A CASE FOR RANDOM TOPOLOGIES IN HPC INTERCONNECTS

Henri Casanova

Univ. of Hawai`i at Manoa

with M. Koibuchi (NII, Japan)

H. Matsutani and H. Amano (Keio Univ., Japan)

D.F. Hsu (Fordham Univ., U.S.A.)

Disclaimer

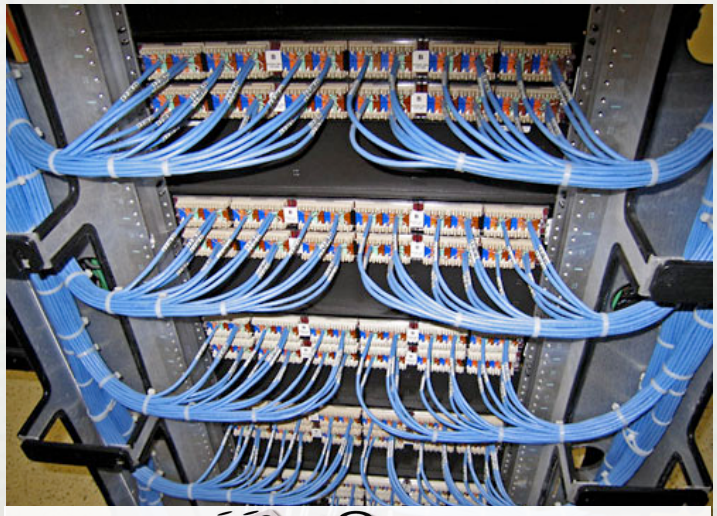
- This is the first talk at the Scheduling Workshop
- Yet, I won't talk about scheduling at all
- Instead, I'll talk mostly about graphs and networking hardware

Why give this talk?

- Pseudo-reason #1 - Among the research I did last year, this is probably the most fun I had
  - And after all it got published in ISCA 2012
- Pseudo-reason #2 - It could revolutionize cluster interconnects (by tomorrow or so...)
  - at least for some kinds of applications/workloads
  - impact on mapping applications to compute nodes

Main Idea

- Forget age-old topologies (tori, grids, hypercubes, trees) that try to be economical or clever
- Instead, just run around the machine room and pull cables into routers at random



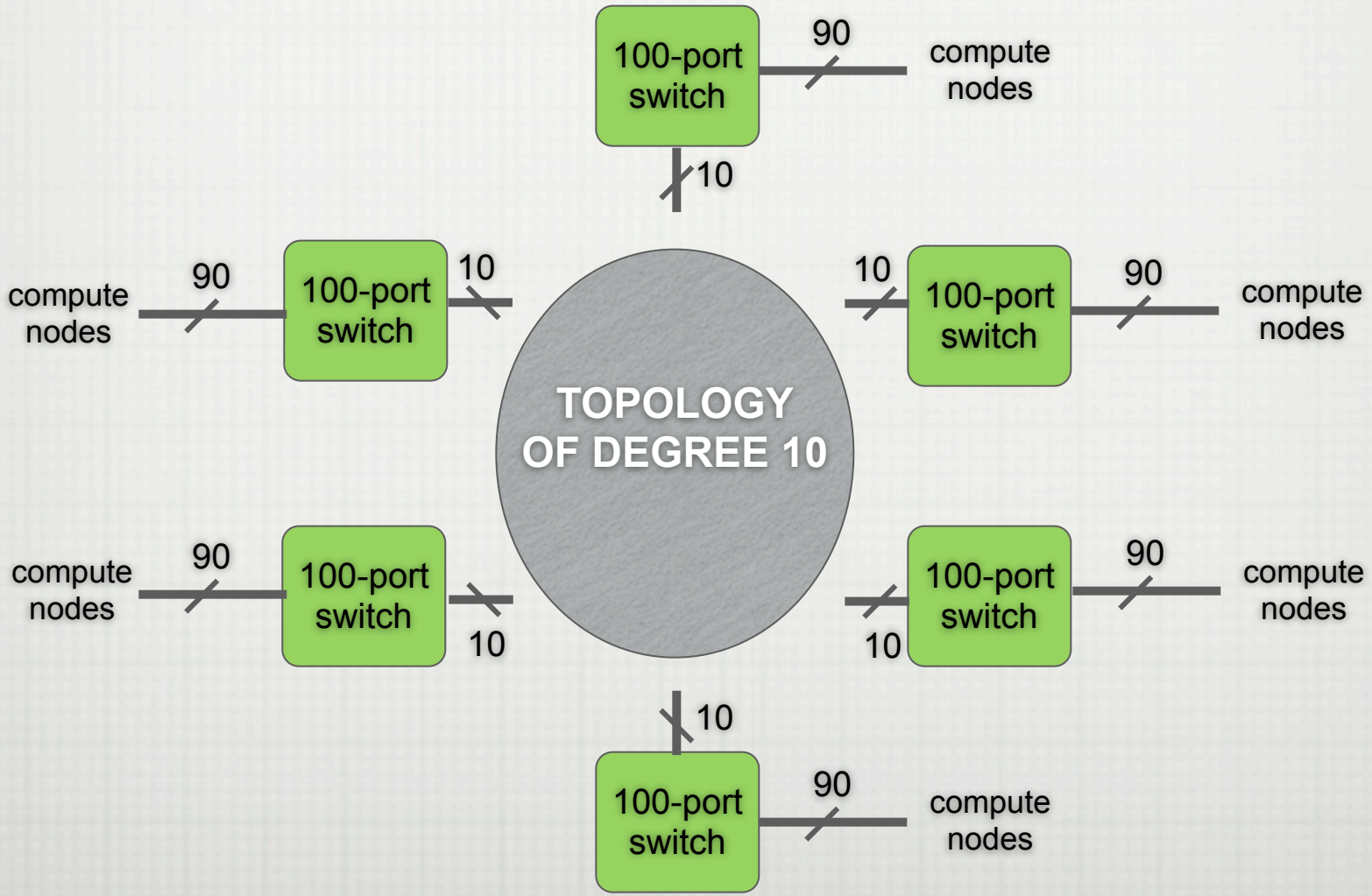
Quest for "Good" Topologies

- Diameter of a graph: longest shortest path between any two vertices
  - Highly correlated to communication latency in network topologies
- Typical problem: maximize the number of vertices in a graph for a given diameter and degree
  - or equivalently: given vertices and a bound on the degree, add edges so as to minimize diameter
- Studied by graph theoreticians for decades
  - Moore bound gives an upper bound on (regular) graph size
  - Many interesting graphs (De Bruijn, (n,k)-star, etc.)
- Several graphs used in practice for HPC interconnects strike different compromises between diameter and degree:
  - grids and tori, hypercube (with many variations), omega and butterfly networks (with many variations), fat trees, etc.

Why would we care today?

- Isn't this all done already?
- Platforms scales are increasing and platforms are built as networks of switches
- Switch delay > 100ns, link delay ~ 5ns/m
- As usual, we want low diameter (i.e., few hops on node-to-node paths)
- But switches with high radix (e.g., > 100 ports) are becoming cheaper
- Therefore, we can use topologies with relatively high degree without incurring too high a cost
  - Different from the "hypercube days" in which increasing the degree by 1 led to an n-fold increase in cost

