Parallel and Distributed Computing
CS3006 (BDS-6A)
Lecture 05

Instructor: Dr. Syed Mohammad Irteza

Assistant Professor, Department of Computer Science, FAST

14 February, 2023

Previous Lecture

- Flynn's Taxonomy:
  - MIMD
- Shared Memory Interconnection Networks
  - Central data bus
  - Crossbar
  - Multi-stage Omega network
- Advantages and disadvantages of shared memory machines
- Distributed Memory (SIMD, MIMD)

Assigned reading pointers:

- Cache Coherence:
  - When we are in a distributed environment, each CPU’s cache needs to be consistent (continuously needs to be updated for current values), which is known as cache coherence.
- Snooping:
  - Snoopy protocols achieve data consistency between the cache memory and the shared memory through a bus-based memory system. Write-invalidate and write-update policies are used for maintaining cache consistency.
- Branch Prediction:
  - Branch prediction is a technique used in CPU design that attempts to guess the outcome of a conditional operation and prepare for the most likely result.

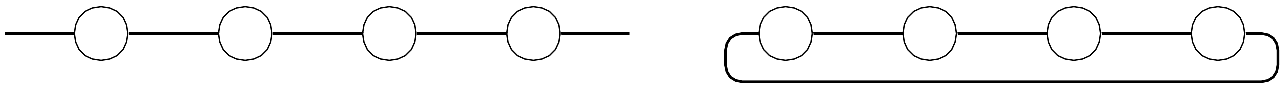
DM-MIMD Routing Mechanism

- Routing mechanism determines the path a message takes through network to reach from source to destination node.
- Data and task decomposition have to be dealt with explicitly!
- The topology and speed of the data paths are crucial and have to be balanced with costs.
- Routing can be classified as:
  - Minimal
  - Non-minimal
- It can also be classified as:
  - Deterministic routing
  - Adaptive routing

Network Topologies

Linear Arrays

- In a linear array, each node has two neighbors, one to its left and one to its right.
- If the nodes at either end are connected, we refer to it as a 1-D torus or a ring.



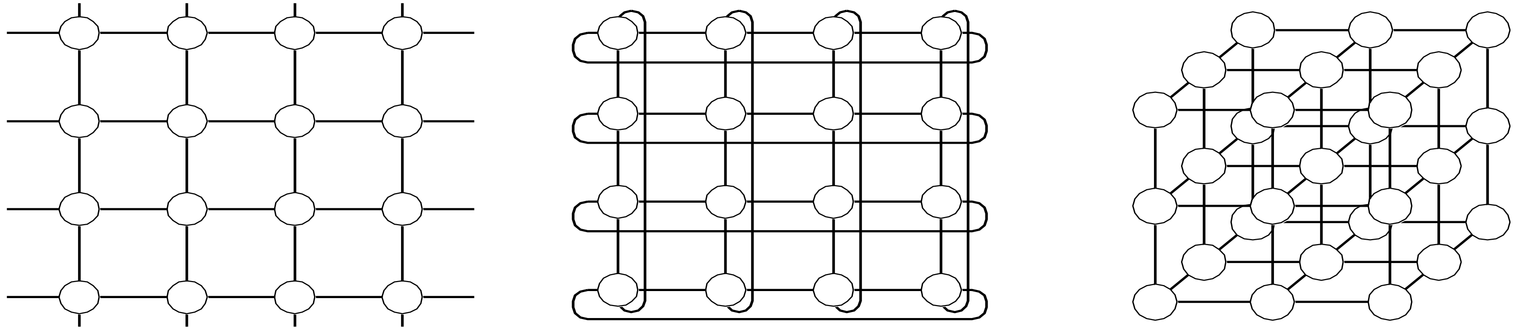
(a) (b)

Linear arrays: (a) with no wraparound links; (b) with wraparound link.

Network Topologies

K-d Meshes

- A generalization has nodes with 4 neighbors, to the north, south, east, and west.
- A further generalization to *d* dimensions has nodes with *2d* neighbors (i.e., 6 neighbors in case of 3d cube).



(a) (b) (c)

Two and three dimensional meshes: (a) 2-D mesh with no wraparound; (b) 2-D mesh with wraparound link (2-D torus); and (c) a 3-D mesh with no wraparound.

