Interconnection Topologies

BITS Pilani
Pilani Campus

K Hari Babu
Department of Computer Science & Information Systems

Interconnection Networks

- All computers with multiple processors must provide a way for processors to interact
  - In some systems processors use the interconnection network to access a shared memory.
  - In other systems processors use the interconnection network to send messages to each other
- Two principal types of interconnection media
  - Shared medium
  - Switched interconnection media

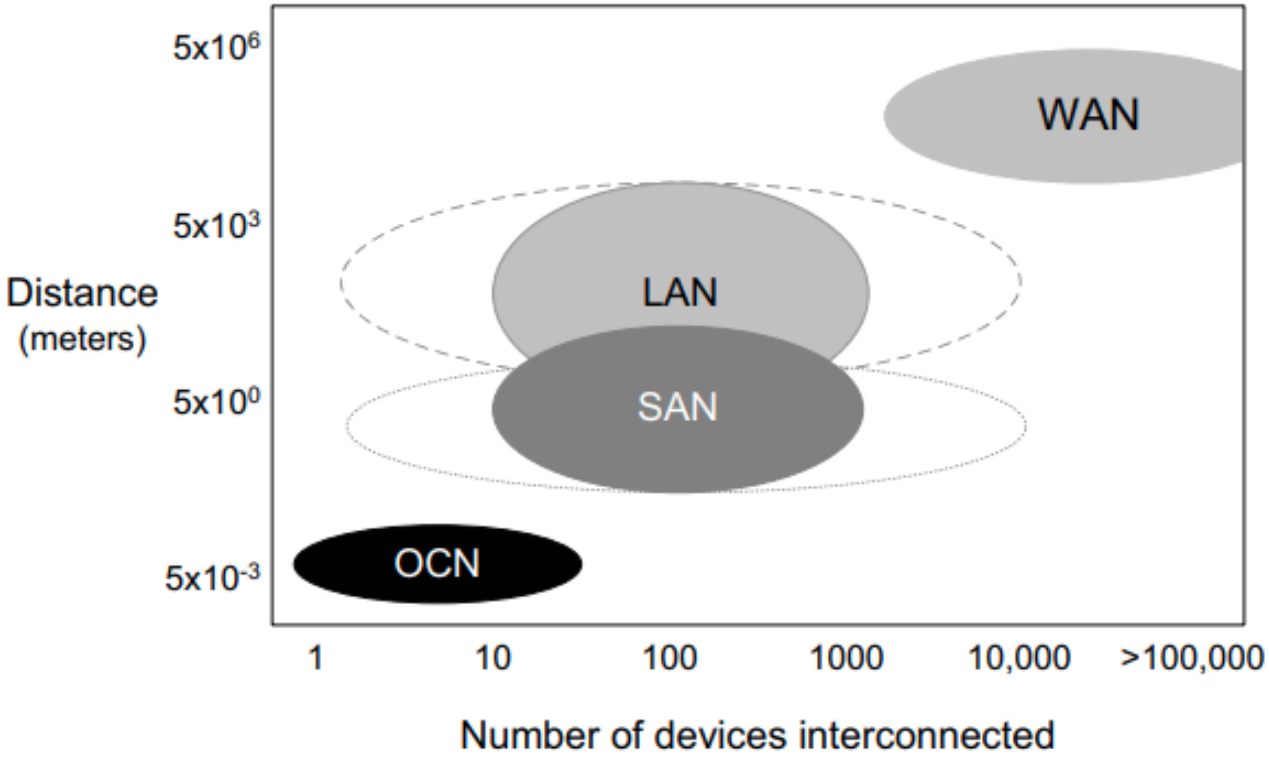
Interconnection Networks at Various Levels

- Interconnection networks are designed for use at different levels
  - On-chip networks (OCNs)
    - interconnecting microarchitecture functional units, register files, caches, processor and IP cores within chips or multichip modules
    - connecting up to only a few tens of such devices with a maximum interconnection distance on the order of centimeters
    - E.g. IBM's CoreConnect, ARM's AMBA, and Sonic's Smart Interconnect, AMD HyperTransport, Intel QuickPath
  - System/storage area networks (SANs)
    - used for interprocessor and processor-memory interconnections within multiprocessor and multicomputer systems
    - connecing hundreds of such devices can be connected, although some supercomputer SANs support the interconnection of many thousands of devices. Distance on the order of a few tens of meters usually— but some SANs have distances spanning a few hundred meters
    - E.g. InfiniBand 120 Gbps, 300 mts

Interconnection Networks at Various Levels

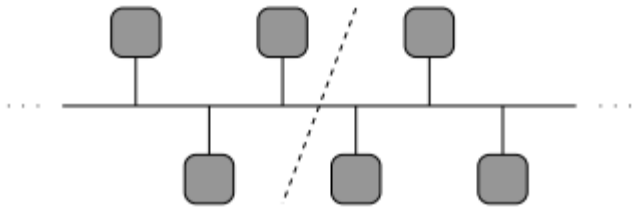
- Local area networks (LANs)
  - used for interconnecting autonomous computer systems distributed across a machine room or throughout a building or campus environment. Interconnecting PCs in a cluster is a prime example.
  - connected only up to a hundred devices, but with bridging, LANs can now connect up to a few thousand devices
  - The maximum interconnect distance covers an area of a few kilometers
  - E.g. Ethernet has a 10 Gbps standard version that supports maximum performance over a distance of 40 kms.
- Wide area networks (WANs)
  - called as long-haul networks, WANs connect computer systems distributed across the globe, which requires internetworking support. WANs connect many millions of computers over distance of many thousands of kilometers. ATM is an example of a WAN.

Interconnection Networks at Various Levels



Shared Medium

- A shared medium allows only one message at a time to be sent
  - If two processors attempt to send messages simultaneously, there will be collisions
  - E.g. PCI (peripheral component interconnect) bus, Ethernet
  - Buses used by commodity systems to connect I/O components, a bus interconnects multiple cores with main memory
- It is blocking. All devices share constant BW. If more devices are connected, lower the available BW. The distance between any two nodes in the network is constant ($O(1)$).



Bisection BW is independent of nodes.

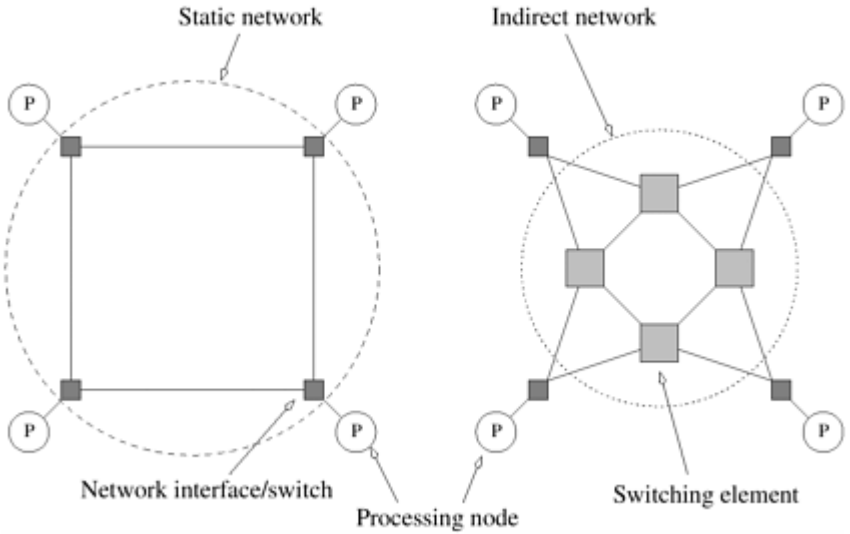
Switch Network Topologies

- A switch network can be represented by a graph in which nodes represent processes and switches and edges represent communication paths
  - Each processor is connected to one switch. Switches connect processors and/or other switches.
- Direct topology or static networks:
  - the ratio of switch nodes t0 processor nodes is 1:1. Every switch node is connected to one processor node and one or more other switch nodes
- Indirect topology or dynamic networks:
  - the ratio of switch nodes to processor nodes is greater than 1:1. Some of the switches simply connect other switches.

Switch Network Topologies

BITS Pilani

- Direct topology or static networks:
  - the ratio of switch nodes t0 processor nodes is 1:1. Every switch node is connected to one processor node and one or more other switch nodes
- Indirect topology or dynamic networks:
  - the ratio of switch nodes to processor nodes is greater than 1:1. Some of the switches simply connect other switches.