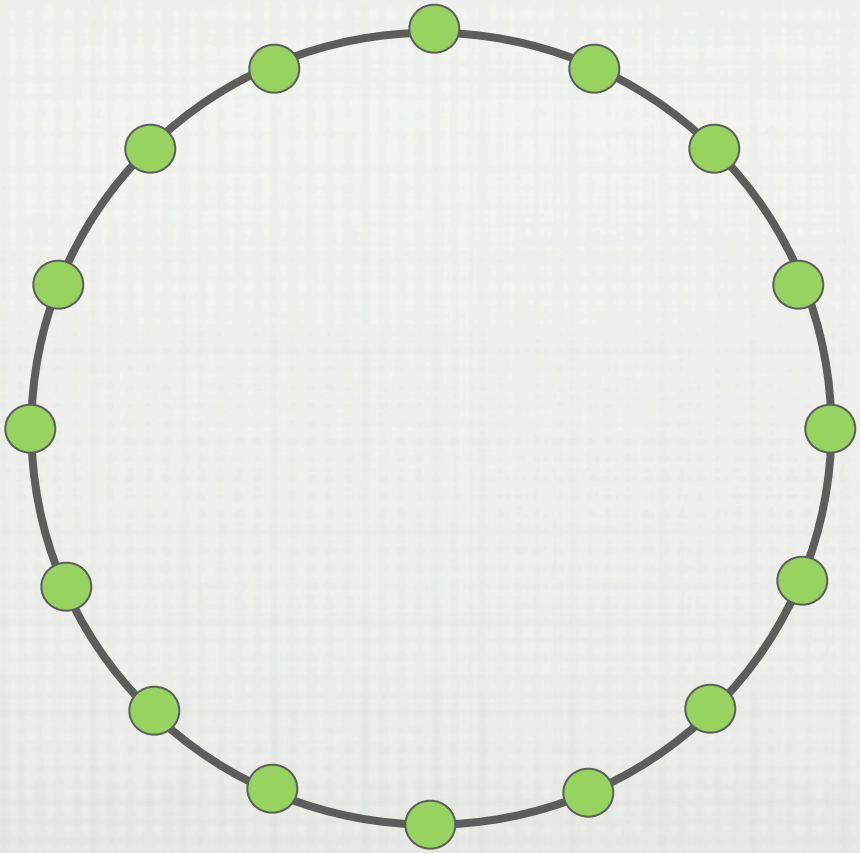
TOPOLOGIES OF SWITCHES



Topologies of Switches

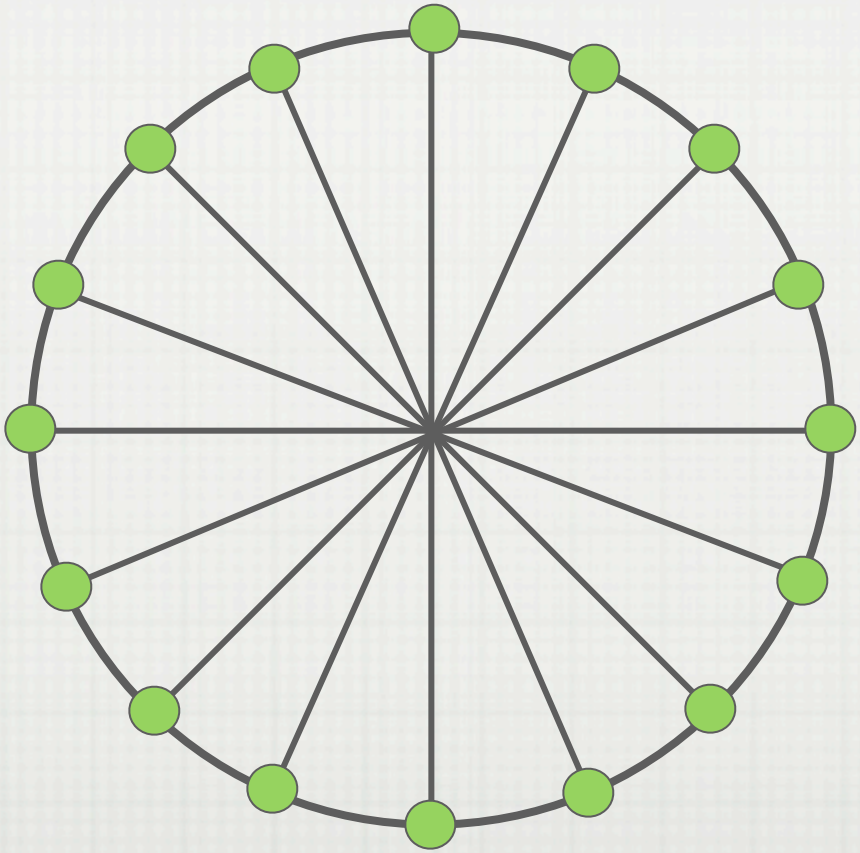
- What graph should we pick for creating a topology of high-radix switches?
- M. Koibuchi came to visit my lab and asked this question
- Our initial attempt: borrow some ideas from structured peer-to-peer networks
  - Degree is O(log n) to keep routing tables “small”
  - So perhaps we can do something similar, but that’s better than, say, a hypercube?
    - and without constraints on the number of nodes
- Common approach in p2p networks: add shortcut edges to a ring to build a Distributed Loop Network (DLN)
  - DLN-x: DLN with degree x

DLN-2



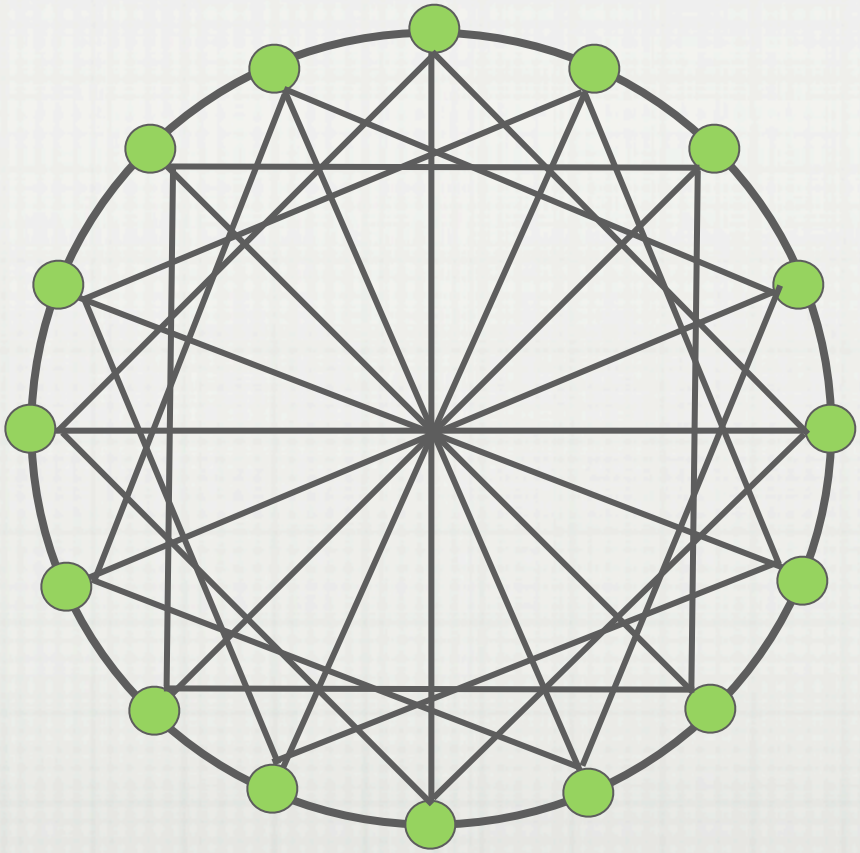
diameter ~ n/2

DLN-3



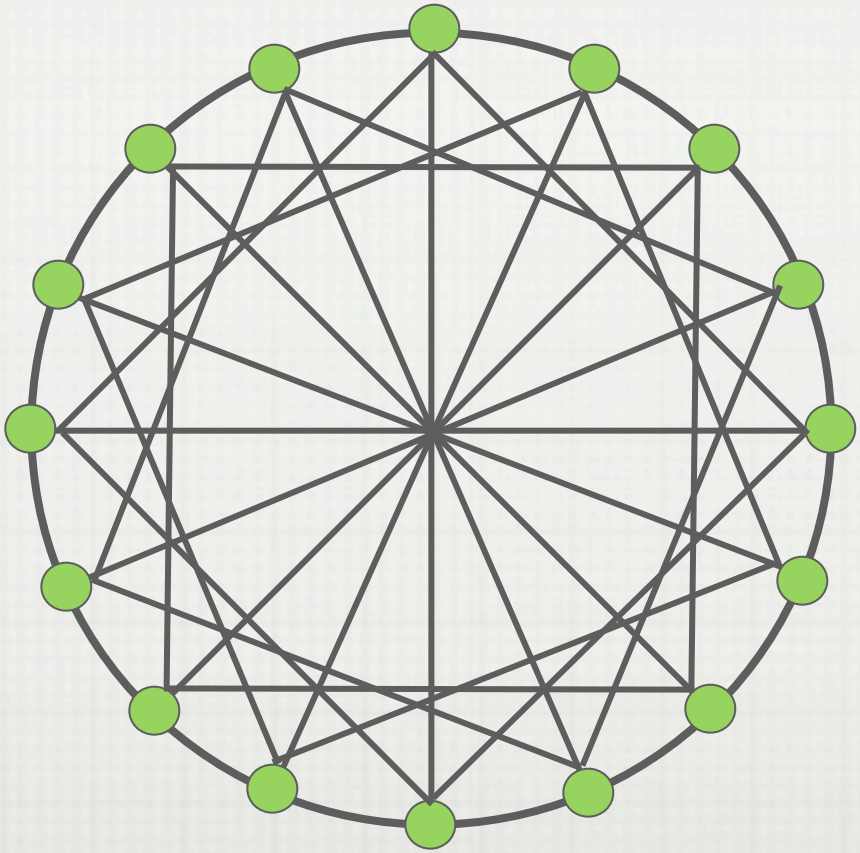
diameter ~ n/4

DLN-5



diameter ~ n/8

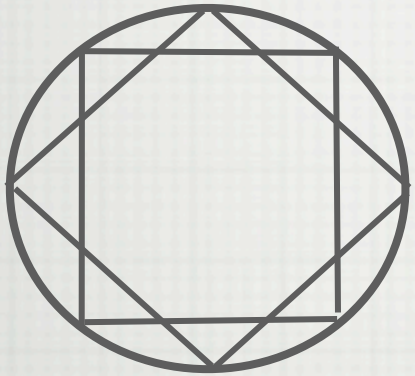
DLN-5



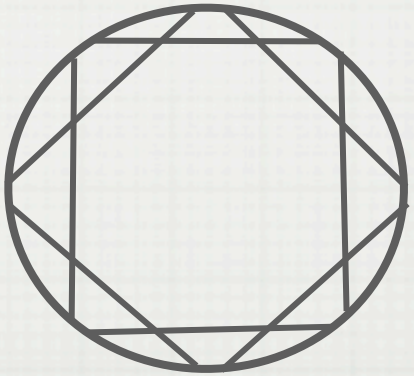
diameter ~ n/8

DLN Topologies

- Many smarter (cheaper) ways to organize the shortcut links likely if your goal is the diameter
- For instance with irregular graphs



diam ~ n/16 + 1 + n/16 ~ n/8
(degree ≤ 4)



diam ~ n/8
(degree ≤ 3)

- What's a good (optimal) deterministic construction here for a bounded degree?
  - For regular graphs or irregular graphs
- This is when we starting reading graph theory literature...

