
Parallel Computing Platforms

Network Topologies

John Mellor-Crummey

**Department of Computer Science
Rice University**

`johnmc@rice.edu`

Topics for Today

Interconnection networks

- Taxonomy
- Metrics
- Topologies
- Characteristics
 - cost
 - performance

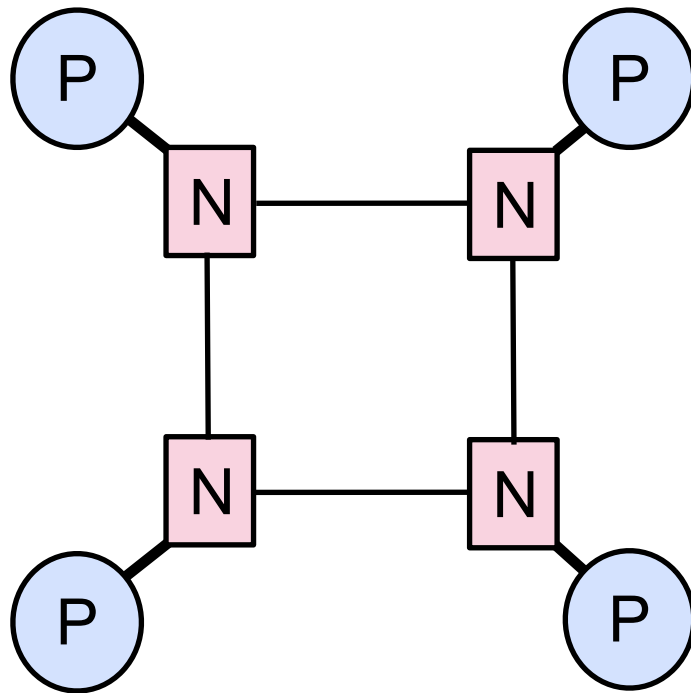
Interconnection Networks

Carry data between processors and to memory

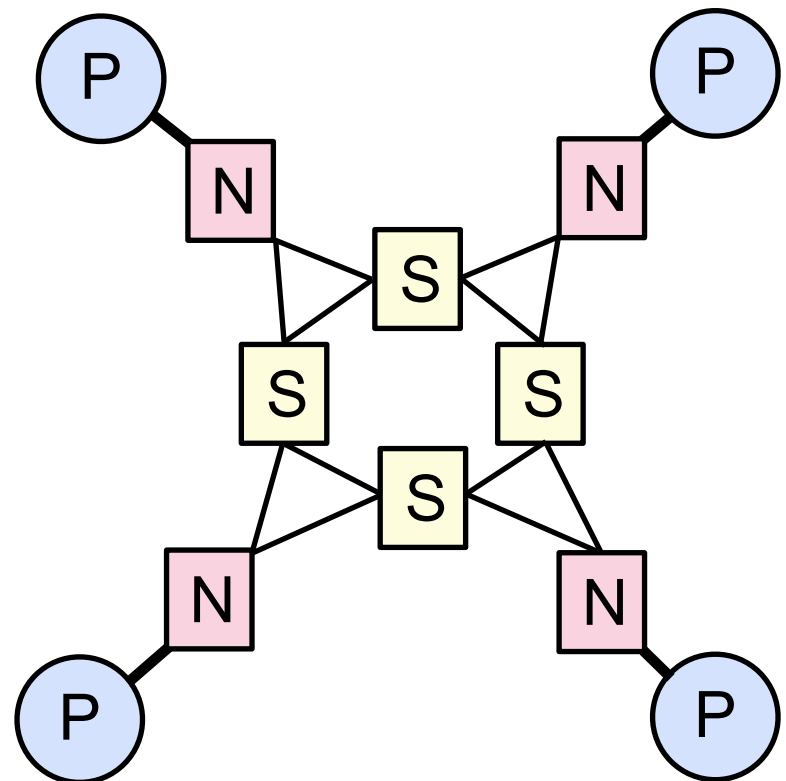
- **Interconnect components**
 - switches
 - links (wires, fiber)
- **Interconnection network flavors**
 - static networks: point-to-point communication links
 - *AKA direct networks*
 - dynamic networks: switches and communication links
 - *AKA indirect networks*

Static vs. Dynamic Networks

static/direct network



dynamic/indirect network



Sometimes, the processor and network interface are on the same chip, e.g., Blue Gene

Dynamic Network Switch

- Maps a fixed number of inputs to outputs
- Number of ports on a switch = *degree* of the switch
- Switch cost
 - grows as the square of switch degree
 - packaging cost grows linearly with the number of pins
- Key property: blocking vs. non-blocking
 - blocking
 - path from p to q may conflict with path from r to s for independent p, q, r, s
 - non-blocking
 - disjoint paths between each pair of independent sources and sinks

Network Interface

Processor node's link to the interconnect

- **Network interface responsibilities**
 - packetizing communication data
 - computing routing information
 - buffering incoming/outgoing data
- **Network interface connection**
 - I/O bus: Peripheral Component Interface Express (PCIe)
 - memory bus: e.g. AMD HyperTransport, Intel QuickPath
 - higher bandwidth and tighter coupling than I/O bus
- **Network performance**
 - depends on relative speeds of I/O and memory buses

Network Topologies

- Many network topologies
- Tradeoff: performance vs. cost
- Machines often implement hybrids of multiple topologies
 - why?
 - packaging
 - cost
 - available components

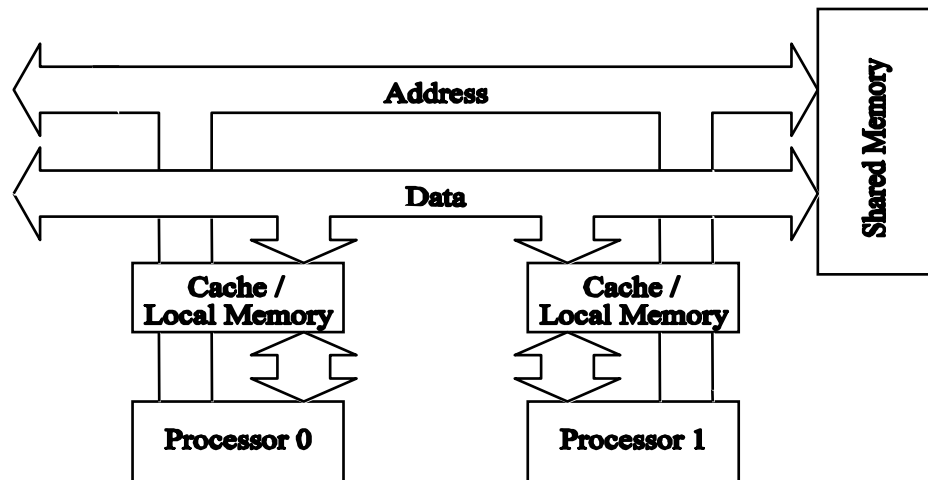
Metrics for Interconnection Networks

- **Degree**
number of links per node
- **Diameter**
longest distance between two nodes in the network
- **Bisection Width**
min # of wire cuts to divide the network in 2 halves
- **Cost:**
~ # links and switches

Network Topologies: Bus

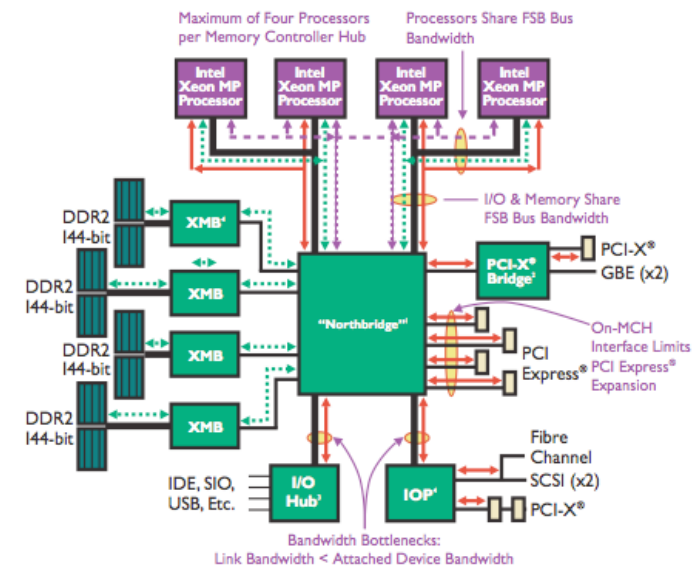
- All processors access a common bus for exchanging data
- Used in simplest and earliest parallel machines
- Advantages
 - distance between any two nodes is $O(1)$
 - provides a convenient broadcast media
- Disadvantages
 - bus bandwidth is a performance bottleneck

Bus



Bus-based interconnect
with local memory/cache

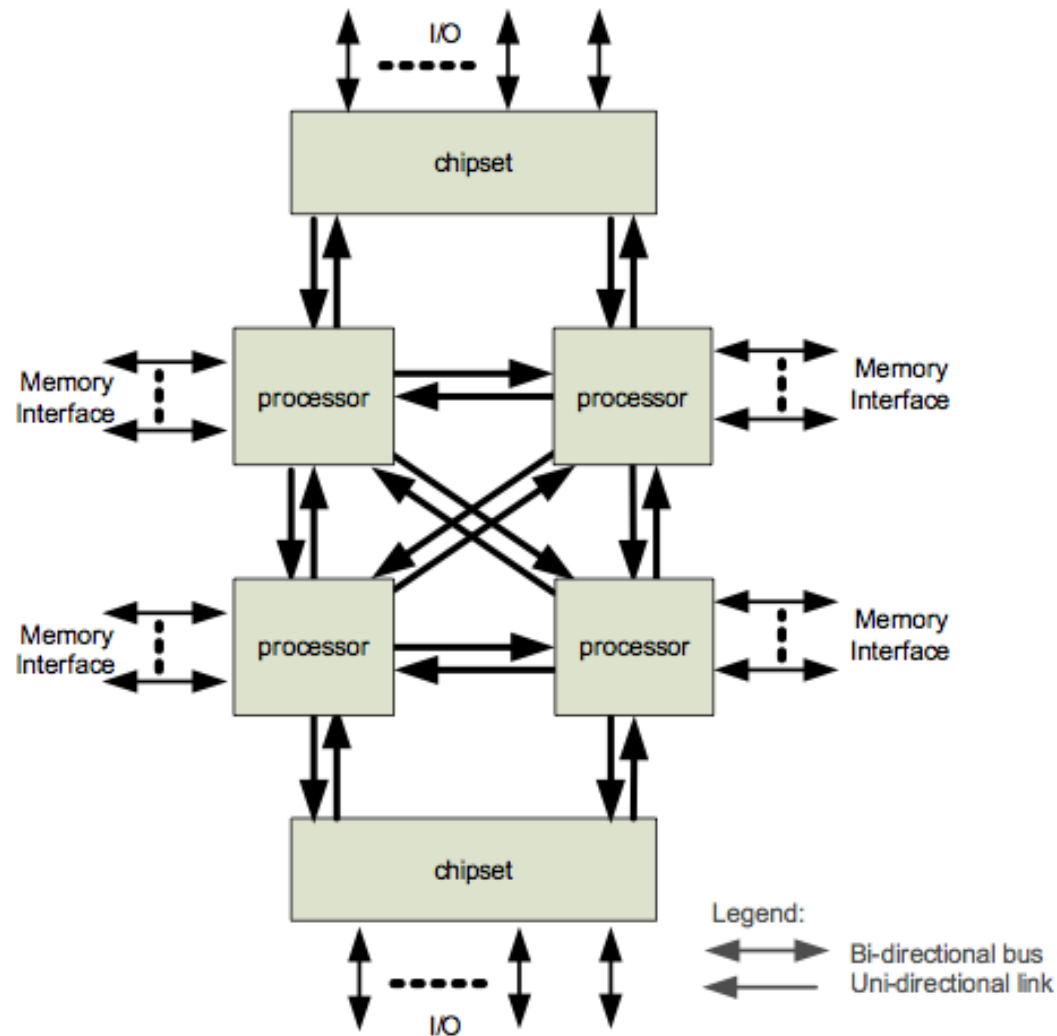
Intel Xeon MP Processor-based 4P Server



Dual-bus (circa 2005)

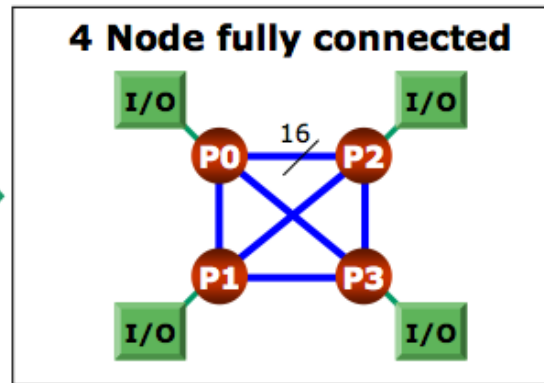
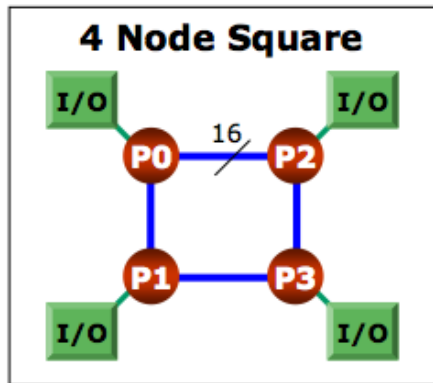
Since much of the data accessed by processors is local to the processor, cache is critical for the performance of bus-based machines

