Layout-conscious Random Topologies for HPC Off-chip Interconnects

Michihiro Koibuchi — National Institute of Informatics, JP

Ikki Fujiwara — National Institute of Informatics, JP

Hiroki Matsutani — Keio University, JP

Henri Casanova — University of Hawaii at Manoa, US

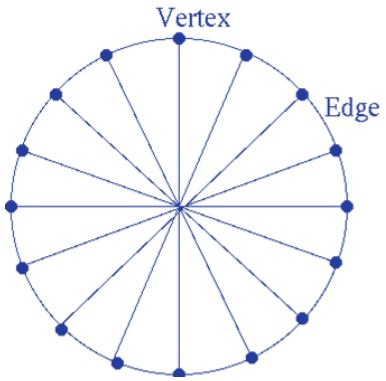
The 19th International Symposium on High-Performance Computer Architecture (HPCA), Feb. 2013

Outline

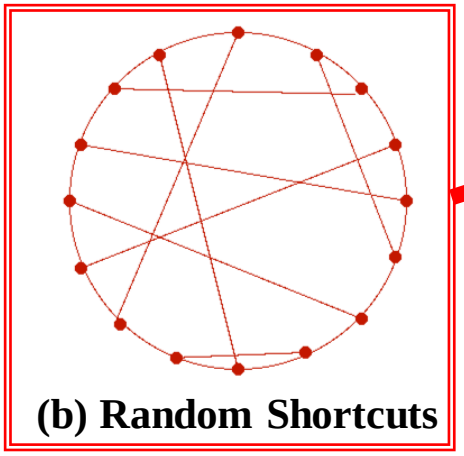
- **Introduction**
- Motivation
- Layout-conscious random topologies
  - (method A) link permutation
  - (method B) constrained random shortcutting
- Comparison of both methods
- Conclusions

Good Point of Random Topology

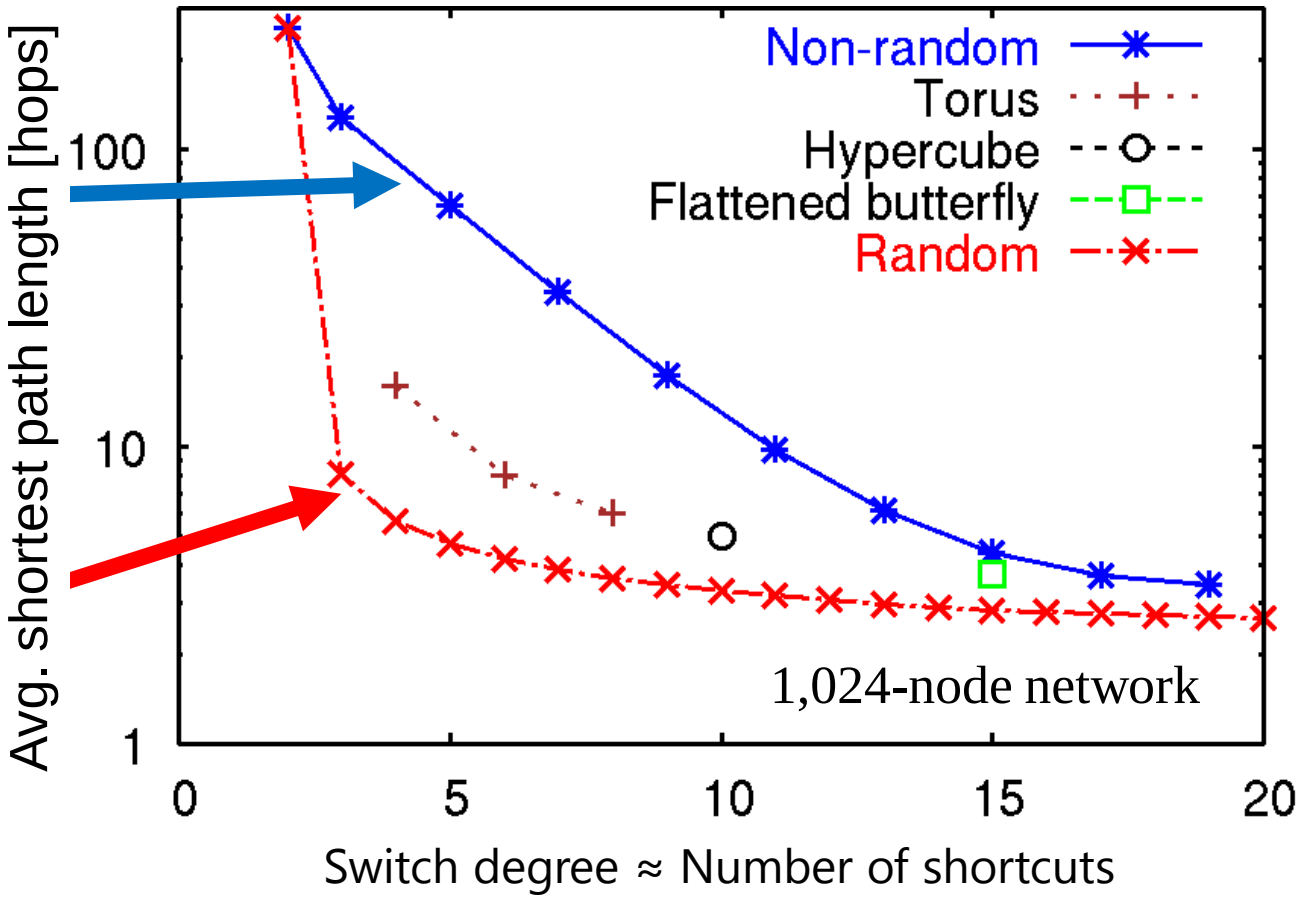
- Goal: to make a low-latency topology for HPC networks
  - low diameter and low average path hops
- Random topology is best [Koibuchi et al, ISCA2012]