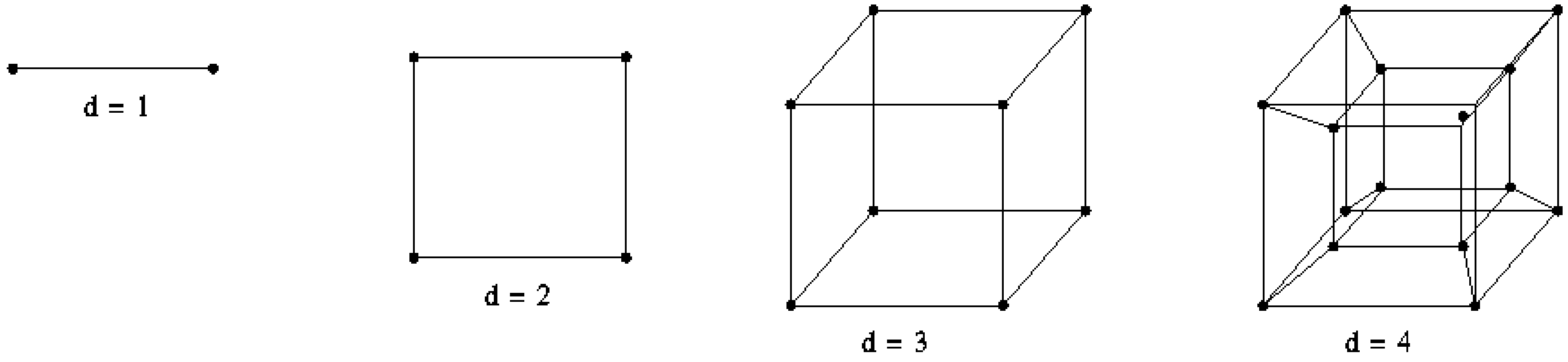
Network Topologies

Hypercube

- For a hypercube with 2^d nodes the number of steps to be taken between any two nodes is at most d (logarithmic grow)

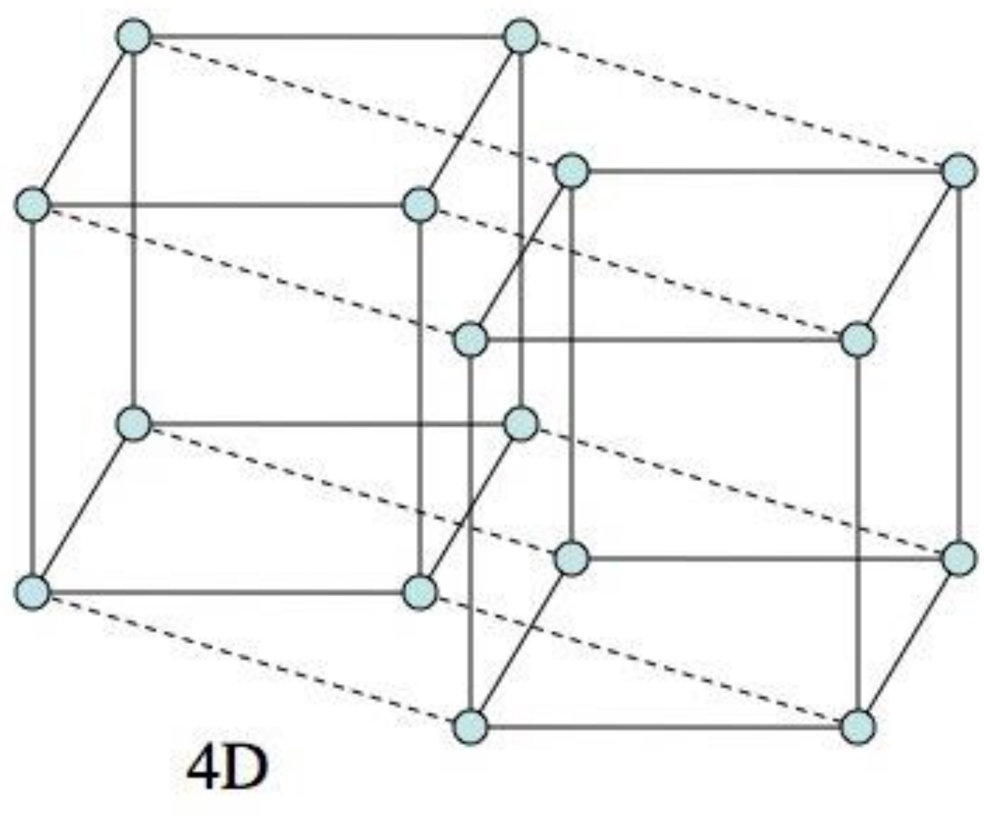
$$d = \log p \ (\text{dimensions} = \log(\text{nodes}))$$

- The distance between any two nodes is at most *log p*.
- Each node has *log p* neighbors.



