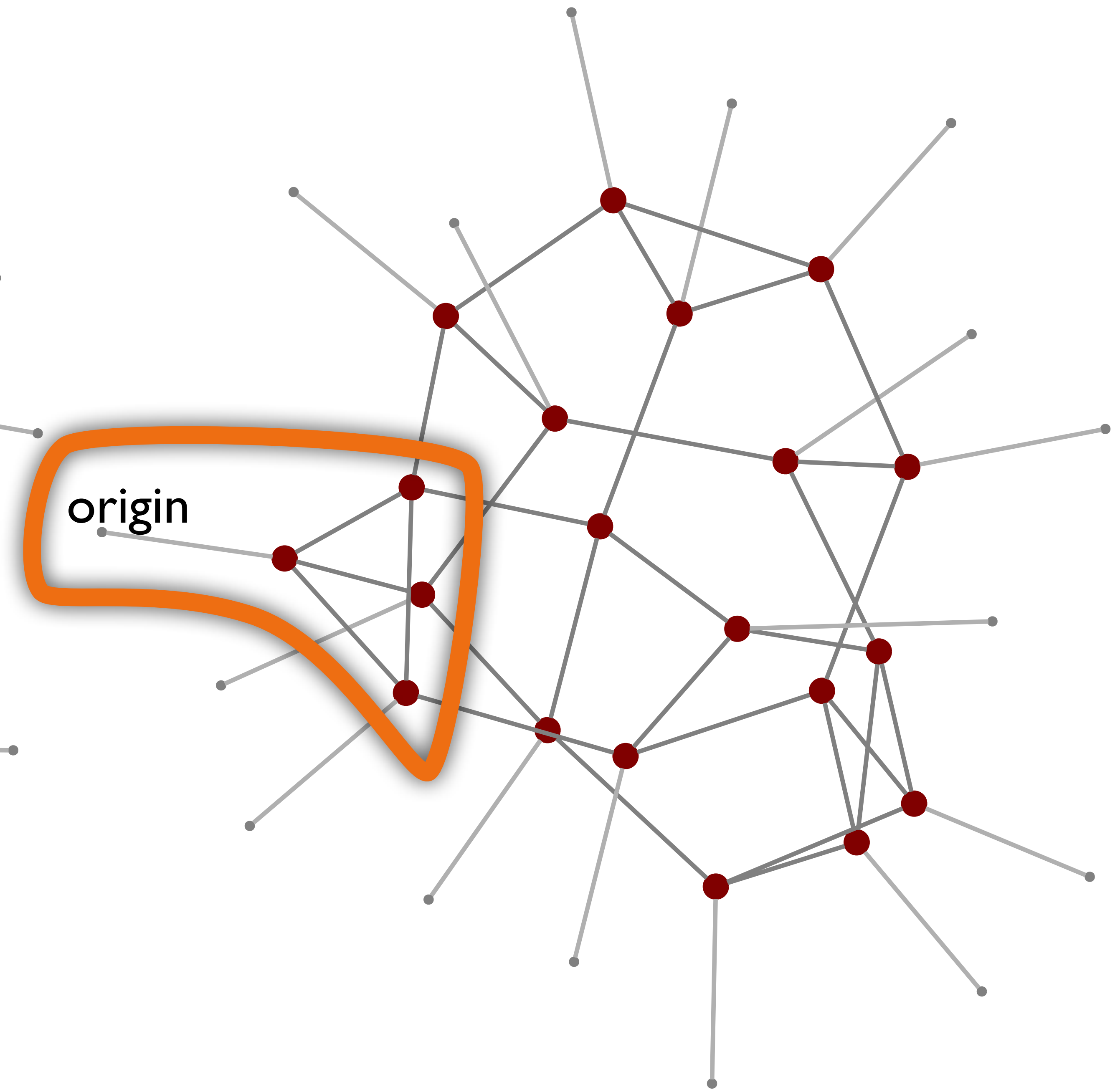
Jellyfish random graph
16 servers, 20 switches, degree 4

Example



Fat tree

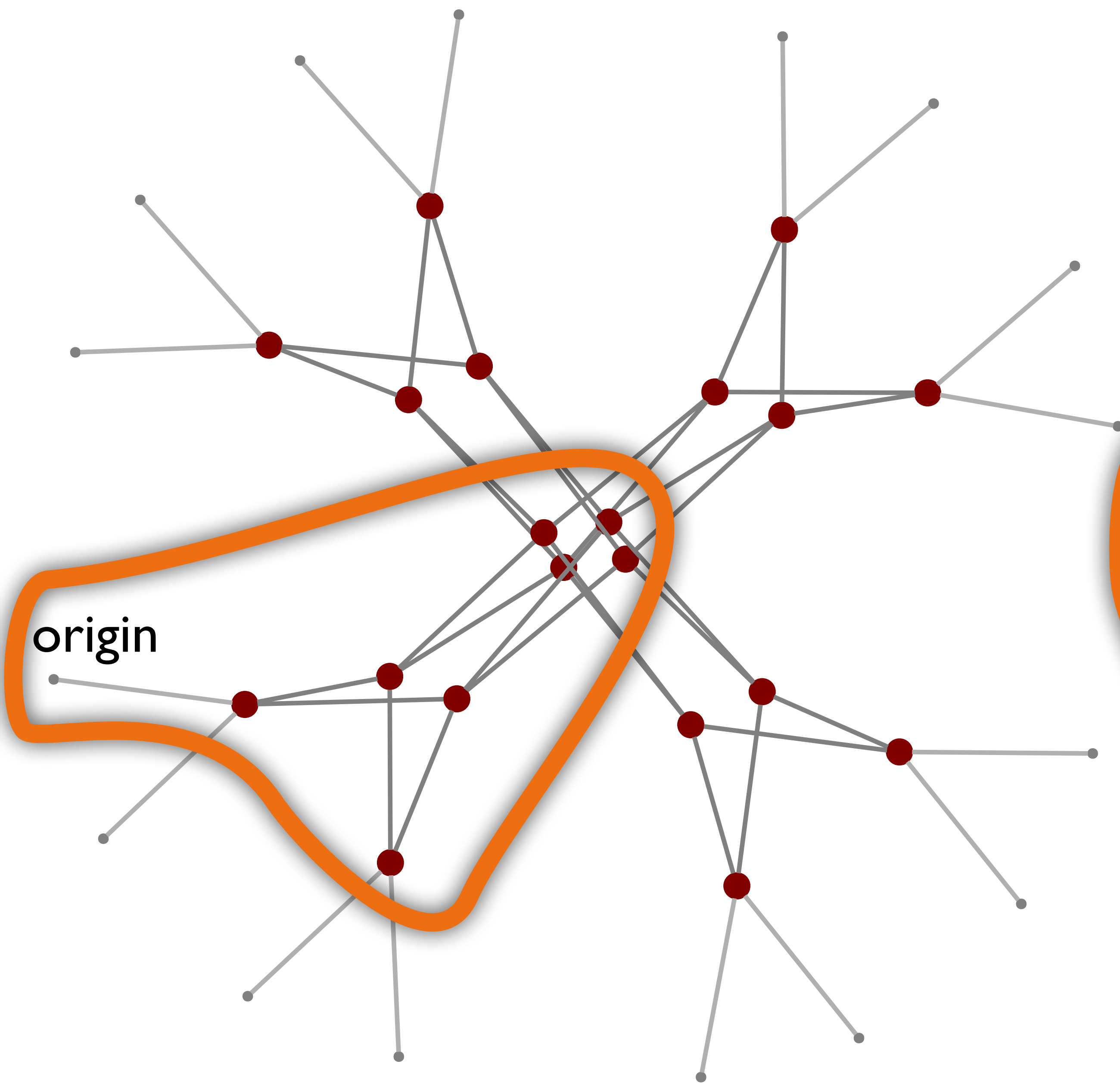
16 servers, 20 switches, degree 4



Jellyfish random graph

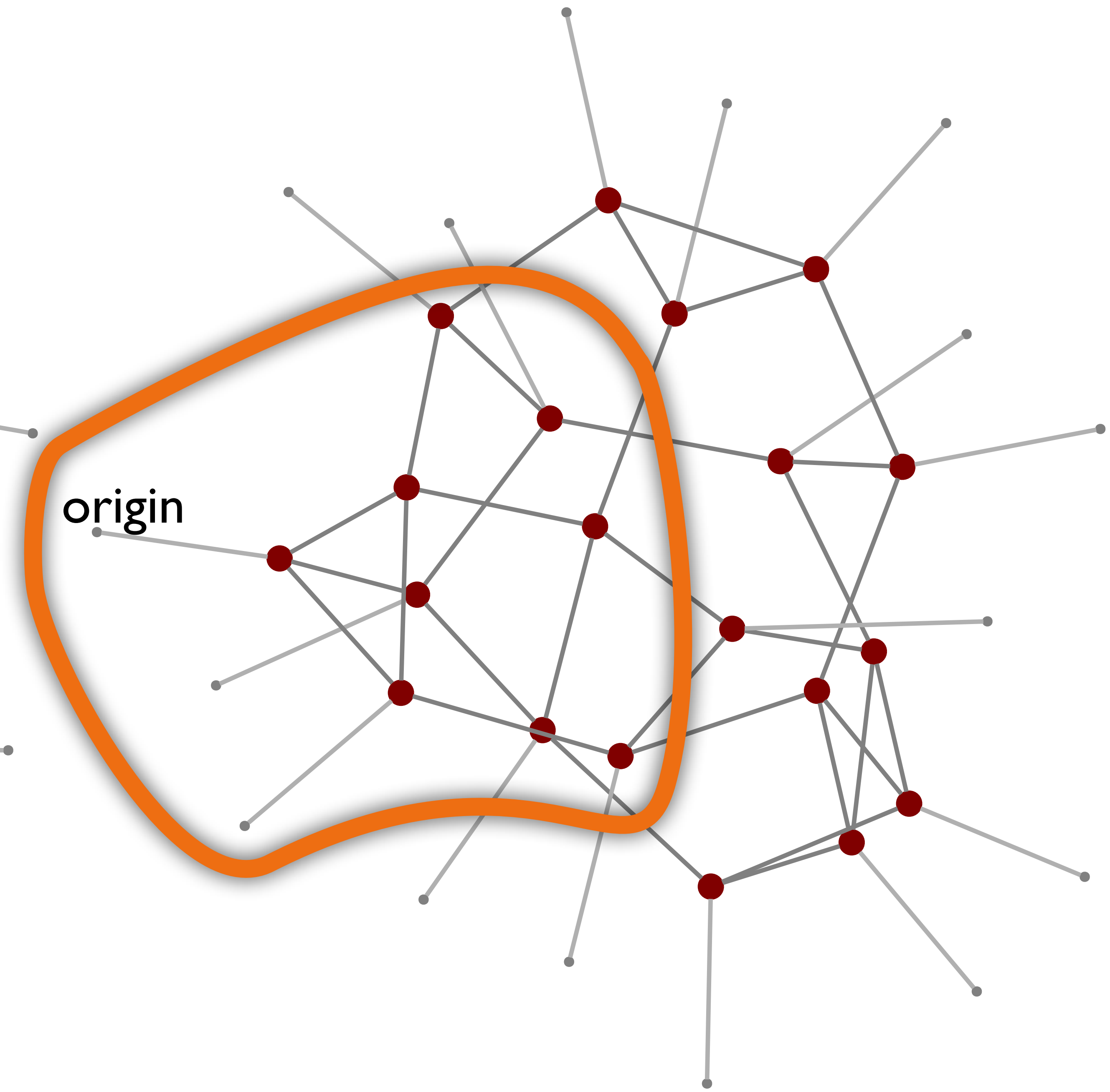
16 servers, 20 switches, degree 4

Example



Fat tree

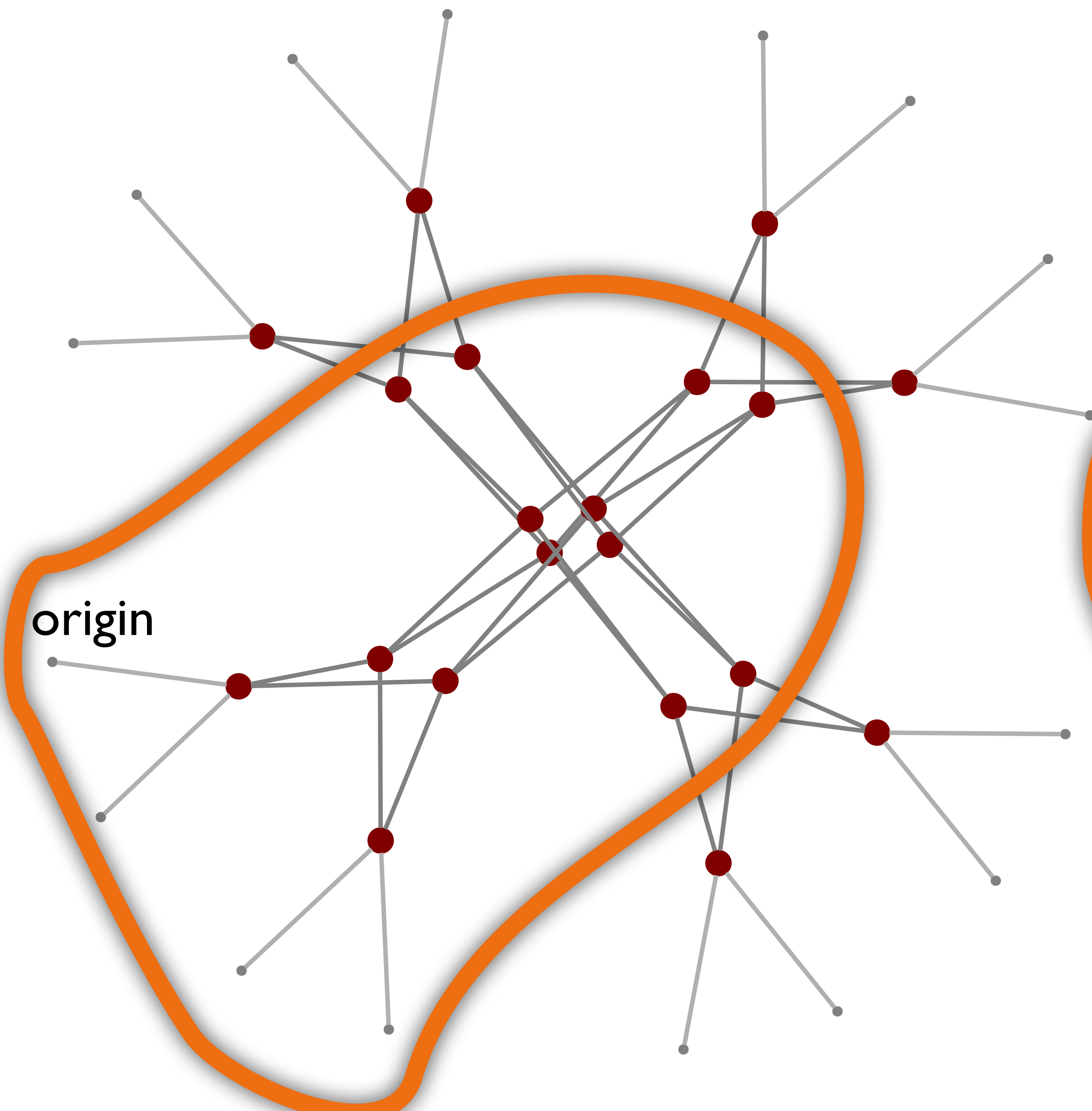
16 servers, 20 switches, degree 4



Jellyfish random graph

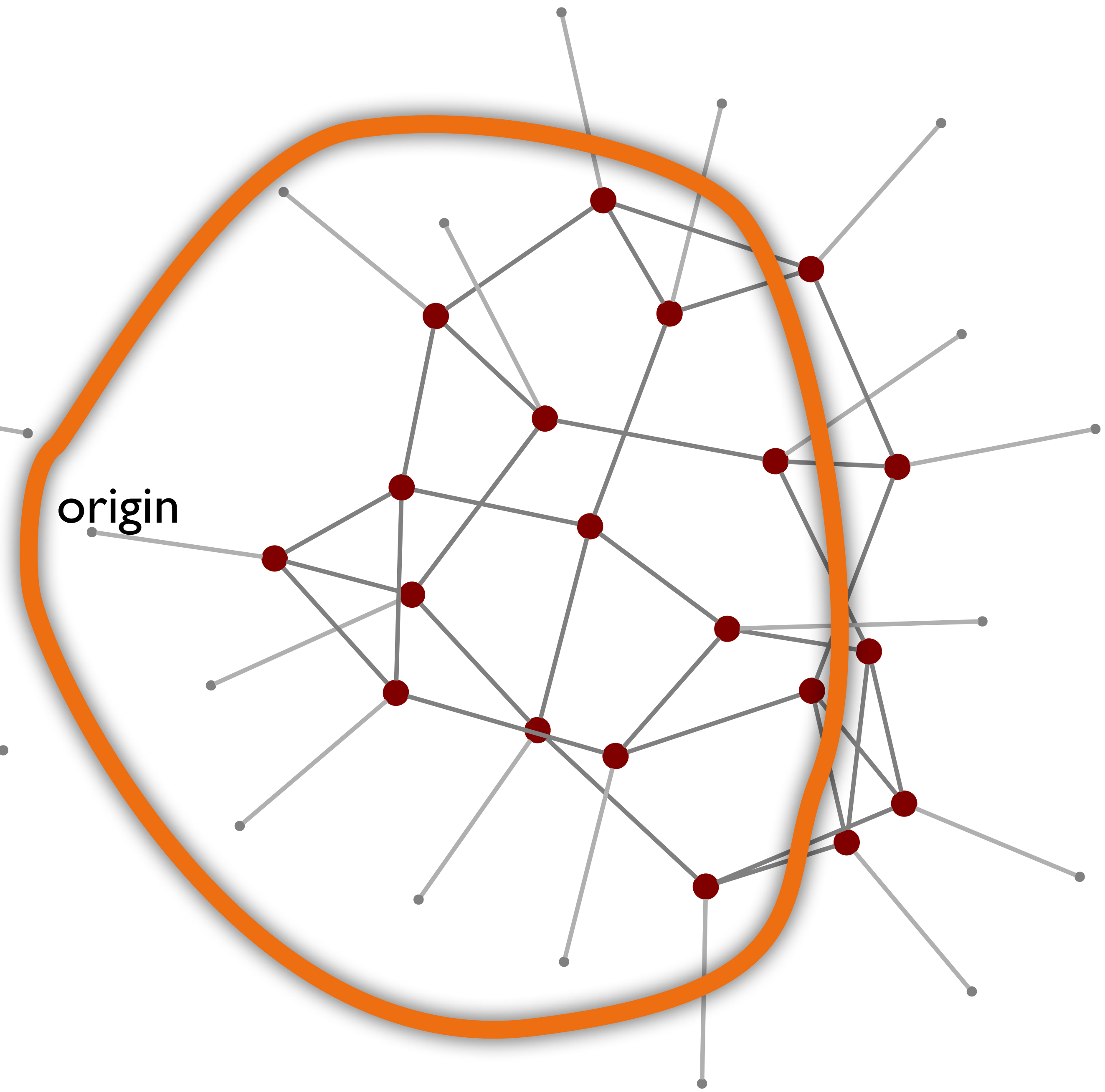
16 servers, 20 switches, degree 4

Example



Fat tree

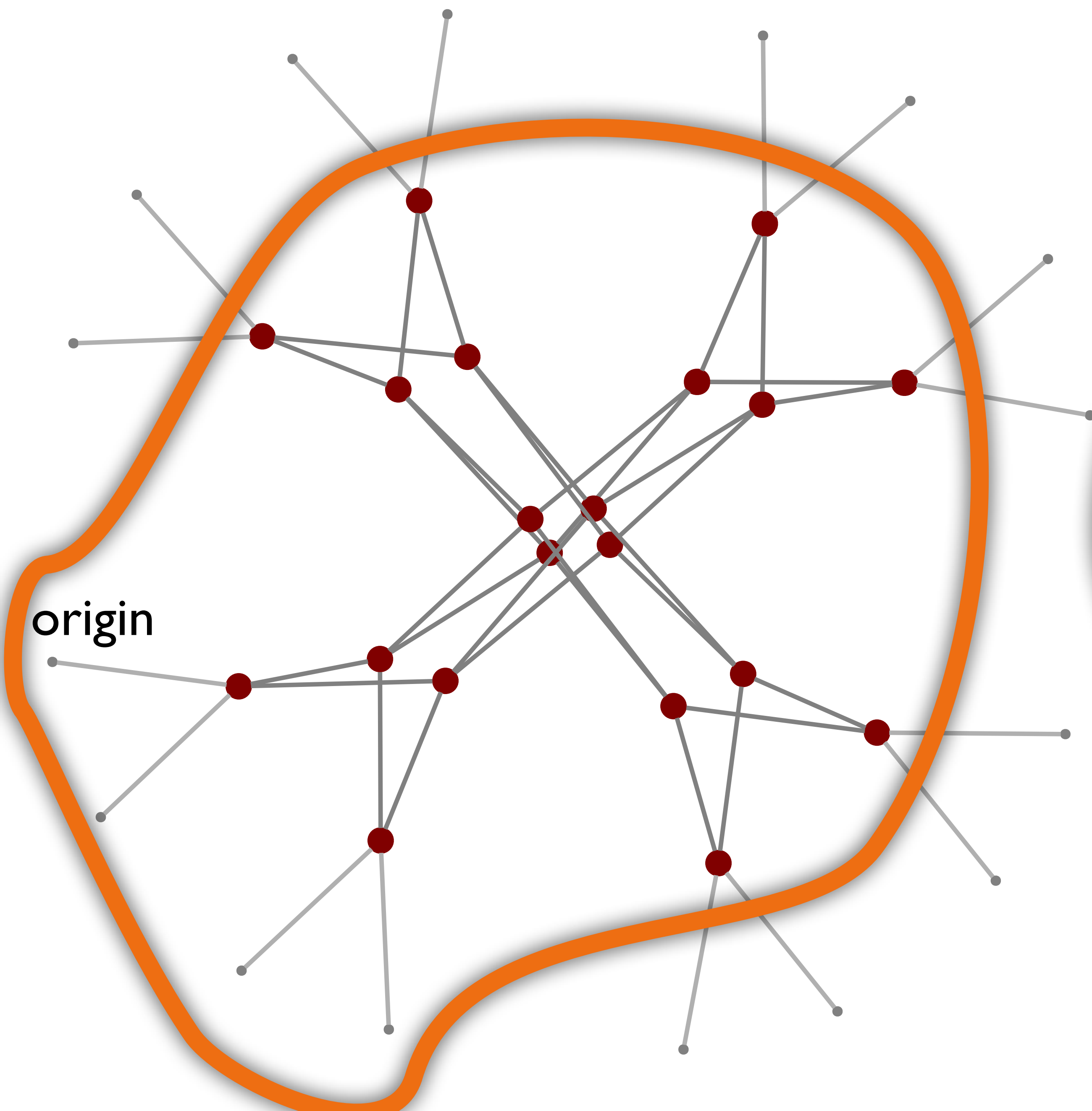
16 servers, 20 switches, degree 4



Jellyfish random graph

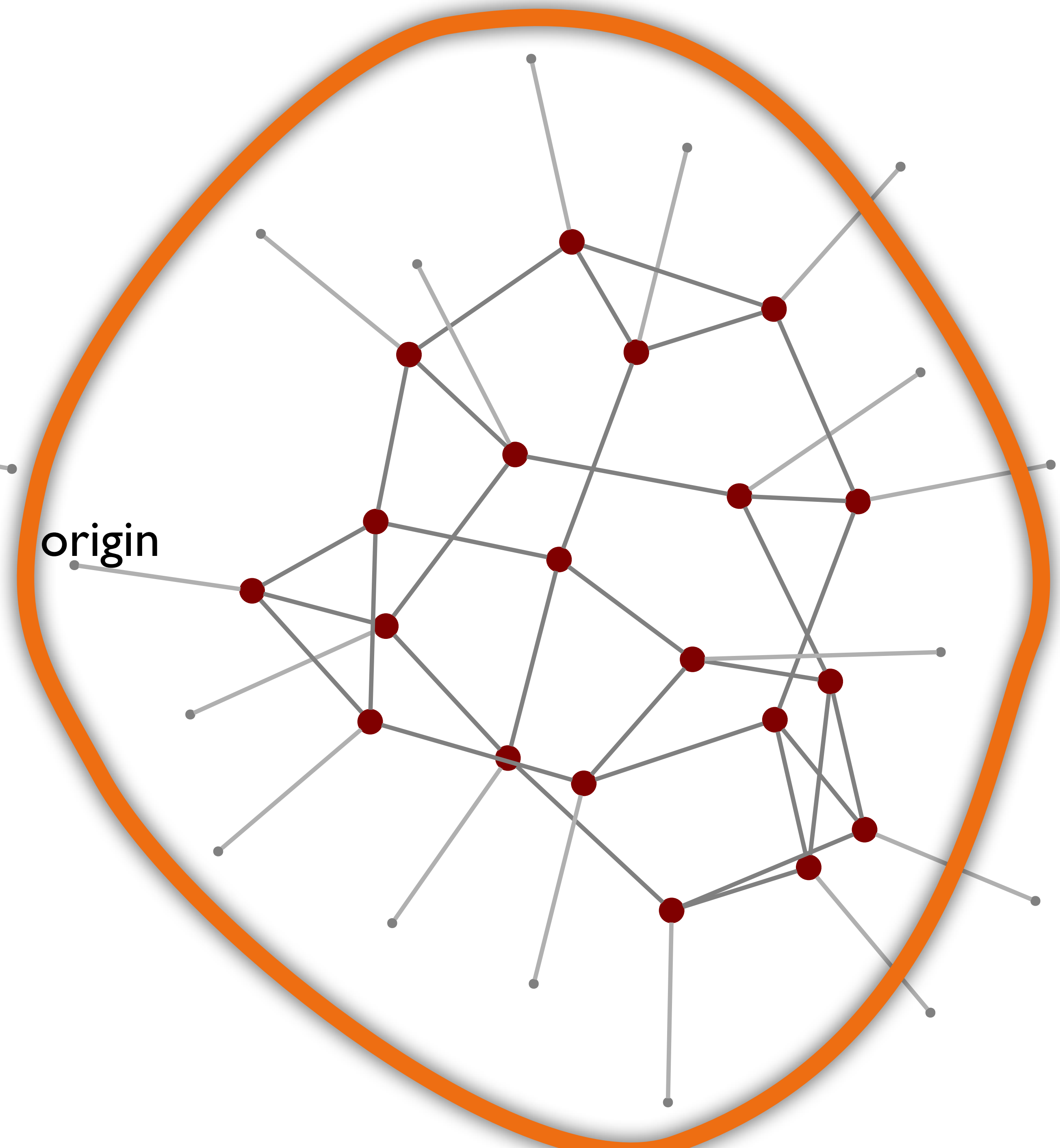
16 servers, 20 switches, degree 4

Example



Fat tree

16 servers, 20 switches, degree 4



Jellyfish random graph

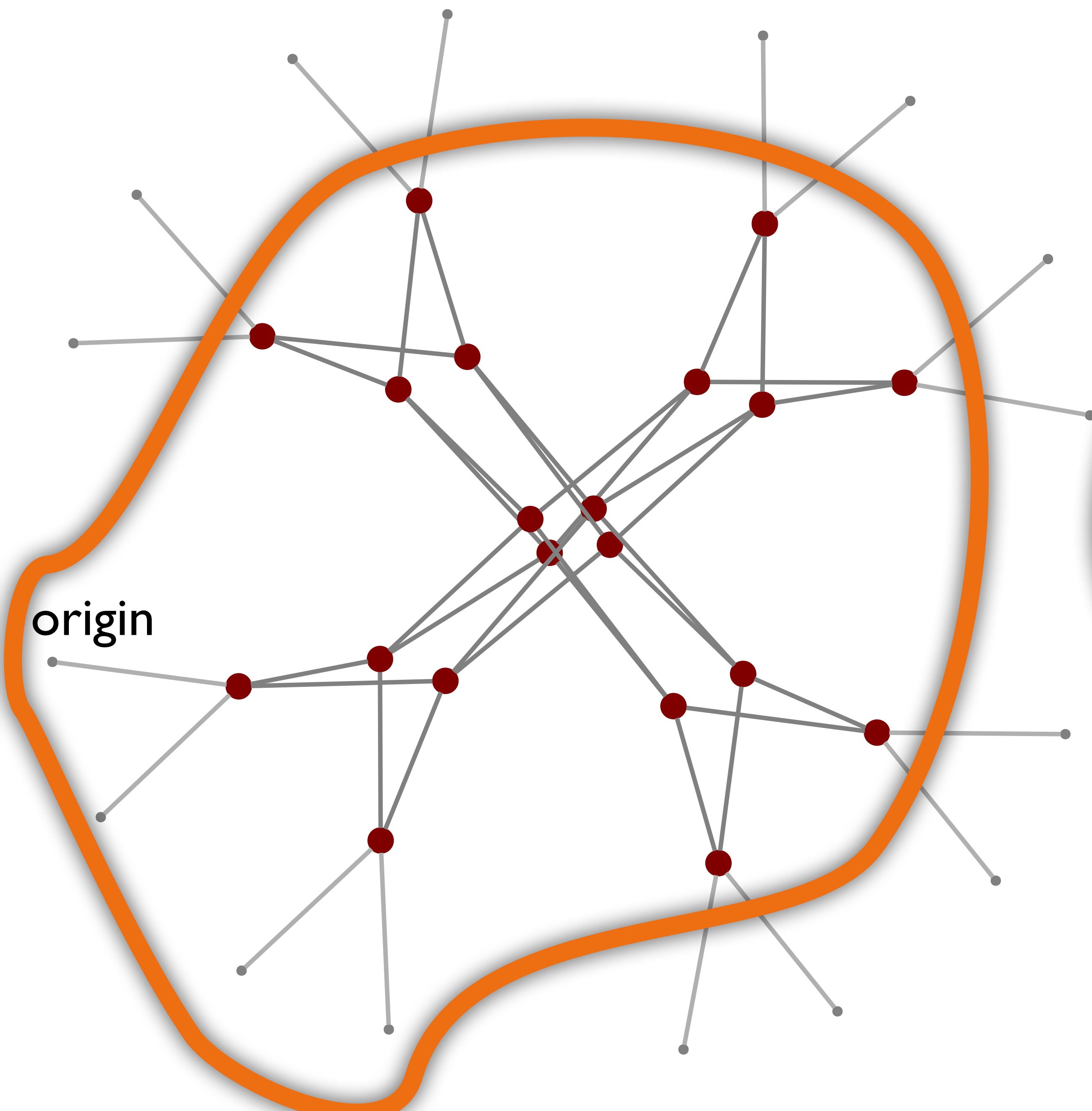
16 servers, 20 switches, degree 4

Example

Fat tree
16 servers, 20 switches, degree 4

Jellyfish random graph
16 servers, 20 switches, degree 4

Example



Fat tree

16 servers, 20 switches, degree 4

12 of 16

reachable in

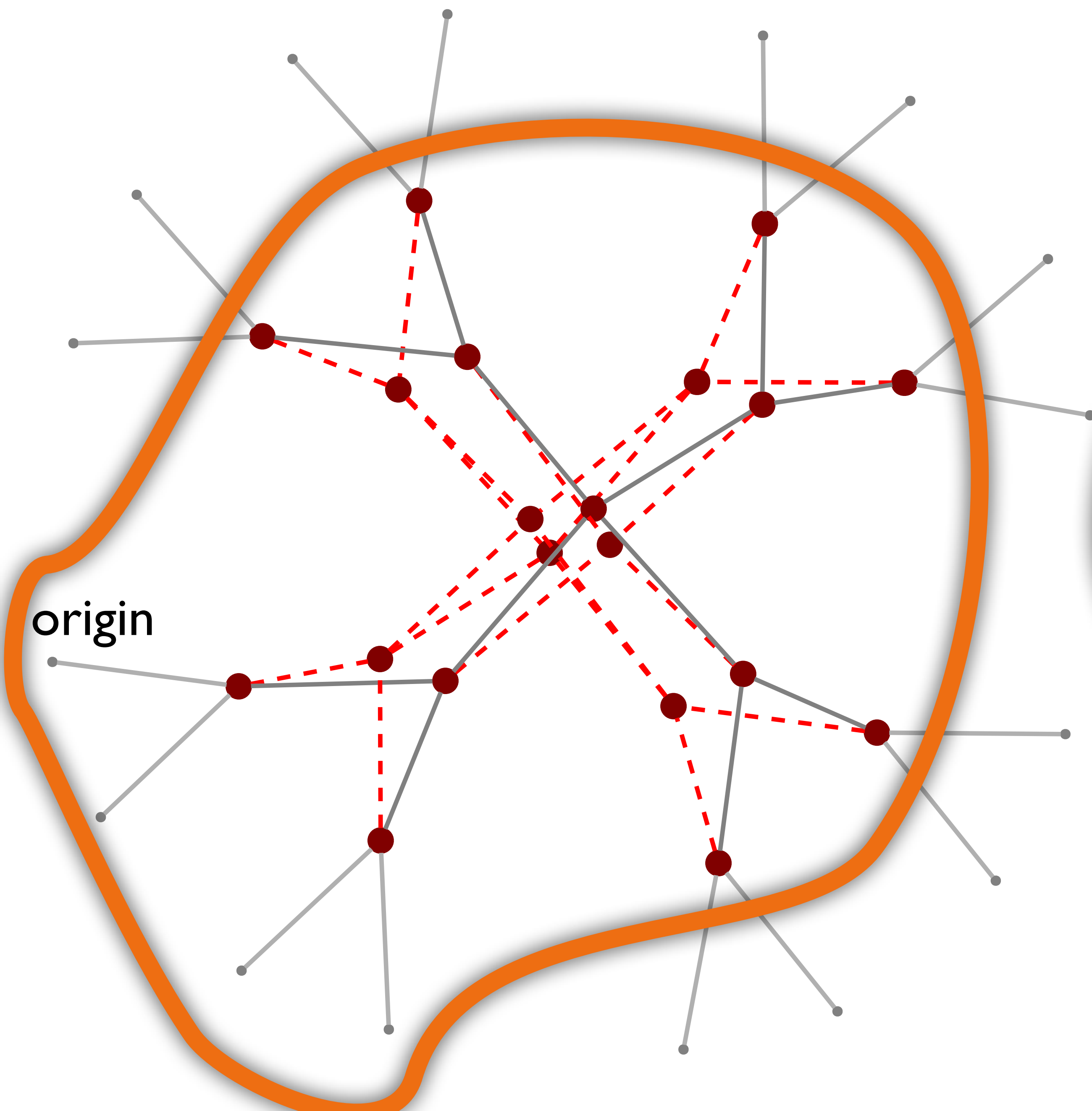
≤ 5 hops

(good expander)

