Interconnection Networks: Topology and Routing

Natalie EnrightJerger

Topology Overview

- Definition: determines arrangement of channels and nodes in network
- Analogous to road map
- Often first step in network design
- Routing and flow control build on properties of topology

Abstract Metrics

- Use metrics to evaluate performance and cost of topology
- Also influenced by routing/flow control
  - At this stage
    - Assume ideal routing (perfect load balancing)
    - Assume ideal flow control (no idle cycles on any channel)
- Switch Degree: number of links at a node
  - Proxy for estimating cost
    - Higher degree requires more links and port counts at each router

Latency

- Time for packet to traverse network
  - Start: head arrives at input port
  - End: tail departs output port
- Latency = Head latency + serialization latency
  - Serialization latency: time for packet with Length L to cross channel with bandwidth b (L/b)
- Hop Count: the number of links traversed between source and destination
  - Proxy for network latency
  - Per hop latency with zero load

Impact of Topology on Latency

- Impacts average minimum hop count
- Impact average distance between routers
- Bandwidth

Throughput

- Data rate (bits/sec) that the network accepts per input port
- Max throughput occurs when one channel saturates
  - Network cannot accept any more traffic
- Channel Load
  - Amount of traffic through channel c if each input node injects 1 packet in the network

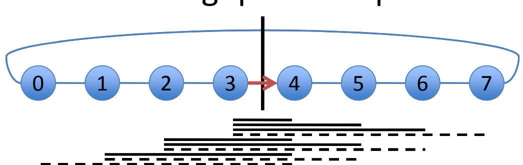
Maximum channel load

- Channel with largest fraction of traffic
- Max throughput for network occurs when channel saturates
  - Bottleneck channel

Bisection Bandwidth

- Cuts partition all the nodes into two disjoint sets
  - Bandwidth of a cut
- Bisection
  - A cut which divides all nodes into nearly half
  - Channel bisection → min. channel count over all bisections
  - Bisection bandwidth →min. bandwidth over all bisections
- With uniform traffic
  - ½ of traffic cross bisection

Throughput Example



- Bisection = 4 (2 in each direction)
- With uniform random traffic
  - 3 sends 1/8 of its traffic to 4,5,6
  - 3 sends 1/16 of its traffic to 7 (2 possible shortest paths)
  - 2 sends 1/8 of its traffic to 4,5
  - Etc
- Channel load = 1

Path Diversity

- Multiple minimum length paths between source and destination pair
- Fault tolerance
- Better load balancing in network
- Routing algorithm should be able to exploit path diversity
- We'll see shortly
  - Butterfly has no path diversity
  - Torus can exploit path diversity

Path Diversity (2)

- Edge disjoint paths: no links in common
- Node disjoint paths: no nodes in common except source and destination
- If j = minimum number of edge/node disjoint paths between any source-destination pair
  - Network can tolerate j link/node failures

Symmetry

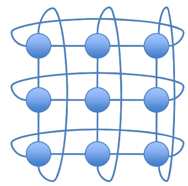
- Vertex symmetric:
  - An automorphism exists that maps any node a onto another node b
  - Topology same from point of view of all nodes
- Edge symmetric:
  - An automorphism exists that maps any channel a onto another channel b

Direct & Indirect Networks

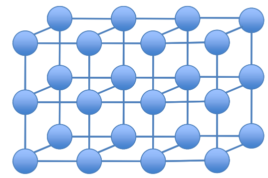
- Direct: Every switch also network end point
  - Ex: Torus
- Indirect: Not all switches are end points
  - Ex: Butterfly

Torus (1)

- K-ary n-cube: k^n network nodes
- n-dimensional grid with k nodes in each dimension



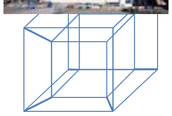
3-ary 2-cube



2,3,4-ary 3-mesh

Torus (2)

- Topologies in Torus Family
  - Ring k-ary 1-cube
  - Hypercubes 2-ary n-cube
- Edge Symmetric
  - Good for load balancing
  - Removing wrap-around links for mesh loses edge symmetry
    - More traffic concentrated on center channels
- Good path diversity
- Exploit locality for near-neighbor traffic



Torus (3)

$$H_{\min} = \begin{cases} \frac{nk}{4} & k \text{ even} \\ n\left(\frac{k}{4} - \frac{1}{4k}\right) & k \text{ odd} \end{cases}$$

