High Performance Computing Systems (CMSC714)

Lecture 10: Fat-tree and Dragonfly Networks

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

Summary of last lecture

- Key requirements of HPC networks
  - extremely low latency, high bandwidth, scalable
  - low network diameter, high bisection bandwidth
- Torus networks (less common now)
  - Network diameter grows as $O(\sqrt[3]{N})$ where N is the number of nodes
- Different types of routing algorithms:
  - Shortest path vs. non-minimal
  - Static vs. dynamic

Fat-tree network

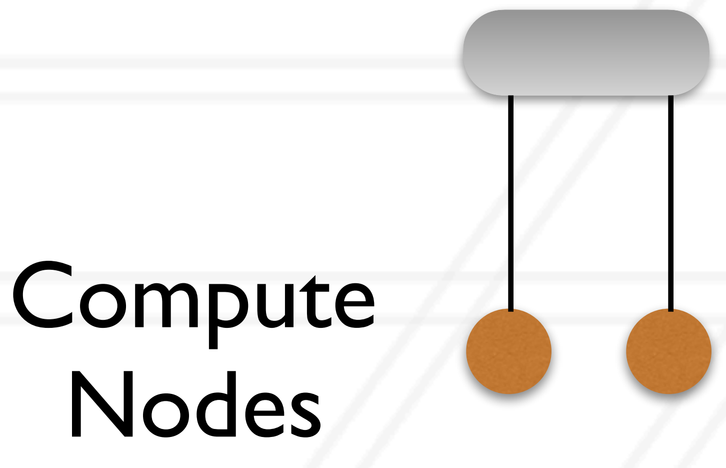
- Most popular network topology
  - Low network diameter, high bandwidth

Fat-tree network

- Most popular network topology
  - Low network diameter, high bandwidth

Fat-tree network

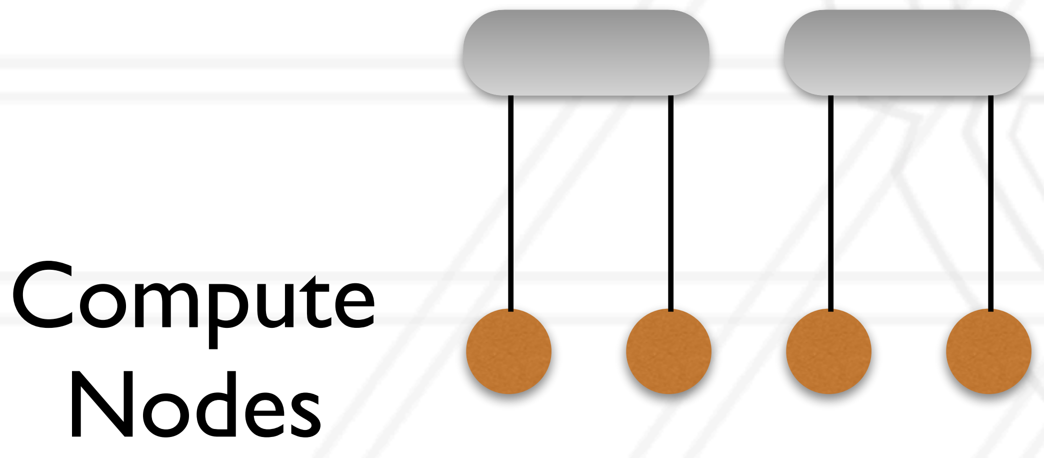
- Most popular network topology
  - Low network diameter, high bandwidth



Router/switch radix = number of ports = k

Fat-tree network

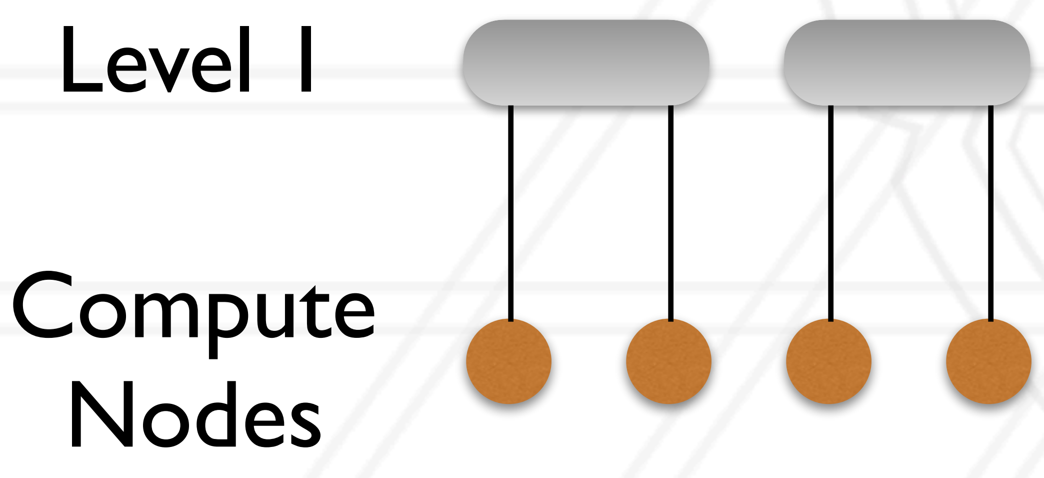
- Most popular network topology
  - Low network diameter, high bandwidth



Router/switch radix = number of ports = k

Fat-tree network

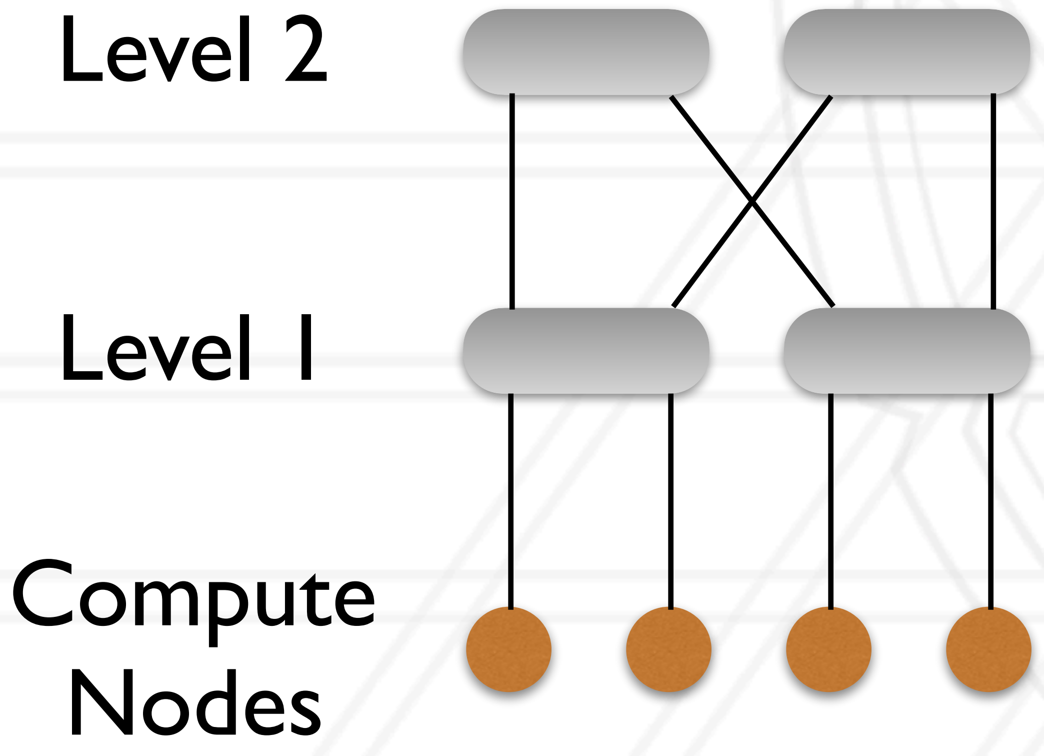
- Most popular network topology
  - Low network diameter, high bandwidth