Bus Replacement: Direct Connect



Intel Quickpath interconnect
(2009 - present)

Direct Connect: 4 Node Configurations



4N SQ (2GT/s HyperTransport)

Diam 2 Avg Diam 1.00

XFIRE BW 14.9GB/s

+ 2 EXTRA LINKS

4N FC (2GT/s HyperTransport)

Diam 1 Avg Diam 0.75

XFIRE BW 29.9GB/s

W/ HYPERTRANSPORT3

4N FC (4.4GT/s HyperTransport3)

Diam 1 Avg Diam 0.75

XFIRE BW 65.8GB/s

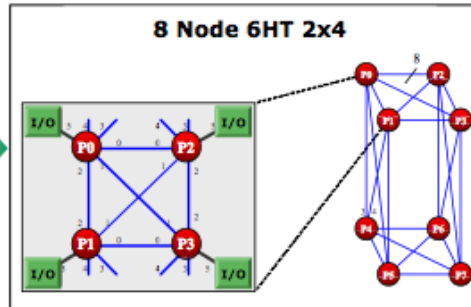
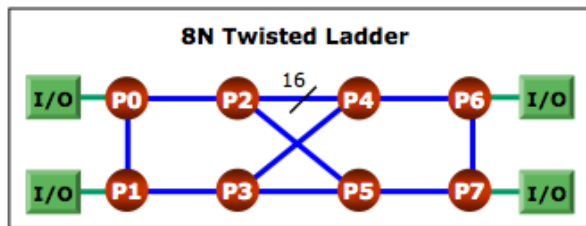
2x

4x

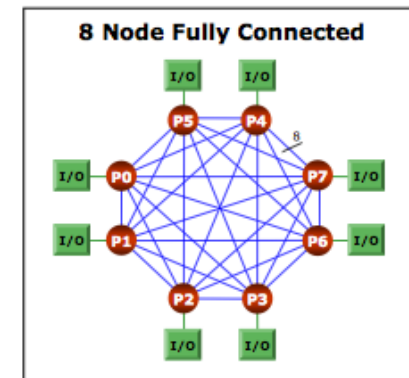
“crossfire” bandwidth is
the link-limited all-to-all
bandwidth (data only)

Figure Credit : The Opteron CMP NorthBridge
Architecture, Now and in the Future, AMD , Pat
Conway, Bill Hughes , HOT CHIPS 2006

Direct Connect: 8 Node Configurations



OR



8N TL (2GT/s HyperTransport)

Diam 3 Avg Diam 1.62

XFIRE BW 15.2GB/s

8N 2x4 (4.4GT/s HyperTransport3)

Diam 2 Avg Diam 1.12

XFIRE BW 72.2GB/s

(5X)

8N FC (4.4GT/s HyperTransport3)

Diam 1 Avg Diam 0.88

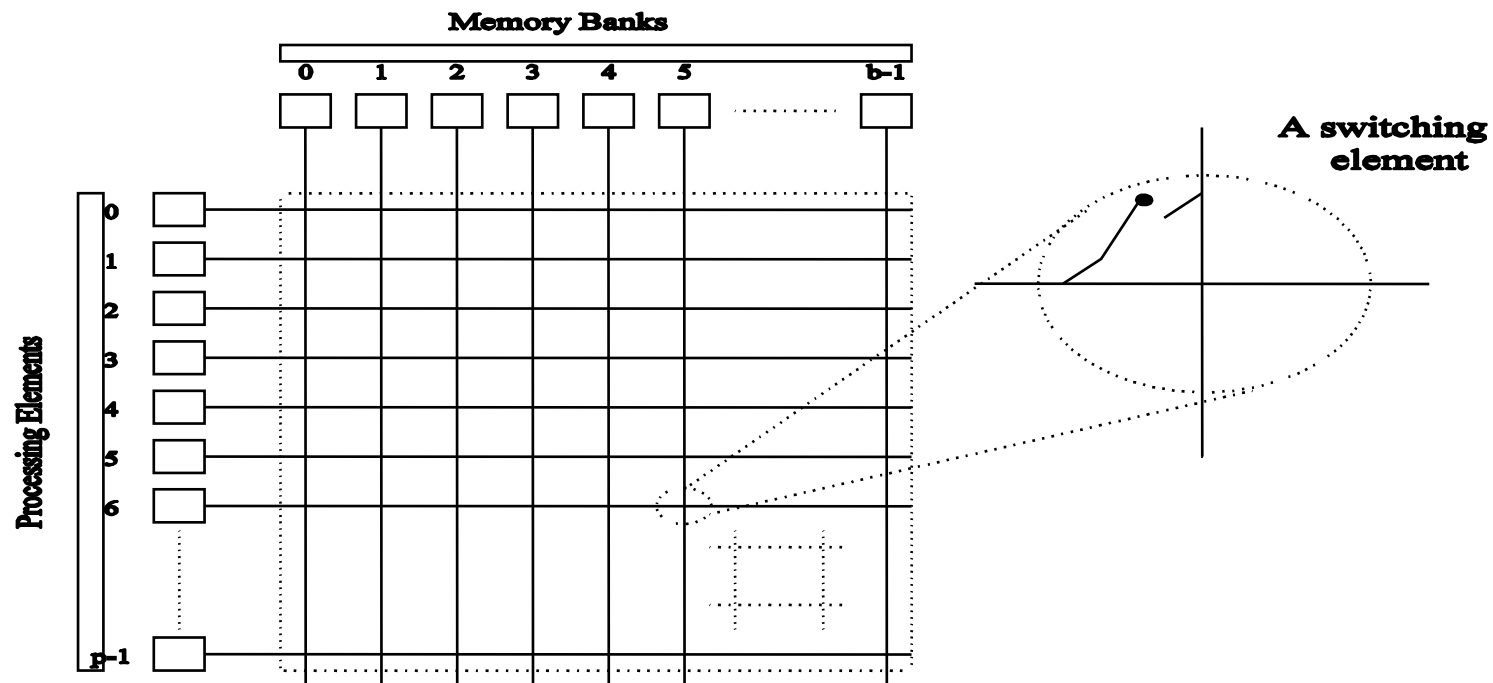
XFIRE BW 94.4GB/s

(6X)

Figure Credit : The Opteron CMP NorthBridge Architecture, Now and in the Future, AMD , Pat Conway, Bill Hughes , HOT CHIPS 2006

Crossbar Network

A crossbar network uses an $p \times m$ grid of switches to connect p inputs to m outputs in a non-blocking manner



A non-blocking crossbar network connecting p processors to b memory banks

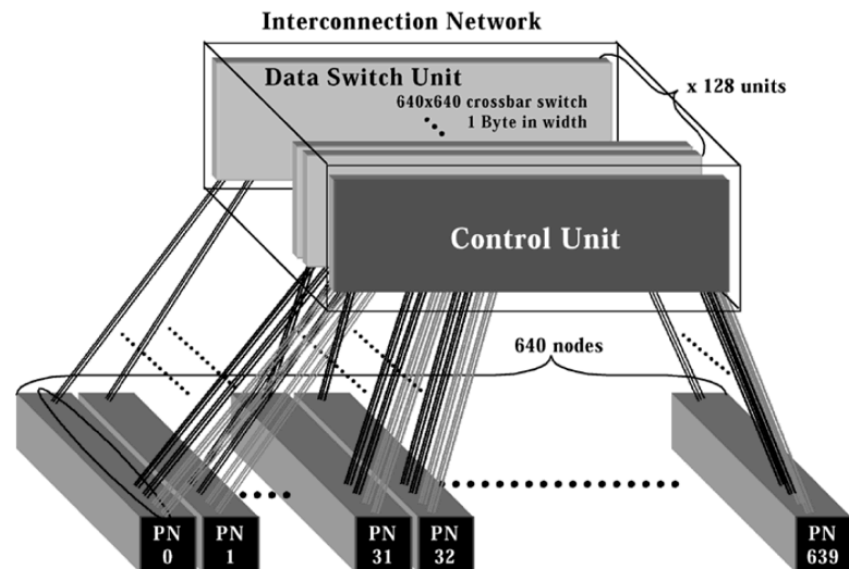
Crossbar Network

- Cost of a crossbar: $O(p^2)$
- Generally difficult to scale for large values of p
- Examples

- full crossbar

- Earth Simulator:
custom 640-way single-stage
crossbar

S. Habata et al. Earth Simulator System
NEC Res. & Develop., 44(1), Jan 2003.



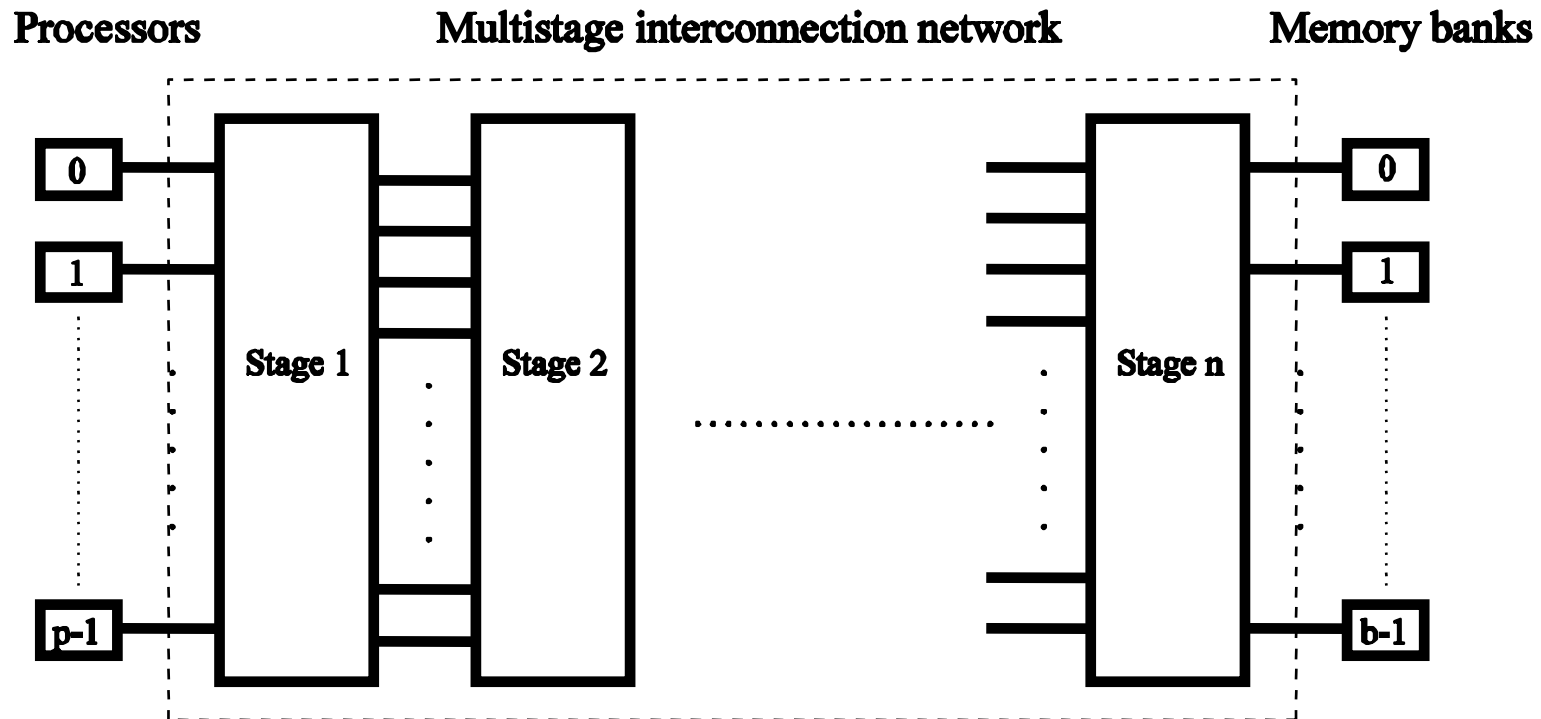
- crossbar as building block

- Rice Terascale Cluster (retired in 2008): Myrinet 2000 interconnect
16-way crossbar switches in 128-way Clos network

Assessing Network Alternatives

- **Buses**
 - excellent cost scalability
 - poor performance scalability
- **Crossbars**
 - excellent performance scalability
 - poor cost scalability
- **Multistage interconnects**
 - compromise between these extremes

Multistage Network



Schematic of processor-to-memory
multistage interconnection network

(e.g., BBN Monarch)

