Parallel and Distributed Computing

CS3006

Lecture 4

Network Topologies

14th February 2024

Dr. Rana Asif Rehman

Agenda

- A Quick Review
- Static Interconnection vs Dynamic interconnections
- Some Basic Interconnections
- Evaluating Static Interconnections

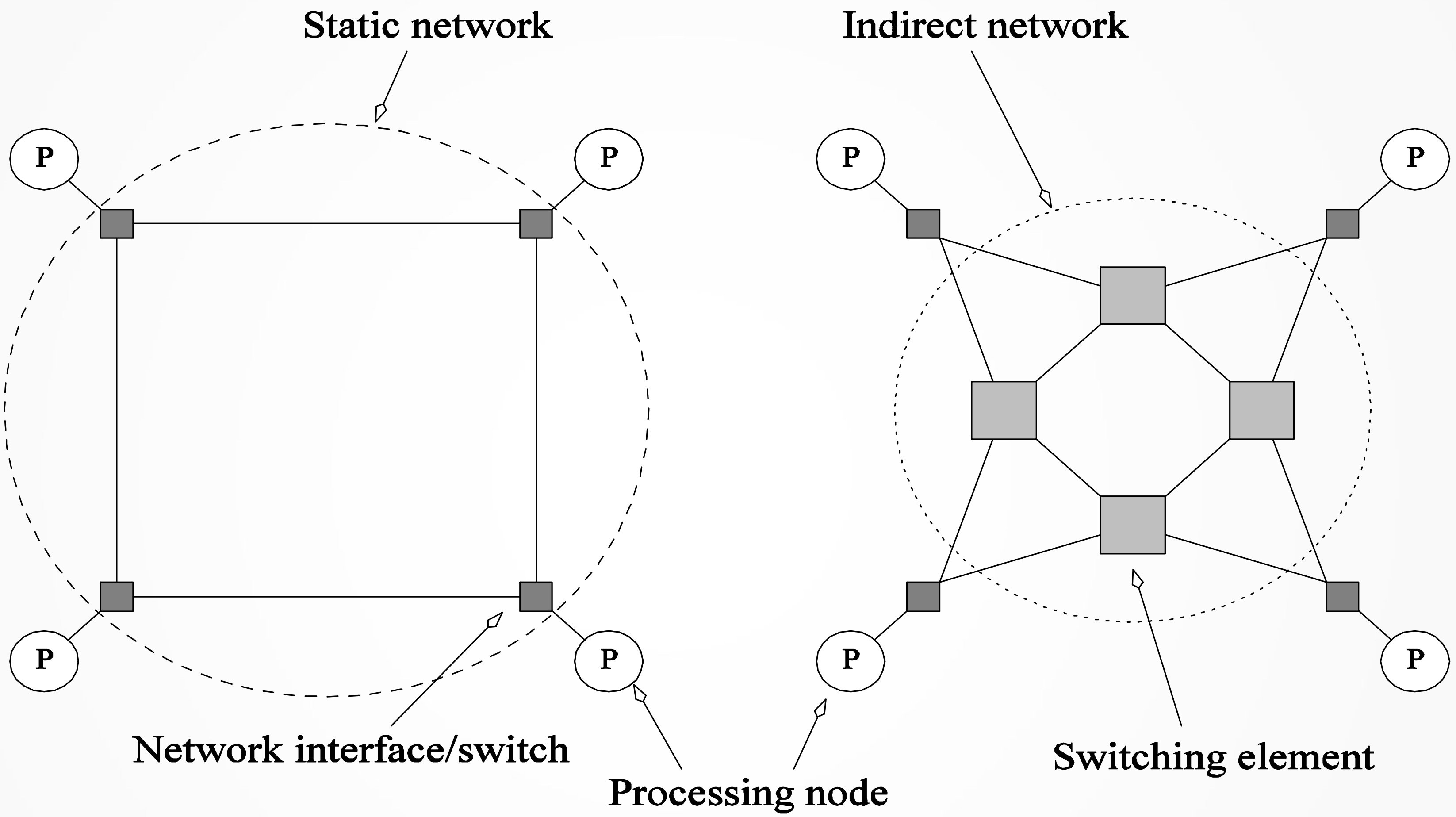
Quick Review to the Previous Lecture

- **Flynn's Taxonomy**
  - SISD
  - MISD
  - SIMD
  - MIMD
- **PRAM Model**
  - Types
  - Arbitration protocols
- **Routing techniques and Costs**

Static vs Dynamic Interconnections

- Interconnection networks carry data between processors and to memory.
- Interconnects are made of processing elements, switches and links (wires, fiber).
- Interconnects are classified as static or dynamic.
- **Static** networks consist of point-to-point communication links among processing nodes and are also referred to as *direct* networks.
- **Dynamic** networks are built using switches and communication links. Dynamic networks are also referred to as *indirect* networks.