Criteria

- Diameter:
  - largest distance between two switch nodes
  - low diameter is better, because the diameter puts a lower bound on the complexity of communication
- Bisection width:
  - the minimum number of edges between switch nodes that must be removed in order to divide the network into two halves (within one)
  - High bisection width is better, because in algorithms requiring large amounts of data movement, the size of the data set divided by the bisection width puts a lower bound on the complexity of the parallel algotithm

Criteria

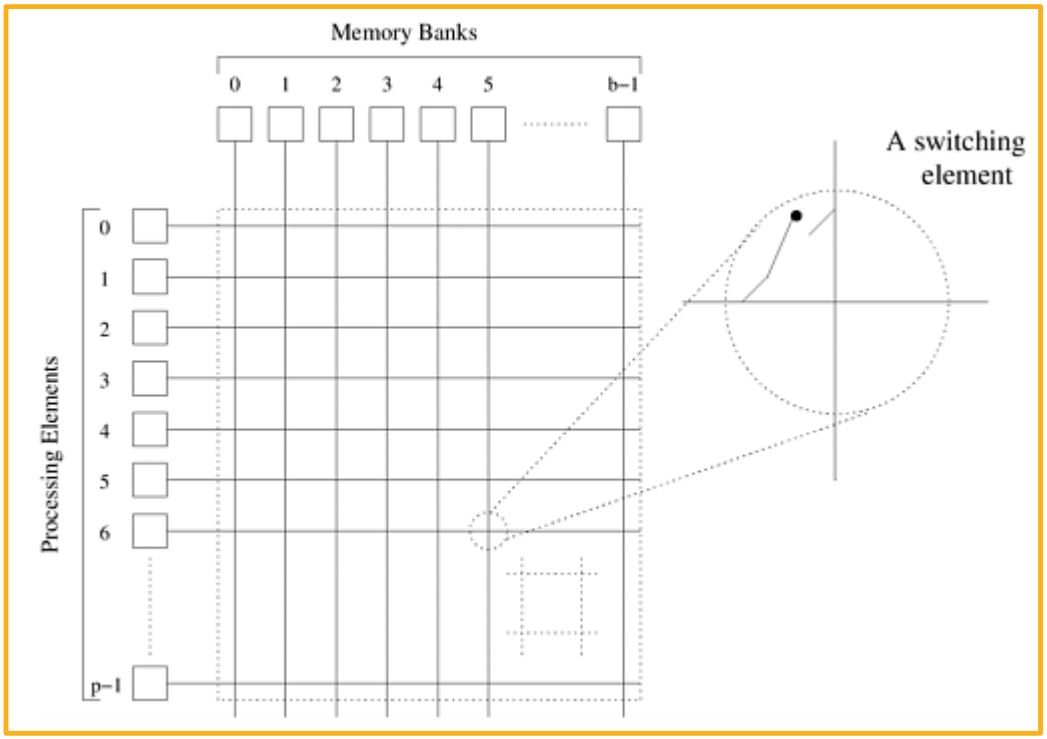
BITS Pilani

- Edges per switch node:
  - It is best if the number of edges per switch node is a constant independent of the network size because then the processor organization scales more easily to systems with large numbers of nodes
- Constant edge length:
  - Although the edges in a network topology do have length, we assume that nodes cannot be infinitely small. As a consequence, the definition of the topology itself can imply that, as the number of nodes increases, the physical distance between them must increase.
    - The binary tree does not have a constant maximum edge length, because as the size of the tree gets larger, the leaf nodes must be placed further apart, which in turn implies that eventually the edges that leave the root of the tree must get longer

Crossbar Networks

BITS Pilani

- A crossbar network employs a grid of switches or switching nodes
  - is a non-blocking network that the connection of a processing node to a memory bank does not block the connection of any other processing nodes to other memory banks.
  - All ports can communicate all time.
- The total number of switching nodes required is O(pb). p>=b. (component count) of the switching network grows as $O(p^2)$
- not very scalable in terms of cost.



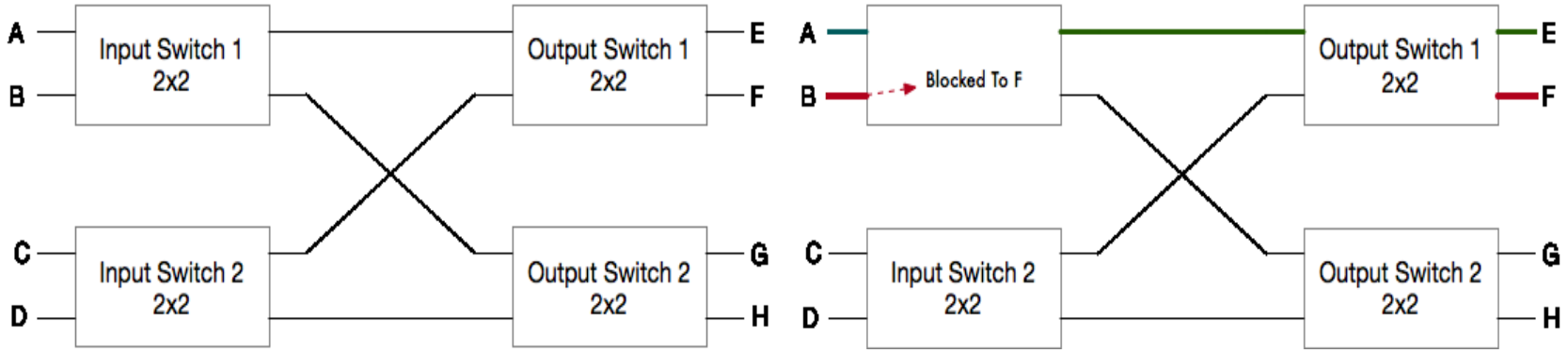
Multi-stage Networks

- The crossbar interconnection network is scalable in terms of performance but unscalable in terms of cost
- The shared bus network is scalable in terms of cost but unscalable in terms of performance
- An intermediate class of networks called multistage interconnection networks lies between these two extremes
  - It is more scalable than the bus in terms of performance and more scalable than the crossbar in terms of cost

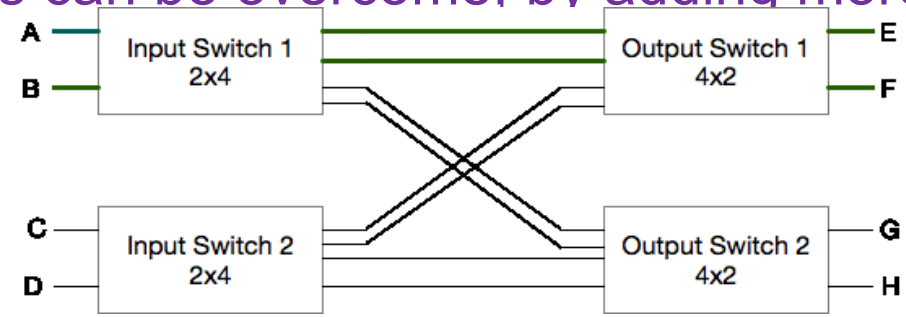
Blocking Network

BITS Pilani

- The following network emulates 4 X 4 cross-bar switch, but is blocking in nature
  - If there is a connection between A and E , then B can't talk to F



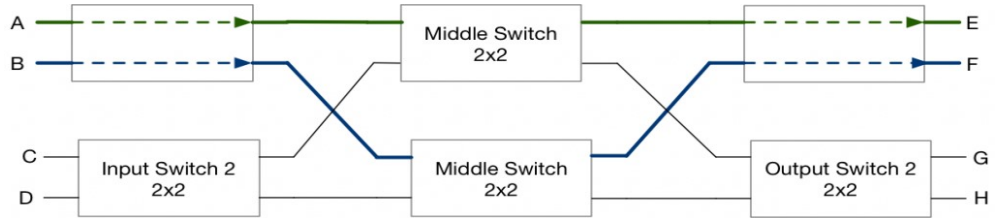
  - This can be overcome, by adding more paths, but very costly.



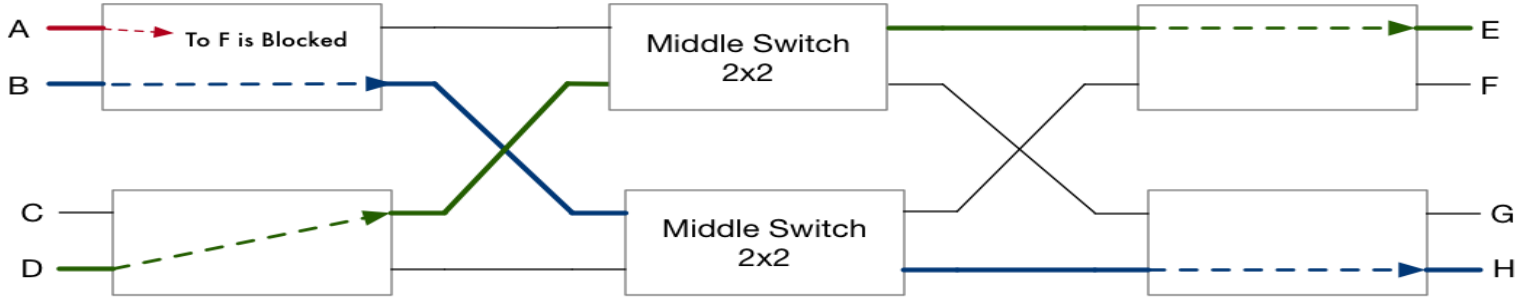
Clos Networks

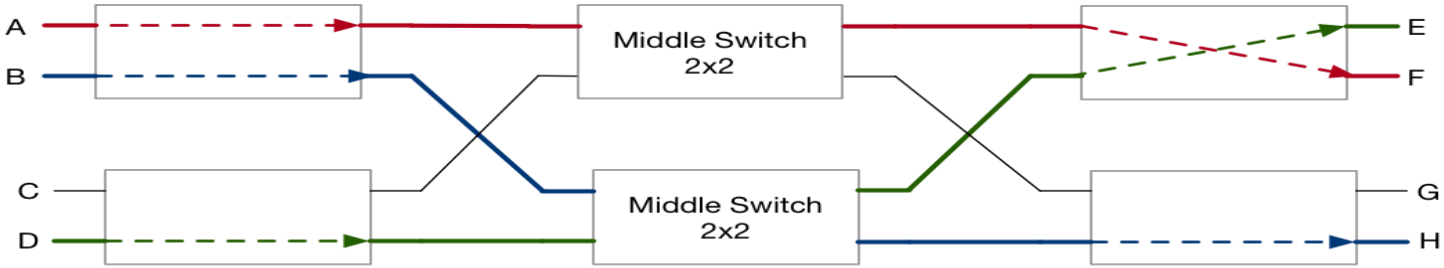
BITS Pilani

- Clos introduced switches in the middle stage, and made the network non-blocking
  - Here A → E and B → F are possible simultaneously



This network is a rearrangeable non-blocking network. Suppose if B→H and D → E are connected, then A→F is not possible unless rearranged.



