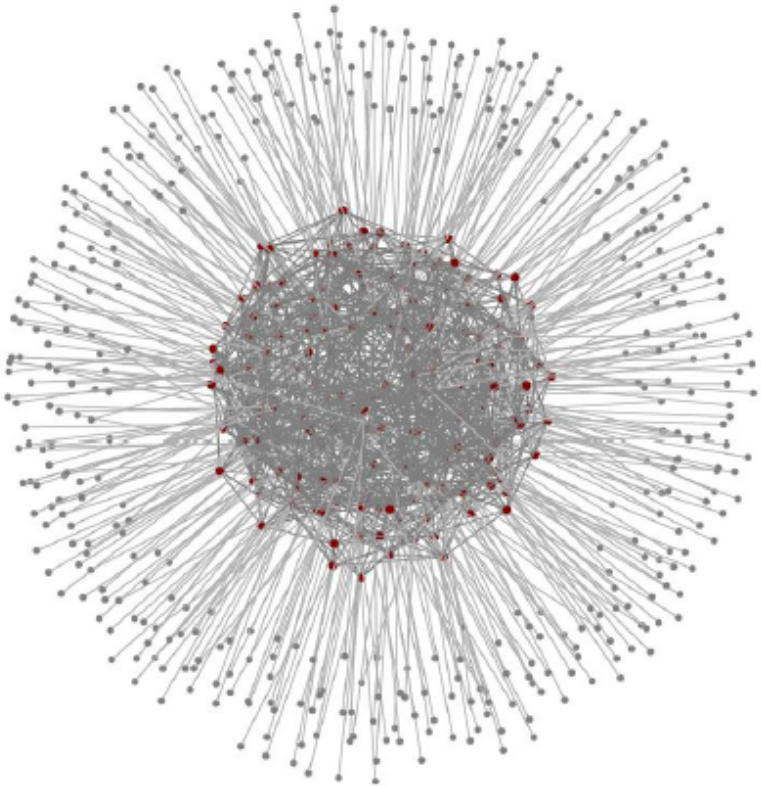
Jellyfish: Networking Data Centers Randomly

Ankit Singla, Chi-Yao Hong, Lucian Popa,
Brighten Godfrey

Presented by Rashmi K. Vinayak
11/16/2015

(Many of the slides sourced from authors' presentations at NSDI '12 & DIMACS workshop '11)

How cool!



Jellyfish random graph

432 servers, 180 switches, degree 12

Jellyfish

Arctapodema

Two goals

High throughput

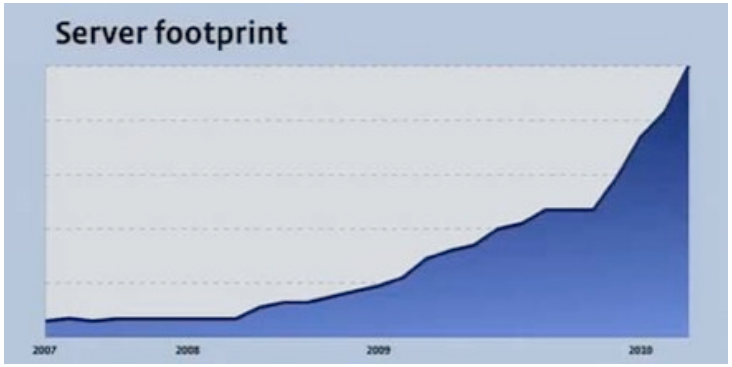
Eliminate bottlenecks
Agile placement of VMs

Incremental expandability

Easily add/replace
servers & switches

Incremental expansion

Facebook "adding capacity on a daily basis"



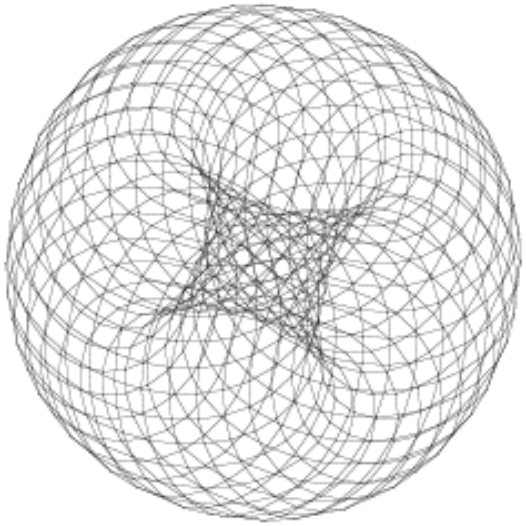
2007 08 09 10

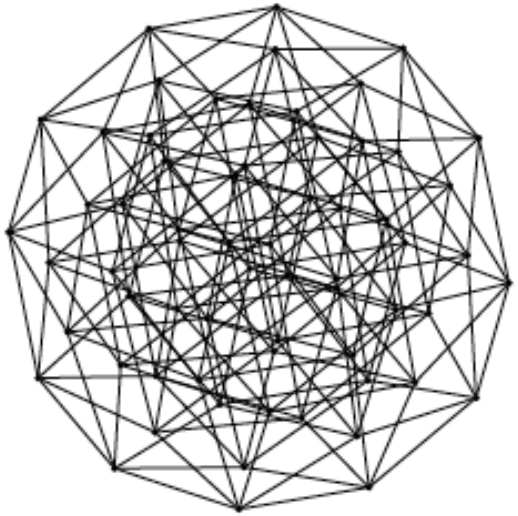
Commercial products

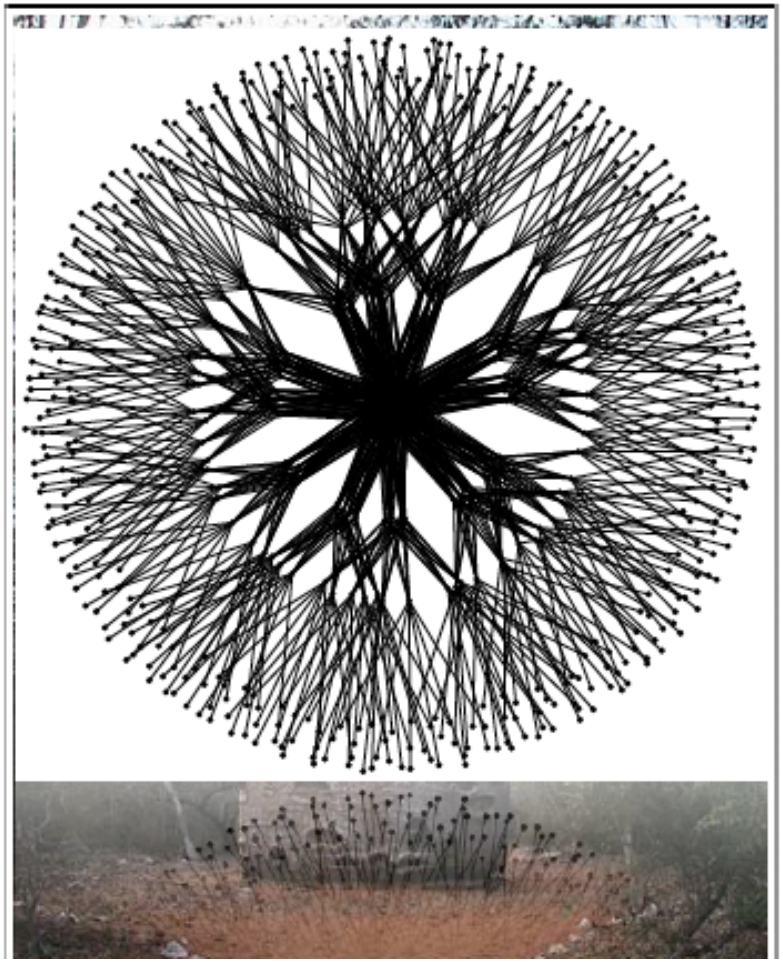
- SGI Ice Cube ("Expandable Modular Data Center")
- HP EcoPod ("Pay-as-you-grow")

You can add servers, but what about the network?