Static vs Dynamic Interconnections



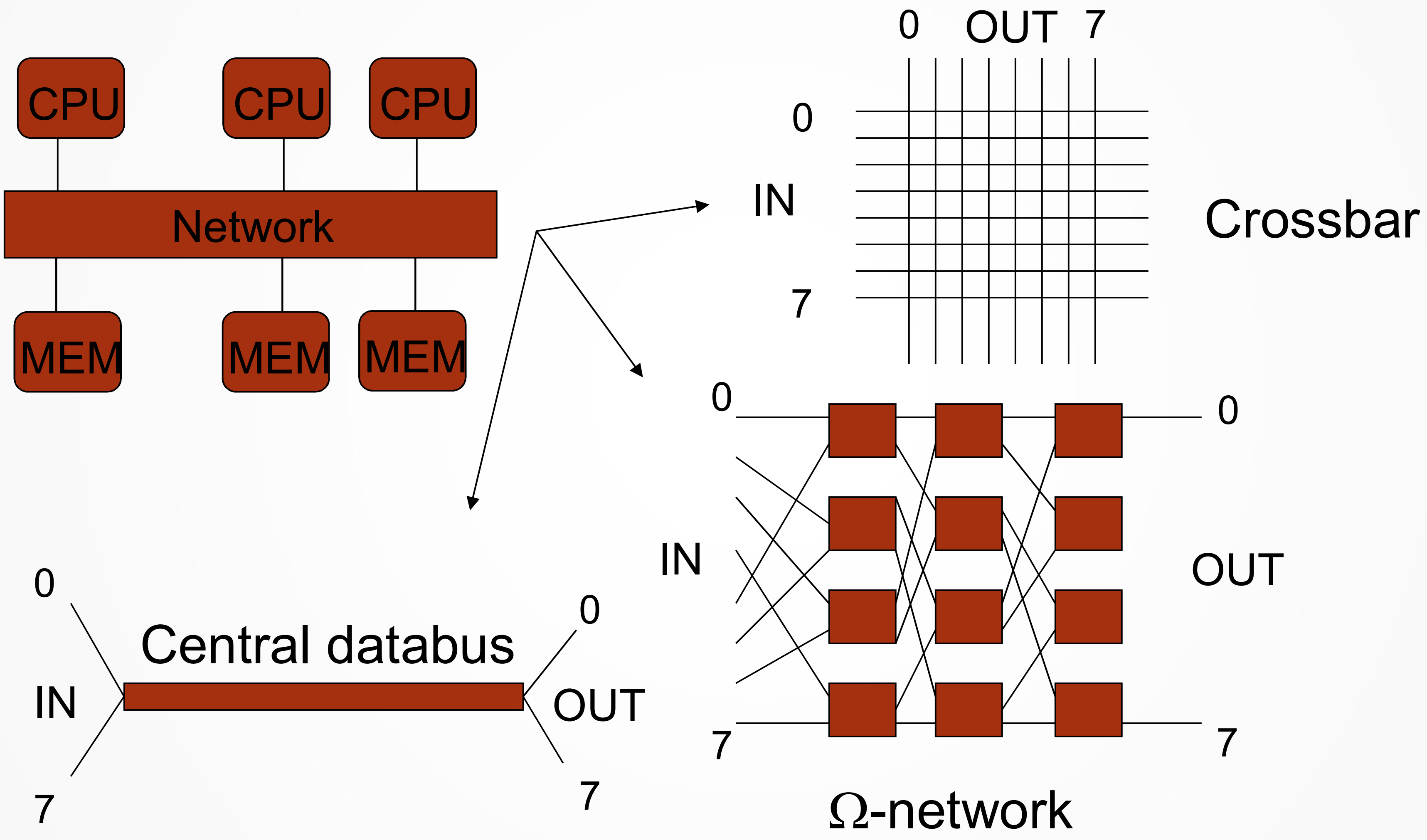
Classification of interconnection networks: (a) a static network; and (b) a dynamic network.