Router/switch radix = number of ports = k

Fat-tree network

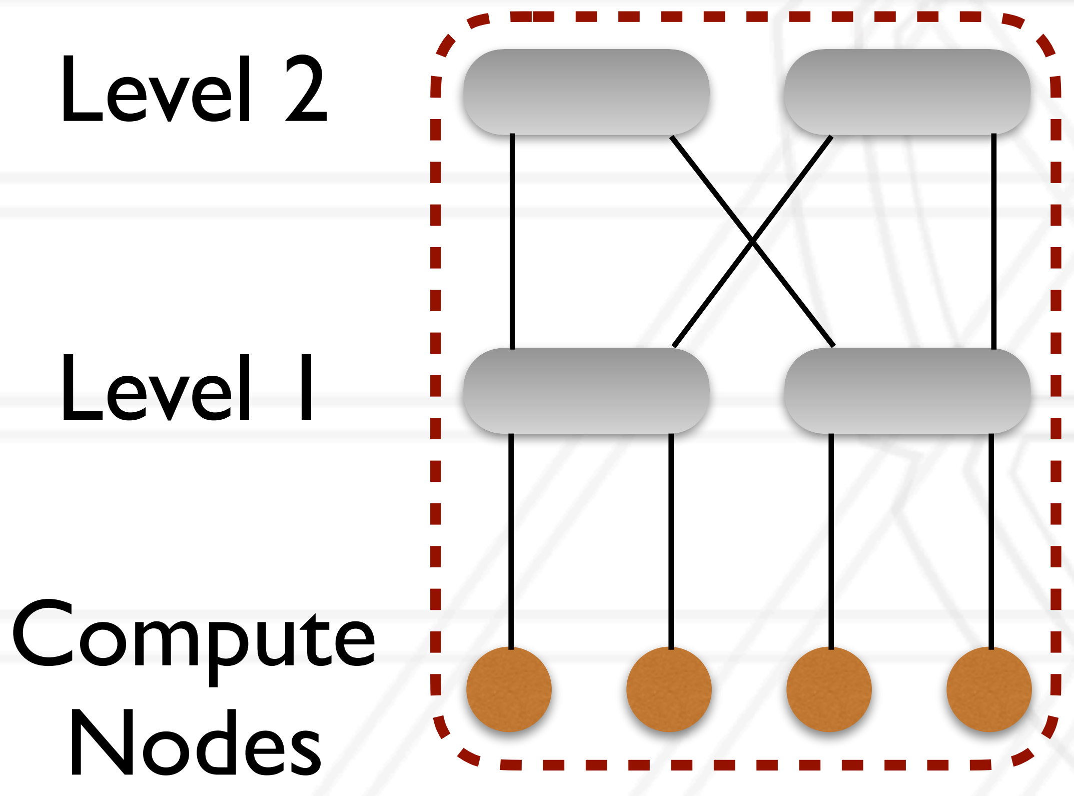
- Most popular network topology
  - Low network diameter, high bandwidth