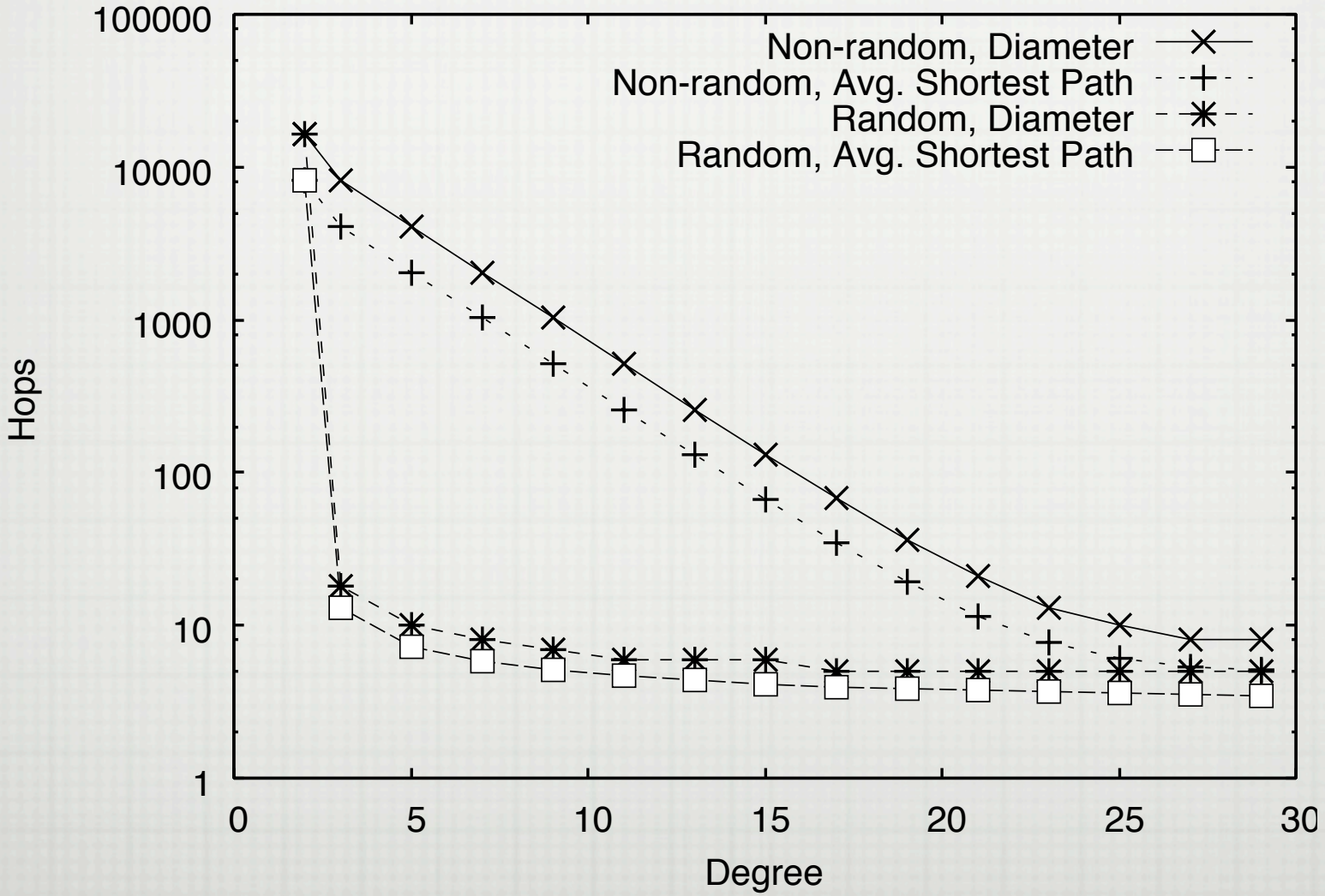
Random DLN???

- *The Diameter of a Cycle Plus a Random Matching*, Bollobás, SIAM J. Discrete Math., 1988
  - Consider a ring of degree 2 (with an even number of vertices)
  - Add one edge between two randomly picked vertices until all vertices have degree 3
  - Question: how good is the diameter?
  - Answer: very close to optimal w.h.p. as n gets large
- General lesson: for a given degree and given bound on the diameter, random graphs are much larger than all cleverly designed non-random graphs
- In other words, random graphs have low diameter
- We quit looking for a deterministic DLN and instead went random!
  - Edges are cheap, we like regular graphs, so perfect matchings are fine

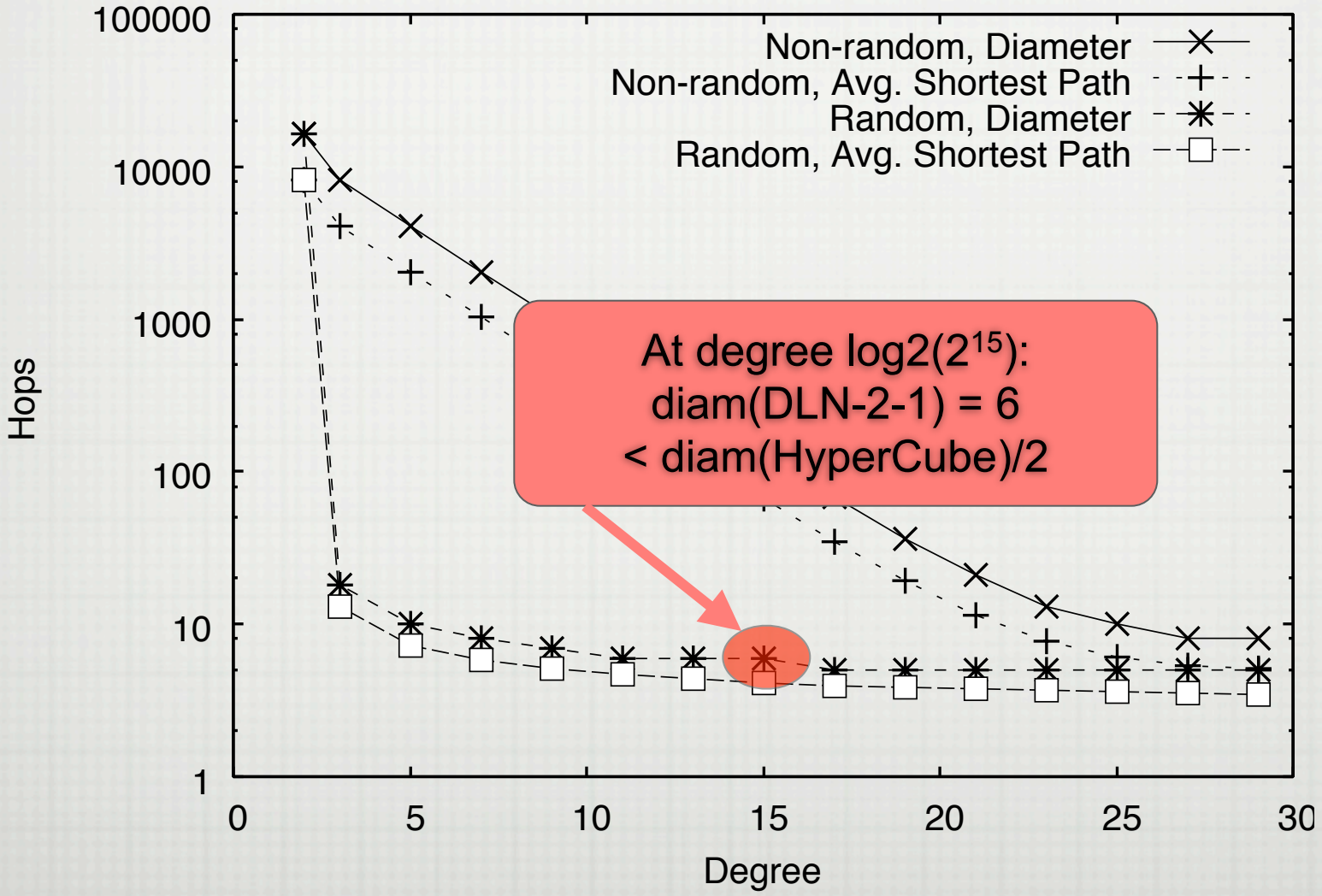
RANDOM DLN

- DLN-x-y: DLN with degree x+y, where y “additional” random shortcut edges are added at each vertex
  - DLN-x-0 is a non-random DLN
- y perfect matches are added to the DLN-x-0 graph using a simple algorithm
- Pick the best generated DLN-x-y sample (best diameter, best average shortest path length for equal diameters) among 100 trials
- Let’s compute the diameter and average shortest path length of DLN-2-(d-2) d for 2^{15} vertices?
  - And show a comparison to DLN-2-0, just for kicks

DLN vs. Random DLN ($n=2^{15}$)



DLN vs. Random DLN ($n=2^{15}$)