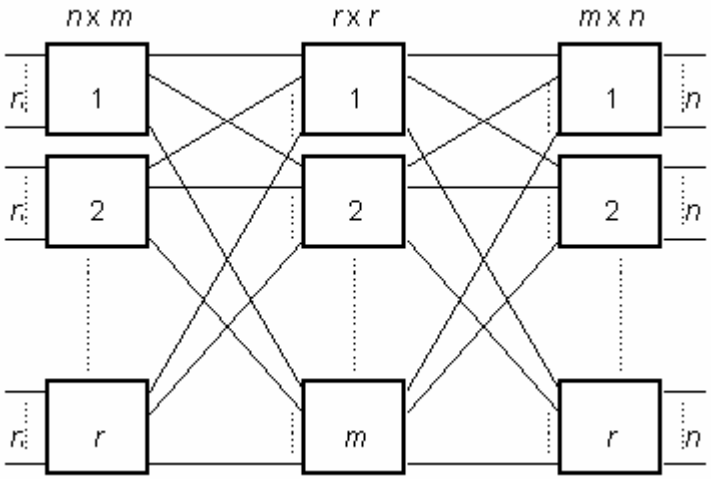


After Rearranging connections

Clos Networks

BITS Pilani

- A symmetric Clos is characterized by a triple,(m, n, r) where m is the number of middle-stage switches, n is the number of input (output) ports on each input (output) switch, and r is the number of input and output switches
- In a Clos network, each middle stage switch has one input link from every input switch and one output link to every output switch
  - The r input switches are n × m crossbars to connect n input ports to m middle switches, the m middle switches are r × r crossbars to connect r input switches to r output switches, and the r output switches are m × n crossbars to connect m middle switches to n output ports.

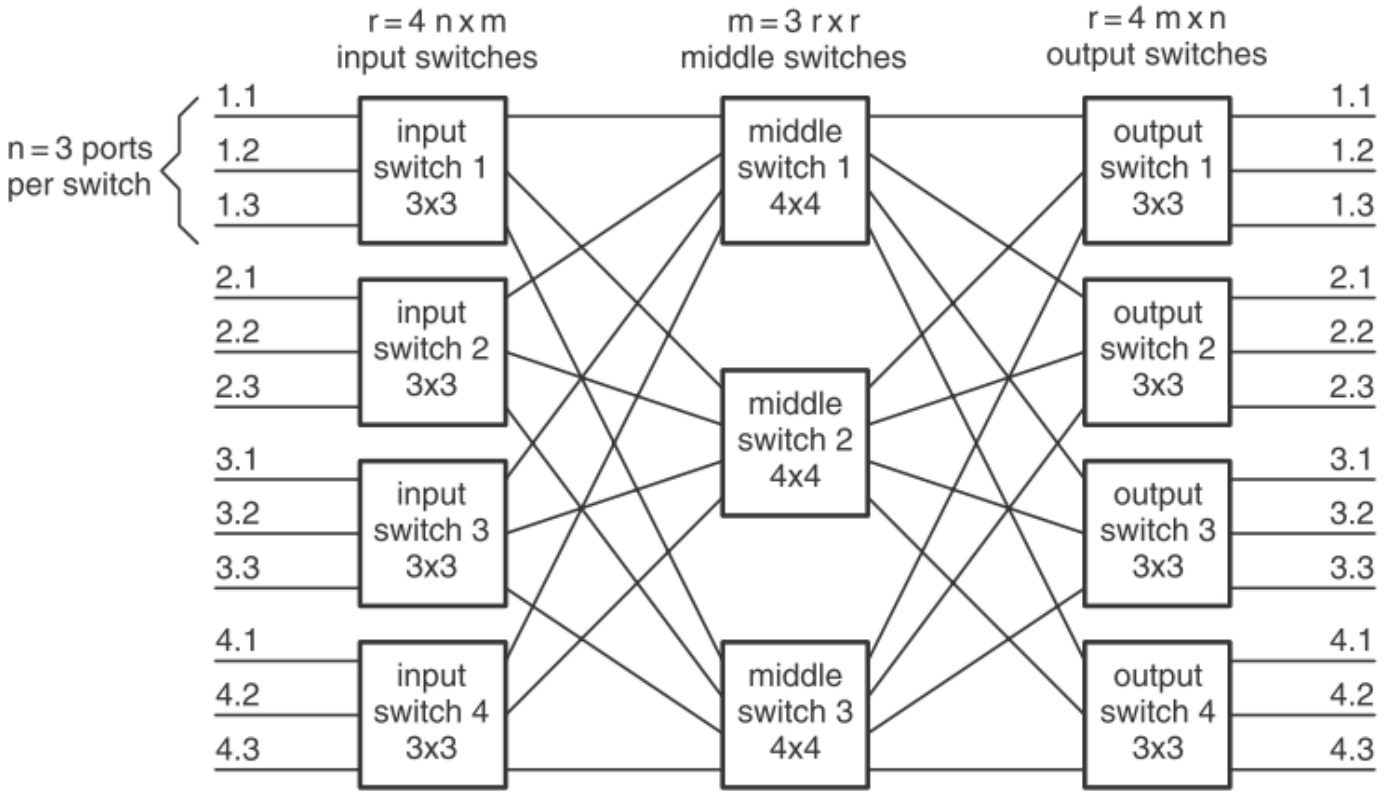


https://en.wikipedia.org/wiki/Multistage_interconnection_networks

Clos Networks

BITS Pilani

- A symmetric Clos is characterized by a triple,(m, n, r) where m is the number of middle-stage switches, n is the number of input (output) ports on each input (output) switch, and r is the number of input and output switches

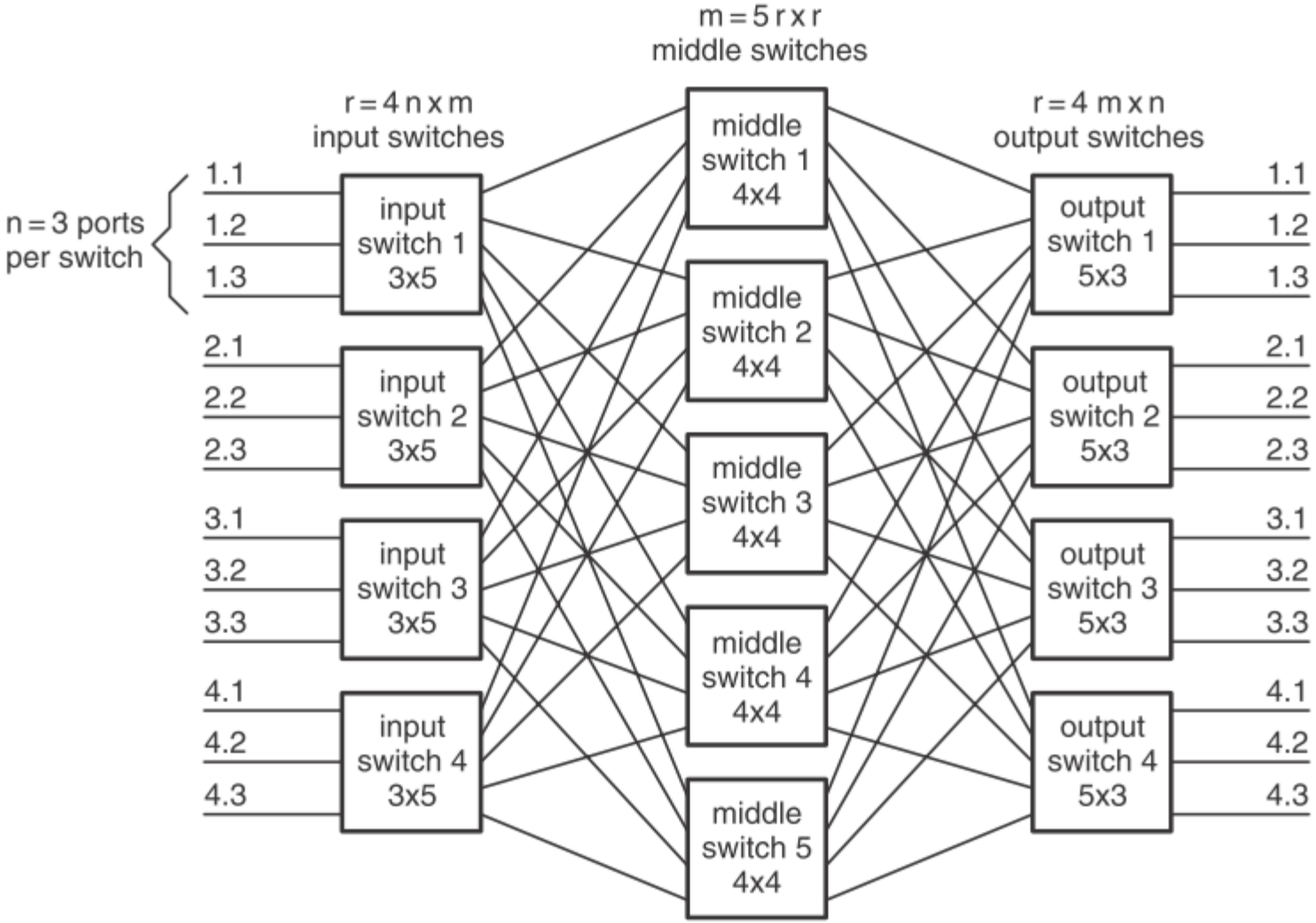


An (m = 3, n = 3, r = 4) symmetric Clos network has r = 4 n × m input switches, m = 3 r × r middle-stage switches, and r = 4 m × n output switches. All switches are crossbars.