(a) Non-random Shortcuts

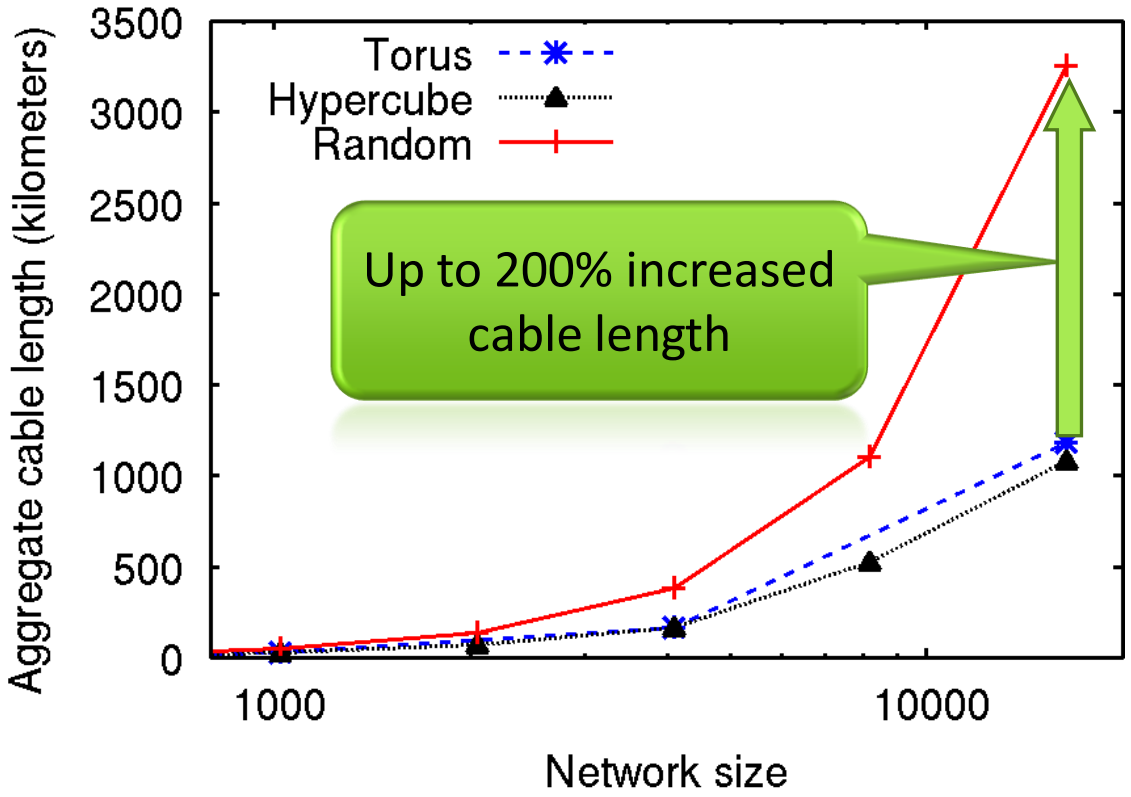


(b) Random Shortcuts



Bad Point of Random Topology

- Random topology leads to lowest latency [Koibuchi et al, ISCA2012] but causes longer cables



- Idea: to design a **layout-friendly quasi-random** topologies
  - **shorter cable length** (near to tori & hypercube)
  - **low path hops** (near to fully random)

Outline

- Introduction
- **Motivation**
- Layout-conscious random topologies
  - (method A) link permutation
  - (method B) constrained random shortcutting
- Comparison of both methods
- Conclusions

Why Should We Care about Topology Now?

	2011		2015		2019	
System Size Sockets Peak PF TF/Socket	32,768 32 1.0		32,768 200 6.1		32,768 800 25.0	
	Expect	Want	Expect	Want	Expect	Want
NIC B/W (B/F)	0.01 - 0.1	1.0	0.005 - 0.03	1.0	0.025 - 0.25	1.0
Link B/W (B/F)	0.01 - 0.1	1.0	0.005 - 0.03	1.0	0.025 - 0.25	1.0
MPI Latency (ns)	750 - 1500	500	500 - 1000	400	400 - 750	300
MPI Throughput (M Msg/s)	20	50	80	300	300	1200
Load/Store (M Msg/s)	75	400	150	1,600	300	6400
Load/Store Latency (ns)	300	100	300	100	300	100

1µs system-across latency is desired [Henmmert, 2008]

[Tomkins, 2008]

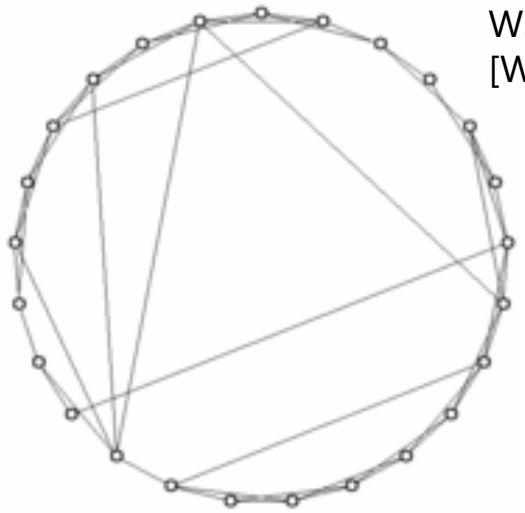
Sandia National Laboratories

Switch delay>100ns, Link delay=5ns/m

New low-diameter/hop topology is needed

Randomness Makes Graphs Smaller

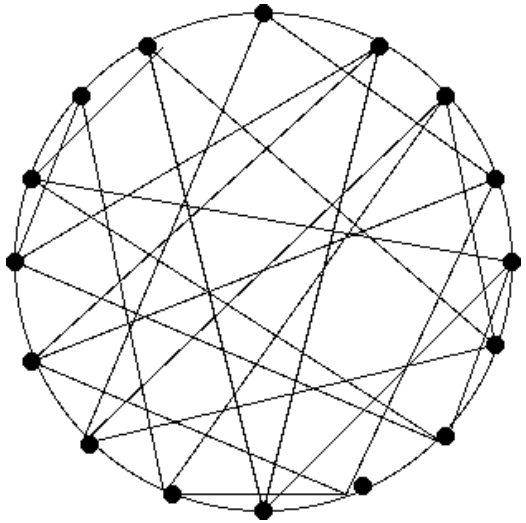
- Small-world phenomenon
  - Social network
  - P2P network
  - Airline network



WS model
[Watts, 1998]