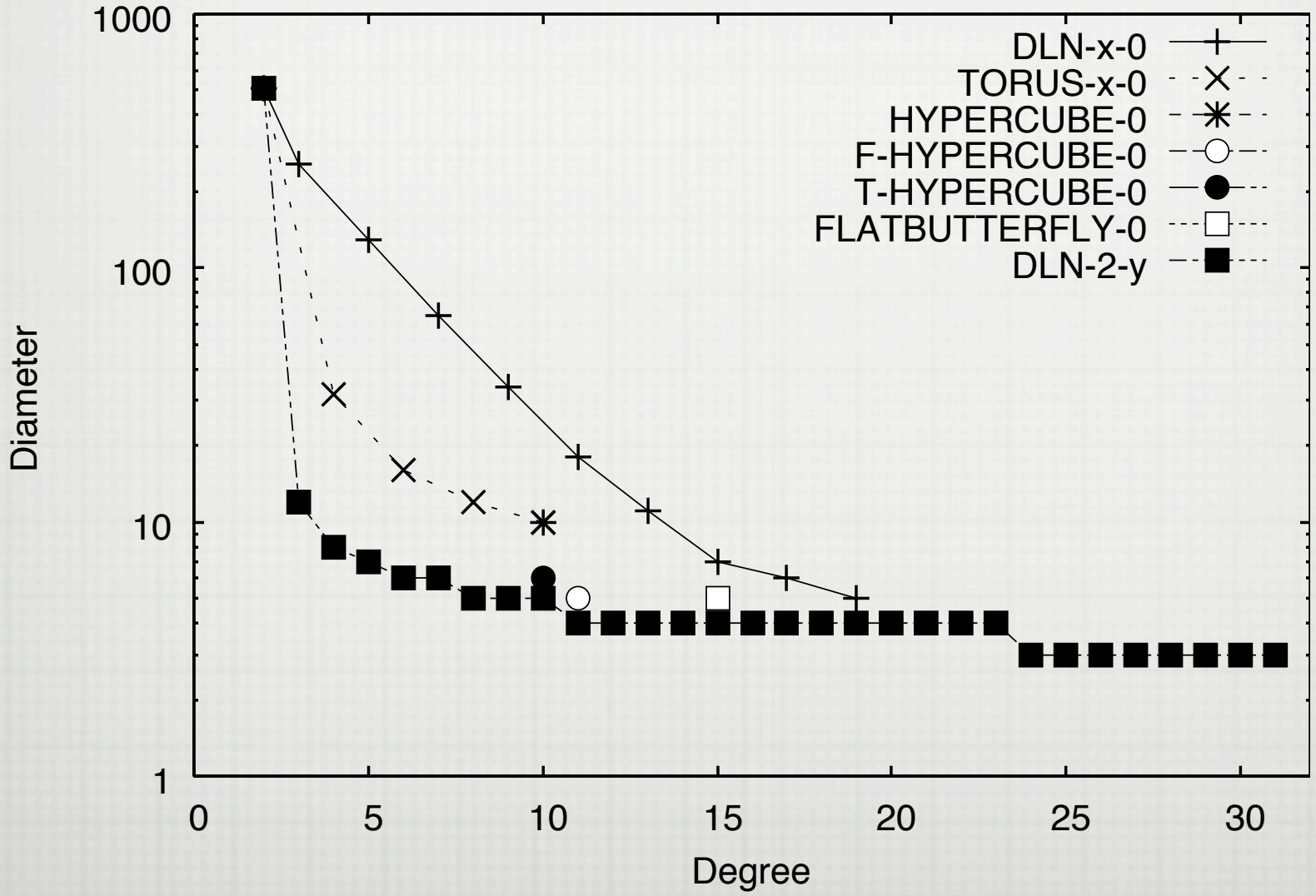
Outline

- It is still important to think of topologies today
- A few random shortcuts drastically reduce diameter
- Comparison to other topologies
- How random is it?
- Network simulations for throughput and latency
- Caveats
- Does any of this matter?

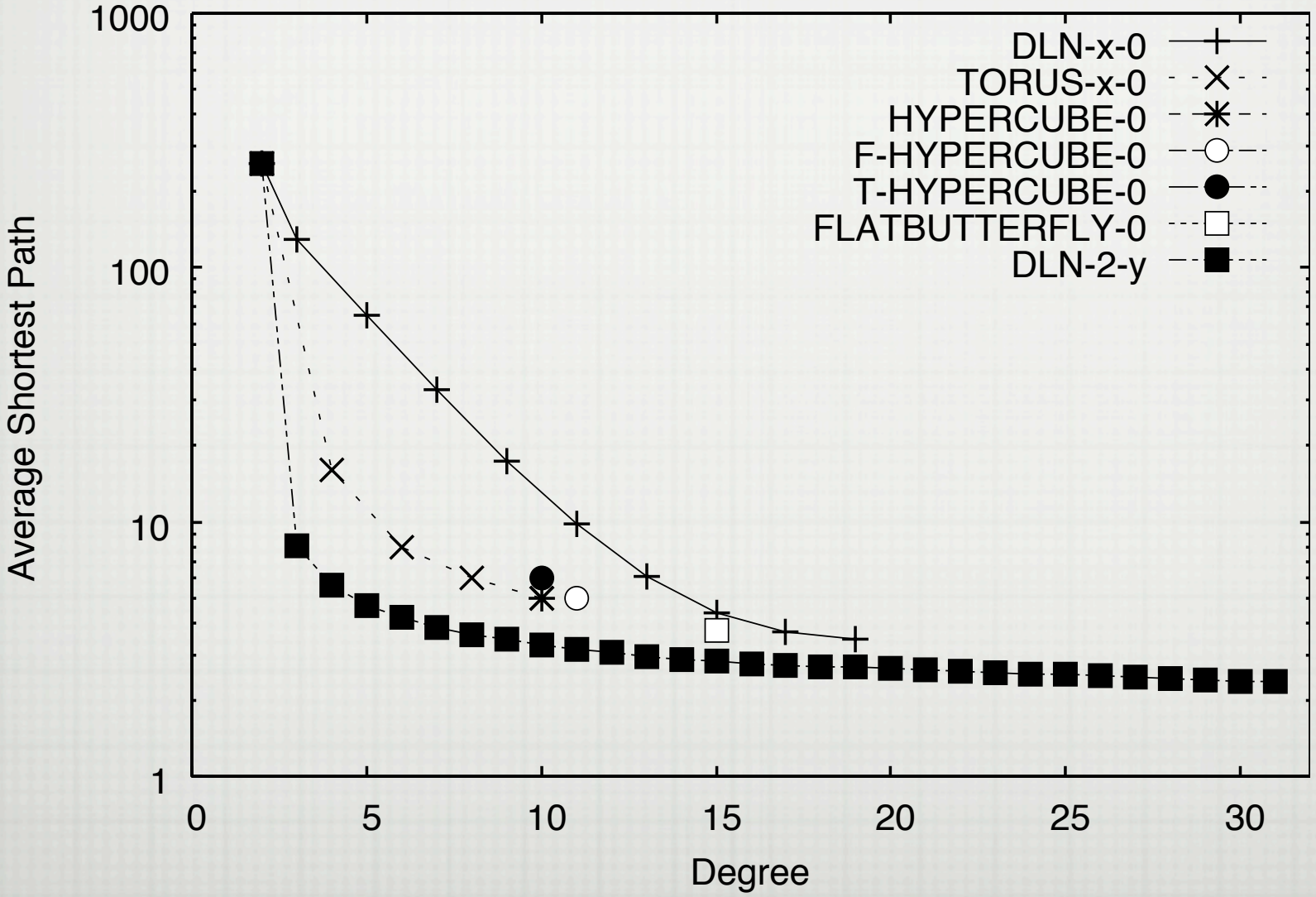
Comparison to Other Topologies

- TORUS-d: Torus of degree d
  - Not at all designed for good diameter of course
- HYPERCUBE
- F-HYPERCUBE: Folded Hypercube [El-Amawy et al., 1991]
  - degree n+1 for 2^n vertices
  - add an edge between vertex x and !x
- T-HYPERCUBE: Multiply-twisted Hypercube [Efe, 1991]
  - degree n for 2^n vertices
  - achieves a lower diameter than the hypercube
- FLATBUTTERLY: Flattened Butterfly [Kim et al., 2007]
  - start with a k-ary, n-layer butterfly network
  - then merge switches into higher-radix switches
  - can be seen as a more extreme hypercube
    - for 2^n vertices, we use the lowest degree flattened butterfly with degree > n

