Physical Organization

Slides from Grama et. al. “Introduction to Parallel Computing”

for Computation

Dichotomy of Parallel Computing Platforms

An explicitly parallel program must specify concurrency and interaction between concurrent subtasks.

The former is sometimes also referred to as the control structure and the latter as the communication model.

Control Structure of Parallel Programs

Parallelism can be expressed at various levels of granularity - from instruction level to processes.

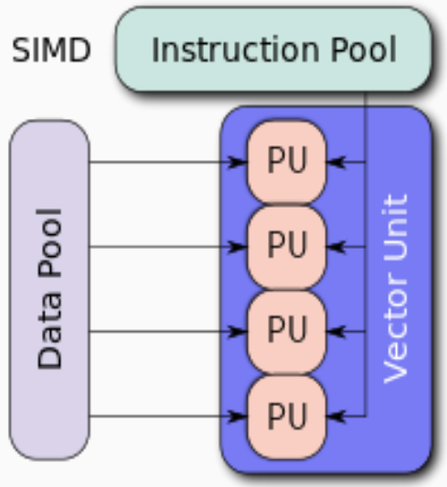
Between these extremes exist a range of models, along with corresponding architectural support.

Control Structure of Parallel Programs

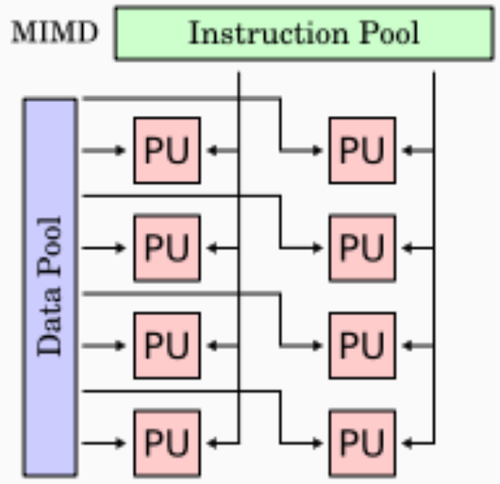
Processing units in parallel computers either operate under the centralized control of a single control unit or work independently.

If there is a single control unit that dispatches the same instruction to various processors (that work on different data), the model is referred to as single instruction stream, multiple data stream (SIMD).

If each processor has its own control control unit, each processor can execute different instructions on different data items. This model is called multiple instruction stream, multiple data stream (MIMD).



SIMD



MIMD

Figures from wikipedia entries for SIMD and MIMD

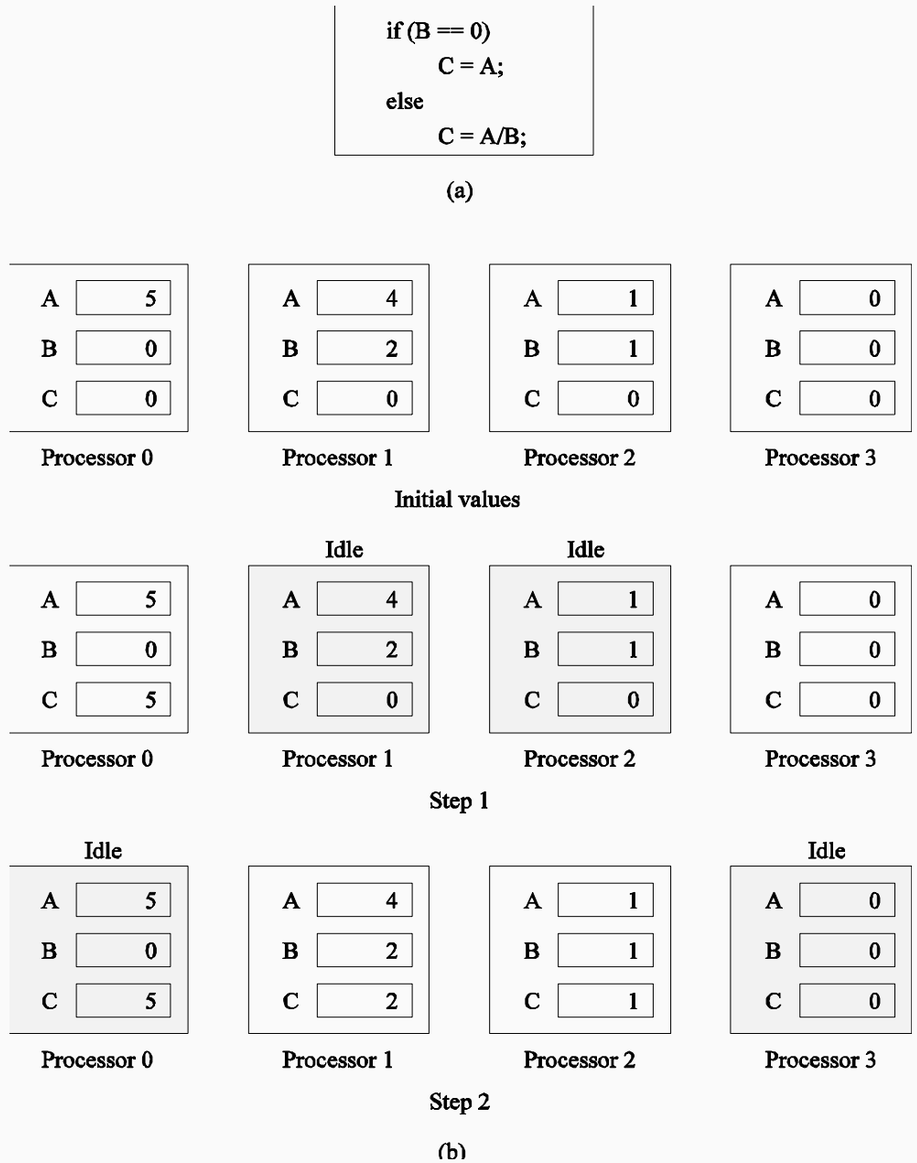
SIMD Processors

Some of the earliest parallel computers such as MasPar MP-1.