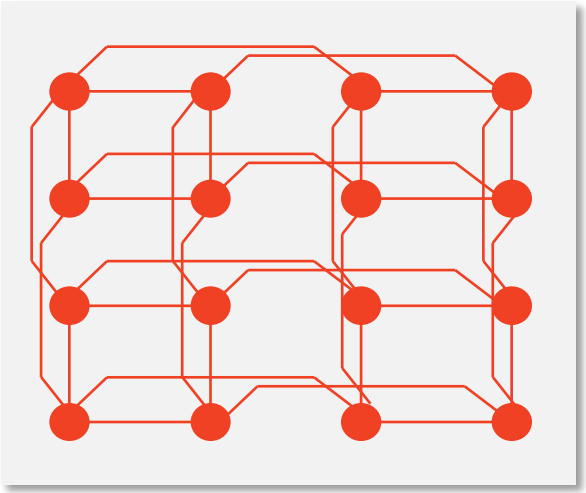
Vertex = Person/Computer/Airport

- Its use for HPC interconnects
  - Relatively high radix
  - More uniform degree
  - Considering rack layout

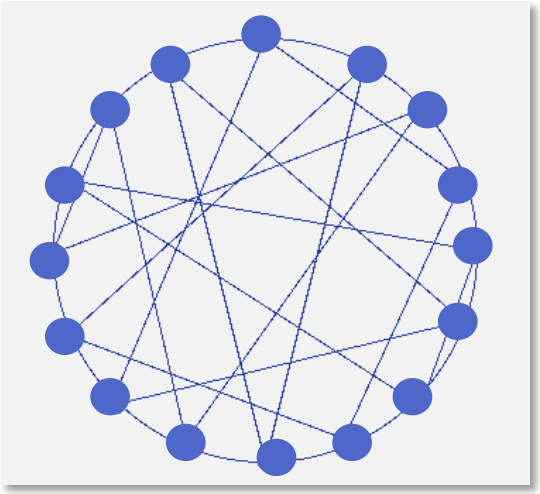


Vertex = Switch
Edge = Links

Random Topology Minimizes Latency

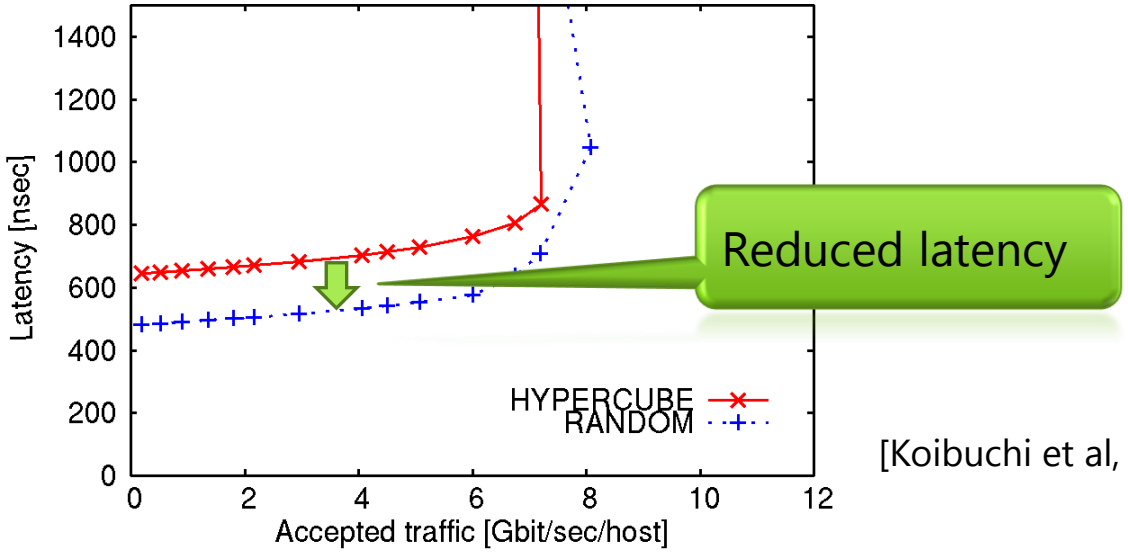


Toward low-latency network era



Non-random topology (tori, fat-tree, etc.) in current HPC & DC networks

Random shortcut topology (ring + random shortcuts)



[Koibuchi et al, ISCA 2012]

But, Cabling is Enormous

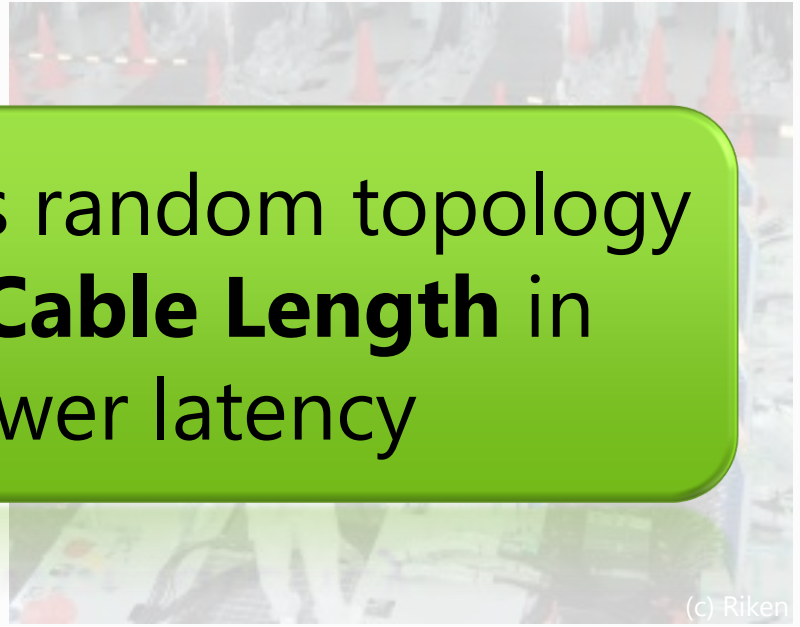
- Even for non-random topology…

Earth Simulator, 1st gen. (crossbar)

83,200 cables
2,400 km
140 tons

K Computer (6-D mesh/torus)