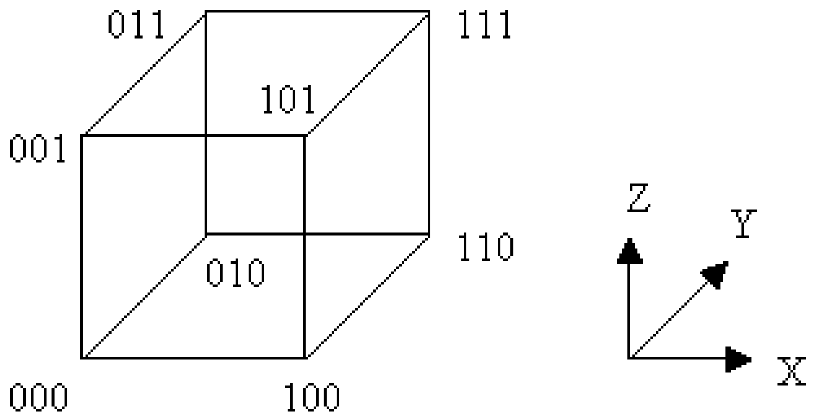
Network Topologies: hypercube

- Rule of thumb is: “d-dimensional hypercube can be constructed by connecting corresponding nodes of two (d-1)-dimensional hypercubes”



Network Topologies: hypercube

- The processors are numbered with 3-bit binary numbers which represent the X-Y-Z coordinates
- The distance between two nodes is given by the number of bit positions at which the two nodes differ.

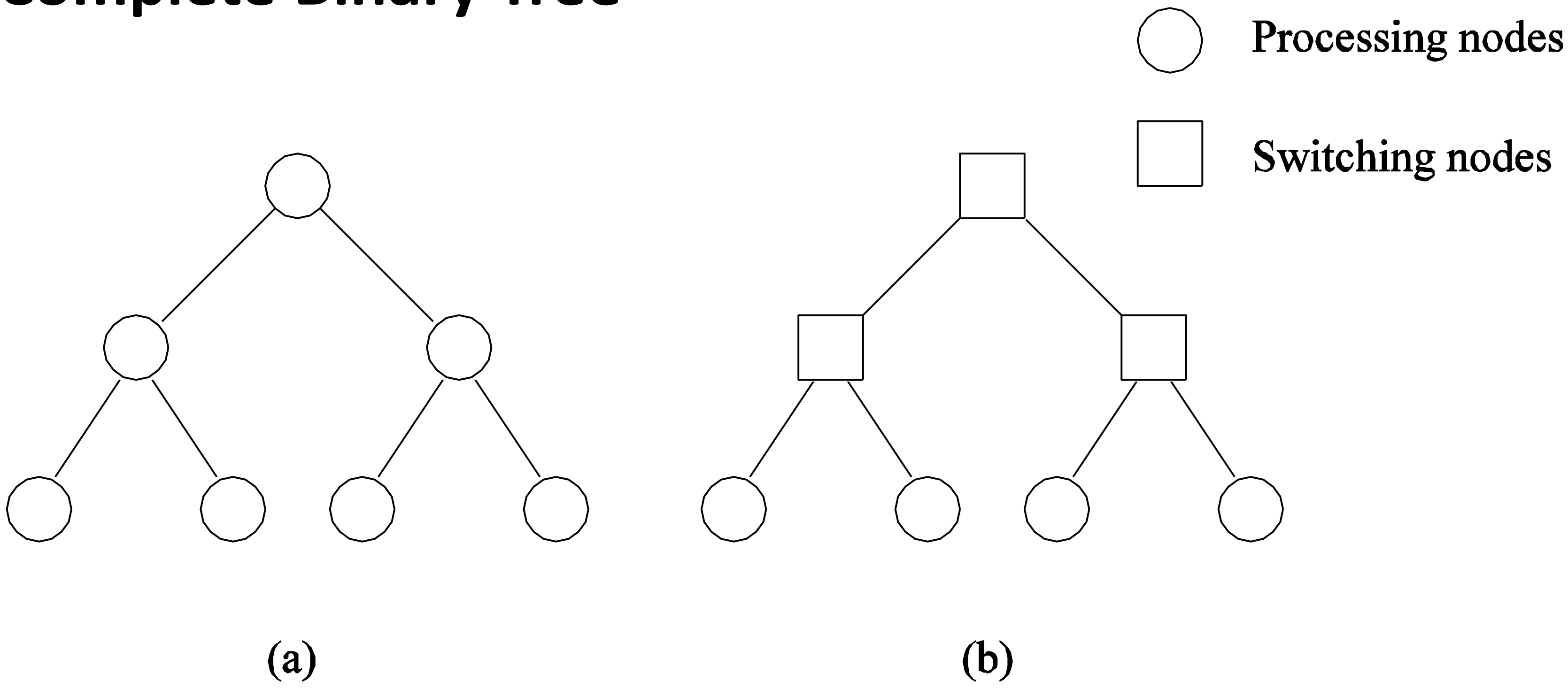


Network Topologies: Tree based Networks

- A tree network is one in which there is one path between any pair of nodes
- Linear arrays and star-connected networks are special cases of tree-based networks
- In static tree network, each node represent a processing element
- In dynamic tree network, leaf nodes represent processing element while internal nodes are switching elements.
- The source node sends the message up the tree until it reaches the node at the root of the smallest subtree containing both the source and destination nodes.
- Trees can be laid out in 2D with no wire crossings. This is an attractive property of trees.
- The distance between any two nodes is no more than $(2 * \log_2 p)$