Variants of this concept → co-processing units like Intel MMX.

SIMD relies on the regular structure of computations (such as those in image processing).

It is often necessary to selectively turn off operations on certain data items. For this reason, most SIMD programming paradigms allow for an 'activity mask', which determines if a processor should participate in a computation or not.



Grama et. al.

Executing a conditional statement on an SIMD computer with four processors:

(a) conditional;

(b) execution.

SIMD-MIMD Comparison

1. SIMD computers require less hardware than MIMD computers (single control unit).
2. However, since SIMD processors ae specially designed, they tend to be expensive and have long design cycles.
3. In contrast, platforms supporting the SPMD paradigm can be built from inexpensive off-the-shelf components with relatively little effort in a short amount of time.
4. Not all applications are naturally suited to SIMD processors.

Contemporary SIMD vs. MIMD

- SIMD →SSE instructions, GPU (vector ops)
- MIMD/SPMD
  - All work off the same program/code
  - At any given time, some maybe working on one part of the code, while others are on a different part of the code.
  - Conditional control flow
  - Vulnerable to load imbalance

for Communication

Communication Model of Parallel Platforms

There are two primary forms of data exchange between parallel tasks:

accessing a shared data space

exchanging messages.

1. Platforms (or programming frameworks) that provide a shared data space are called shared-address-space machines or multiprocessors.

2. Platforms (or programming frameworks) that support messaging are also called message passing platforms.

1. Shared-Address-Space Platforms

Part (or all) of the memory is accessible to all processors.

Processors interact by modifying data objects stored in this shared-address-space.

If the time taken by a processor to access any memory word in the system global or local is identical, the platform is classified as a uniform memory access (UMA), else, a non-uniform memory access (NUMA) machine.

2. Message-Passing Platforms

These platforms comprise of a set of processors and their own (exclusive) memory.

Instances of such a view come naturally from clustered workstations and non-shared-address-space multicore.

These platforms are programmed using (variants of) send and receive primitives.

Libraries such as MPI and PVM provide such primitives.

Message Passing vs. Shared Address Space Platforms

Message passing requires little hardware support, other than a network.

Shared address space platforms can easily emulate message passing. The reverse is more difficult to do (in an efficient manner).

Architecture of an Ideal Parallel Computer

A natural extension of the Random Access Machine (RAM) serial architecture is the Parallel Random Access Machine, or PRAM — The Ideal Machine

PRAMs consist of *p* processors and a global memory of unbounded size that is uniformly accessible to all processors.

Processors share a common clock but may execute different instructions in each cycle.

Architecture of an Ideal Parallel Computer

Depending on how simultaneous memory accesses are handled, PRAMs can be divided into four subclasses.

Exclusive-read, exclusive-write (EREW) PRAM.

Concurrent-read, exclusive-write (CREW) PRAM.

Exclusive-read, concurrent-write (ERCW) PRAM.

Concurrent-read, concurrent-write (CRCW) PRAM.

How do you deal w/ concurrency

4 possible policies

- Common: write only if all values are identical.
- Arbitrary: write the data from a randomly selected processor.
- Priority: follow a predetermined priority order.
- Sum: Write the sum of all data items.

What's the Complexity?

1. Processors (p) and memories (m words) are connected via switches

2. Complexity of Switch?

   O(1) – Ideal Machine!!

3. Complexity of ideal machine?

   Since these switches must operate in *O(1)* time at the level of words, for a system of *p* processors and *m* words, the switch complexity is *O(mp).*

Clearly, for meaningful values of *p* and *m*, a true PRAM is not realizable.

Interconnects

How do you connect >2 ‘nodes'?
Ring/Star/Fully connected/Mesh/…

What are Interconnects

Interconnection networks carry data processors ←→ memory.

Interconnects are made of 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.

Switch

Switches map a fixed number of inputs to outputs.

The total number of ports on a switch is the *degree* of the switch.

The cost of a switch grows as??

> the square of the degree of the switch,
> the peripheral hardware linearly as the degree, and the packaging costs linearly as the number of pins.

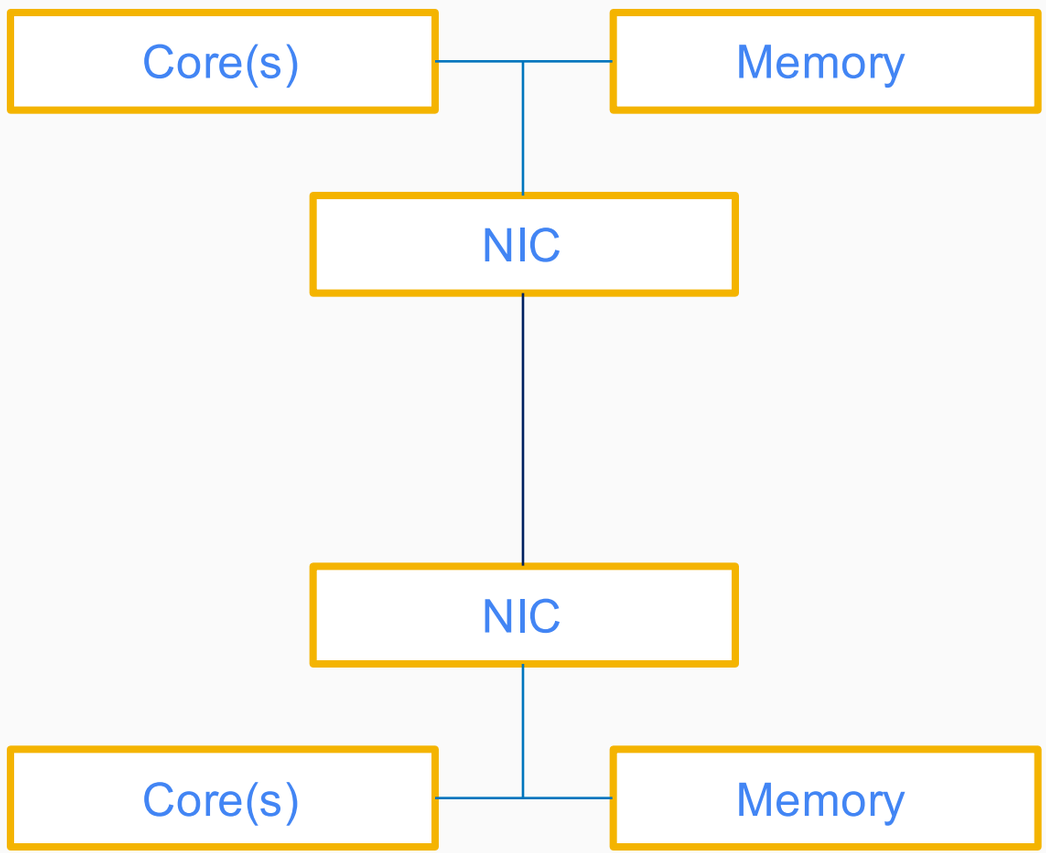
Network Interfaces (NIC)