Jellyfish random graph

16 servers, 20 switches, degree 4

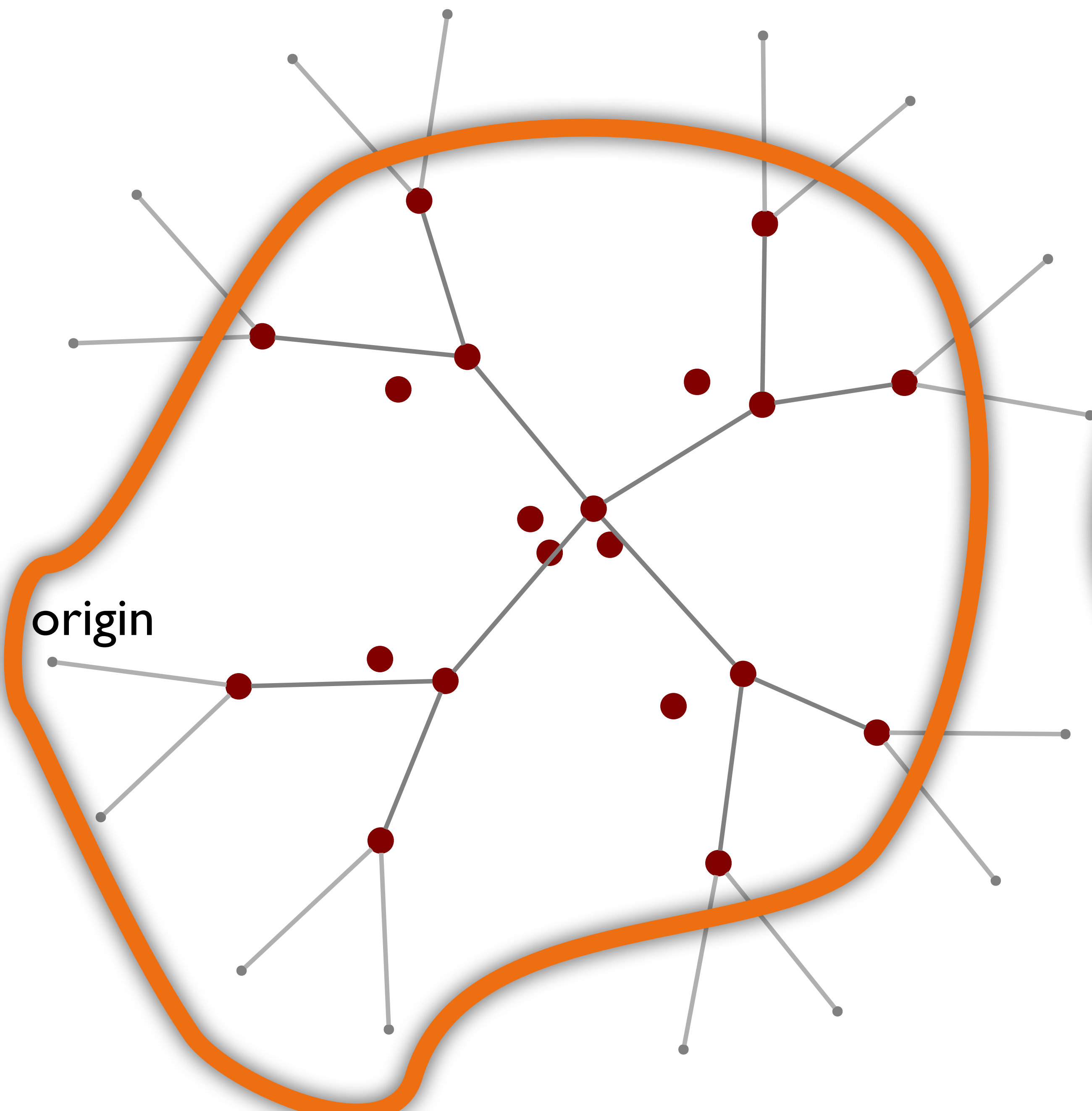
Example



Fat tree
16 servers, 20 switches, degree 4

Jellyfish random graph
16 servers, 20 switches, degree 4

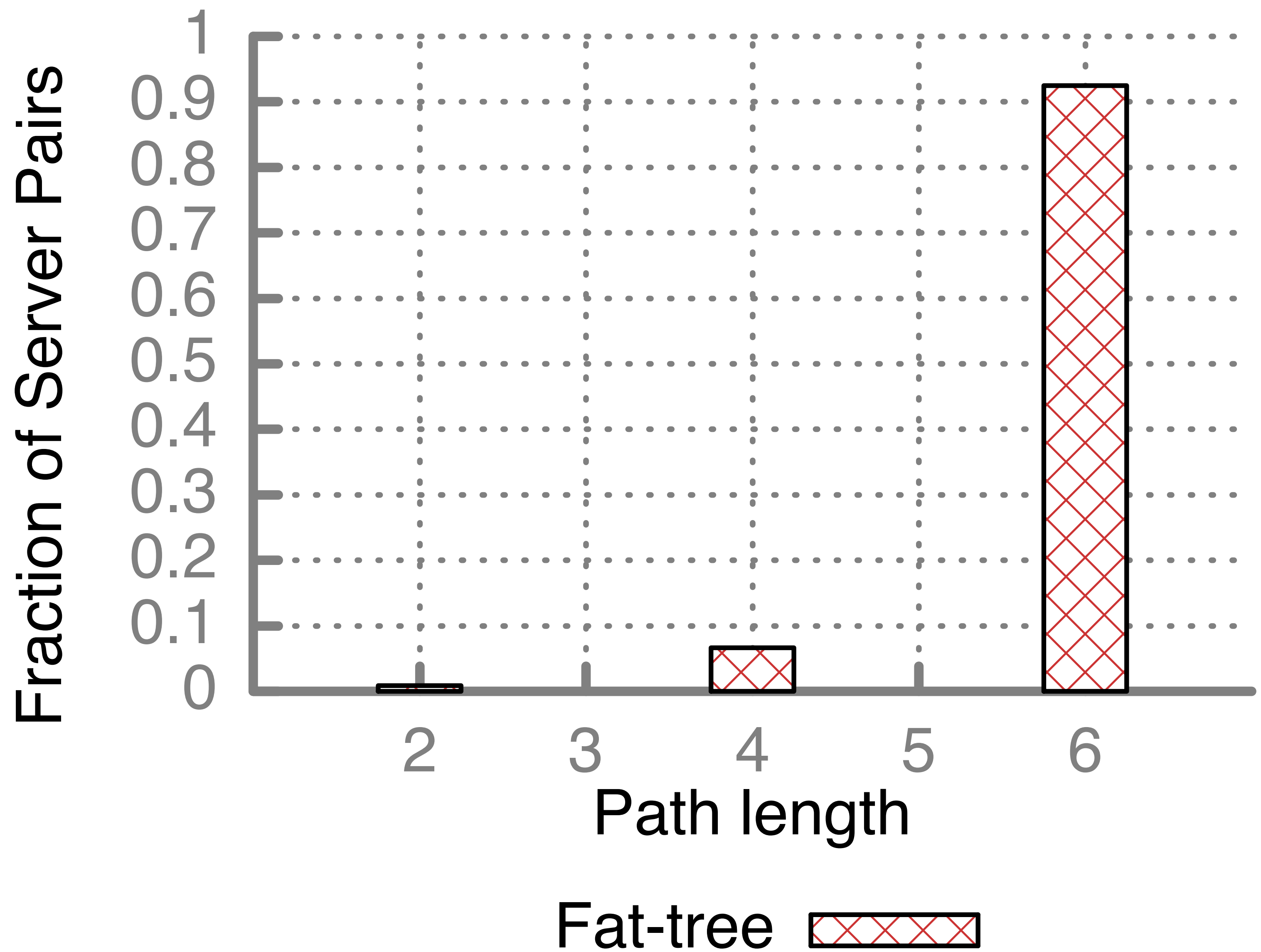
Example



Fat tree
16 servers, 20 switches, degree 4

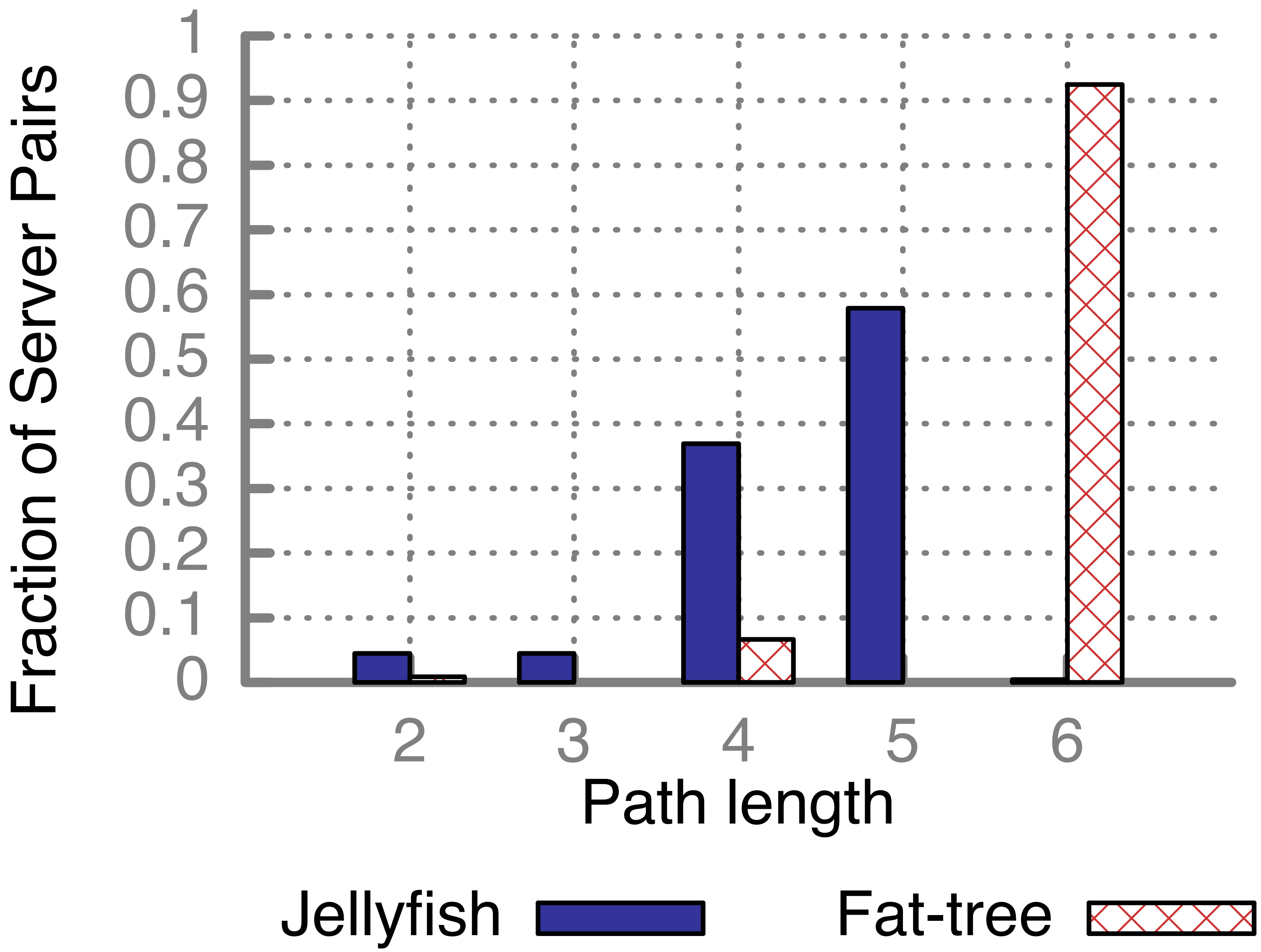
Jellyfish random graph
16 servers, 20 switches, degree 4

Jellyfish has short paths



Fat-tree with 686 servers

Jellyfish has short paths



Jellyfish, same equipment

System Design:

Performance Consistency

Is performance more variable?

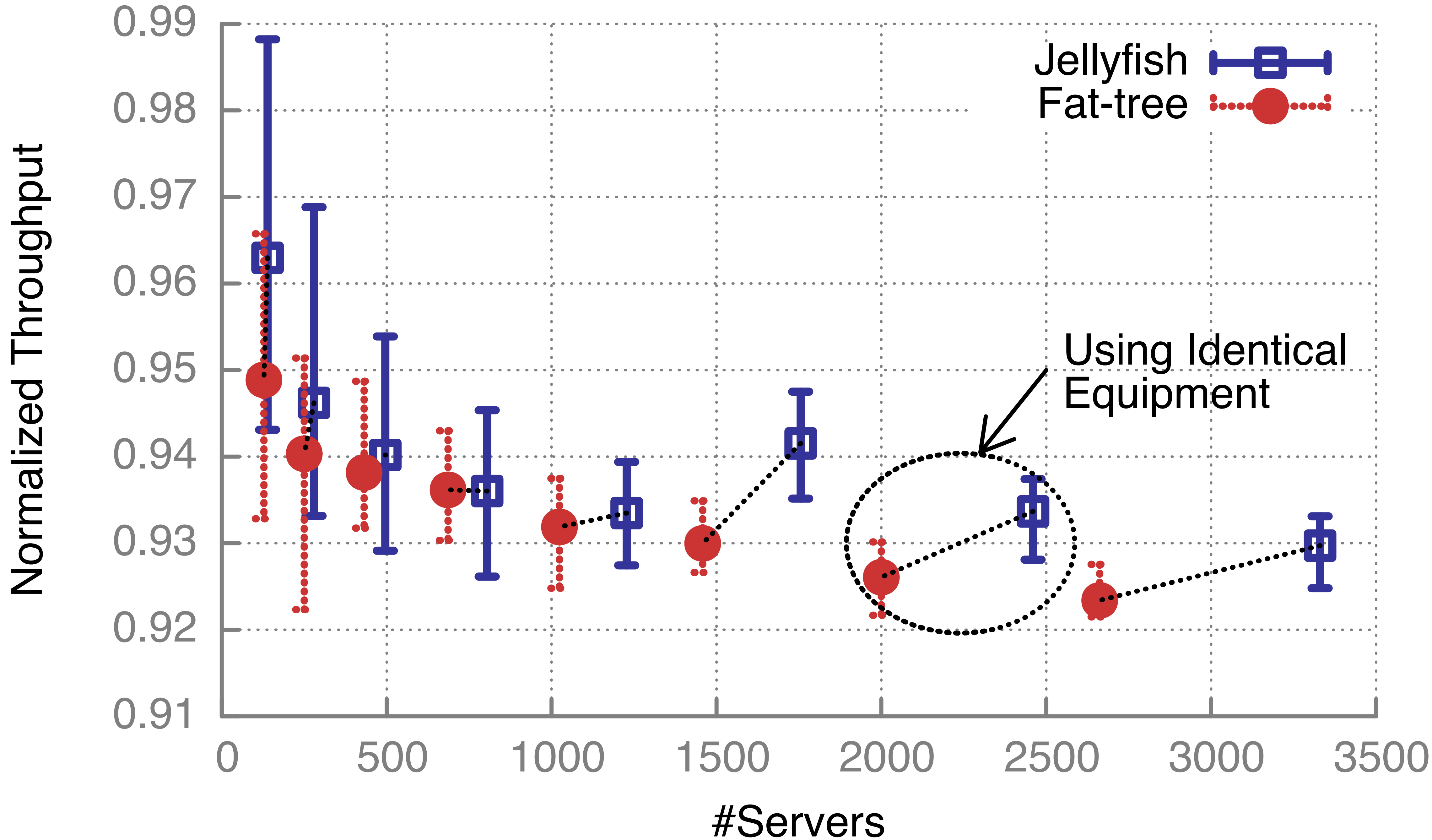
Performance depends on choice of random graph

- if you expand the network, would performance change dramatically?

Extreme case: graph could be disconnected!

- never happens, with high probability

Little variation if size is moderate



{min, avg, max} of 20 trials shown

System Design:

Routing

Routing

Intuition

if we fully utilize all available capacity ...

$$\#\ 1\ \text{Gbps flows} = \frac{\text{total capacity}}{\text{used capacity per flow}}$$

How do we effectively utilize capacity without structure?

Routing without structure

In theory, just a multicommodity flow (MCF) problem

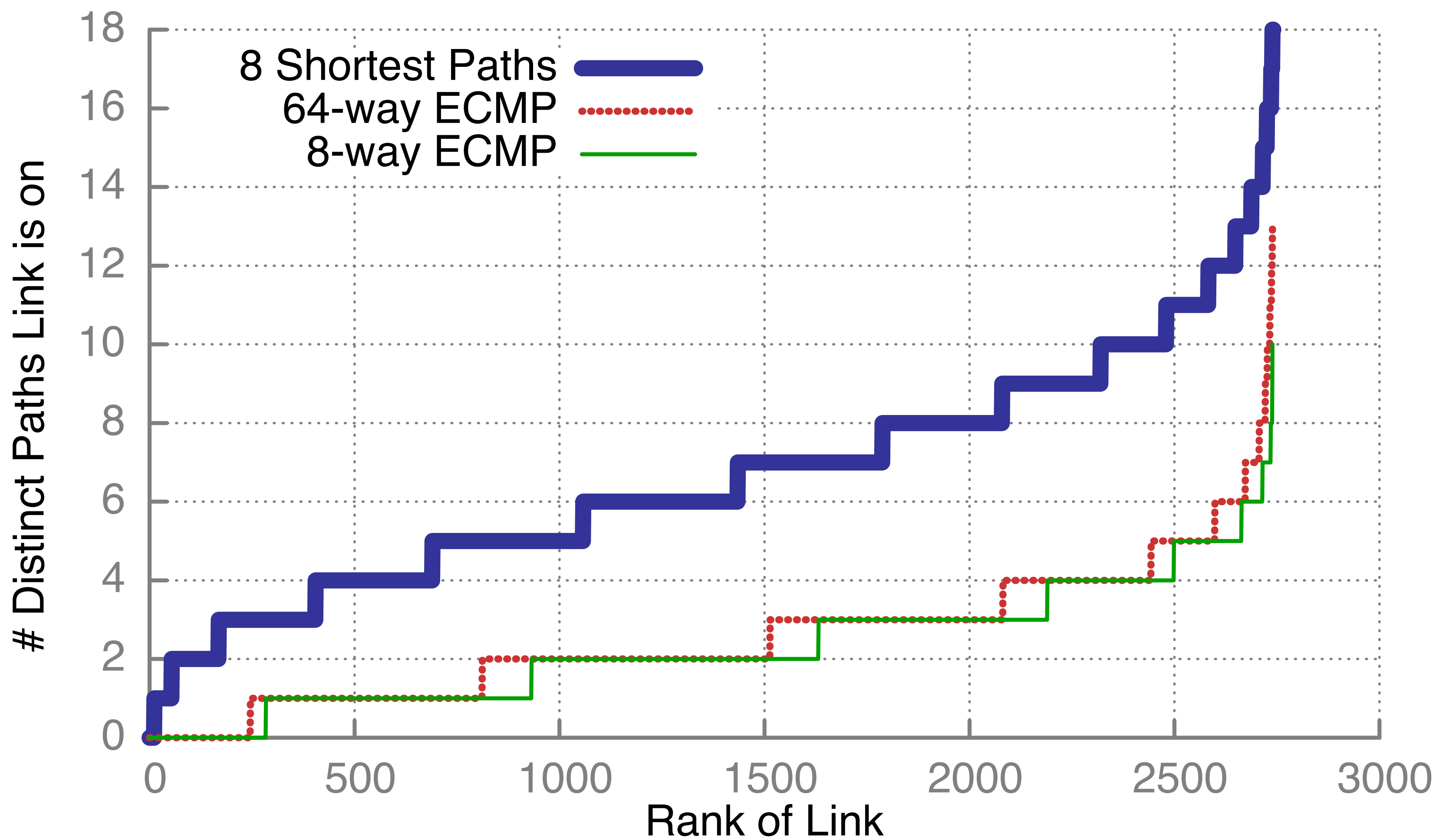
Potential issues:

- Solve MCF using a distributed protocol?
- Optimal solution could have too many small subflows

Routing

Does ECMP work?

- No
- ECMP doesn't use Jellyfish's path diversity

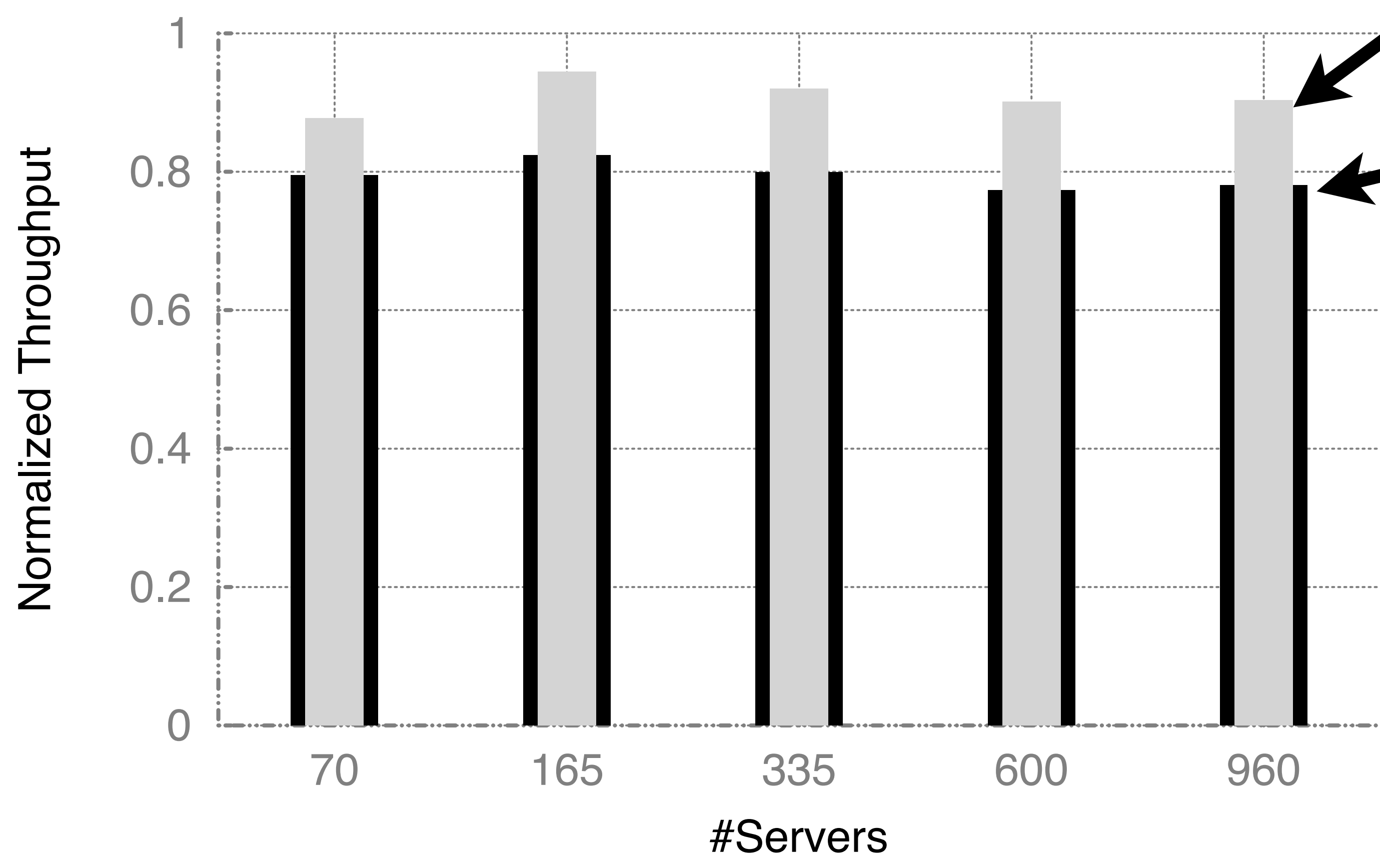


Routing: a simple solution

Find *k* shortest paths

Let Multipath TCP do the rest

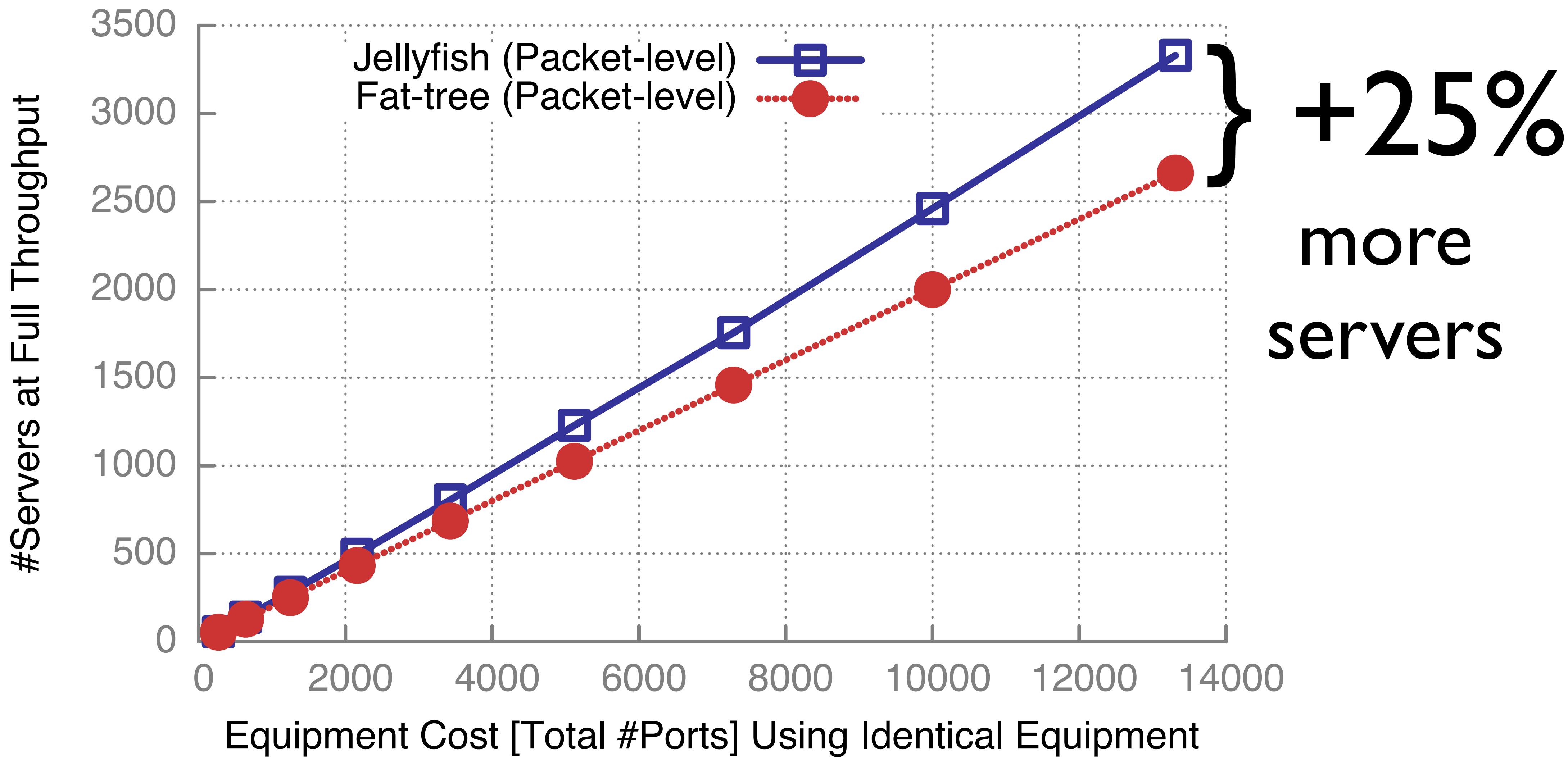
- [Wischik, Raiciu, Greenhalgh, Handley, NSDI'10]



