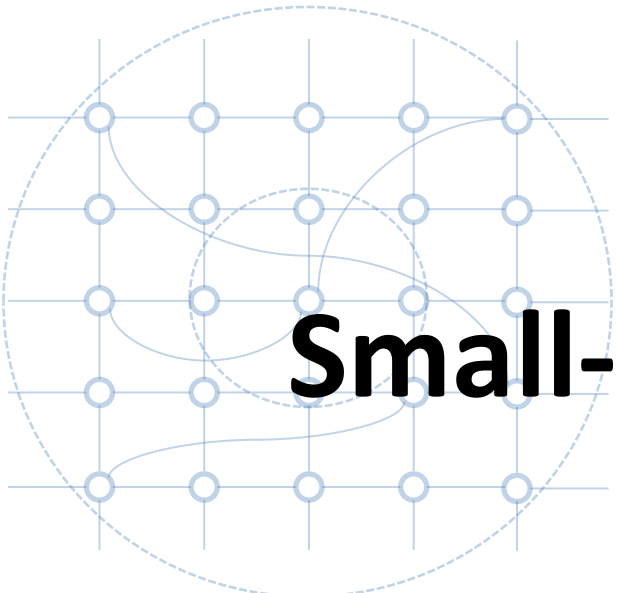


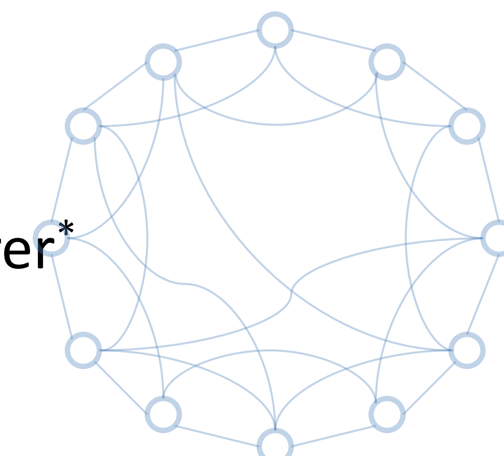
Small-World Datacenters

Ji-Yong Shin[*]

Bernard Wong[+], and Emin Gün Sirer[*]

[*]Cornell University

[+]University of Waterloo



Motivation

- Conventional networks are hierarchical
  - Higher layers become bottlenecks
- Non-traditional datacenter networks are emerging
  - Fat Tree, VL2, DCell and BCube
    - Highly structured or sophisticated regular connections
    - Redesign of network protocols
  - CamCube (3D Torus)
    - High bandwidth and APIs exposing network architecture
    - Large network hops

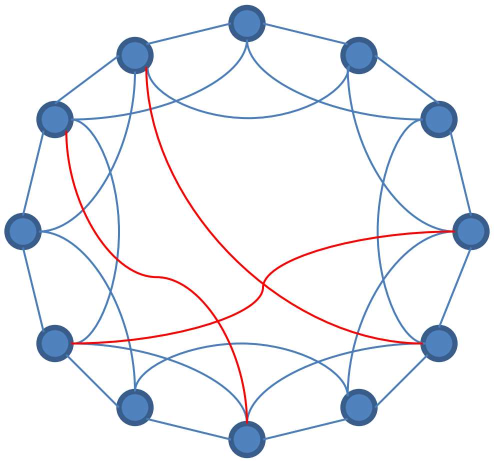
Small-World Datacenters

- Regular + random connections
  - Based on a simple underlying grid
  - Achieves low network diameter
  - Enables content routing

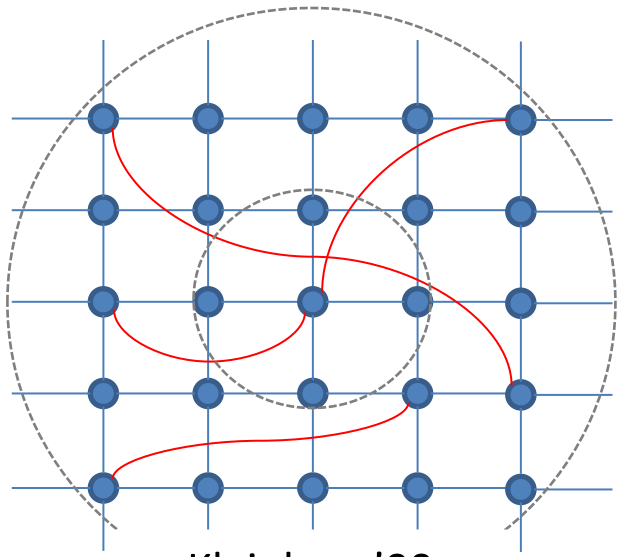
- Characteristics
  - High bandwidth
  - Fault tolerant
  - Scalable