Multistage Omega Network

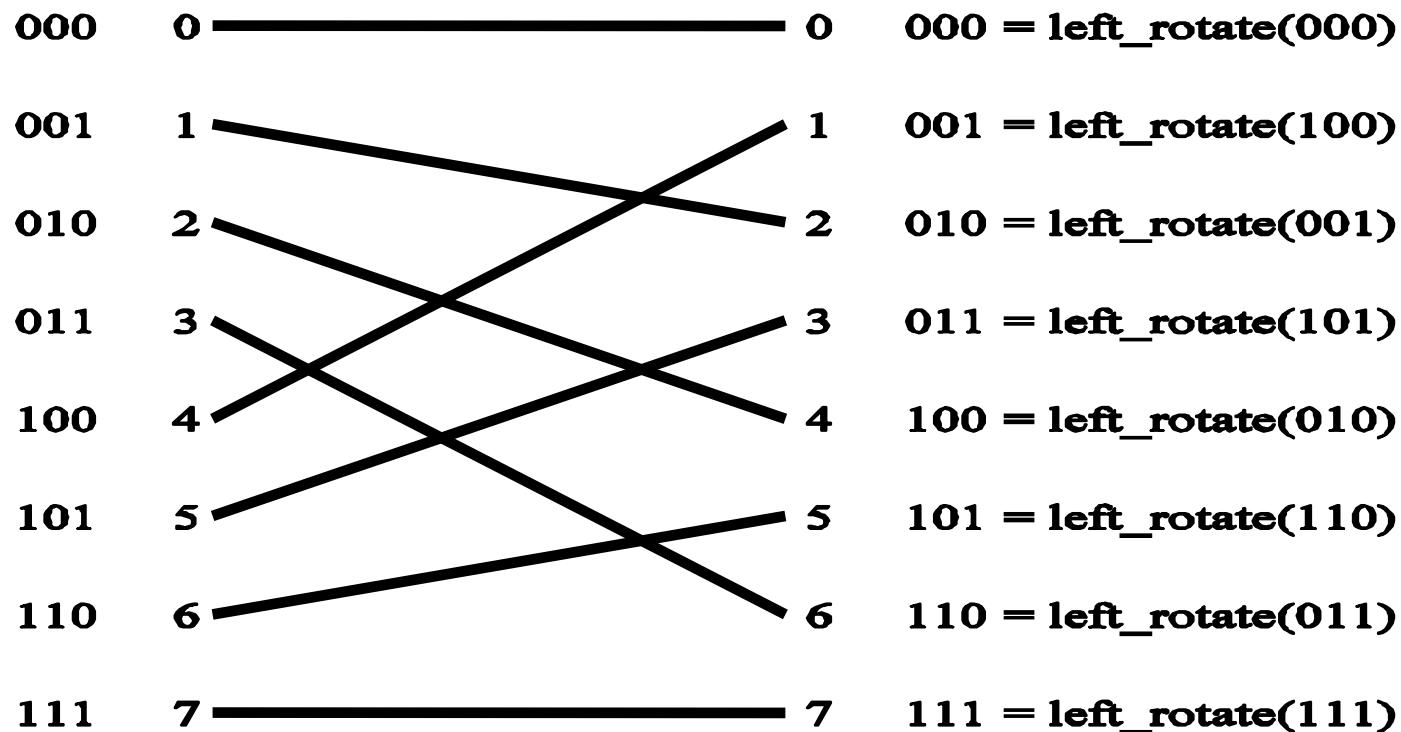
- **Organization**
 - $\log p$ stages
 - p inputs/outputs
- **At each stage, input i is connected to output j if:**

$$j = \begin{cases} 2i, & 0 \leq i \leq p/2 - 1 \\ 2i + 1 - p, & p/2 \leq i \leq p - 1 \end{cases}$$

if $p = 2^k$ then $j = \text{left_rotate}(i)$

Omega Network Stage

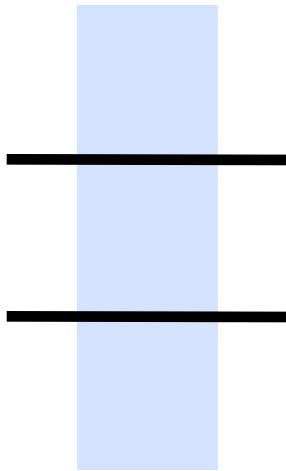
Each Omega stage is connected in a perfect shuffle



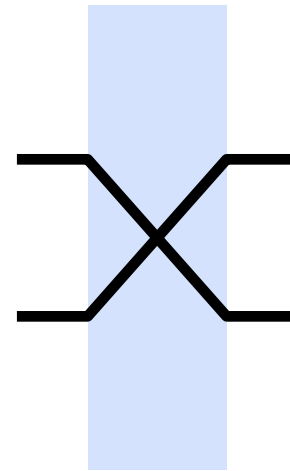
A perfect shuffle interconnection for eight inputs and outputs

Omega Network Switches

- **2×2 switches connect perfect shuffles**
- **Each switch operates in two modes**

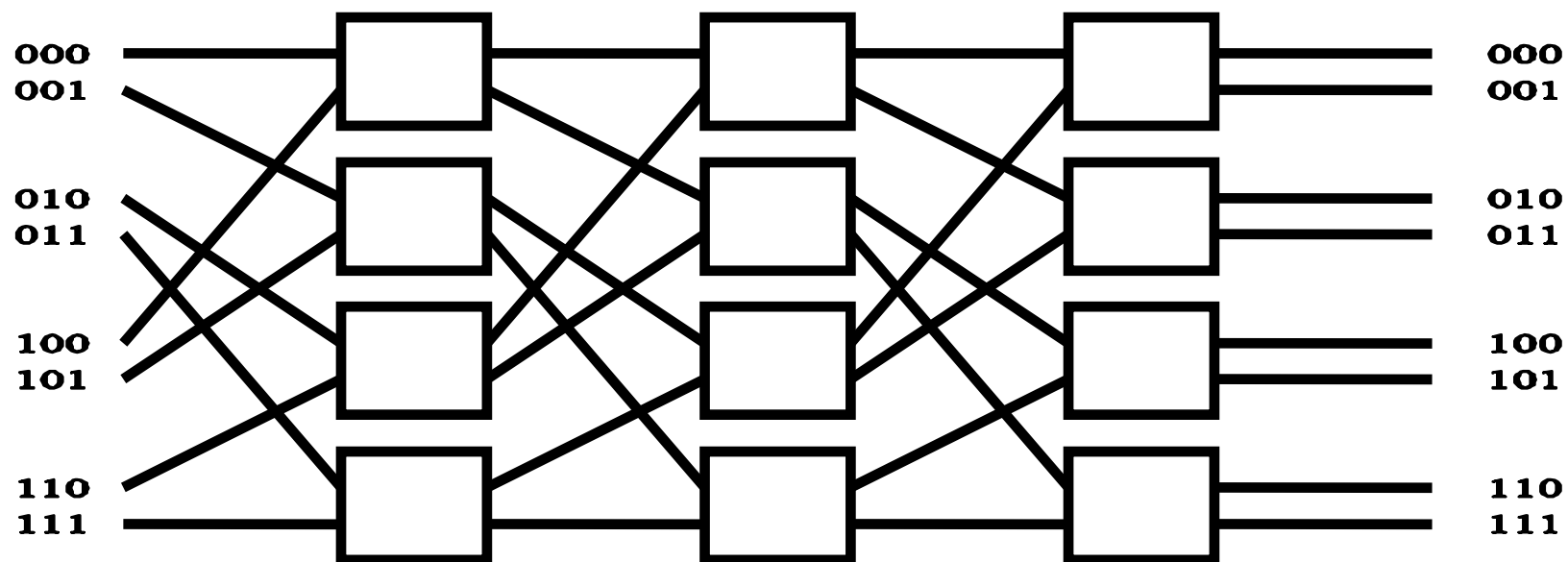


Pass-through



Cross-over

Multistage Omega Network



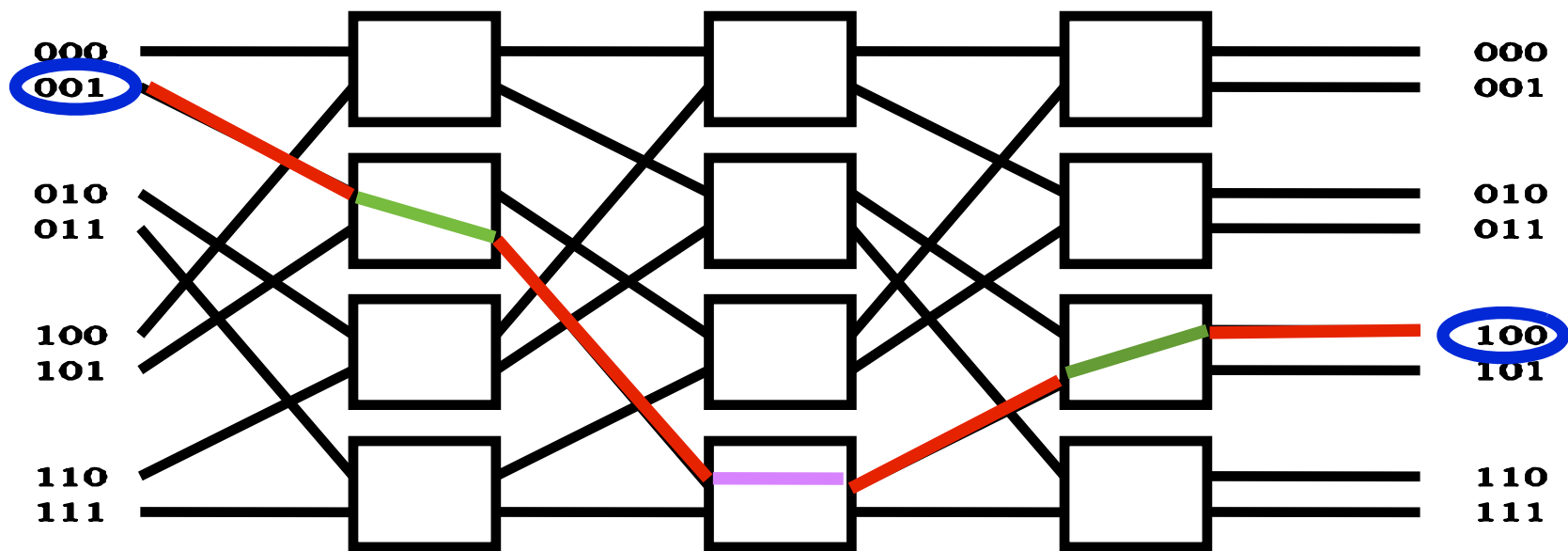
Omega network connecting eight inputs and eight outputs

Cost: $p/2 \times \log p$ switching nodes $\rightarrow O(p \log p)$

Omega Network Routing

- **Let**
 - s = binary representation of the source processor
 - d = binary representation of the destination processor or memory
- **The data traverses the link to the first switching node**
 - if the most significant bit of s and d are the same
route data in pass-through mode by the switch
 - else
use crossover path
- **Strip off leftmost bit of s and d**
- **Repeat for each of the $\log p$ switching stages**