Interconnection Networks

- Main problem is how to do interconnections of the CPUs to each other and to the memory

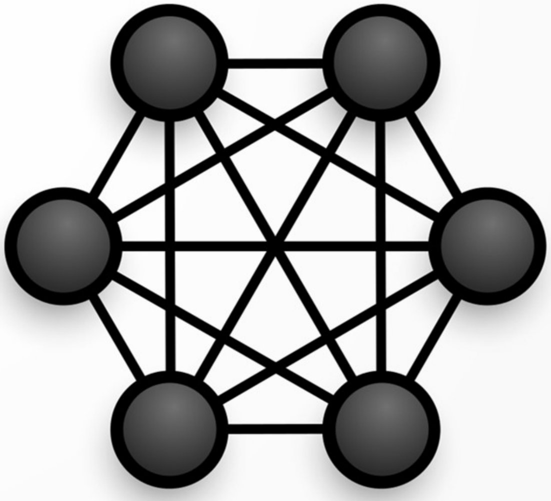
- There are three main dynamic network topologies available:
  - Crossbar (n^2 connections – data path without sharing)
  - Multi-stages network ($n \log_2 n$ connections - $\log_2 n$ switching stages and shared on a path)
  - Central databus (1 connections - n shared)

Dynamic Interconnection Networks



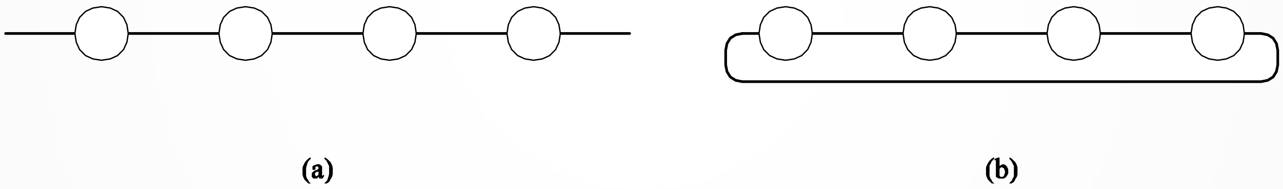
Static Network Topologies: Linear Arrays, Meshes, and *k-d* Meshes

- Each processor is connected to every other processor (Complete connected network).
- The number of links in the network scales as $O(p^2)$.
- While the performance scales very well, the hardware complexity is not realizable for large values of *p*.
- Star connected networks



Static Network Topologies: Linear Arrays, Meshes, and *k-d* Meshes

- 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.

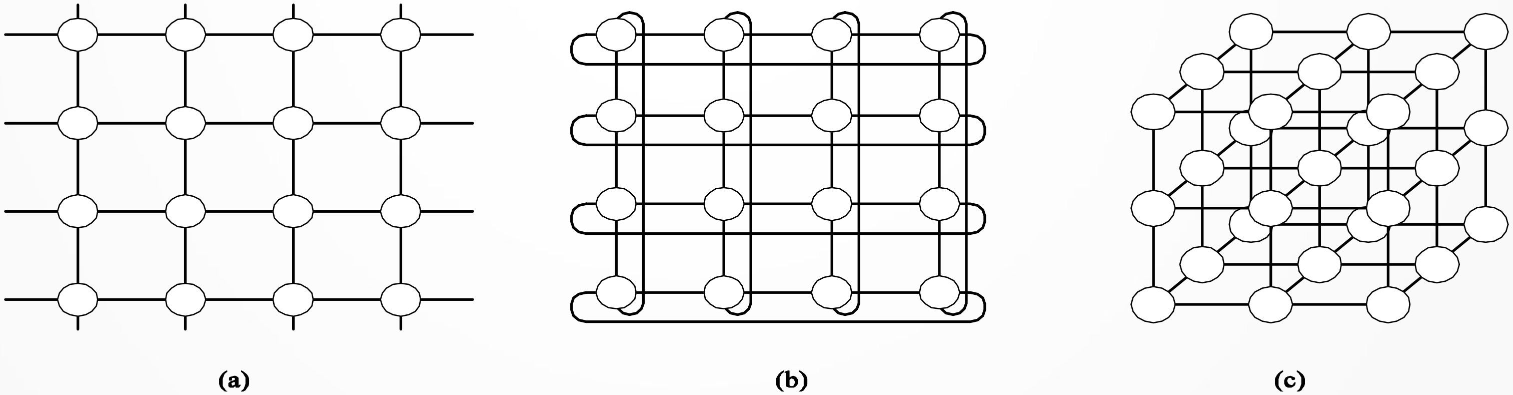


Linear arrays: (a) with no wraparound links; (b) with wraparound link (1-D torus)