Small-World Networks

- Watts and Strogatz
  - Multiple connections to neighbors on a ring + random connections
- Kleinberg
  - Lattice + random links
  - Probability of connecting a random pair decreases with *d*th power of distance between the pair in *d*-dimensional network
  - Path length becomes O(log n)



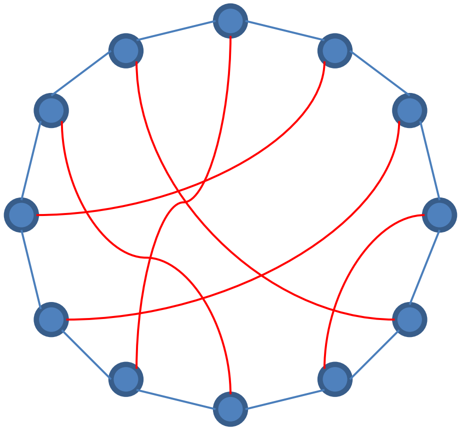
Watts and Strogatz'98



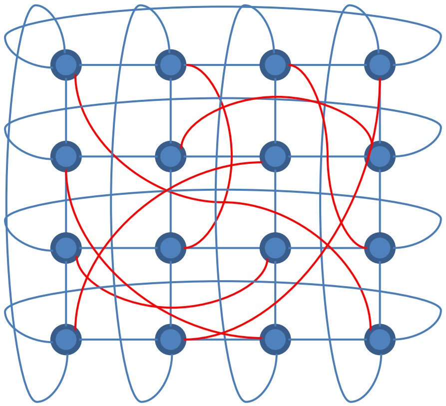
Kleinberg'00

Small-World Datacenter Design

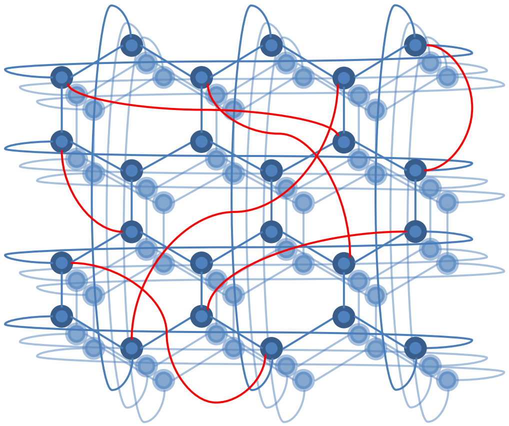
- Possible topologies for 6 links per node



Small World Ring
(2 reg + 4 rand)



Small World 2D Torus
(4 reg + 2 rand)



Small World 3D Hexagon Torus
(5 reg + 1 rand)

- Direct connections from server to server
  - No need for switches
  - Software routing approach

Routing in Small-World Datacenters

- Shortest path
  - Link state protocol (OSPF)
  - Expensive due to TCAM cost
- Greedy geographical
  - Find min distance neighbor
  - Coordinates in lattice used as ID
  - Maintain info of 3 hop neighbors
  - Comparable to shortest path for 3DHexTorus

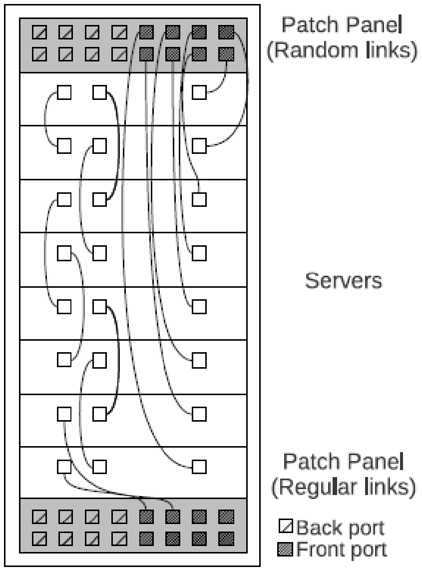
Content Routing in Small-Worlds

- Content routing
  - Logical coordinate space and network to map data
  - Logical and physical network do not necessarily match
- Geographical identity + simple regular network in SWDC
  - Logical topology can be mapped physically
  - Random links only accelerates routing
- SWDC can support DHT and key value stores directly
  - Similar to CamCube

Packaging and Scaling

- SWDCs can be constructed from preconfigured, reusable, scalable components
- Reusable racks
  - Regular links: only short cables necessary
  - Random links:
    - Predefined Blueprint
    - Random number generator
    - Pre-cut wires based on known probability

*



- Ease of construction
  - Connect rack-> cluster (or container) -> datacenter
  - Switches, repeaters, or direct wires for inter-cluster connections