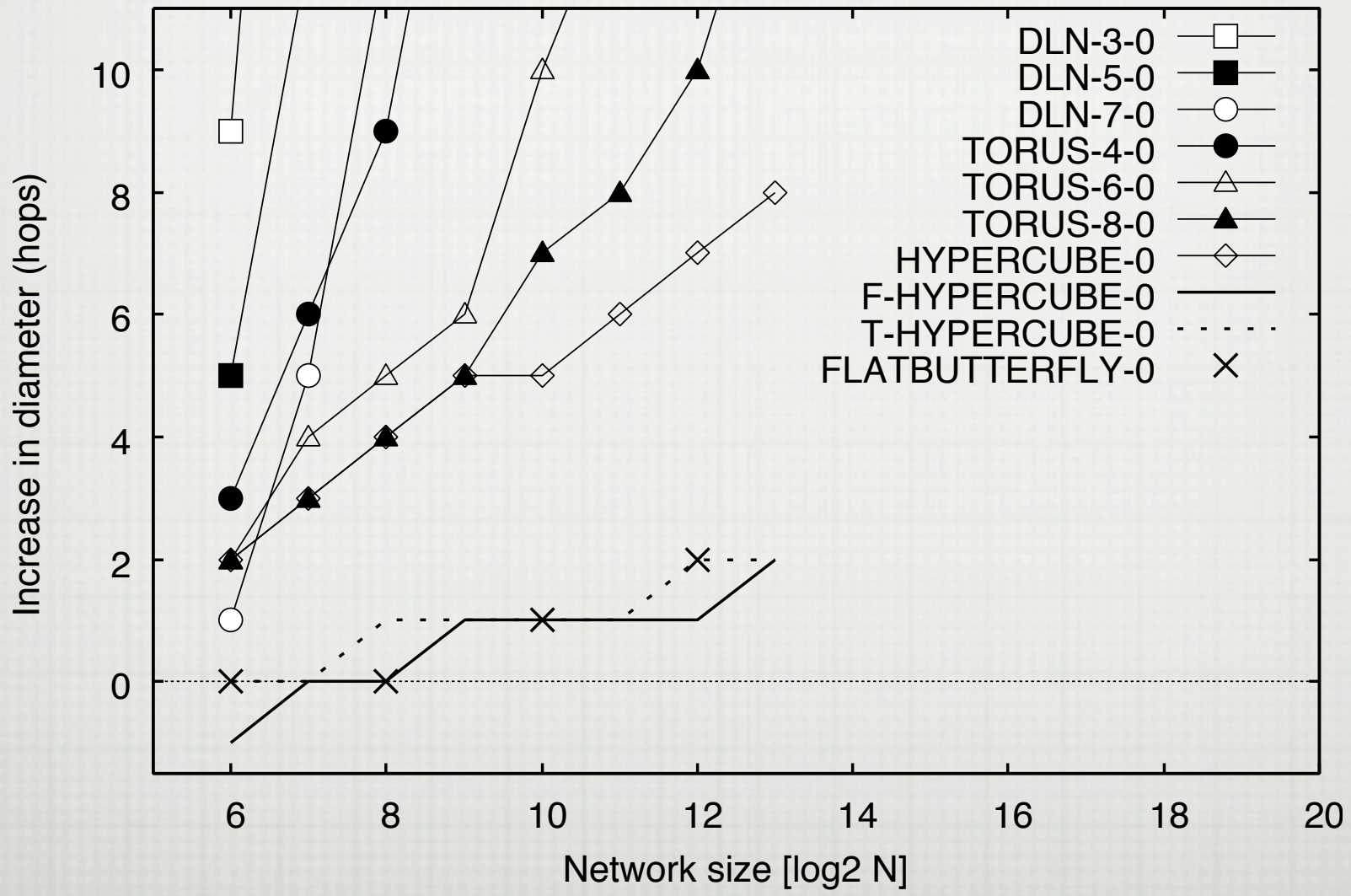
DIAMETER COMPARISON ($n=2^{10}$)



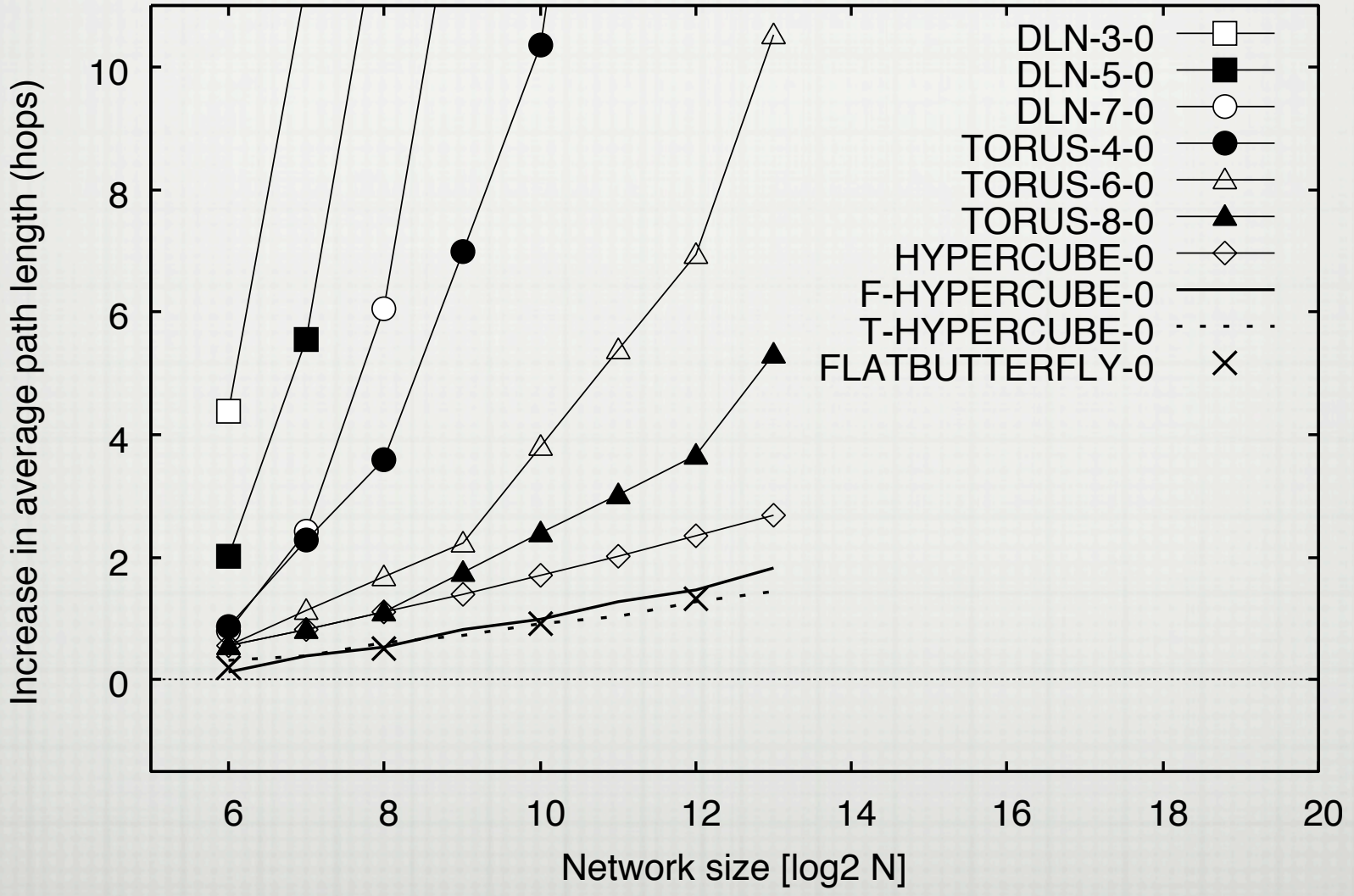
ASPL Comparison ($n=2^{10}$)



Diameter Improvement Scaling



ASPL Improvement Scaling

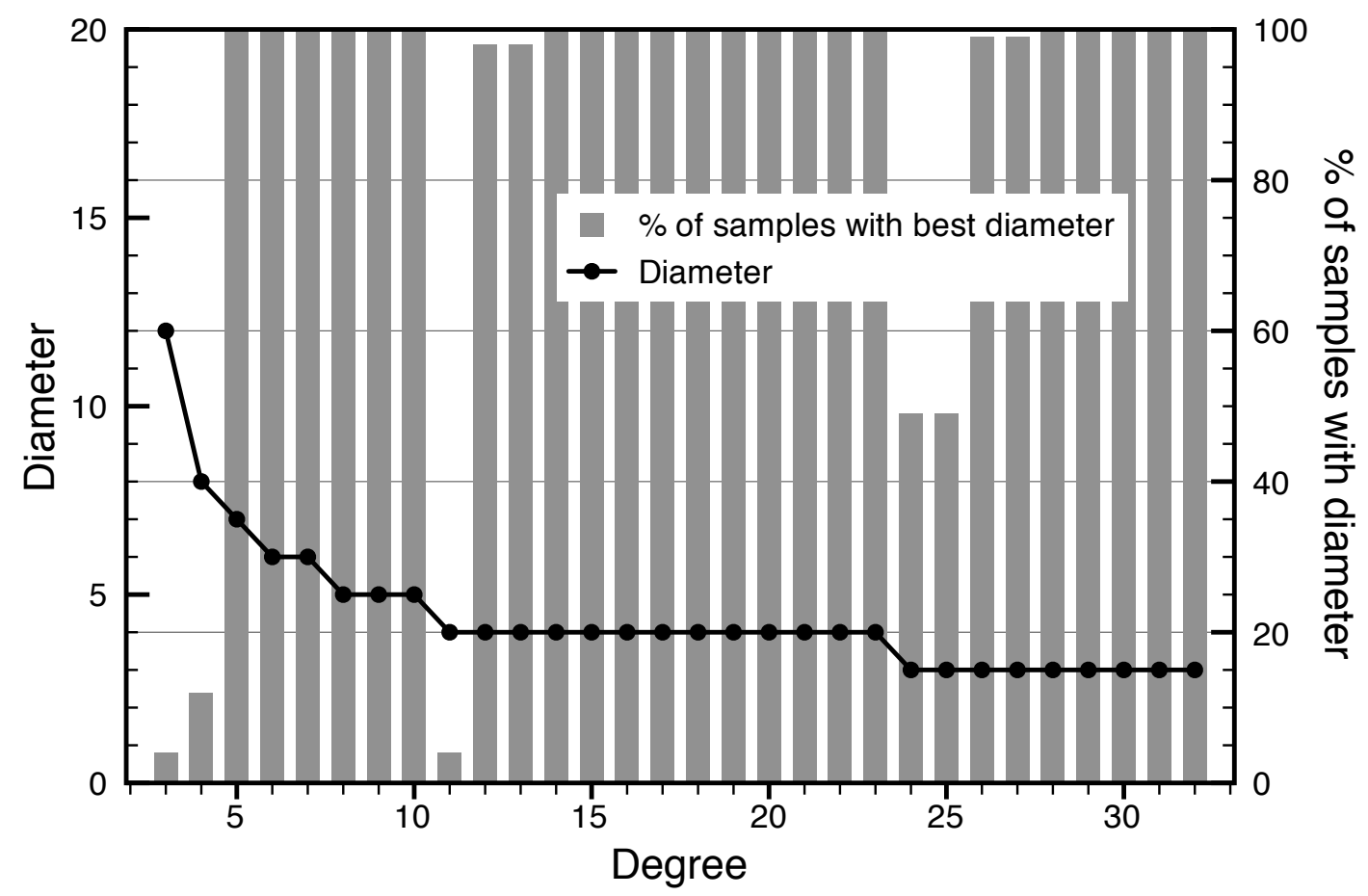


Outline

- It is still important to think of topologies today
- A few random shortcuts drastically reduce diameter
- Comparison to other topologies
- Observations on randomness
- Network simulations for throughput and latency
- Caveats
- Does any of this matter?