Clos Networks

BITS Pilani

- A strictly non-blocking (5,3,4) Clos network



Properties of Clos Networks

BITS Pilani

- The most interesting property of the Clos, and the one from which its non- blocking properties derive, is its path diversity.
  - For a Clos network with m middle switches, there are $|R_{ab}| = m$ routes from any input a to any output b, one through each middle stage switch
- The degree of the input and output switches is $n + m$ and of the middle switches is $2r$
- During routing, the decision is at the input switch only, where any of the m middle switches can be chosen
  - The middle switches must choose the single link to the output switch (and the route is not possible if this link is busy)
  - Similarly, the output switch must choose the selected output port
- Thus, the problem of routing in a Clos network is reduced to the problem of assigning each circuit (or packet) to a middle

Properties of Clos Networks

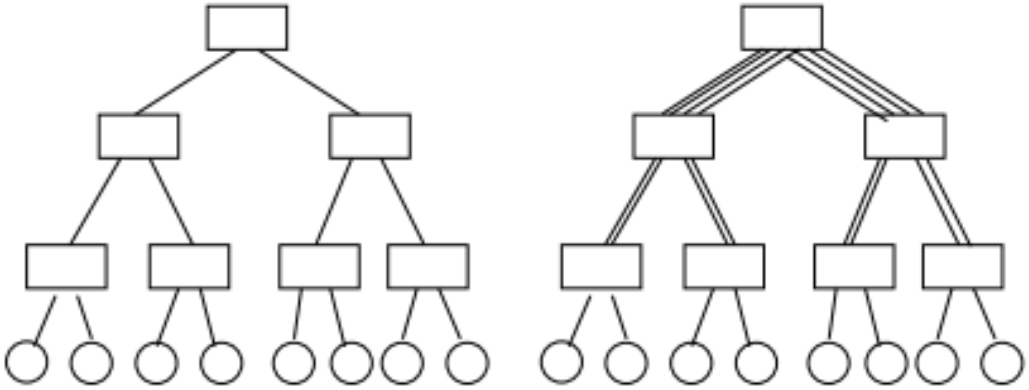
BITS Pilani

- A Clos network is strictly non-blocking for unicast traffic iff $m \geq 2n - 1$
- A Clos network with $m \geq n$ is re-arrangeable
- The channel bisection of Clos is 2*n*r = 2N = 2* No of Hosts
- In Clos network, there are *m* routes for any input and output

Fat Tree

BITS Pilani

- In the binary tree, the number of links (and thus the aggregate bandwidth) is reduced by half at each level from the leaves to the root
  - This can cause serious congestion towards the root
- The binary fat-tree topology remedies this situation by maintaining the same bandwidth at each level of the network



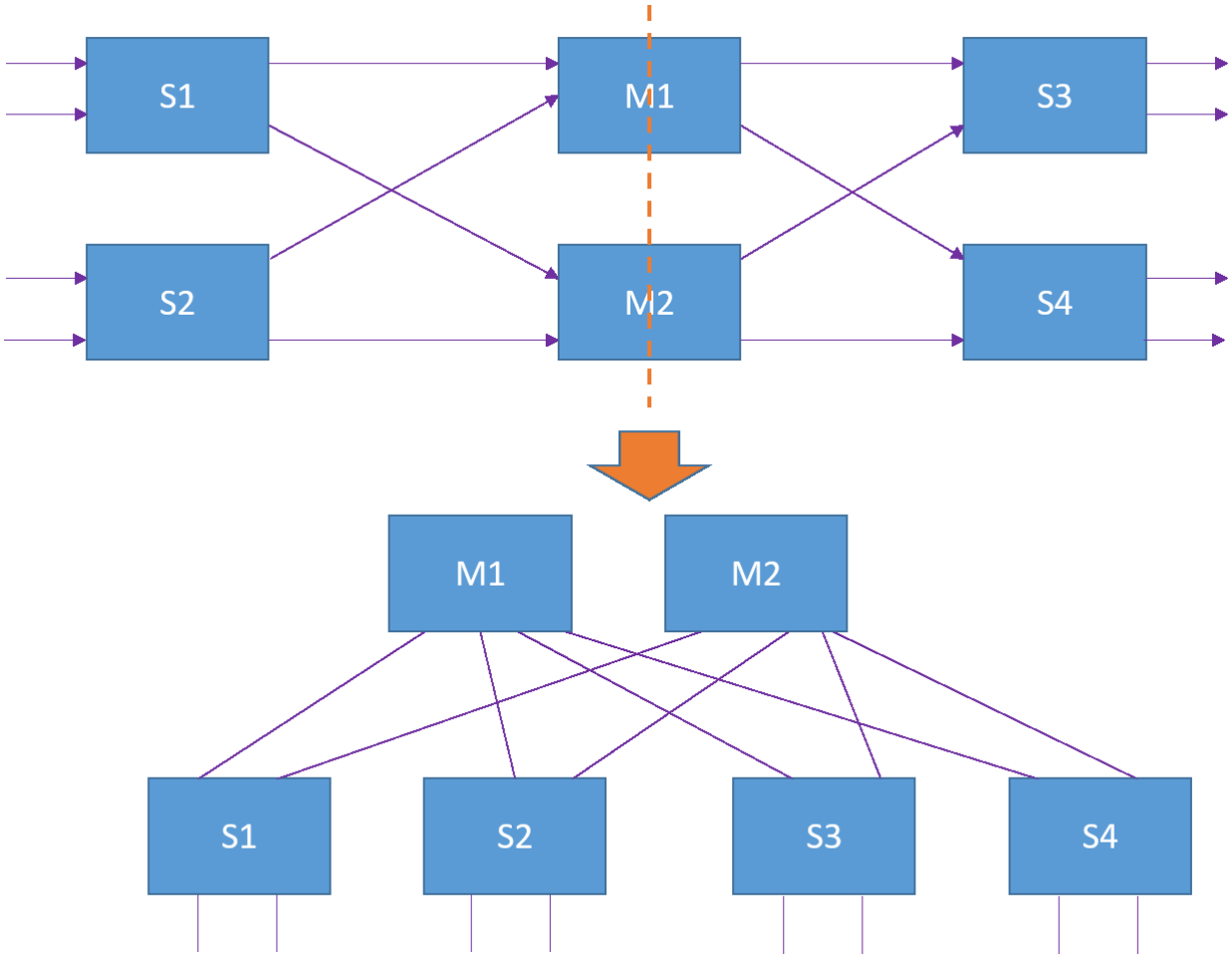
(a) Binary tree (b) Binary fat–tree

Source: http://www.cs.fsu.edu/~xyuan/paper/07sigmetrics.pdf

Fat Tree

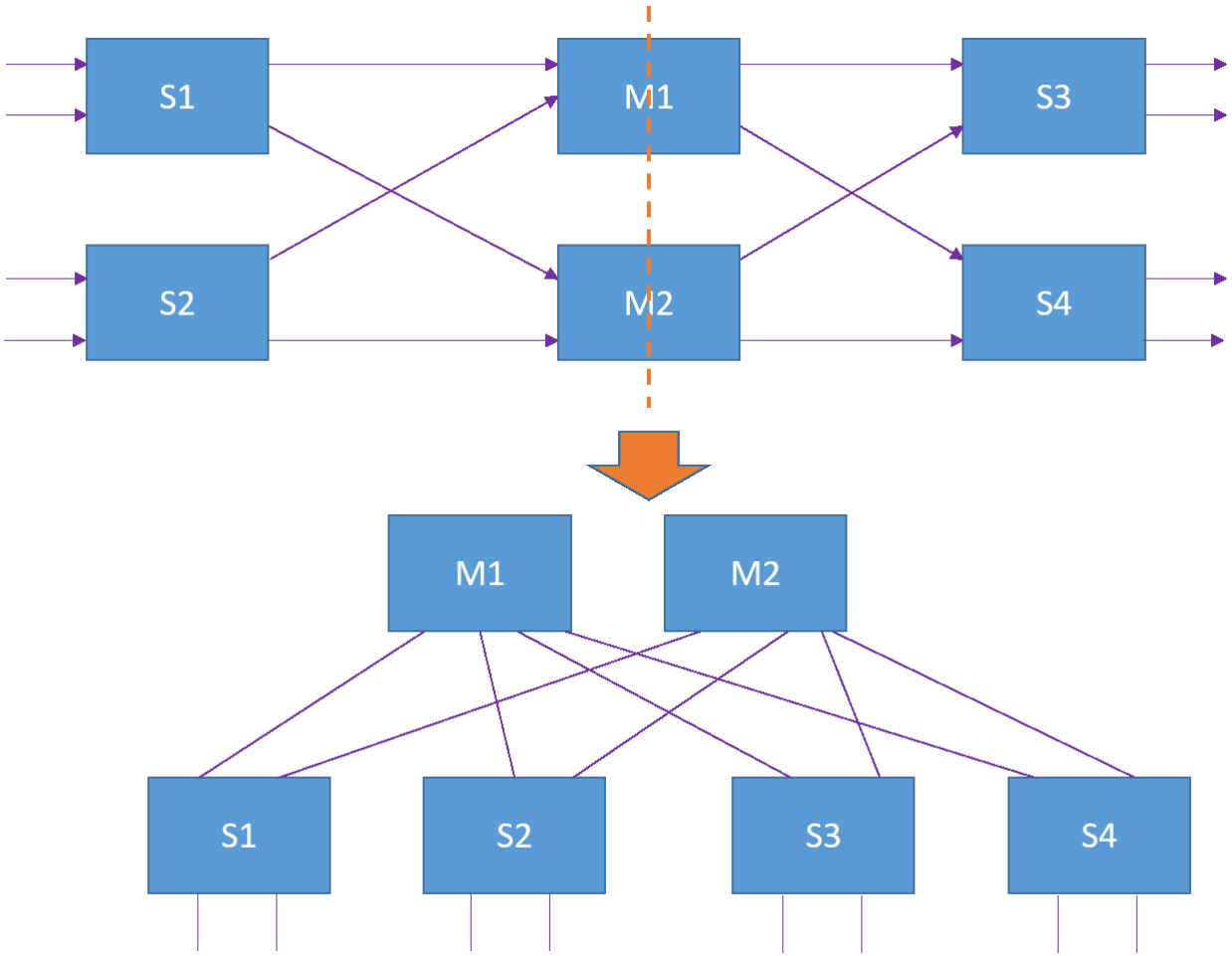
BITS Pilani

- A Fat-Tree is generally represented by FT(k,n) where k is the radix of the switch and “n” is the levels of the Fat-Tree