Static Network Topologies: Linear Arrays, Meshes, and *k-d* Meshes

Mesh

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



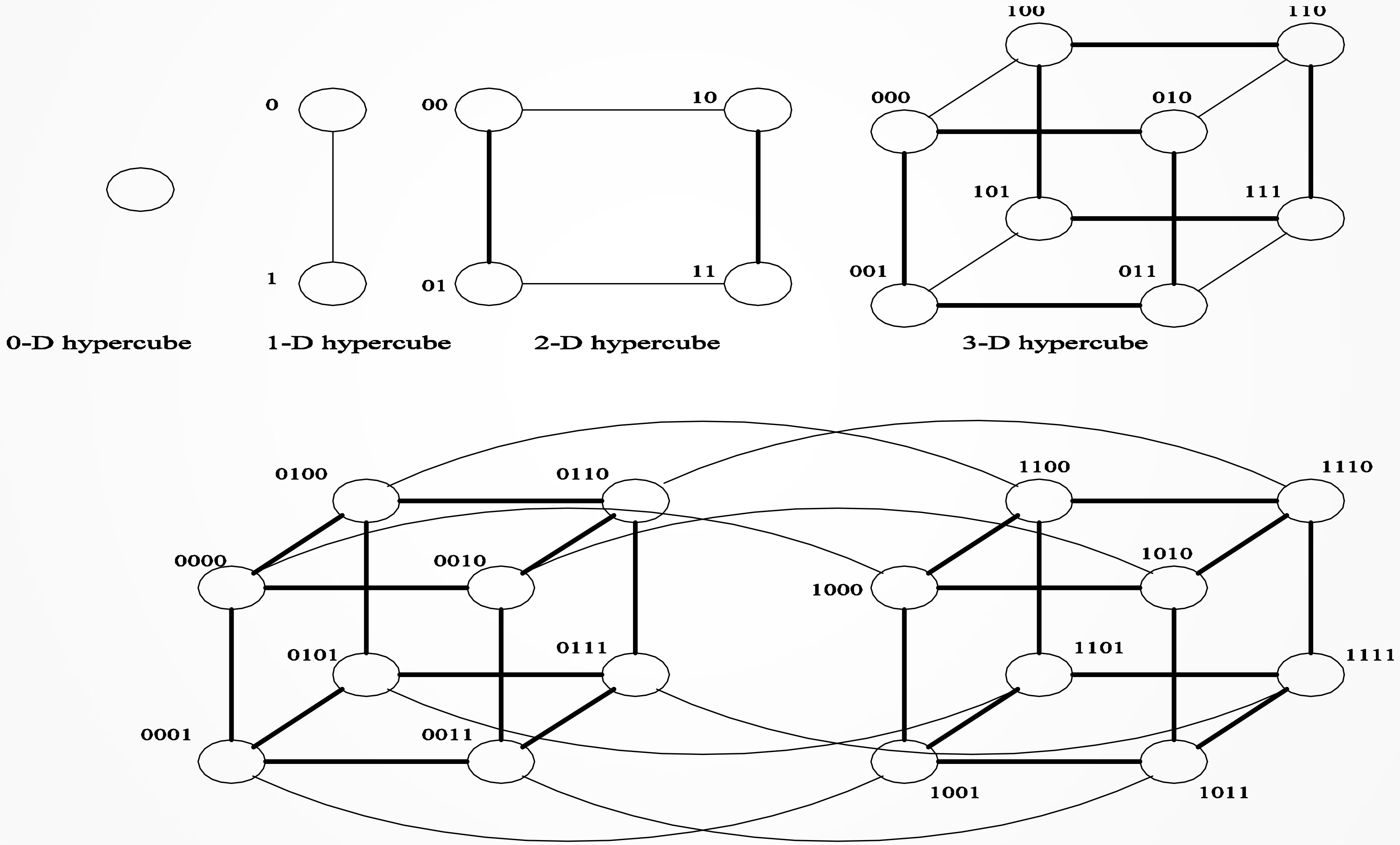
Two and three dimensional meshes: (a) 2-D mesh with no wraparound; (b) 2-D mesh with wraparound link (2-D torus); and (c) a 3-D mesh with no wraparound.

