Parallel Systems

Prof. James L. Frankel
Harvard University

Version of 6:50 PM 4-Dec-2018

Architectures

- SISD (Single Instruction, Single Data)
- MIMD
- SIMD
  - GPU is an instance of SIMD
- (MISD – not used)

- SPMD (Single Program, Multiple Data)
- MPMD

Access to Memory

- Time to access memory can dominate a system's performance – or lack of it
- How data is laid out across processor is very important

- Local memory per processor
  - All ***remote*** *memory* is accessed through a network
- No local memory
  - *All memory* is accessed through a network
- Of course, caching is always important
- NUMA – Non-Uniform Memory Access

Shared Memory

- Does the system provide a synchronization primitive across processors (an atomic read-modify-write cycle)?
- Can shared memory be cached?
- Are snoopy caches supported?

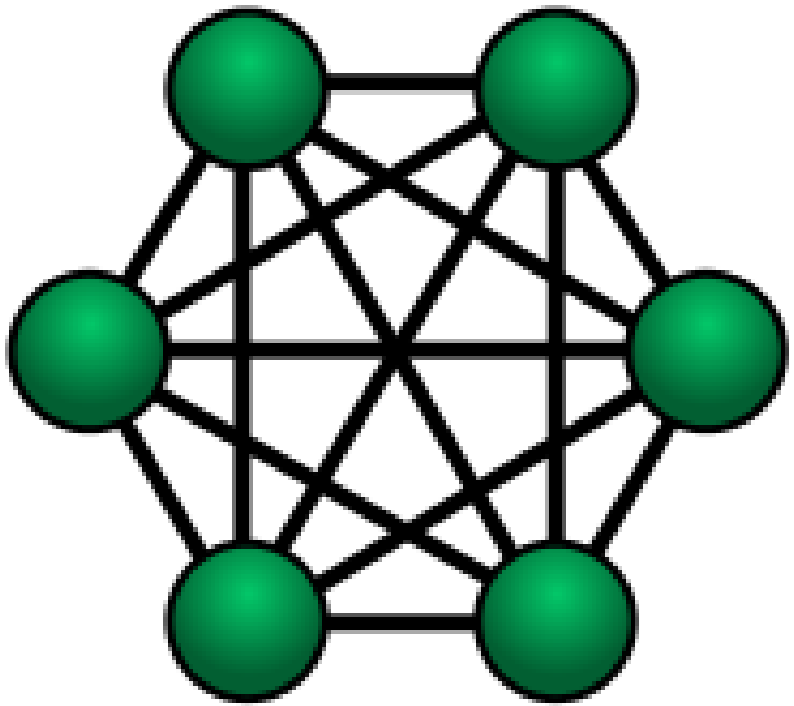
Network Architecture: Attributes

- Number of communication paths
- Number of switching points
- Width of each communication path (Single-bit wide vs. multi-bit bit wide)
- Is parallel communication allowed (*i.e.,* can more than one message propagate through the network concurrently)?
- Network diameter – the maximum distance between any two communicating nodes
- Worm-hole routing
- Are intermediate switches autonomous or is there a control unit? Are switches message-switched or circuit-switched?
- How is the destination addressed?
- Does the speed of accessing remote memory differ by location (NUMA)?
- How many nodes can the network support? Is the limitation determined by cost or size or other feasibility?
- Are memory and processing co-located (peer-to-peer) or separated?
- Do intermediate network switching nodes have the ability to perform computation?
- Do messages flow in one direction through the network or can responses travel backwards?

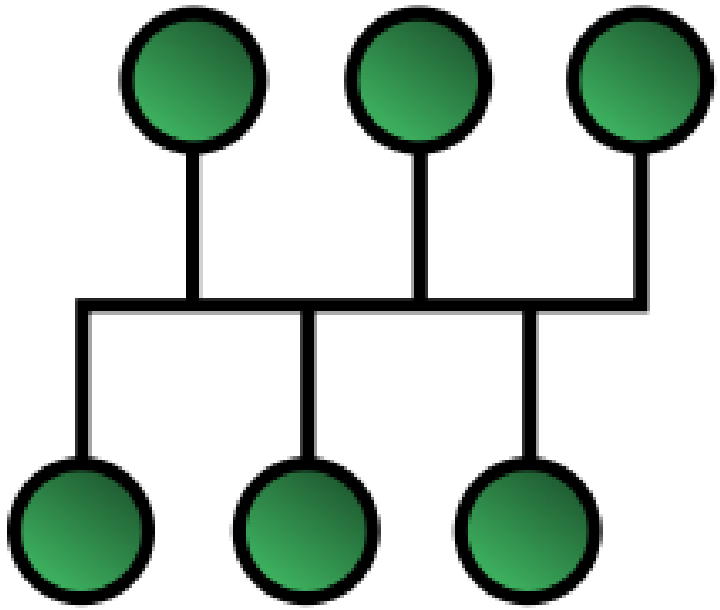
Network Architecture: Examples

- Arbitrary
- Fully-connected
- Bus
  - Includes network of workstations (NOW) over Ethernet
  - Electrical limit on the number of nodes that can be connected to a single bus
    - Also limited because of conflicts accessing a single bus
- Token Ring
- Two-dimensional grid (mesh) with possible wrapping at edges (torus)
- Three-dimensional grid with possible wrapping at edges (torus)
- Crossbar
- Omega Network
- Butterfly Network
- Tree
- Fat Tree
- Hypercube