Fat Tree

- A three stage Clos network when folded gives 2-level Fat Tree FT(k,2)
  - These networks are called folded Clos, Leaf and Spine or Fat Tree
- All switches in Fat Tree have same radix (#ports)
- A 4 port 2 level FT(4,2) is shown here

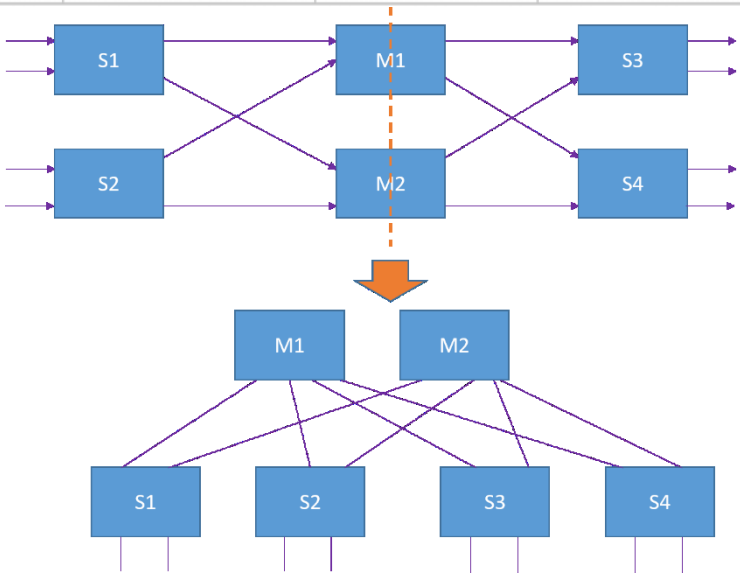


Fat Tree

BITS Pilani

- A FT(4,2) can support
  - 4/2 #core switches =2
  - $(4/2)^2$ * 2 hosts = 8
  - Total #switches 3*(4/2) = 6
  - Total # edge switches = 4

	Fat-tree with L levels	Two level Fat Tree L=2	Three Level Fat Tree L = 3	Four Level Fat Tree L=4
Number of Core switches	$\left(\frac{k}{2}\right)^{L-1}$	$\frac{k}{2}$	$\left(\frac{k}{2}\right)^{2}$	$\left(\frac{k}{2}\right)^{3}$
Number of Hosts supported	$2\left(\frac{k}{2}\right)^{L}$	$2\left(\frac{k}{2}\right)^{2}$	$2\left(\frac{k}{2}\right)^{3}$	$2\left(\frac{k}{2}\right)^{4}$
Total Switches	$(2L-1)\left(\frac{k}{2}\right)^{L-1}$	$3(\frac{k}{2})$	$5\left(\frac{k}{2}\right)^{2}$	$7\left(\frac{k}{2}\right)^{2}$
Number of Edge Switches	$2\left(\frac{k}{2}\right)^{L-1}$	k	$2\left(\frac{k}{2}\right)^{2}$	$2\left(\frac{k}{2}\right)^{3}$
Number of Pods	$2\left(\frac{k}{2}\right)^{L-2}$	NA	k	$k^2/2$

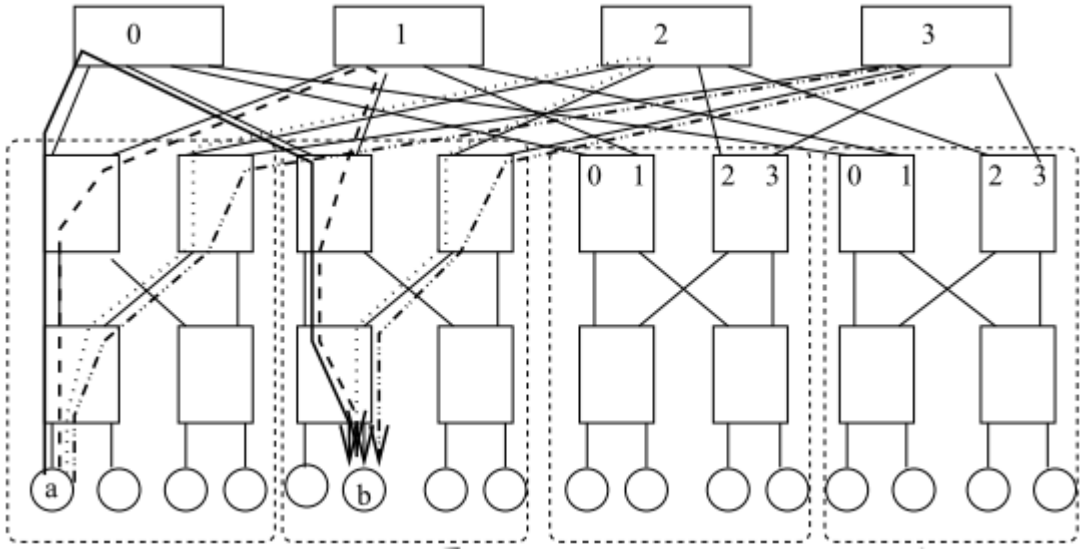


Fat Tree

BITS Pilani

- A FT(4,3) can support
  - $(4/2)^2$ #core switches =4
  - $(4/2)^3$ * 2 hosts = 16
  - Total #switches $5*(4/2)^2 = 20$
  - Total # edge switches = $(2*(4/2)^2)=8$
  - Number of pods= $2(4/2)^{3-2}=4$

	Fat-tree with L levels	Two level Fat Tree L=2	Three Level Fat Tree L = 3	Four Level Fat Tree L=4
Number of Core switches	$\left(\frac{k}{2}\right)^{L-1}$	$\frac{k}{2}$	$\left(\frac{k}{2}\right)^{2}$	$\left(\frac{k}{2}\right)^{3}$
Number of Hosts supported	$2\left(\frac{k}{2}\right)^{L}$	$2\left(\frac{k}{2}\right)^{2}$	$2\left(\frac{k}{2}\right)^{3}$	$2\left(\frac{k}{2}\right)^{4}$
Total Switches	$(2L-1)\left(\frac{k}{2}\right)^{L-1}$	$3(\frac{k}{2})$	$5\left(\frac{k}{2}\right)^{2}$	$7\left(\frac{k}{2}\right)^{2}$
Number of Edge Switches	$2\left(\frac{k}{2}\right)^{L-1}$	k	$2\left(\frac{k}{2}\right)^{2}$	$2\left(\frac{k}{2}\right)^{3}$
Number of Pods	$2\left(\frac{k}{2}\right)^{L-2}$	NA	k	$k^2/2$



Source: http://www.cs.fsu.edu/~xyuan/paper/07sigmetrics.pdf

Network Topologies: Multistage Omega Network

BITS Pilani

- One of the most commonly used multistage interconnects is the Omega network.
- This network consists of *log p* stages, where *p* is the number of inputs/outputs.
- At each stage, input *i* is connected to output *j* if:

$$j = \begin{cases} 2i, & 0 \leq i \leq p/2 - 1 \\ 2i + 1 - p, & p/2 \leq i \leq p - 1 \end{cases}$$

