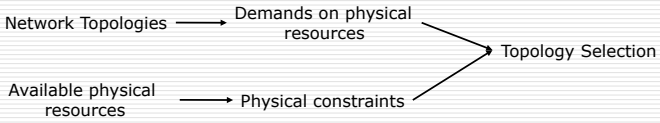
Analysis of Networks

Reading

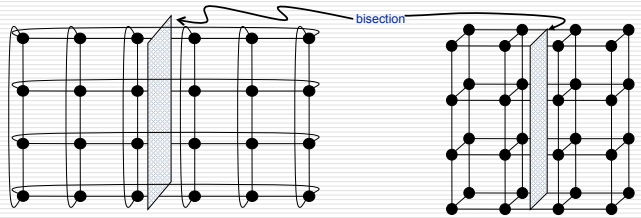
- Section 7.1
- Papers (see Tsquare)

Topology Optimization



- Network design is subject to physical constraints
  - Wiring area → bisection bandwidth
  - I/O → pin out
  - Speed → wire length
  - Power → wiring and buffers

Evaluation Metrics



- Bisection bandwidth
  - This is minimum bandwidth across any bisection of the network
- Bisection bandwidth is a limiting attribute of performance

Metrics

Network	Bisection Width	Node Size (I/O)
k-ary *n*-cube	$2Wk^{n-1}$	$2Wn$
Binary *n*-cube	$NW/2$	nW
n-dimensional mesh	Wk^{n-1}	$2Wn$
Omega Network	NW	$2tW$

A Few Observations

- For non-uniform k-ary n-cubes pick the bisection orthogonal to the largest radix

Bisection width = $\frac{2WN}{k_m}$

where k_m is the largest radix

Node Size

- Number of available I/Os
- What node size does the bisection width imply?
  - Normalize to that of a binary hypercube

$$Bisection\ Width = 2Wk^{n-1} = N$$

$$W = \frac{k}{2}$$

  - This is the maximum channel width that is permitted by the given bisection width and leads to a pin out of nk

Node Size (cont.)

- Channel width must satisfy two constraints

$$W \le \frac{PinOut}{2n}$$

$$W \le \frac{Bisection Width}{2 \times k^{n-1}}$$

Node Size (cont.)

- Effect of channel width on latency
  - Consider the following comparison between a $N = 2^n$ binary hypercube and $N = k^m$ k-ary m-cube (equal node size)

$$W_m = W_h \times \frac{n}{2} \times \frac{1}{m}$$

  - Keeping the number of nodes constant

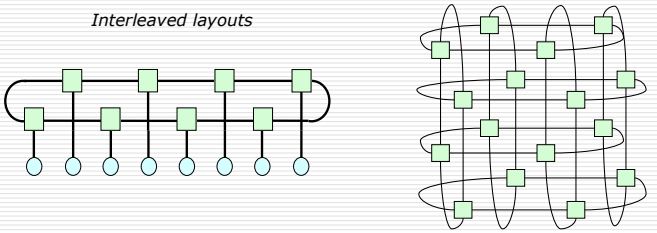
$$k = N^{\frac{1}{m}}$$

Linear increase in channel width ⇑ ⇓ *Exponential increase distance*

As m decreases

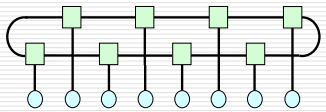
Wire Delay

Interleaved layouts



- Mapping higher dimensional networks to 2D and 3D implementations
  - Normalize to the wire length in 2D implementations
  - Layout arguments including/excluding wire width

Wire Delay (cont.)



- Length of the longest wire is greater by a factor of

$$k^{\frac{n-2}{2}}\left(\sqrt{k} \times k^{n-2}\right)$$

- Wire delay models
  - Linear
  - Logarithmic
  - constant

Basic Latency Model

- No load latency model

$$t_{wormhole} = [D \times (t_r + t_s + t_w)] + [\max(t_s, t_w) \times \left\lceil \frac{L}{W} \right\rceil]$$

  - Routing delays can be non-negligible
  - Assumes a router that uses both input buffering and output buffering
    - For an input buffered switch $\max(t_s, t_w) \rightarrow (t_s + t_w)$

Improved Latency Model

- The preceding model can be modified for
  - Wire length: normalize delays to 2D wire delay
  - Random traffic effects: can modify D

$$t_{wormhole} = [\underbrace{\frac{nk}{4}}_{\text{Uniform traffic, k even}} \times \underbrace{(r + s + k^{\frac{n-2}{2}})}_{\text{Normalize delays to that of 2D wire delay}}] + [\max(s, k^{\frac{n-2}{2}}) \times \left\lceil \frac{L}{W} \right\rceil]$$

Captures parameters of topology, physical constraints, router microarchitecture, & application

Application of the Latency Model

- Alternative delay models
- Latency expression can be viewed as the sum of a *distance* component and a *message* component
  - Behavior of each component differs for increasing dimensionality
  - Minimum occurs when both are equal
- Further generalizations
  - Routing and switching delays as a function of dimension
  - Delays due to the use of virtual channels

Analysis

- Constant bisection width
  - Assumes wiring density is the limiting resource
- Constant node size
  - Assumes that pin out is the limiting resource
- Constant wire throughput
  - Assumes wire length is the limiting resource