NEEDLE IN HAY STACK?

- Question: what's the variation among our 100 samples?



Needle in Hay Stack?

- In fact, at degree d, topologies have diameters that vary by at most 1 hop
  - Some have diameter x, some diameter x+1
- Say that x decreases at degree d+1
- Question: Is there a “lucky” topology with degree d and diameter x-1?
- Empirical answer: No improvement when using 10,000 samples

- In practice, a “good” topology is found in the first 100 samples

Better Randomness?

- We have generated random shortcut edges without caring about the "quality" of the shortcut
  - e.g., if two vertices already have a short shortest path, then it's not useful to add a shortcut between them
- When generating a shortcut, generate k candidate shortcuts and pick the one between the vertices that have the longest shortest path
- k=2 improves diameter over k=1 in < 8% of the cases
- k=5 improves diameter over k=2 in < 4% of the cases
- The improvement is one hop (and increasing the degree by 1 "negates" the improvement)
- Improvements in ASPL are at most 0.02%
- In the end, "stupid" shortcuts are fine

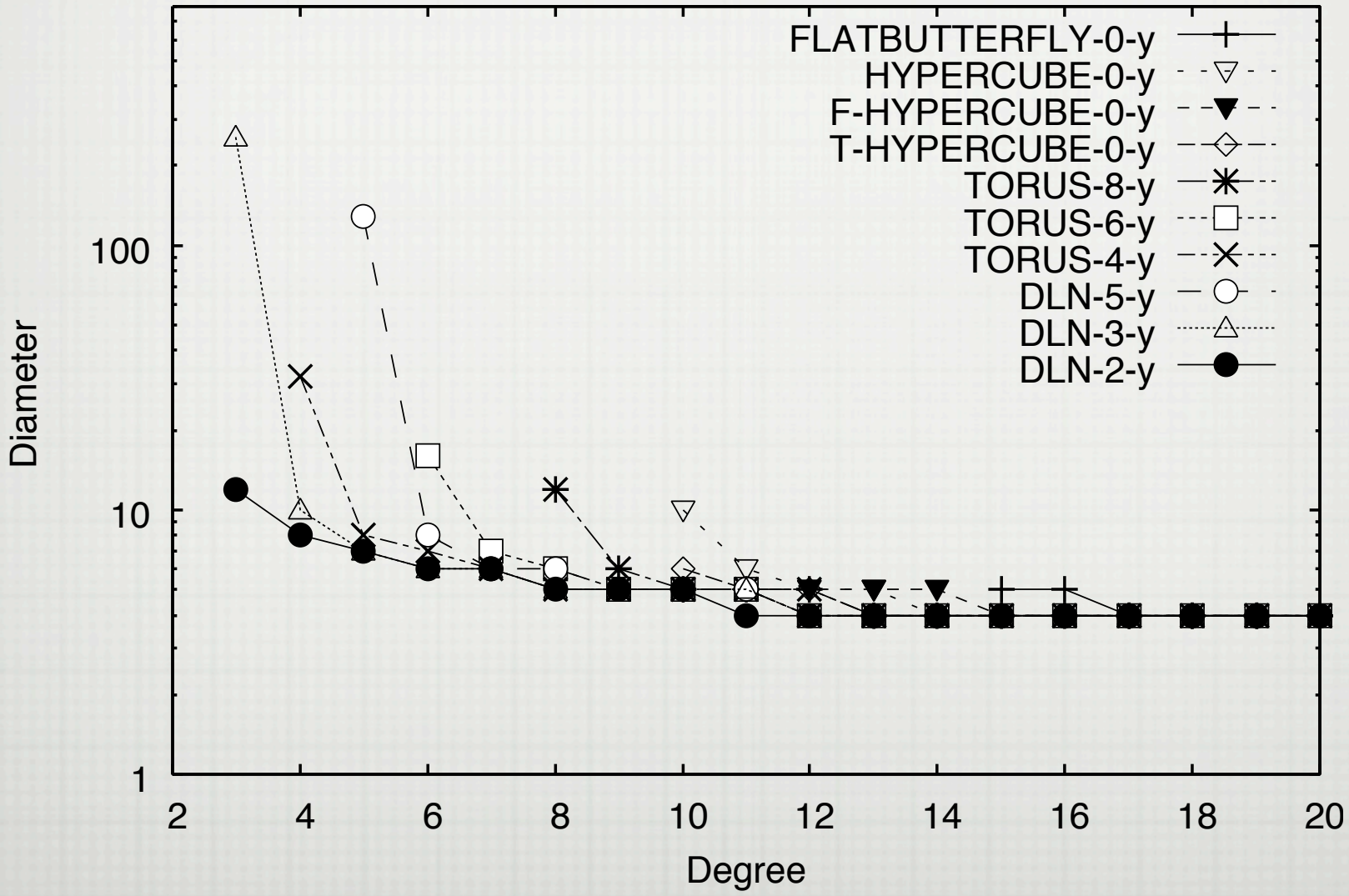
Non-regular Topologies

- How about not enforcing that the graph is regular
  - vertices can have different degree
  - which is fine for a topology of high-radix switches
- Makes shortcut generation simpler
- But in fact leads to slightly less good diameter and average path length
- In the end, enforcing regularity is a good idea

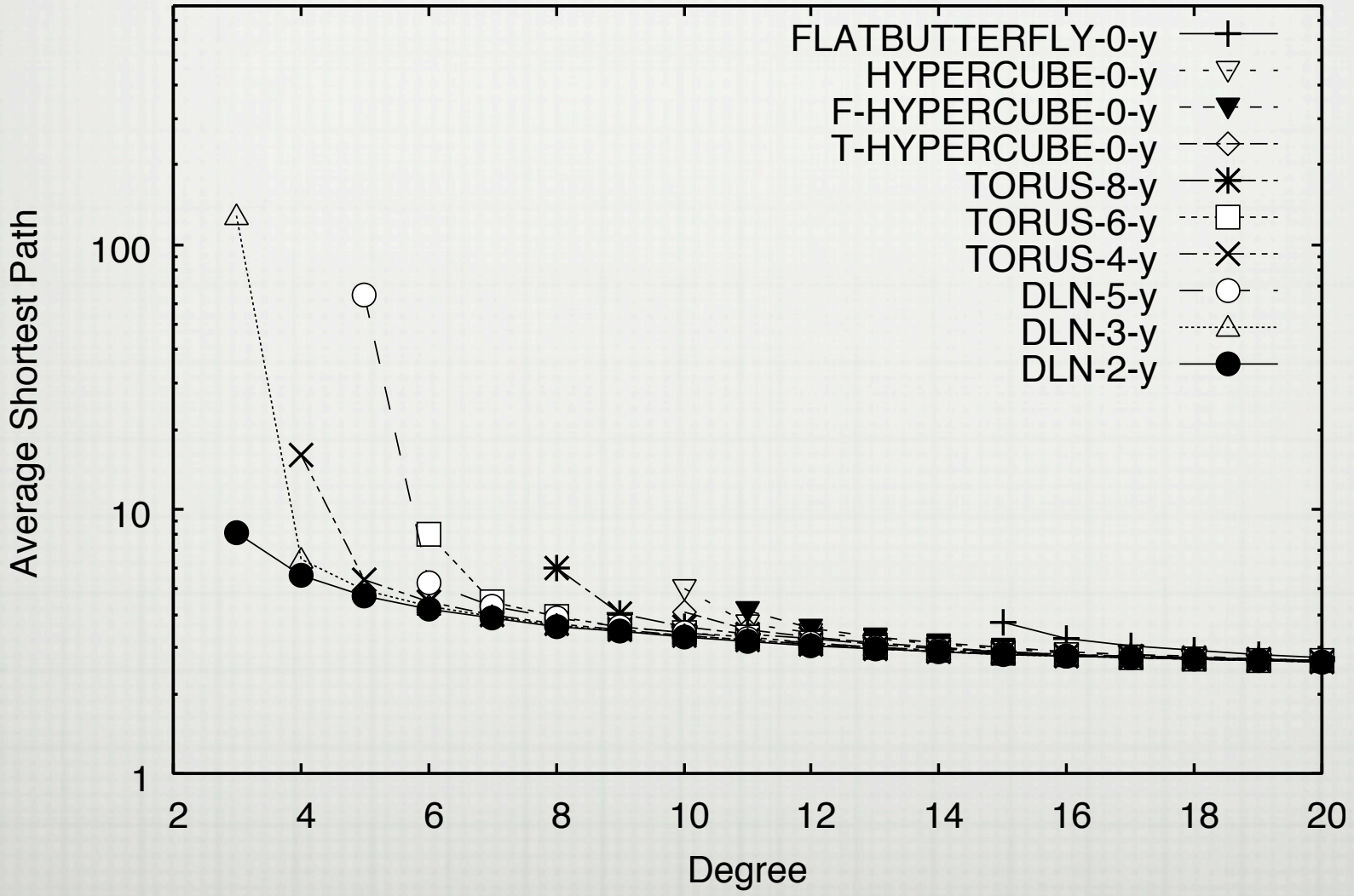
Less Randomness

- How about replacing DLN-2 by a better base topology before adding shortcut?
- Perhaps enhancing a smart topology with a few random edges will lead to good results...