Router/switch radix = number of ports = k

Fat-tree network

- Most popular network topology
  - Low network diameter, high bandwidth

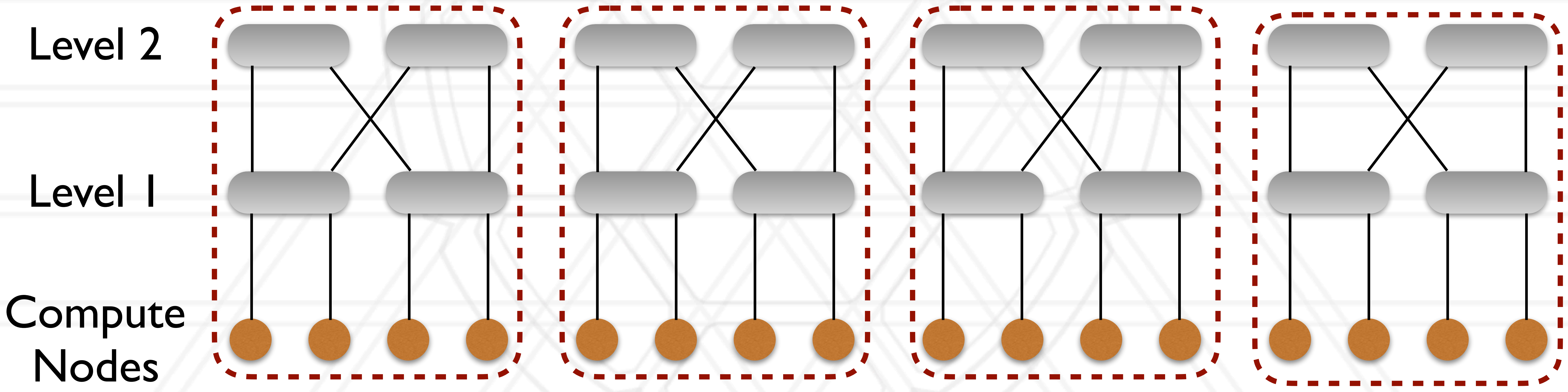


Router/switch radix = number of ports = k

Pod = group of switches = k/2 switches

Fat-tree network

- Most popular network topology
  - Low network diameter, high bandwidth

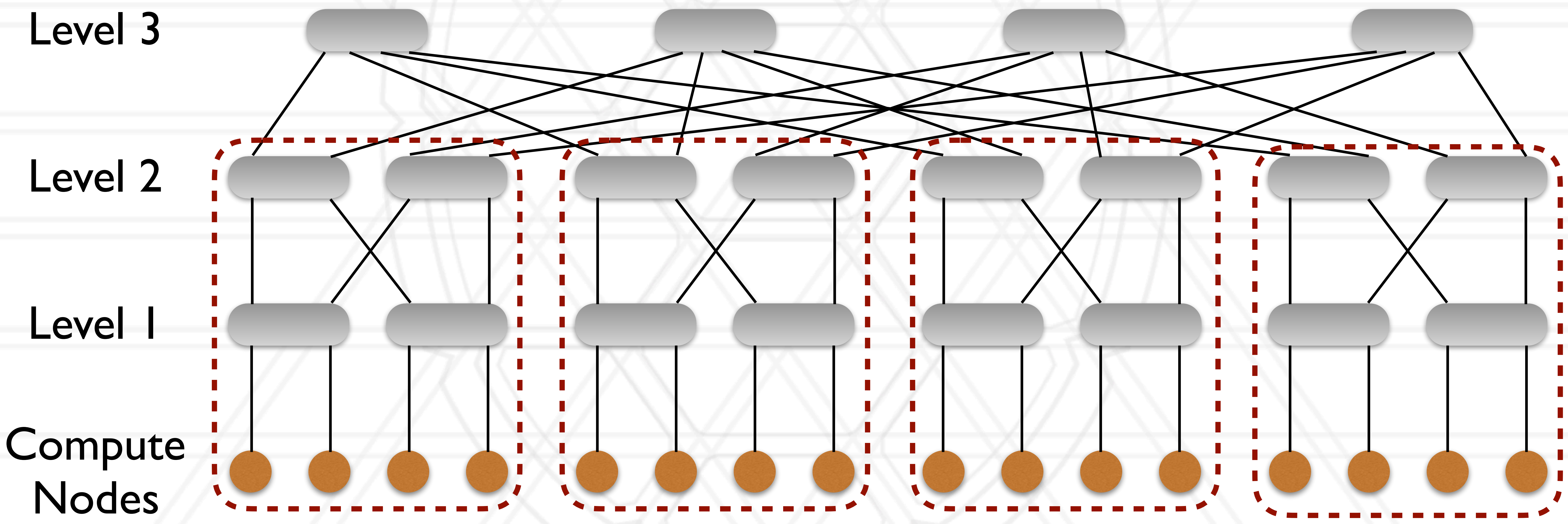


Router/switch radix = number of ports = k

Pod = group of switches = k/2 switches

Fat-tree network

- Most popular network topology
  - Low network diameter, high bandwidth

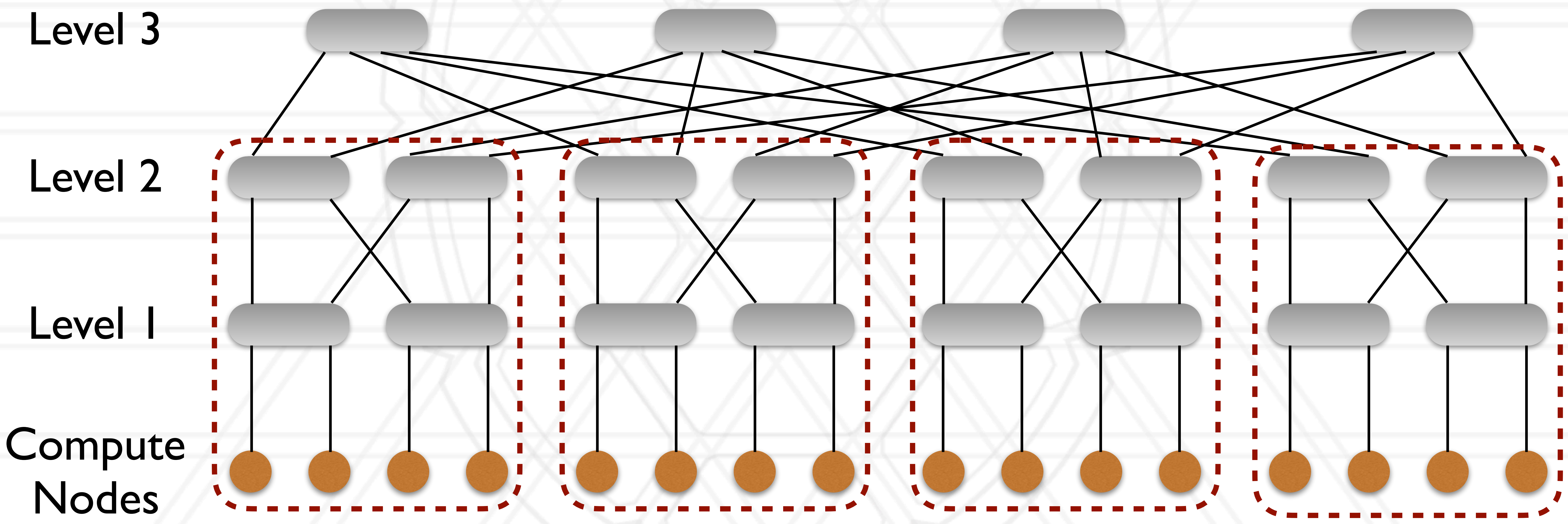


Router/switch radix = number of ports = k