Today's structured networks







Structure constrains expansion

Coarse design points

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

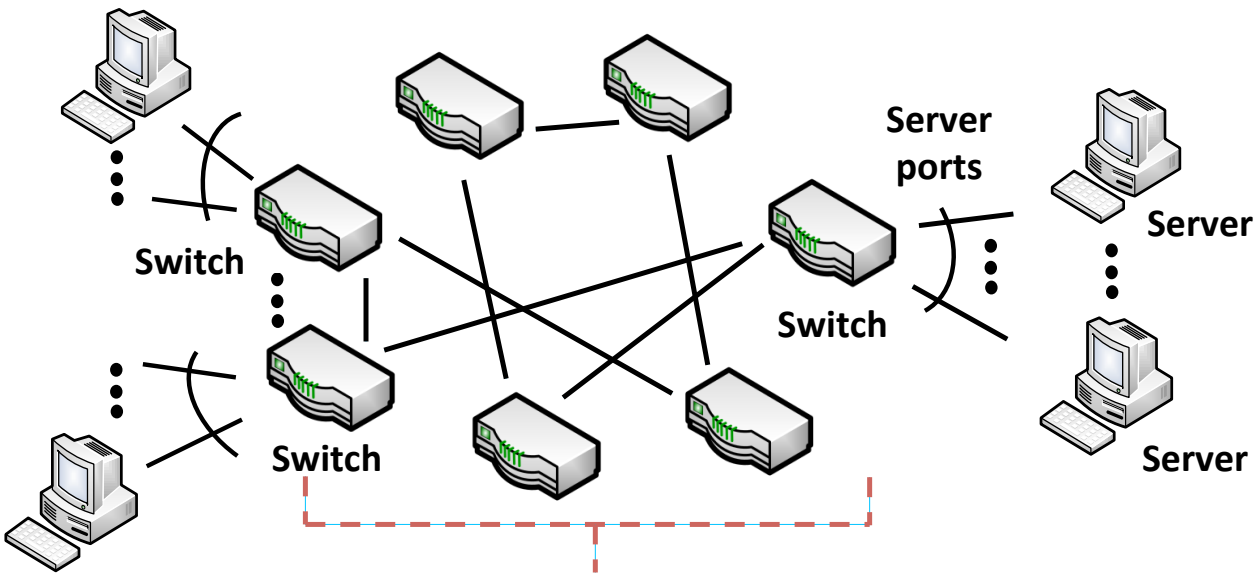
Fat trees by the numbers:

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

Unclear how to maintain structure incrementally

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

Jellyfish: The Topology



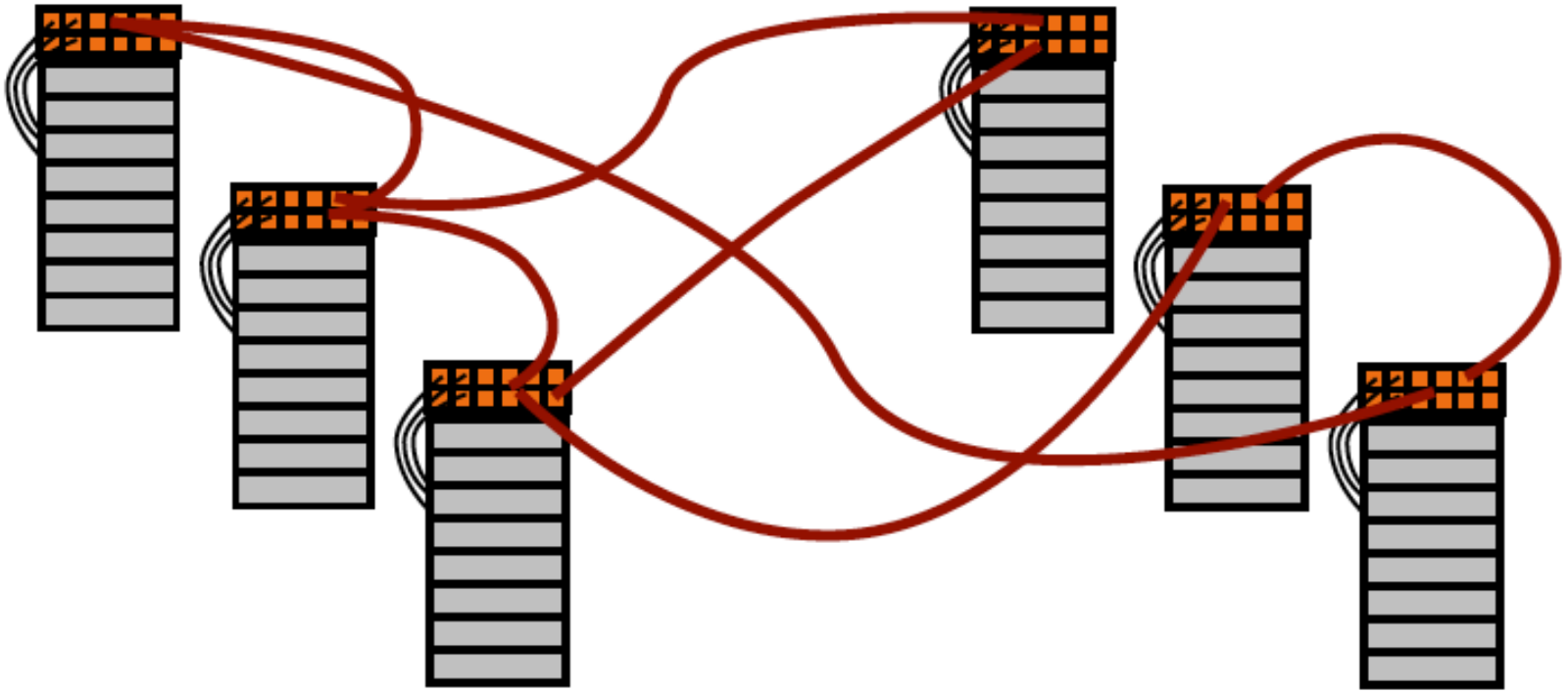
Random Regular Graph

Uniform randomly selected from all regular graphs

Each node has the same degree

Switches are nodes

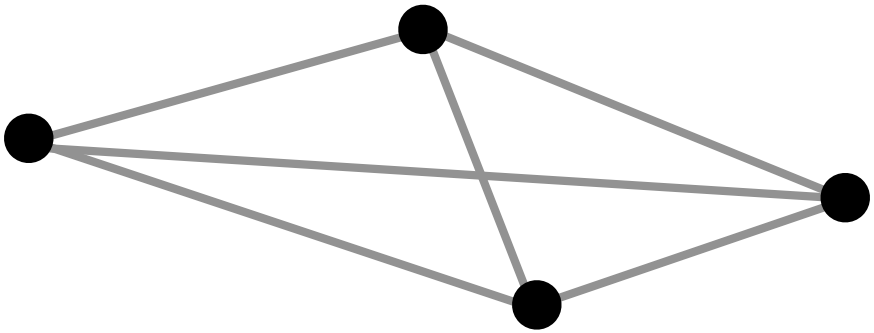
Jellyfish: The Topology



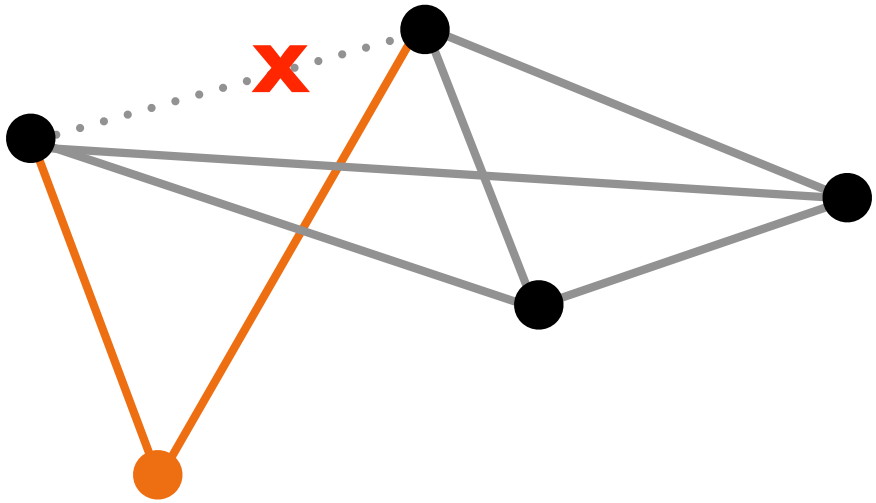
Servers connected to top-of-rack switch

Switches form uniform-random interconnections

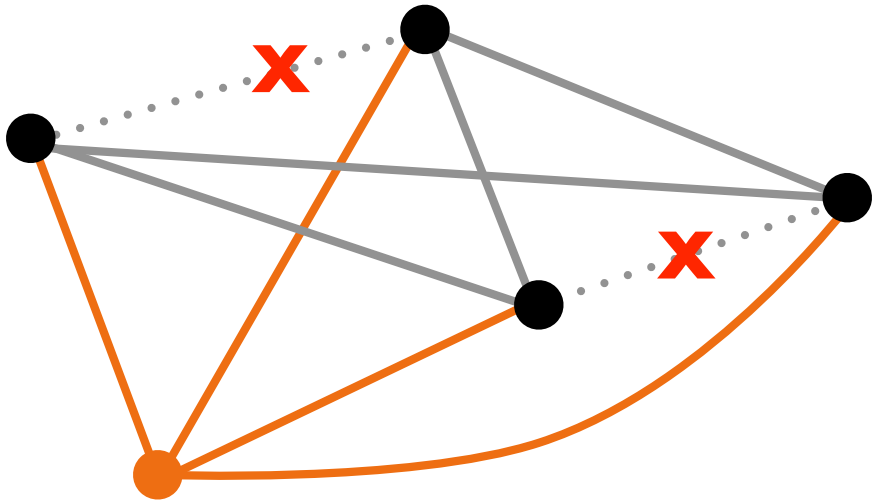
Building Jellyfish



Building Jellyfish



Building Jellyfish



Same procedure for initial construction
and incremental expansion

Highly flexible

– few random swaps to incorporate additional components (both server racks and switches)