Omega Network Routing



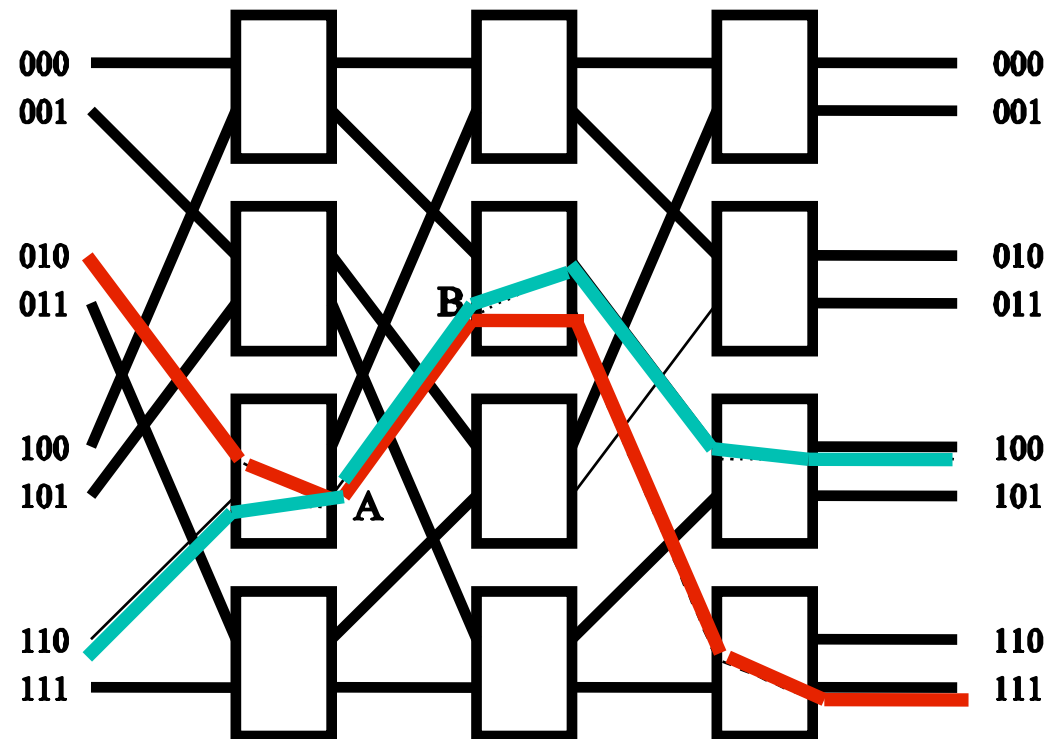
Example: $s = 001 \rightarrow d = 100$

stage 1: leftmost bit $s \neq d \rightarrow$ **crossover**

stage 2: middle bit $s = d \rightarrow$ **pass-through**

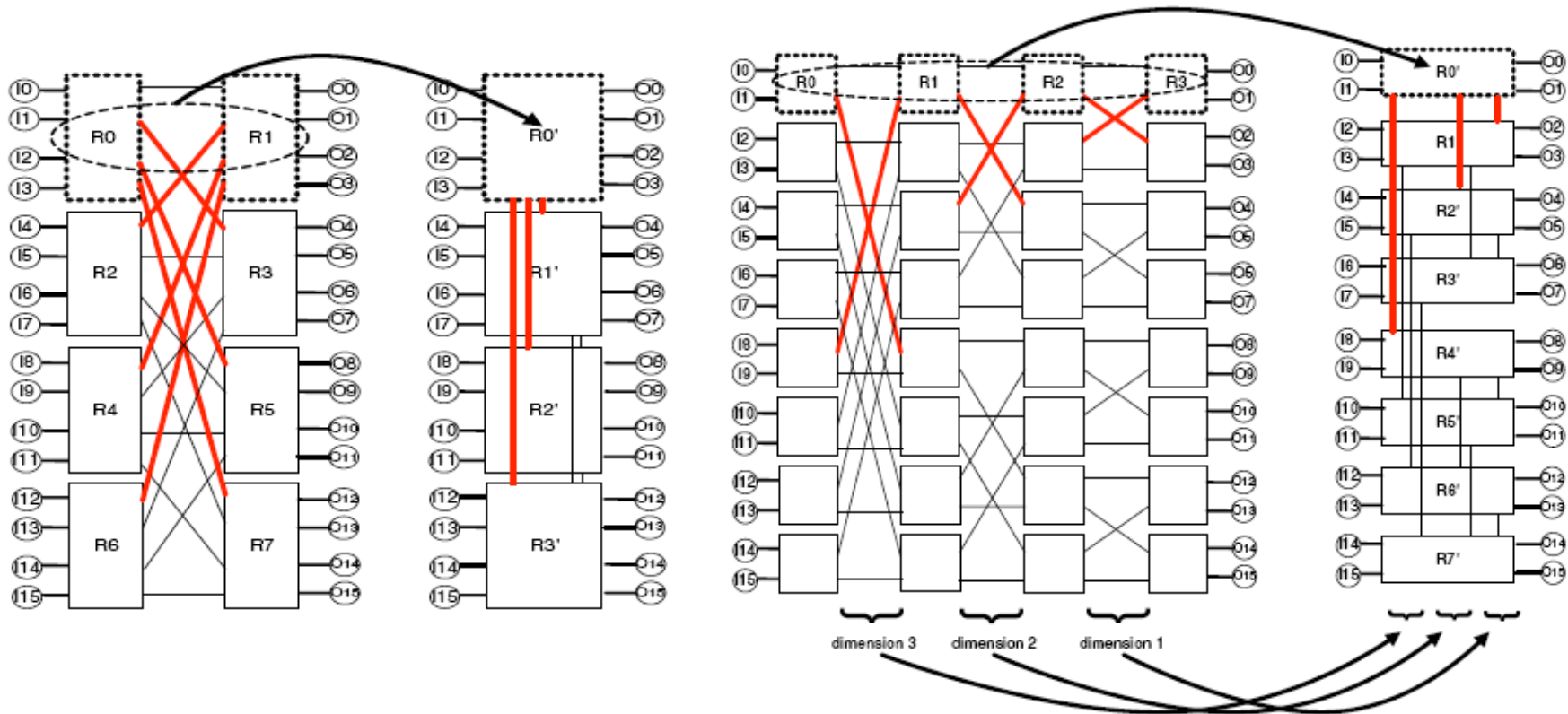
stage 3: rightmost bit $s \neq d \rightarrow$ **crossover**

Blocking in an Omega Network



One of the messages (010 to 111 or 110 to 100) blocks at link AB

Butterfly and Flattened Butterfly



4-ary, 2 fly → 4-ary, 2-flat

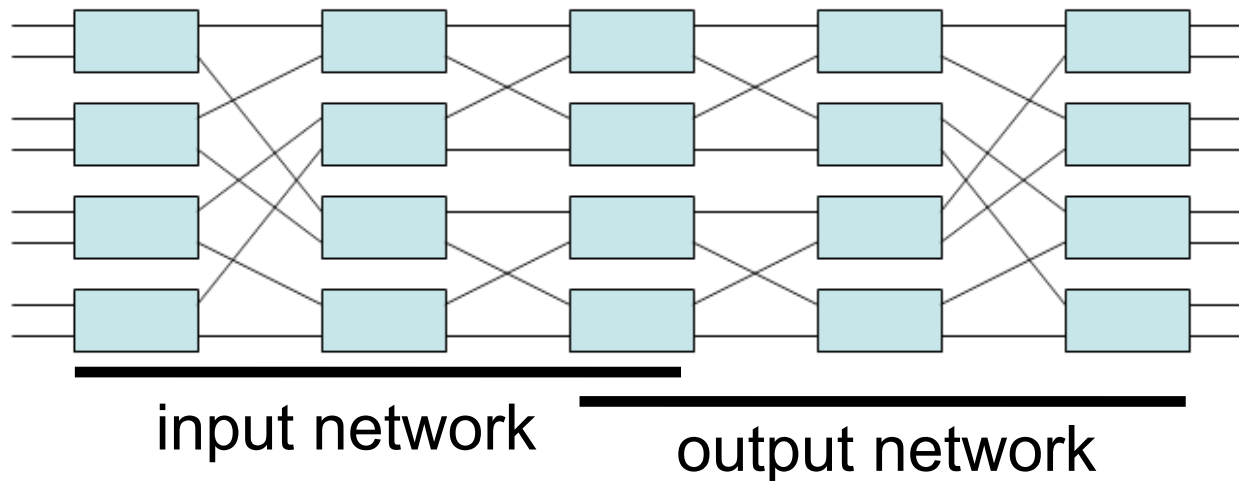
2-ary, 4 fly → 2-ary, 4-flat

- Start with conventional butterfly k-ary n-fly
- Flatten routers in each row of the network into single router
- Flattened butterfly has better performance and path diversity

John Kim, William J. Dally, Dennis Abts: Flattened butterfly: a cost-efficient topology for high-radix networks. *ISCA 2007*: 126-137

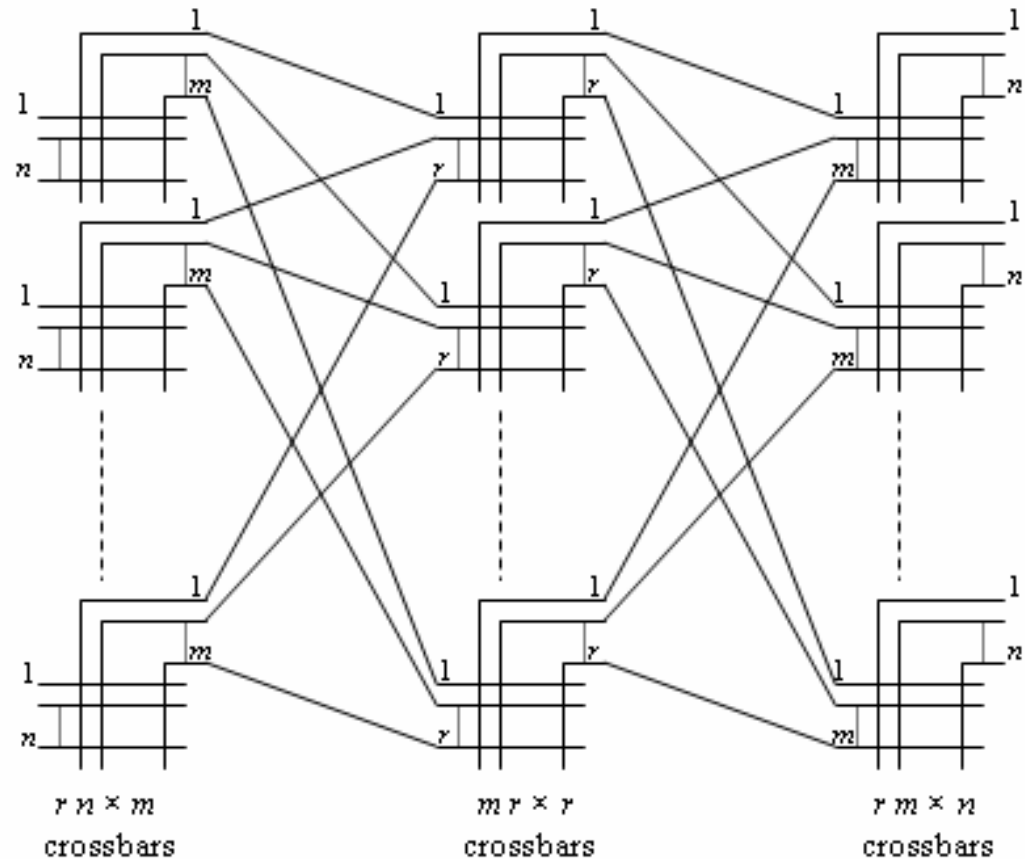
Clos Network

- Multistage non-blocking network with odd number of stages
 - uses fewer switches than a complete crossbar
- Equivalent to two back-to-back butterfly networks
 - last stage of input network fused w/ first stage of output network
- Input network
 - routes from any input to any middle stage switch
- Output network
 - routes from any middle stage switch to any output



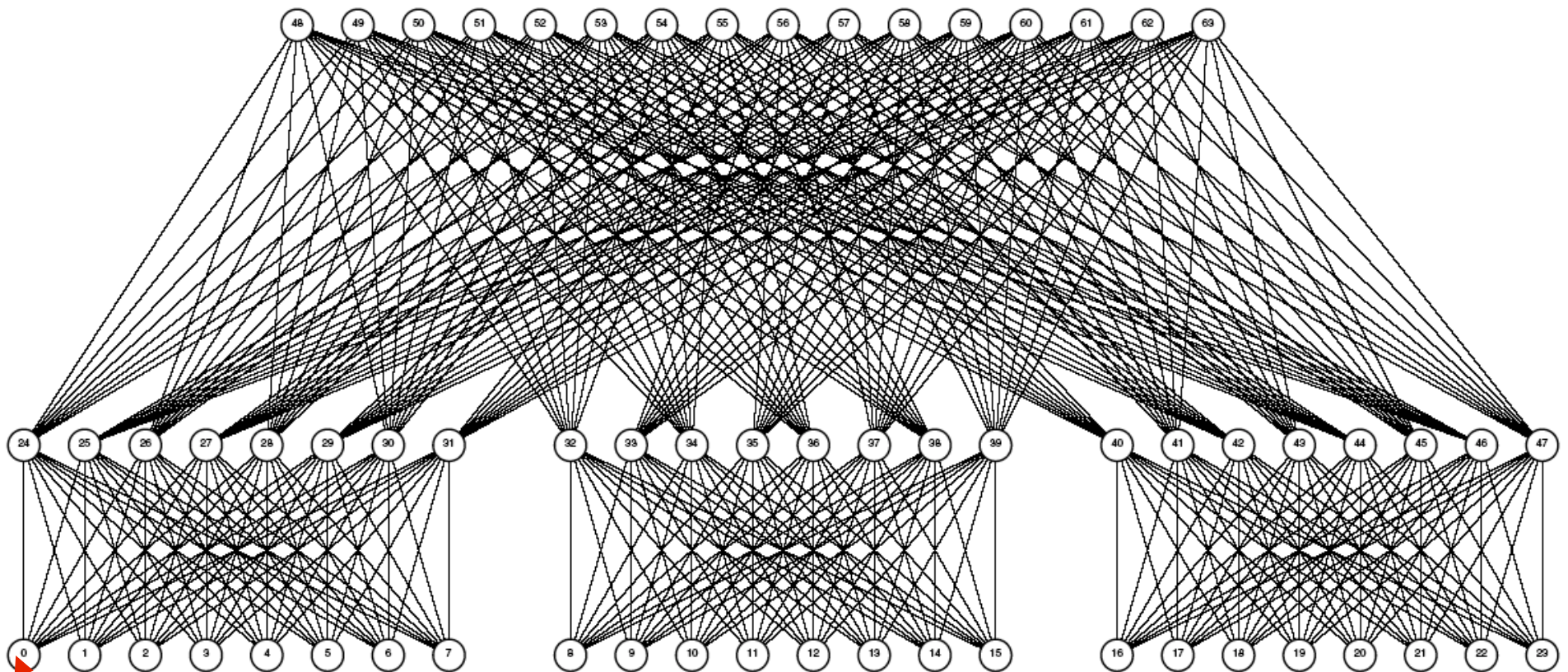
Clos Network

- Multistage non-blocking network with odd number of stages
 - uses fewer switches than a complete crossbar
- Input network
 - routes from any input to any middle stage switch
- Output network
 - routes from any middle stage switch to any output



Folded Clos Network

192 hosts, 64 16-way crossbar switches



8 hosts attach to each 16-port switch node

Charles Clos, "A study of Non-blocking Switching Networks,"
Bell System Technical Journal, 1953, 32(2):406-424.