- Hop Count:
- Degree = 2n, 2 channels per dimension

Channel Load for Torus

- Even number of k-ary (n-1)-cubes in outer dimension
- Dividing these k-ary (n-1)-cubes gives a 2 sets of k^{n-1} bidirectional channels or $4k^{n-1}$
- ½ Traffic from each node cross bisection

$$\text{channel load} = \frac{N}{2} \times \frac{k}{4N} = \frac{k}{8}$$

- Mesh has ½ the bisection bandwidth of torus

Torus Path Diversity

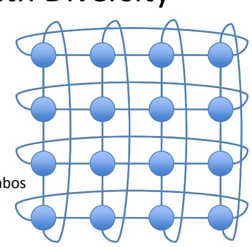
$$|R_{xy}| = \binom{\Delta x + \Delta y}{\Delta x}$$

2 dimensions*

$$\Delta x = 2, \Delta y = 2$$

$$|R_{xy}| = 6$$

$|R_{xy}| = 24$ NW, NE, SW, SE combos



$$|R_{xy}| = \prod_{i=0}^{n-1} \binom{\sum_{j=i}^{n-1} \Delta j}{\Delta i} = \frac{\left(\sum_{i=0}^{n-1} \Delta i\right)!}{\prod_{i=0}^{n-1} \Delta i!}$$

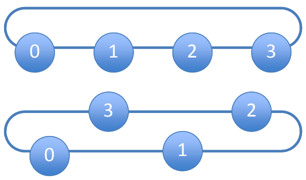
2 edge and node disjoint minimum paths

n dimensions with Δi hops in i dimension

*assume single direction for x and y

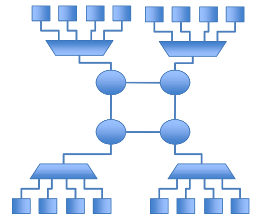
Implementation

- Folding
  - Equalize path lengths
    - Reduces max link length
    - Increases length of other links



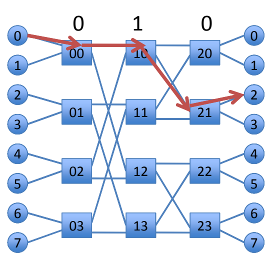
Concentration

- Don't need 1:1 ratio of network nodes and cores/memory
- Ex: 4 cores concentrated to 1 router



Butterfly

- K-ary n-fly: k^n network nodes
- Example: 2-ary 3-fly
- Routing from 000 to 010
  - Dest address used to directly route packet
  - Bit n used to select output port at stage n



Butterfly (2)

- No path diversity $|R_{xy}| = 1$
- Hop Count
  - $\log_k n + 1$
  - Does not exploit locality
    - Hop count same regardless of location
- Switch Degree = 2k
- Channel Load → uniform traffic

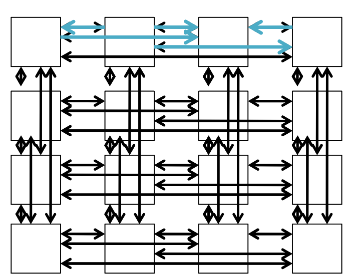
$$\frac{NH_{min}}{C} = \frac{k^n(n+1)}{k^n(n+1)} = 1$$

  - Increases for adversarial traffic

Flattened Butterfly

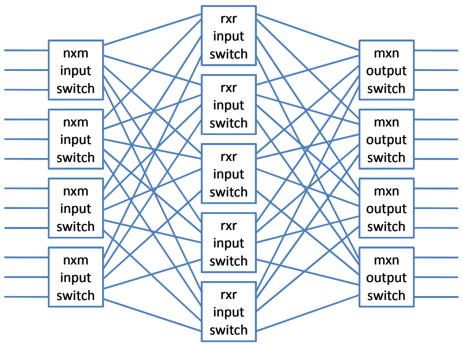
- Proposed by Kim et al (ISCA 2007)
  - Adapted for on-chip (MICRO 2007)
- Advantages
  - Max distance between nodes = 2 hops
  - Lower latency and improved throughput compared to mesh
- Disadvantages
  - Requires higher port count on switches (than mesh, torus)
  - Long global wires
  - Need non-minimal routing to balance load

Flattened Butterfly



- Path diversity through non-minimal routes

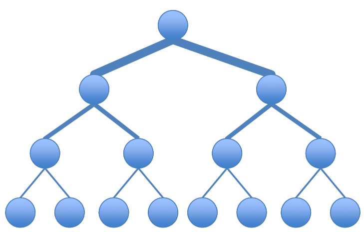
Clos Network



Clos Network

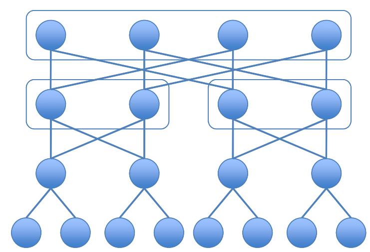
- 3-stage indirect network
- Characterized by triple (m, n, r)
  - M: # of middle stage switches
  - N: # of input/output ports on input/output switches
  - R: # of input/output switching
- Hop Count = 4