– supports heterogeneity naturally

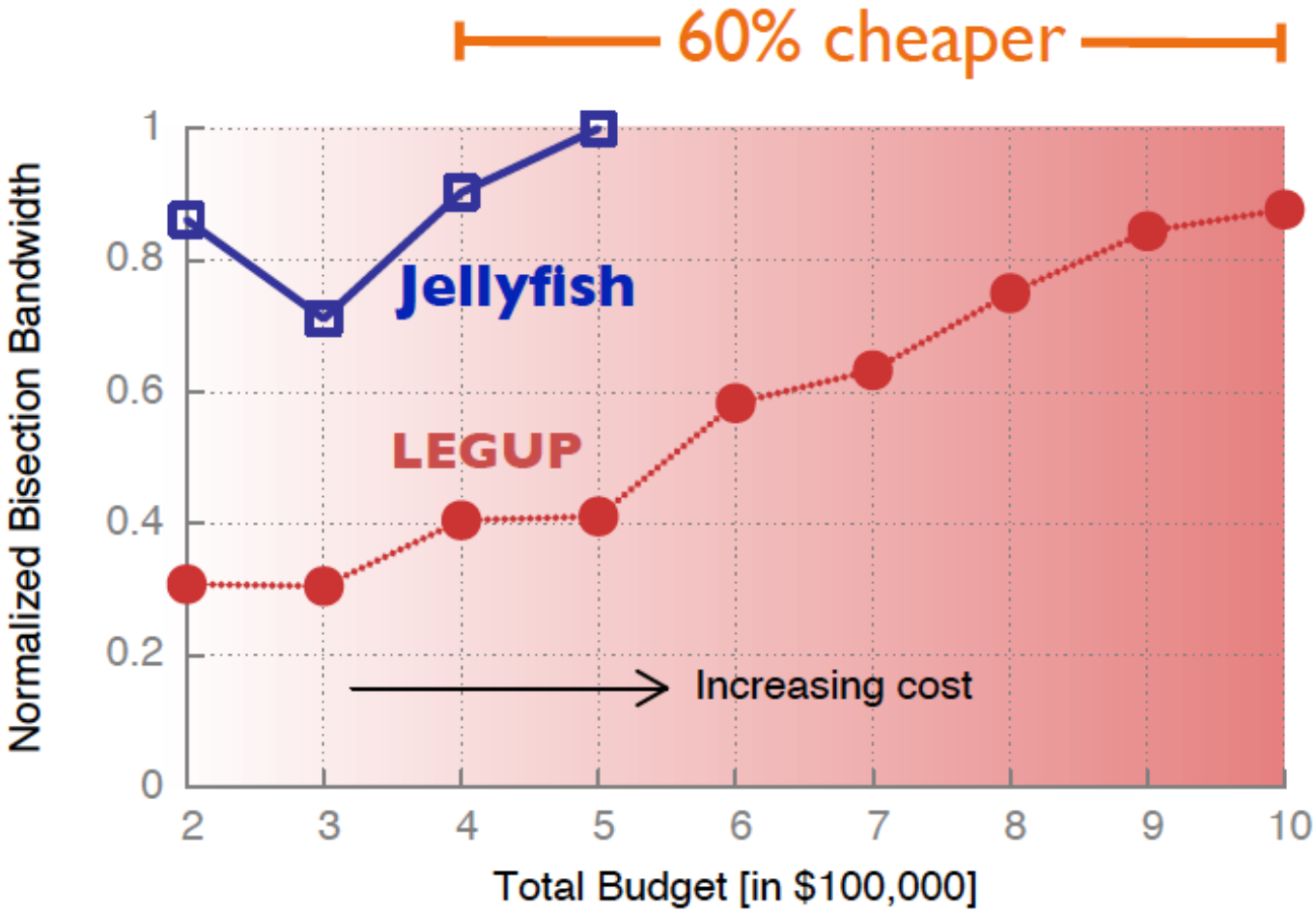
- newer network elements can have higher port counts