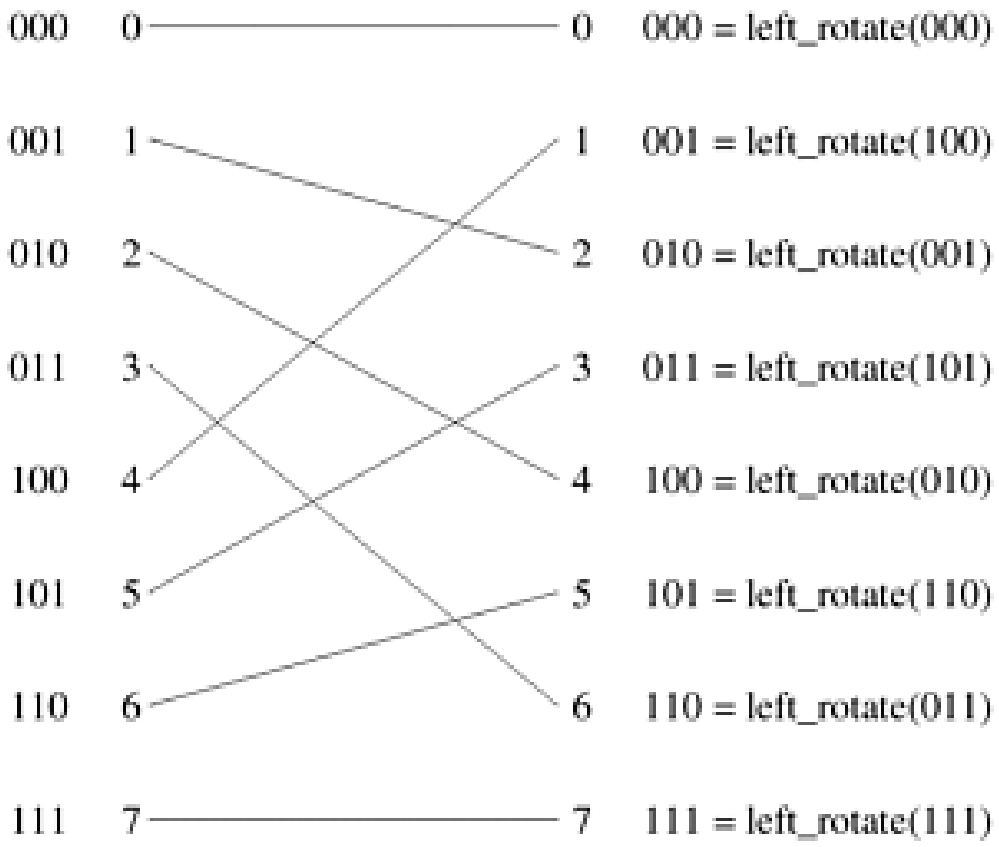
An omega network has *p/2* × *log p* switching nodes, and the cost of such a network grows as *(p log p).*
Cost of crossbar grows as O(p^2)
Cost of bus grows as O(p)

Network Topologies: Multistage Omega Network

BITS Pilani

- Each stage of the Omega network implements a perfect shuffle as follows:

A perfect shuffle interconnection for eight inputs and outputs.



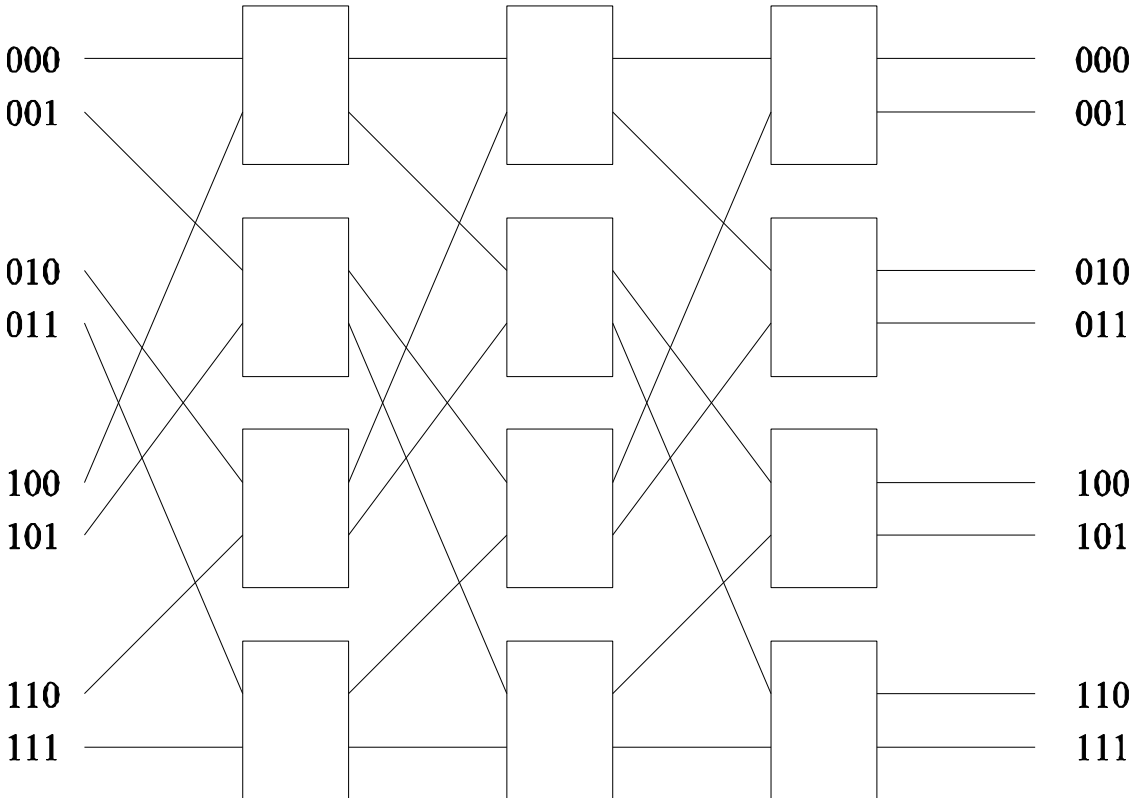
Network Topologies: Multistage Omega Network

BITS Pilani

- A complete Omega network with the perfect shuffle interconnects and switches can now be illustrated:

A complete omega network connecting eight inputs and eight outputs.

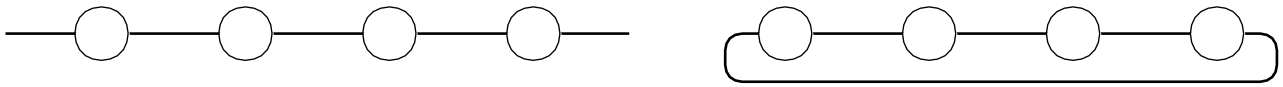
An omega network has $p/2 \times \log p$ switching nodes, and the cost of such a network grows as $(p \log p)$.



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.
- A generalization to 2 dimensions has nodes with 4 neighbors, to the north, south, east, and west.
- A further generalization to d dimensions has nodes with 2d neighbors.
- A special case of a d-dimensional mesh is a hypercube. Here, d = log p, where p is the total number of nodes.

Network Topologies: Linear Arrays

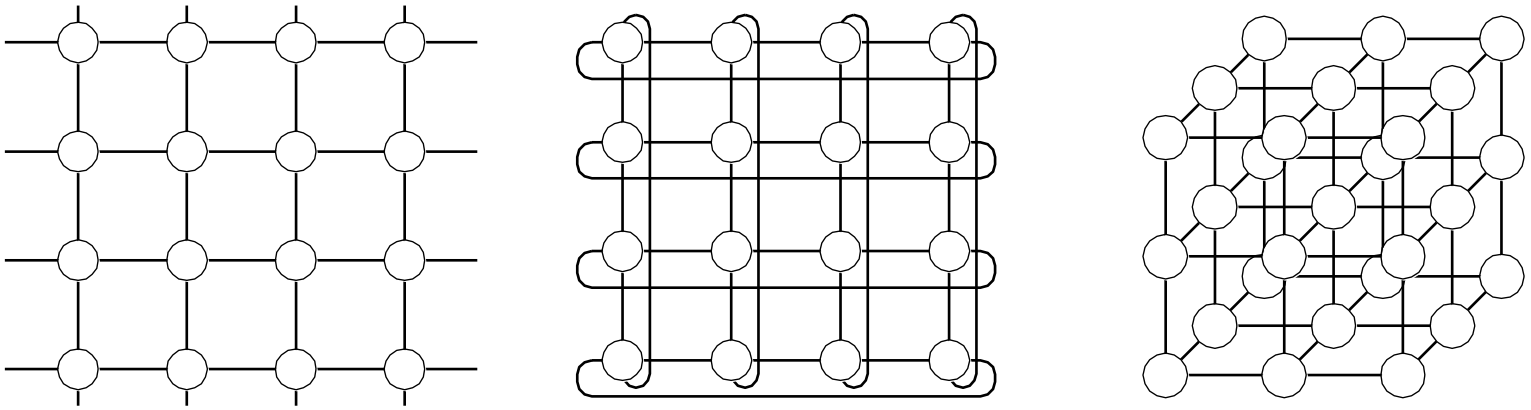


(a) (b)

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

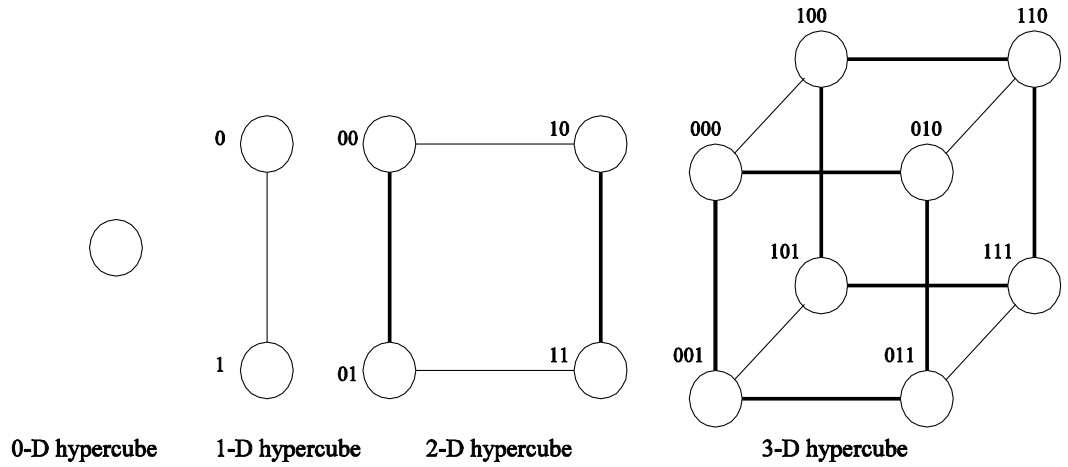
Network Topologies: Two- and Three Dimensional Meshes