*Image from http://www.chuck-a-con.net/Dice.html

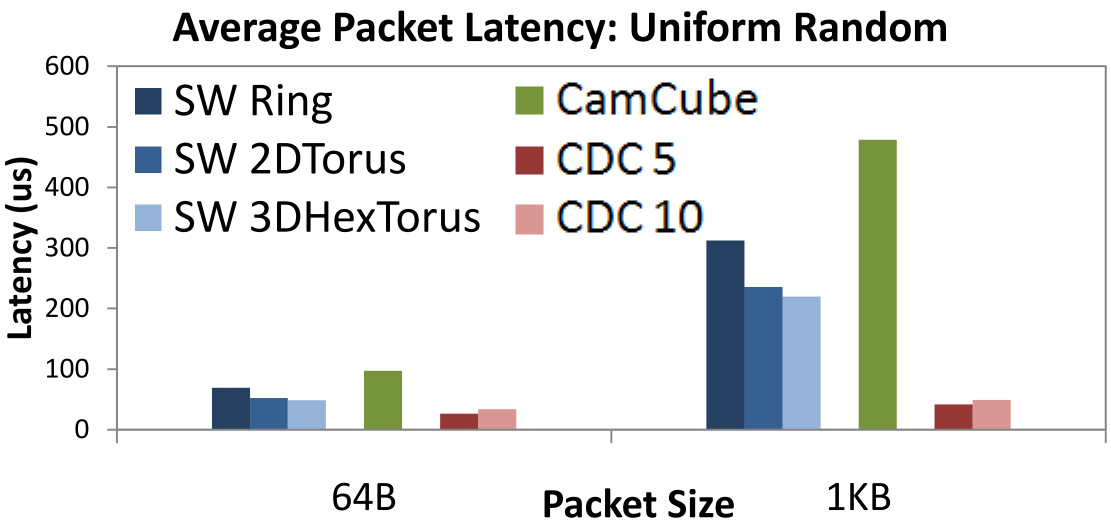
Evaluation Setup

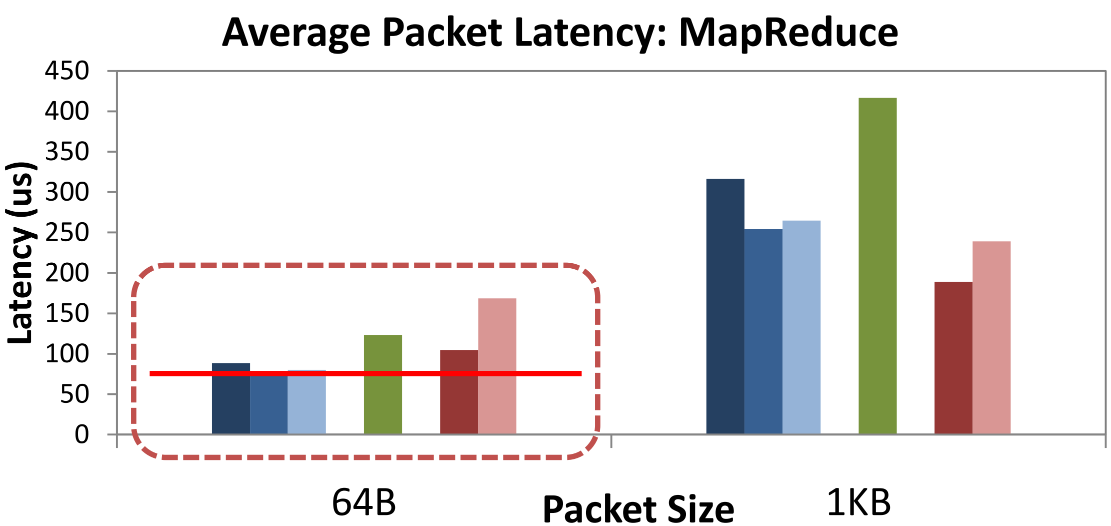
- Simulation of 10,240 nodes in three settings:
  - **Small-World Datacenters (SWDC)**
  - **CamCube**

SW Ring SW 2DTorus SW 3DHexagonal Torus CamCube	• 6 x 1GigE links per server • Greedy routing • NetFPGA Setup – 64B packet 4.6 us – 1KB packet 15 us

  - **Conventional hierarchical data centers (CDC)**
    - 1 x 1GigE link per server
    - 10 GigE links among switches
    - 3 layer switches (Uniform delays: 6us, 3.2 us, and 5us in each layer)
    - Oversubscriptions: 5 and 10