86-90% of optimal

(TCP is within 3 percentage points of MPTCP)

Throughput: Jellyfish vs. fat tree



8-shortest paths + MPTCP

Deploying k-shortest paths

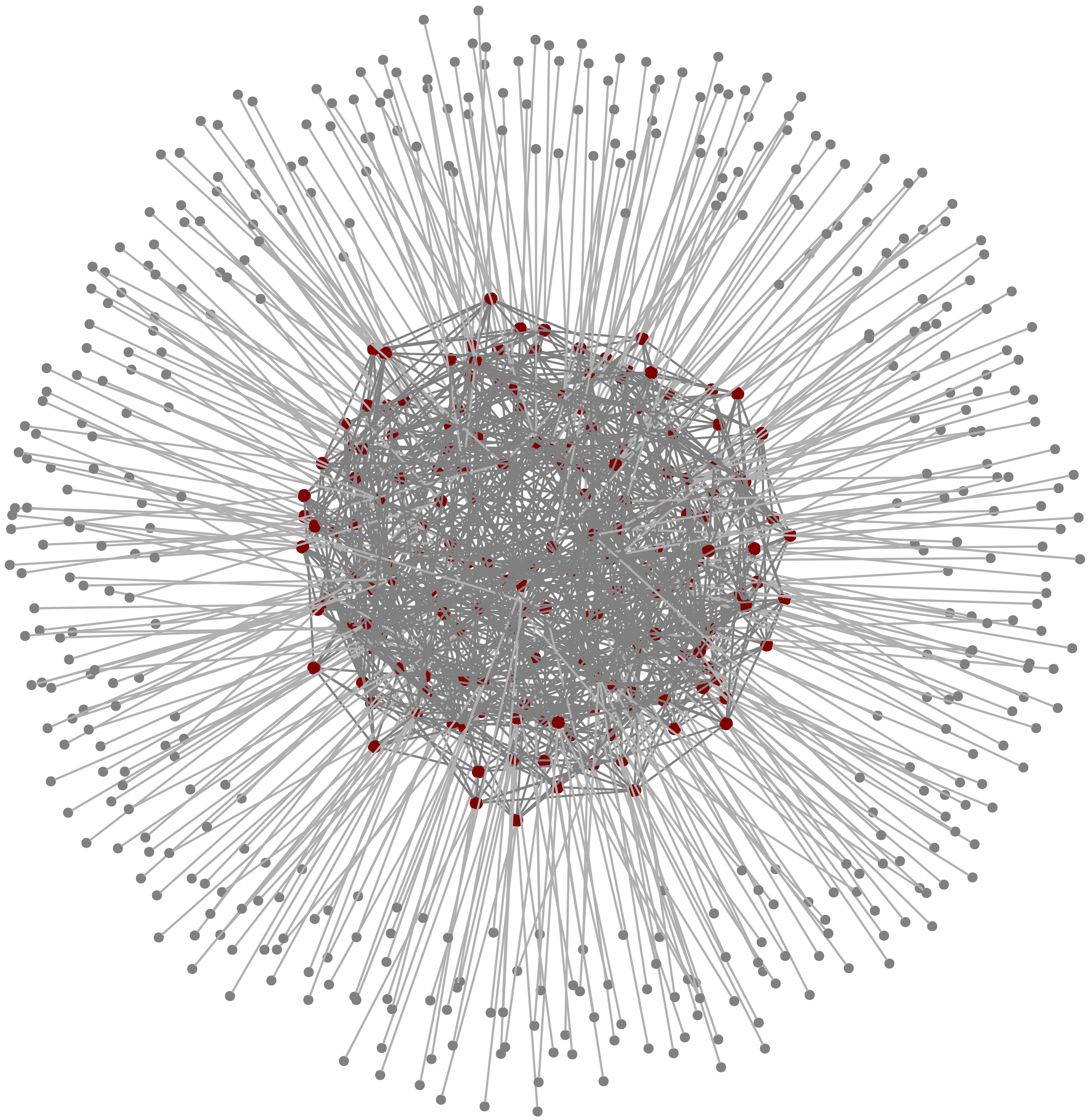
Multiple options:

- SPAIN [Mudigonda, Yalagandula, Al-Fares, Mogul, NSDI' 10]
- Equal-cost MPLS tunnels
- IBM Research's SPARTA [CoNEXT 2012]
- SDN controller based methods

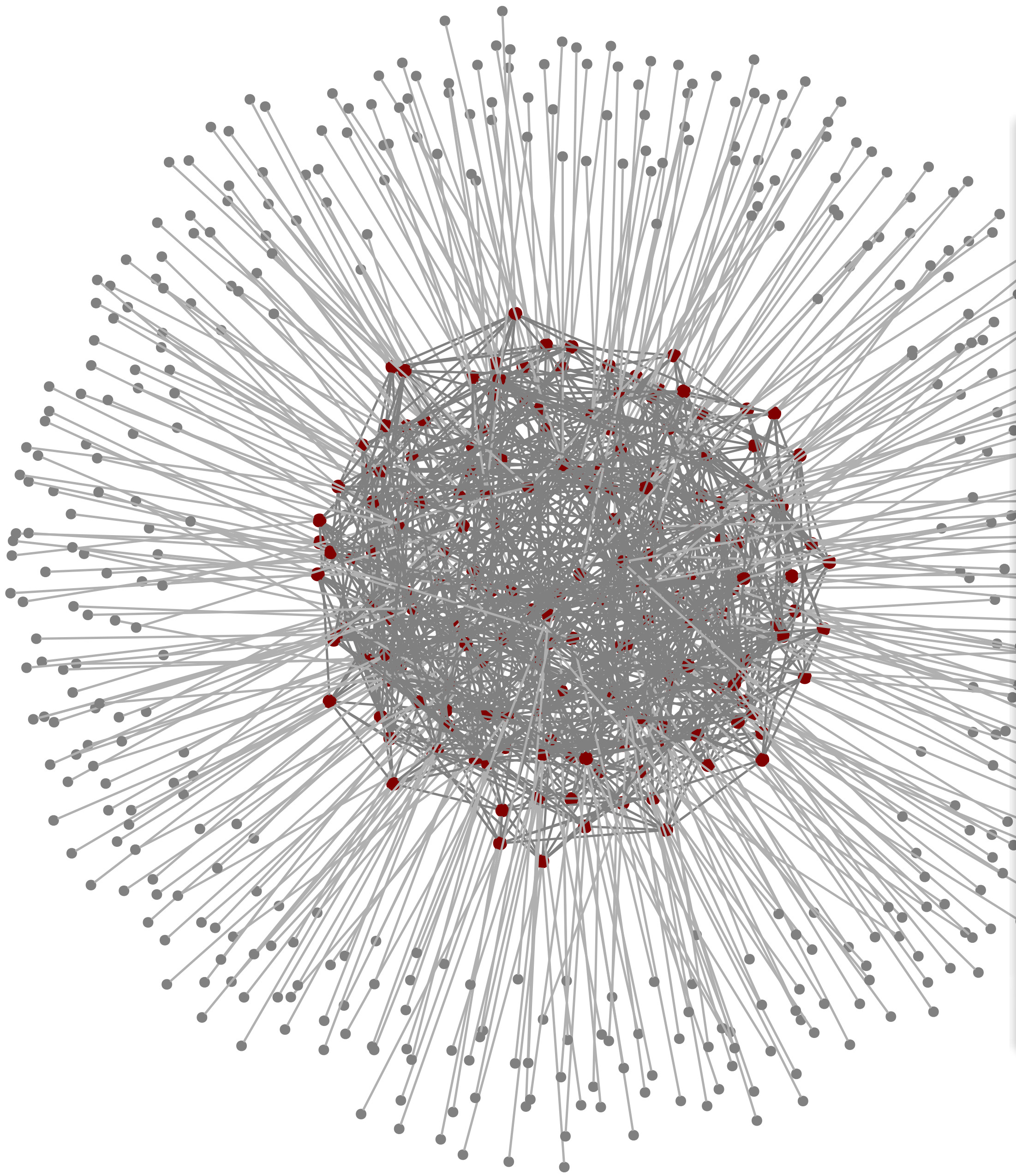
System Design:

Cabling

Cabling



Cabling



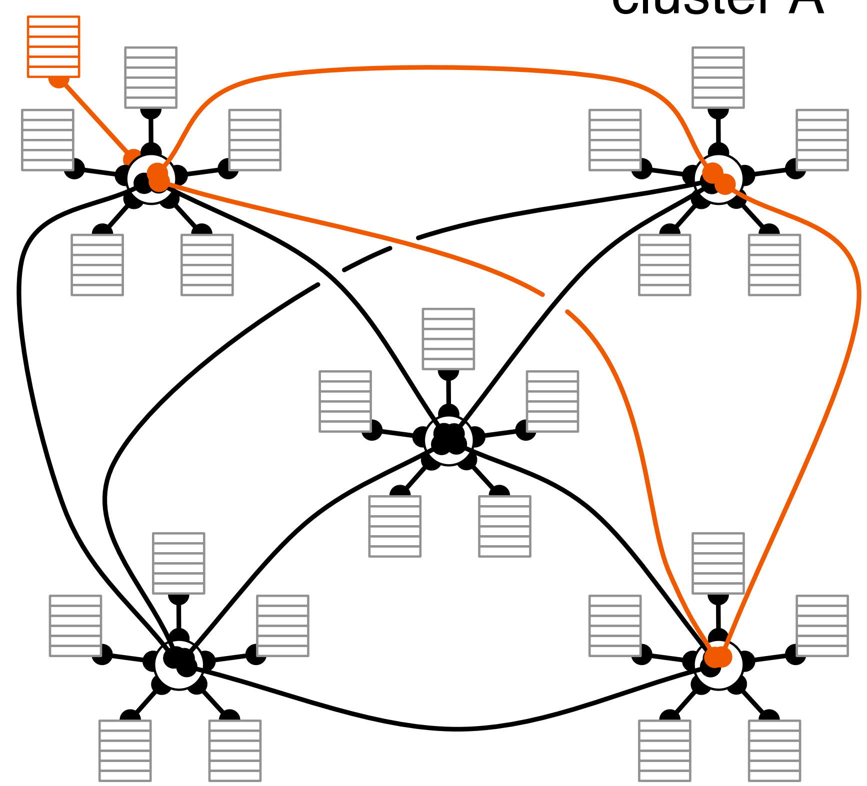
[Photo: Javier Lastras / Wikimedia]

Cabling solutions

Fewer cables

for same # servers as fat tree

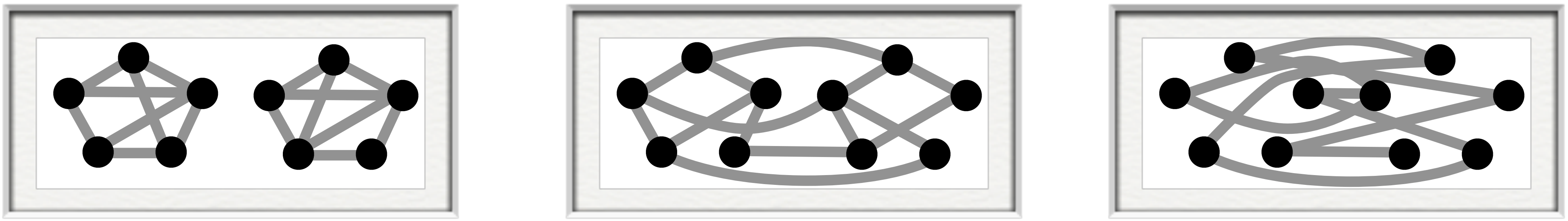
Aggregate bundles



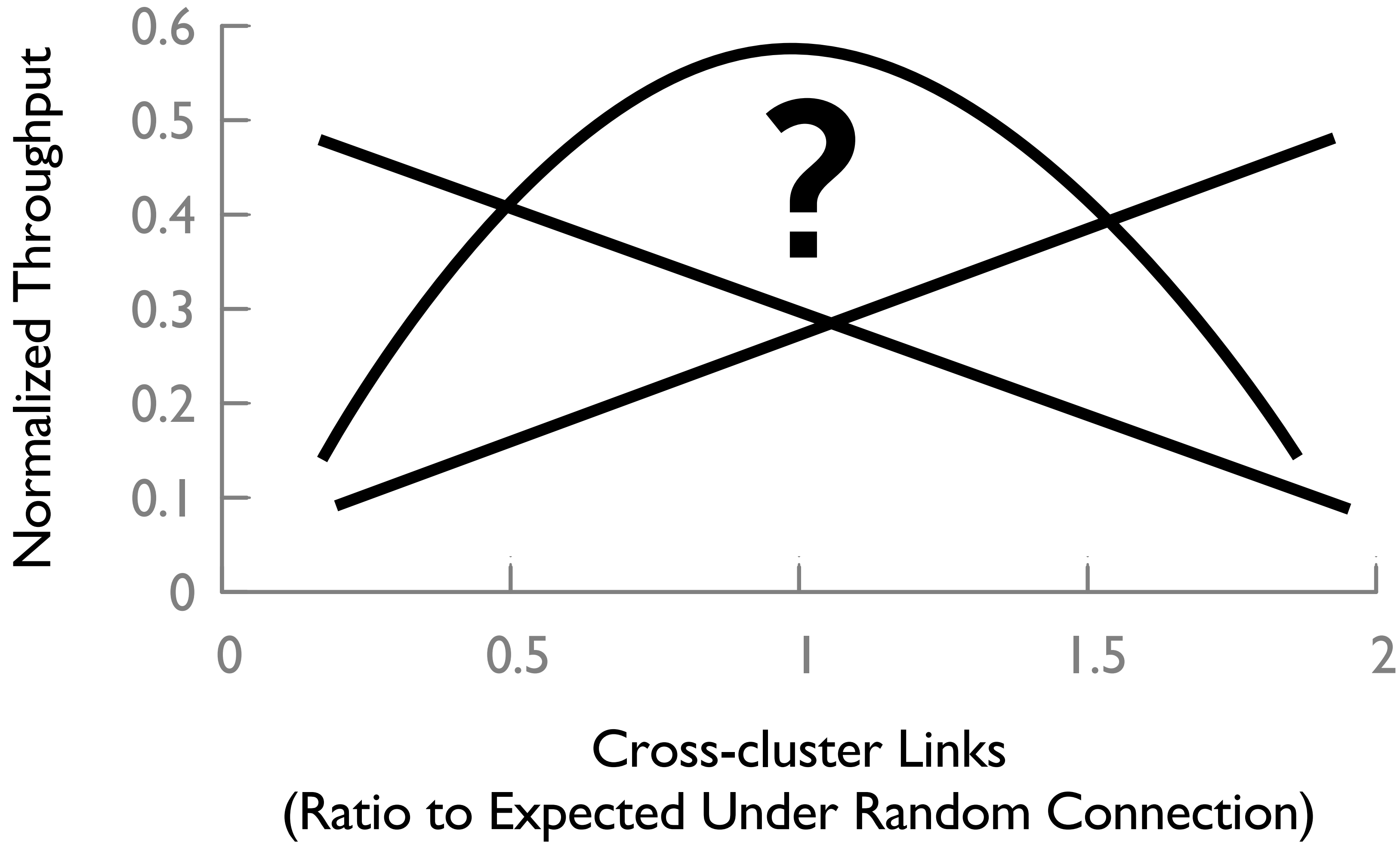
Generic optimization: Place all switches centrally

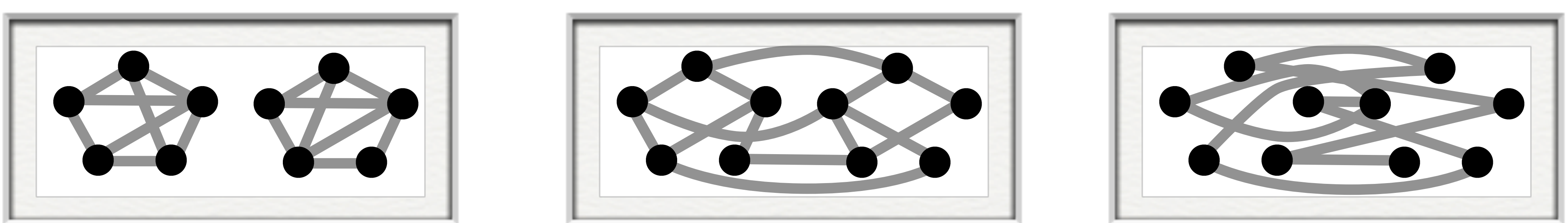
Interconnecting clusters

How many “long” cables do we need?