Processors talk to the network via a network interface.

The network interface may hang off the I/O bus or the memory bus.

In a physical sense, this distinguishes a cluster from a tightly coupled multicore machine.

The relative speeds of the I/O and memory buses impact the performance of the network.



Layout

Network vs. Bus

Network Topologies

A variety of network topologies have been proposed and implemented.

These topologies tradeoff performance for cost.

Commercial machines often implement hybrids of multiple topologies for reasons of packaging, cost, and available components.

Network Topologies: Buses

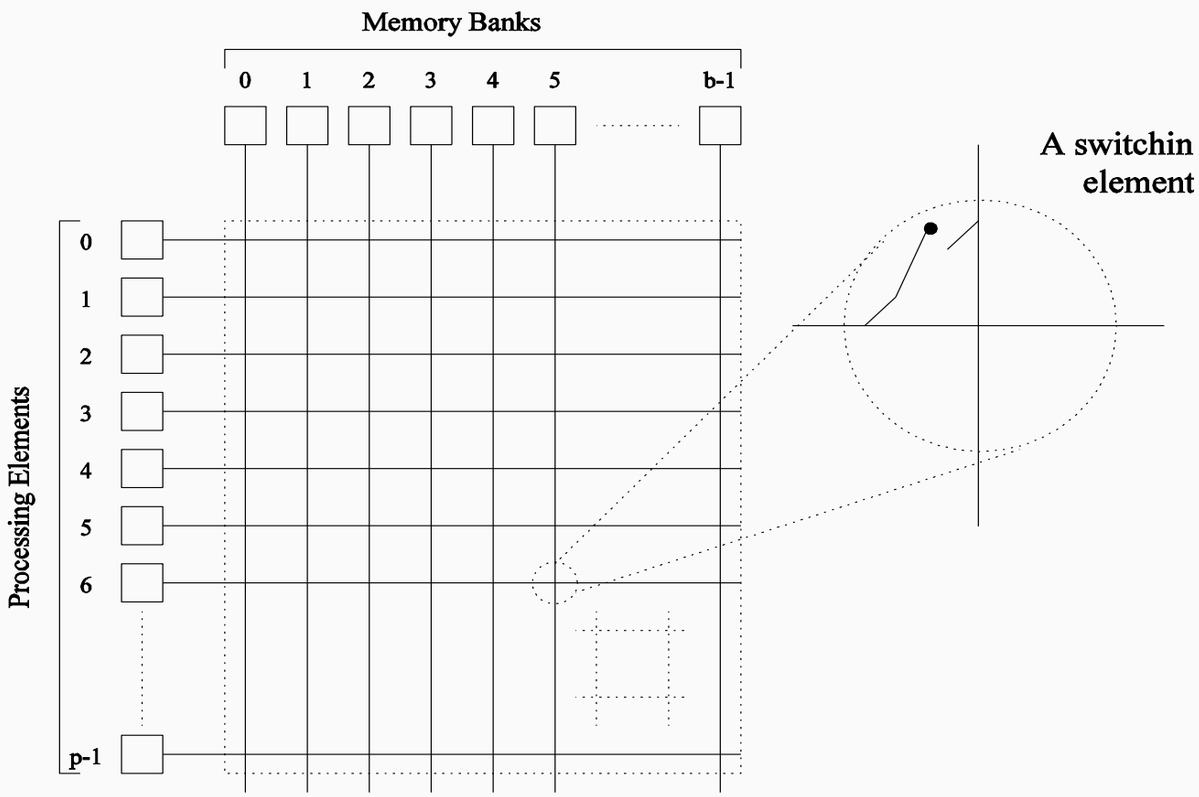
Some of the simplest and earliest parallel machines used buses.

All processors access a common bus for exchanging data.

The distance between any two nodes is *O(1)* in a bus. The bus also provides a convenient broadcast media.

However, the bandwidth of the shared bus is a major bottleneck ☹

Network Topologies: Crossbars



Grama et. al.

A crossbar network uses an $p \times m$ grid of switches to connect p inputs to m outputs in a non-blocking manner.

A completely non-blocking crossbar network connecting p processors to b memory banks.

Network Topologies: Crossbars

The cost of a crossbar of p processors grows as $O(p^2)$.

This is generally difficult to scale for large values of p.

