Topological Properties

- Routing Distance - number of links on route
- Diameter - maximum routing distance
- Average Distance
- A network is partitioned by a set of links if their removal disconnects the graph

Interconnection Topologies

- Class of networks scaling with N
- Logical Properties:
  - distance, degree
- Physical properties
  - length, width
- Fully connected network
  - diameter = 1
  - degree = N
  - cost?
    - bus => O(N), but BW is O(1) - actually worse
    - crossbar => O(N^2) for BW O(N)
- VLSI technology determines switch degree

Example: Linear Arrays and Rings

- Linear Array
- Torus
  - Torus arranged to use short wires
  - Linear Array
    - Diameter?
    - Average Distance?
    - Bisection bandwidth?
    - Route A -> B given by relative address R = B-A
- Torus?
- Examples: FDDI, SCI, FiberChannel Arbitrated Loop, KSR1
Example: Multidimensional Meshes and Tori

- **n-dimensional array**
  - $N = k_{n-1} \times \ldots \times k_0$ nodes
  - described by $n$-vector of coordinates $(i_{n-1}, \ldots, i_0)$
- **n-dimensional $k$-ary mesh**: $N = k^n$
  - $k = \sqrt[n]{N}$
  - described by $n$-vector of radix $k$ coordinate
- **n-dimensional $k$-ary torus (or $k$-ary $n$-cube)?

On Chip: Embeddings in two dimensions

- Embed multiple logical dimension in one physical dimension using long wires
- When embedding higher-dimension in lower one, either some wires longer than others, or all wires long

Trees

- Diameter and ave distance logarithmic
  - $k$-ary tree, height $n = \log_k N$
  - address specified $n$-vector of radix $k$ coordinates describing path down from root
- Fixed degree
- Route up to common ancestor and down
  - $R = B \oplus A$
  - let $i$ be position of most significant 1 in $R$, route up $i+1$ levels
  - down in direction given by low $i+1$ bits of $B$
- $H$-tree space is $O(N)$ with $O(\sqrt[3]{N})$ long wires
- Bisection BW?

Fat-Trees

- Fatter links (really more of them) as you go up, so bisection BW scales with $N$
**Butterflies**

- Tree with lots of roots!
- $N \log N$ (actually $N/2 \times \log N$)
- Exactly one route from any source to any dest
- $R = A \oplus B$, at level $i$ use ‘straight’ edge if $r_i=0$, otherwise cross edge
- Bisection $N/2$ vs $N^{(n-1)/n}$ (for n-cube)

**k-ary n-cubes vs k-ary n-flies**

- degree $n$ vs degree $k$
- $N$ switches vs $N \log N$ switches
- diminishing BW per node vs constant
- requires locality vs little benefit to locality
- Can you route all permutations?

**Benes network and Fat Tree**

- Back-to-back butterfly can route all permutations
- What if you just pick a random mid point?

**Hypercubes**

- Also called binary n-cubes. # of nodes = $N = 2^n$.
- $O(\log N)$ Hops
- Good bisection BW
- Complexity
  - Out degree is $n = \log N$
  - correct dimensions in order
  - with random comm. 2 ports per processor
**Some Properties**

- **Routing**
  - relative distance: \( R = (b_{n-1} - a_{n-1}, \ldots, b_0 - a_0) \)
  - traverse \( r_i = b_i - a_i \) hops in each dimension
  - *dimension-order routing? Adaptive routing?*

- **Average Distance**
  - Wire Length?
  - \( n \times 2k/3 \) for mesh
  - \( nk/2 \) for cube

- **Degree?**

- **Bisection bandwidth?**
  - \( k^{n-1} \) bidirectional links

- **Physical layout?**
  - 2D in \( O(N) \) space
  - higher dimension?

---

**The Routing problem: Local decisions**

- Routing at each hop: Pick next output port!

---

**How do you build a crossbar?**

- Independent routing logic per input
  - FSM

- Scheduler logic arbitrates each output
  - priority, FIFO, random

- Head-of-line blocking problem
  - Message at head of queue blocks messages behind it

---

**Input buffered switch**

- Independent routing logic per input
  - FSM

- Scheduler logic arbitrates each output
  - priority, FIFO, random

- Head-of-line blocking problem
  - Message at head of queue blocks messages behind it
Output Buffered Switch

• How would you build a shared pool?