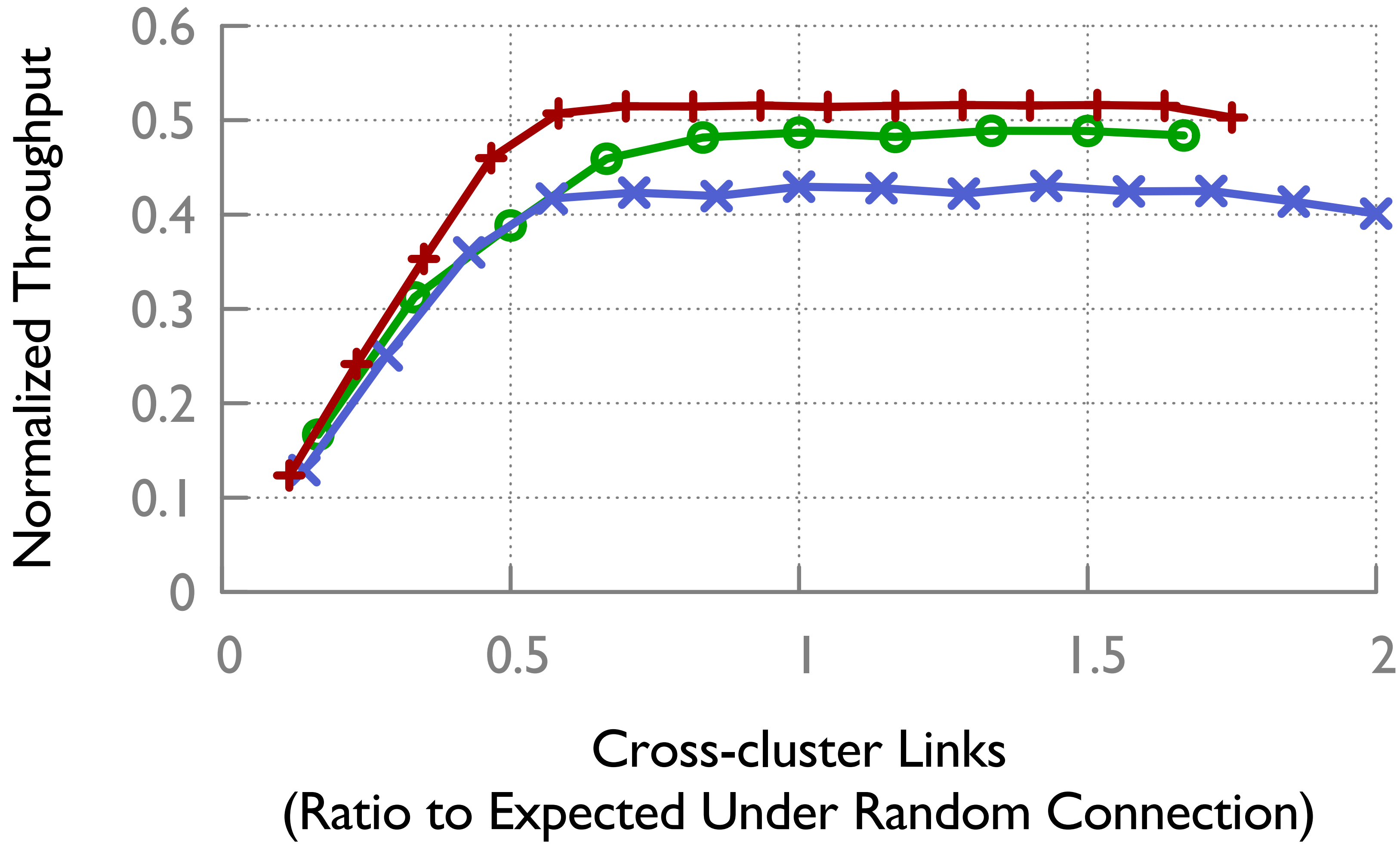


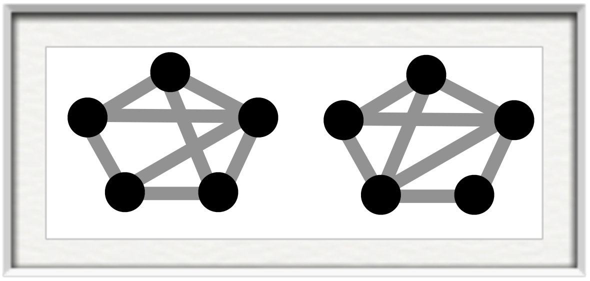
Interconnecting clusters

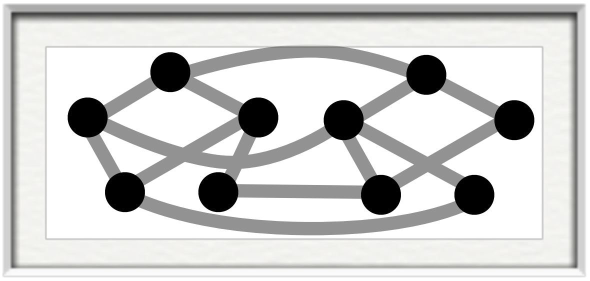


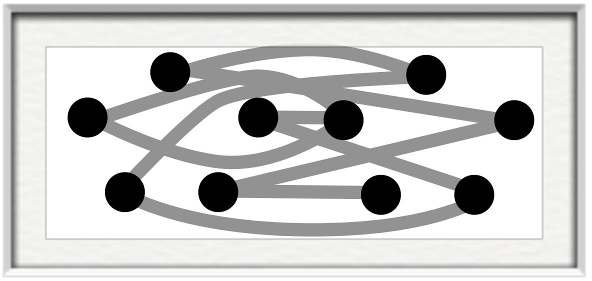


Interconnecting clusters

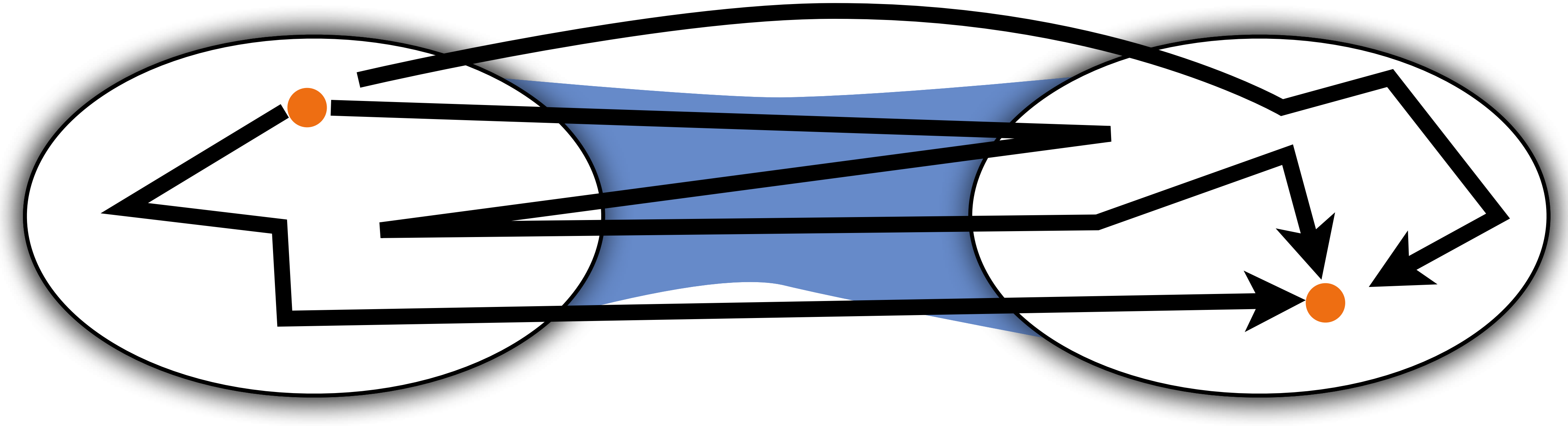




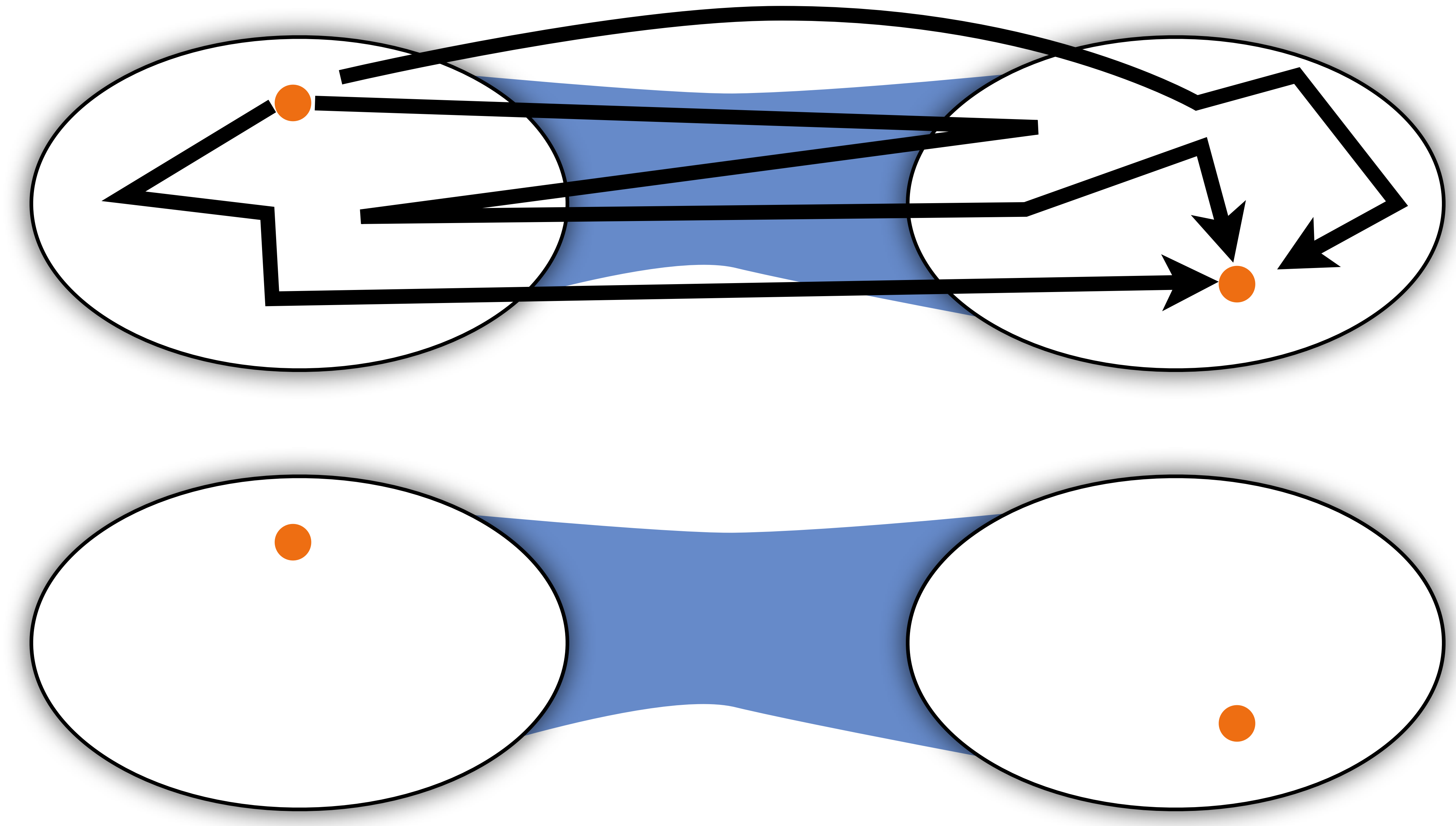




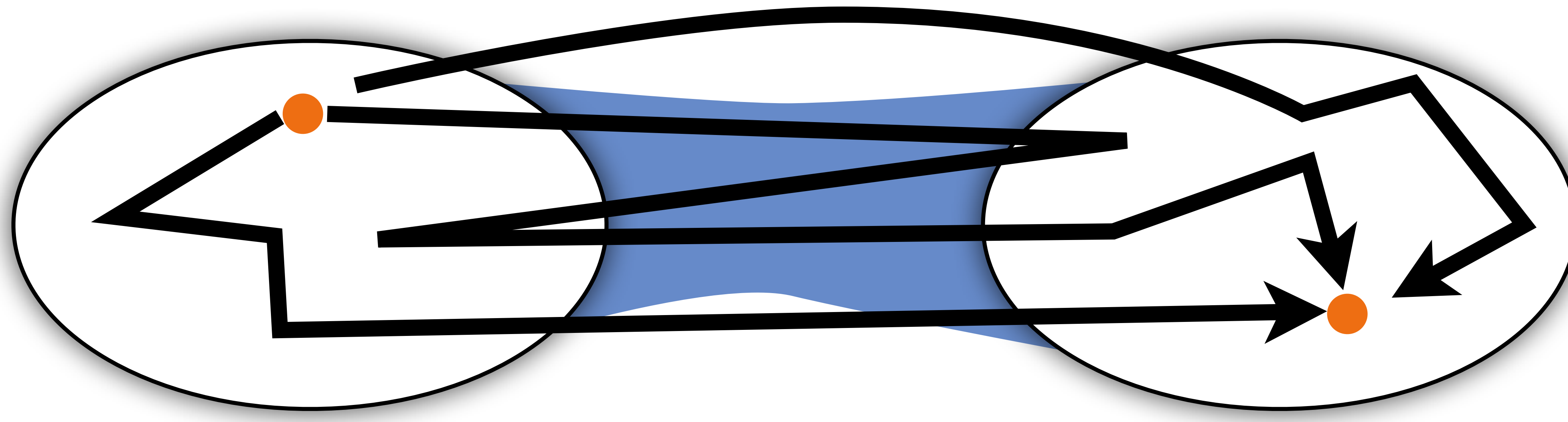
Intuition

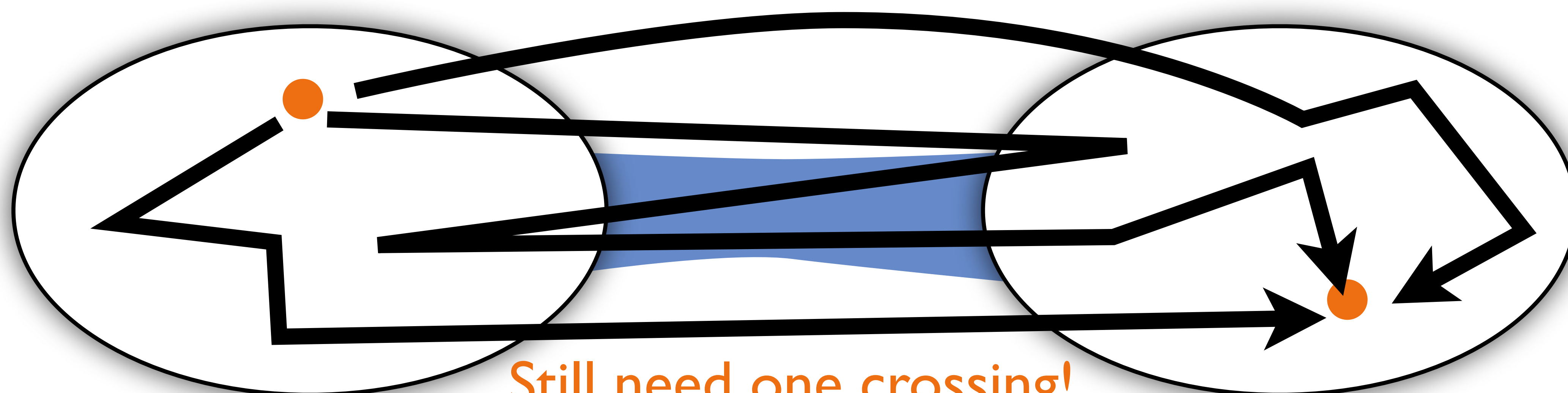


Intuition



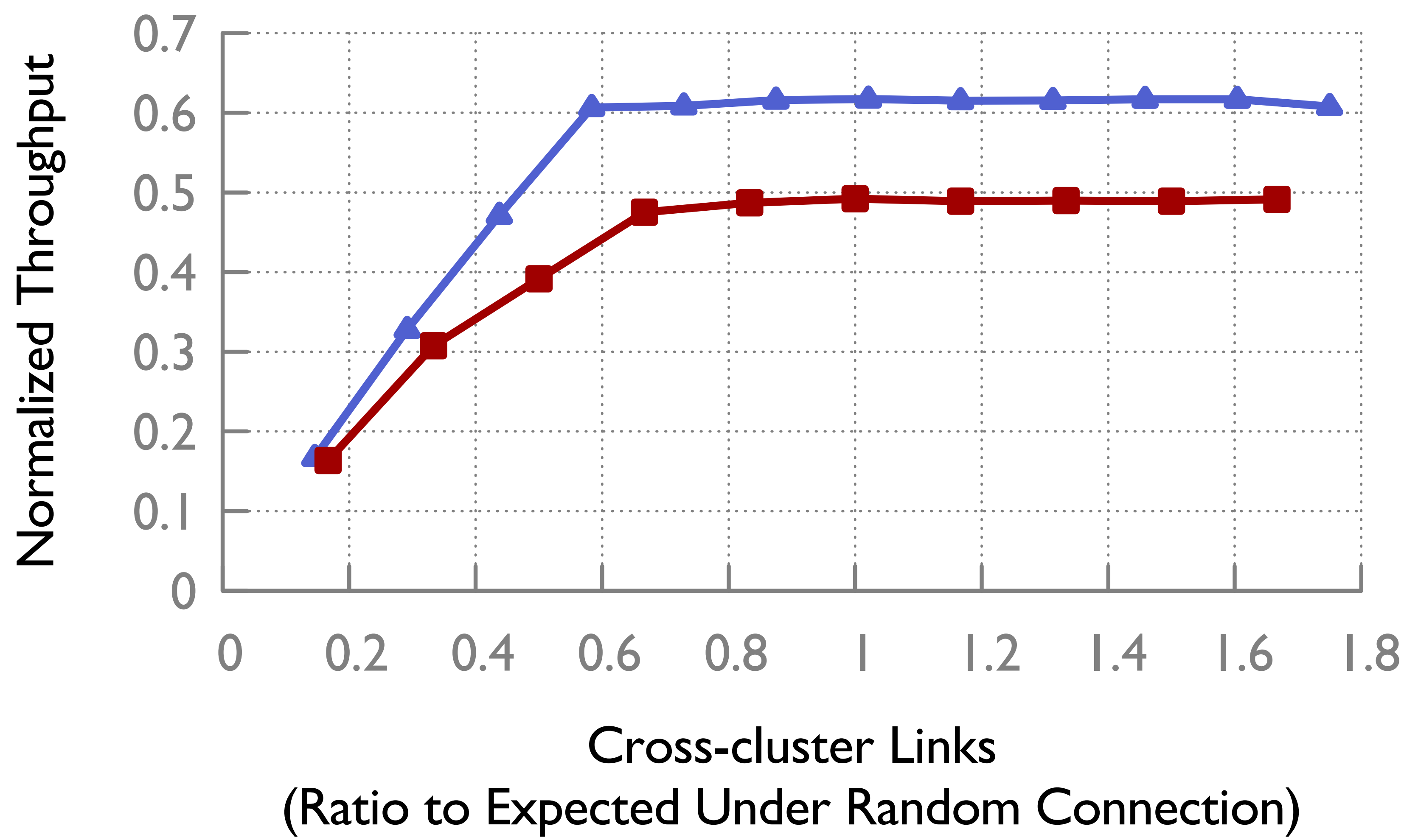
Intuition





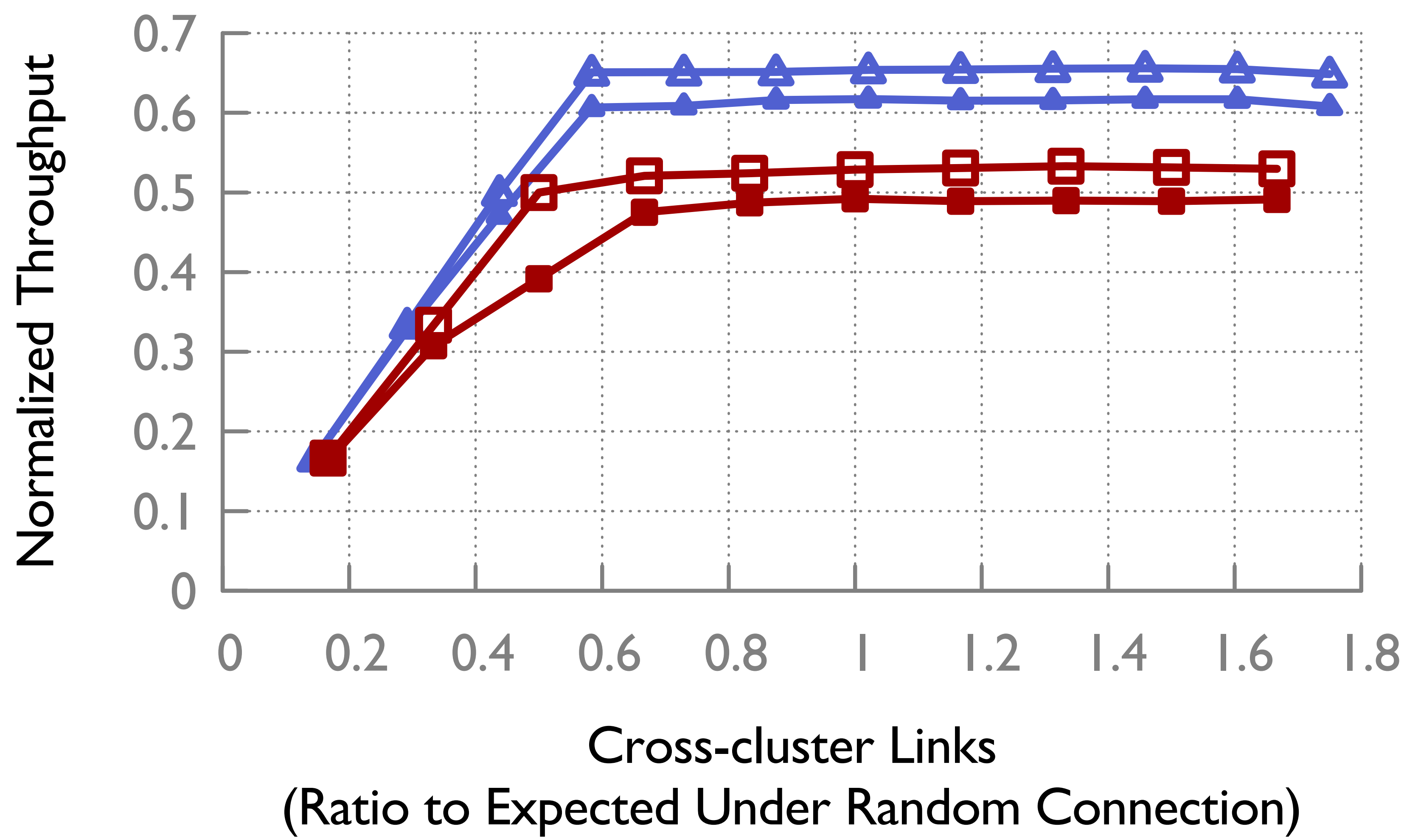
Throughput should drop when less than $\Theta\left(\frac{1}{APL}\right)$ of total capacity crosses the cut!

Explaining throughput



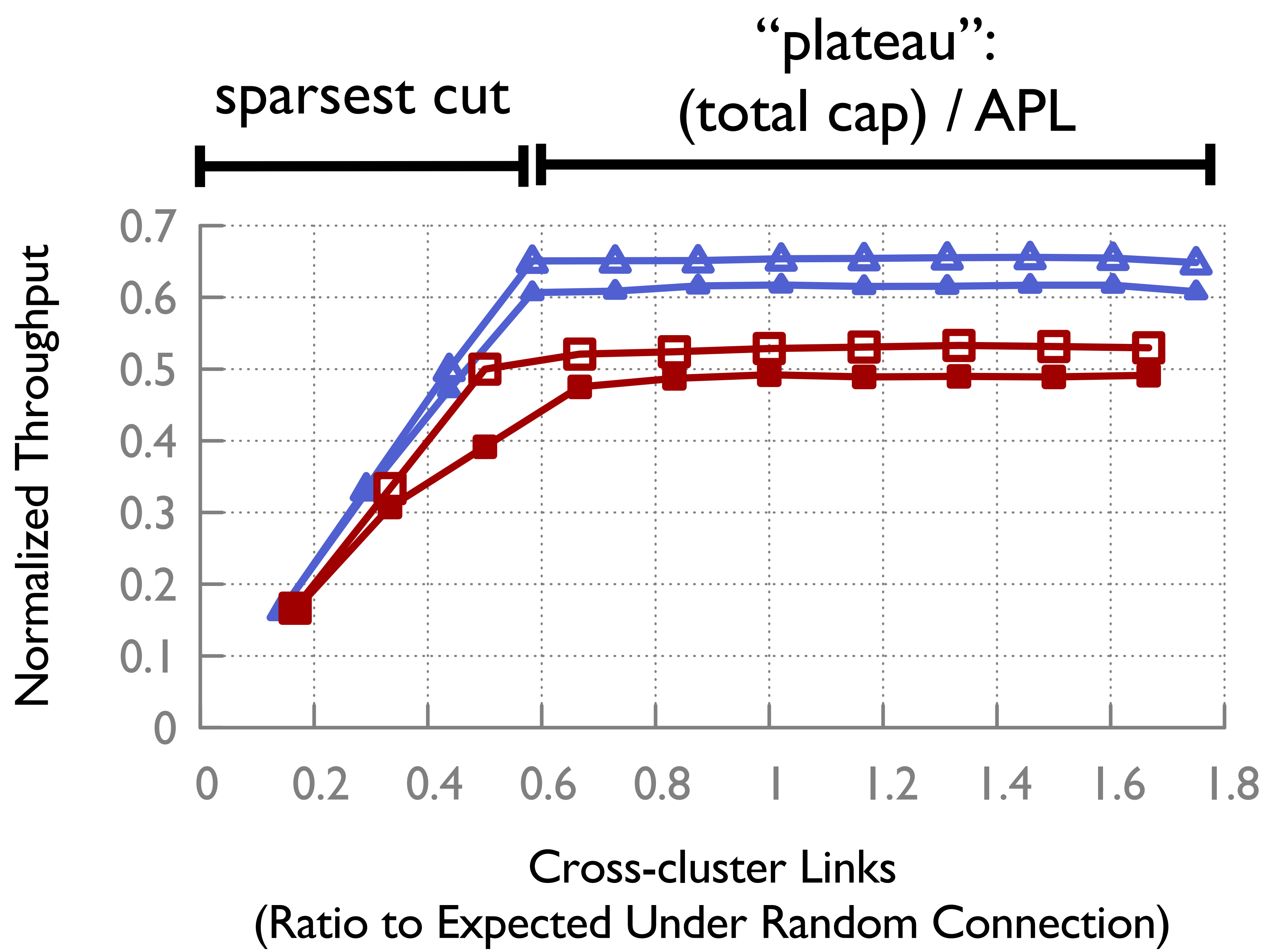
Explaining throughput

Upper bounds...

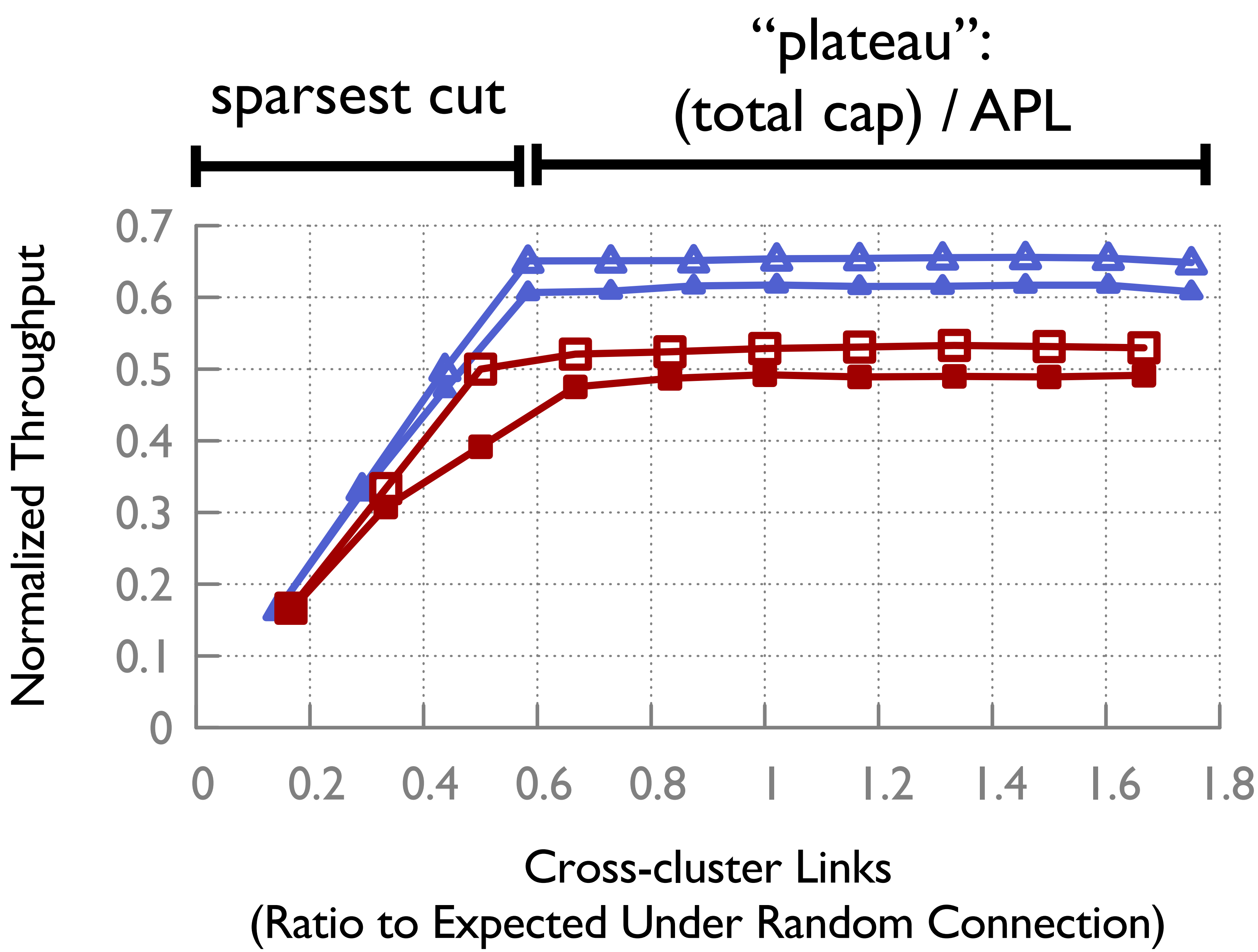


And constant-factor matching lower bounds in special case.

Two regimes of throughput



Two regimes of throughput



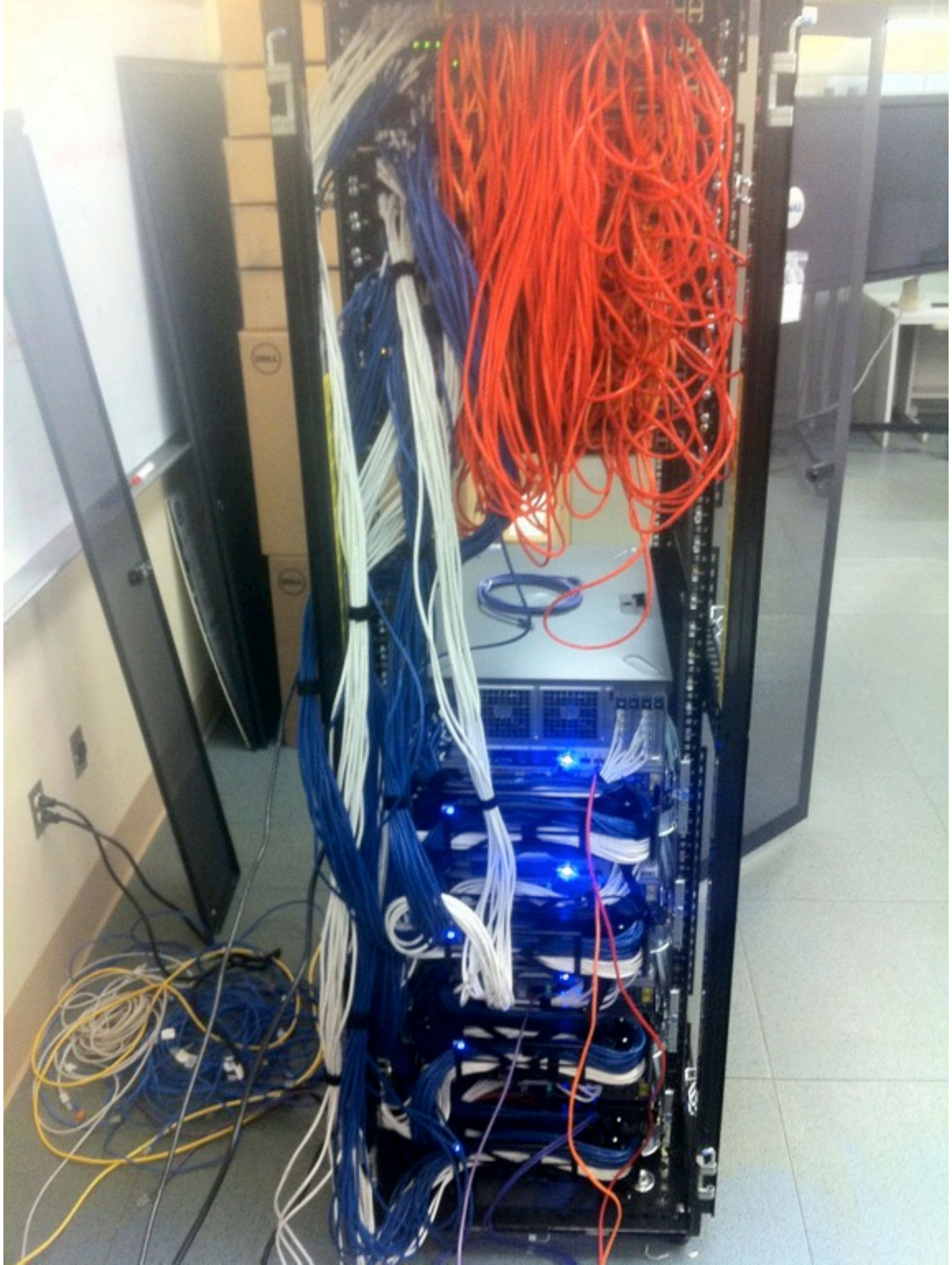
High-capacity switches needn’t be clustered

Bisection bandwidth is poor predictor of performance!

Cables can be localized

What's Next


min
12
2
1
0
13 × 13 × 2
144
12
156
312
contention
contention



Research agenda

Prototype in the lab

- High throughput routing even in unstructured networks
- New techniques for near-optimal TE applicable generally
- SDN-based implementation

Topology-aware application & VM placement

Tech transfer

For more...

"Networking Data Centers Randomly"

A. Singla, C. Hong, L. Popa, P. B. Godfrey
NSDI 2012

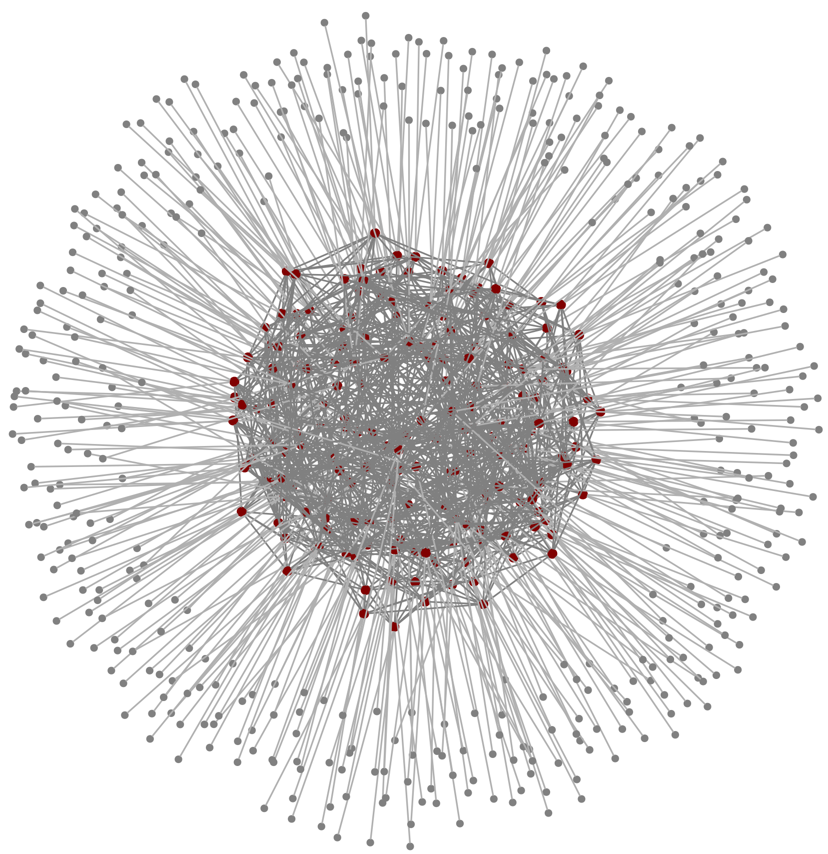
"High throughput data center topology design"

A. Singla, P. B. Godfrey, A. Kolla
Manuscript (check arxiv soon!)

Conclusion

High throughput

Expandability

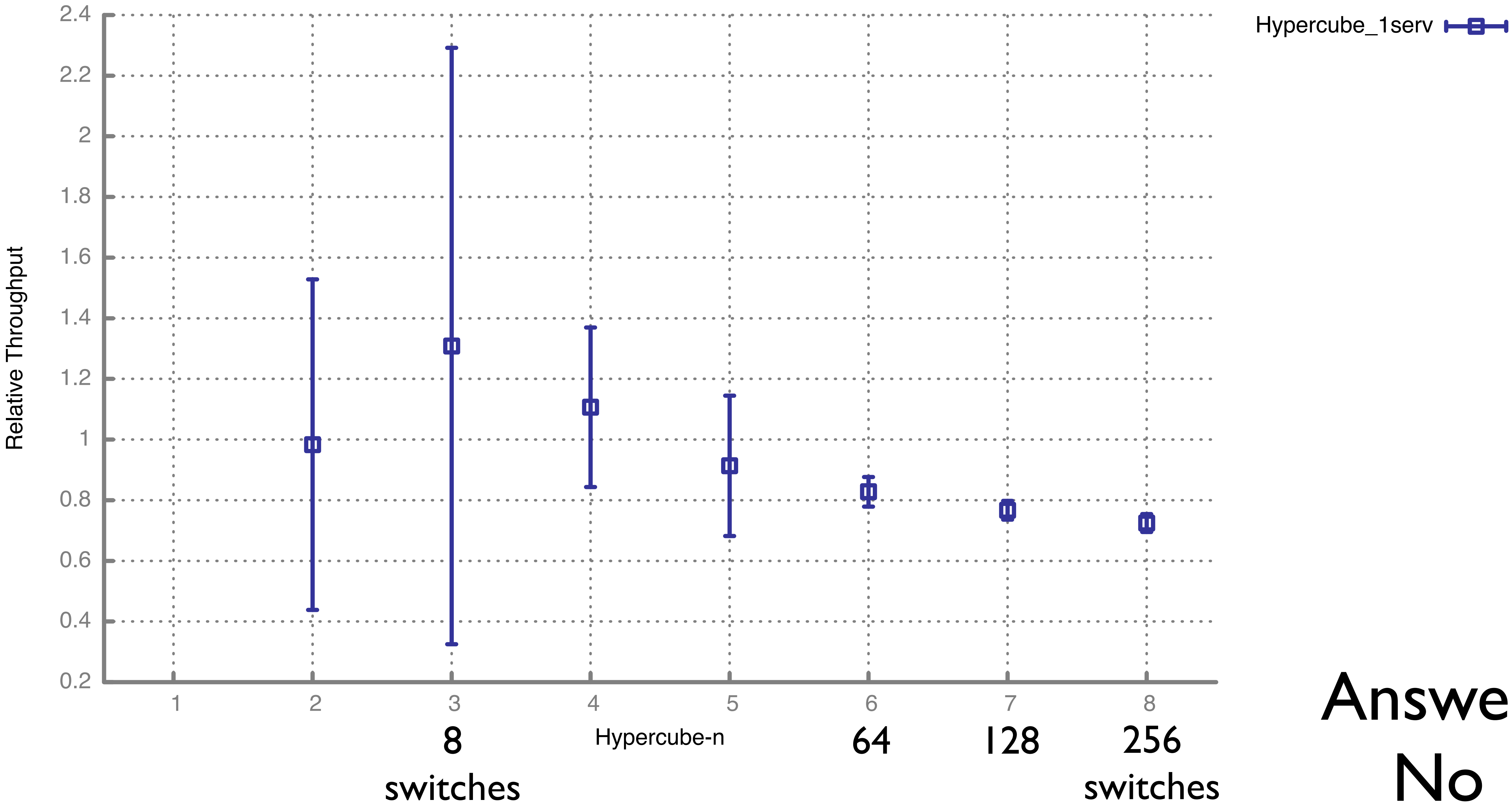


[Photo: Kevin Raskoff]

Backup Slides

Hypercube vs. Random Graph

Is Jellyfish's advantage just that it's a "direct" network?