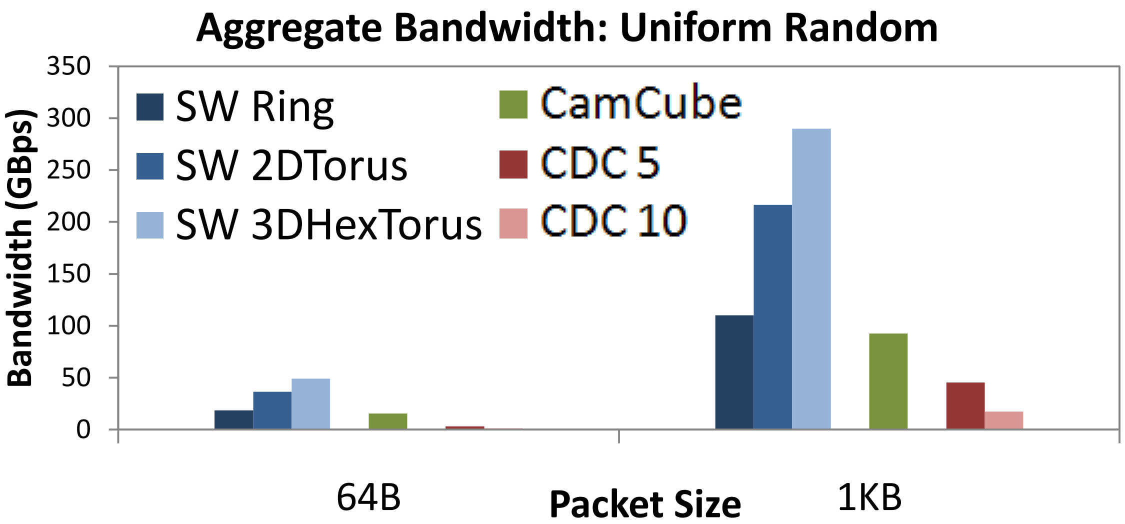
Evaluation: Average Packet Latency

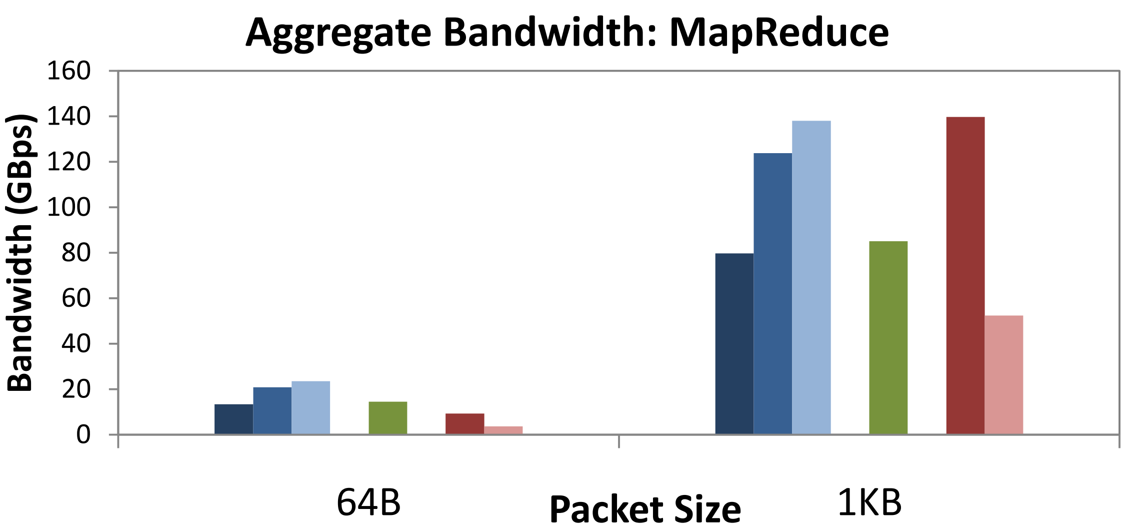




- SWDCs always outperform CamCube
- SWDCs can outperform CDC for MapReduce
  - SWDC has multiple ports
- SWDC latencies are packet size dependent
  - Limitations of software routers