LESS RANDOMNESS (DIAMETER)



LESS RANDOMNESS (ASPL)



Less Randomness

- Adding a few shortcut links to a base topology leads to large payoffs
- But starting with a good base topology doesn't work better than using DLN-2
- In the end, the more non-random edges the higher the diameter/APST

Guidelines for Random Topologies

- Use DLN-2 as a base topology
- Add perfect matchings to it to maintain regularity
- Few random samples are sufficient to obtain a good topology
- Generating high-quality shortcuts only pays off a little bit
- Great pay-offs at low degree
- And it can all be done for whatever number of switches

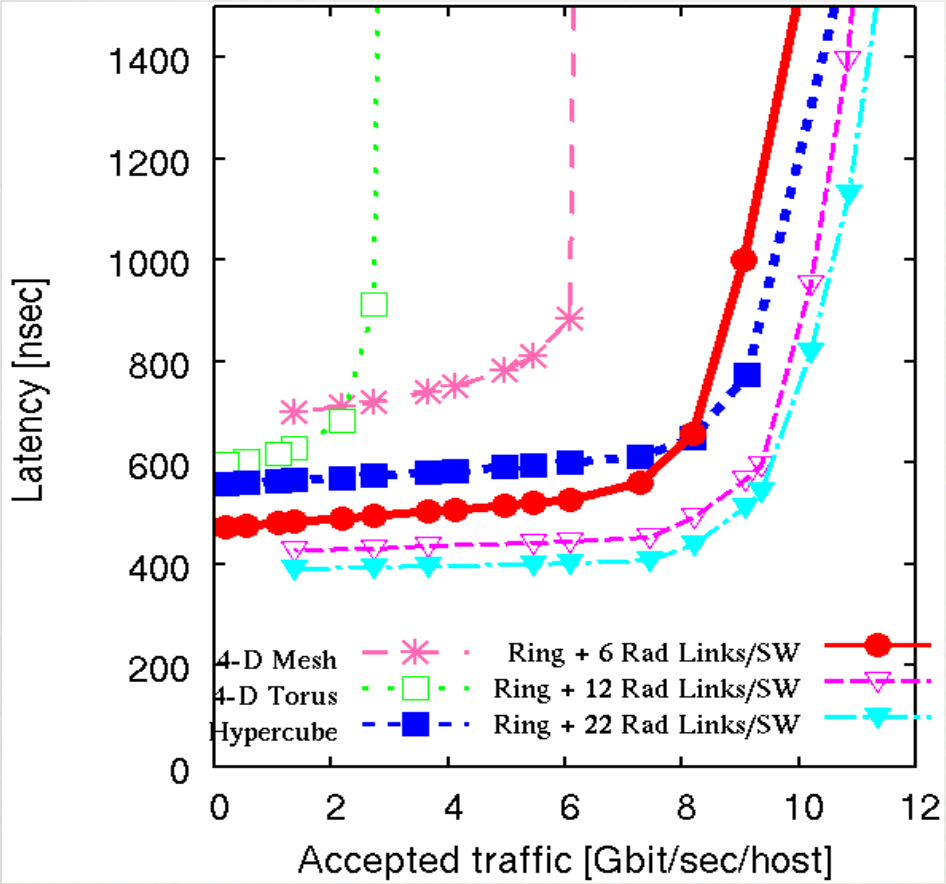
Outline

- It is still important to think of topologies today
- A few random shortcuts drastically reduce diameter
- Comparison to other topologies
- How random is it?
- Network simulations for throughput and latency
- Caveats
- Does any of this matter?

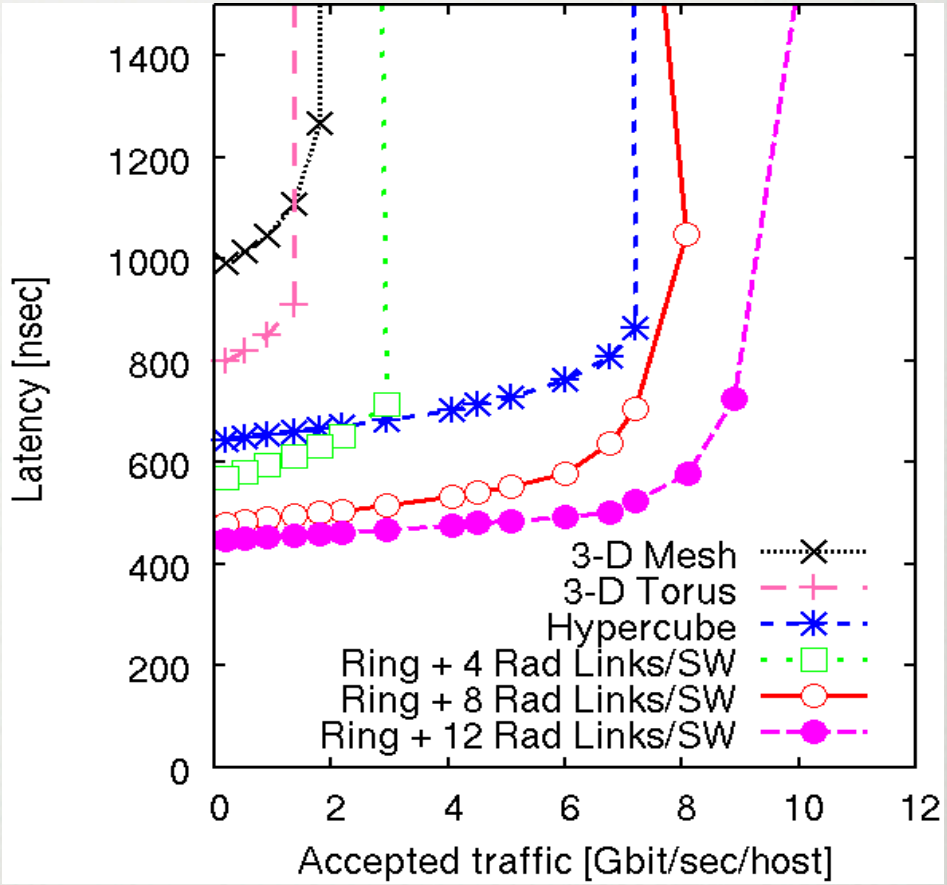
Simulation Environment

- We use a cycle-accurate flit-level network simulator of cluster interconnects
  - 1 packet is 33 flits, 1 flit is 256 bits
- Switch delay = 100ns, Link delay = 20ns
- Link bandwidth = 96 Gbps
- Three classic traffic patterns used in “how good is my network?” studies:
  - Uniform (random)
  - Matrix transpose
  - Bit reversal
- We measure latency and throughput