Answer:
No

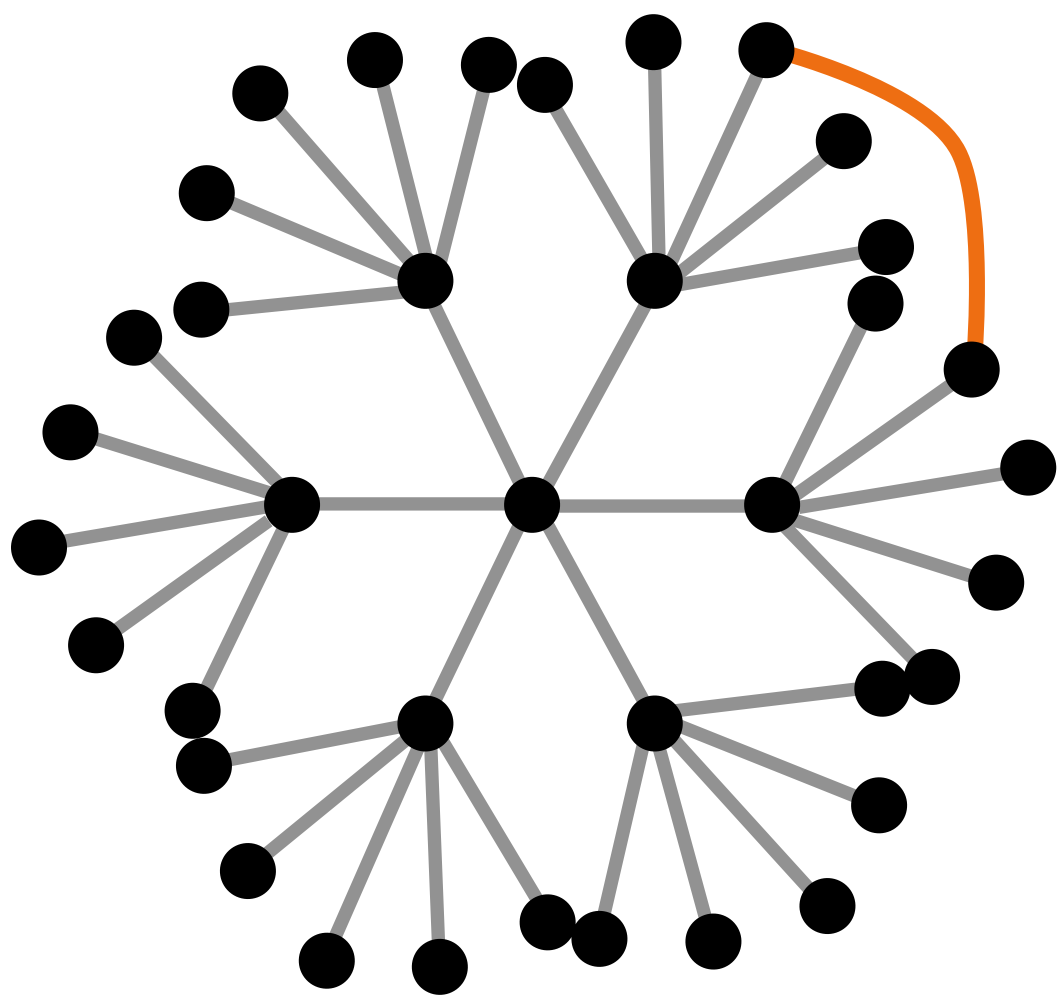
Are There Even Better Topologies?

A simple upper bound

$$\text{Throughput per flow} \leq \frac{\sum_{\text{links}} \text{capacity}(\mathit{link})}{\#\ \text{flows} \cdot \text{mean path length}}$$

Lower bound this!

Lower bound on mean path length

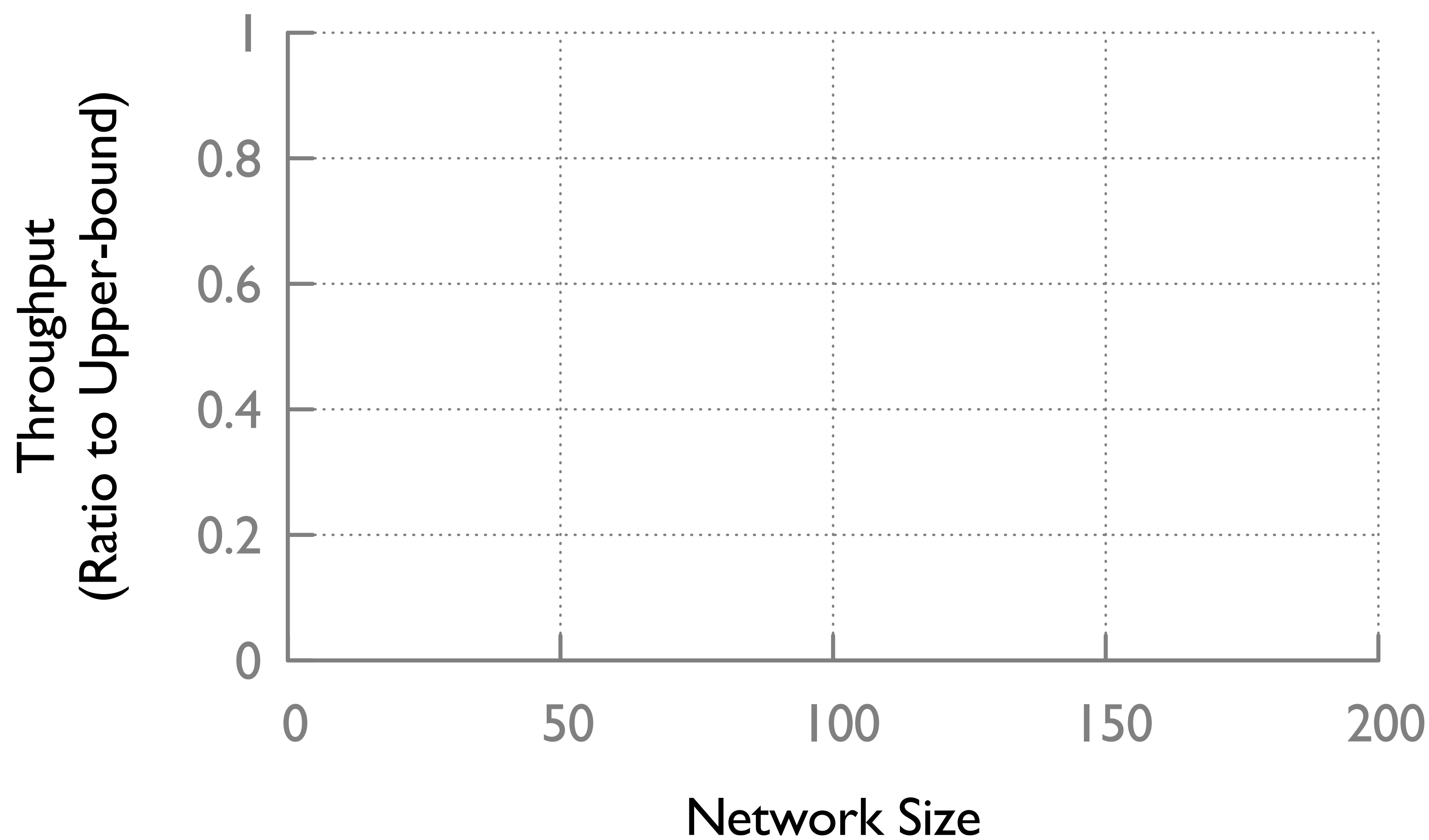


Distance	# Nodes
1	6
2	$6^2 - 6$

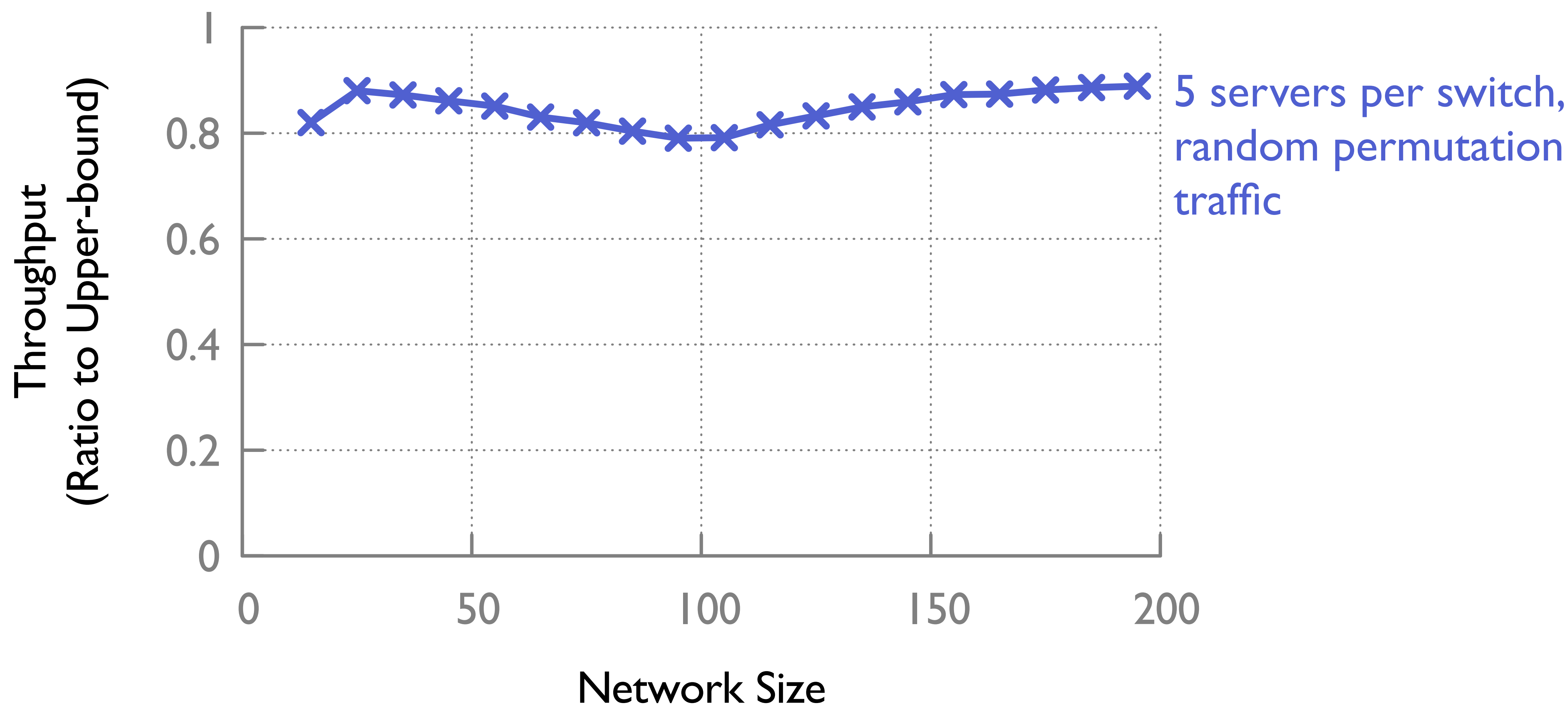
Ugliness omitted!

[Cerf et al., "A lower bound on the average shortest path length in regular graphs", 1974]

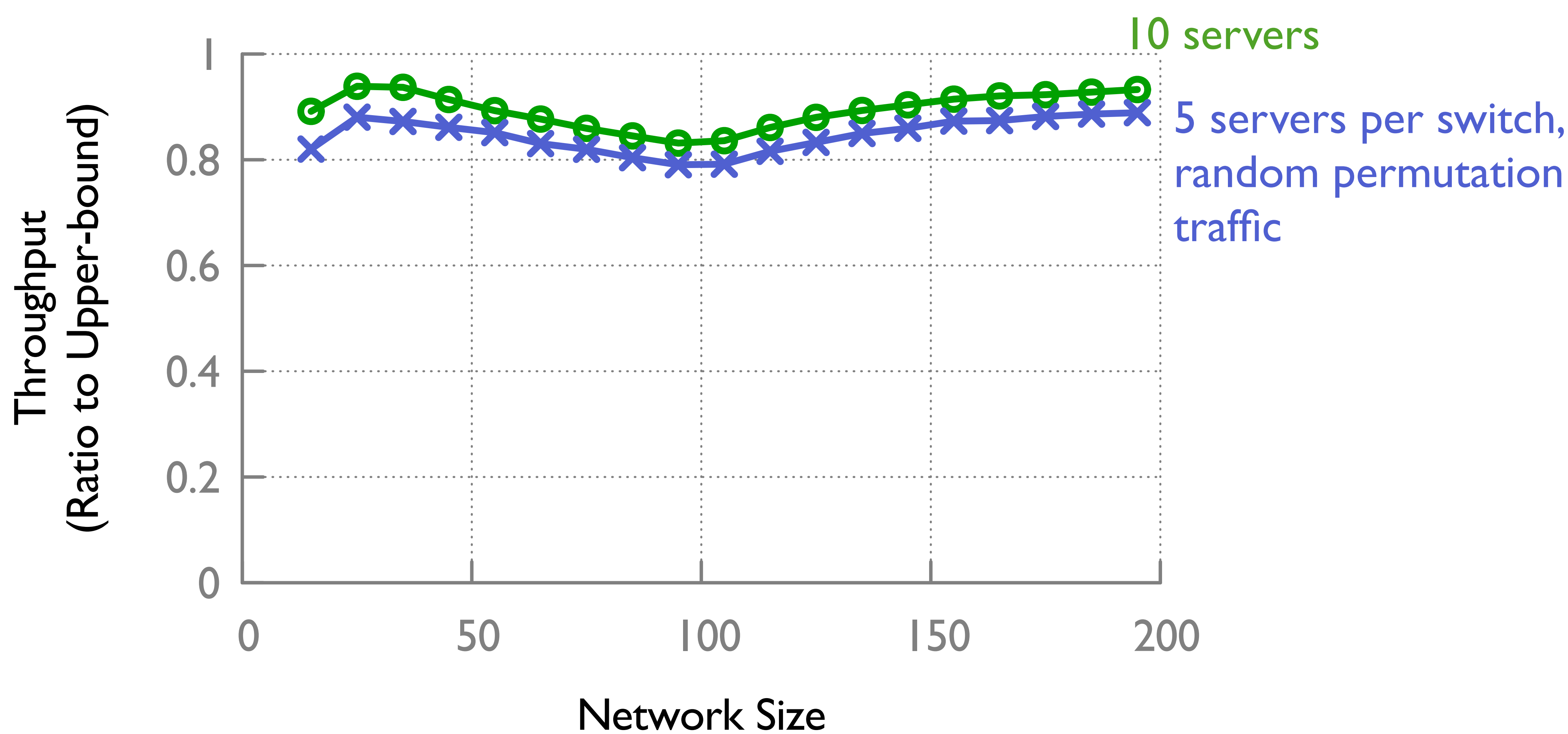
Random graphs vs. upper bound



Random graphs vs. upper bound

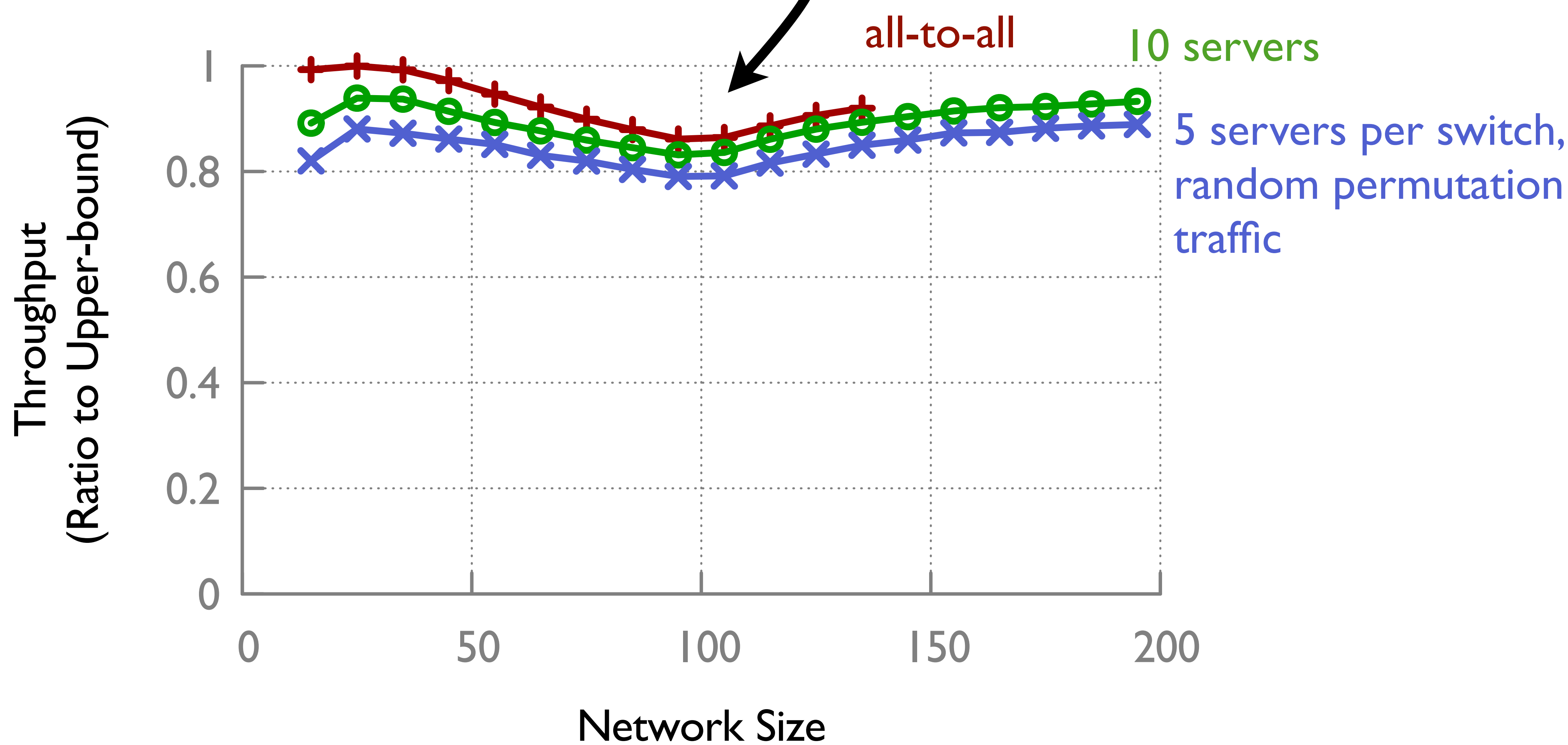


Random graphs vs. upper bound



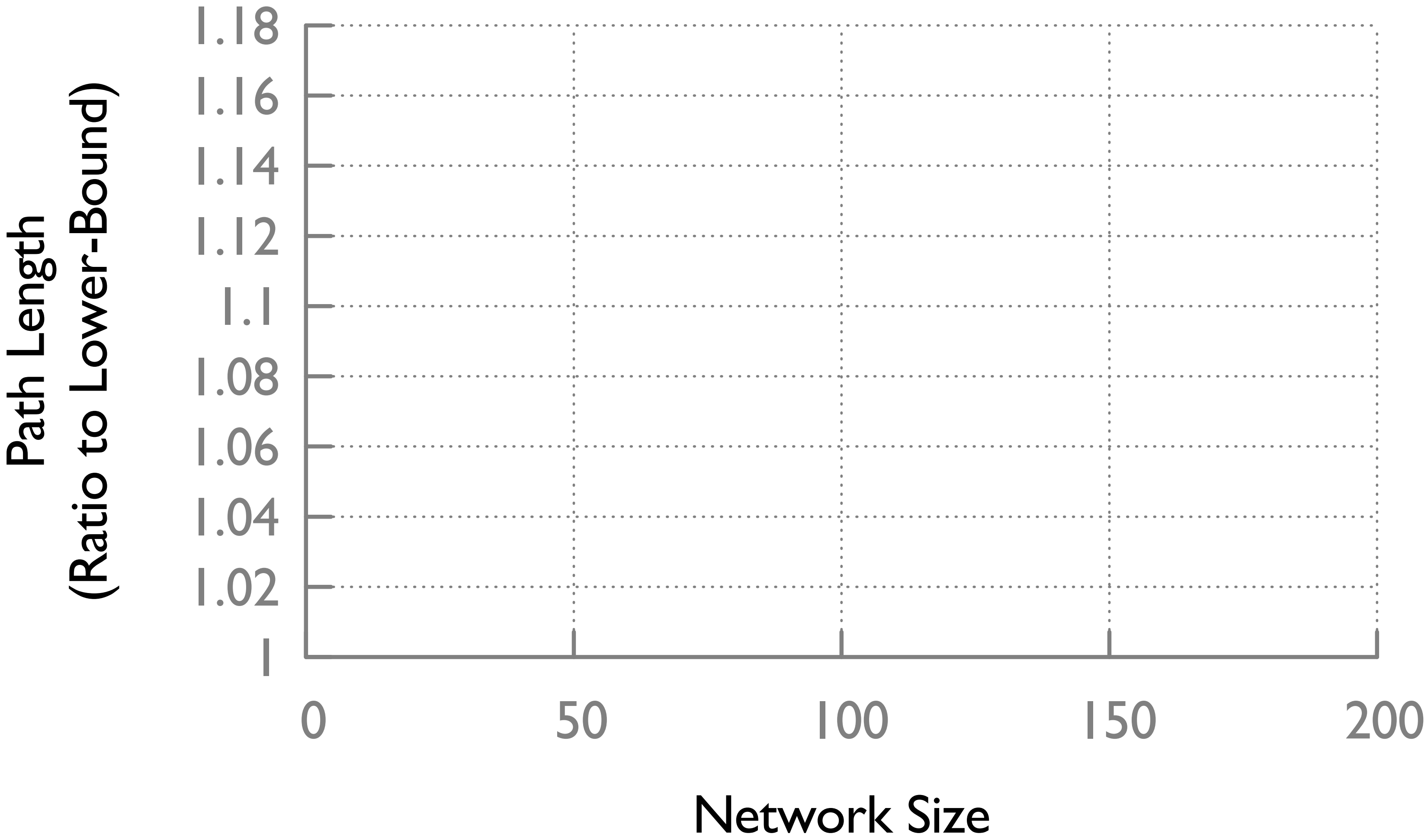
Random graphs vs. upper bound

(Aside: is any topology closer to the bound?)

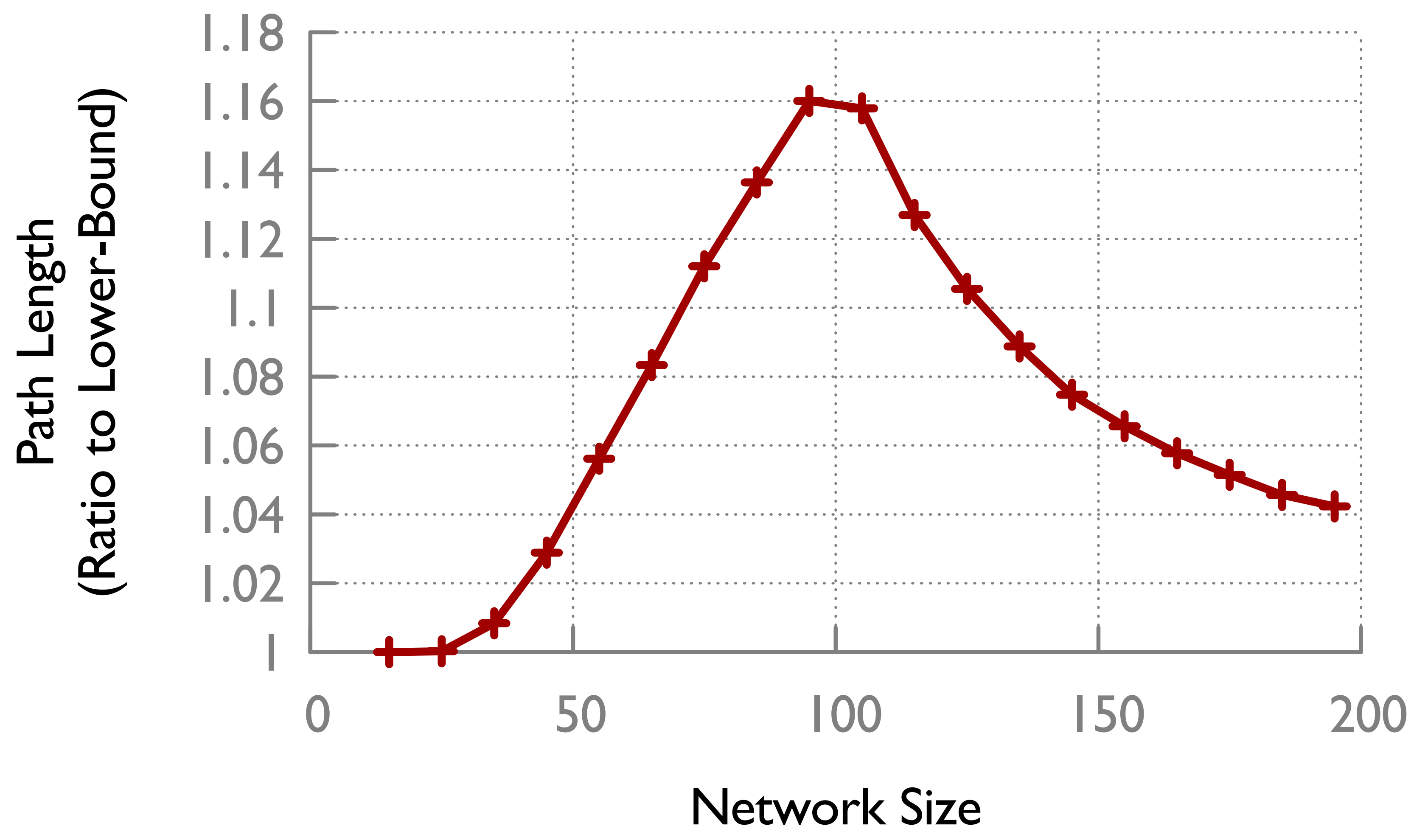


Random graphs within a few percent of optimal!