Static Network Topologies: Linear Arrays, Meshes, and *k-d* Meshes

Hypercube

- The hypercube has two nodes along each dimension except *0d* hypercube.
- *d = log p (dimensions = log(nodes))*
- The distance between any two nodes is at most *log p*.
- Each node has *log p* neighbors.
- The distance between two nodes is given by the number of bit positions at which the two nodes differ.
- Rule of thumb is: "d-dimensional hypercube can be constructed by connecting corresponding nodes of two (d-1)-dimensional hypercubes"

Static Network Topologies: Linear Arrays, Meshes, and *k-d* Meshes



0-D hypercube

1-D hypercube

2-D hypercube

3-D hypercube

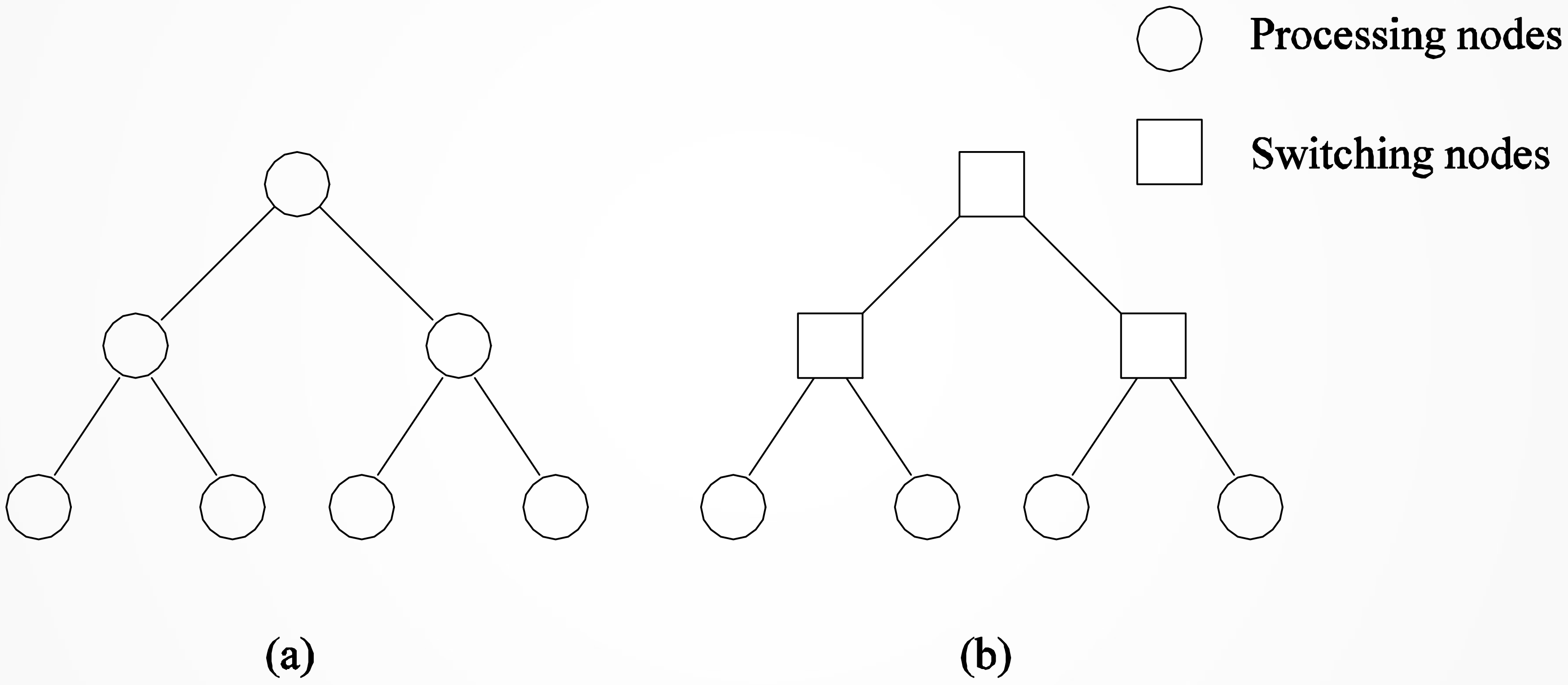
4-D hypercube

Static 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.