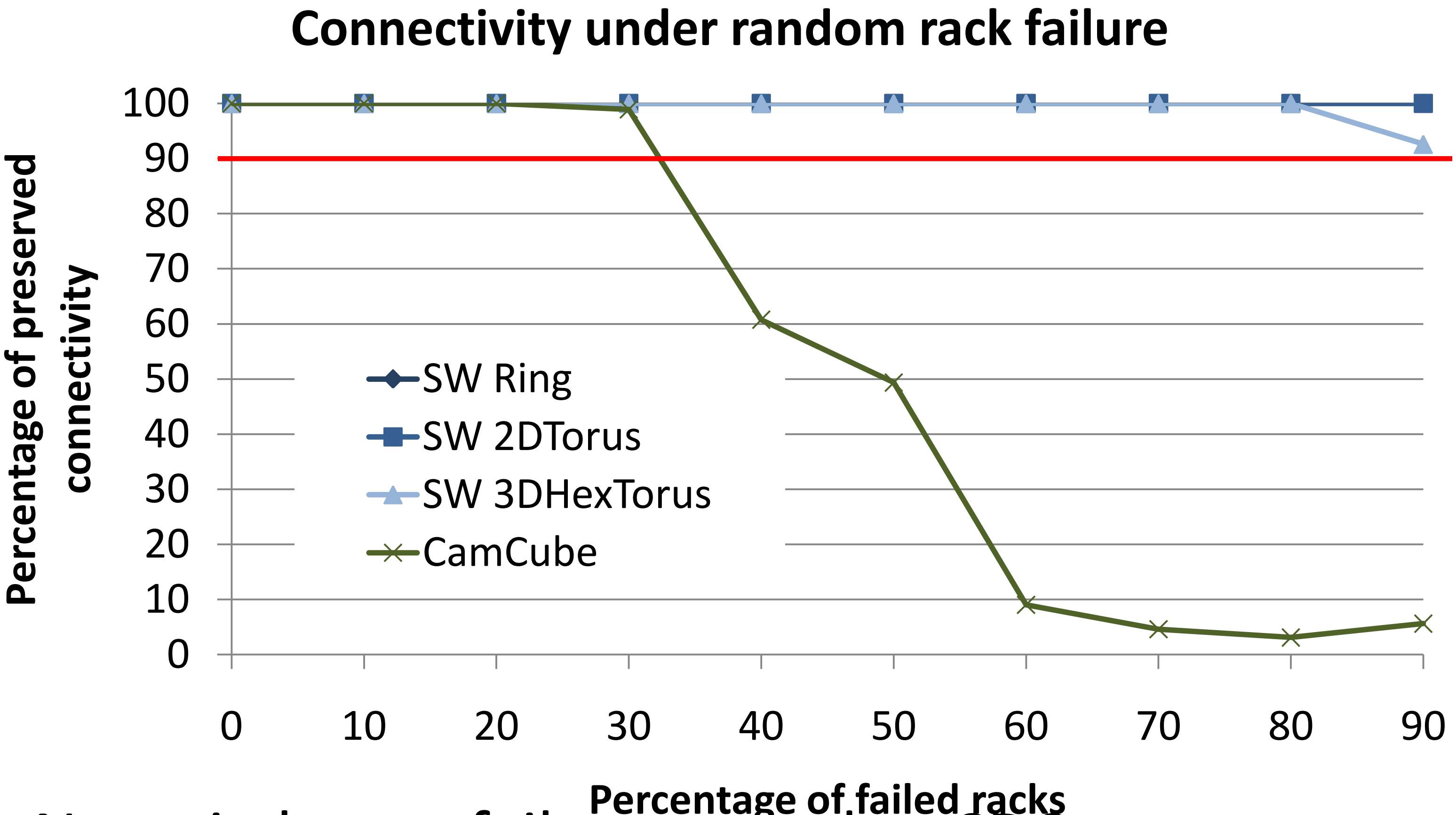
Evaluation: Aggregate Bandwidth





- SWDCs outperform CamCube in general
  - 1.5x - 3x better
- SWDCs outperform CDCs in general
  - 1x - 16x better

Evaluation: Fault Tolerance



- No switches to fail compared to CDCs
- Random links enable stronger connections

Related Concurrent Work

- Scafida and Jellyfish
  - Rely on random connections
  - Achieve high bandwidth