Random graphs vs. upper bound

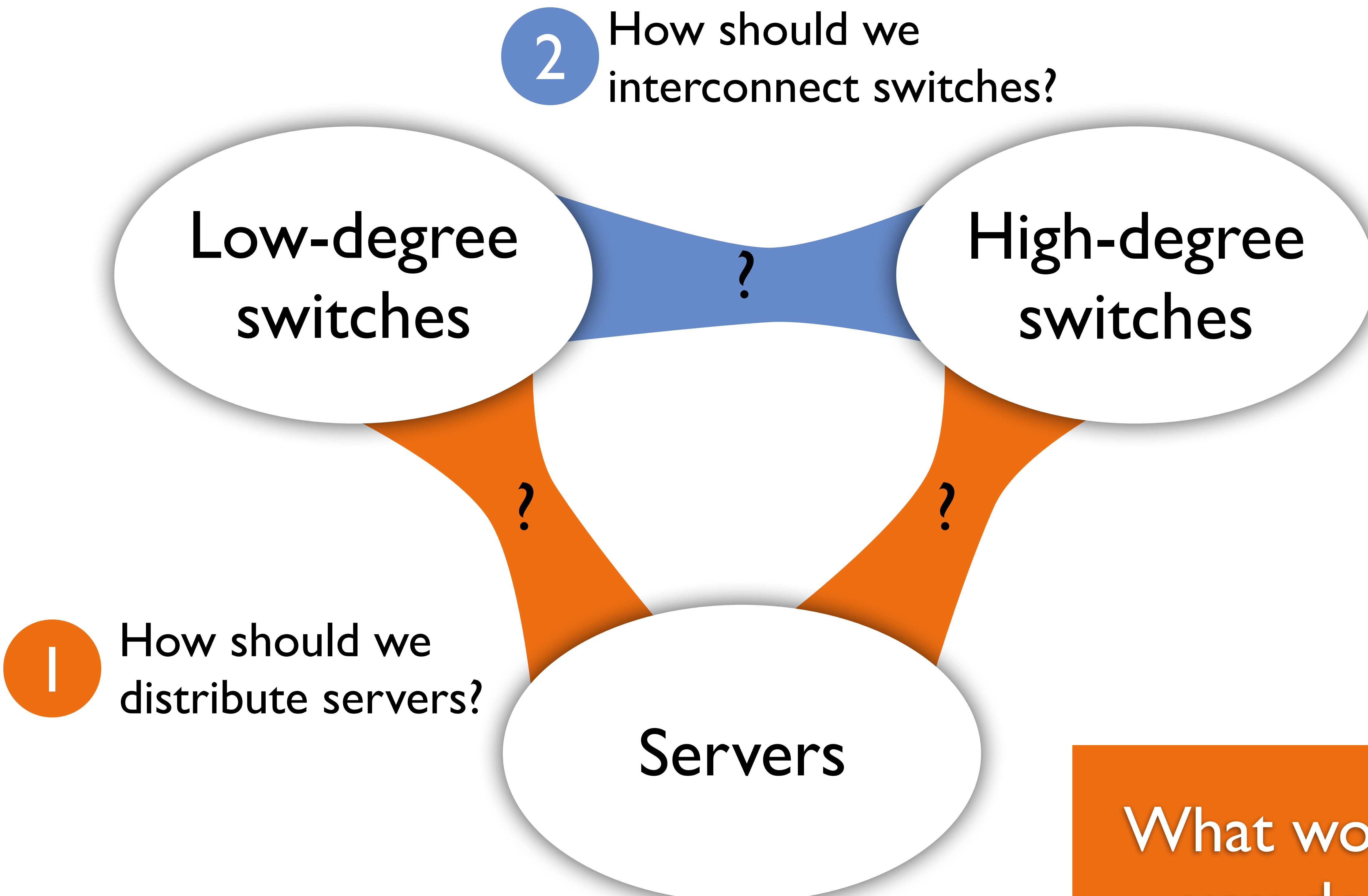


Random graphs vs. upper bound



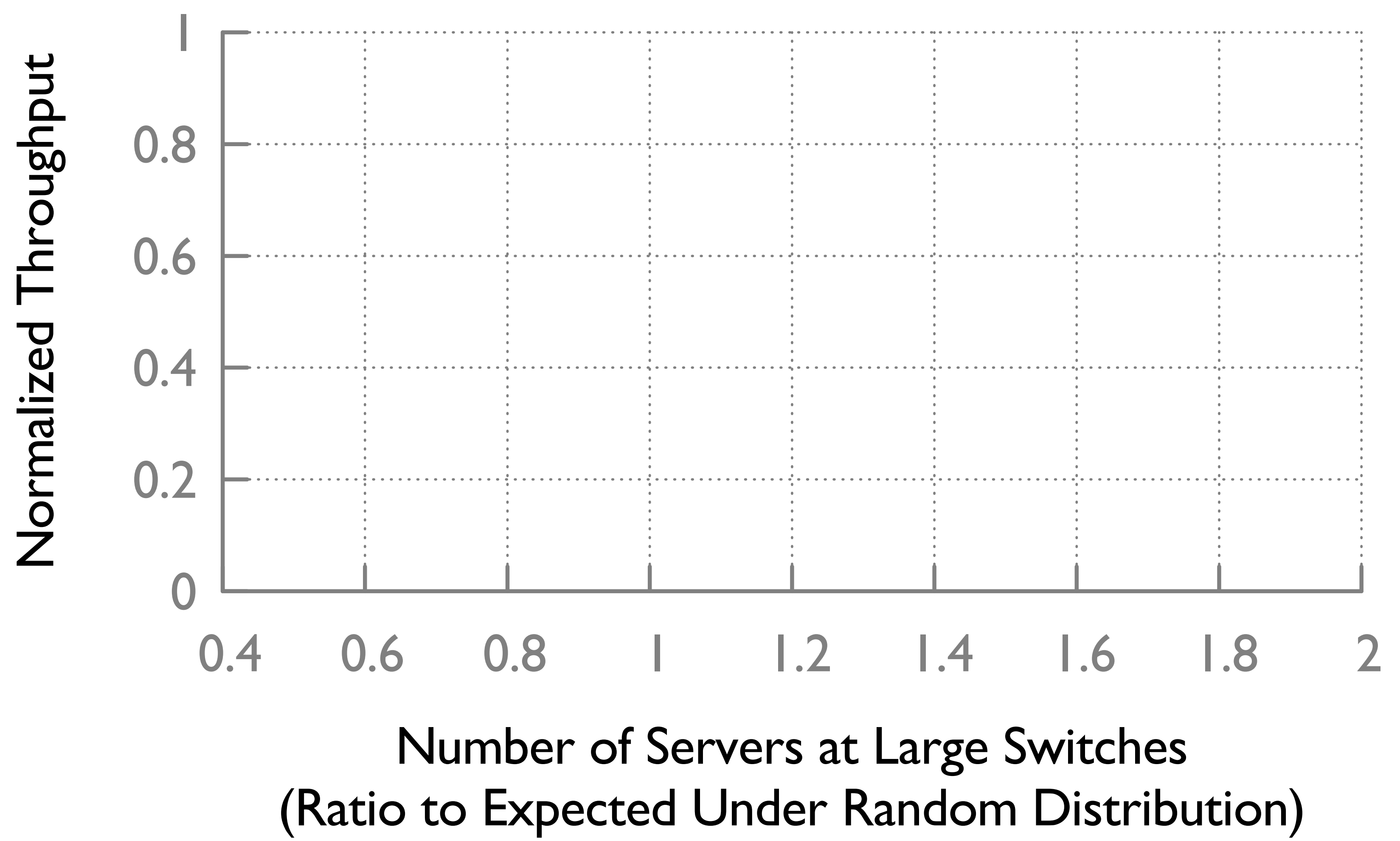
Designing Heterogeneous Networks

Random graphs as a building block



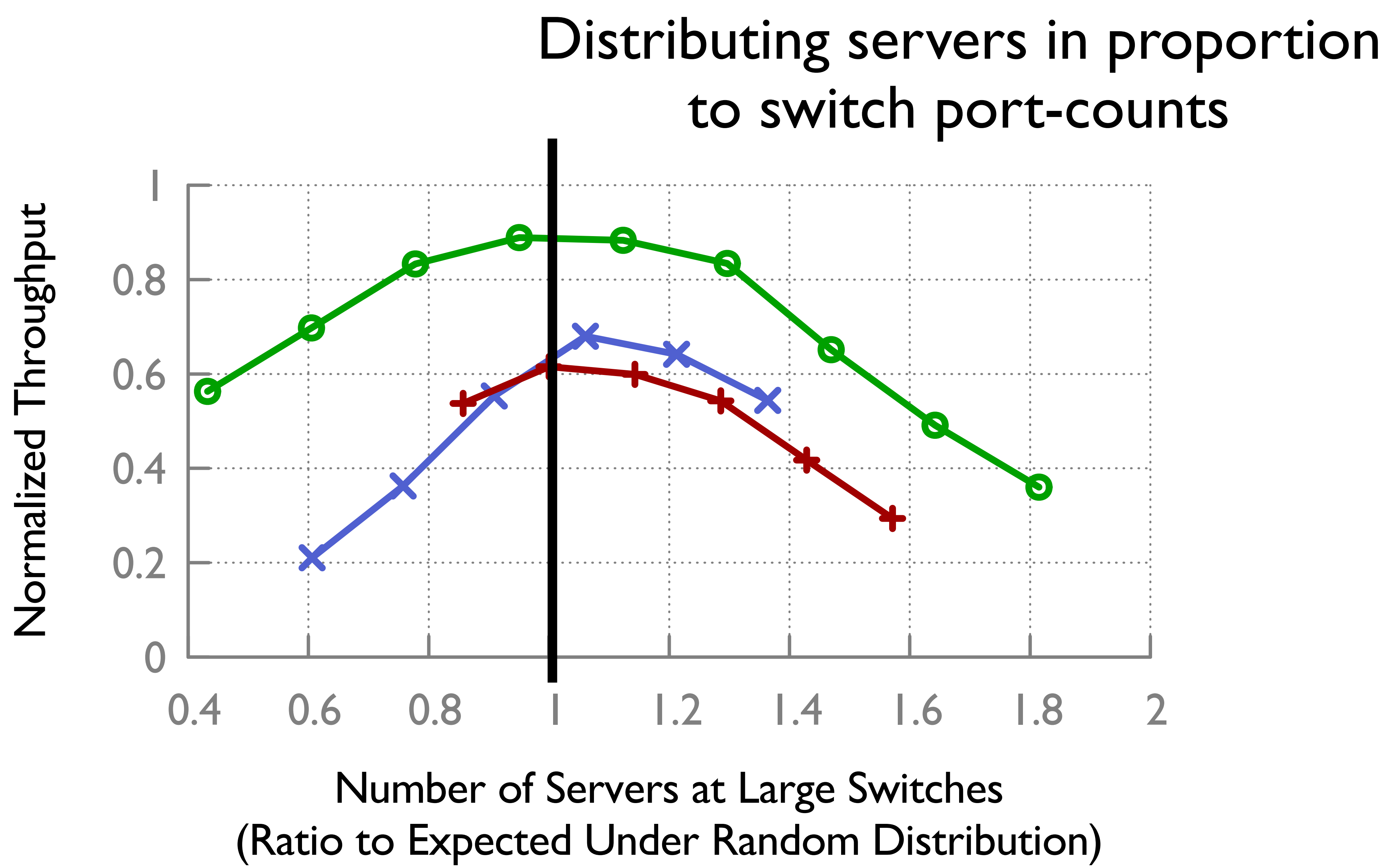
What would you do?

Distributing servers



(The switch interconnect being vanilla random)

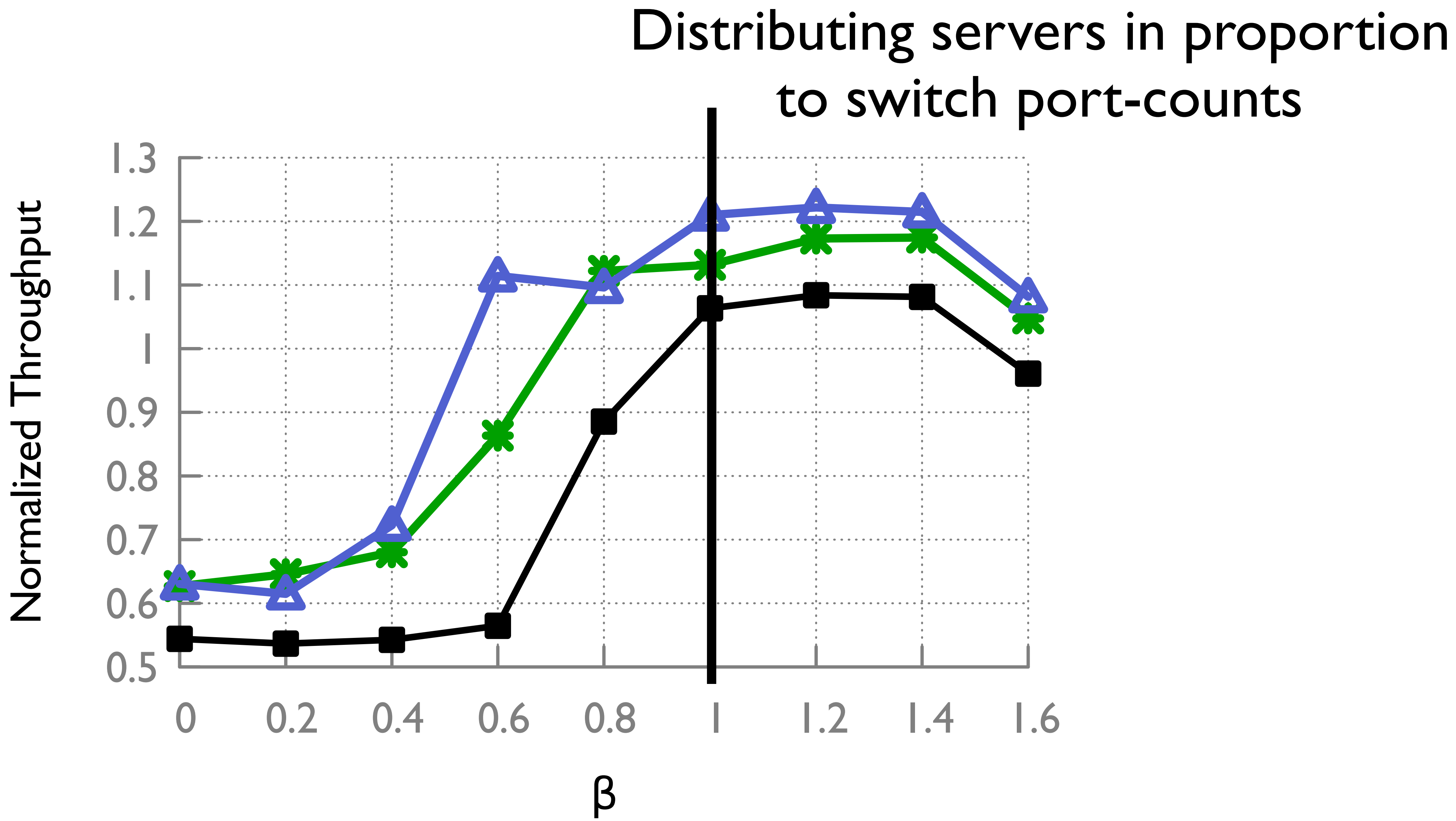
Distributing servers



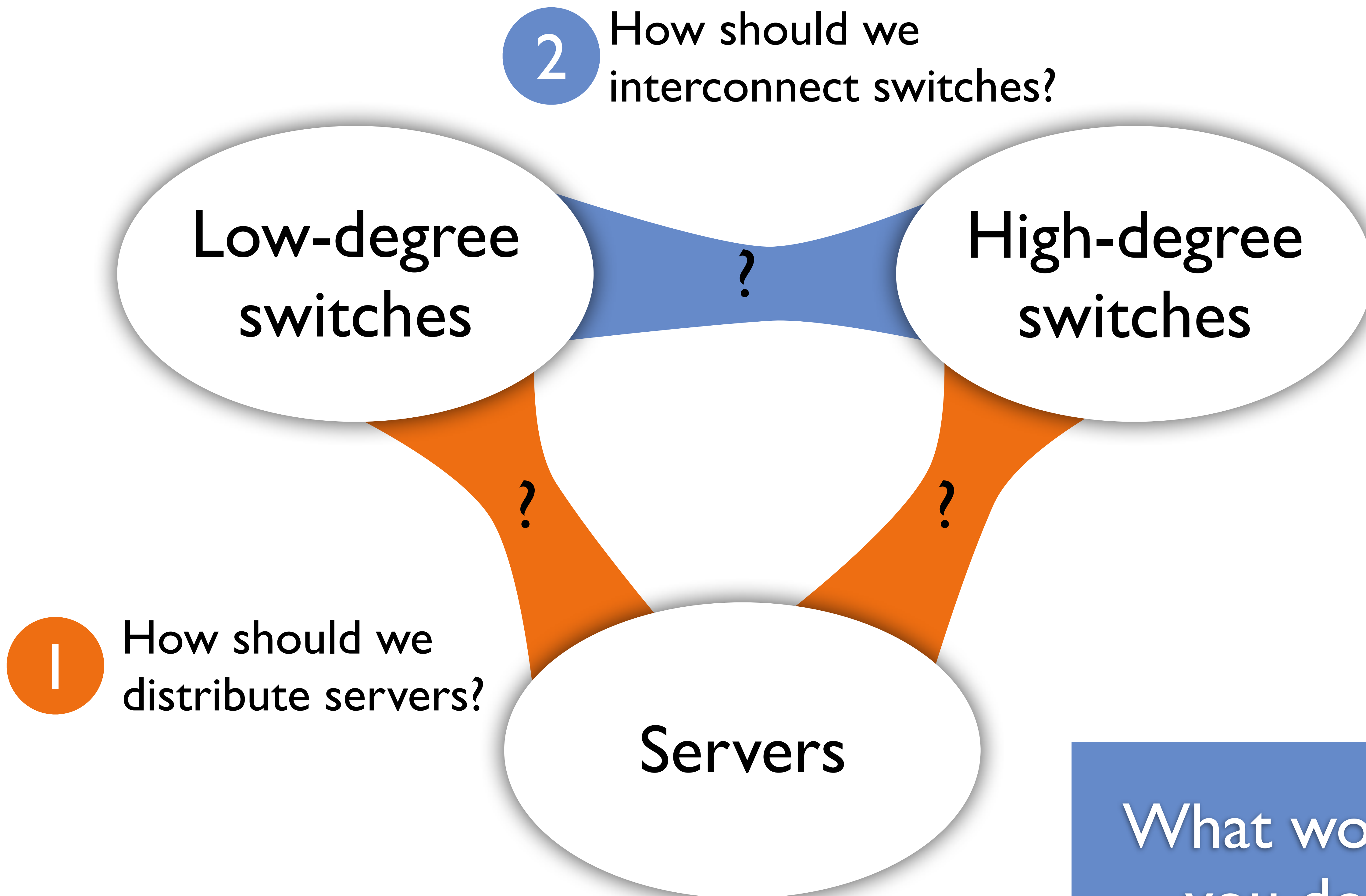
(The switch interconnect being vanilla random)

Distributing servers

#Servers on switch *i* $\propto$ (port-count of *i*)$^{\beta}$

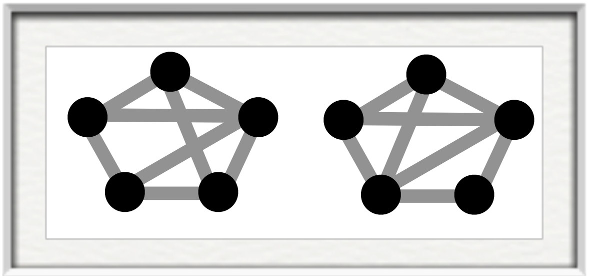


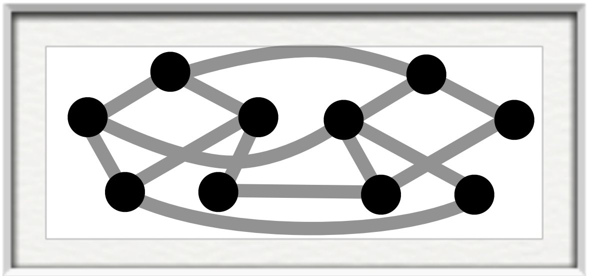
Random graphs as a building block

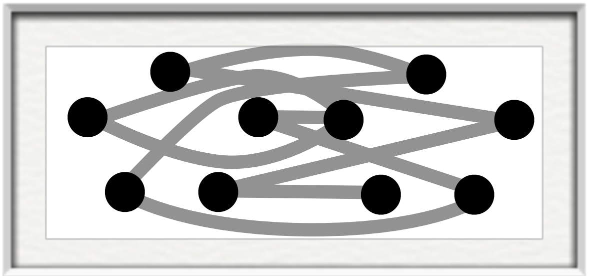


What would you do?