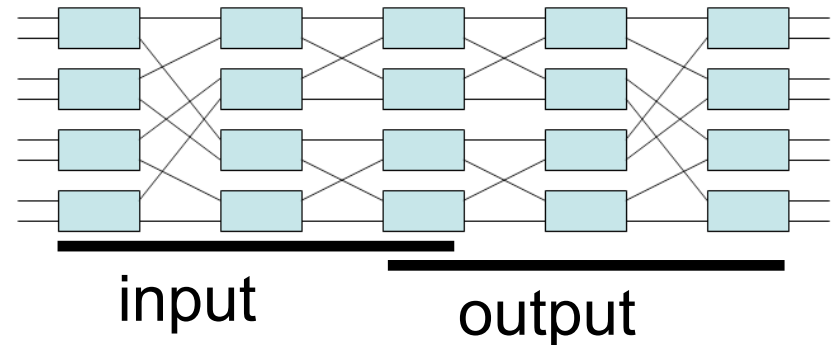
Clos Network

- **Advantages**

- provides many paths between each pair of nodes
- path diversity enables Clos to route arbitrary traffic patterns without a loss of throughput

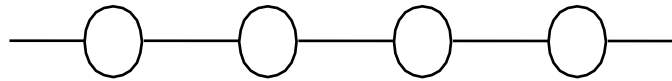
- **Disadvantages**

- cost that is nearly double of a butterfly with equal capacity
- greater latency than a butterfly
- why higher cost and latency?
 - need to route packets to arbitrary middle stage & then destination
 - double number of long cables = double cost
 - doubles number of inter-router channels traversed: doubles latency

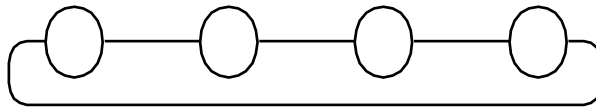


Linear Array

- Each node has two neighbors: left & right

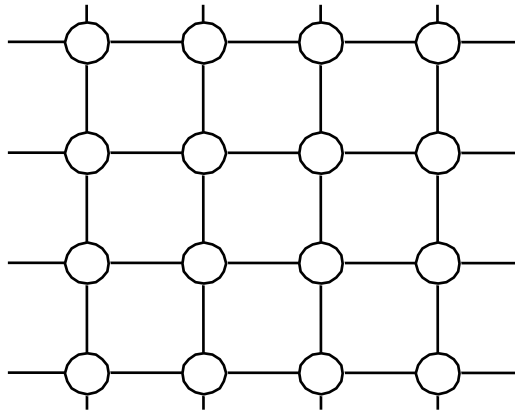


- If connection between nodes at ends: 1D torus (ring)

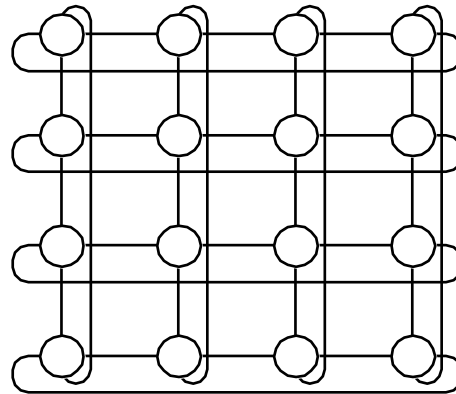


Meshes and k -dimensional Meshes

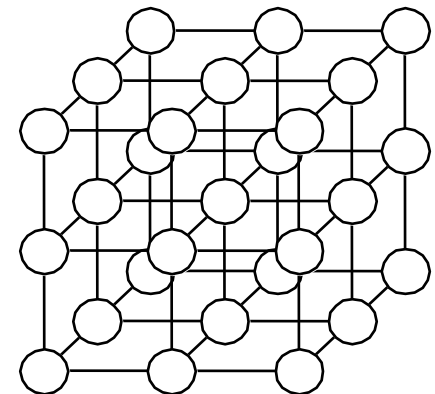
- **Mesh:** generalization of linear array to 2D
 - nodes have 4 neighbors: north, south, east, and west.
- **k -dimensional mesh**
 - node have $2k$ neighbors



2D mesh



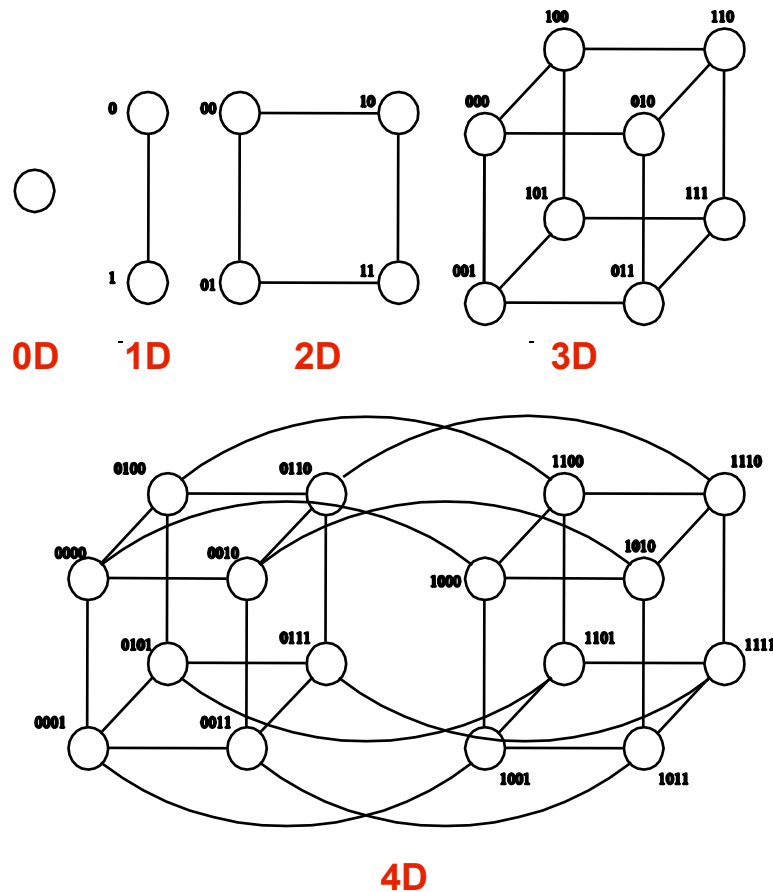
2D torus



3D mesh

Hypercubes

Special d-dimensional mesh: p nodes, $d = \log p$

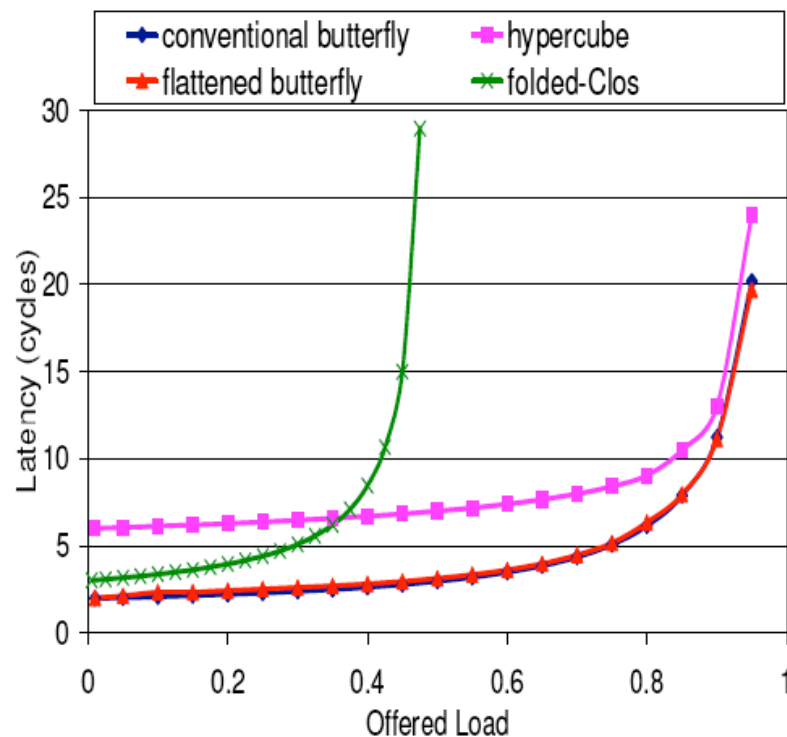


Constructing hypercubes from hypercubes of lower dimension

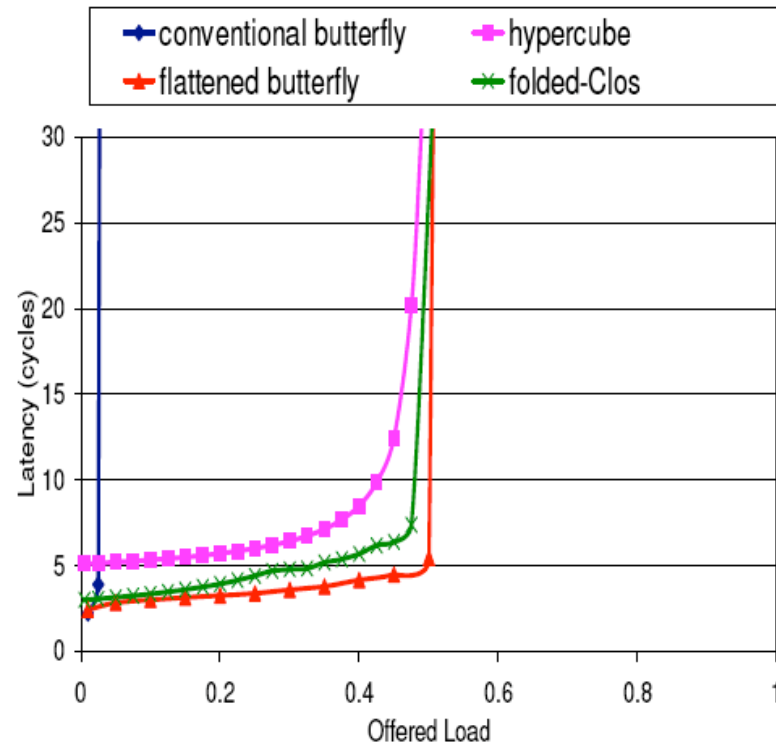
Hypercube Properties

- Distance between any two nodes is at most $\log p$.
- Each node has $\log p$ neighbors
- Distance between two nodes =
 # of bit positions that differ between node numbers

Comparing Network Performance



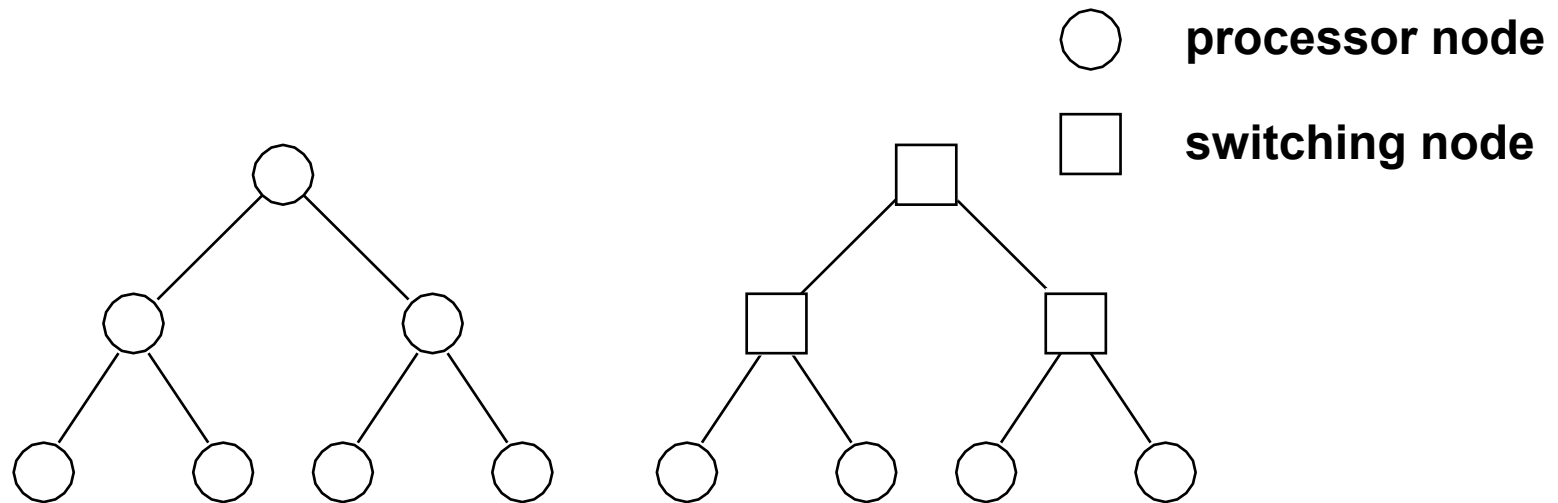
Uniform Random Traffic



Worst Case Traffic

John Kim, William J. Dally, Dennis Abts: Flattened butterfly: a cost-efficient topology for high-radix networks. *ISCA 2007*: 126-137