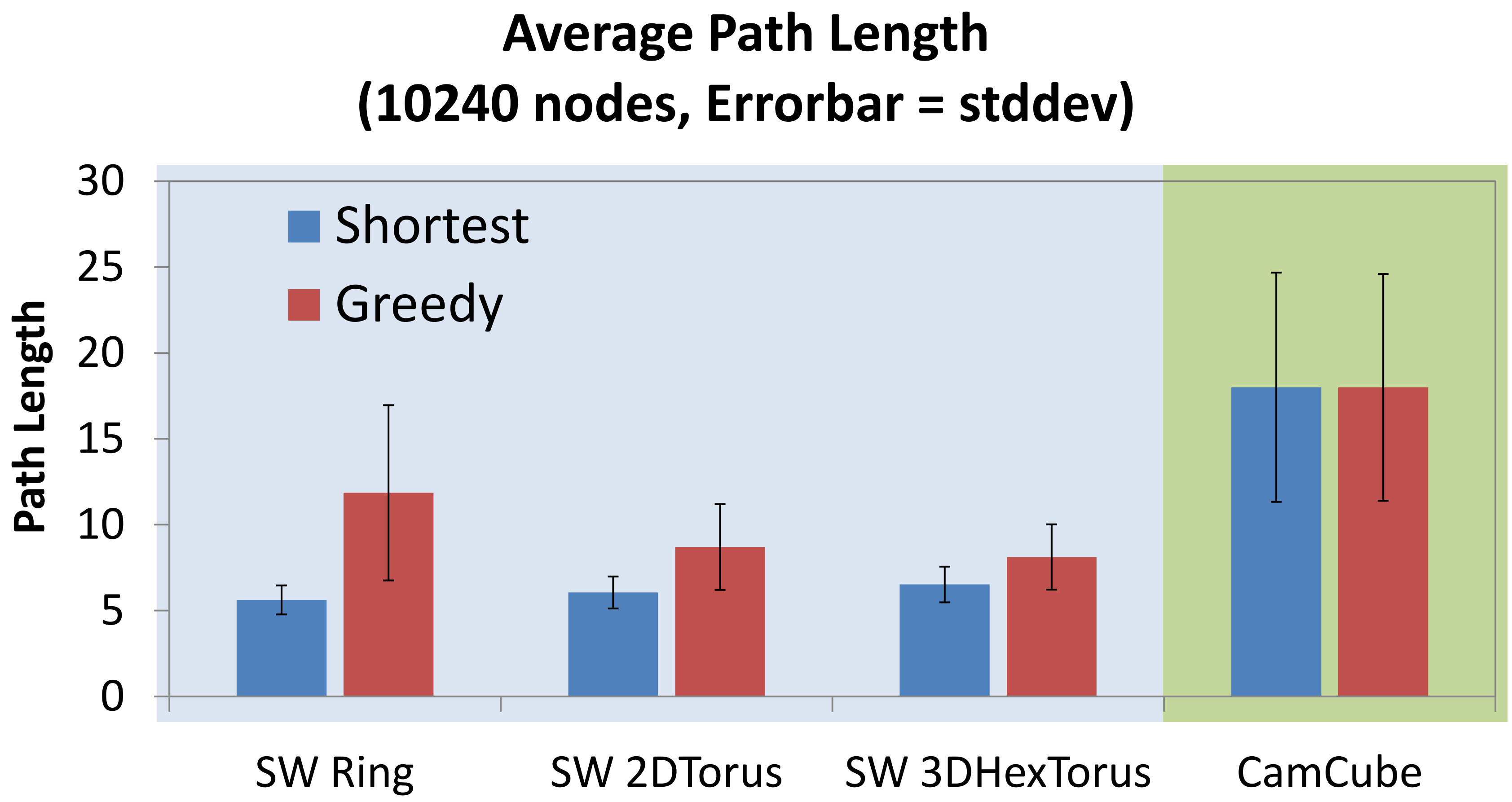
- Comparison to SWDC
  - SWDCs have more regular links
  - Routing can be simpler

Summary

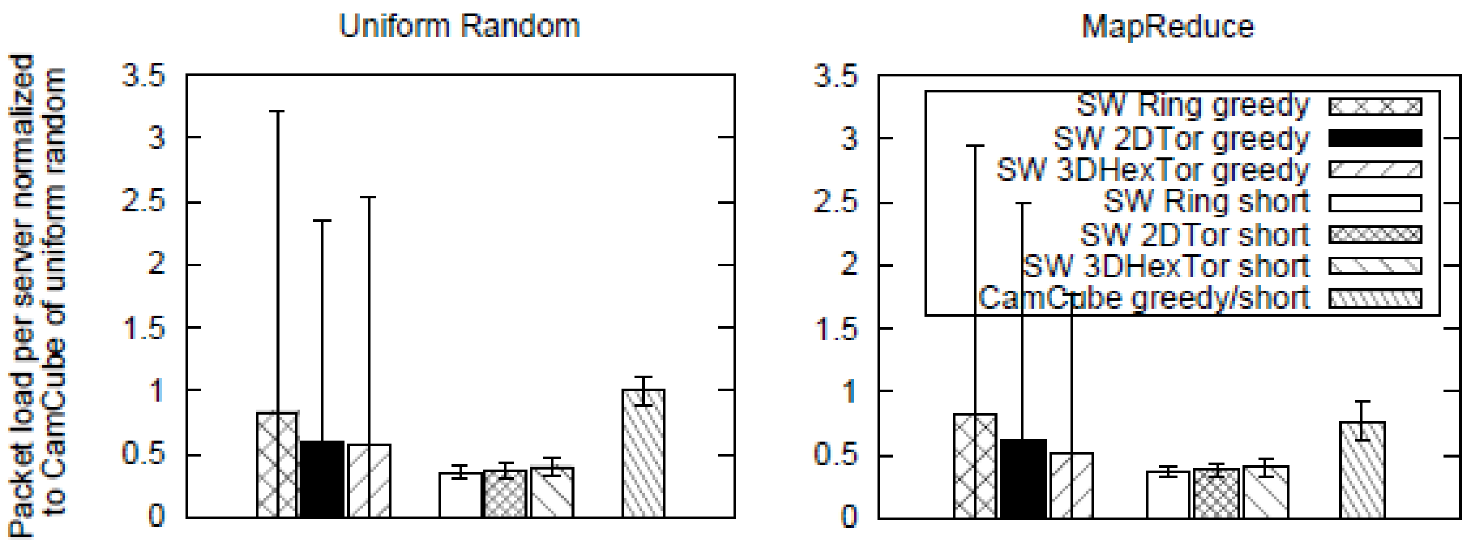
- Unorthodox topology comprising a mix of regular and random links can yield:
  - High performance
  - Fault tolerant
  - Easy to construct and scalable