– allows construction of arbitrary sized networks

- almost continuous design space
- can add one rack or a switch at a time

Great for incremental expansion

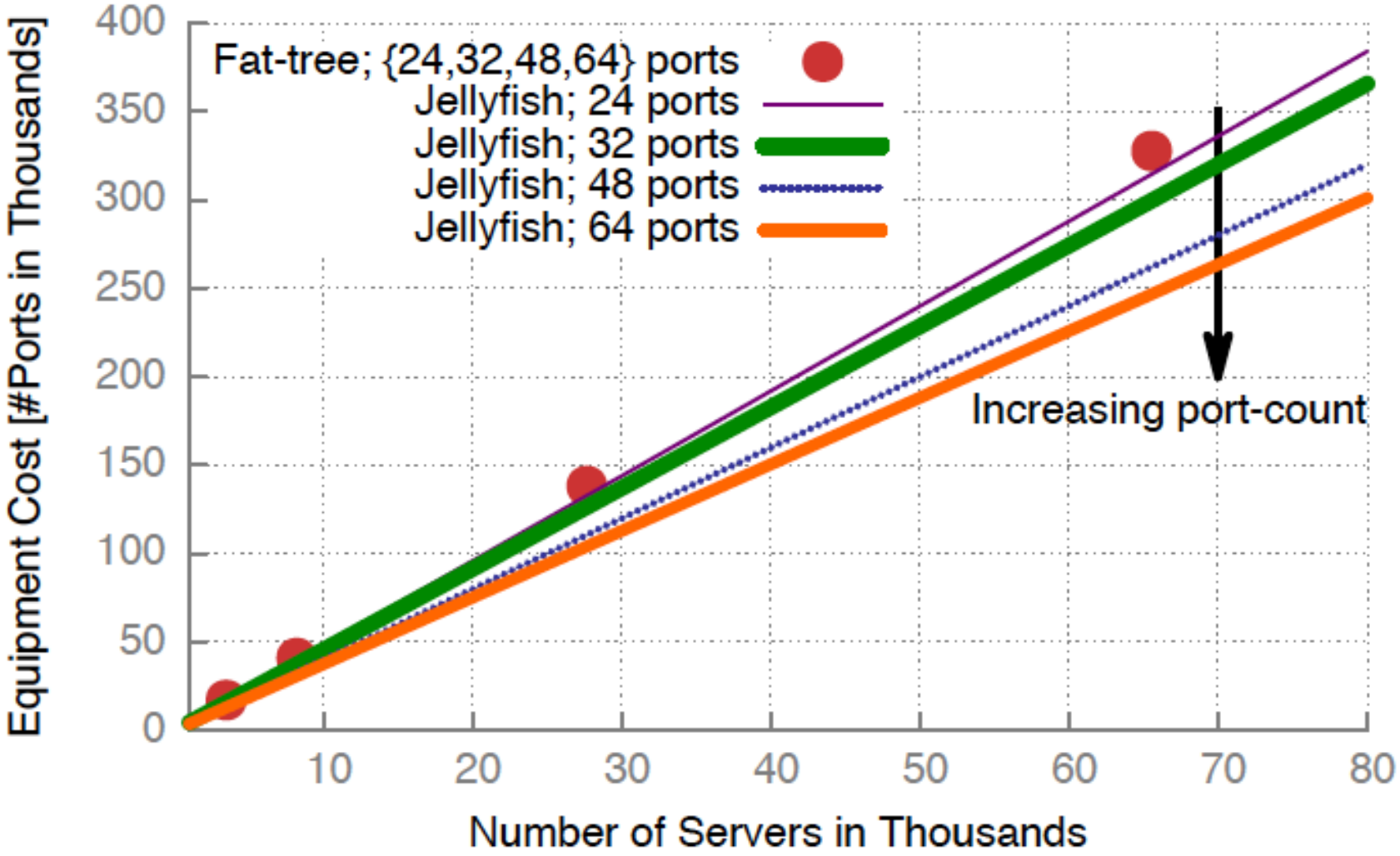
Quantifying expandability



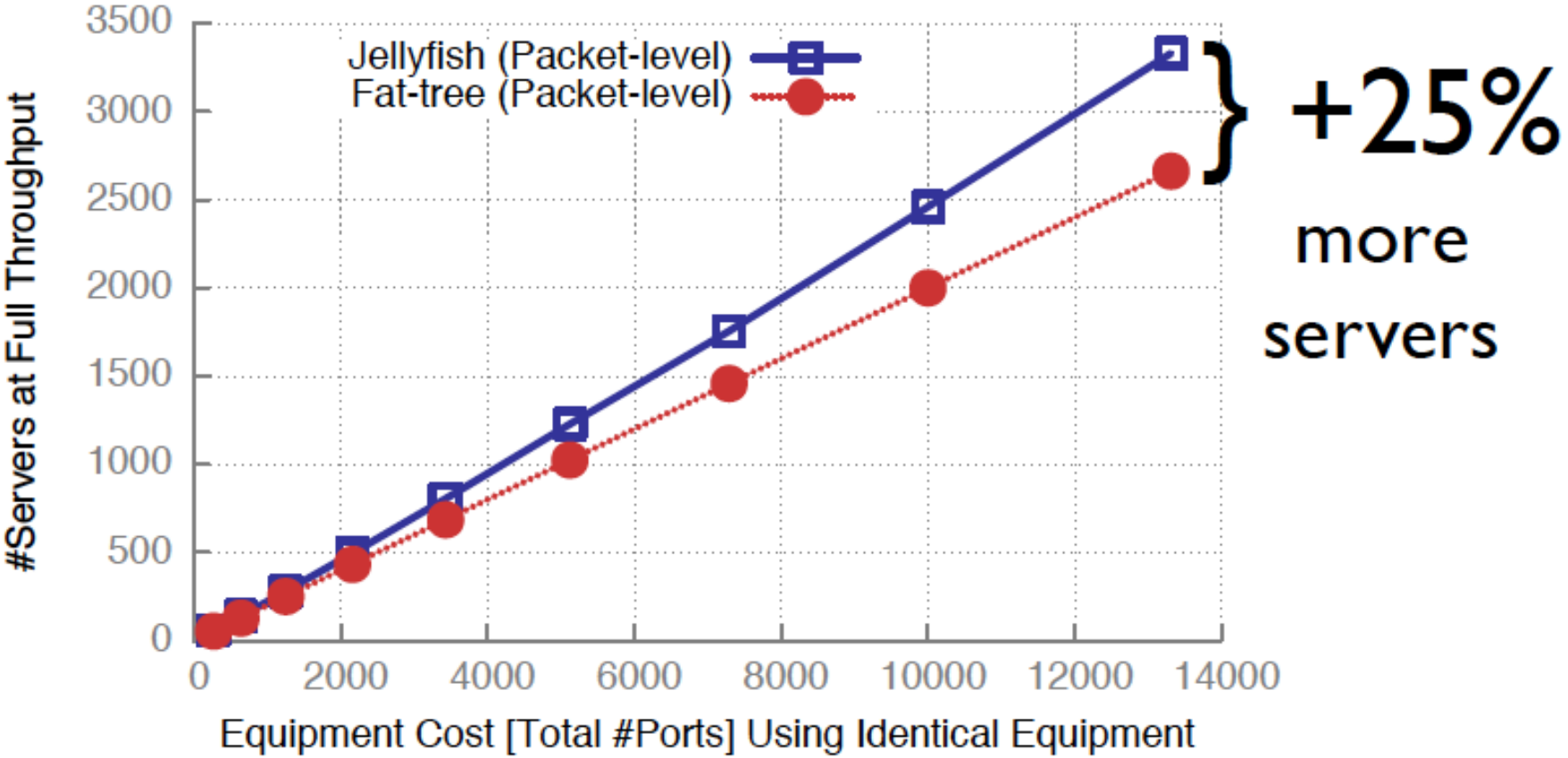
- Initial 2 stages adds both servers and switches
- Then on only switches added
- *LEGUP leaves some ports free for incremental expansion*

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

Cost of building full bisection BW network



Throughput: Jellyfish vs. fat tree



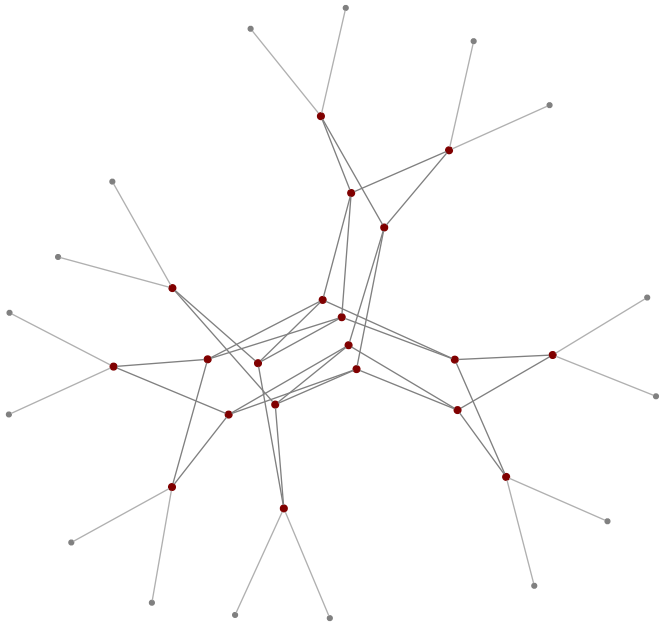
Intuition

if we fully utilize all available capacity ...

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

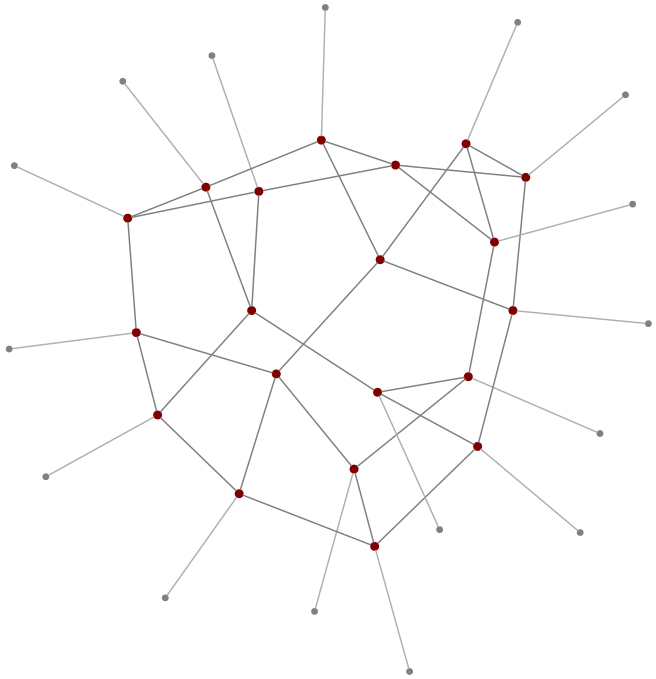
Mission:
minimize average path length

Example



Fat tree

16 servers, 20 switches, degree 4

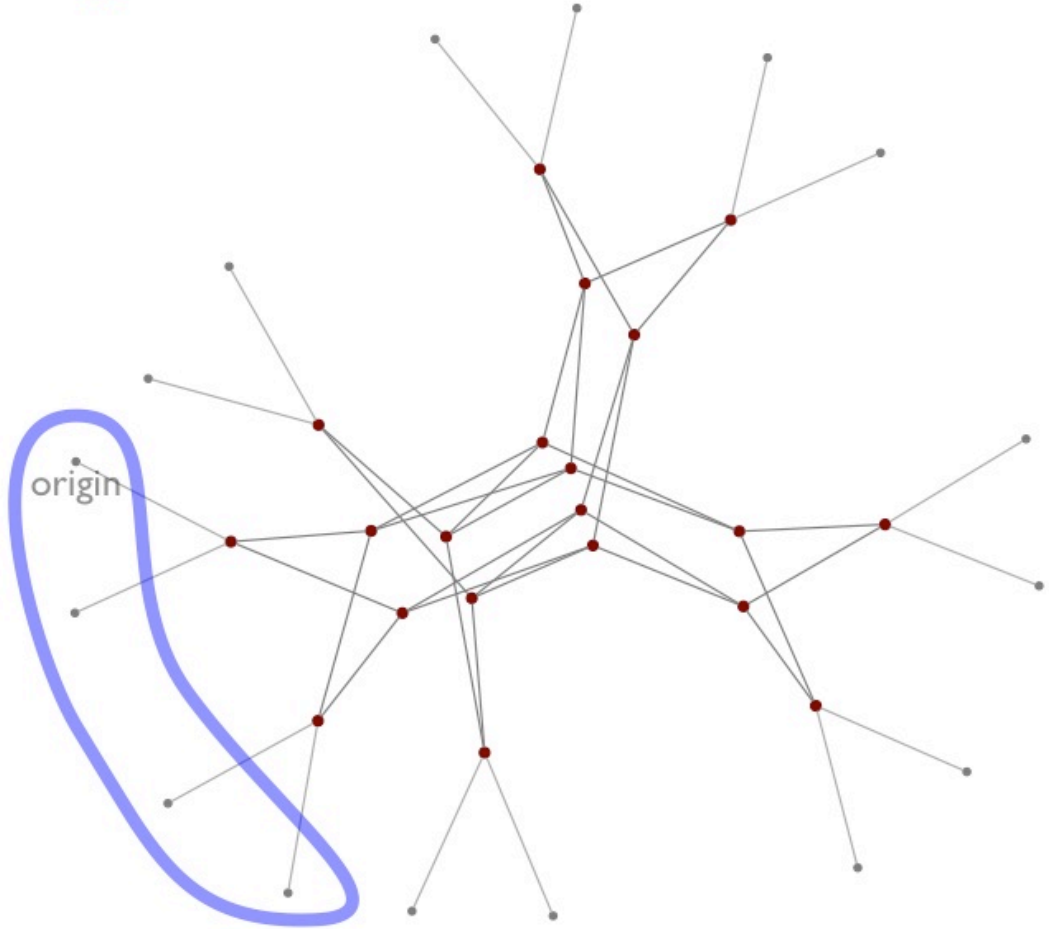


Jellyfish random graph

16 servers, 20 switches, degree 4

A more manageable example, actually...