Trees



static tree network

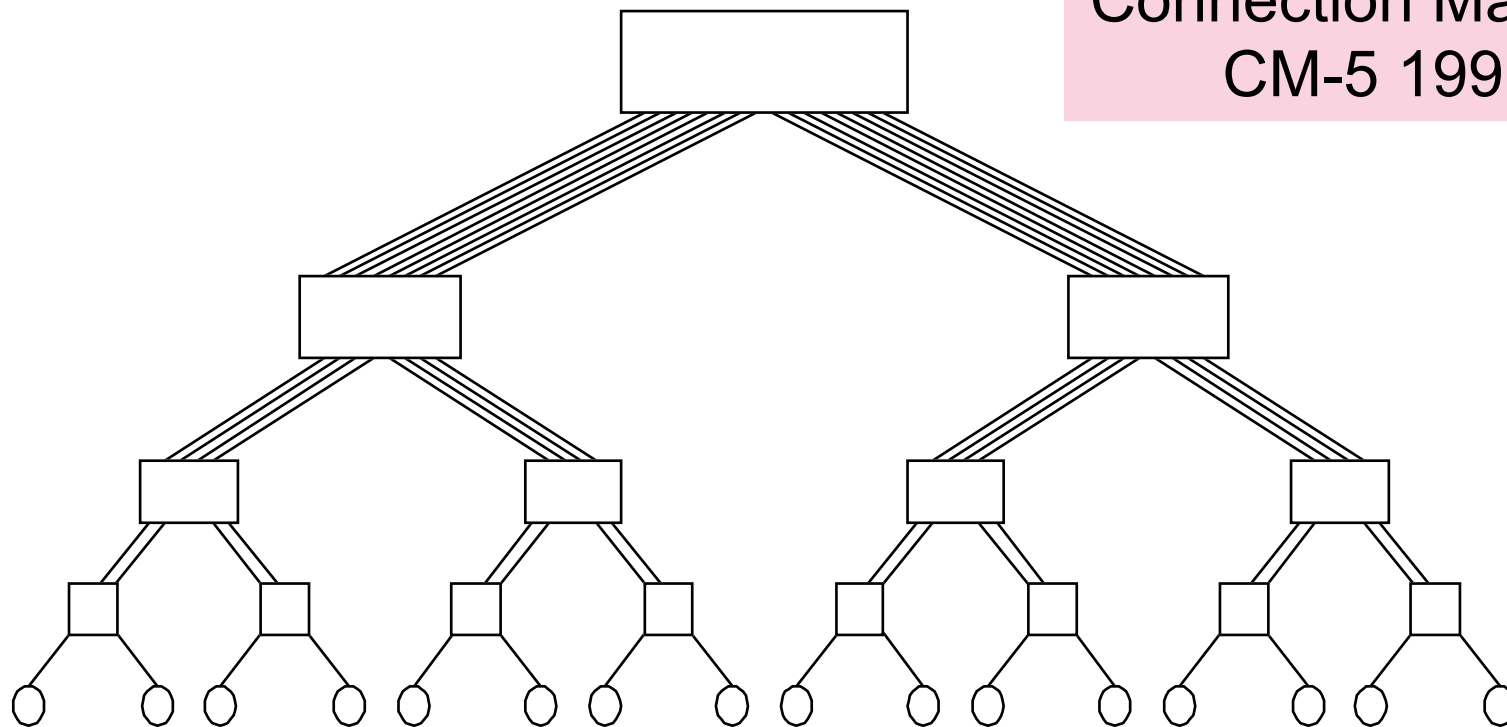
dynamic tree network

Examples of complete binary tree networks

Tree Properties

- Distance between any two nodes is no more than $2 \log p$
- Trees can be laid out in 2D with no wire crossings
- Problem
 - links closer to root carry > traffic than those at lower levels.
- Solution: fat tree
 - widen links as depth gets shallower
 - copes with higher traffic on links near root

Fat Tree Network



Connection Machine
CM-5 1991

Fat tree network for 16 processing nodes

- Can judiciously choose “fatness” of links
 - take full advantage of technology and packaging constraints

Charles Leiserson. Fat Trees: Universal Networks for Hardware-Efficient Supercomputing. IEEE Transactions on Computers, C-34:10, Oct. 1985.

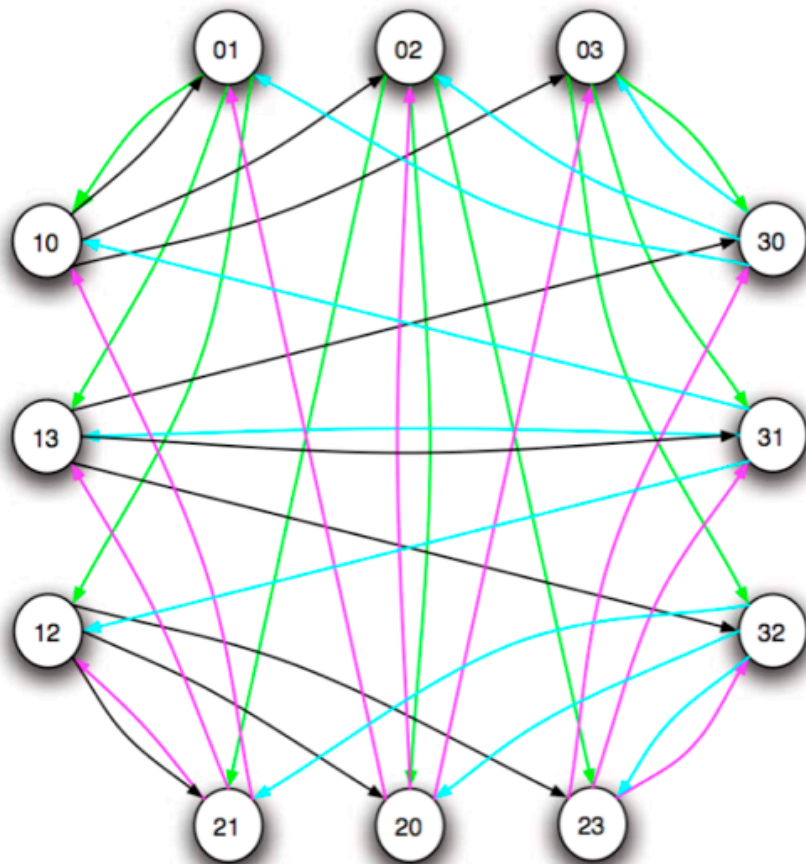
Fat Tree Properties

“We prove that for any given amount of communications hardware, a fat-tree built from that amount of hardware can simulate every other network built from the same amount of hardware, using only slightly more time (a polylogarithmic factor greater). The basic assumption we make of competing networks is the following. In unit time, at most $O(a)$ bits can enter or leave a closed 3D region with surface area a .”

Charles Leiserson. Fat Trees: Universal Networks for Hardware-Efficient Supercomputing. IEEE Transactions on Computers, C-34:10, Oct. 1985.

This paper proves the universality result for off-line simulations only.

Kautz Graph Network



Diameter	2	3	4	5	6	7
Degree 2	6	12	24	48	96	192
Degree 3	12	36	108	324	972	2916
Degree 4	20	80	320	1280	5129	20480

W. H. Kautz, Bounds on directed (d,k) graphs, Theory of cellular logic networks and machines, AFCRL-68-0668 Final report, pp. 20-28, 1968.

SiCortex5832: 5832 cores (6-core MIPS), 972 nodes, diameter 6, 2916 links. (2003-2009).

Properties

- Largest # nodes for a fixed degree and diameter
- Diameter grows as $\log(\# \text{ nodes})$

Metrics for Interconnection Networks

- **Diameter:** longest distance between two nodes in the network
 - examples
 - linear array: $p - 1$
 - mesh: $2(\sqrt{p} - 1)$
 - tree and hypercube: $O(\log p)$
 - completely connected network: $O(1)$
- **Bisection Width:** min # of wire cuts to divide the network in 2 halves
 - examples
 - linear array and tree: 1
 - mesh: \sqrt{p}
 - hypercube: $p/2$
 - completely connected network: $p^2/4$
- **Cost:** \sim # links or switches (whichever is asymptotically higher)
 - other cost issues
 - ability to layout the network
 - length of wires

Case Study: SGI Altix UV

SGI Altix UV (2010): Node

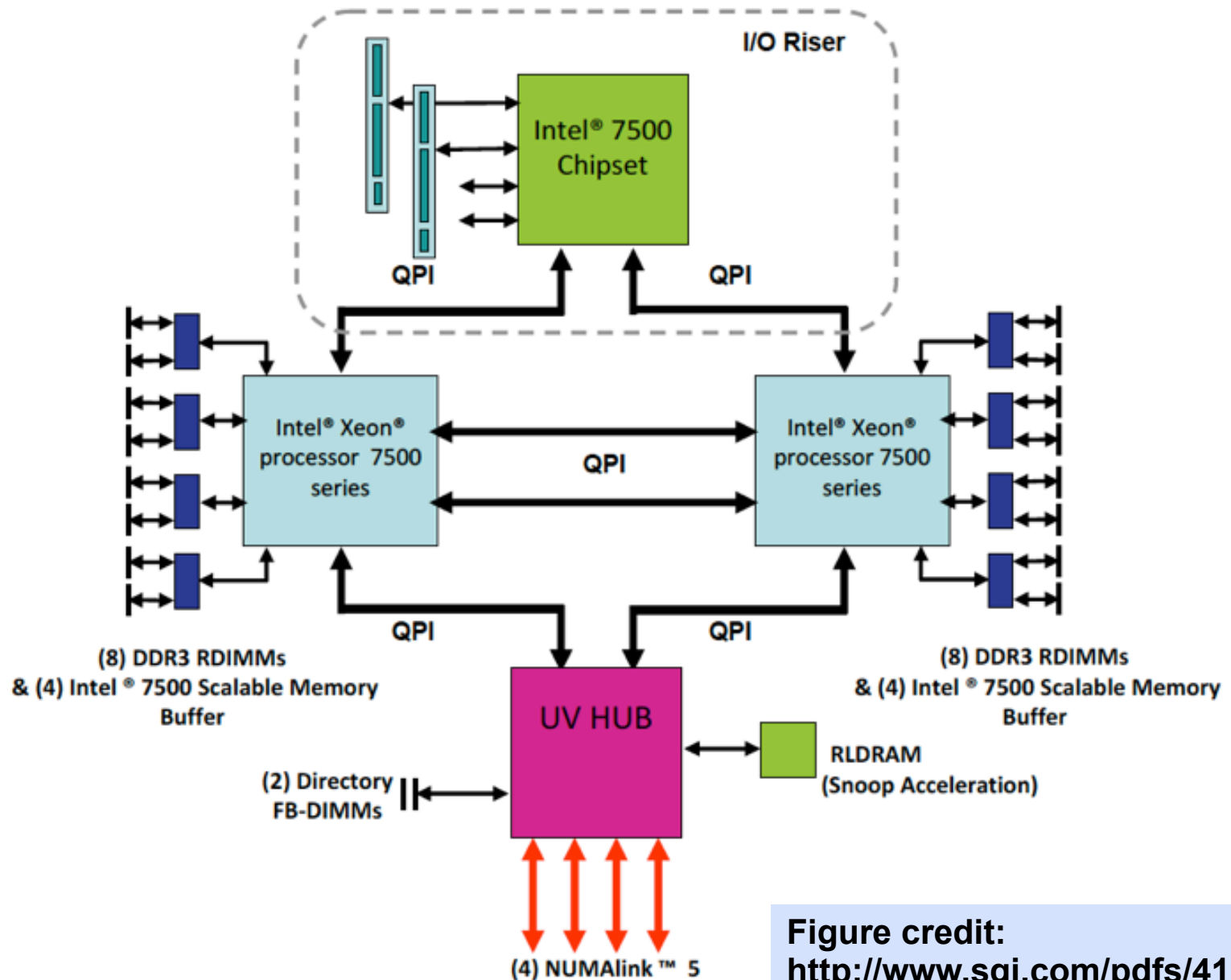
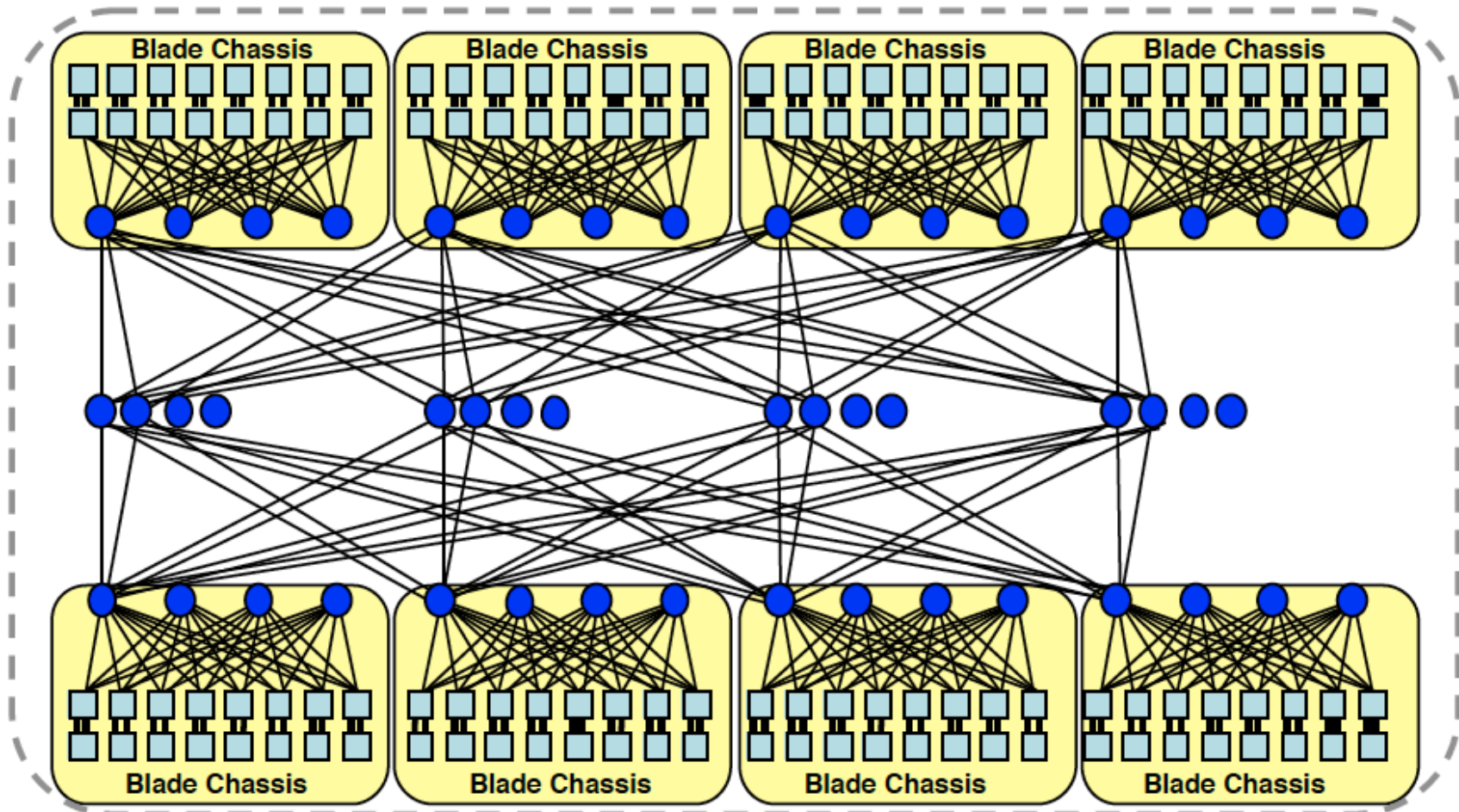