200,000 cables
1,000 km



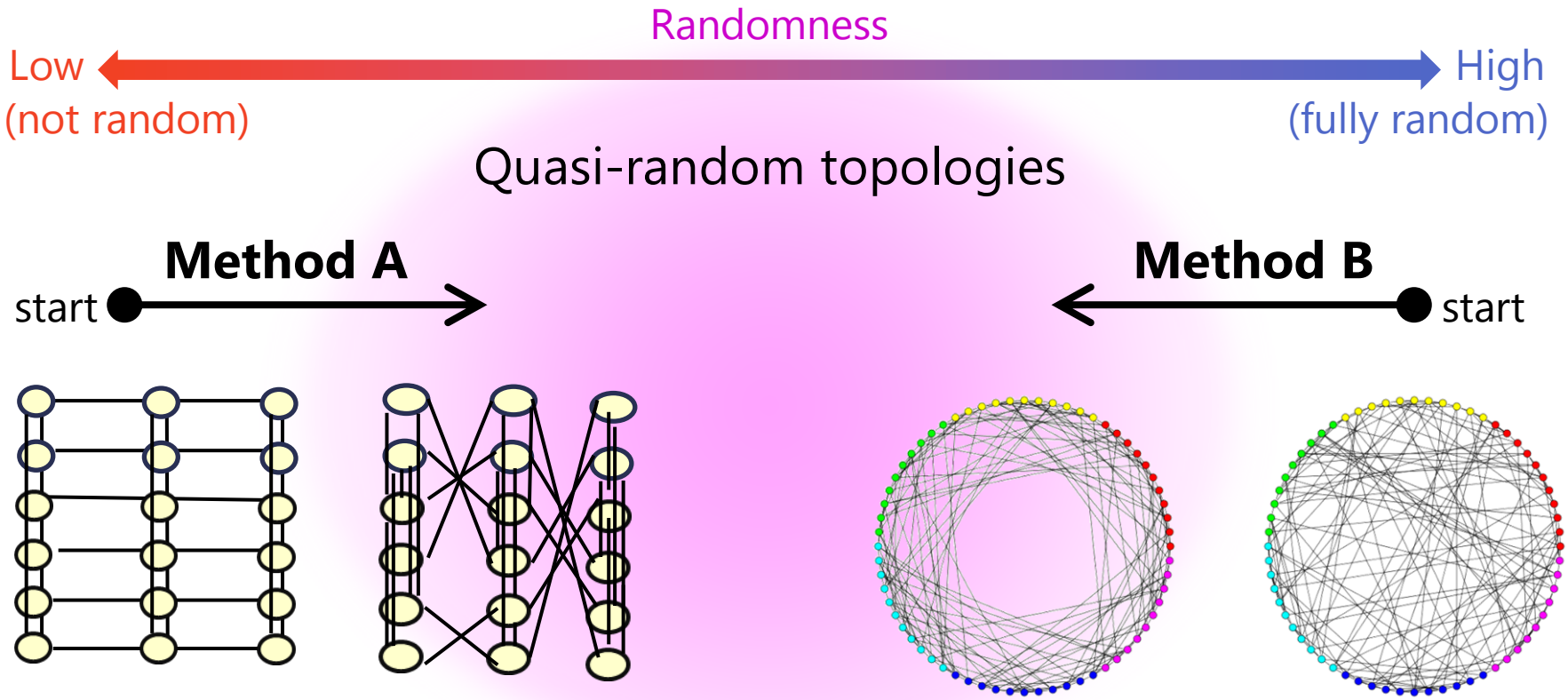
Our layout-conscious random topology provides **Shorter Cable Length** in addition to lower latency

Outline

- Motivation
- **Layout-conscious random topologies**
  - (method A) link permutation
  - (method B) constrained random shortcutting
- Comparison of both methods
- Conclusions

Two Approaches to Quasi-randomness

- **Method A** makes a non-random topology random
- **Method B** makes a random topology layout-friendly