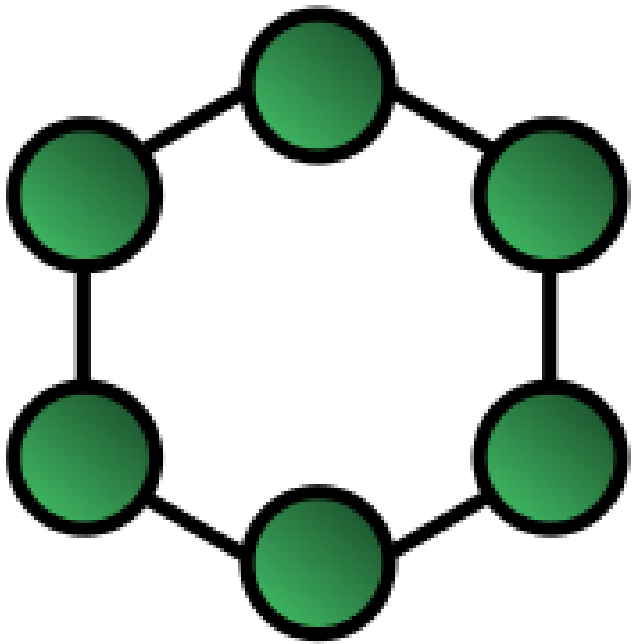
Fully-Connected (sometimes referred to as a Mesh)



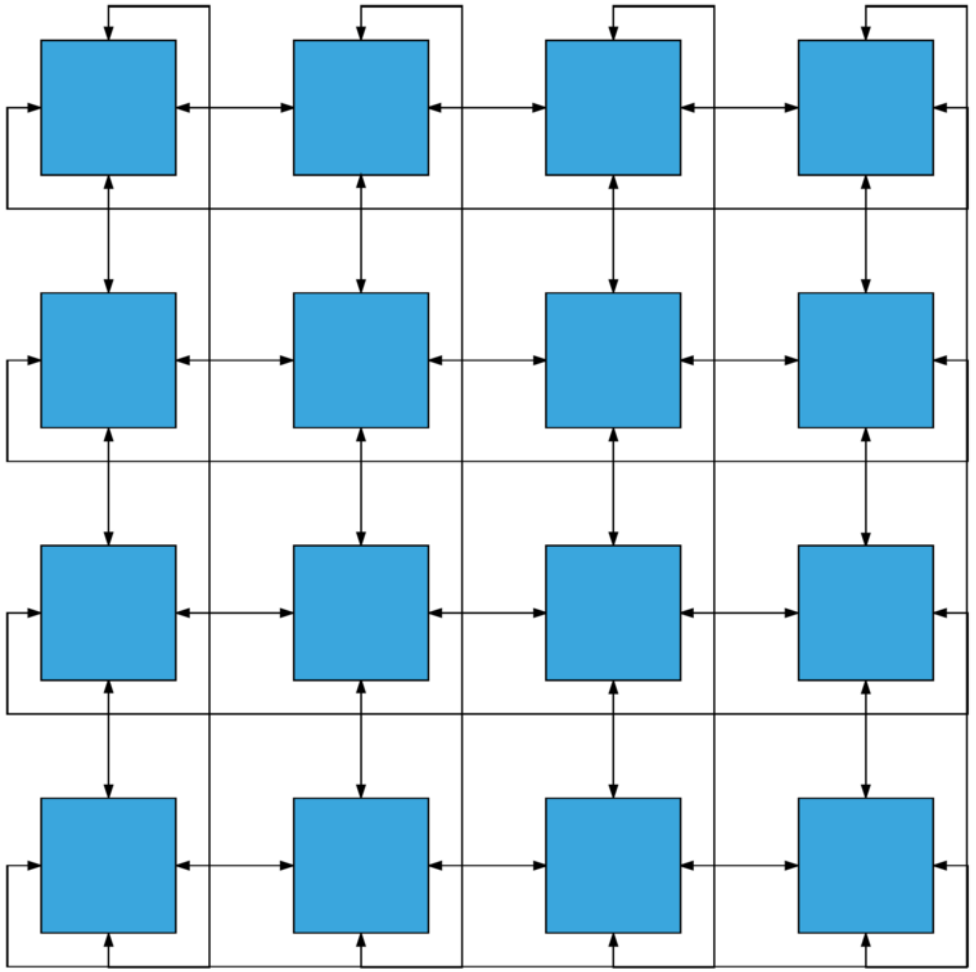
Bus (including Thicknet & Thinnet Ethernet)