Folded Clos (Fat Tree)



- Bandwidth remains constant at each level
- Regular Tree: Bandwidth decreases closer to root

Fat Tree (2)



- Provides path diversity

Common On-Chip Topologies

- Torus family: mesh, concentrated mesh, ring
  - Extending to 3D stacked architectures
  - Favored for low port count switches
- Butterfly family: Flattened butterfly

Topology Summary

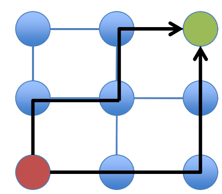
- First network design decision
- Critical impact on network latency and throughput
  - Hop count provides first order approximation of message latency
  - Bottleneck channels determine saturation throughput

Routing Overview

- Discussion of topologies assumed ideal routing
- Practically though routing algorithms are not ideal
- Discuss various classes of routing algorithms
  - Deterministic, Oblivious, Adaptive
- Various implementation issues
  - Deadlock

Routing Basics

- Once topology is fixed
- Routing algorithm determines path(s) from source to destination



Routing Algorithm Attributes

- Number of destinations
  - Unicast, Multicast, Broadcast?
- Adaptivity
  - Oblivious or Adaptive? Local or Global knowledge?
- Implementation
  - Source or node routing?
  - Table or circuit?

Oblivious

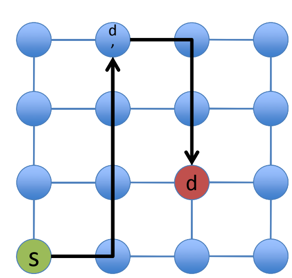
- Routing decisions are made without regard to network state
  - Keeps algorithms simple
  - Unable to adapt
- Deterministic algorithms are a subset of oblivious

Deterministic

- All messages from Src to Dest will traverse the same path
- Common example: Dimension Order Routing (DOR)
  - Message traverses network dimension by dimension
  - Aka XY routing
- Cons:
  - Eliminates any path diversity provided by topology
  - Poor load balancing
- Pros:
  - Simple and inexpensive to implement
  - Deadlock free

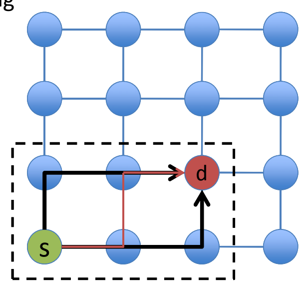
Valiant's Routing Algorithm

- To route from s to d, randomly choose intermediate node d'
  - Route from s to d' and from d' to d.
- Randomizes any traffic pattern
  - All patterns appear to be uniform random
  - Balances network load
- Non-minimal



Minimal Oblivious

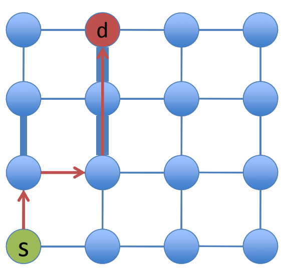
- Valiant's: Load balancing comes at expense of significant hop count increase
  - Destroys locality
- Minimal Oblivious: achieve some load balancing, but use shortest paths
  - d' must lie within minimum quadrant
  - 6 options for d'
  - Only 3 different paths



Adaptive

- Uses network state to make routing decisions
  - Buffer occupancies often used
  - Couple with flow control mechanism
- Local information readily available
  - Global information more costly to obtain
  - Network state can change rapidly
  - Use of local information can lead to non-optimal choices
- Can be minimal or non-minimal

Minimal Adaptive Routing

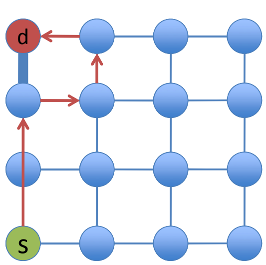


- Local info can result in sub-optimal choices

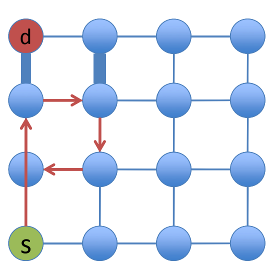
Non-minimal adaptive

- Fully adaptive
- Not restricted to take shortest path
  - Example: FBfly
- Misrouting: directing packet along non-productive channel
  - Priority given to productive output
  - Some algorithms forbid U-turns
- Livelock potential: traversing network without ever reaching destination
  - Mechanism to guarantee forward progress
    - Limit number of misroutings