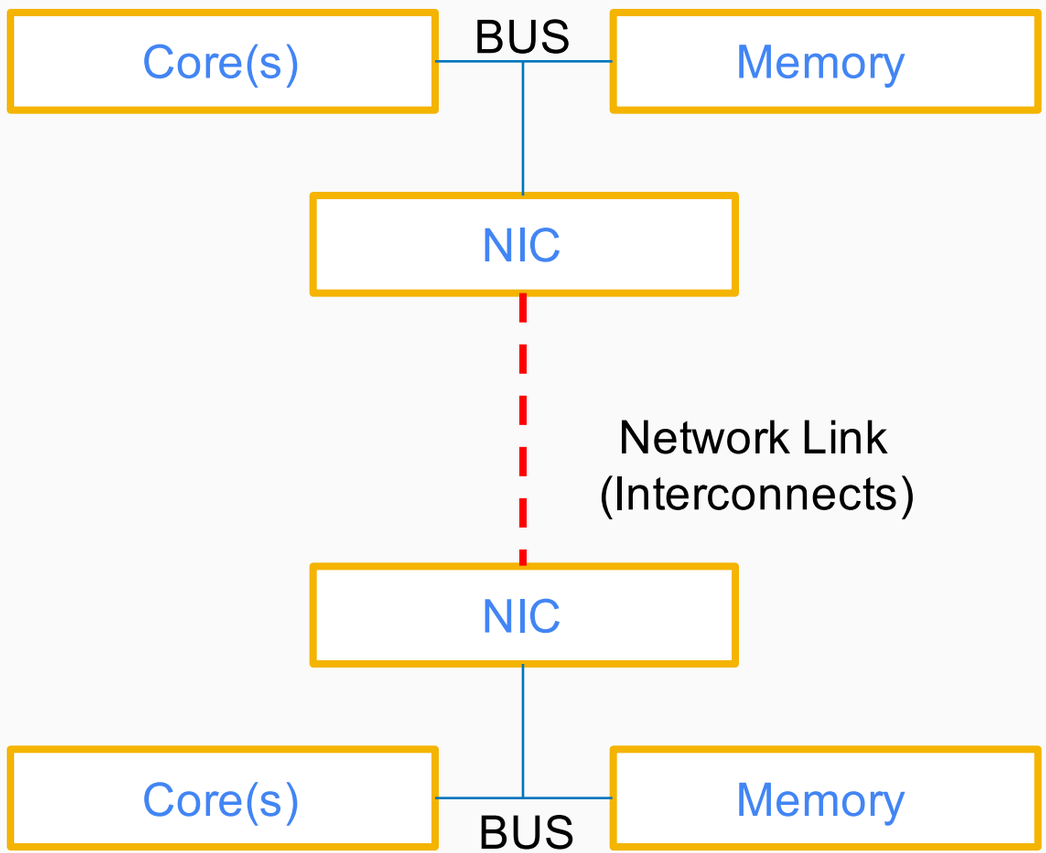
Examples of machines that employ crossbars include NEC Earth Simulator (640 x 640)

Network Topologies: Multistage Networks

Crossbars have excellent performance scalability but poor cost scalability.

Buses have excellent cost scalability, but poor performance scalability.

Multistage interconnects strike a compromise between these extremes.



Topology

Networks

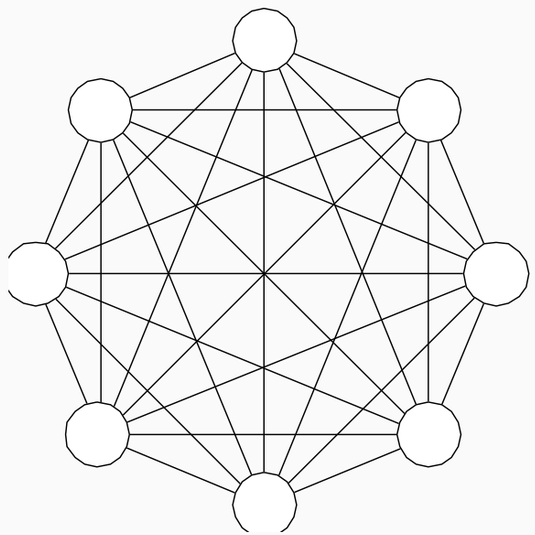
Network Topologies: Completely Connected

Each processor is connected to every other processor.

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.

In this sense, these networks are static counterparts of crossbars.



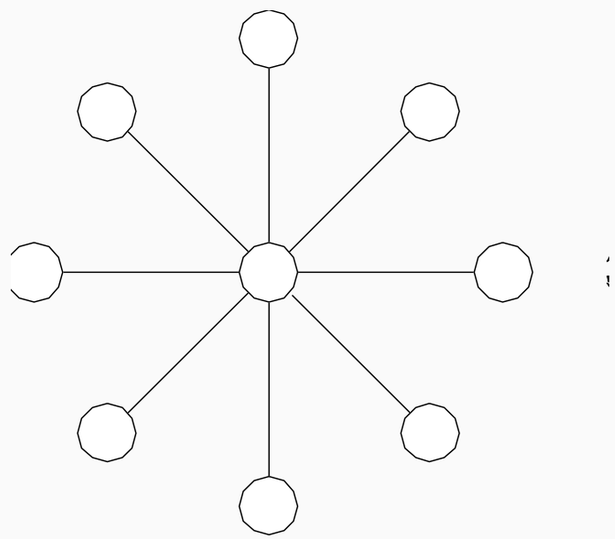
Grama et. al.

Network Topologies: Star Connected Network

Every node is connected only to a common node at the center.

Distance between any pair of nodes is *O(1)*. However, the central node becomes a bottleneck.

In this sense, star connected networks are static counterparts of buses.



Grama et. al.