Properties of Routing Algorithms

• Routing algorithm:
  – \( R : N \times N \rightarrow C \), which at each switch maps the destination node \( n_d \) to the next channel on the route
  – which of the possible paths are used as routes?
  – how is the next hop determined?
    » arithmetic
    » source-based port select
    » table driven
    » general computation
• Deterministic
  – route determined by (source, dest), not intermediate state (i.e. traffic)
• Adaptive
  – route influenced by traffic along the way
• Minimal
  – only selects shortest paths
• Deadlock free
  – no traffic pattern can lead to a situation where packets are deadlocked and never move forward

Example: Simple Routing Mechanism

• need to select output port for each input packet
  – in a few cycles
• Simple arithmetic in regular topologies
  – ex: \( \Delta x, \Delta y \) routing in a grid
    » west (-x) \( \Delta x < 0 \)
    » east (+x) \( \Delta x > 0 \)
    » south (-y) \( \Delta x = 0, \Delta y < 0 \)
    » north (+y) \( \Delta x = 0, \Delta y > 0 \)
    » processor \( \Delta x = 0, \Delta y = 0 \)
• Reduce relative address of each dimension in order
  – Dimension-order routing in k-ary d-cubes
  – e-cube routing in n-cube

Communication Performance

• Typical Packet includes data + encapsulation bytes
  – Unfragmented packet size \( S = S_{data} + S_{encapsulation} \)
• Routing Time:
  – \( \text{Time}(S)_{s-d} = \text{overhead} + \text{routing delay} + \text{channel occupancy} + \text{contention delay} \)
  – Channel occupancy = \( \frac{S_{data} + S_{encapsulation}}{b} \)
  – Routing delay in cycles (\( \Delta \)):
    » Time to get head of packet to next hop
  – Contention?
Store & Forward vs Cut-Through Routing

- Time: \( h(S/b + \Delta/\tau) \) vs \( S/b + h/\tau \)
- OR(cycles): \( h(S/w + \Delta) \) vs \( S/w + h/\Delta \)

- what if message is fragmented?
- wormhole vs virtual cut-through

Contention

- Two packets trying to use the same link at same time
  - limited buffering
  - drop?
- Most parallel mach. networks block in place
  - link-level flow control
  - tree saturation
- Closed system - offered load depends on delivered
  - Source Squelching

Bandwidth

- What affects local bandwidth?
  - packet density: \( b \times S_{data}/S \)
  - routing delay: \( b \times S_{data} / (S + w\Delta) \)
  - contention
    - endpoints
    - within the network
- Aggregate bandwidth
  - bisection bandwidth
    - sum of bandwidth of smallest set of links that partition the network
  - total bandwidth of all the channels: \( Cb \)
  - suppose \( N \) hosts issue packet every \( M \) cycles with ave dist
    - each msg occupies \( h \) channels for \( l = S/w \) cycles each
    - \( C/N \) channels available per node
    - link utilization for store-and-forward:
      \( \rho = (hl/M \text{ channel cycles/node})/(C/N) = Nh/M \text{ } < 1! \)
    - link utilization for wormhole routing?

Saturation

- Delivered Bandwidth vs Offered Bandwidth
- Latency vs Saturation
How Many Dimensions?

- n = 2 or n = 3
  - Short wires, easy to build
  - Many hops, low bisection bandwidth
  - Requires traffic locality
- n >= 4
  - Harder to build, more wires, longer average length
  - Fewer hops, better bisection bandwidth
  - Can handle non-local traffic
- k-ary n-cubes provide a consistent framework for comparison
  - N = kn
  - scale dimension (n) or nodes per dimension (k)
  - assume cut-through

Traditional Scaling: Latency scaling with N

- Assumes equal channel width
  - independent of node count or dimension
  - dominated by average distance

Average Distance

ave dist = n(k-1)/2

- but, equal channel width is not equal cost!
- Higher dimension => more channels

