Multiprocessor Interconnection Networks

CS 418
Lecture 26

(CSG Sections 1.1-1.10)

Topics
- Network design issues
- Network Topology

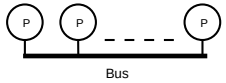
Networks

- How do we move data between processors?
- Design Options:
  - Topology
  - Routing
  - Physical implementation

Evaluation Criteria:

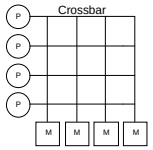
- Latency
- Bisection Bandwidth
- Contention and hot-spot behavior
- Partitionability
- Cost and scalability
- Fault tolerance

Buses



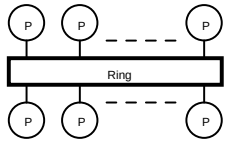
- Simple and cost-effective for small-scale multiprocessors
- Not scalable (limited bandwidth; electrical complications)

Crossbars



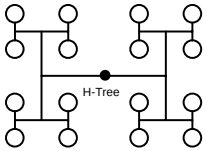
- Each port has link to every other port
+ Low latency and high throughput
- Cost grows as O(N^2) so not very scalable.
- Difficult to arbitrate and to get all data lines into and out of a centralized crossbar.
- Used in small-scale MPs (e.g., C.mmp) and as building block for other networks (e.g., Omega).

Rings

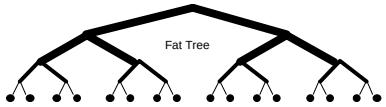


- Cheap: Cost is O(N).
- Point-to-point wires and pipelining can be used to make them very fast.
+ High overall bandwidth
- High latency O(N)
- Examples: KSR machine, Hector

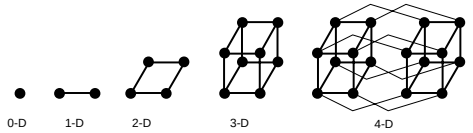
Trees



- Cheap: Cost is O(N).
- Latency is O(logN).
- Easy to layout as planar graphs (e.g., H-Trees).
- For random permutations, root can become bottleneck.
- To avoid root being bottleneck, notion of Fat-Trees (used in CM-5)



Hypercubes



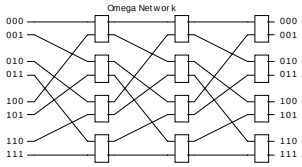
- Also called binary n-cubes. # of nodes = N = 2^n.
- Latency is O(logN); Out degree of PE is O(logN)
- Minimizes hops; good bisection BW; but tough to layout in 3-space
- Popular in early message-passing computers (e.g., intel iPSC, NCUBE)
- Used as direct network ==> emphasizes locality

Multistage Logarithmic Networks

Key Idea: have multiple layers of switches between destinations.

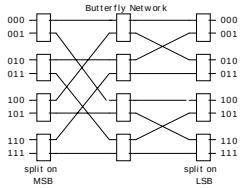
- Cost is O(NlogN); latency is O(logN); throughput is O(N).
- Generally indirect networks.
- Many variations exist (Omega, Butterfly, Benes, ...).
- Used in many machines: BBN Butterfly, IBM RP3, ...

Omega Network



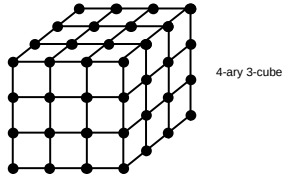
- All stages are same, so can use recirculating network.
- Single path from source to destination.
- Can add extra stages and pathways to minimize collisions and increase fault tolerance.
- Can support combining. Used in IBM RP3.

Butterfly Network



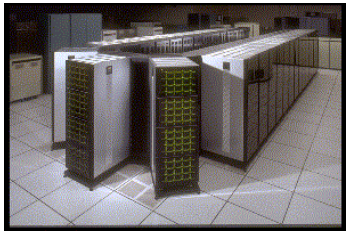
- Equivalent to Omega network. Easy to see routing of messages.
- Also very similar to hypercubes (direct vs. indirect though).
- Clearly see that bisection of network is (N / 2) channels.
- Can use higher-degree switches to reduce depth.

k-ary n-cubes



- Generalization of hypercubes (k-nodes in a string)
- Total # of nodes = N = k^n.
- k > 2 reduces # of channels at bisection, thus allowing for wider channels but more hops.

Real World 2D mesh



1824 node Paragon: 16 x 114 array

Advantages of Low-Dimensional Nets

What can be built in VLSI is often wire-limited

LDNs are easier to layout:

- more uniform wiring density (easier to embed in 2-D or 3-D space)
- mostly local connections (e.g., grids)

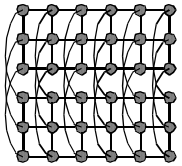
Compared with HDNs (e.g., hypercubes), LDNs have:

- shorter wires (reduces hop latency)
- fewer wires (increases bandwidth given constant bisection width)
  - increased channel width is the major reason why LDNs win!

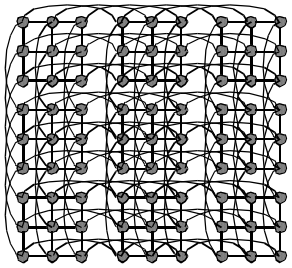
LDNs have better hot-spot throughput

- more pins per node than HDNs

Embedding into 2 Dimensions



6 x 3 x 2



Embed multiple logical dimension in one physical dimension using long wires