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

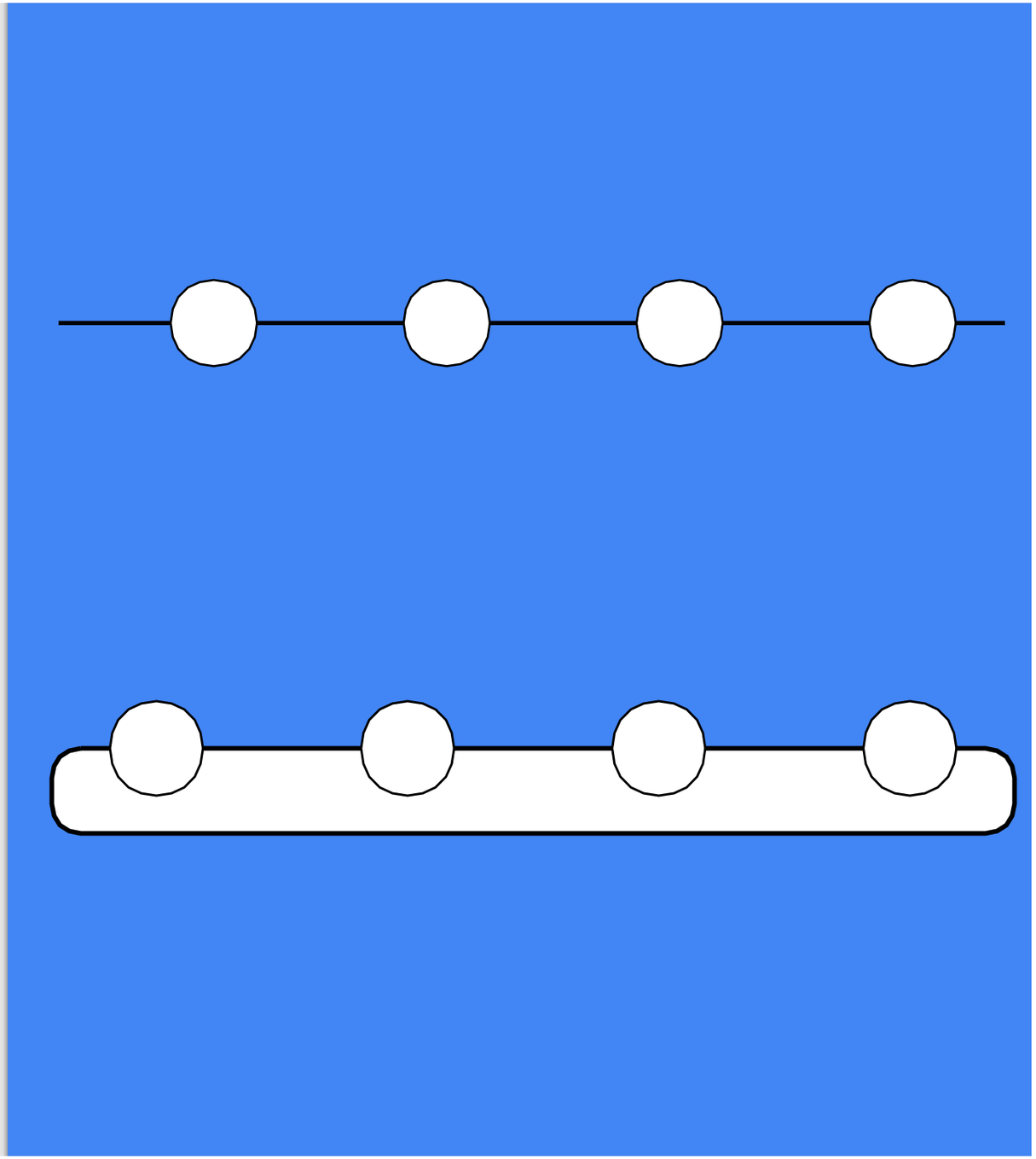
In a linear array, each node has two neighbors, one to its left and one to its right. If the nodes at either end are connected, we refer to it as a 1-D torus or a ring.

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.

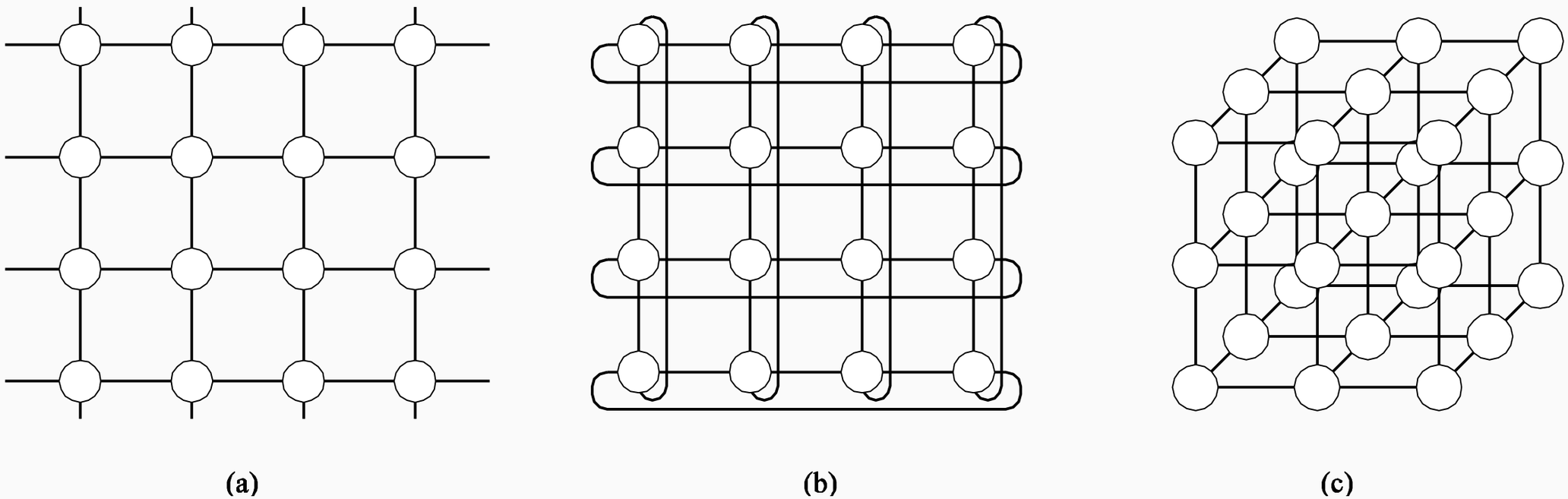
A special case of a *d*-dimensional mesh is a hypercube. Here, $d = \log p$, where p is the total number of nodes.