Dally Paper: In the 3D world

- For N nodes, bisection area is O(N^{2/3})

  - Bill Dally, IEEE TPDS, [Dal90a]
  - For fixed bisection bandwidth, low-dimensional k-ary n-cubes are better (otherwise higher is better)
  - i.e., a few short fat wires are better than many long thin wires
  - What about many long fat wires?
Dally paper (con’t)

- Equal Bisection, W=1 for hypercube \( \Rightarrow W = \frac{1}{2}k \)
- Three wire models:
  - Constant delay, independent of length
  - Logarithmic delay with length (exponential driver tree)
  - Linear delay (speed of light/optimal repeaters)

Equal cost in k-ary n-cubes

- Equal number of nodes?
- Equal number of pins/wires?
- Equal bisection bandwidth?
- Equal area?
- Equal wire length?

What do we know?

- switch degree: \( n \) diameter = \( n(k-1) \)
- total links = \( Nn \)
- pins per node = \( 2wn \)
- bisection = \( k^{n-1} = \frac{N}{k} \) links in each directions
- \( 2Nw/k \) wires cross the middle

Latency for Equal Width Channels

- \( \text{total links}(N) = Nn \)

Latency with Equal Pin Count

- Baseline \( n=2 \), has \( w = 32 \) (128 wires per node)
- fix 2nw pins \( \Rightarrow w(n) = \frac{64}{n} \)
- distance up with \( n \), but channel time down
Latency with Equal Bisection Width

- N-node hypercube has $N$ bisection links
- 2d torus has $2N^{1/2}$
- Fixed bisection $\Rightarrow w(n) = N^{1/n}/2 = k/2$
- 1 M nodes, $n=2$ has $w=512$!

Larger Routing Delay (w/ equal pin)

- Dally’s conclusions strongly influenced by assumption of small routing delay
  - Here, Routing delay $\Delta=20$

Saturation

- Fatter links shorten queuing delays

Discuss of paper: Virtual Channel Flow Control

- Basic Idea: Use of virtual channels to reduce contention
  - Provided a model of k-ary, n-flies
  - Also provided simulation
- Tradeoff: Better to split buffers into virtual channels
  - Example (constant total storage for 2-ary 8-fly):
When are virtual channels allocated?

- Two separate processes:
  - Virtual channel allocation
  - Switch/connection allocation
- Virtual Channel Allocation
  - Choose route and free output virtual channel
  - Really means: Source of link tracks channels at destination
- Switch Allocation
  - For incoming virtual channel, negotiate switch on outgoing pin

Reducing routing delay: Express Cubes

- Problem: Low-dimensional networks have high $k$
  - Consequence: may have to travel many hops in single dimension
  - Routing latency can dominate long-distance traffic patterns
- Solution: Provide one or more “express” links

- Like express trains, express elevators, etc
  - Delay linear with distance, lower constant
  - Closer to “speed of light” in medium
  - Lower power, since no router cost
- Another Idea: route with pass transistors through links

Summary

- Network Topologies:
  - Topology | Degree | Diameter | Ave Dist | Bisection | $D (D \text{ ave}) @ P=1024$
  - 1D Array | 2 | $N-1$ | $N / 3$ | 1 | huge
  - 1D Ring | 2 | $N/2$ | $N/4$ | 2 |
  - 2D Mesh | 4 | $2 (N^{1/2} - 1)$ | $2 / 3 N^{1/2}$ | $N^{1/2}$ | 63 (21)
  - 2D Torus | 4 | $N^{1/2}$ | $1 / 2 N^{1/2}$ | $2N^{1/2}$ | 32 (16)
  - k-ary n-cube | 2n | nk/2 | nk/4 | nk/4 | 15 (7.5) @n=3
  - Hypercube | $n = \log N$ | n/2 | N/2 | 10 (5)

- Fair metrics of comparison
  - Equal cost: area, bisection bandwidth, etc
- Routing Algorithms restrict set of routes within the topology
  - simple mechanism selects turn at each hop
  - arithmetic, selection, lookup
- Virtual Channels
  - Adds complexity to router
  - Can be used for performance
  - Can be used for deadlock avoidance