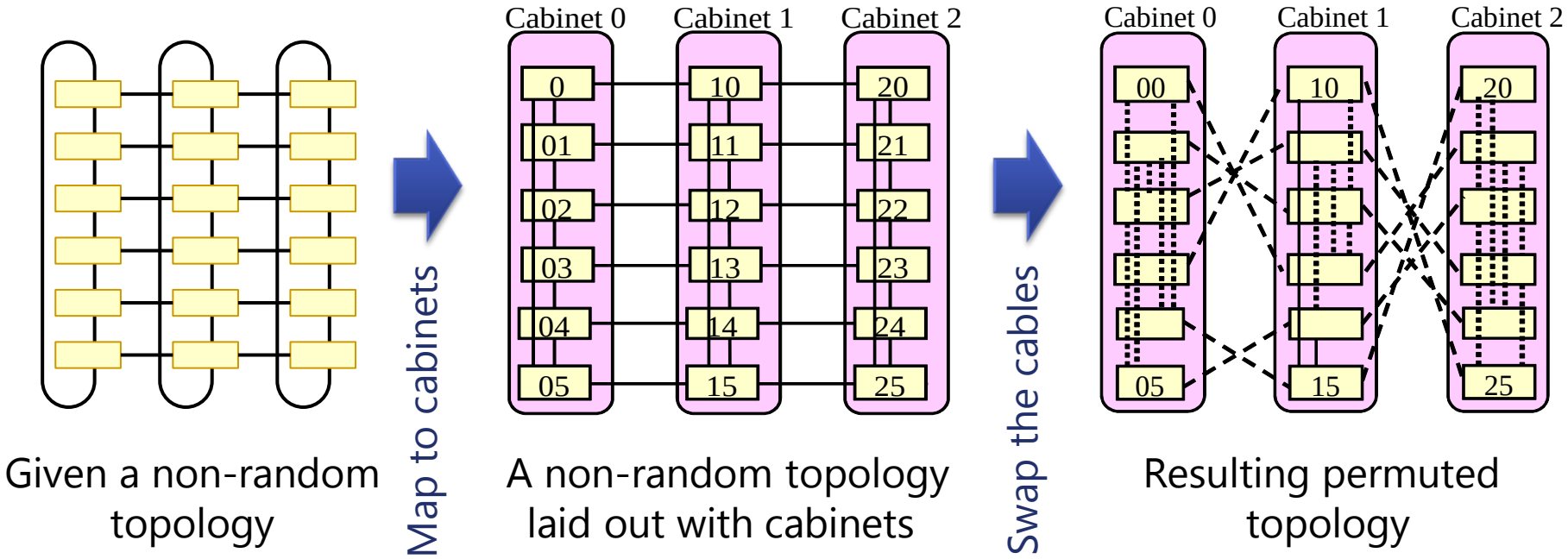


Outline

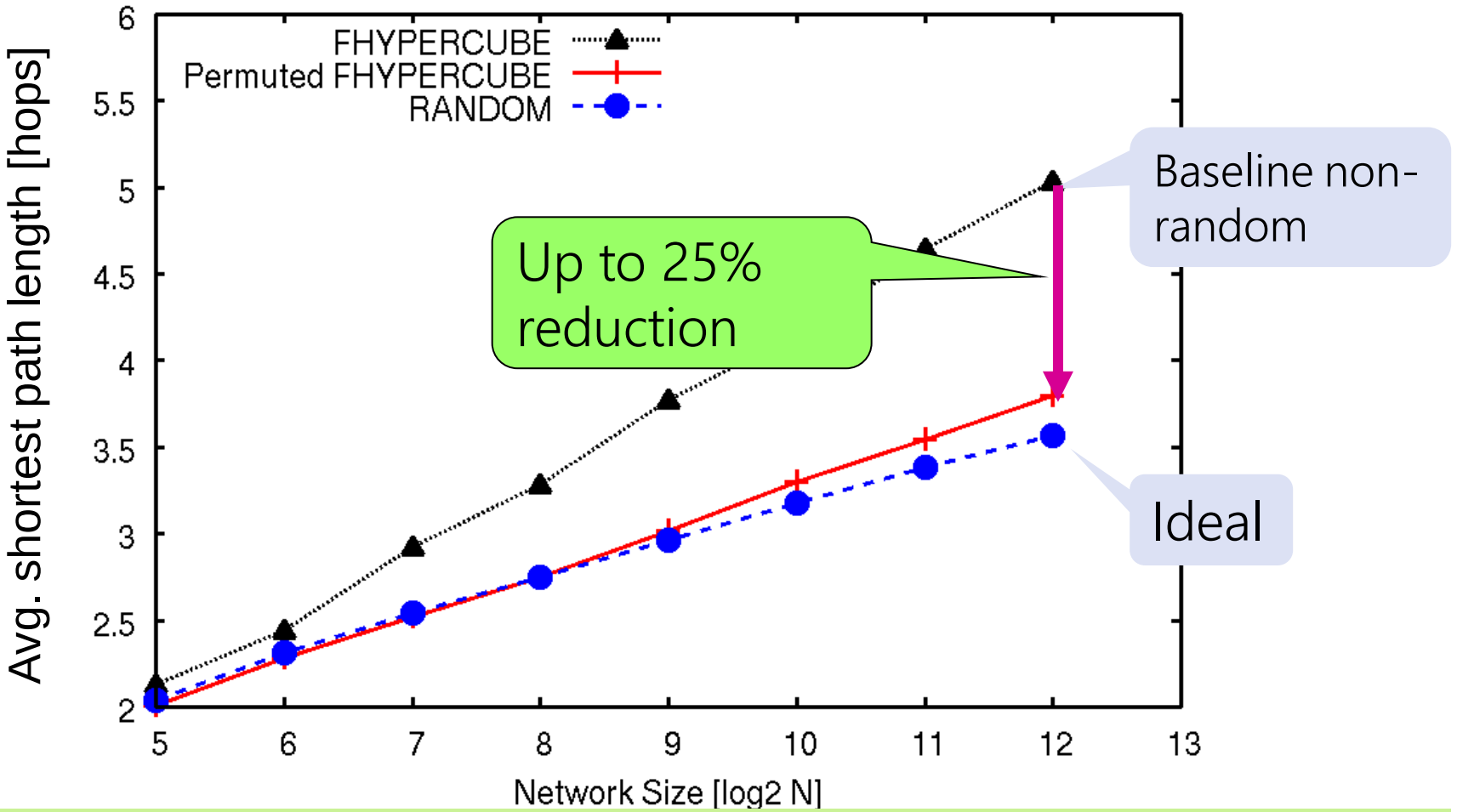
- Motivation
- **Layout-conscious random topologies**
  - **(method A) link permutation**
  - (method B) constrained random shortcutting
- Comparison of both methods
- Conclusions

Method A: Link Permutation

- Typical non-random topologies have great layout
- Randomly **swap the ends of cables** on its layout
  - Maintaining the great layout and the lengths of cables



Path Hops vs. Network Size

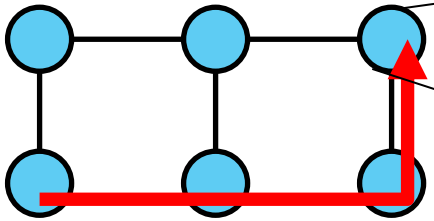


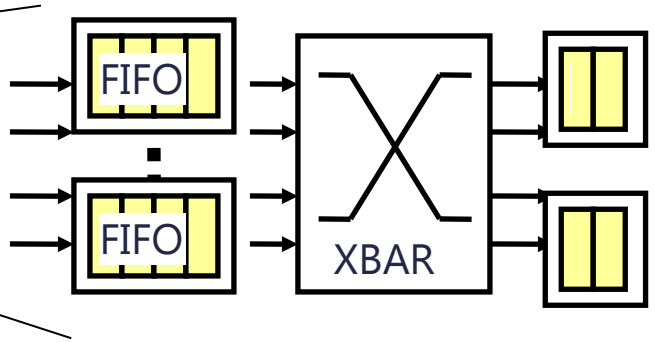
Permuted random topology provides better path hops
- Permuted_tori, and Permuted_Hypercube also do

14

Network Simulation

How many cycles ?





Cycle-accurate network simulation

Comm. Latency and Throughput

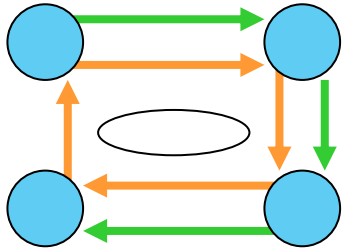
Switch structure

Switch & network parameters

Packet length	33 flits (1 flit = 256 bit)
Switching technique	Virtual-cut through
Traffic Pattern	Uni, Matrix-t, Bit-rev
Number of VCs	2
Switch delay	>100 ns
Link delay	20 ns