Static Network Topologies: Tree based Networks

Complete Binary Tree



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

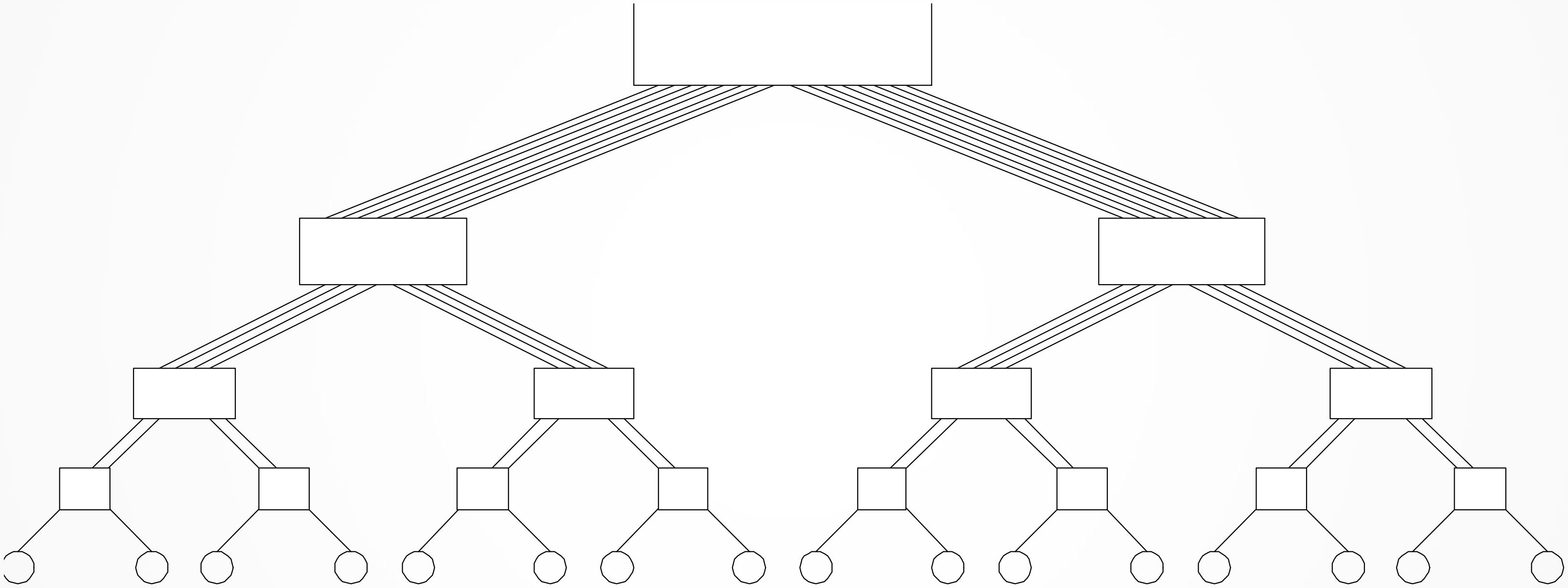
Static Network Topologies: Tree based Networks

Properties of Complete Binary Tree Network

- The distance between any two nodes is no more than *2logp*.
- Links higher up the tree potentially carry more traffic than those at the lower levels.
- For this reason, a variant called a fat-tree, fattens the links as we go up the tree.
- Trees can be laid out in 2D with no wire crossings. This is an attractive property of trees.

Static Network Topologies: Tree based Networks

Properties of Complete Binary Tree Network



A fat tree network of 16 processing nodes.