Example: Fat Tree

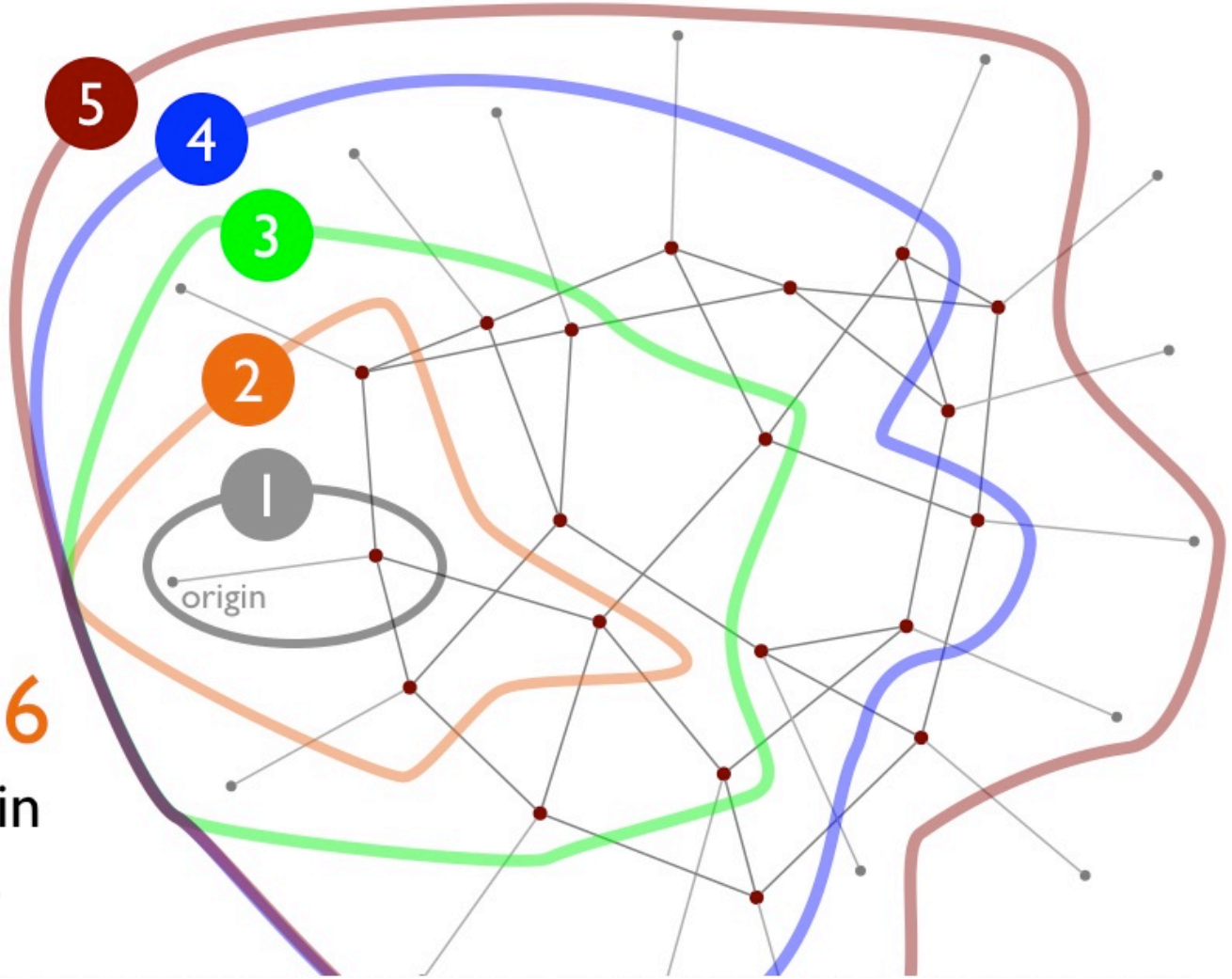


4 of 16

reachable

in < 6 hops

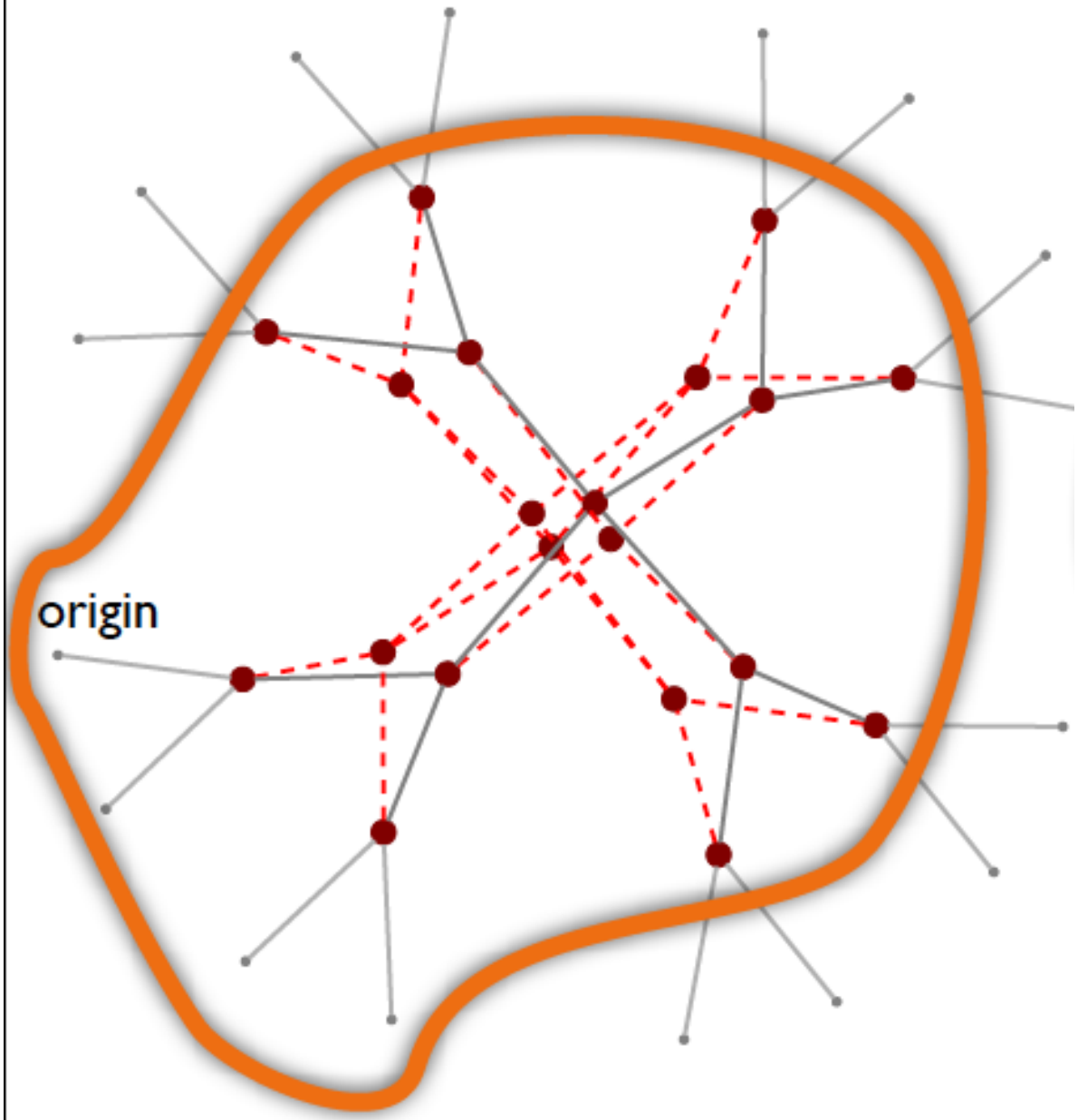
Example: Jellyfish



13 of 16