Topology & Routing

Mesh/Hypercube	Duato
Torus	DOR
Ring + Random	Irregular

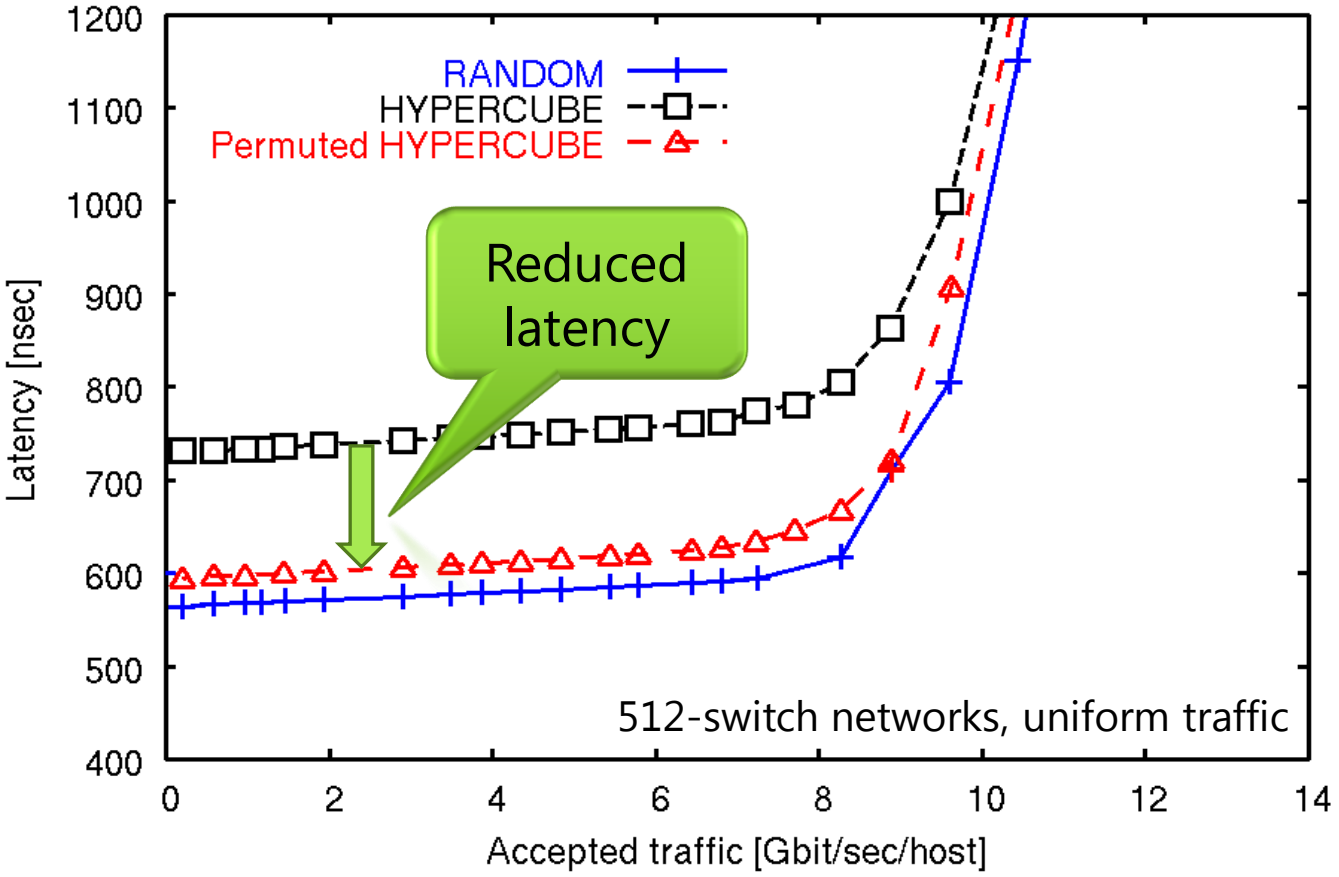


Applying Deadlock-free Routing theory

Adaptive channel

Escape channel

Latency vs. Throughput



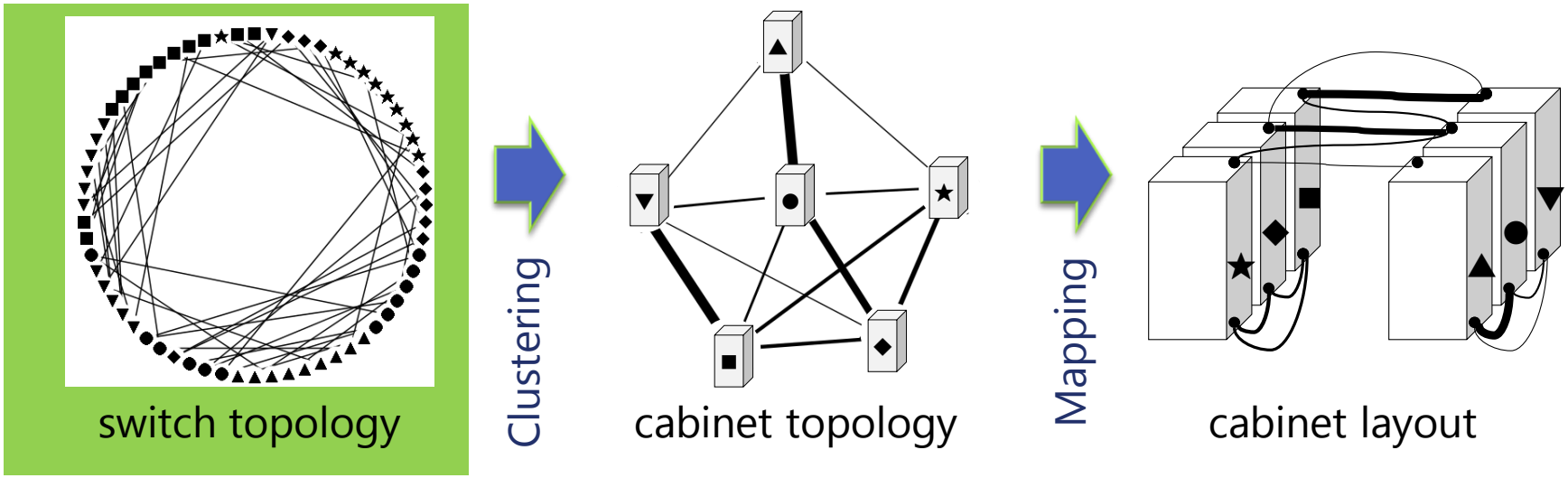
- Up to 20% reduced latency: close to full-random
- Similar results found with other traffic patterns and with other baseline topologies (e.g. fat tree)

Outline

- Motivation
- **Layout-conscious random topologies**
  - (method A) link permutation
  - **(method B) constrained random shortcutting**
- Comparison of both methods
- Conclusions