Pod = group of switches = k/2 switches

Fat-tree network

- Most popular network topology
  - Low network diameter, high bandwidth

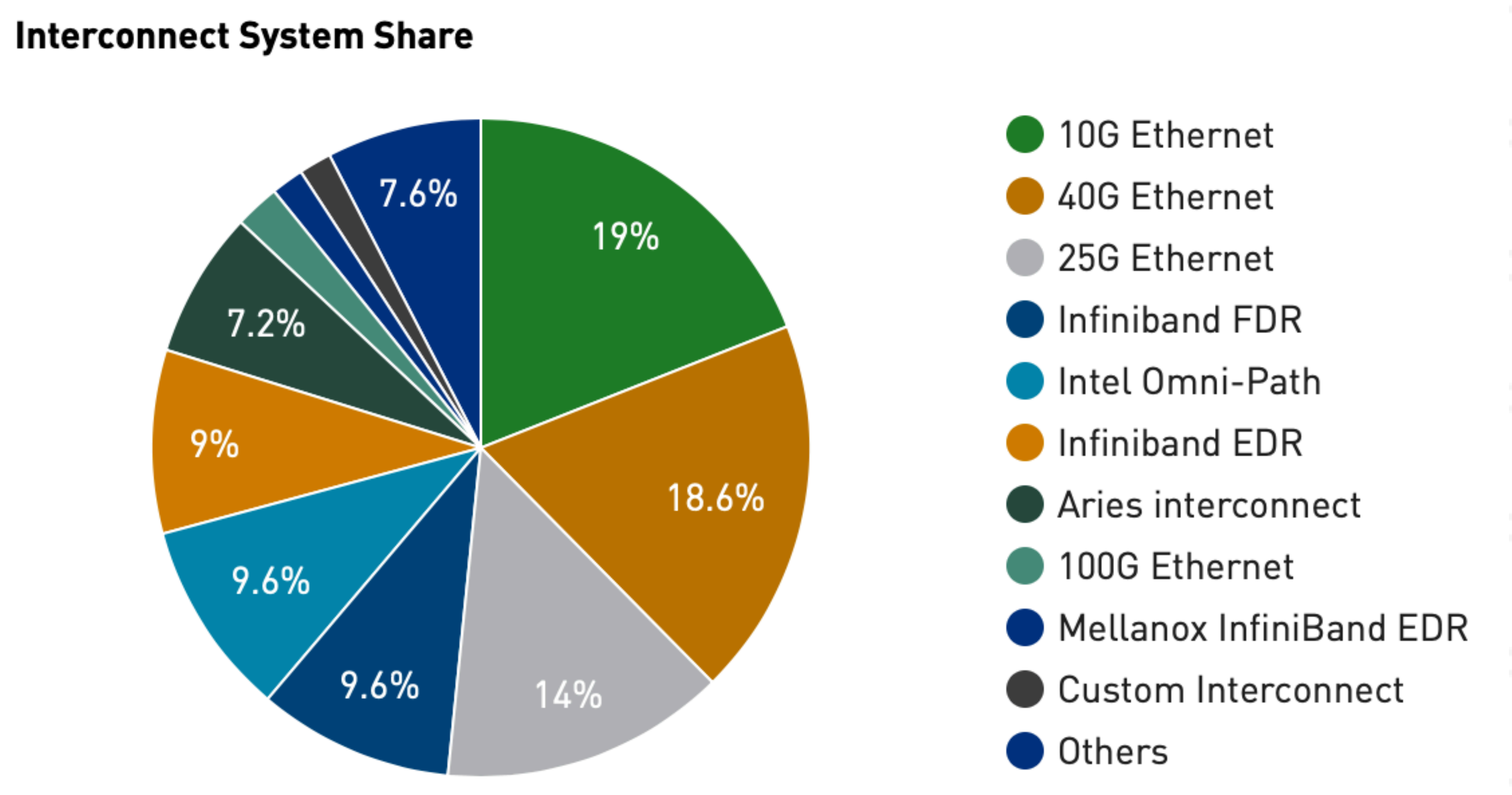


Router/switch radix = number of ports = k

Pod = group of switches = k/2 switches   Max. number of pods = k

Fat-tree networks on the top500 list

- Infiniband EDR/FDR
- Intel Omni-Path

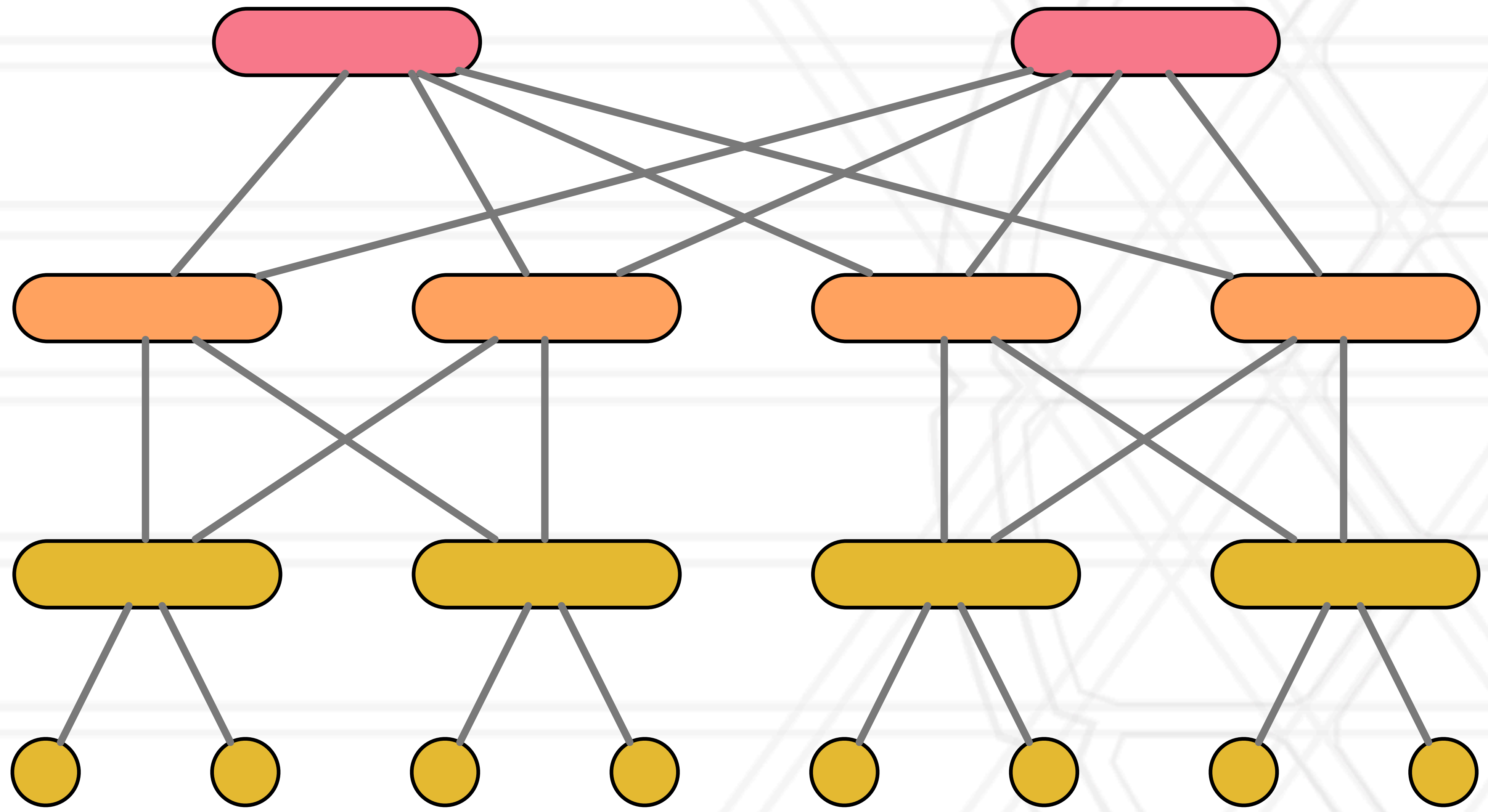


https://www.top500.org/statistics/list/

Routing on a fat-tree

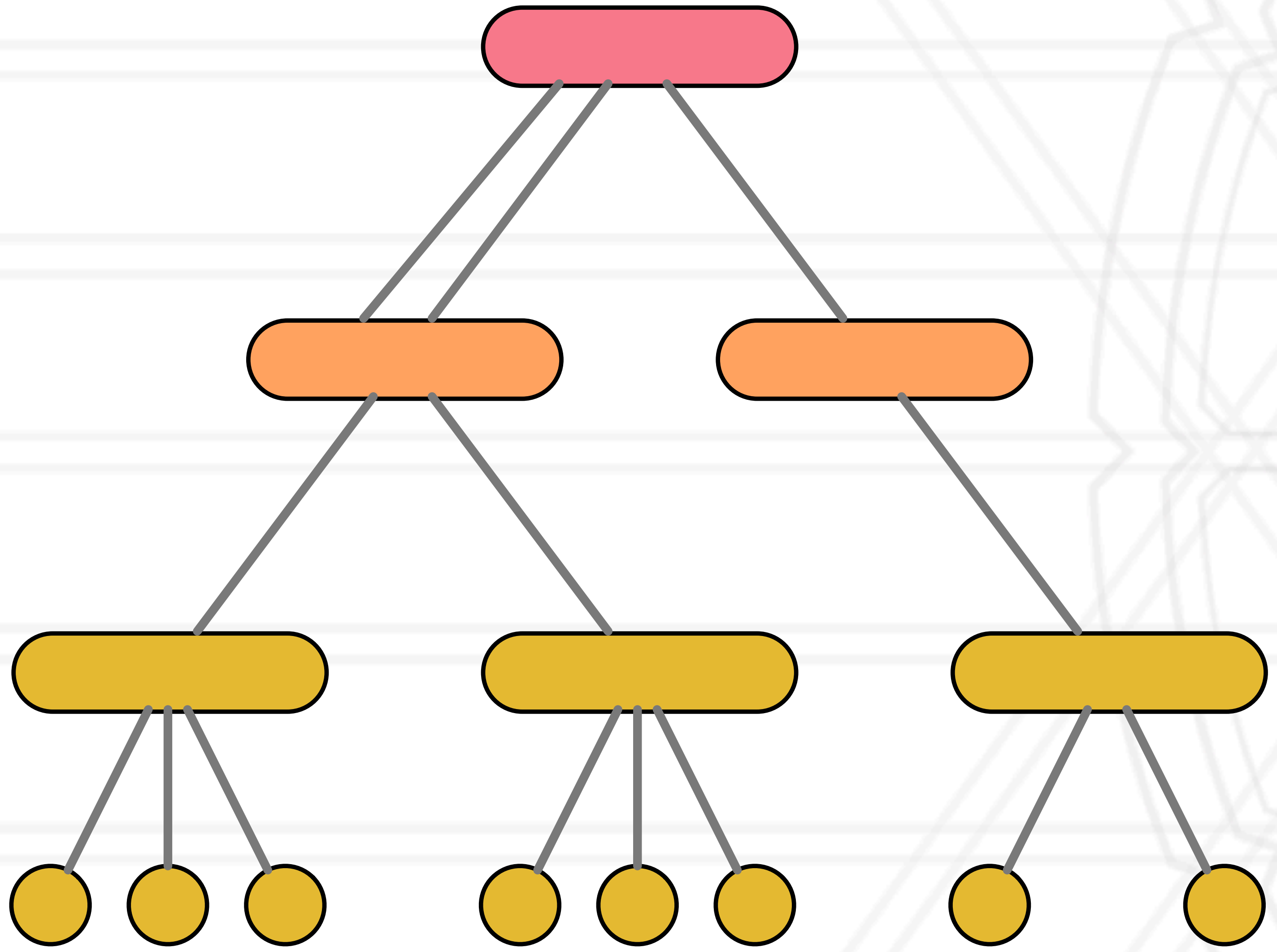
- Until recently, most fat-tree installations used static routing
  - Destination-mod-k (D-mod-k) routing
- Adaptive routing is now starting to be used