Evaluating Static Interconnections

The parameters to evaluate a static interconnection:-

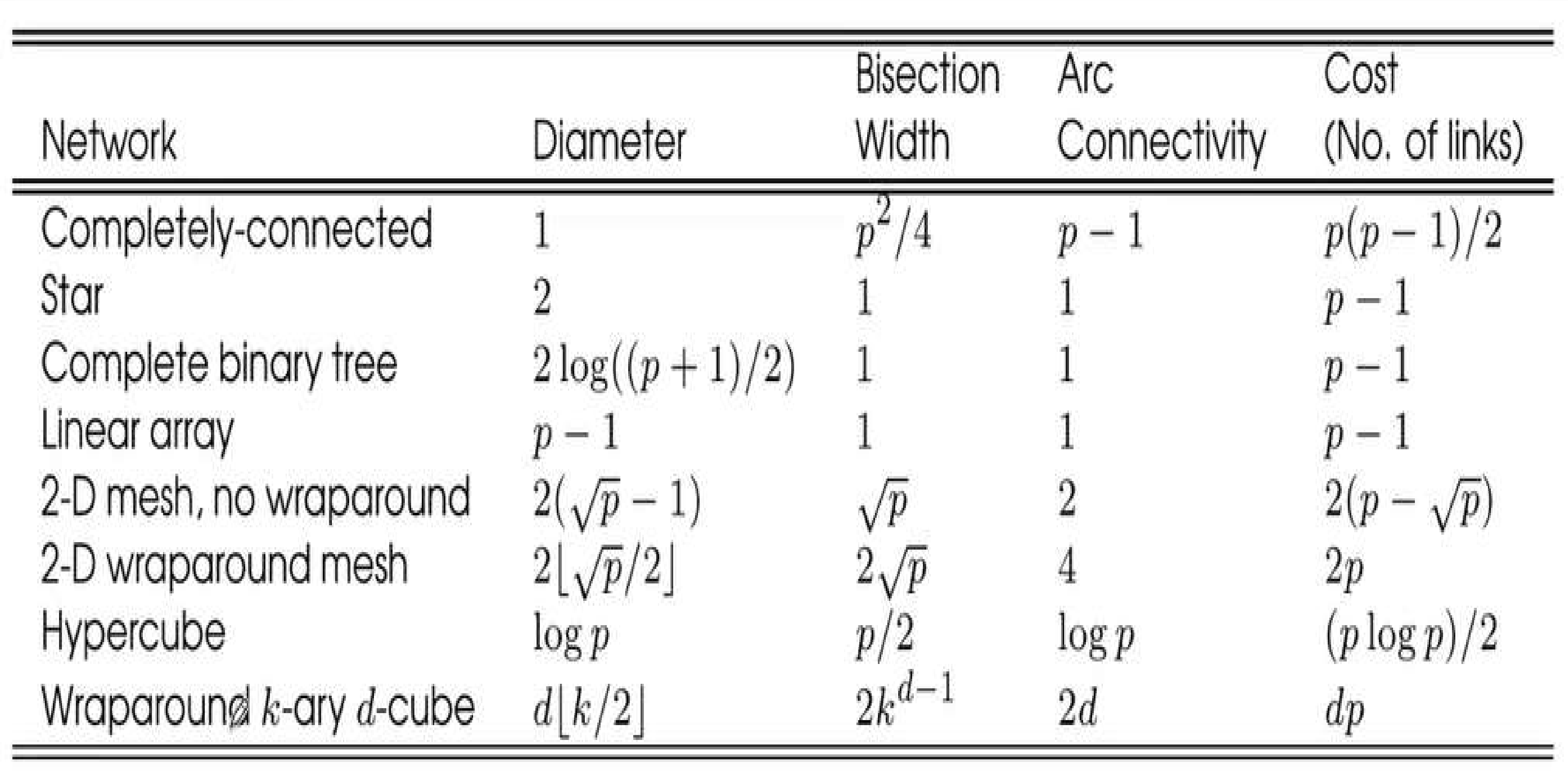
- ***Cost:*** Usually depends on number of links for communication. E.g., cost for linear array is *p-1*.
  - *Lower values are favorable*
- ***Diameter:*** The shortest distance between the farthest two nodes in the network. The diameter of a linear array is $p - 1$.
  - *Lower values are favorable*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