Figure credit:
<http://www.sgi.com/pdfs/4192.pdf>

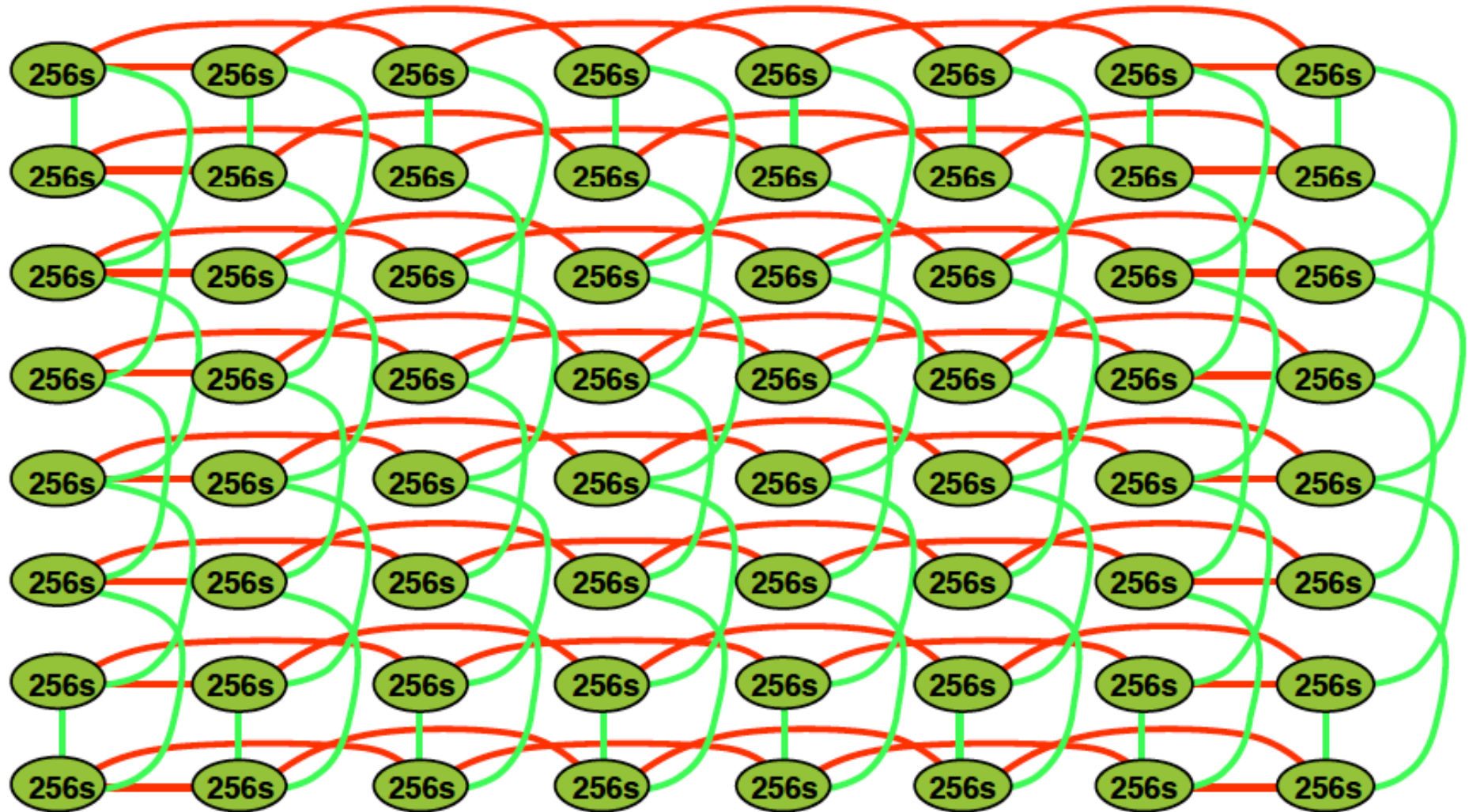
SGI Altix UV - Scalable Unit



256-socket building block; fat tree (indirect)

Figure credit: <http://www.sgi.com/pdfs/4192.pdf>

SGI Altix UV - System

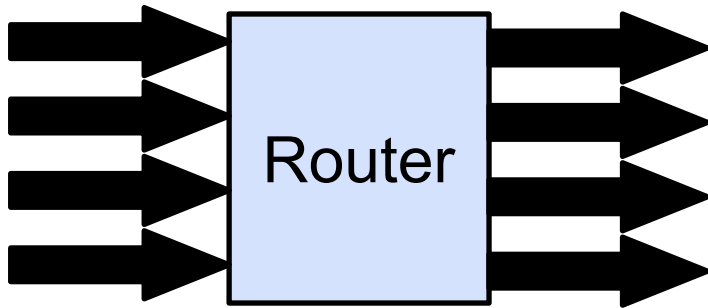


16,384 socket (131,072 core); torus (direct)

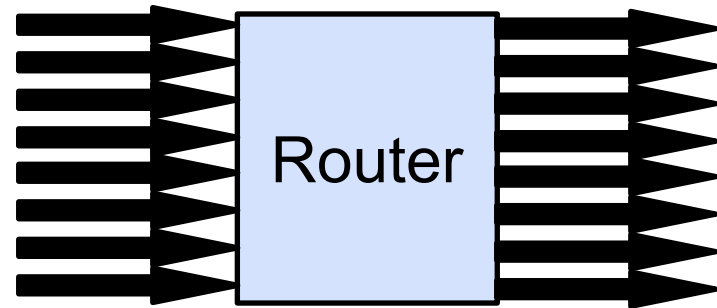
Figure credit: <http://www.sgi.com/pdfs/4192.pdf>

Dragonfly

The Trend in Routers

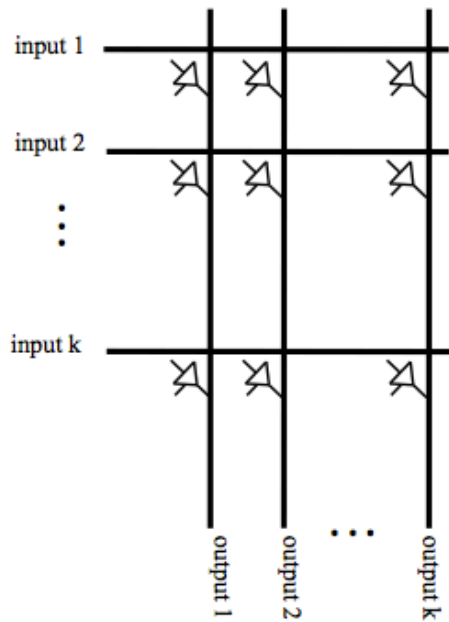


Low radix router
(small number of fat ports)

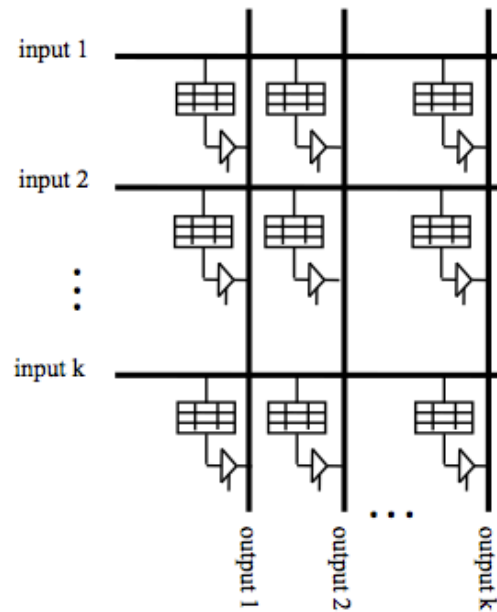


High radix router
(large number of skinny ports)