reachable in

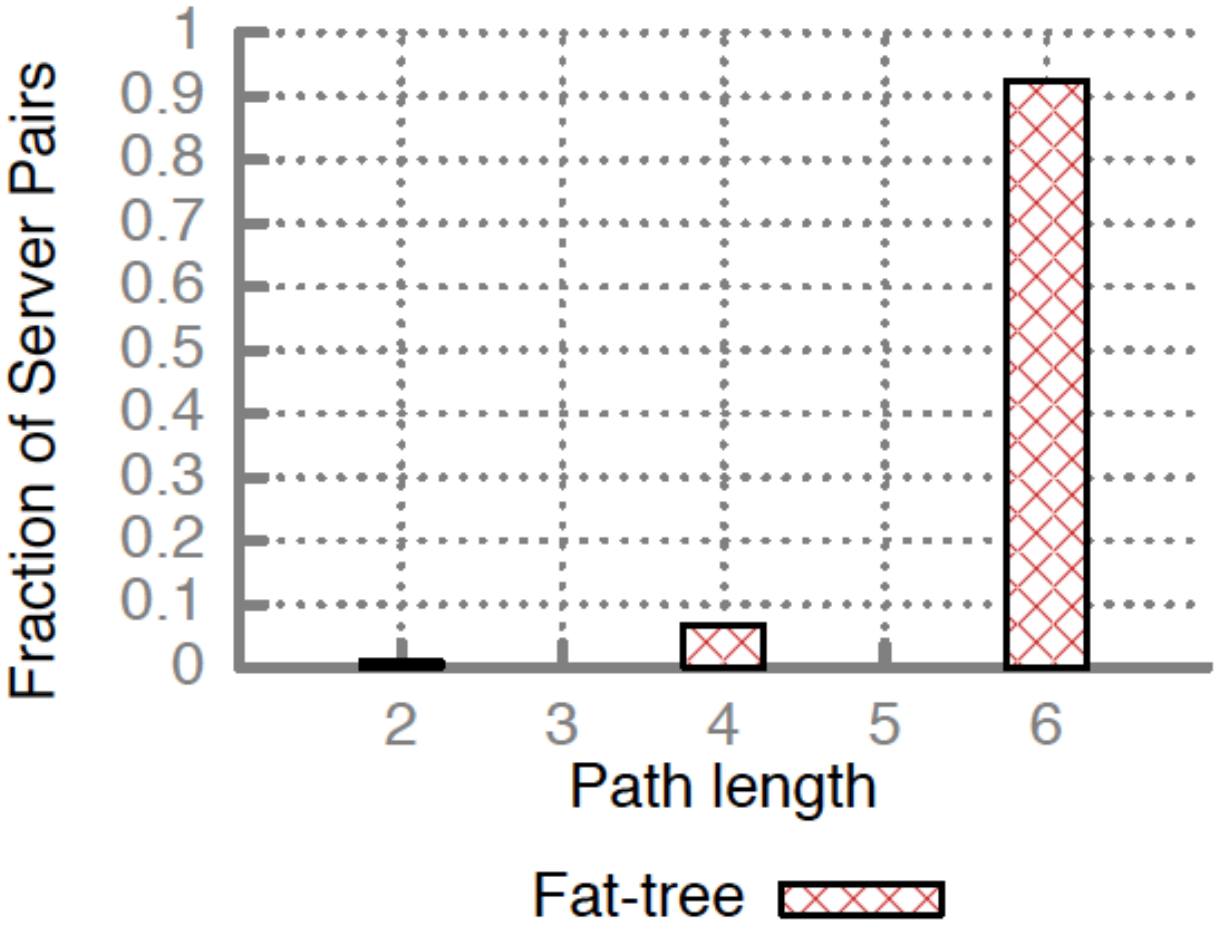
< 6 hops



Fat tree

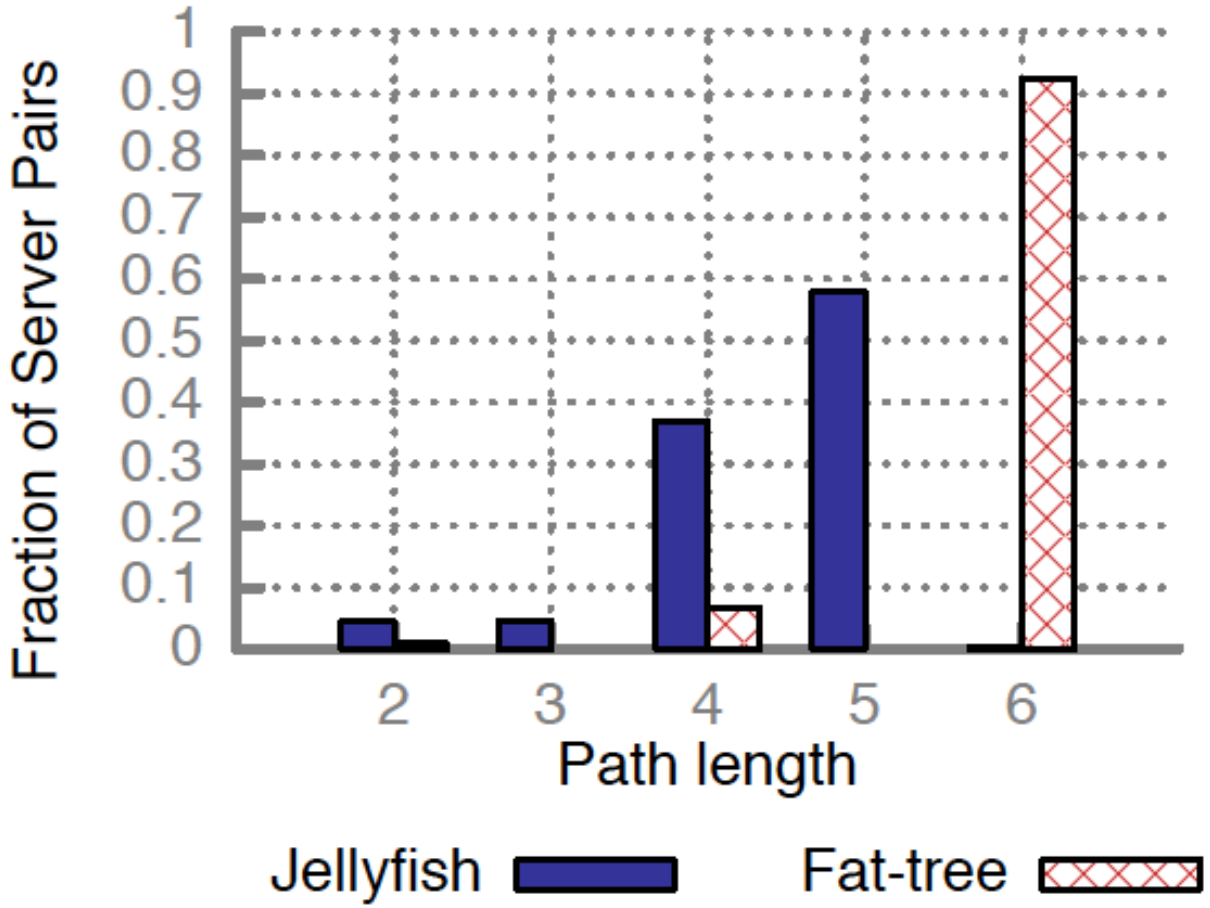
- Edges not helping in reducing path lengths