Variations on a full bandwidth fat-tree



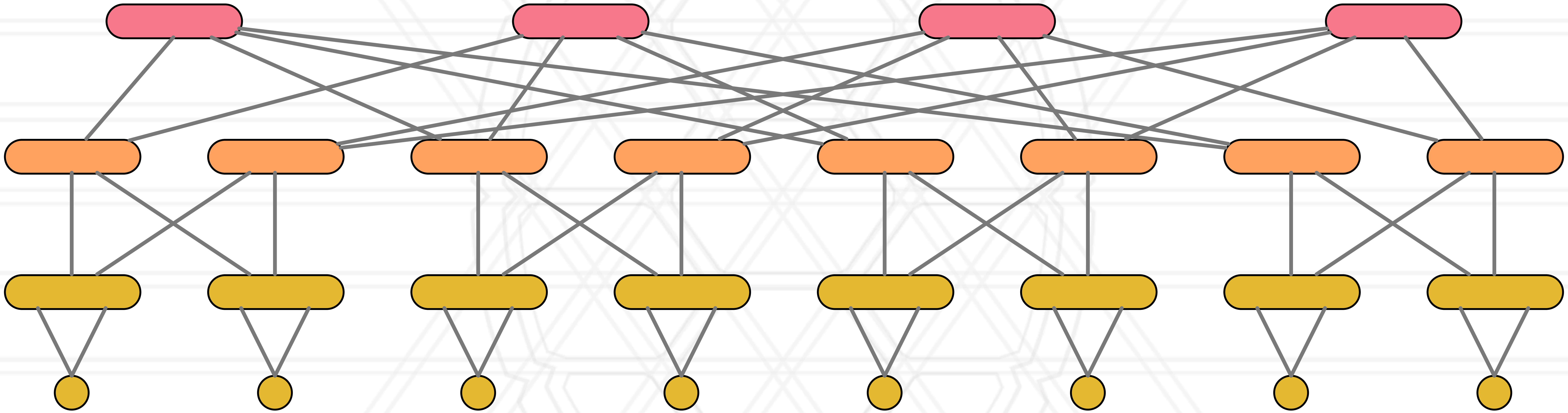
Single-rail single-plane fat-tree

Variations on a full bandwidth fat-tree



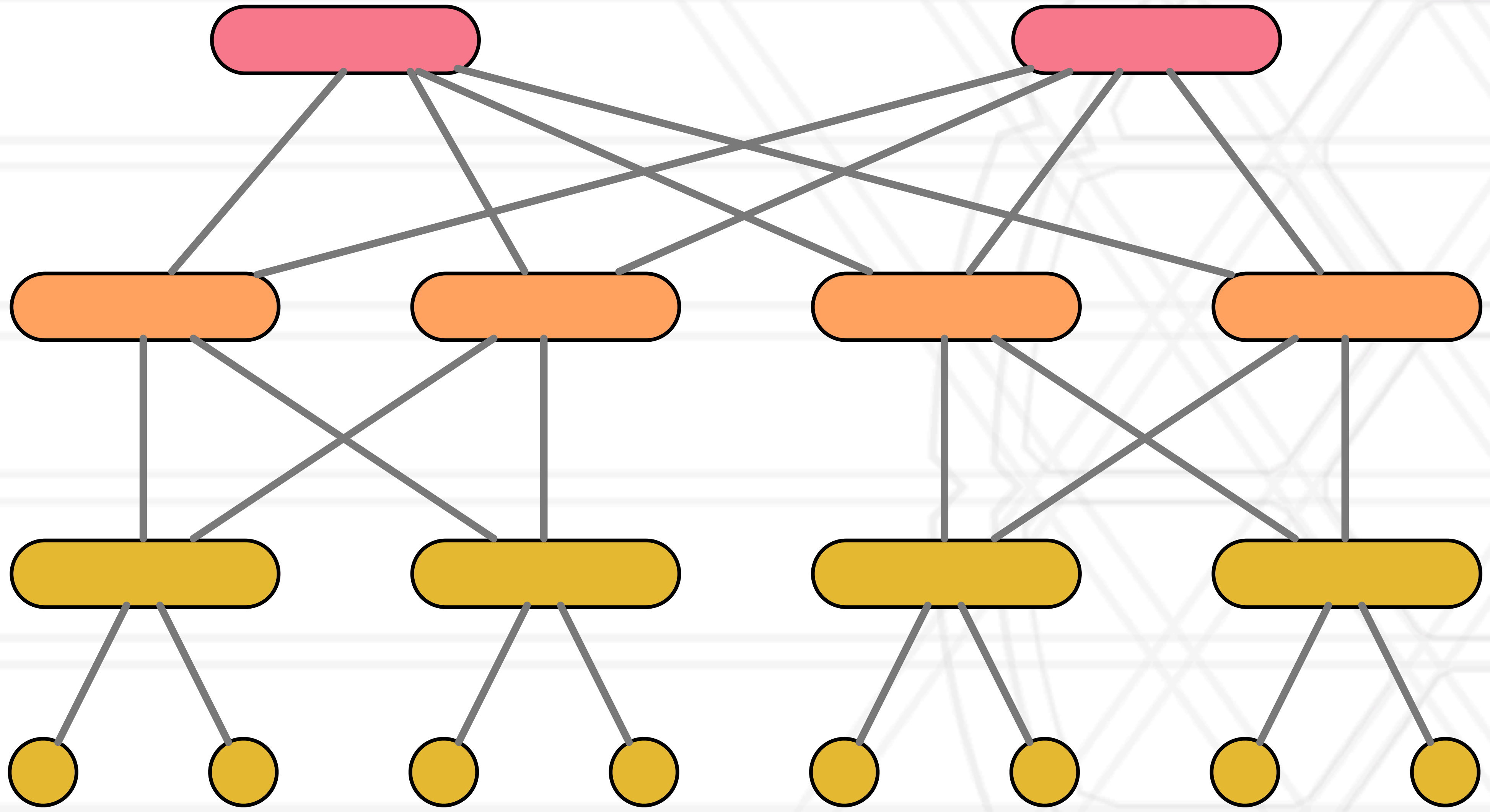
Single-rail single-plane fat-tree (tapered)