BITS Pilani

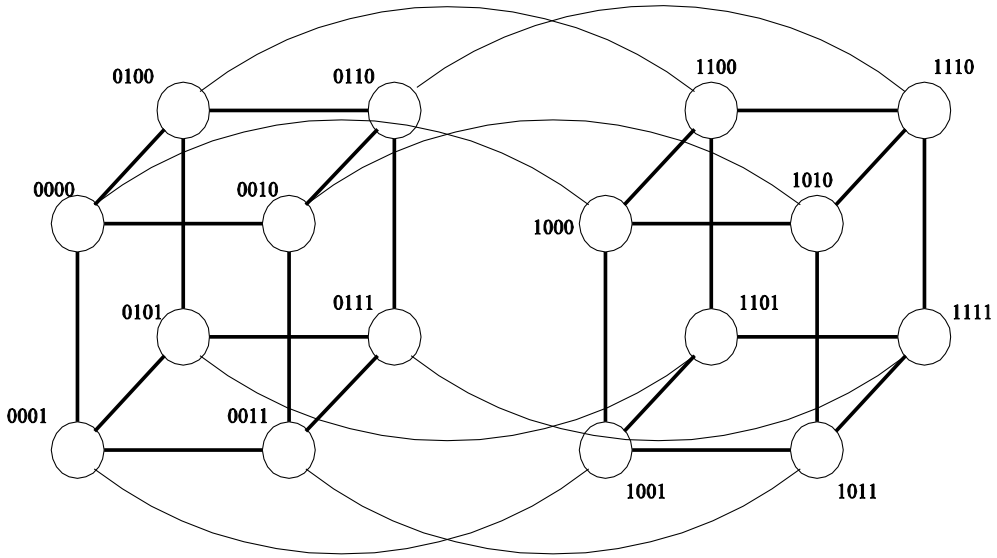


Two and three dimensional meshes: (a) 2-D mesh with no wraparound; (b) 2-D mesh with wraparound link (2-D torus); and (c) a 3-D mesh with no wraparound.

Network Topologies: Hypercubes and their Construction



0-D hypercube 1-D hypercube 2-D hypercube 3-D hypercube

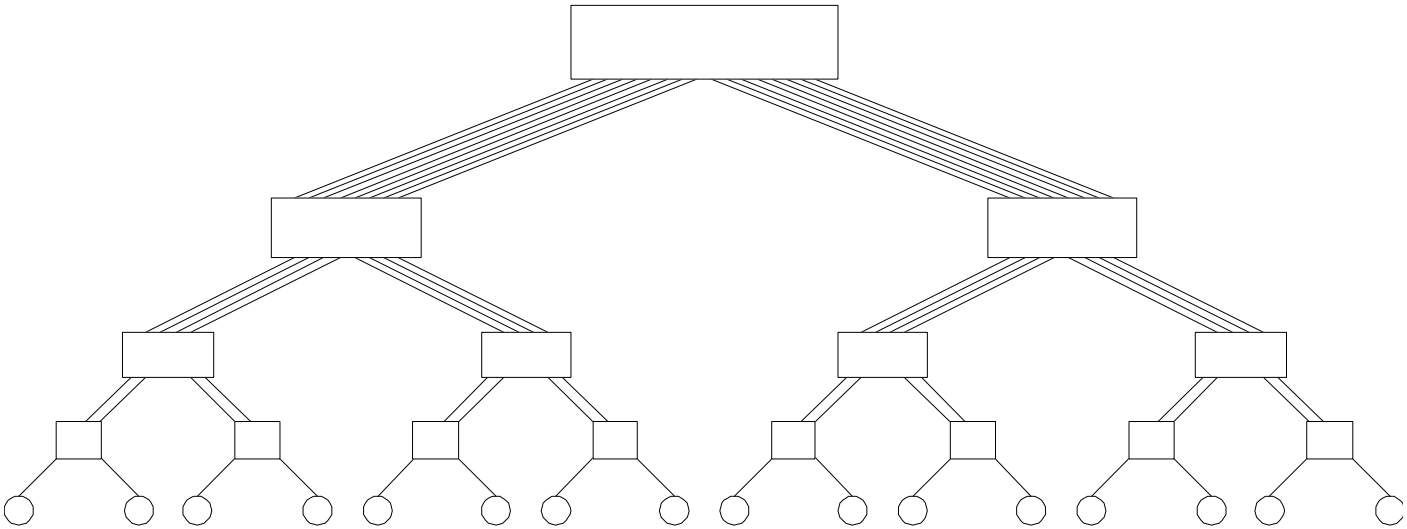


4-D hypercube

Construction of hypercubes from hypercubes of lower dimension.

- 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.

Network Topologies: Fat Trees



A fat tree network of 16 processing nodes.

Evaluating Static Interconnection Networks

- *Diameter:* The distance between the farthest two nodes in the network. The diameter of a linear array is $p - 1$, that of a mesh is $2(\sqrt{p} - 1)$, that of a tree and hypercube is $\log p$, and that of a completely connected network is $O(1)$.
- *Bisection Width:* The minimum number of wires you must cut to divide the network into two equal parts. The bisection width of a linear array and tree is 1, that of a mesh is $\sqrt{p}$, that of a hypercube is $p/2$ and that of a completely connected network is $p^2/4$.
- *Cost:* The number of links or switches (whichever is asymptotically higher) is a meaningful measure of the cost. However, a number of other factors, such as the ability to layout the network, the length of wires, etc., also factor in to the cost.

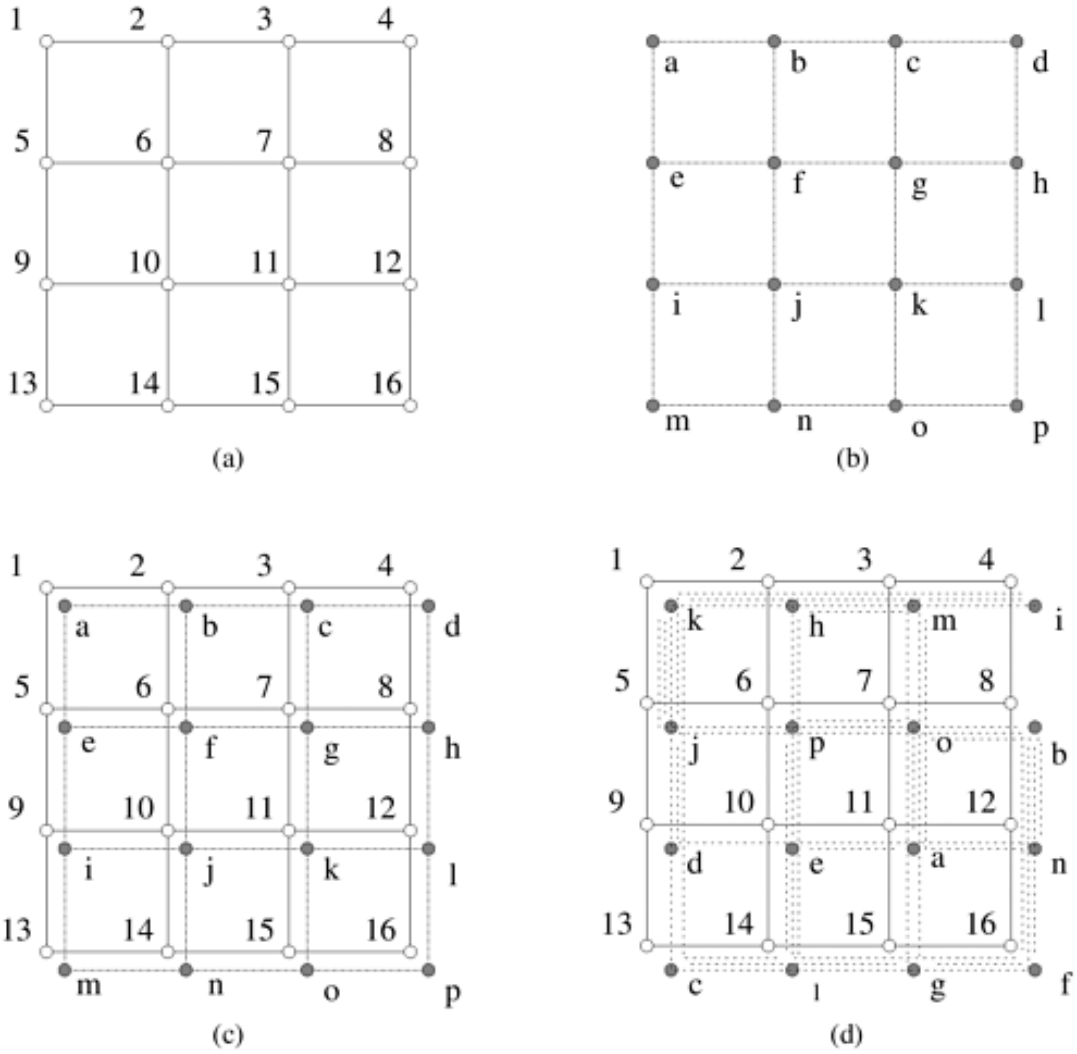
Evaluating Static Interconnection Networks