Linear Arrays and Rings



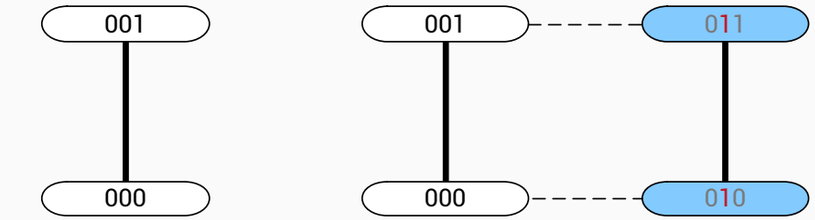
Grama et. al.

Network Topologies:
Two- and Three Dimensional Meshes



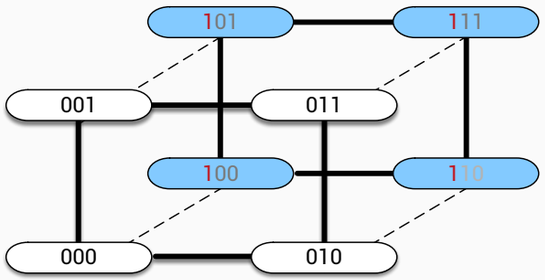
Two and three dimensional meshes: (a) 2-D mesh;
(b) 2-D torus; and (c) a 3-D mesh.

Grama et. al.

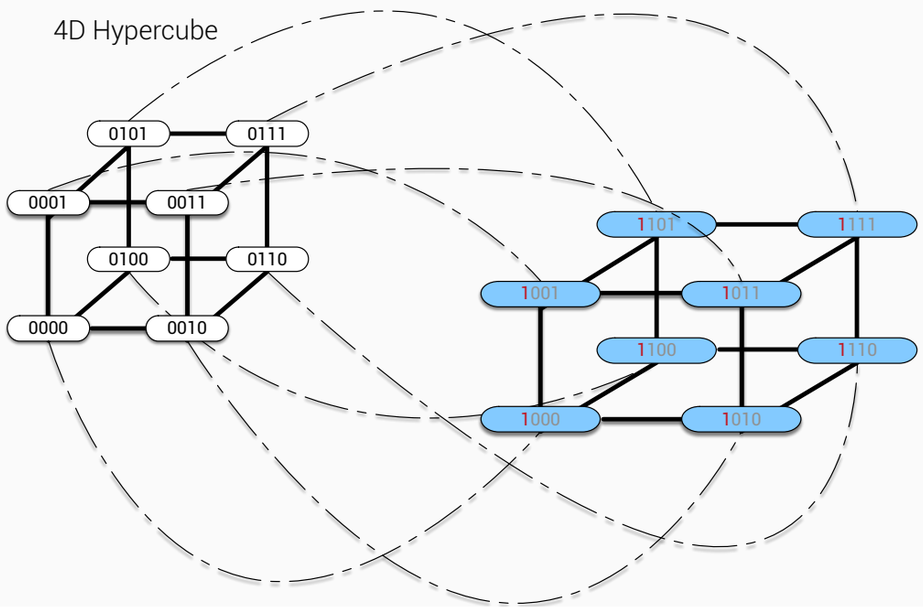


1D Hypercube

2D Hypercube



3D Hypercube



Network Topologies: Hypercubes and their Construction

Construction of hypercubes from hypercubes of lower dimension.

Network Topologies: Properties of Hypercubes

The distance between any two nodes is at most *log p*.

LOTS!!! Of Bandwidth

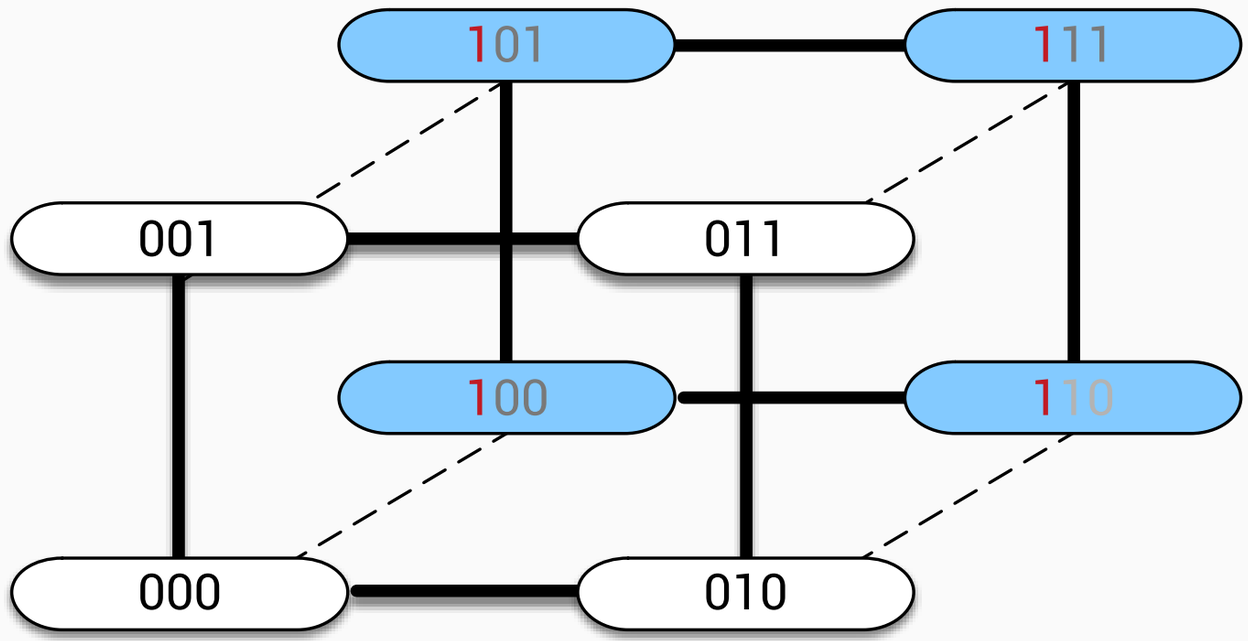
Low contention

Each node has *log p* neighbors.