- Issues of cost at scale, routing around failures, multipath routing, etc. are discussed in the paper

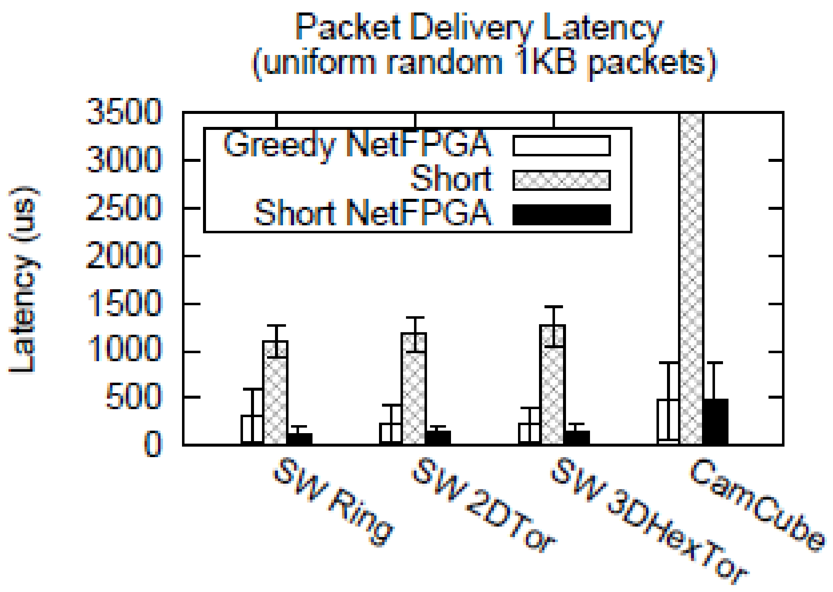
Extra: Path Length Comparison

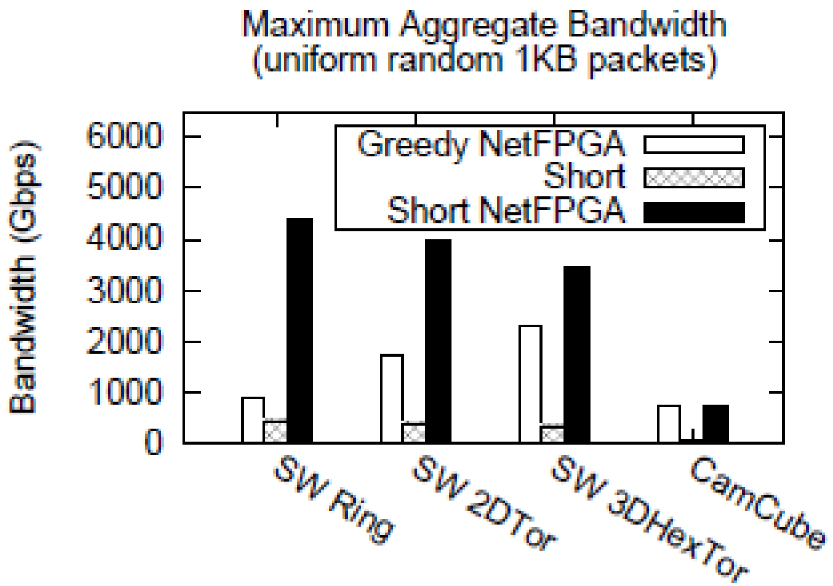


Extra: Load balance