Method B: Constrained Shortcutting

- Optimize layout after randomizing topology

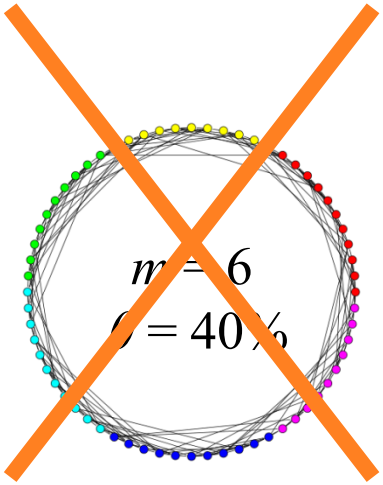


- Constrain the number of bypassed nodes when each random link is added to a ring
  - Virtually maintaining the diameter and the average shortest path hops of a fully random topology

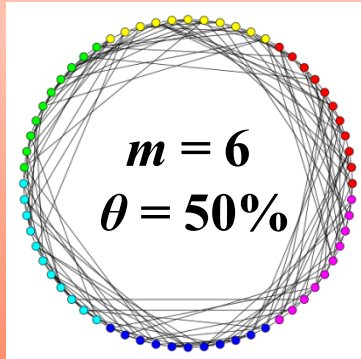
[Fujiwara et al, PDCAT 2012]

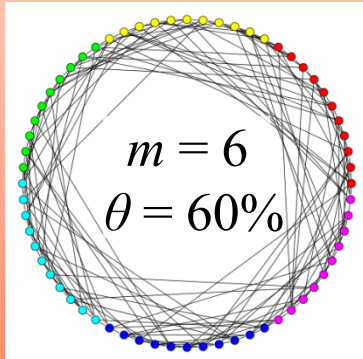
Way to Constrain the Shortcuts

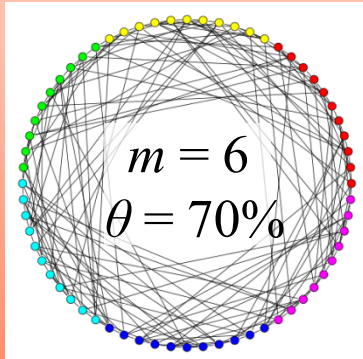
- *N* nodes, *m* degree, distribution θ
- Add $m-2$ shortcuts randomly to a ring ($m=2$)
- So that each **shortcut bypasses up to $\theta/2n$ nodes** along the ring
  - If θ = 100% then it is identical to a fully random topology



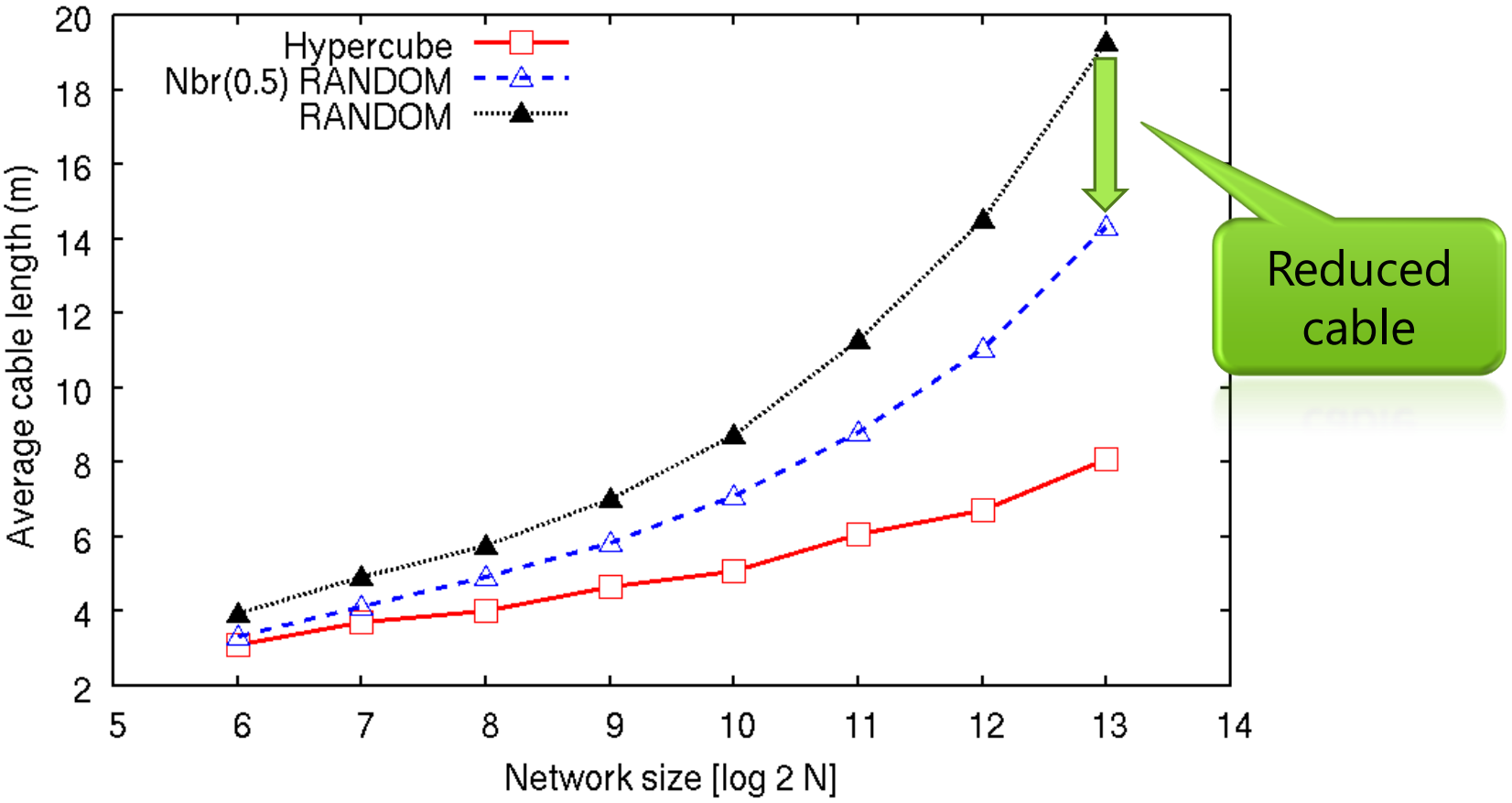
Low path hops close to θ = 100% (ideal)