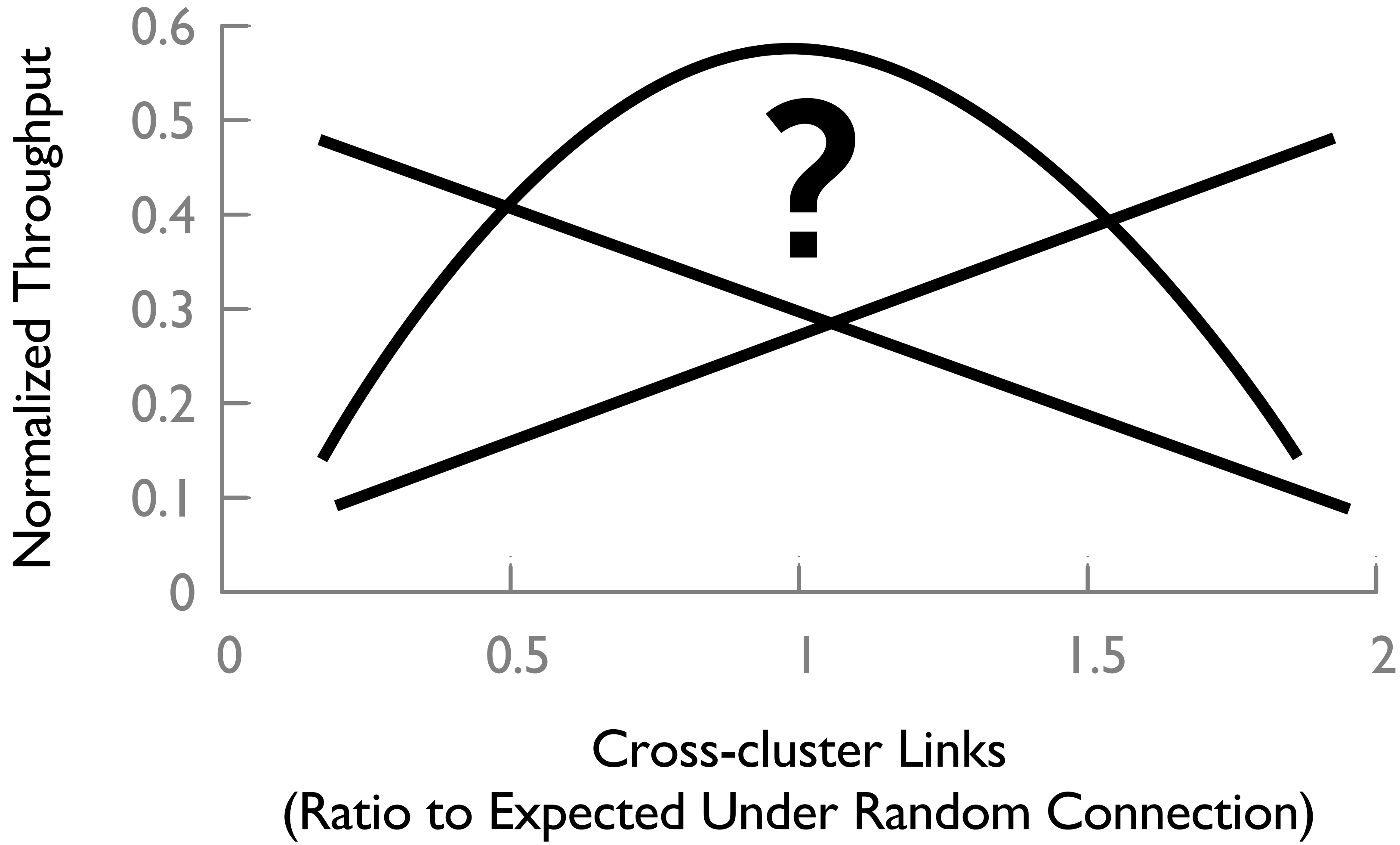
Interconnecting switches

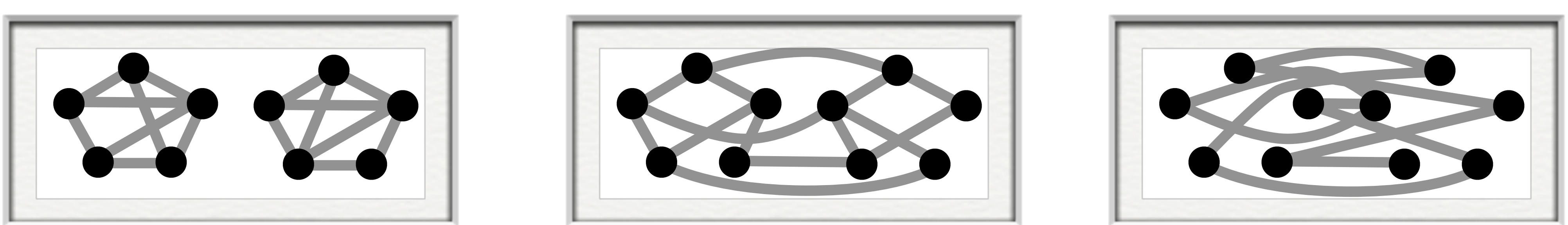




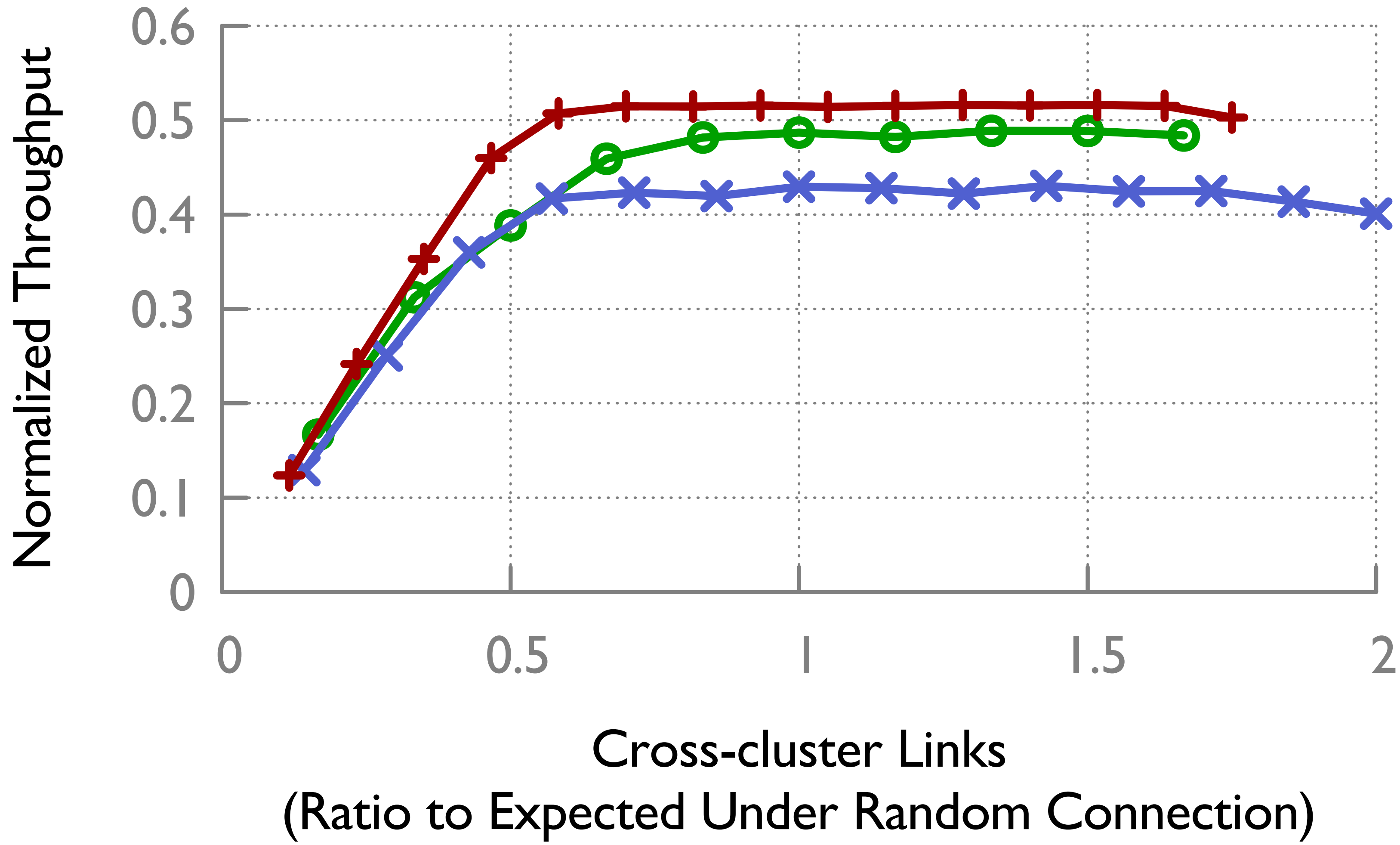


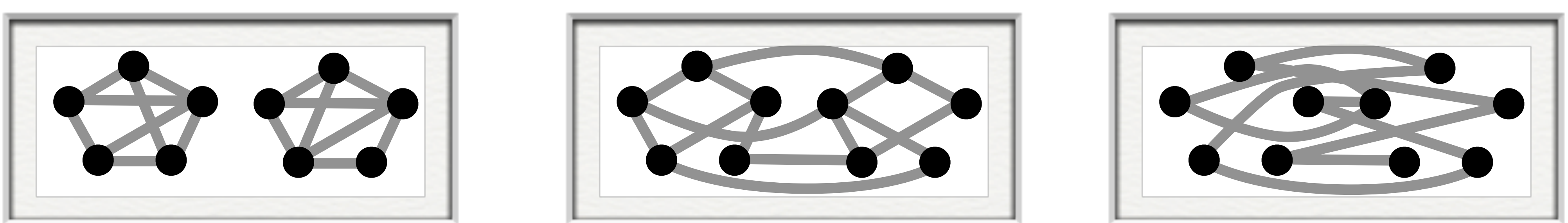
Interconnecting switches



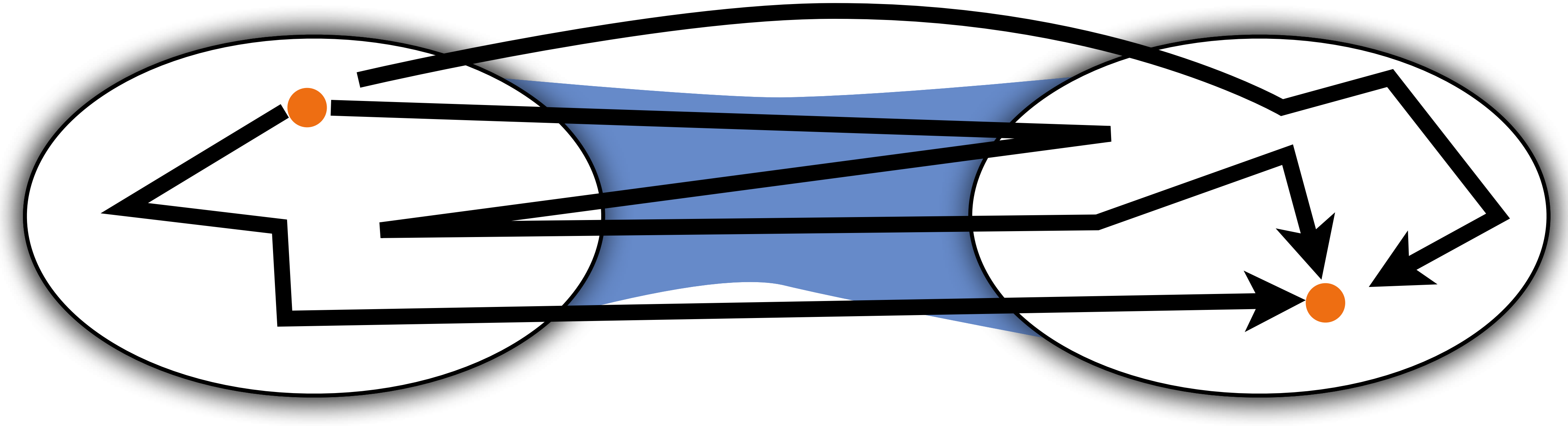


Interconnecting switches

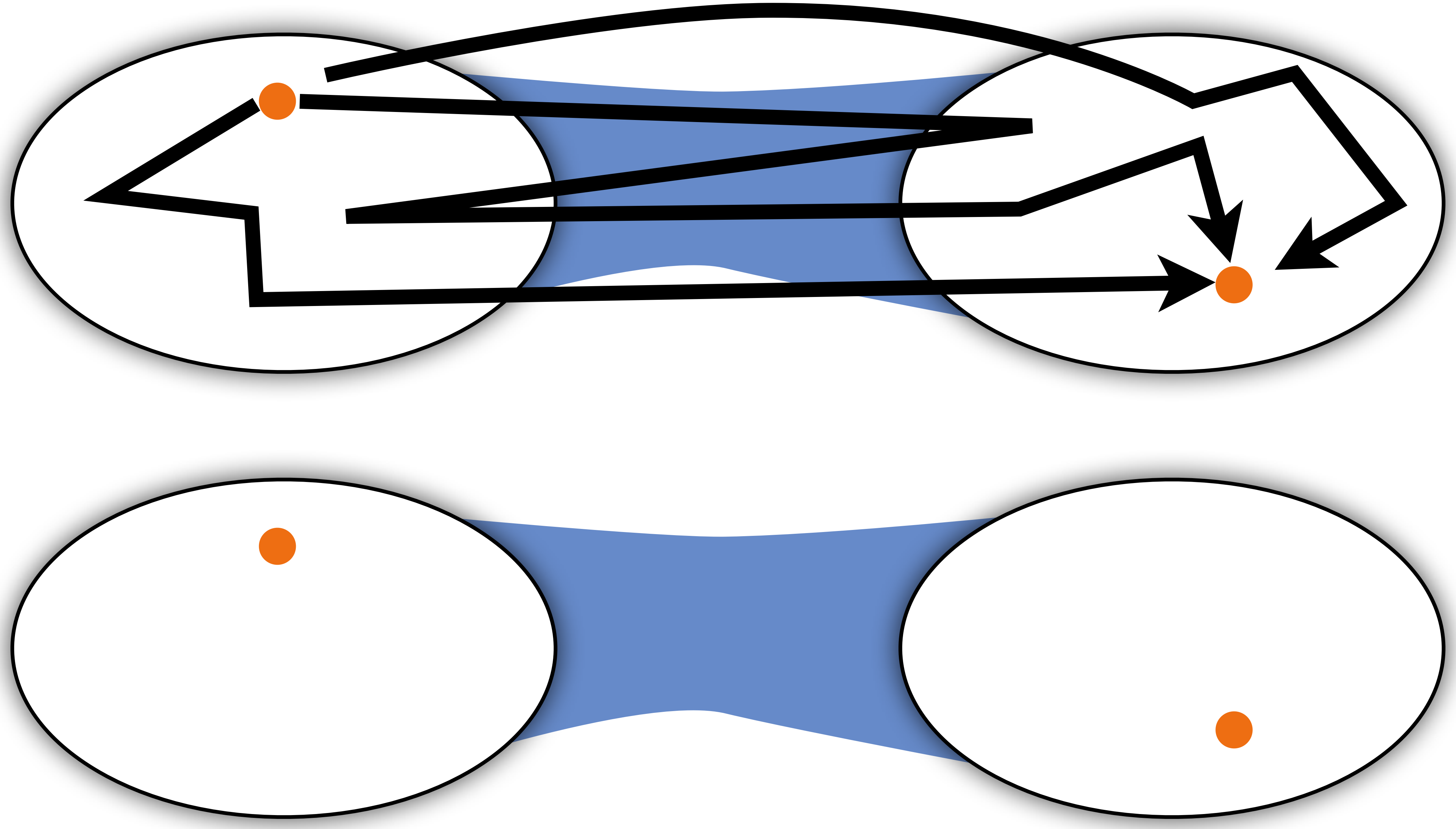




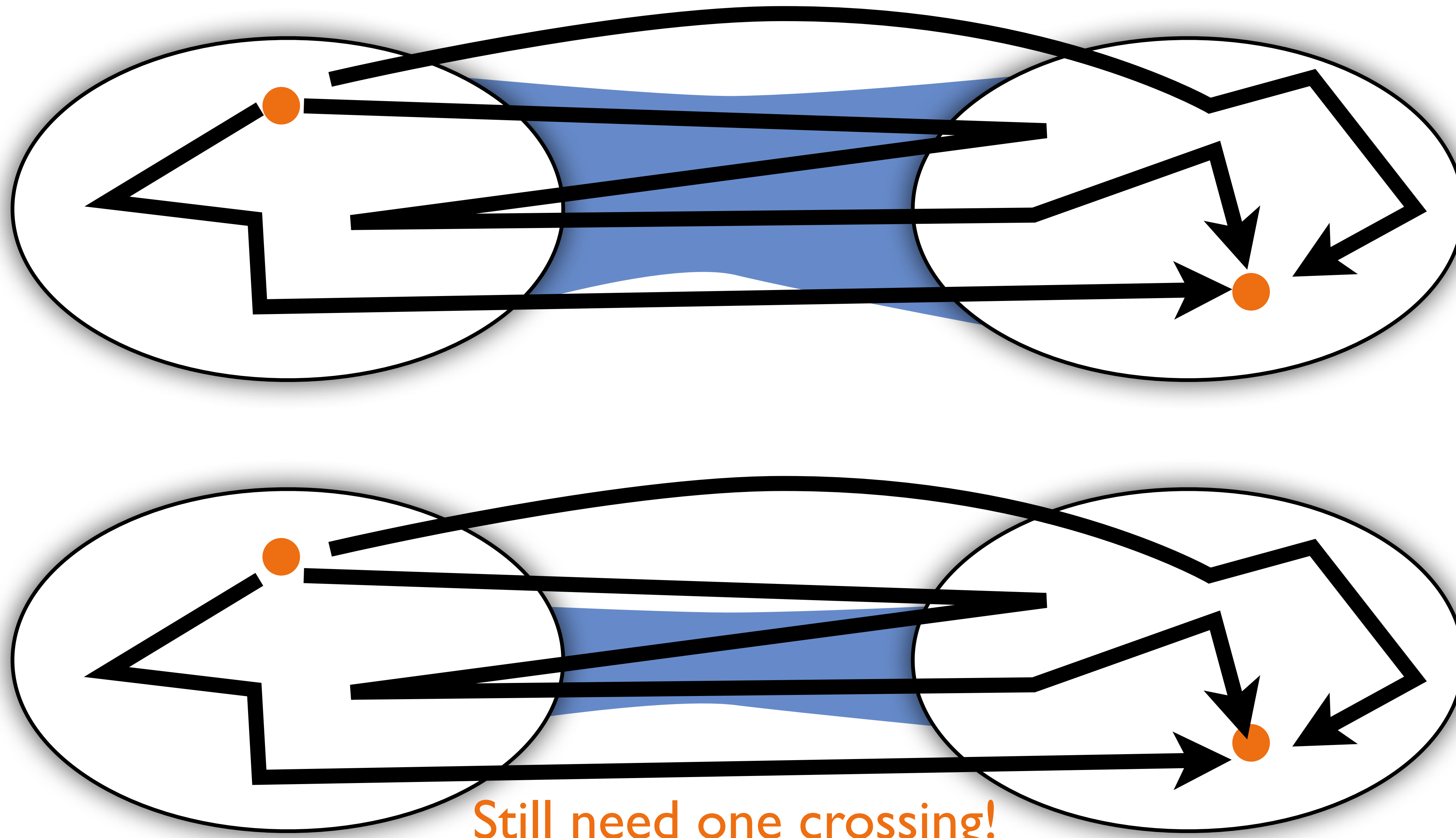
Intuition



Intuition

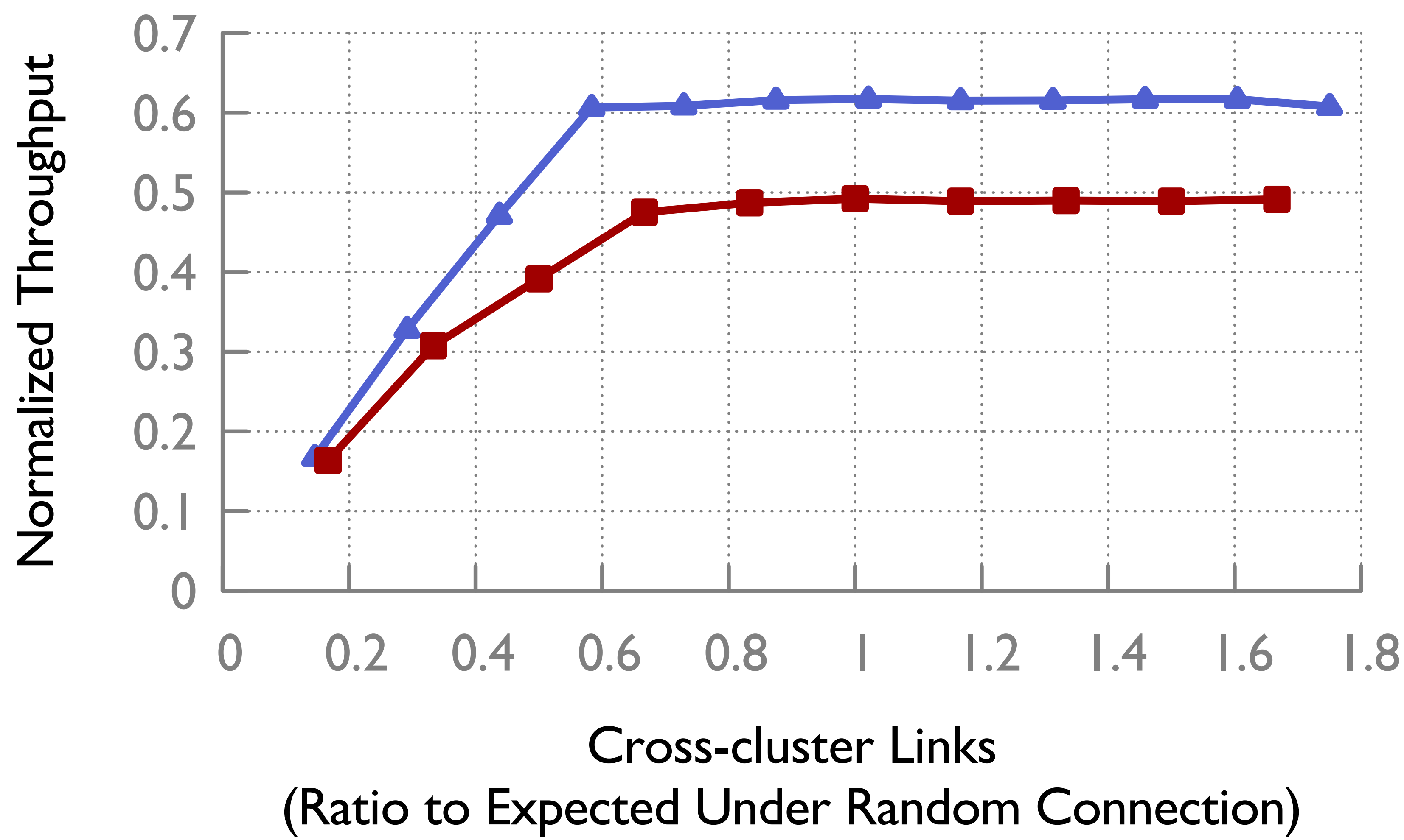


Intuition



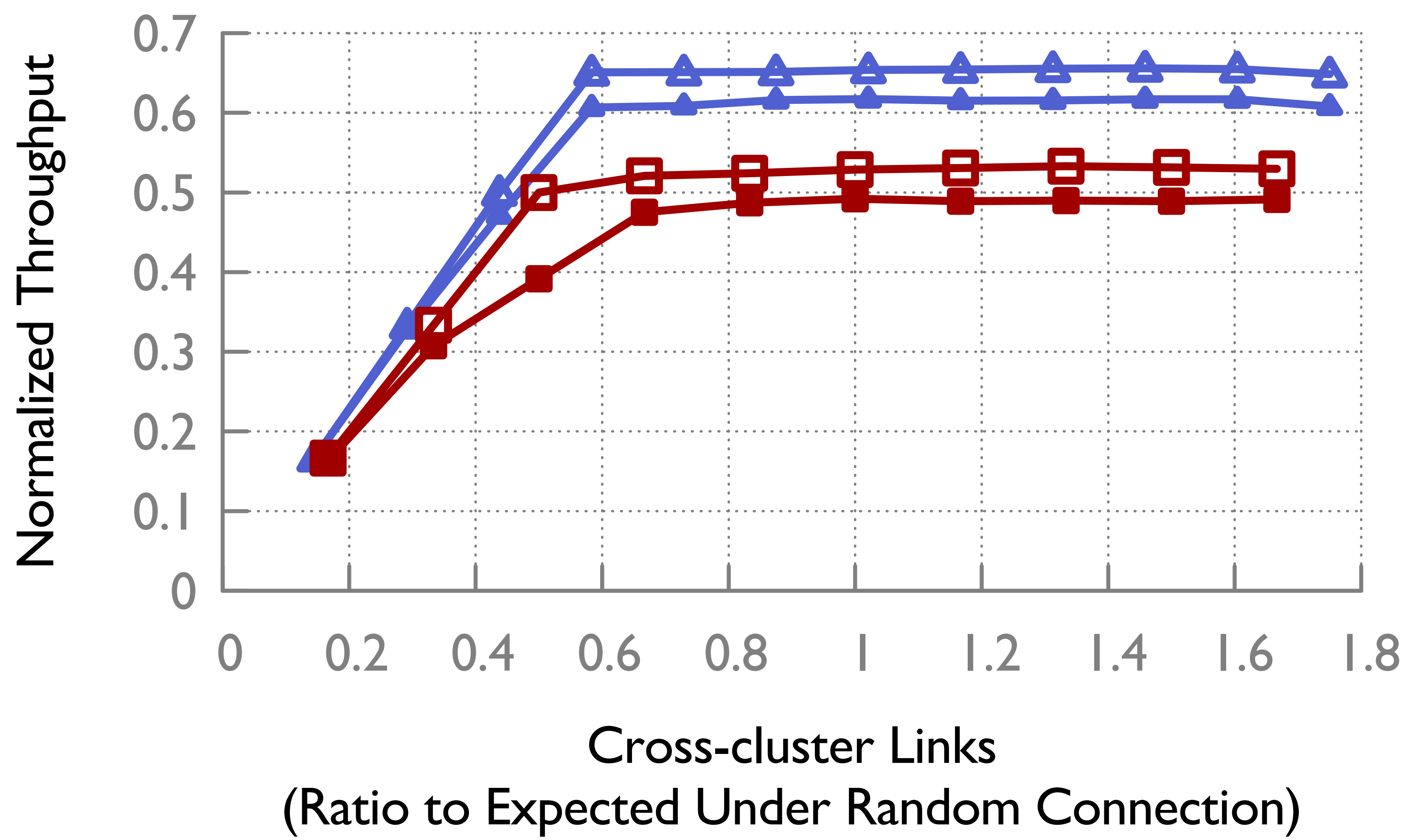
Throughput should drop when less than $\Theta\left(\frac{1}{APL}\right)$ of total capacity crosses the cut!

Explaining throughput



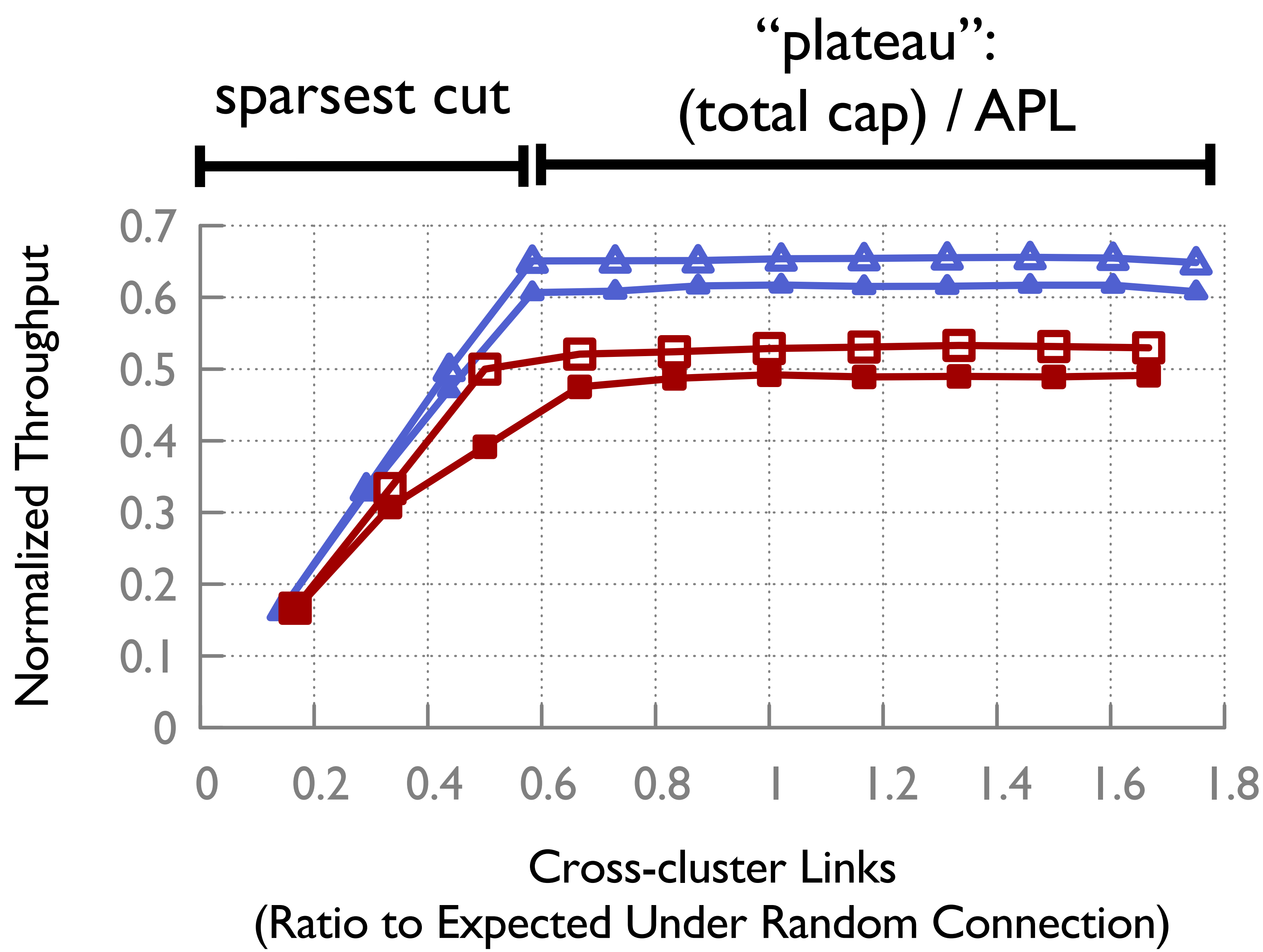
Explaining throughput

Upper bounds...

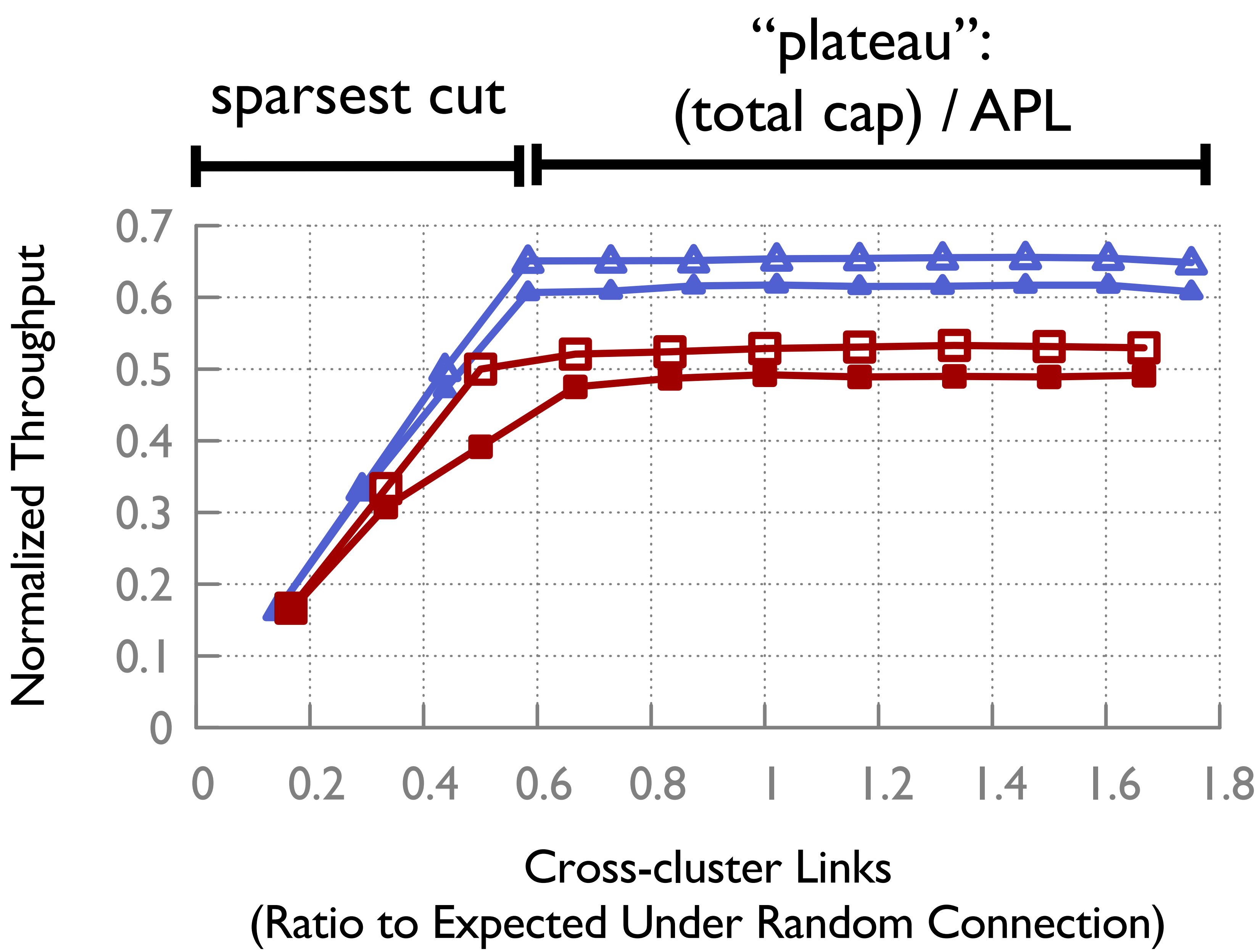


And constant-factor matching lower bounds in special case.

Two regimes of throughput



Two regimes of throughput



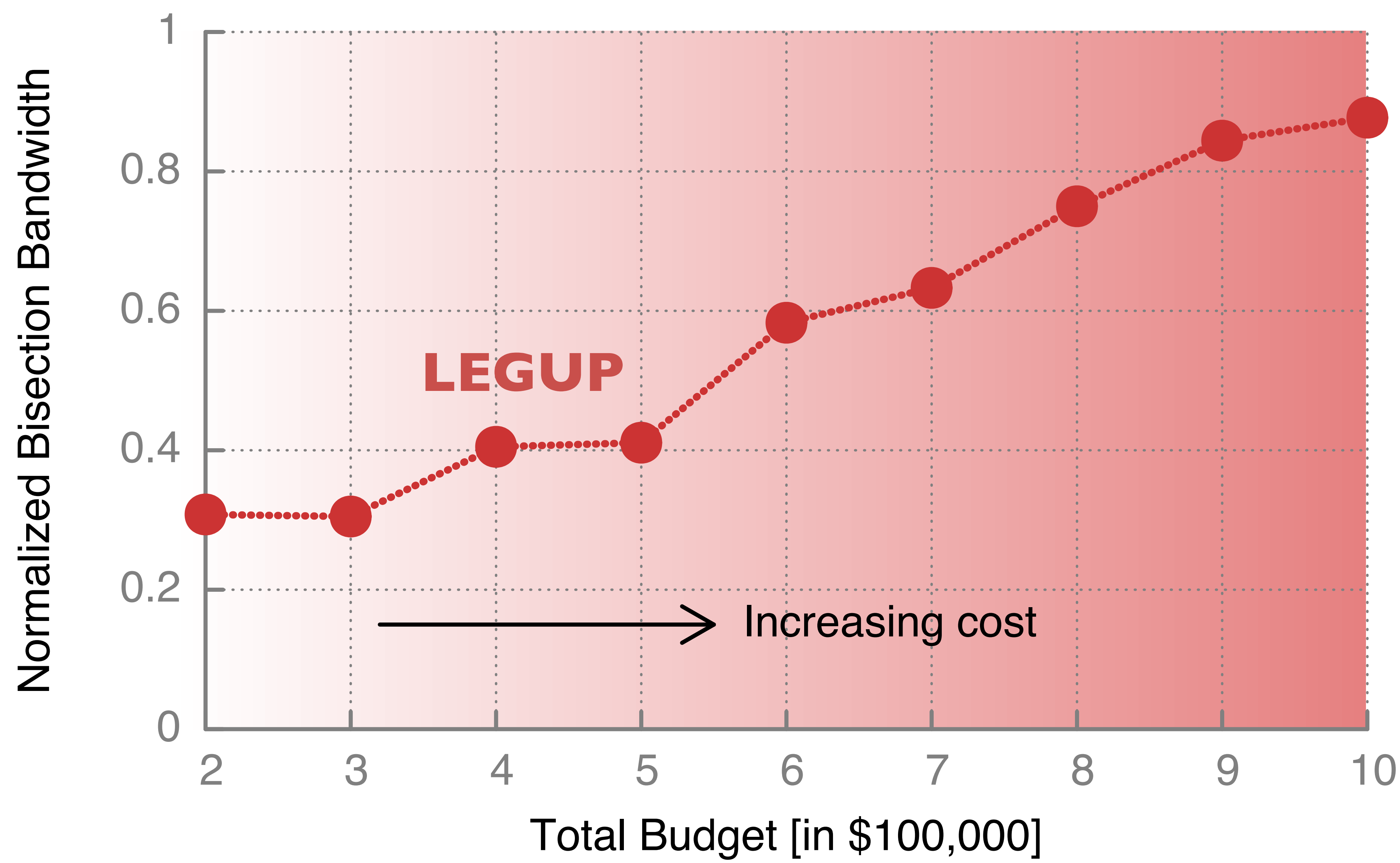
High-capacity switches needn't be clustered

Bisection bandwidth is poor predictor of performance!

Cables can be localized

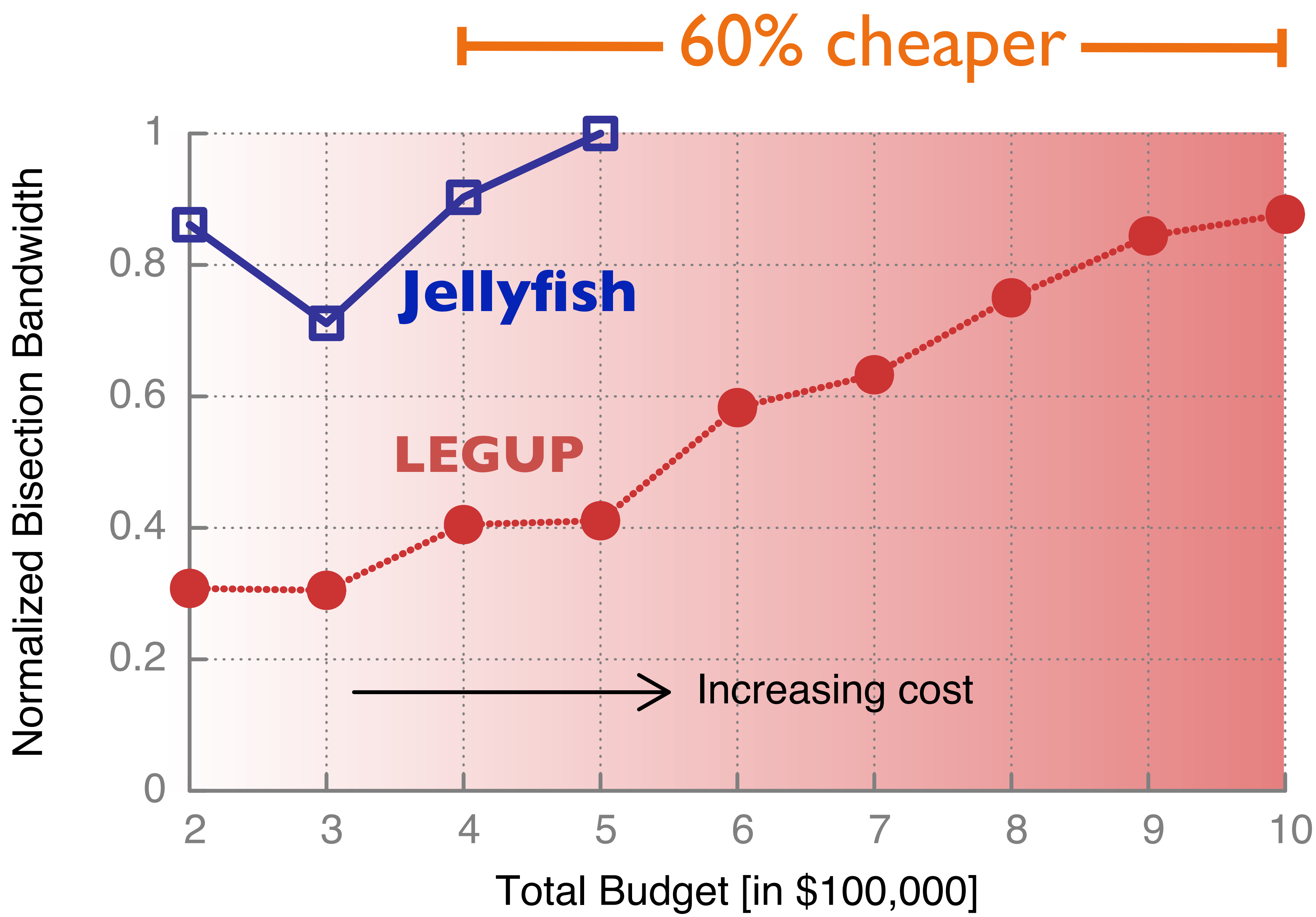
Quantifying Expandability

Quantifying expandability



LEGUP: [Curtis, Keshav, Lopez-Ortiz, CoNEXT'10]