Evaluating Static Interconnections

The parameters to evaluate a static interconnection:-

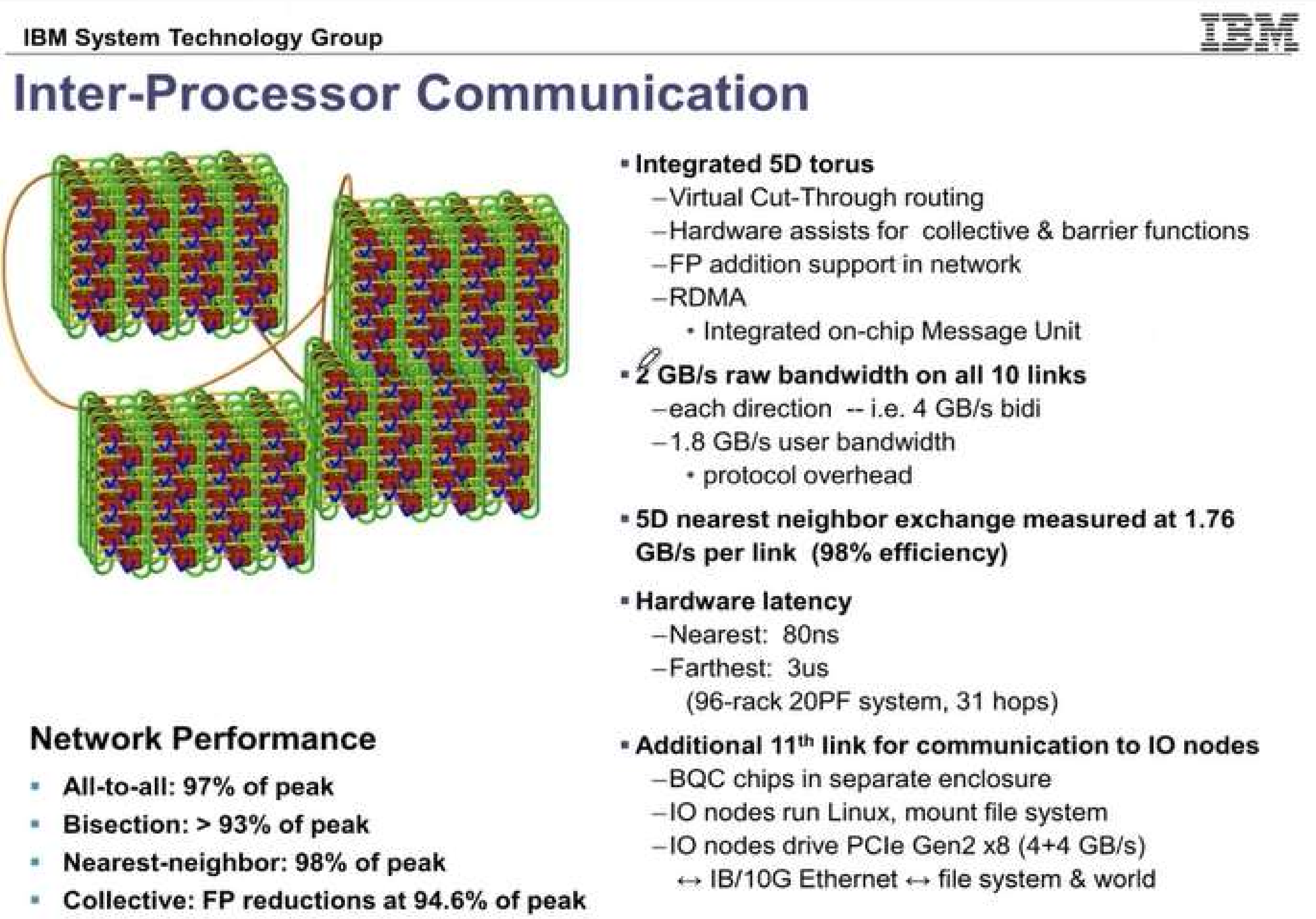
- ***Arc-connectivity:*** *The minimum number of arcs or links that must be removed from the network, to break the network into two disconnected networks*
  - *Higher value are desirable*
  - It is minimum number of the links that must be cut to separate the single node from the network
  - Higher values 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	$\log p$	$p/2$	$\log p$	$(p\log p)/2$
Wraparound k-ary d-cube	$d\lfloor k/2\rfloor$	$2k^{d-1}$	$2d$	dp

Real World Example:



IBM System Technology Group

IBM

Inter-Processor Communication

- **Integrated 5D torus**
  - Virtual Cut-Through routing
  - Hardware assists for collective & barrier functions
  - FP addition support in network
  - RDMA
    - Integrated on-chip Message Unit
- **2 GB/s raw bandwidth on all 10 links**
  - each direction -- i.e. 4 GB/s bidi
  - 1.8 GB/s user bandwidth
    - protocol overhead
- **5D nearest neighbor exchange measured at 1.76 GB/s per link (98% efficiency)**
- **Hardware latency**
  - Nearest: 80ns
  - Farthest: 3us
    (96-rack 20PF system, 31 hops)
- **Additional 11th link for communication to IO nodes**
  - BQC chips in separate enclosure
  - IO nodes run Linux, mount file system
  - IO nodes drive PCIe Gen2 x8 (4+4 GB/s)
    ↔ IB/10G Ethernet ↔ file system & world

Network Performance

- **All-to-all: 97% of peak**
- **Bisection: > 93% of peak**
- **Nearest-neighbor: 98% of peak**
- **Collective: FP reductions at 94.6% of peak**

Questions

References

1. Flynn, M., “Some Computer Organizations and Their Effectiveness,” IEEE Transactions on Computers, Vol. C-21, No. 9, September 1972.
2. Kumar, V., Grama, A., Gupta, A., & Karypis, G. (1994). *Introduction to parallel computing* (Vol. 110). Redwood City, CA: Benjamin/Cummings.
3. Quinn, M. J. Parallel Programming in C with MPI and OpenMP,(2003).

Cache Coherence and snooping

- In a snooping system, all caches on the bus monitor (or snoop) the bus to determine if they have a copy of the block of data that is requested on the bus.
- Every cache has a copy of the sharing status of every block of physical memory it has.

Snooping Protocol Types

- Write-invalidate (mostly used)
  - The processor that is writing data causes copies in the caches of all other processors in the system to be rendered **invalid** before it changes its local copy.
- Write-update
  - The processor that is writing the data broadcasts the new data over the bus
  - All caches that contain copies of the data are then updated