Hypercube Mapping

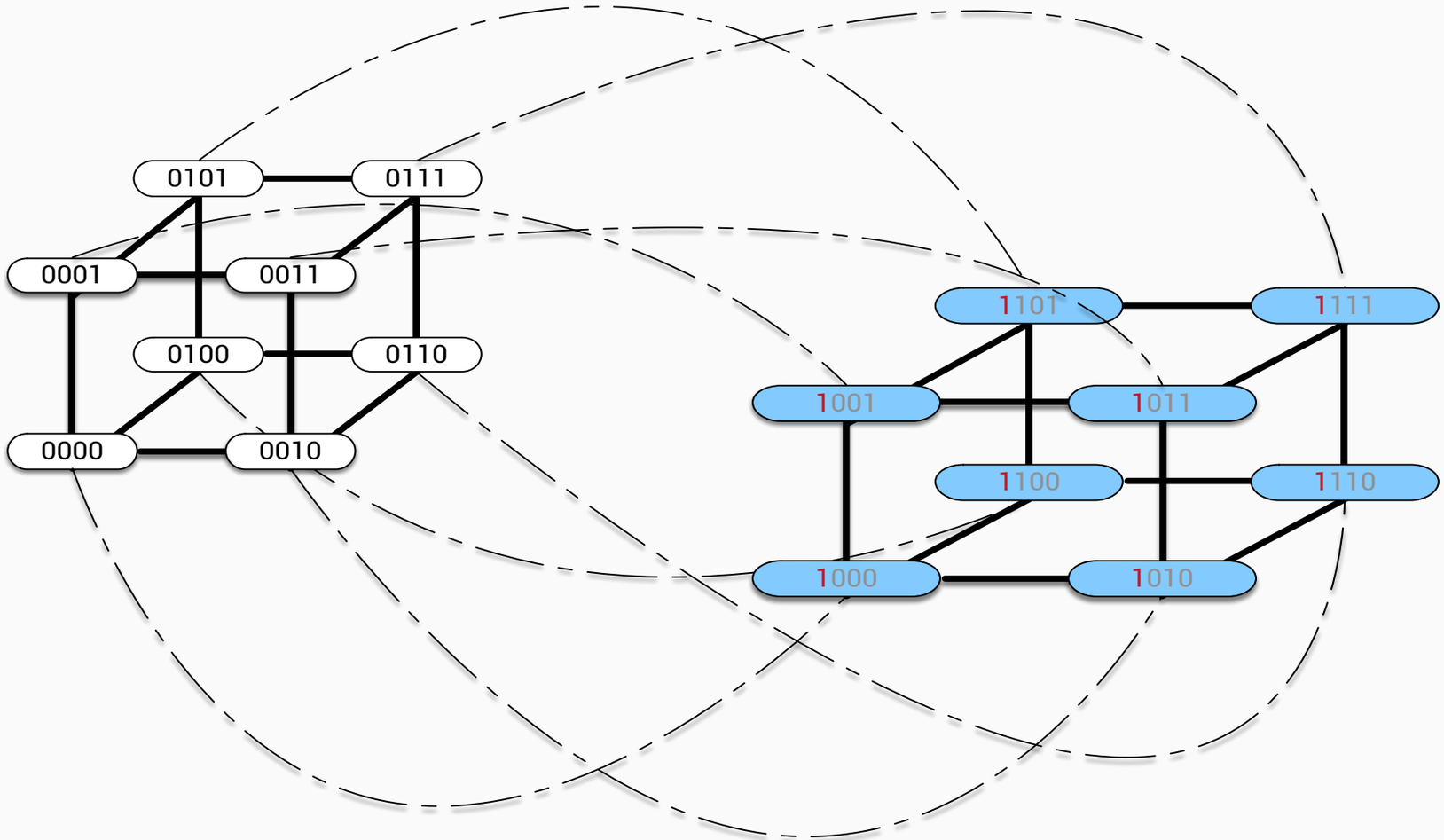
```
  0      |      1

 00      |     01
 10      |     11
```

The distance between two nodes is given by the number of bit positions at which the two nodes differ.



The dotted lines show the new dimension. So in this case, it shows how one would go from 2D → 3D.
The text in Red shows the bit that was flipped when moving from 2D space to 3D.

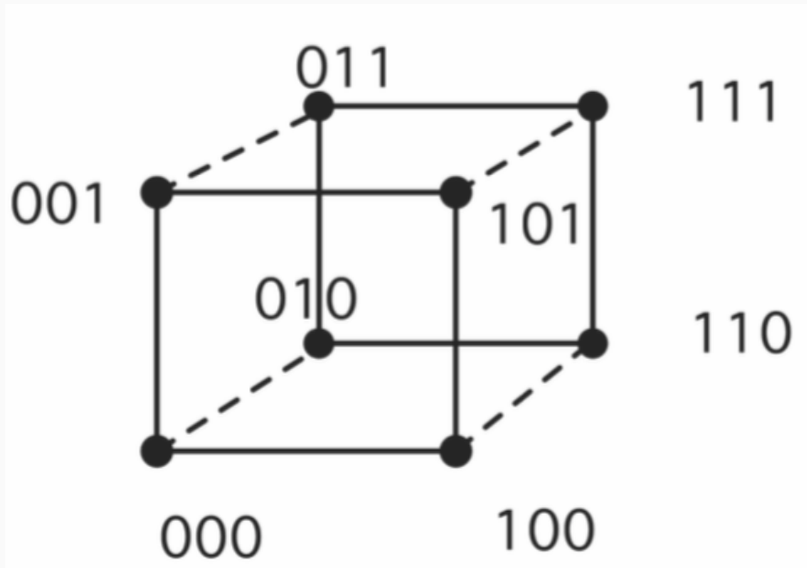
Hypercube Mapping (4D)



The dotted lines show the new dimension. So in this case, it shows how one would go from 3D → 4D.
The text in Red shows the bit that was flipped when moving from 3D space to 4D.