Jellyfish has short paths



Fat-tree with 686 servers

Jellyfish has short paths



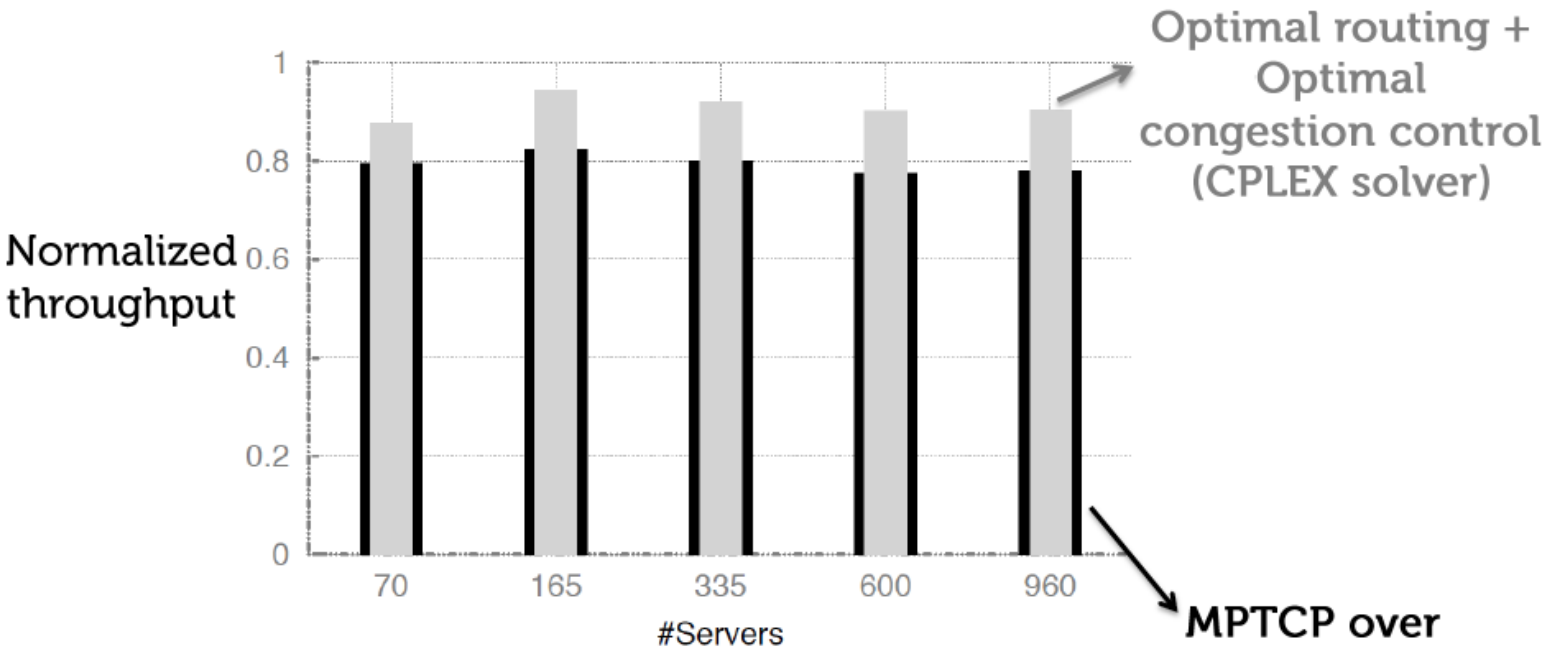
Jellyfish, same equipment

Routing: a simple solution

Find *k* shortest paths

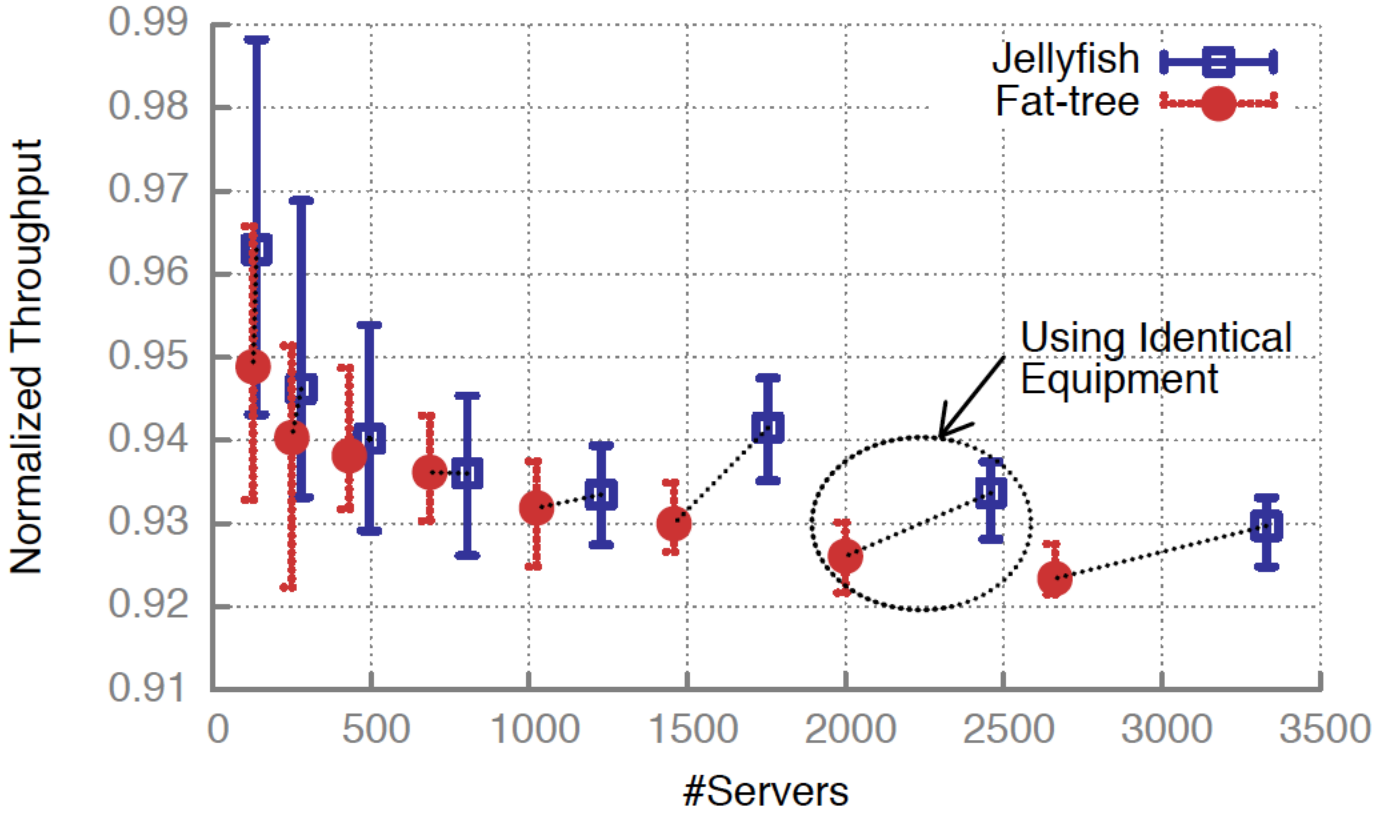
Let Multipath TCP do the rest

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



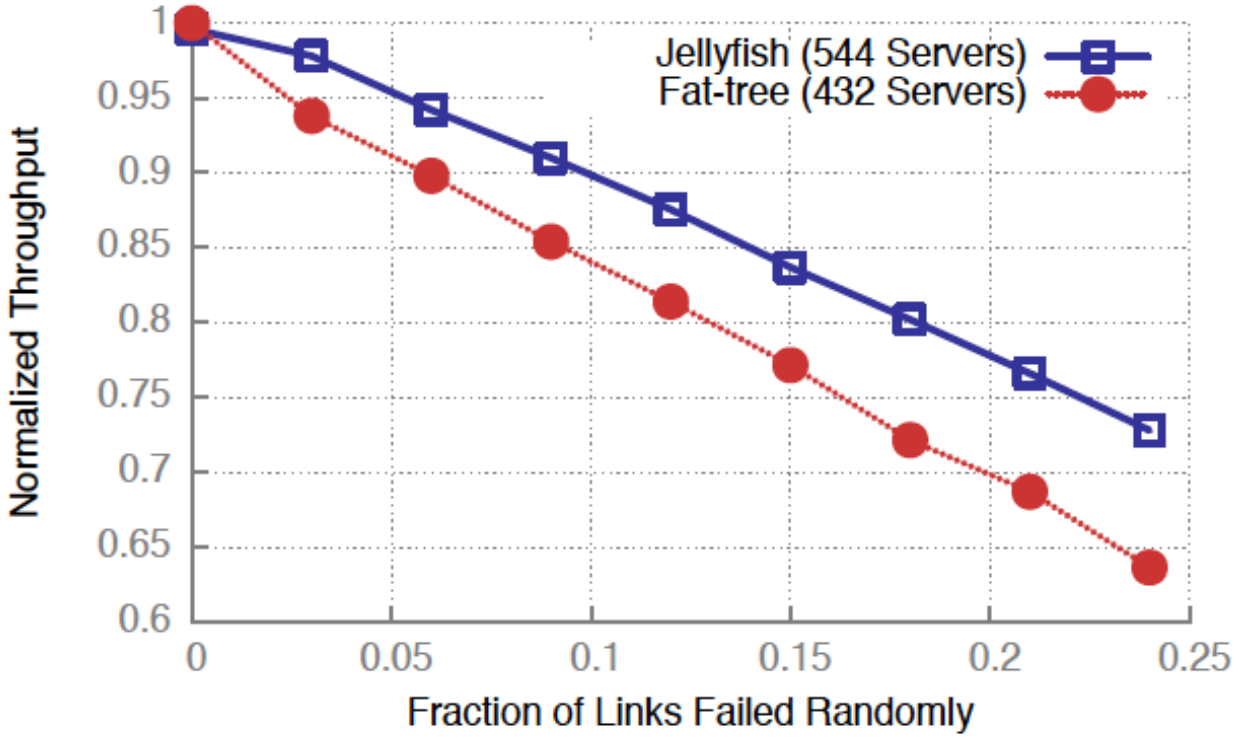
86-90% of optimal!