Network Topologies: Tree based Networks

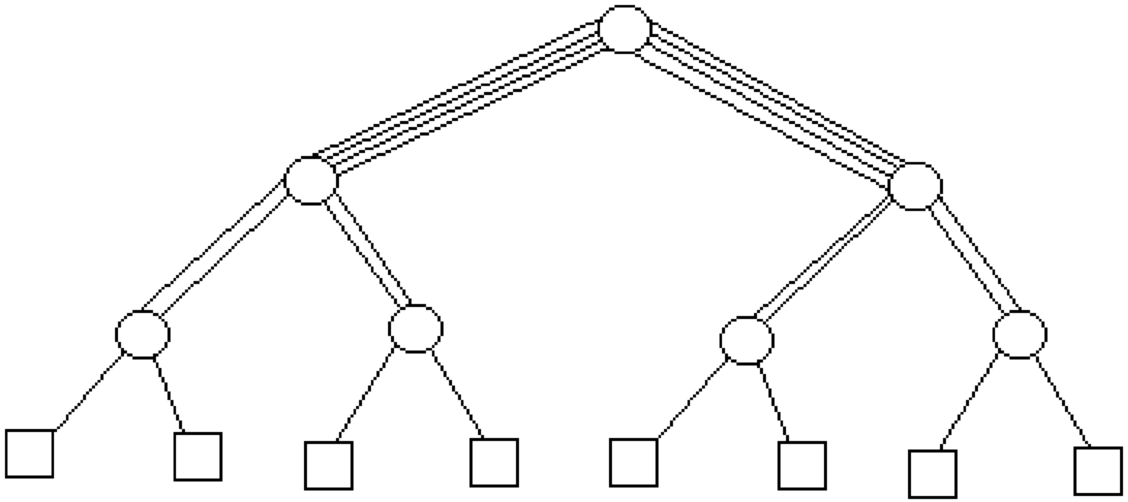
Complete Binary Tree



Complete binary tree networks: (a) a static tree network; and (b) a dynamic tree network.

Routing - Fat Tree

- Another topology is the “fat tree”
- In a tree, a node can speak to another node passing through the root so we have higher traffic at the root node.
- Fat tree amends this by providing more bandwidth (with multiple connections) in the higher levels of the tree
- N-ary fat tree is when the levels towards the root have N times the number of connections in the level below it



Here, leaf nodes are processing nodes, and all intermediate nodes are switches

Evaluating Static Interconnections

The parameters to evaluate a static interconnection:-

- **Cost:** Usually depends on number of links for communication. E.g., cost for linear array is p-1.
  - Lower values are favorable
- **Diameter:** The shortest distance between the farthest two nodes in the network. The diameter of a linear array is p − 1.
  - Lower values are favorable
- **Bisection Width:** The minimum number of wires (i.e., links) you must cut to divide the network into two equal parts. The bisection width of a linear array is 1.
  - What does this tell us about the performance of a topology?

Evaluating Static Interconnections

- **Arc-connectivity:** The minimum number of arcs or links that must be removed from the network, to break the network into two disconnected networks
  - Higher values are desirable
  - It is the minimum number of the links that must be cut to separate the single node from the network
  - Higher values means, that incase of link failure there are multiple other routes to the node.
  - Arc-connectivity of linear array is 1 and 2 for ring.

Evaluating Static Interconnections

Network	Diameter	Bisection Width	Arc Connectivity	Cost (No. of links)
Completely-connected	1	$p^2/4$	$p-1$	$p(p-1)/2$
Star	2	1	1	$p-1$
Complete binary tree	$2\log((p+1)/2)$	1	1	$p-1$
Linear array	$p-1$	1	1	$p-1$
2-D mesh, no wraparound	$2(\sqrt{p}-1)$	$\sqrt{p}$	2	$2(p-\sqrt{p})$
2-D wraparound mesh	$2\lfloor\sqrt{p}/2\rfloor$	$2\sqrt{p}$	4	$2p$
Hypercube	$\log p$	$p/2$	$\log p$	$(p\log p)/2$

Sources

- Slides of Dr. Rana Asif Rahman & Dr. Haroon Mahmood, FAST
- (Chapter 2) Kumar, V., Grama, A., Gupta, A., & Karypis, G. (1994). Introduction to parallel computing (Vol. 110). Redwood City, CA: Benjamin/Cummings.
- Quinn, M. J. Parallel Programming in C with MPI and OpenMP,(2003).