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

Evaluating Dynamic Interconnection Networks

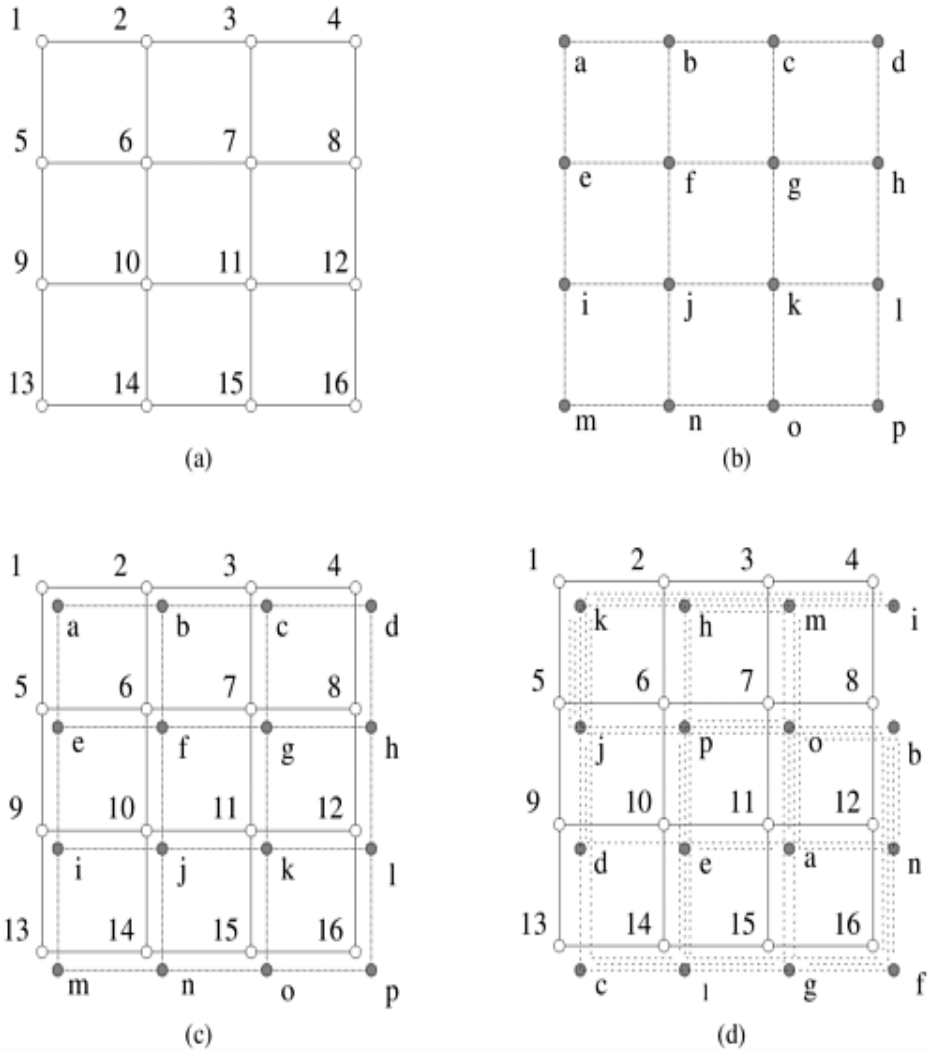
Network	Diameter	Bisection Width	Arc Connectivity	Cost (No. of links)
Crossbar	1	p	1	p^2
Omega Network	$\log p$	$p/2$	2	$p/2$
Dynamic Tree	$2 \log p$	1	2	$p - 1$

Impact of Process-Processor Mapping



- Impact of process mapping on performance: (a) underlying architecture; (b) processes and their interactions; (c) an intuitive mapping of processes to nodes; and (d) a random mapping of processes to nodes.

Impact of Process-Processor Mapping



The underlying architecture is a 16-node mesh with nodes labeled from 1 to and the algorithm has been implemented as 16 processes, labeled 'a' through 'p'. The algorithm has been tuned for execution on a mesh in such a way that there are no congesting communication operations. We now consider two mappings of the processes to nodes as illustrated in Figures 2.29(c) and (d). Figure 2.29(c) is an intuitive mapping and is such that a single link in the underlying architecture only carries data corresponding to a single communication channel between processes. Figure 2.29(d), on the other hand, corresponds to a situation in which processes have been mapped randomly to processing nodes. In this case, it is easy to see that each link in the machine carries up to six channels of data between processes. This may potentially result in considerably larger communication times if the required data rates on communication channels between processes is high

Mapping Techniques for Graphs

- Often, we need to embed a known communication pattern into a given interconnection topology.
- We may have an algorithm designed for one network, which we are porting to another topology.

For these reasons, it is useful to understand mapping between graphs.

Mapping Techniques for Graphs: Metrics

- When mapping a graph *G(V,E)* into *G'(V',E'),* the following metrics are important:
- The maximum number of edges mapped onto any edge in *E'* is called the *congestion* of the mapping.
- The maximum number of links in *E'* that any edge in *E* is mapped onto is called the *dilation* of the mapping.
- The ratio of the number of nodes in the set *V'* to that in set *V* is called the *expansion* of the mapping.

Embedding a Linear Array into a Hypercube

- A linear array (or a ring) composed of 2^d nodes (labeled 0 through $2^d - 1$) can be embedded into a *d*-dimensional hypercube by mapping node *i* of the linear array onto node
- *G(i, d)* of the hypercube. The function *G(i, x)* is defined as follows:

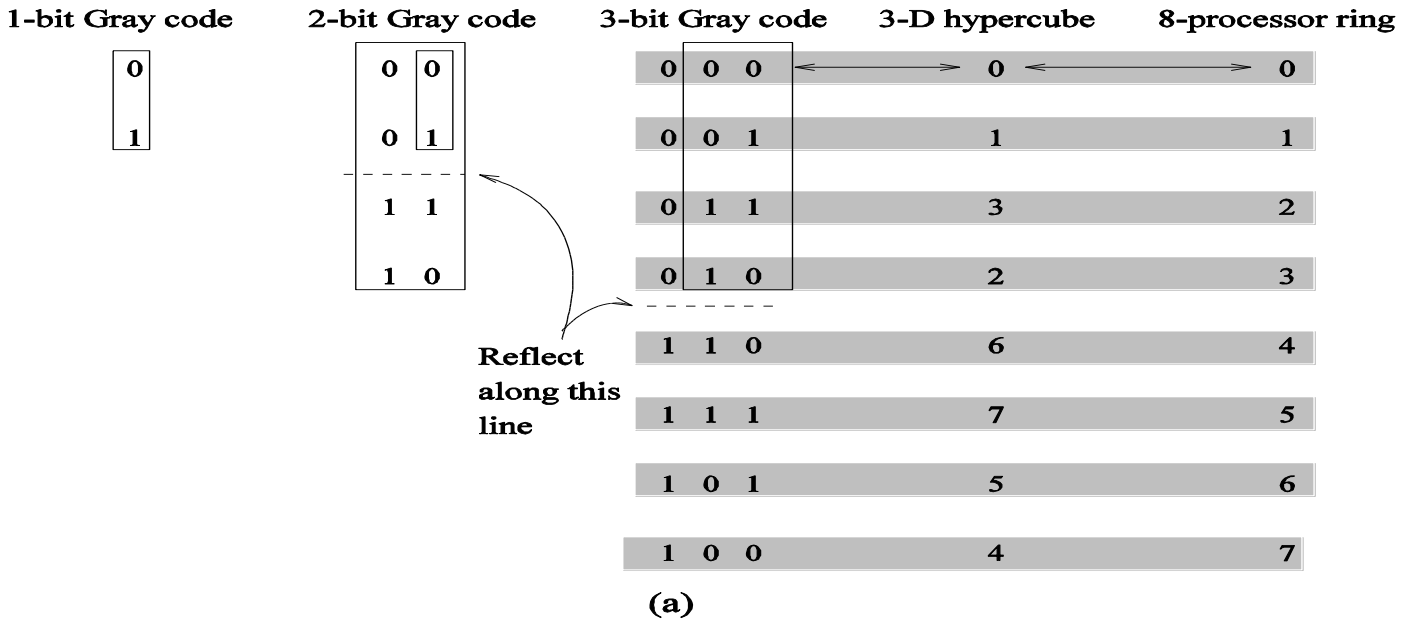
$$
\begin{aligned}
G(0,1) &= 0 \\
G(1,1) &= 1 \\
G(i,x+1) &= \begin{cases} G(i,x), & i < 2^x \\ 2^x + G(2^{x+1} - 1 - i, x), & i \geq 2^x \end{cases}
\end{aligned}
$$

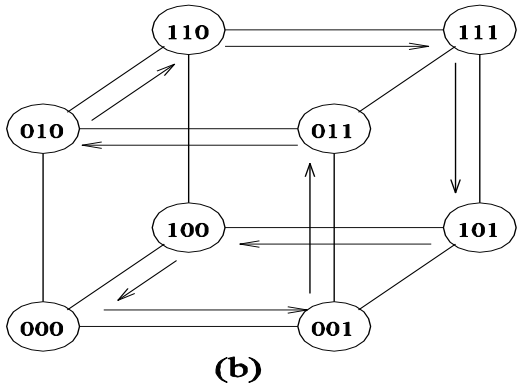
Embedding a Linear Array into a Hypercube

The function *G* is called the *binary reflected Gray code* (RGC).

Since adjoining entries (*G(i, d)* and *G(i + 1, d)*) differ from each other at only one bit position, corresponding processors are mapped to neighbors in a hypercube. Therefore, the congestion, dilation, and expansion of the mapping are all 1.

Embedding a Linear Array into a Hypercube: Example





(a) A three-bit reflected Gray code ring; and (b) its embedding into a three-dimensional hypercube.

References

innovate achieve lead

BITS Pilani
Pilani Campus

Thank You