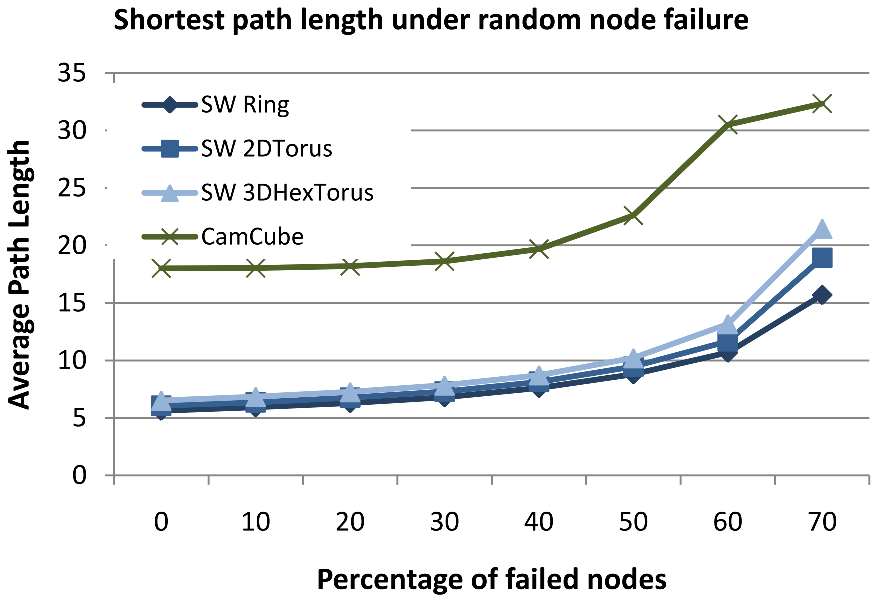


Extra: Need for Hardware

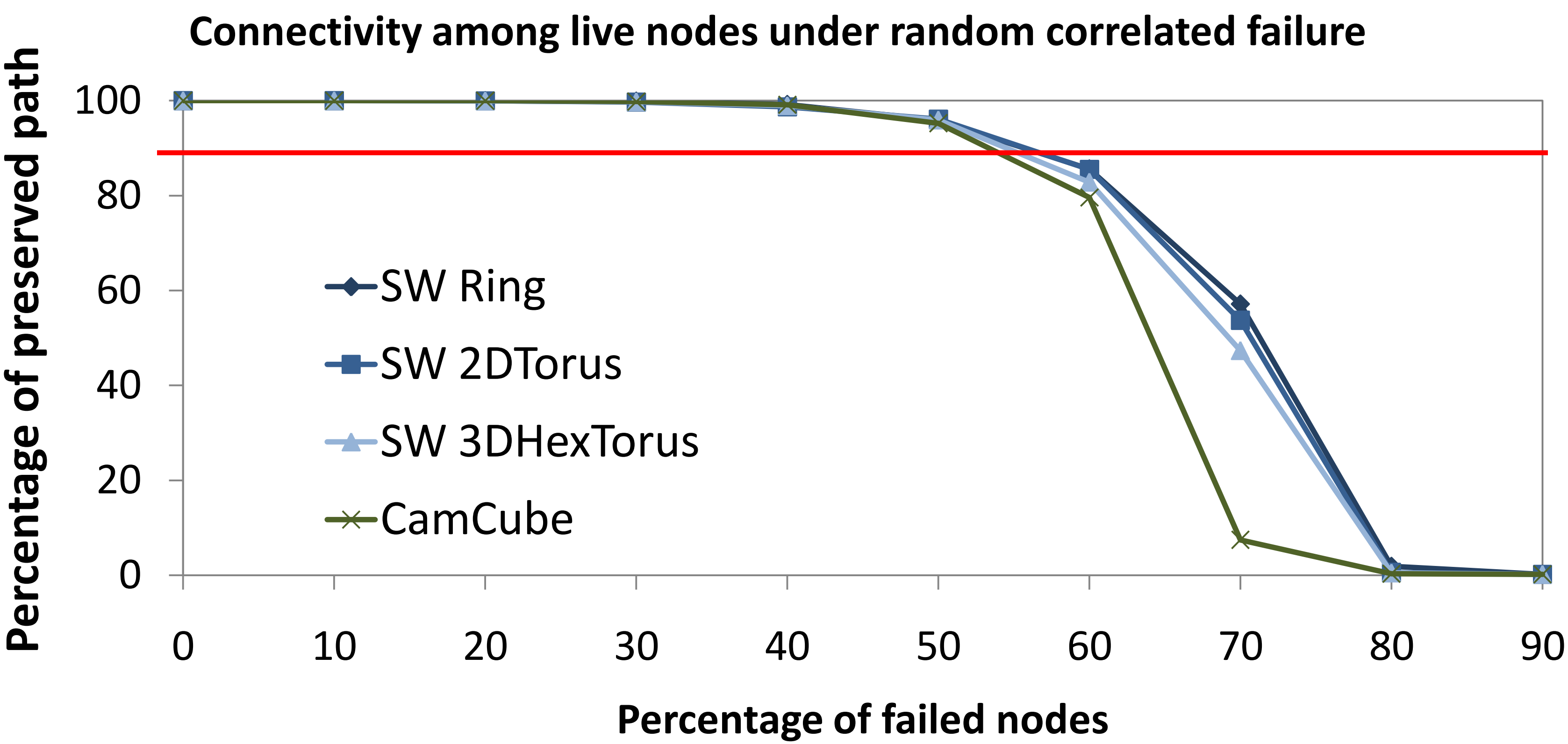




Extra: Path Length Under Failure



Extra: Fault Tolerance (node failure)



- No switches to fail compared to CDCs
- Random links enable stronger connections