SOME RESULTS

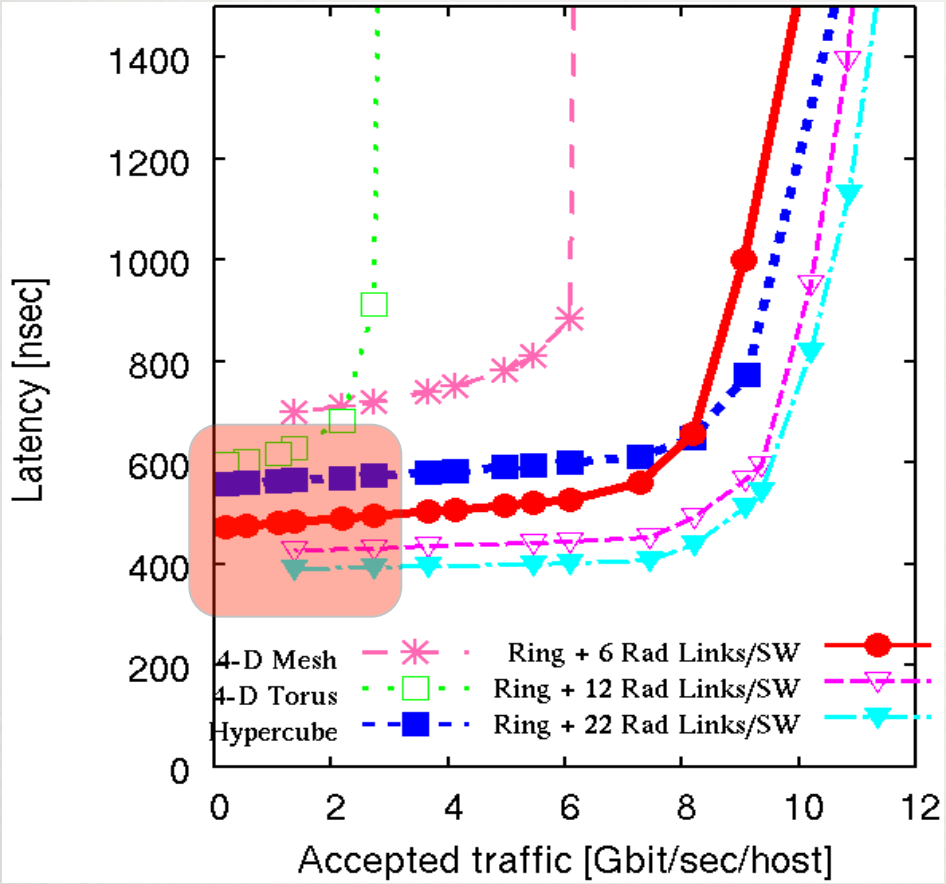


n=256, bit-reversal

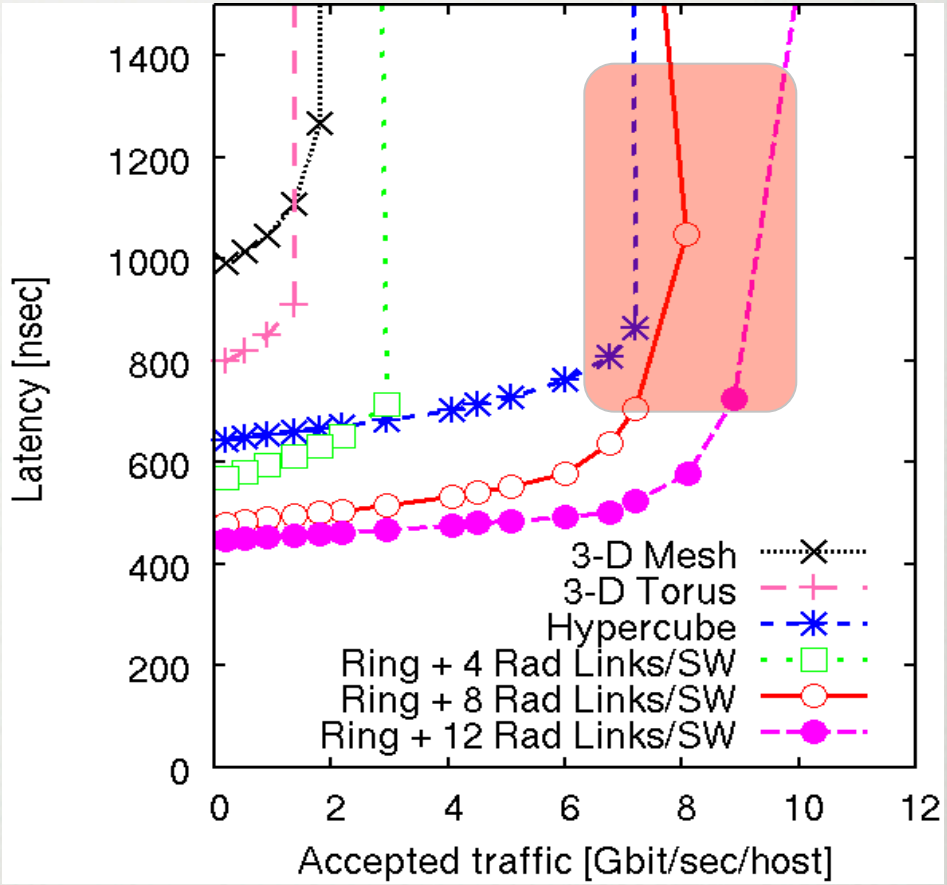


n=512, matrix-transpose

SOME RESULTS



n=256, bit-reversal



n=512, matrix-transpose

Outline

- It is still important to think of topologies today
- A few random shortcuts drastically reduce diameter
- Comparison to other topologies
- How random is it?
- Network simulations for throughput and latency
- Caveats
- Does any of this matter?

Are you serious?

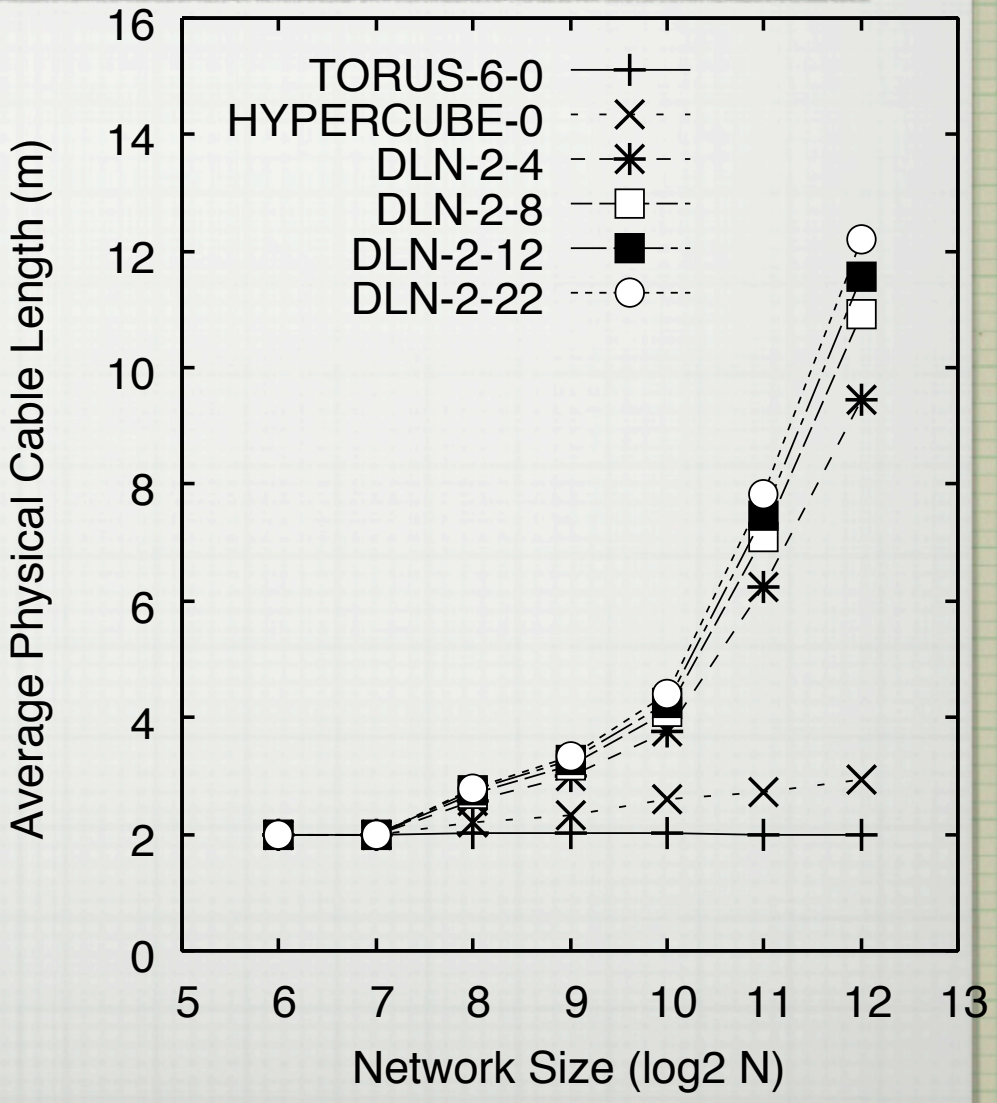
- Graph analysis and simulation results seem to show that random topologies are a good idea
- Good luck trying to convince anybody to “go random” for a real platform today
- Likely complaints:
  - Routing scalability
  - Cabling costs

Routing Scalability

- Because the topology is random, routing must be done with routing tables
  - Routing on a torus is trivial
  - No clever hypercube-like routing scheme with tiny electronics solutions
- But, 87% of Top500 platforms use Ethernet or Infiniband, meaning that they use routing tables
- So the vast majority of high-end HPC platforms suffer from routing table scalability anyway
  - And there are solutions to compact routing tables anyway
- We conclude that routing scalability is not a show stopper for random topologies

CABLING COST

- Cabling cost is proportional to cable length but mostly to link type:
  - passive copper: 10m
  - active copper: 40m
  - optical: ~100m
- Assuming standard cabinet layouts and Manhattan distance



Cabling Cost

- DLN-2-x leads to longer average cable length
- But it can use the same cheap cabling technology as non-random topologies for most cables
- There may be some particularly shortcuts that require long cables

DOES ANY OF THIS MATTER?

- If you're doing a single parallel dense linear algebra app, you want a torus anyway
  - And HPC networks will likely always provide a torus-like network
  - Would be interesting to see how much is lost in practice when switching to a random topology
- If you're running an irregular application then good diameter and ASPL make your life easier
  - No matter your application mapping, you'll do pretty well
  - Coming up with a clever mapping of the application on a particular topology is known to be hard in general
    - but good research topics for students
- Even more true if you're running multiple arbitrary communicating applications/services onto a cluster
  - which is where most of the interest comes from I think

Conclusion

- Future work:
  - What's the penalty for bounding above the maximum cable length of a shortcut edge?
  - Are perfect matchings overkill?
    - Do we care?
- Questions?