Variations on a full bandwidth fat-tree



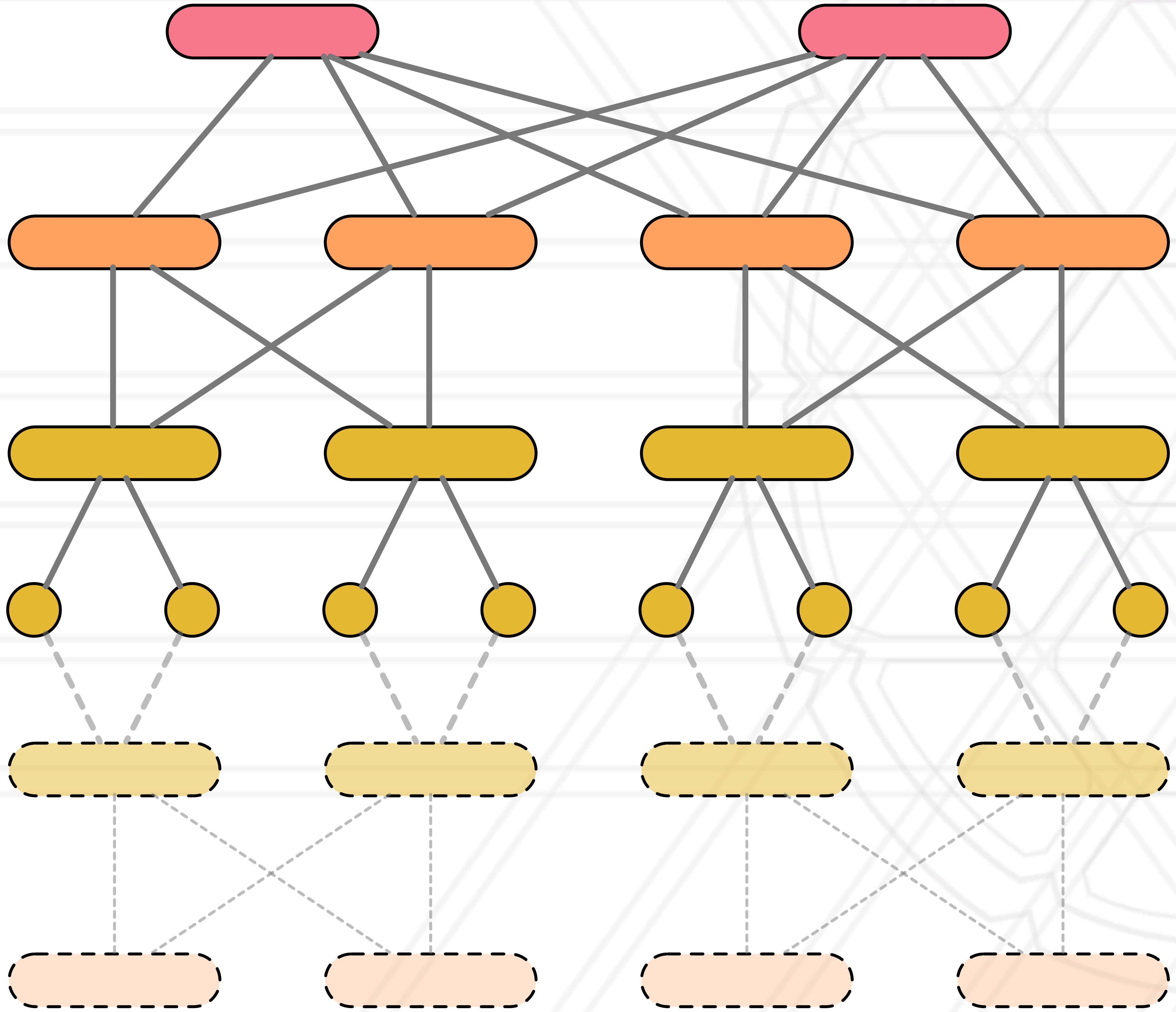
Dual-rail single-plane fat-tree

Variations on a full bandwidth fat-tree



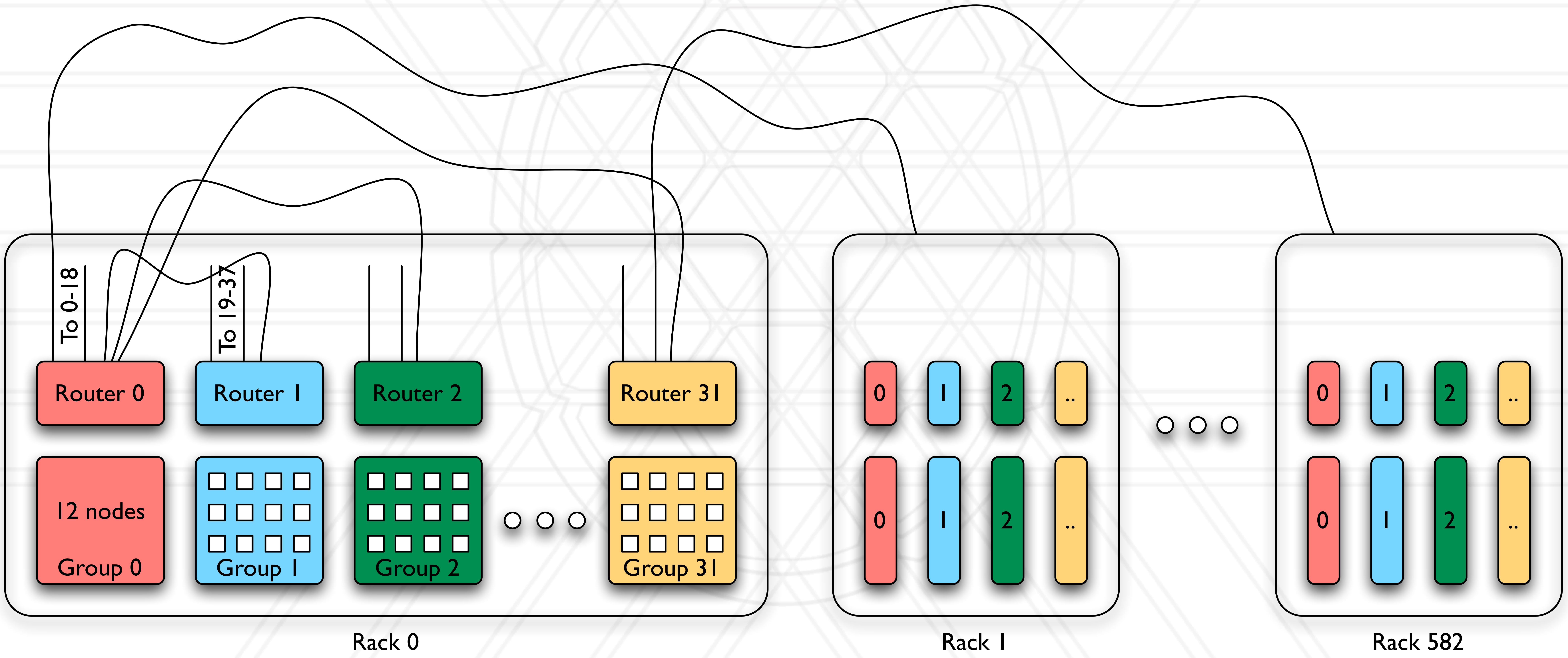
Single-rail single-plane fat-tree

Variations on a full bandwidth fat-tree



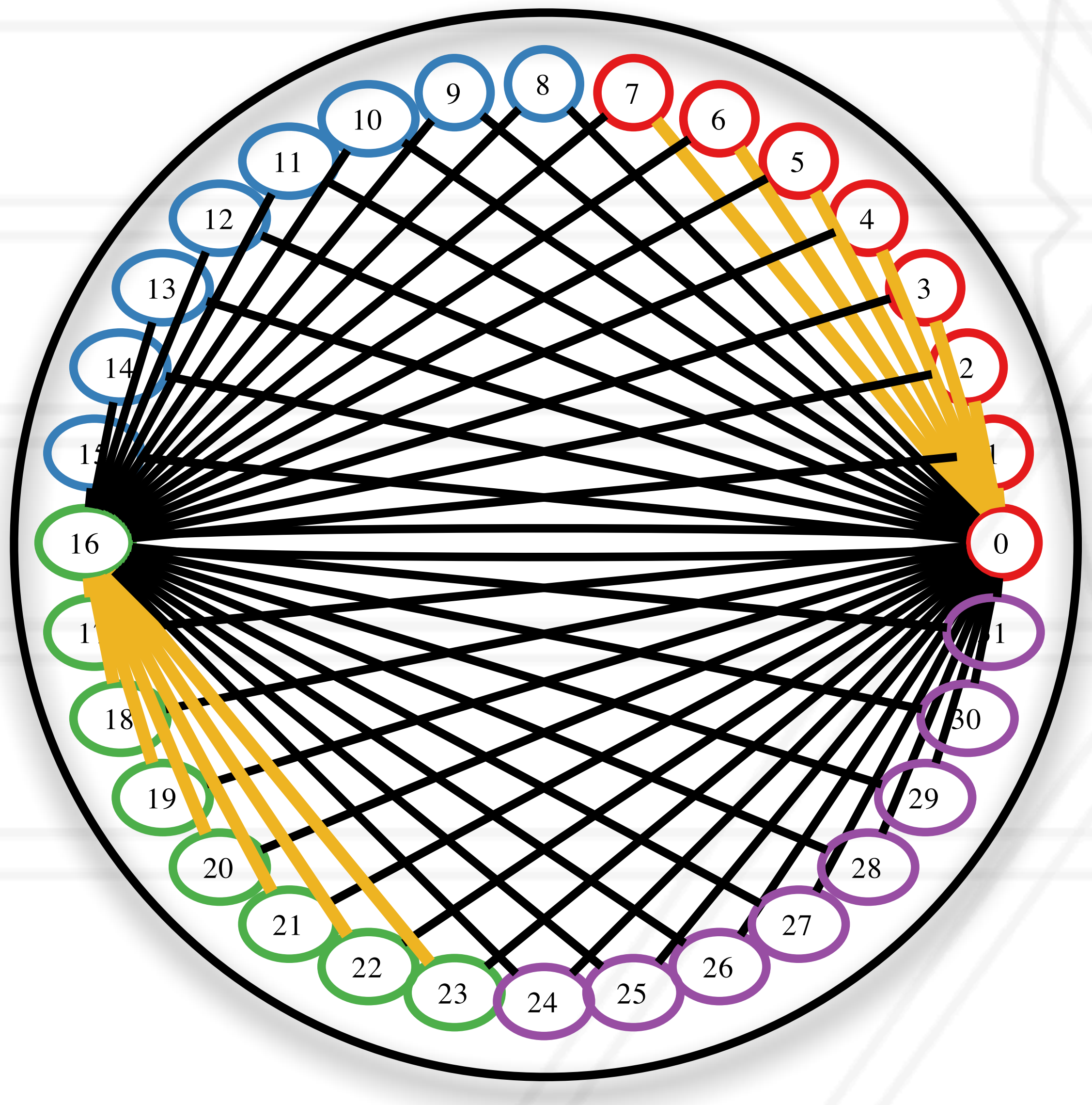
Dual-rail dual-plane fat-tree

Dragonfly network



IBM PERCS network

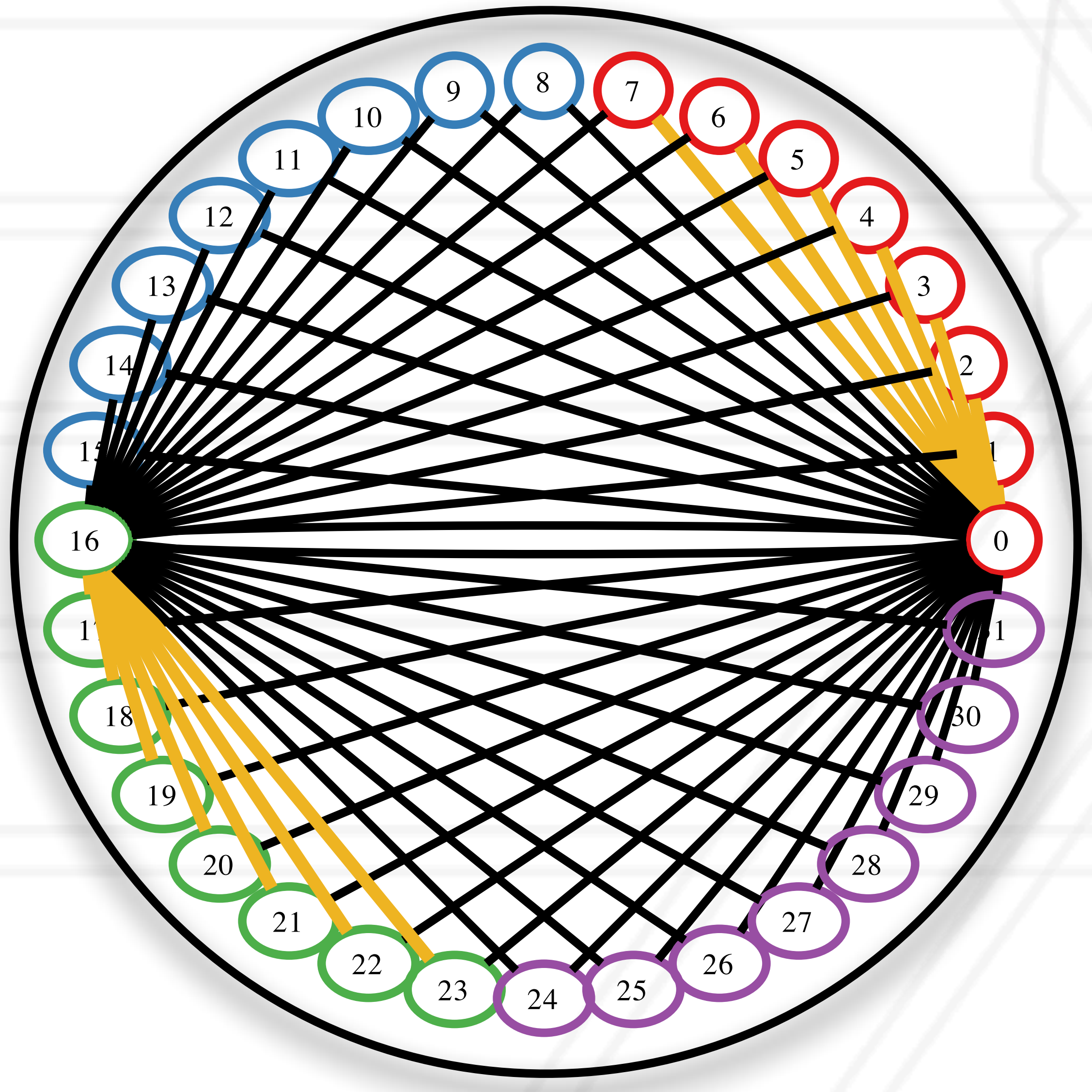
- All-to-all connections within each group