Non-minimal routing example

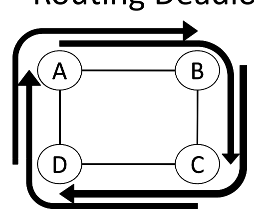


- Longer path with potentially lower latency



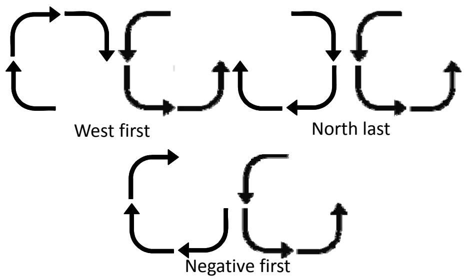
- Livelock: continue routing in cycle

Routing Deadlock



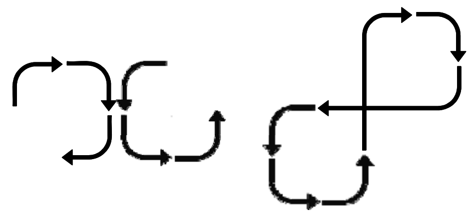
- Without routing restrictions, a resource cycle can occur
  - Leads to deadlock

Turn Model Routing



- Some adaptivity by removing 2 of 8 turns
  - Remains deadlock free (like DOR)

Turn Model Routing Deadlock



- Not a valid turn elimination
  - Resource cycle results

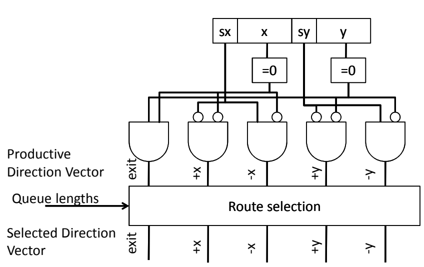
Routing Implementation

- Source tables
  - Entire route specified at source
  - Avoids per-hop routing latency
  - Unable to adapt to network conditions
  - Can specify multiple routes per destination
- Node tables
  - Store only next routes at each node
  - Smaller tables than source routing
  - Adds per-hop routing latency
  - Can adapt to network conditions
    - Specify multiple possible outputs per destination

Implementation

- Combinational circuits can be used
  - Simple (e.g. DOR): low router overhead
  - Specific to one topology and one routing algorithm
    - Limits fault tolerance
- Tables can be updated to reflect new configuration, network faults, etc

Circuit Based



Routing Summary

- Latency paramount concern
  - Minimal routing most common for NoC
  - Non-minimal can avoid congestion and deliver low latency
- To date: NoC research favors DOR for simplicity and deadlock freedom
  - On-chip networks often lightly loaded
- Only covered unicast routing
  - Recent work on extending on-chip routing to support multicast

Bibliography

- Topology
  - William J. Dally and C. L Seitz. The torus routing chip. Journal of Distributed Computing, 1(3):187–196, 1986.
  - Charles Leiserson. Fat-trees: Universal networks for hardware efficient supercomputing. IEEE Transactions on Computers, 34(10), October 1985.
  - Boris Grot, Joel Hestness, Stephen W. Keckler, and OnurMutlu. Express cube topologies for on-chip networks. In Proceedings of the International Symposium on High Performance Computer Architecture, February 2009.
  - Flattened butterfly topology for on-chip networks. In Proceedings of the 40th International Symposium on Microarchitecture, December 2007.
  - J. Balfour and W. Dally. Design tradeoffs for tiled cmp on-chip networks. In Proceedings of the International Conference on Supercomputing, 2006.
- Routing
  - L. G. Valiant and G. J. Brebner. Universal schemes for parallel communication. In Proceedings of the 13th Annual ACM Symposium on Theory of Computing, pages 263–277, 1981.
  - D. Seo, A. Ali, W.-T. Lim, N. Rafique, and M. Thottenhodi. Near-optimal worst- case throughput routing in two dimensional mesh networks. In Proceedings of the 32nd Annual International Symposium on Computer Architecture, June.
  - Christopher J. Glass and Lionel M. Ni. The turn model for adaptive routing. In Proceedings of the International Symposium on Computer Architecture, 1992.
  - P. Gratz, B. Grot, and S. W. Keckler, “Regional congestion awareness for load balance in networks-on-chip,” in Proceedings of the 14th IEEE International Symposium on High-Performance Computer Architecture, February 2008.
  - N. EnrightJerger, L.-S. Peh, and M. H. Lipasti, “Virtual circuit tree multi- casting: A case for on-chip hardware multicast support,” in Proceedings of the International Symposium on Computer Architecture (ISCA-35), Beijing, China, June 2008.