Cable Length vs. Network Size



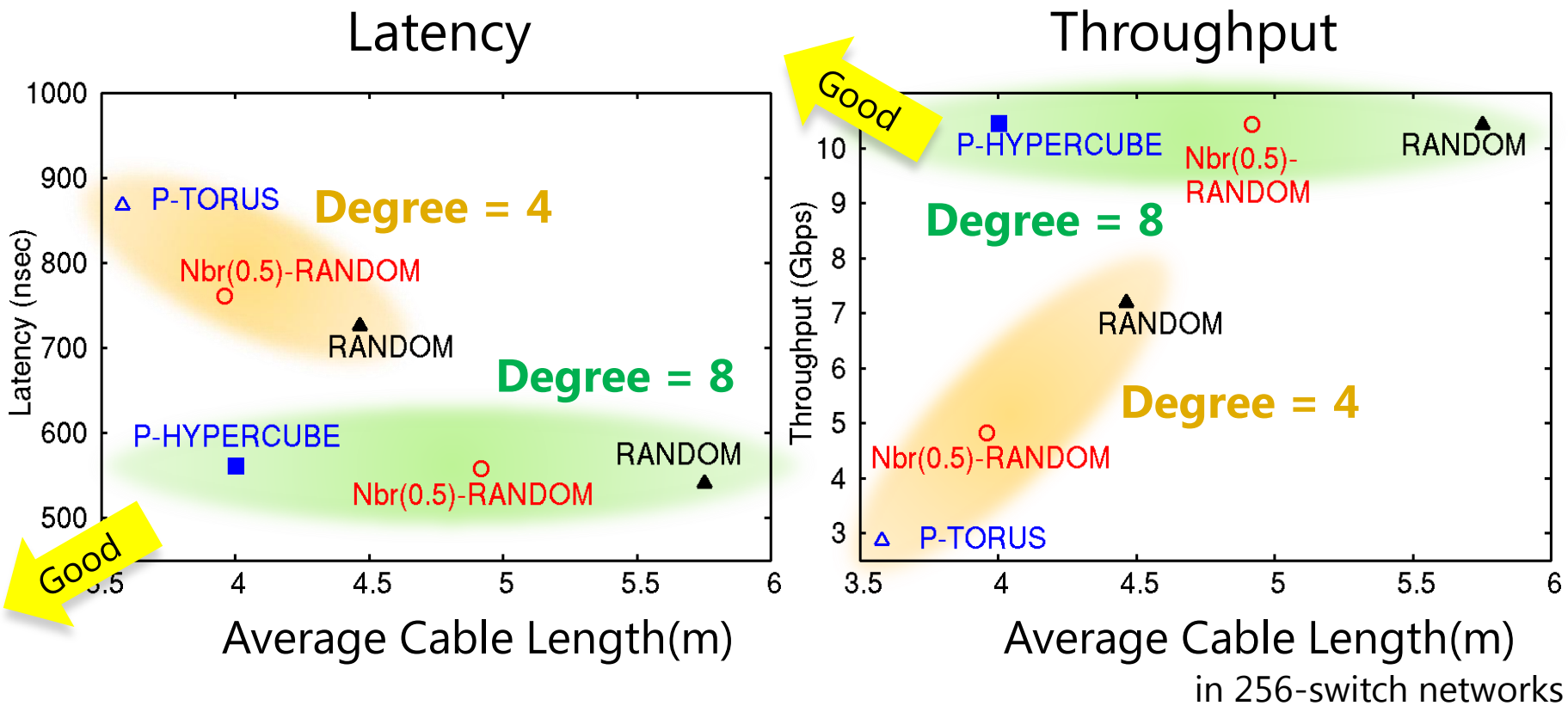
- Constrained shortcutting successfully reduced the cable length by up to 26%

Floorplan follows [ANSI/TIA/EIA-942] and [J.Kim, ISCA2007]

Outline

- Motivation
- Layout-conscious random topologies
  - (method A) link permutation
  - (method B) constrained random shortcutting
- **Comparison of both methods**
- **Conclusions**

Permutation vs. Constrained Shortcuts

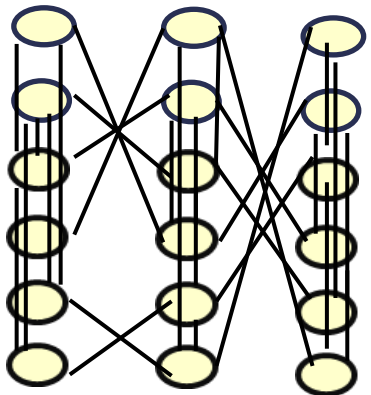


- **Link permutation** is usually recommended
- **Constrained shortcutting** (θ=50%) is better in low-radix networks

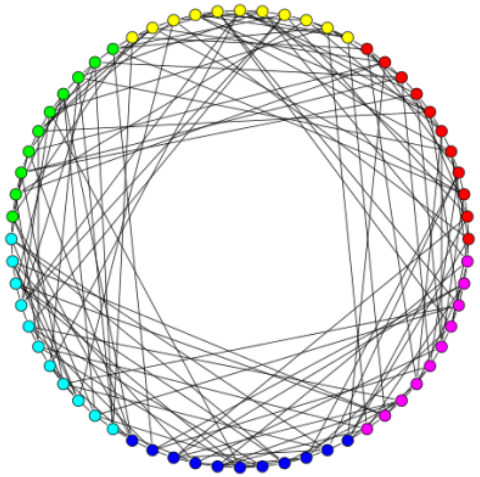
Conclusions

- Two randomizing methods for practical quasi-random topology in off-chip networks
  - **Link permutation** randomizes links after layout is fixed
  - **Constrained shortcutting** optimizes layout after randomizing topology

A resulting quasi-random topology has
- **Shorter cable length** (≒tori/hypercube)
- **Low path hops** (≒ fully random)



Link permutation



Constrained shortcutting

Thank you!

Routing Computation Cost

- Address and routing-table size at switch
  - InfiniBand LID: 48k
  - General issue regardless of topology
- Computational cost of path search
  - Topology-agnostic deadlock-free routing [Flich,TPDS2012]
    - O ((N+E) logN) priority-queue Dijkstra algo.