Fat-tree Throughput Comparison

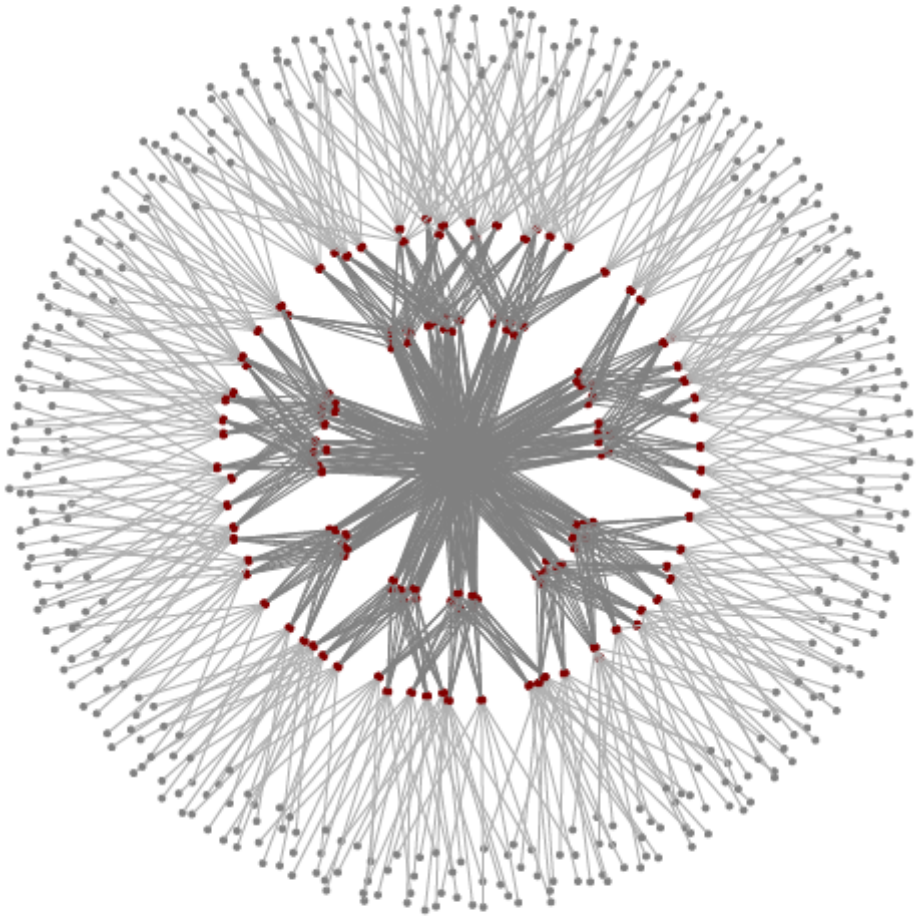


- Based on packet level simulations, including *routing overheads*
- Same hardware; more number of servers in Jellyfish than Fat-tree
- Similar stability

Throughput under link failures

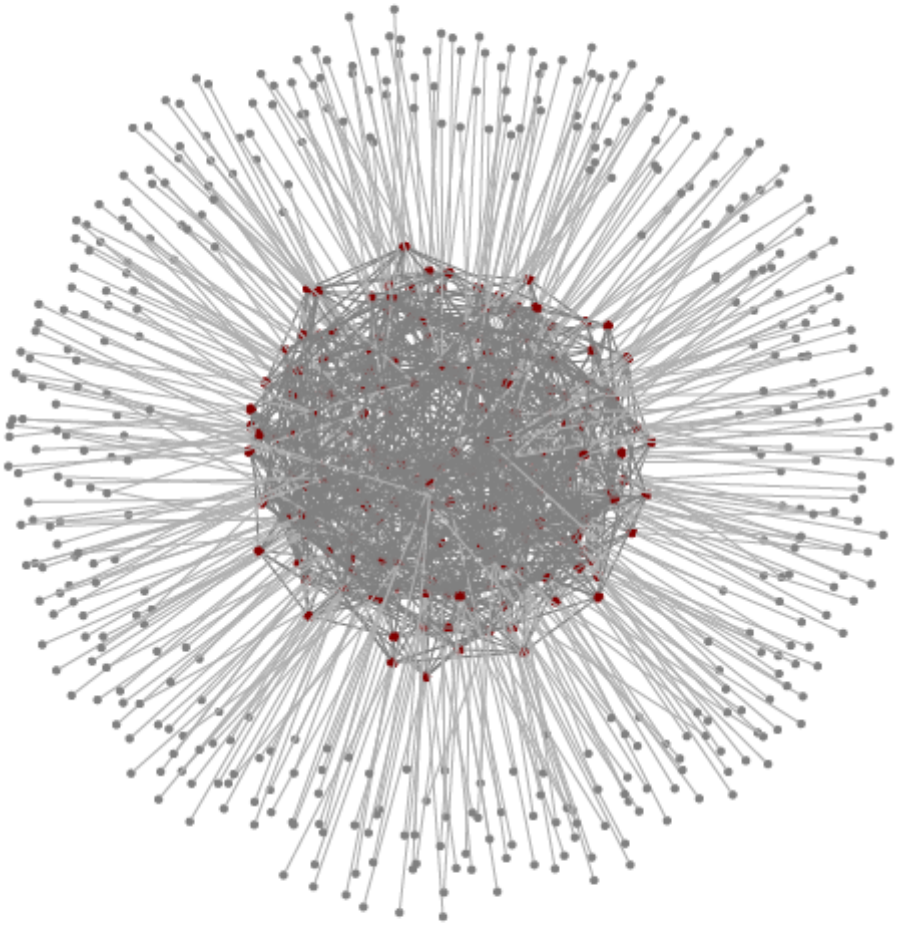


Example



Fat tree

432 servers, 180 switches, degree 12



Jellyfish random graph

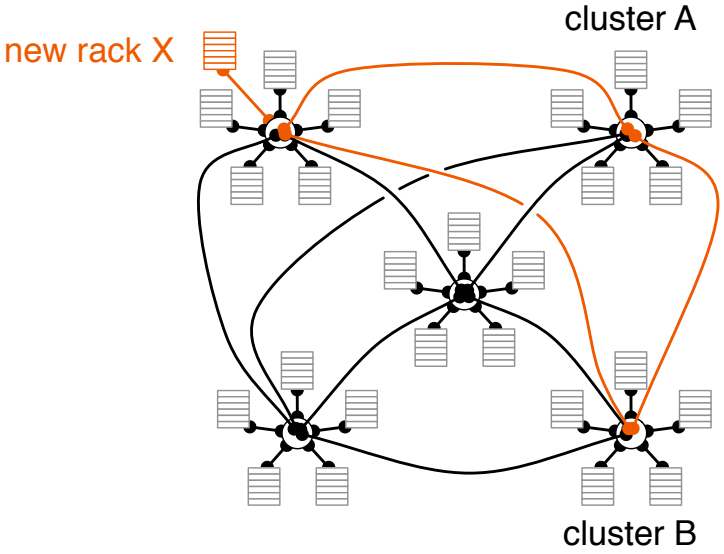
432 servers, 180 switches, degree 12

Cabling solutions

Fewer cables

for same # servers as fat tree

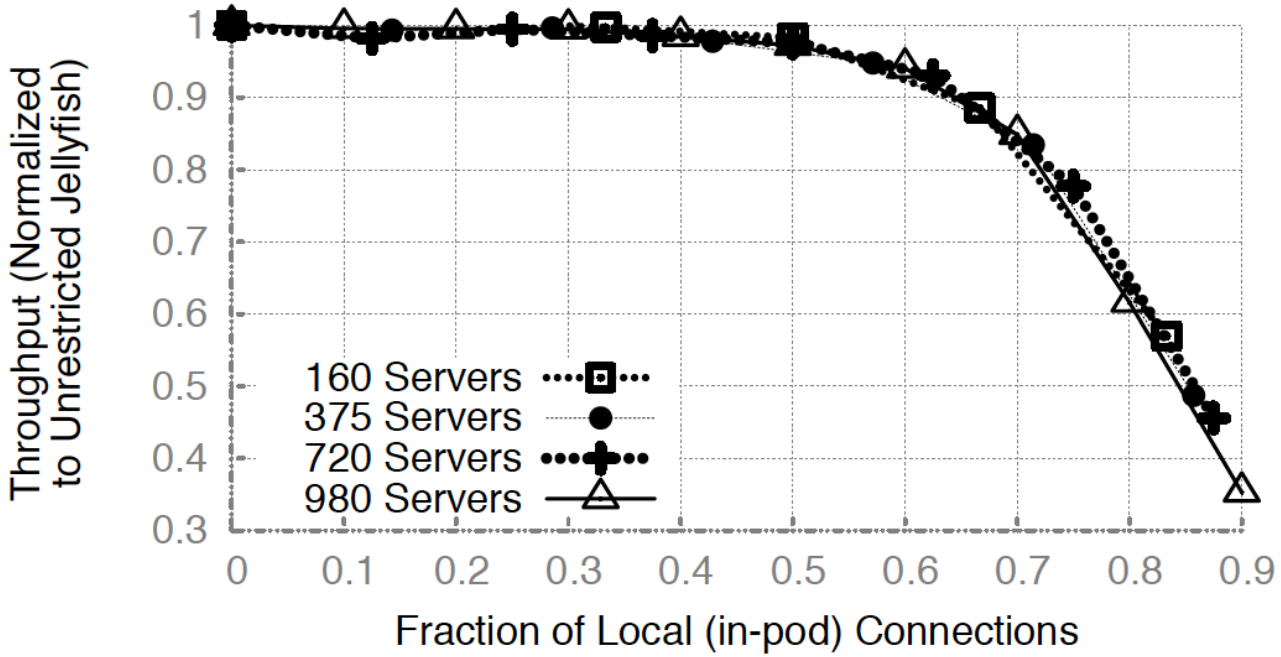
Aggregate bundles



Avoid long cables

< 5% loss of throughput

Cabling Jellyfish in Massive Scale DC



- <6% loss in throughput when 60% of the network connections per switch are localized

Discussion Points

- Real world impact - what is the industry's take on this?
- Is the cabling issue for Jellyfish really resolved from the solutions offered?
  - Patch panels?
  - Is debugging a network of concern?
- Are the routing issues resolved?
  - k-shortest path routing?