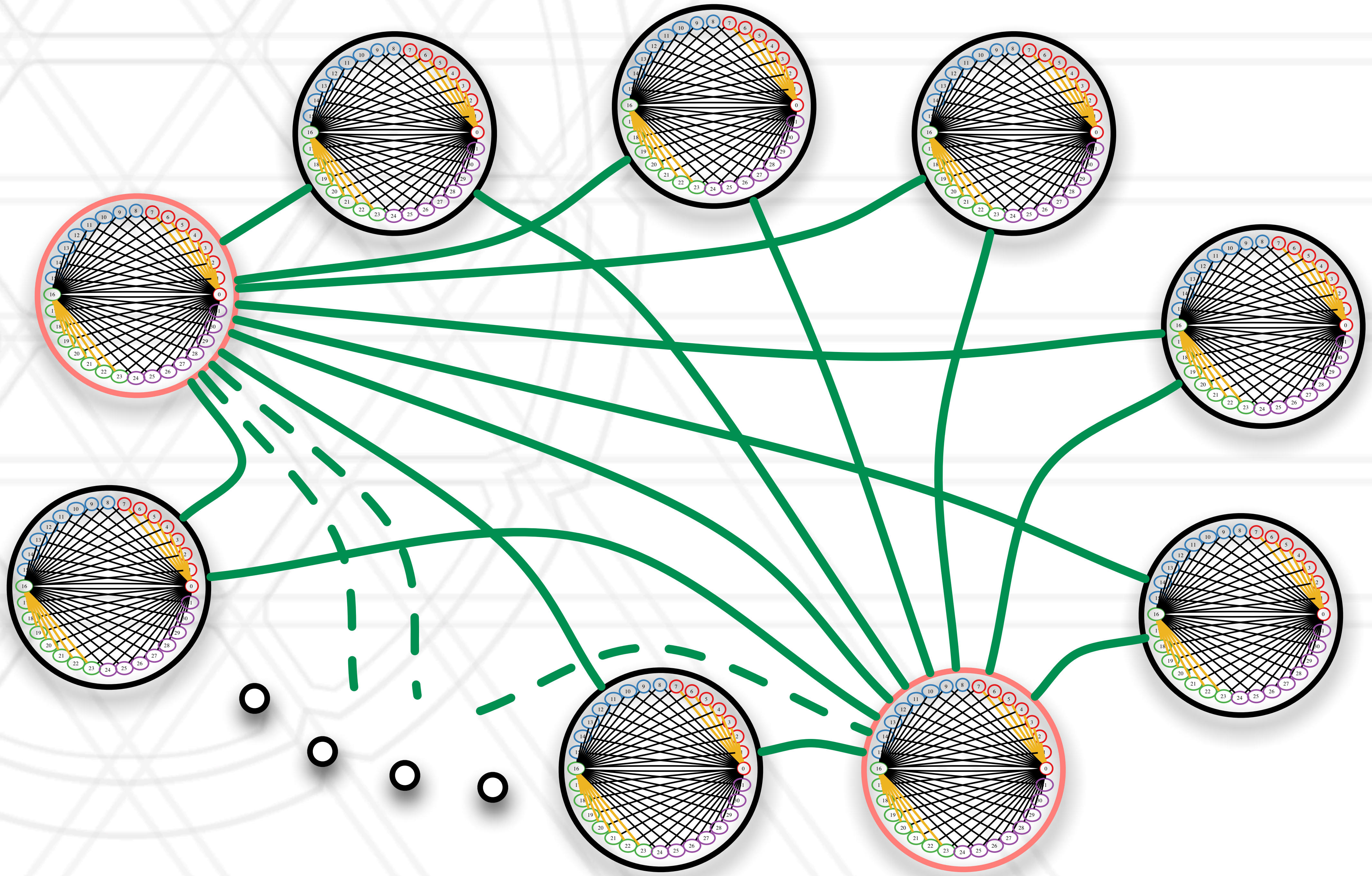
One supernode in the PERCS topology

IBM PERCS network

- All-to-all connections within each group

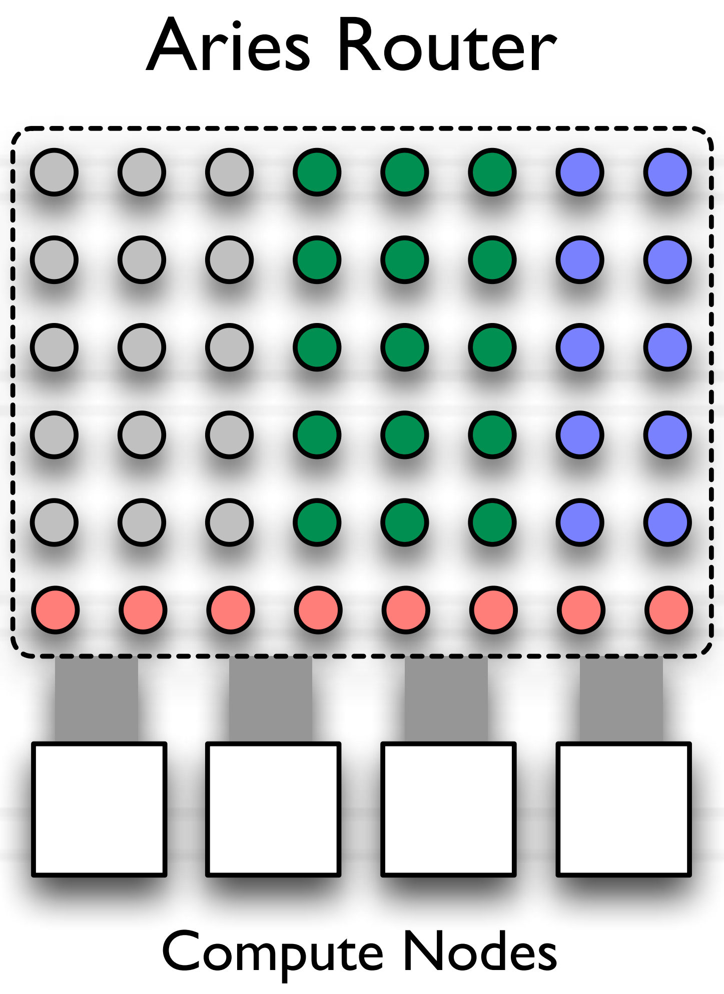


One supernode in the PERCS topology



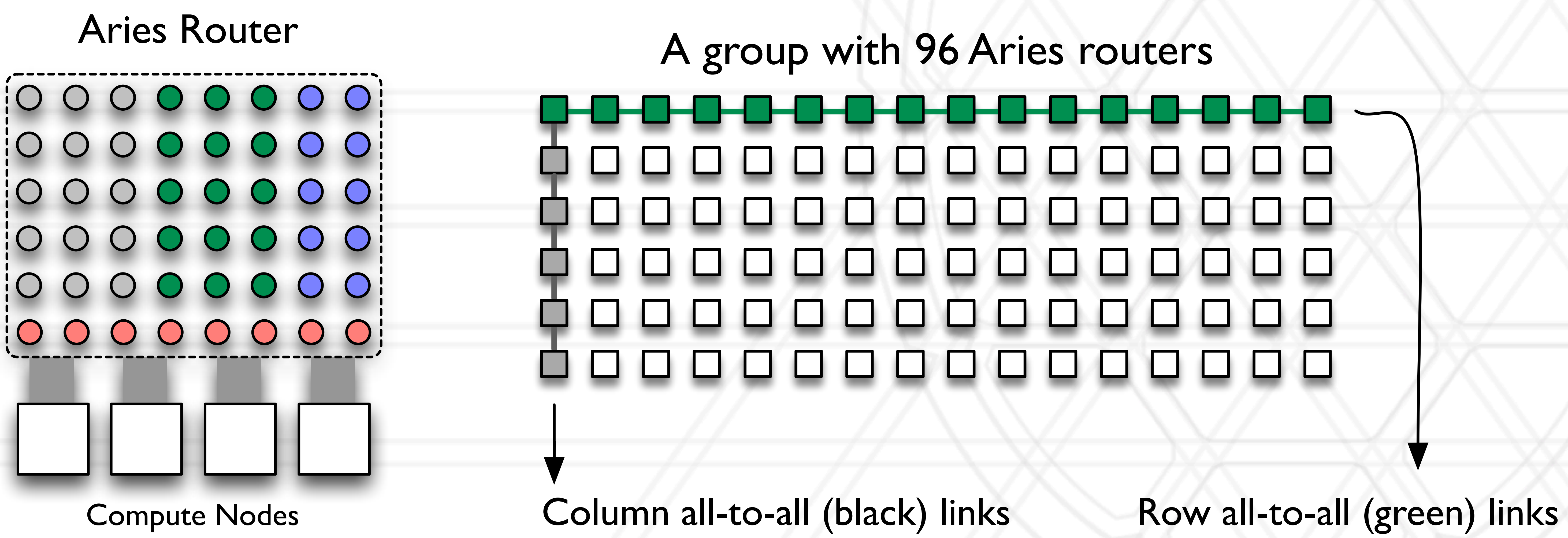
Cray Aries network

- Row and column all-to-all connections within each group



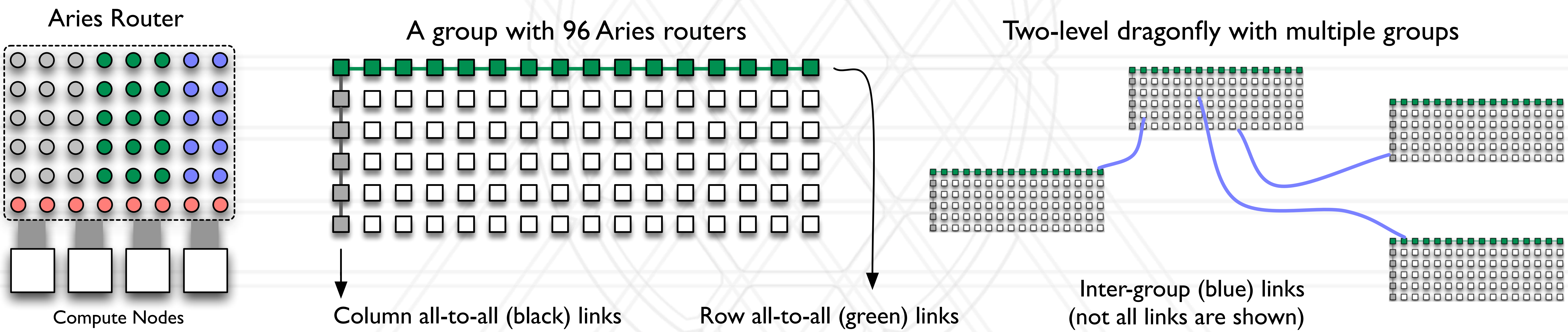
Cray Aries network

- Row and column all-to-all connections within each group



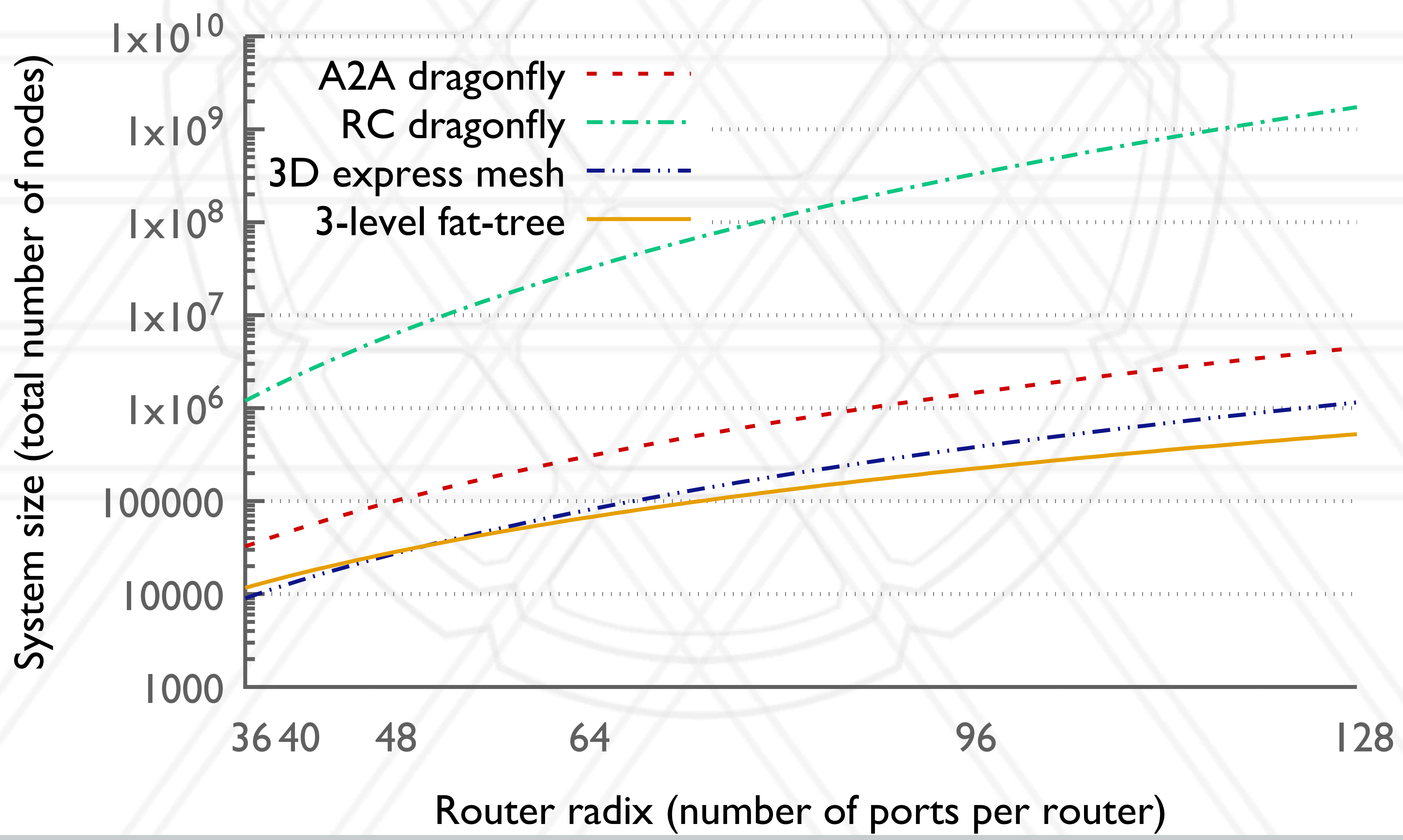
Cray Aries network

- Row and column all-to-all connections within each group



Network comparisons

Network topology	#nodes/router	#links/router	Maximum system size (#nodes)
All-to-all (A2A) dragonfly	k/4	k/2 (**L**), k/4 (**G**)	$(k/2+1)^2 \times (k/4+1) \times k/4$
Row-column (RC) dragonfly	k/6	2k/3 (**L**), k/6 (**G**)	$(k/3+1)^4 \times (k/6+1) \times k/6$
Express mesh (3D, gap=1)	k/4	3k/4	$(k/4+1)^3 \times k/4$
Fat-tree (three-level)	k/2	k/2	$k/2 \times k/2 \times k$



Questions

Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing

- How do you use a partial concentrator graph to construct a good concentrator switch?
- The paper says the capacities of the channels of a universal fat-tree grow exponentially as we go up the tree from the leaves. If so, we must have a large number of wires for the top layers in a big fat tree, which may lead to higher costs in my view. So how can we manage the costs of building a fat-tree network?
- How does fat tree compare with the dragonfly network? Under what kind of circumstance, we prefer one to another?

Questions

Technology-Driven, Highly-Scalable Dragonfly Topology

- It's said in figure 6(b), the effective radix is 32, which I understand as a=8, p=2, h=2 and k'=a(p+h)=32. But it says the radix of each router k=7, which I don't get it. According to the formula, k should be a+p+h-1=11. So why does it say k=7 here?
- In the part introducing the credit round-trip latency technique, it says "the credit is delayed by td(O) − min [td(o)]". Where does the little o come from?
- Is there any hardware technology that supports advanced congestion look ahead nowadays?

Questions?

UNIVERSITY OF MARYLAND

Abhinav Bhatele

5218 Brendan Iribe Center (IRB) / College Park, MD 20742

phone: 301.405.4507 / e-mail: bhatele@cs.umd.edu