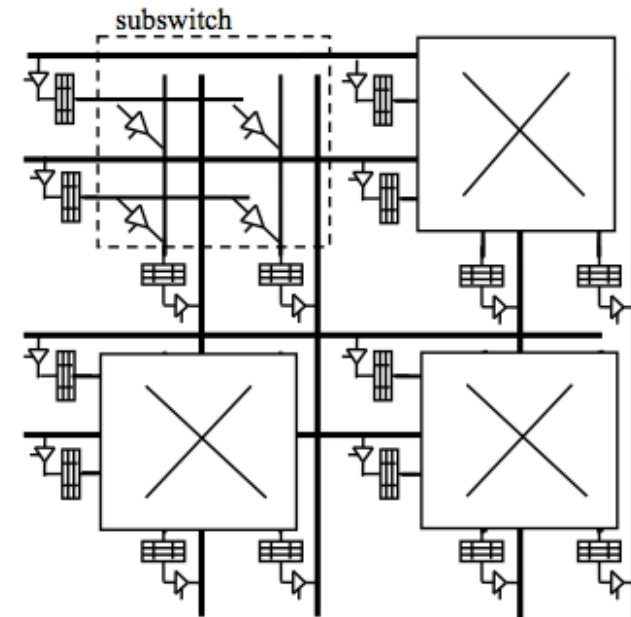
High Radix Routers



(a) Baseline design



(b) Fully buffered crossbar



(c) Hierarchical crossbar

Dragonfly: Three Level Network

- Levels

- router
- group
- system

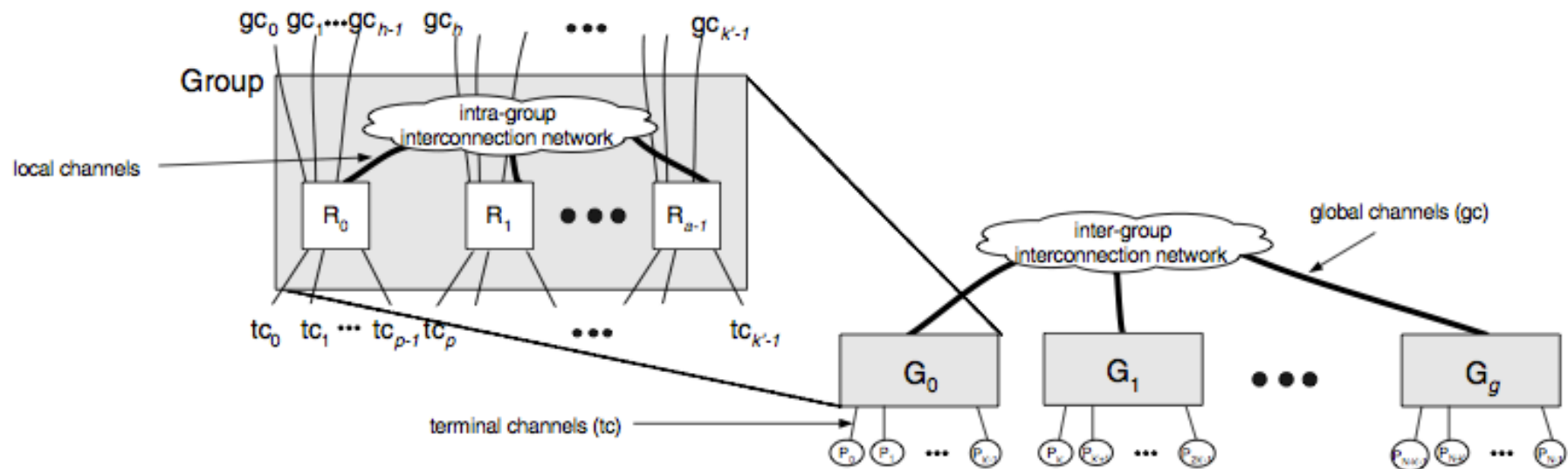
router has to connect to

p terminals

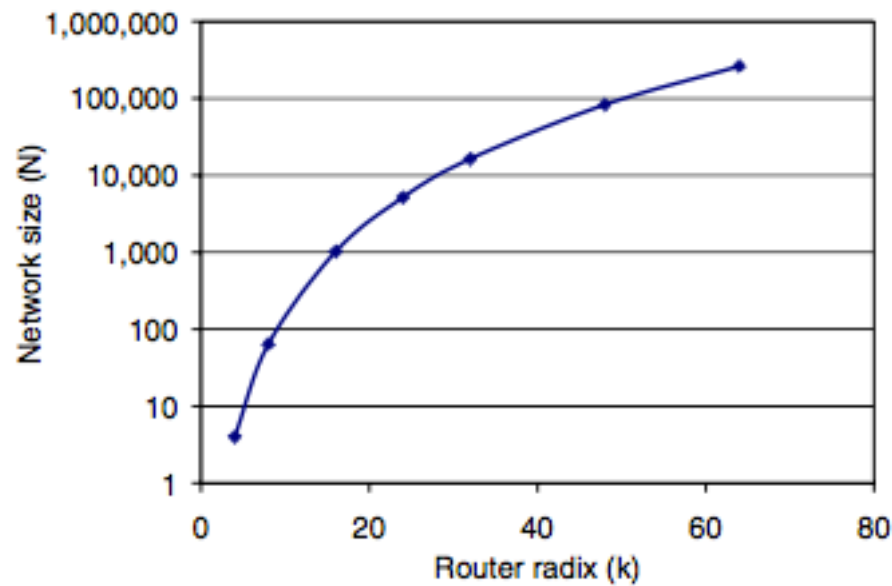
a - 1 routers within the same group

h global channels to other groups

radix = $p + a - 1 + h$



Dragonfly Scalability



network scale vs.
router radix

Valiant's Randomized Routing

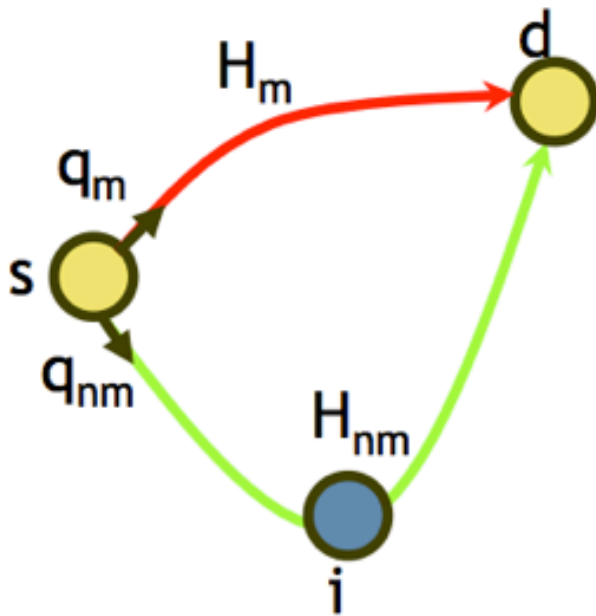
- Avoid hot spots with two step routing
 - message from $s \rightarrow d$
 - first sent to a randomly chosen intermediate processor i
 - then forward from i to destination d
 - reduces a worst case permutation route to two randomized routing steps*
 - *one with randomly picked source nodes*
 - *a second with randomly picked destination nodes*

L. G. Valiant. A scheme for fast parallel communication.
SIAM Journal on Computing, 11(2):350–361, 1982.

Global Adaptive Routing

- VAL gives optimal worst-case throughput
- MIN gives optimal benign traffic performance
- UGAL (Universal Globally Adaptive Load-balance)
 - [Singh '05]
 - Routes benign traffic minimally
 - Starts routing like VAL if load imbalance in channel queues
 - In the worst-case, degenerates into VAL, thus giving optimal worst-case throughput

UGAL



1. H_m = shortest path (SP) length

2. q_m = congestion of the outgoing channel for SP

3. Pick i , a random intermediate node

4. H_{nm} = non-min path ($s \rightarrow i \rightarrow d$) length

5. q_{nm} = congestion of the outgoing channel for $s \rightarrow i \rightarrow d$

6. Choose SP if $H_m q_m \leq H_{nm} q_{nm}$; else route via i , minimally in each phase

Dragonfly Performance

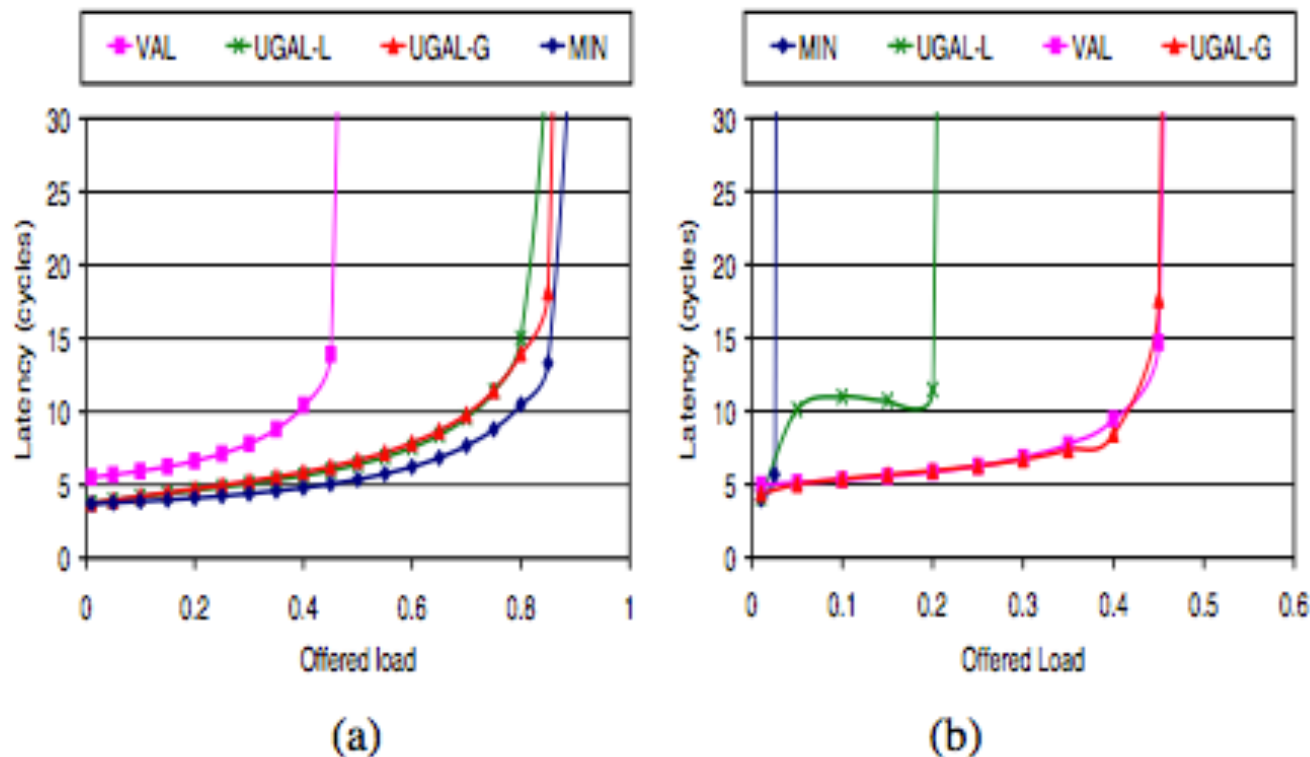
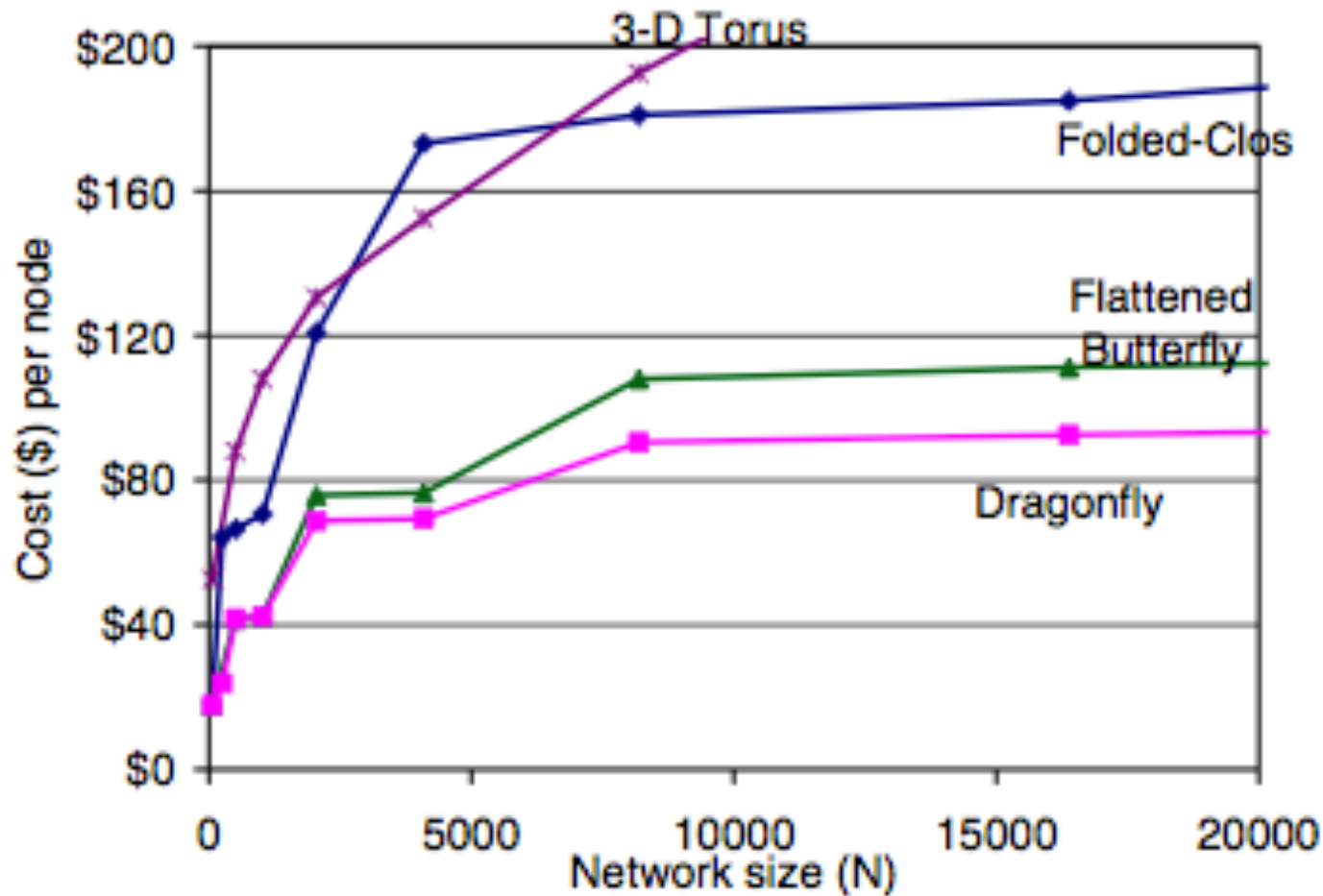


Figure 8. Routing algorithm comparison on the dragonfly for (a) uniform random traffic and (b) adversarial traffic pattern.

adversarial traffic pattern: each node in a group sends to randomly selected node in another group

Cost Comparison



Points to Remember

- Indirect networks
 - high-radix routers are better
 - many thin links rather than fewer fat links
 - networks built from high-radix routers have lower latency and cost
 - Clos or flat butterfly have good cost and performance
- Direct networks
 - 3D Torus popular for very large networks
 - good bisection bandwidth - $2p^{2/3}$
 - modest number of links - $3p$
 - low fixed degree - 6
- Hybrid configurations
 - SGI UV: QPI direct connect, fat tree (indirect), torus (direct)
 - balance latency vs. cost
- Current supercomputers
 - Cray XC30: dragonfly using high radix routers
 - Blue Gene/Q: 5D torus

References

- Adapted from slides “Parallel Programming Platforms” by Ananth Grama
- Based on Chapter 2 of “Introduction to Parallel Computing” by Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar. Addison Wesley, 2003
- John Kim, William J. Dally, Dennis Abts: Flattened butterfly: a cost-efficient topology for high-radix networks. ISCA 2007: 126-137.
- Lawrence C. Stewart and David Gingold. A New Generation of Cluster Interconnect . SiCortex White Paper. December 2006/ revised April 2008.
- Technical Advances in the SGI® Altix® UV Architecture.
<http://www.sgi.com/pdfs/4192.pdf>