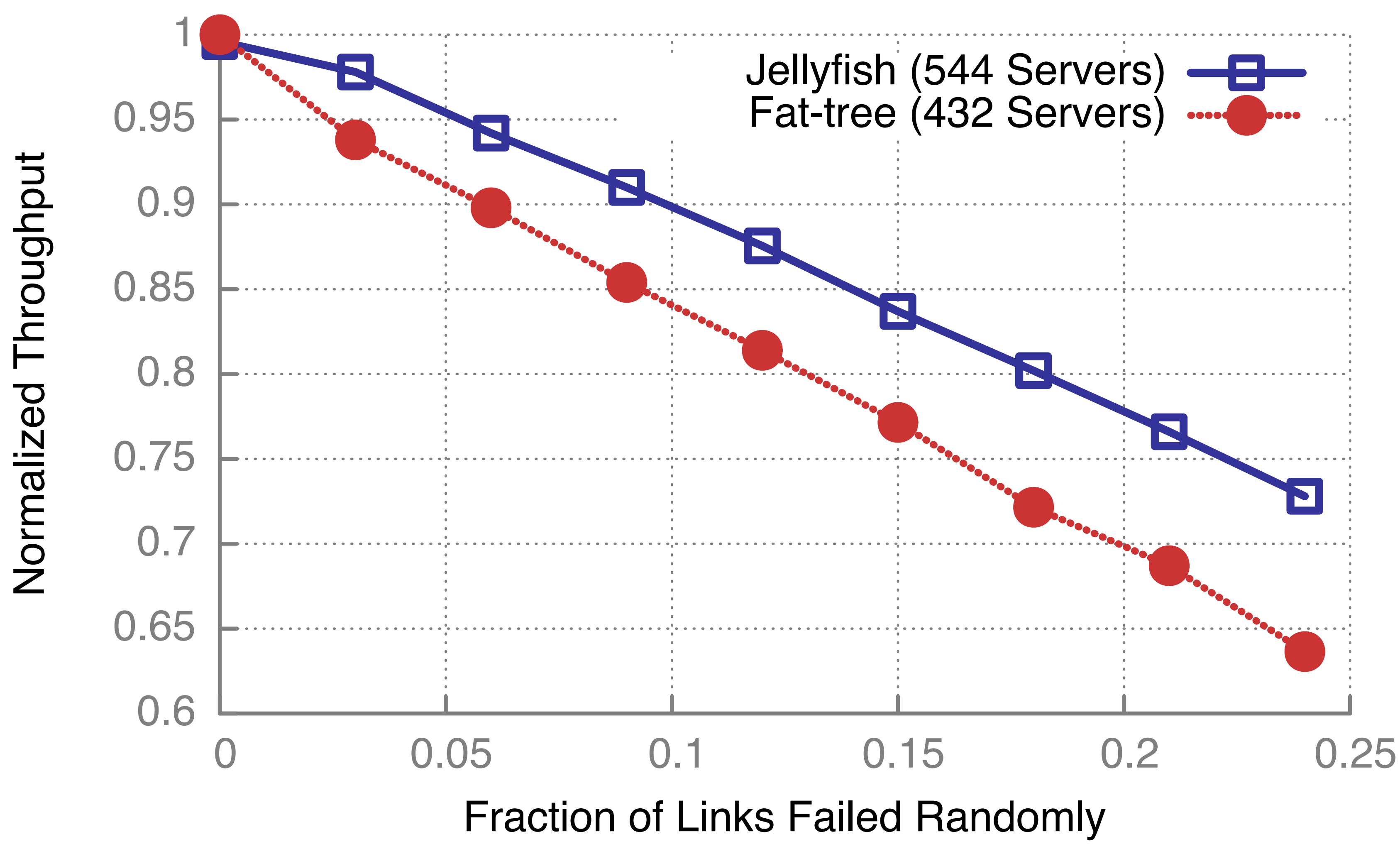
Quantifying expandability



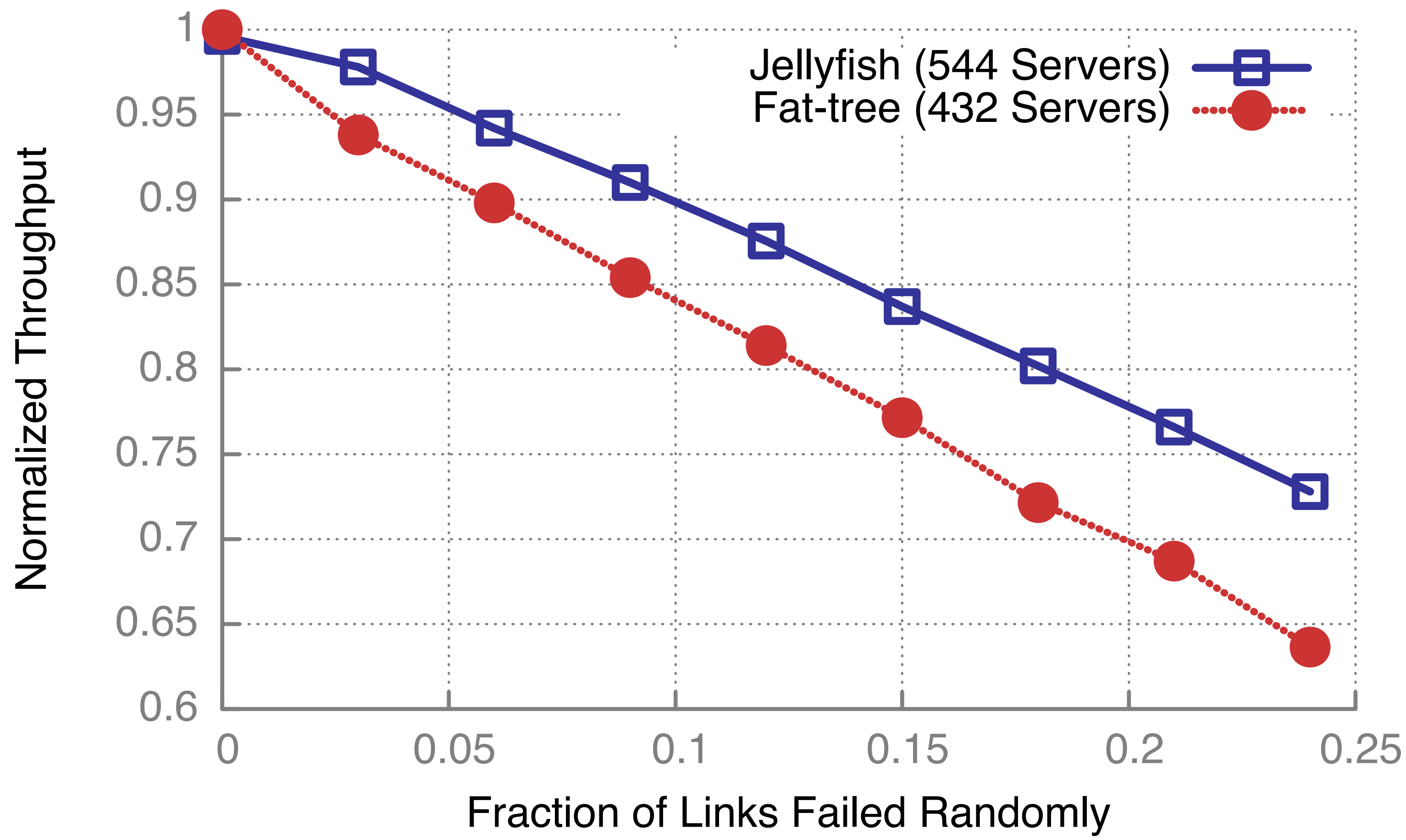
LEGUP: [Curtis, Keshav, Lopez-Ortiz, CoNEXT'10]

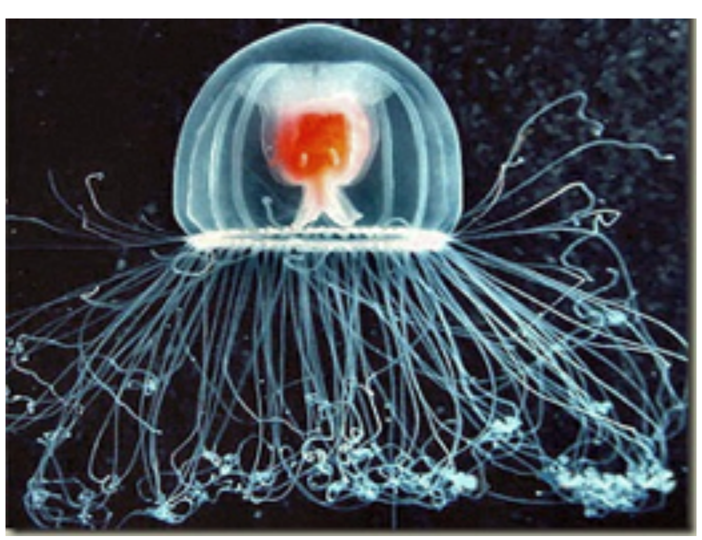
Failure Resilience

Throughput under link failures



Throughput under link failures





Turritopsis Nutricula?

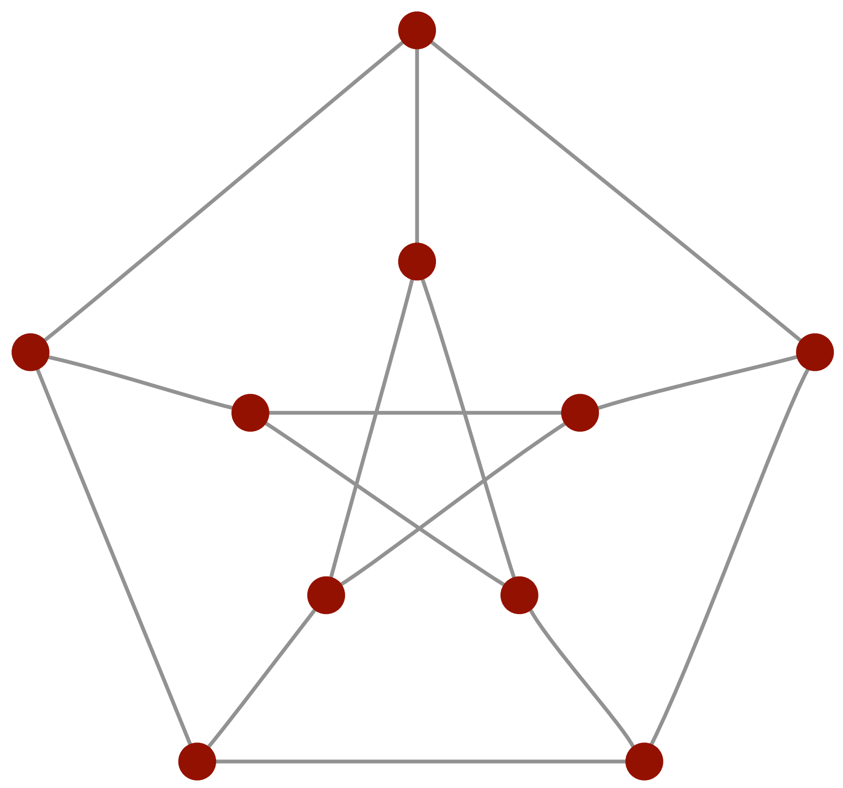
Beyond Random Graphs

Can we do even better?

What is the maximum number of nodes
in any graph with degree ∂ and diameter d?

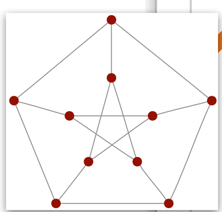
Can we do even better?

What is the maximum number of nodes
in any graph with degree 3 and diameter 2?



Peterson graph

Degree-diameter problem



LARGEST KNOWN (Δ,D)-GRAPHS. June 2010.

Diameter

Degree

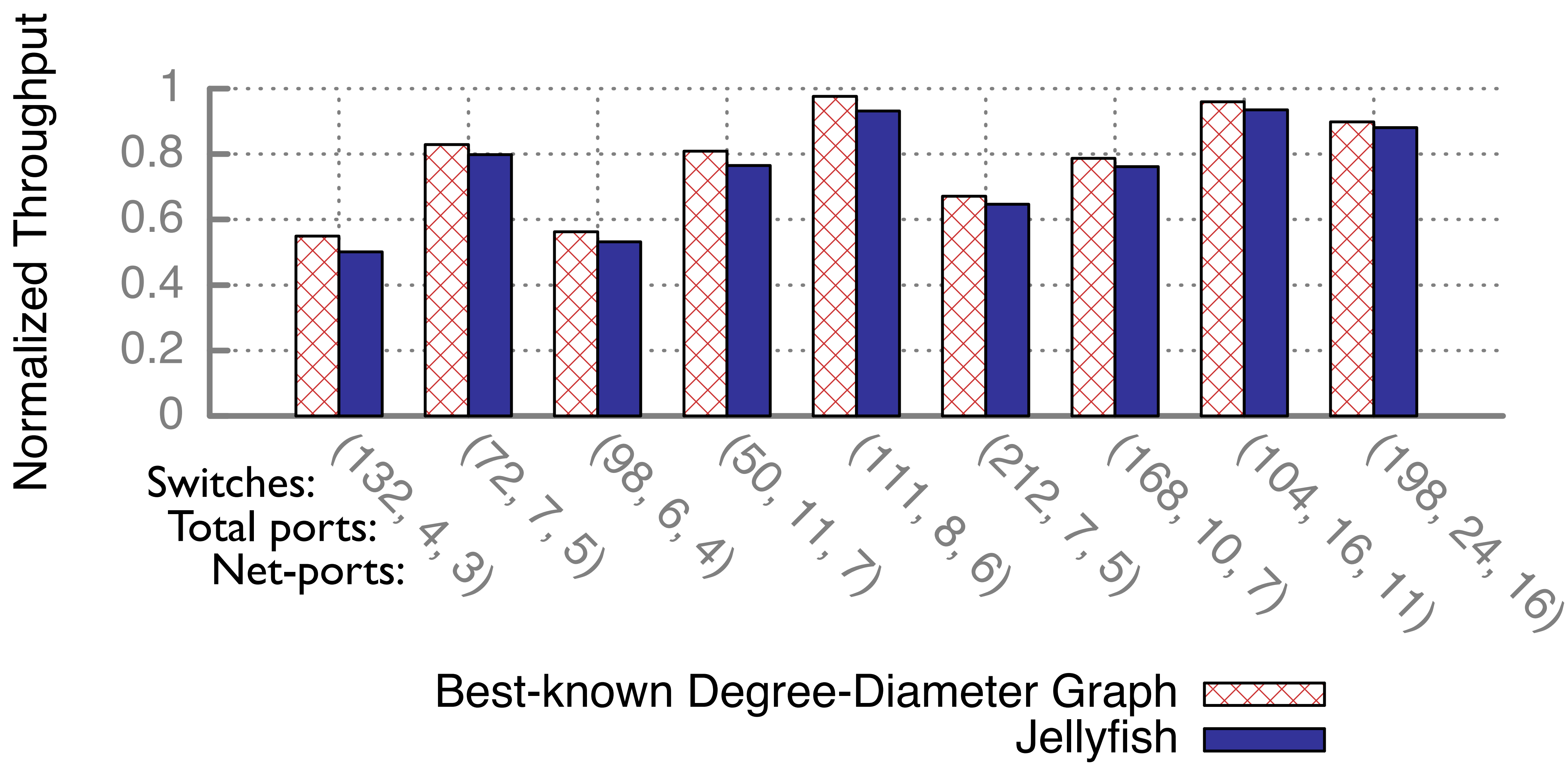
D \ D	2	3	4	5	6	7	8	9	10
3	10	20	38	70	132	196	336	600	1 250
4	15	41	98	364	740	1 320	3 243	7 575	17 703
5	24	72	212	624	2 772	5 516	17 030	53 352	164 720
6	32	111	390	1 404	7 917	19 282	75 157	295 025	1 212 117
7	50	168	672	2 756	11 988	52 768	233 700	1 124 990	5 311 572
8	57	253	1 100	5 060	39 672	130 017	714 010	4 039 704	17 823 532
9	74	585	1 550	8 200	75 893	270 192	1 485 498	10 423 212	31 466 244
10	91	650	2 223	13 140	134 690	561 957	4 019 736	17 304 400	104 058 822
11	104	715	3 200	18 700	156 864	971 028	5 941 864	62 932 488	250 108 668
12	133	786	4 680	29 470	359 772	1 900 464	10 423 212	104 058 822	600 105 100
13	162	851	6 560	39 576	531 440	2 901 404	17 823 532	180 002 472	1 050 104 118
14	183	916	8 200	56 790	816 294	6 200 460	41 894 424	450 103 771	2 050 103 984
15	186	1 215	11 712	74 298	1 417 248	8 079 298	90 001 236	900 207 542	4 149 702 144
16	198	1 600	14 640	132 496	1 771 560	14 882 658	104 518 518	1 400 103 920	7 394 669 856

[Delorme & Comellas: http://www-mat.upc.es/grup_de_grafs/table_g.html/]

Degree-diameter problem

Do the best known degree-diameter graphs also work well for high throughput?

Degree-diameter vs. Jellyfish

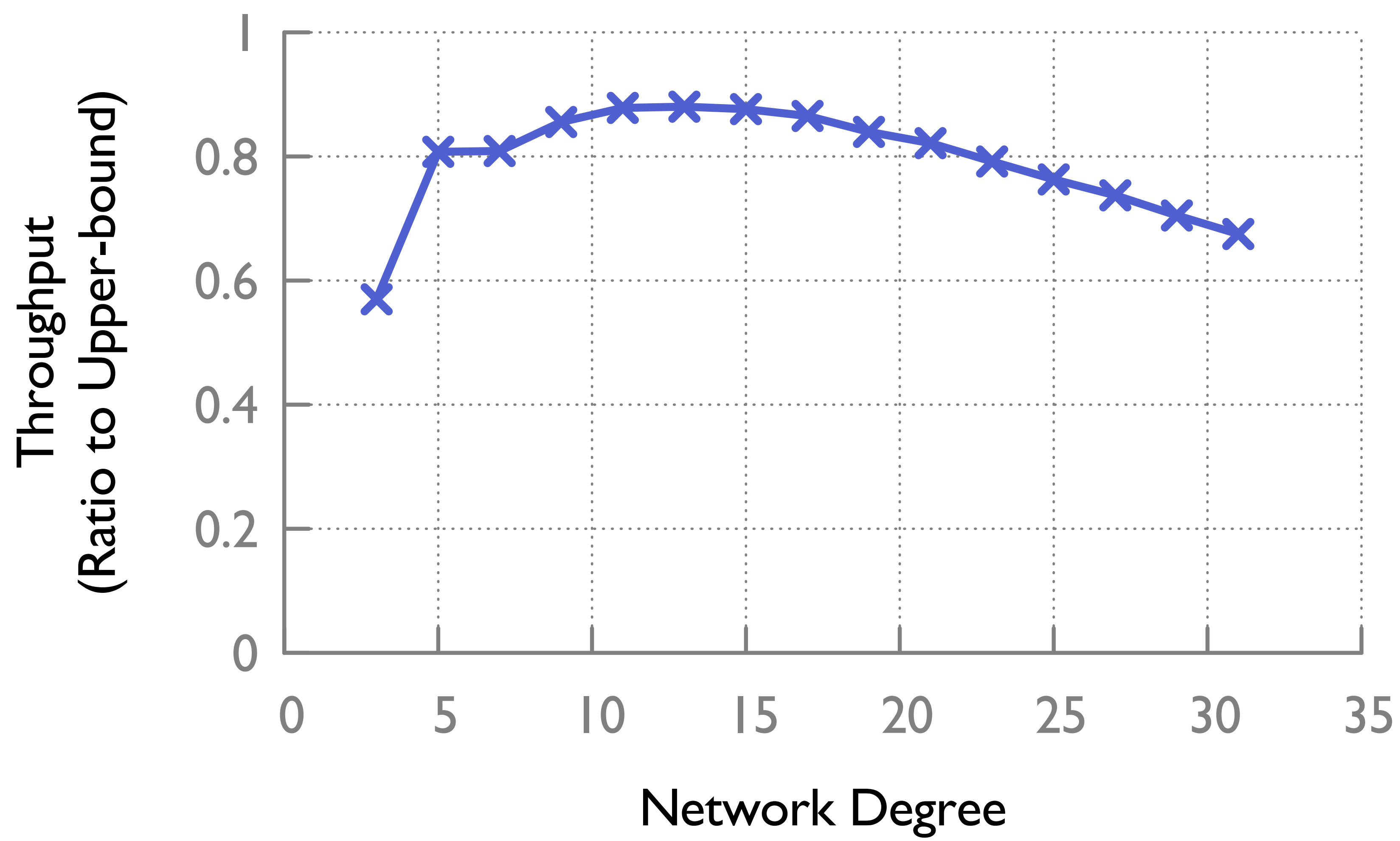


D-D graphs ***do*** have high throughput

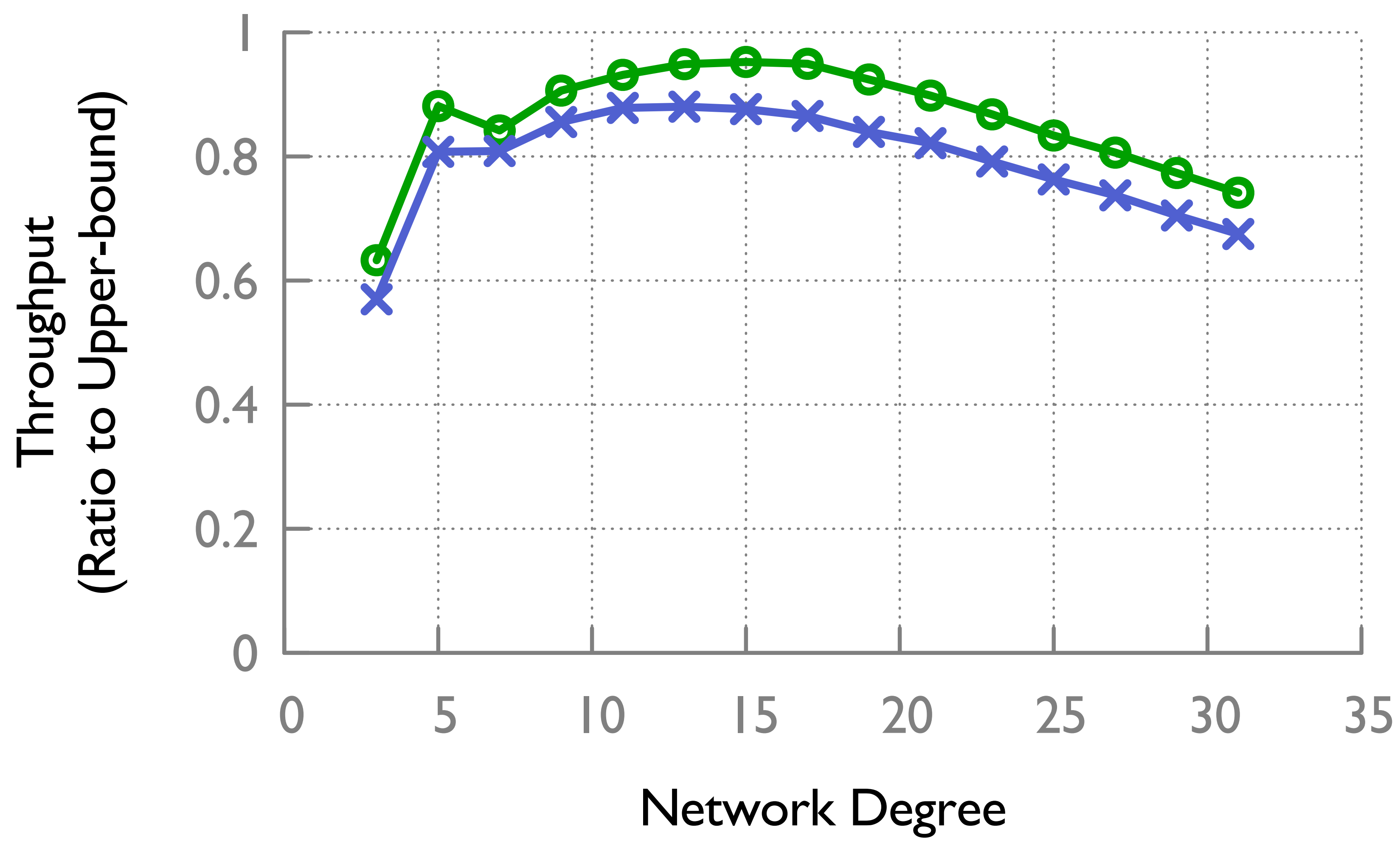
Jellyfish within 9%!

Random graphs vs. upper bound for fixed size and increasing degree

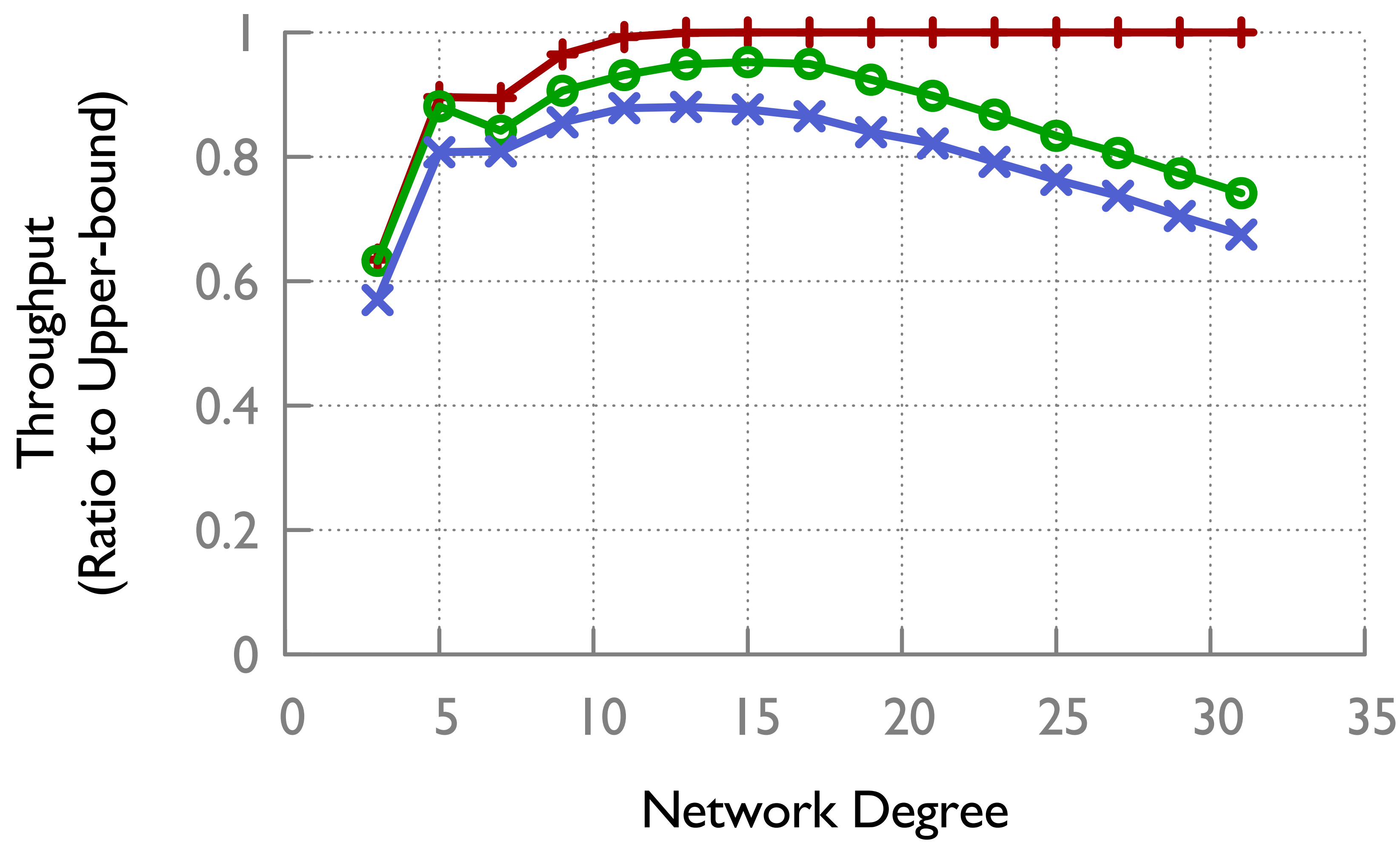
Random graphs vs. upper bound



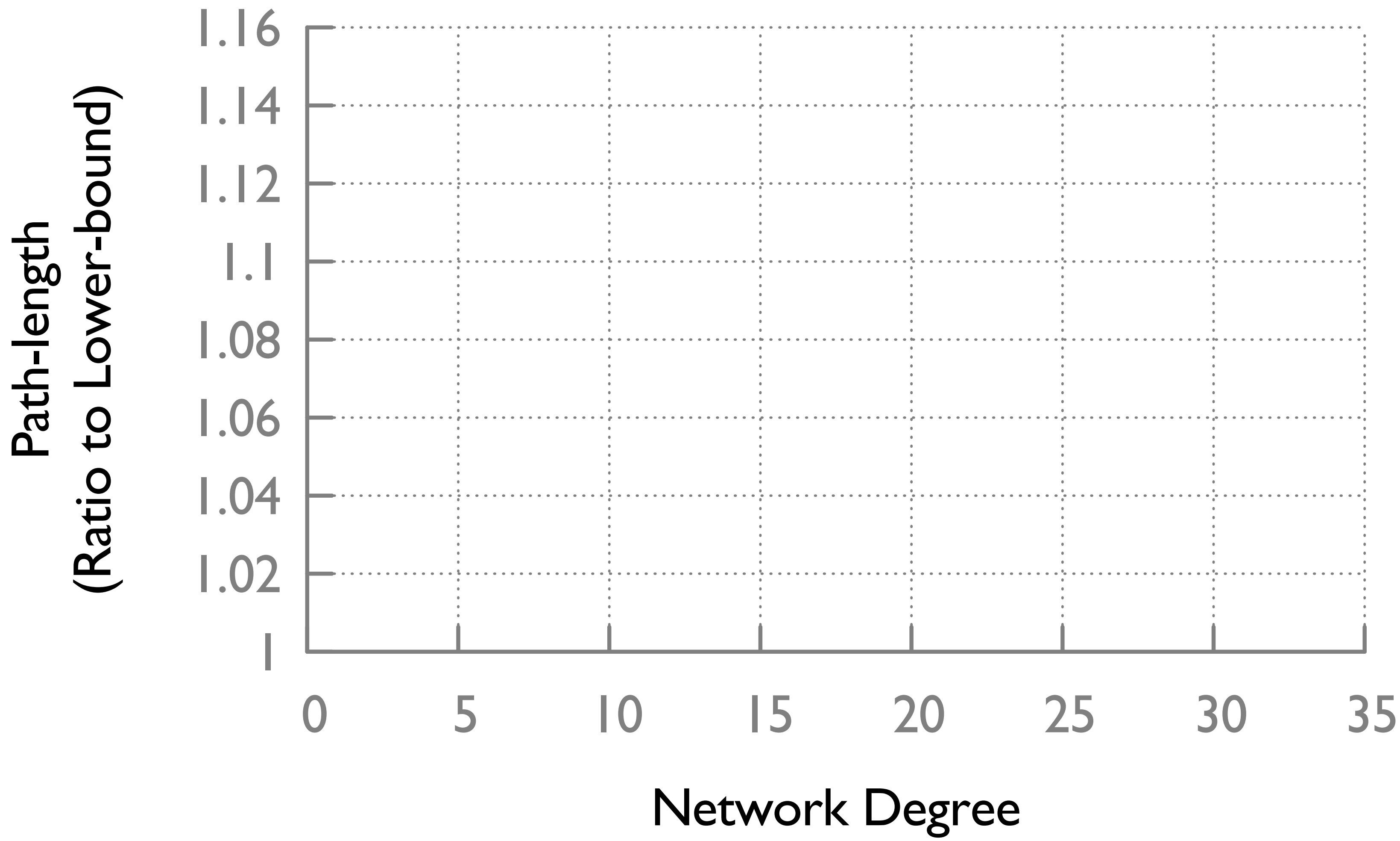
Random graphs vs. upper bound



Random graphs vs. upper bound



Random graphs vs. upper bound



Random graphs vs. upper bound