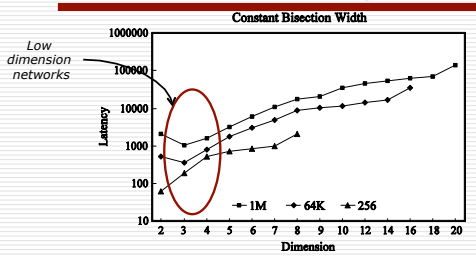
Constant Bisection Width

- Assume bisection width is fixed to that of a *k*-ary *n*-cube with k=2 and W = 1
  - Bisection width = N (note use of full duplex channels)
- For K>1, W = (k/2), latency for linear wire delay is reduced to

$$t_{wormhole} = \underbrace{[\frac{nk}{4} \times (r + s + k^{\frac{n-2}{2}})]}_{\text{Distance component}} + \underbrace{[\max(s, k^{\frac{n-2}{2}}) \times \left\lceil 2 \times \frac{L}{k} \right\rceil]}_{\text{Message component}}$$

$$W = \frac{k}{2}$$

Analysis: Constant BW



- As the number of dimensions increases
  - Channel width decreases
  - Larger network sensitive to switch and wire delays
  - Smaller networks sensitive to channel widths and message sizes

Analysis: Constant Node Size

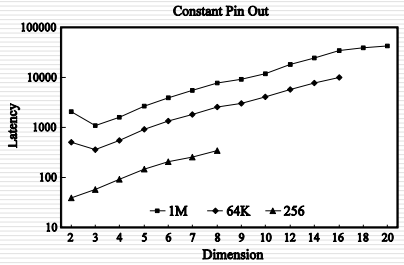
- # of pins is given by P = 2nW → W = P/2n
  - Implicitly assumes half-duplex channels!
- Latency is given by

$$t_{wormhole} = [\frac{nk}{4} \times (r + s + k^{\frac{n-2}{2}})] + [\max(s, k^{\frac{n-2}{2}}) \times \lceil \frac{2nL}{P} \rceil]$$

- Note: bisection width and node size are not independent!

$$BW = \frac{P}{n} \times k^{n-1}$$

Analysis: Constant Node Size



- Linear increase in channel width generally overshadowed by exponential increase in distance
- Note that latency expressions change depending on how channel resources are used

Analysis: Constant Wire Throughput

- Wire delays rather than switching delays dominate
- Pipeline wires in the same manner that datapaths are pipelined
- Under this model the latency expression can be written as

$$t_{wormhole} = [\frac{nk}{4} \times (r + s + w_{avg})] + [s \times \lceil \frac{L}{W} \rceil]$$

  - Message latency component is only dependent upon switching speed: rate at which bits can be placed on the wire

Average Wire Length

- Average wire length in a dimension

$$l_2 \frac{(k-1)}{k}$$

- With (n/2) dimensions mapped to each physical dimension, average wire length in a physical dimension grows by a factor of *k* and is

$$l_2 \frac{(k-1)}{k} + l_2 \frac{(k-1)}{k} k + l_2 \frac{(k-1)}{k} k^2 \cdots$$

- Average across all dimensions we have

$$w_{avg} = 2l_2 \frac{(N^{\frac{n}{2}} - 1)}{nk}$$

Summary

- Latency can be written as

$$Latency = \alpha.Dist(n) + \beta.MsgLength(n)$$

- Minimum latency achieved when components are equal
- Rate of growth of components determined by network parameters
- Congestion can be modeled with the addition of a third term
- **The general case was made for low radix routers**

The Case for High Radix Routers

- Let us study the argument for high radix routers
- Packet Latency is given by

$$T = Ht_r + L/b$$

channel bandwidth

per router hop delay

- For a non-blocking network this can be given as

$$T = 2t_r \log_k N + 2kL/B$$

total router bandwidth

Required of a non-blocking network

Note $b = \frac{B}{2k}$

John Kim, William J. Dally, Brian Towles, Amit K. Gupta, "Microarchitecture of a High-Radix Router," *Proceedings of the International Symposium on Computer Architecture*, June 2005

The Case for High Radix Routers (cont.)

- Find the value of k to minimize the preceding latency expression

$$k \log^2 k = \frac{Bt_r \log N}{L}$$

aspect ratio

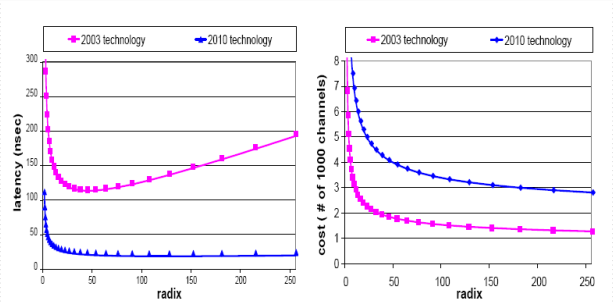
- Analyze dependencies on the radix
  - Keep the switch cycle time constant while #pipeline stages grows

$$t_r = t_{cy} P = t_{cy}(X + Y \log_2 k)$$

Observations/Assumptions

- Since total router bandwidth is fixed, k is assumed to independent of B
- The router delay model
  - As router delays increases (logk) this is offset by a distance reduction (1/logk) in the network
  - Net effect is that the optimal radix does not change
- Note the independence of header delay from k
  - The latency is based on Clos and fat tree style topologies

Optimal Radix



- When does serialization latency begin to dominate?
- Note that cost is roughly proportional to bandwidth (pins)

Reproduced from, John Kim, William J. Dally, Brian Towles1, Amit K. Gupta, "Microarchitecture of a High-Radix Router," *Proceedings of the International Symposium on Computer Architecture*, June 2005

Some Comments

- Target is indirect networks where distance vs. radix properties are different from direct networks
- Targeted towards very large scale networks
  - Commercial implications of indirect networks
- Energy behaviors at scale have quite an influence

Summary

- Simple models of resource demand can inform decisions about interconnect
  - Topology, channel width, router complexity
- The models must track technology
  - ITRS roadmap
- Now onto to some case studies……