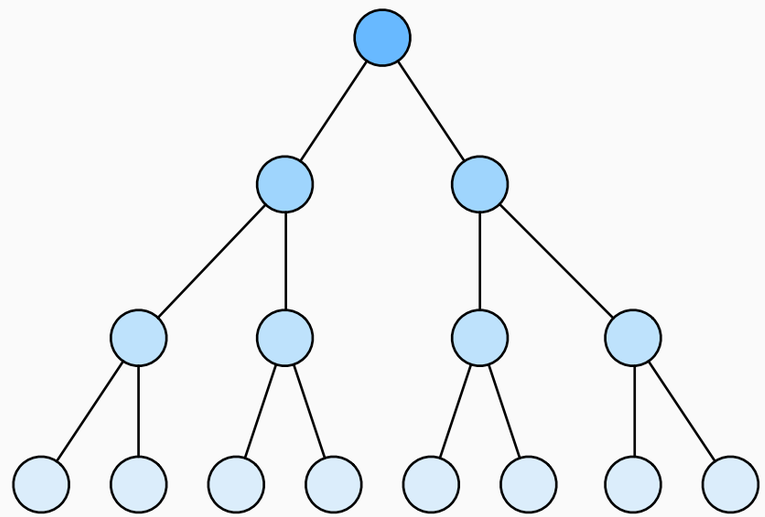
Hypercube Mapping

1. What is the distance between 0 & 1?
2. 3 & 4??
3. Max distance?



Network Topologies: Tree Networks

- 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.



Network Topologies: Fat Trees

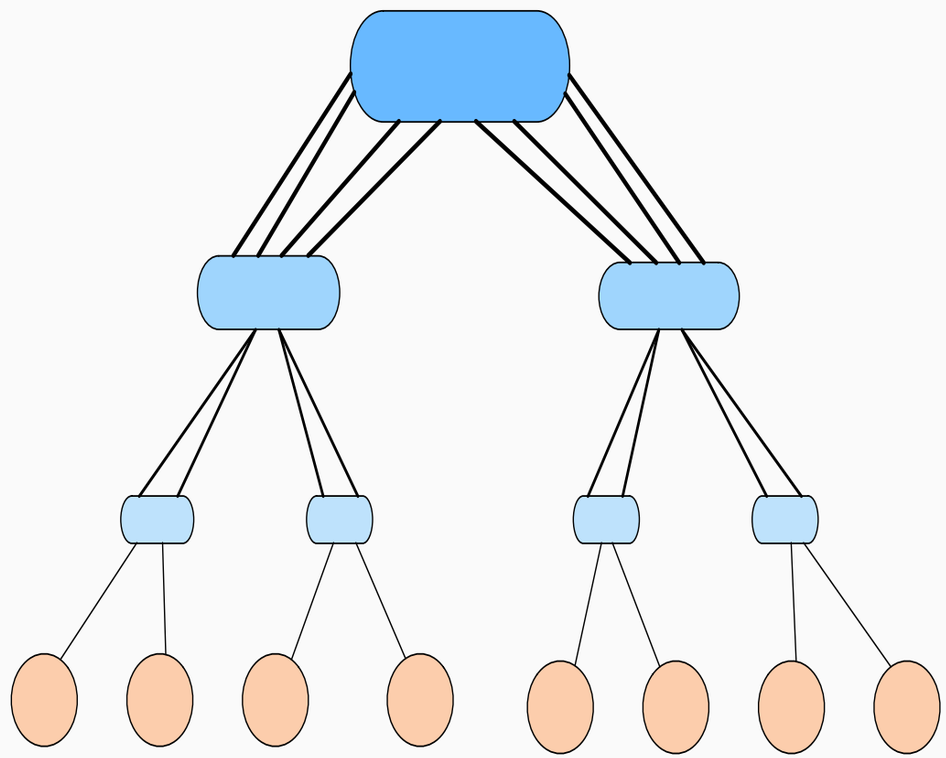
Problems with trees?

Multiple Switches

Each Level has the same number of incoming and outgoing links

As you go up a tree, the number of links increases

Latency increases as you go up a tree



Terminology

Degree → How many links/node? Pros and Cons?

Diameter → Max distance between any two nodes

Distance → shortest path between two nodes

Dia & Dist for:

- Completely Connected
- Star
- Ring
- Hypercube

Connectivity

How robust is the connection (i.e., how many nodes do you need to remove in order to sever the connection)

Do you want high/low connectivity? Why?

Connectivity of:

Linear Array, Star, Ring, Mesh?

Latency and Bandwidth??

Latency: How long does it take to start sending a "message"? Units are generally usec.

Bandwidth: What data rate can be sustained once the message is started?

Units are GBps

Both point-to-point and aggregate bandwidth are of interest

Multiple Connections (Wires) → Latency (same/different)? Bandwidth?

Evaluating Static Interconnection Networks

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*, Ring? Hypercube? mesh is $\sqrt{p}$, 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

Next Week

Communication Costs
Merging Networks →
Best of all involved
Basic Communication Operations

Student Presentation