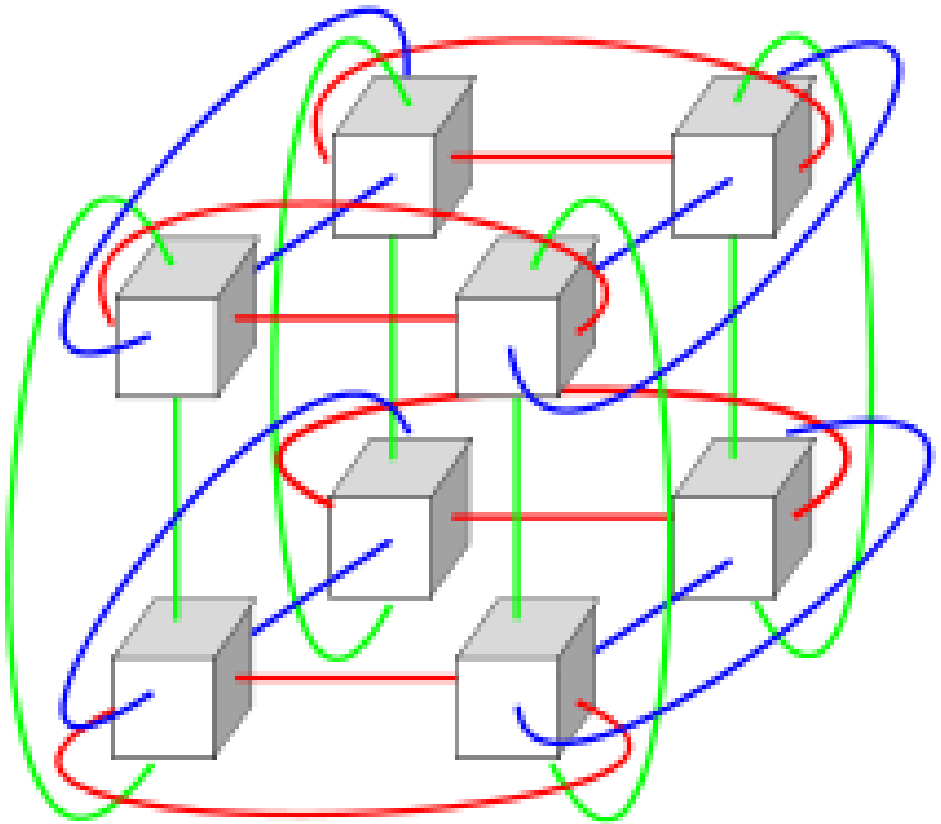
Token Ring



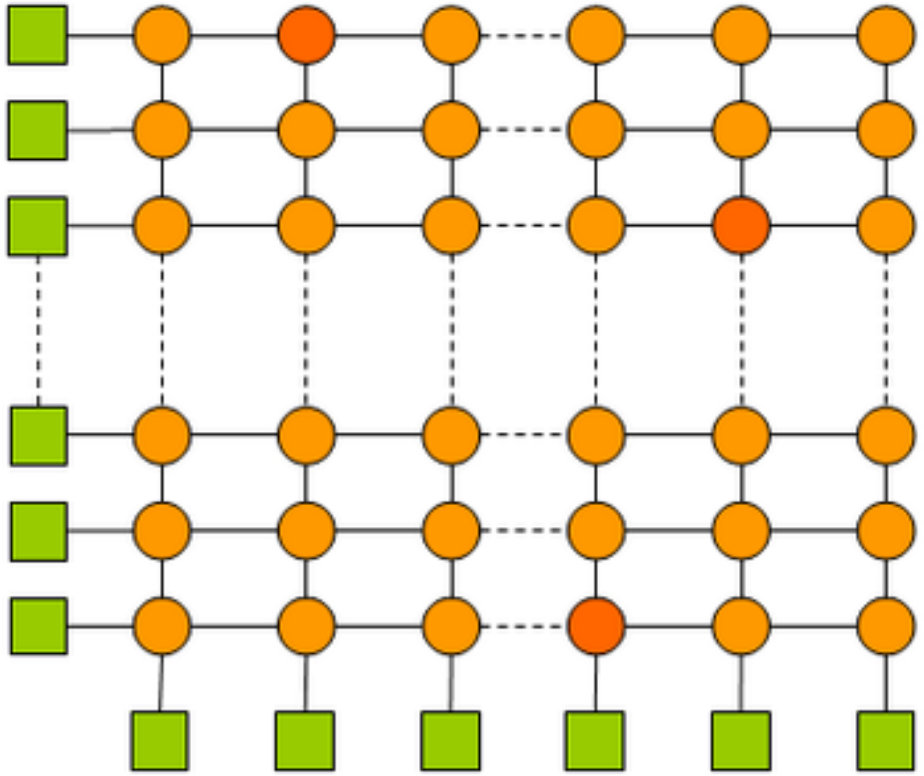
Two-Dimensional Grid



Three-Dimensional Grid



Crossbar



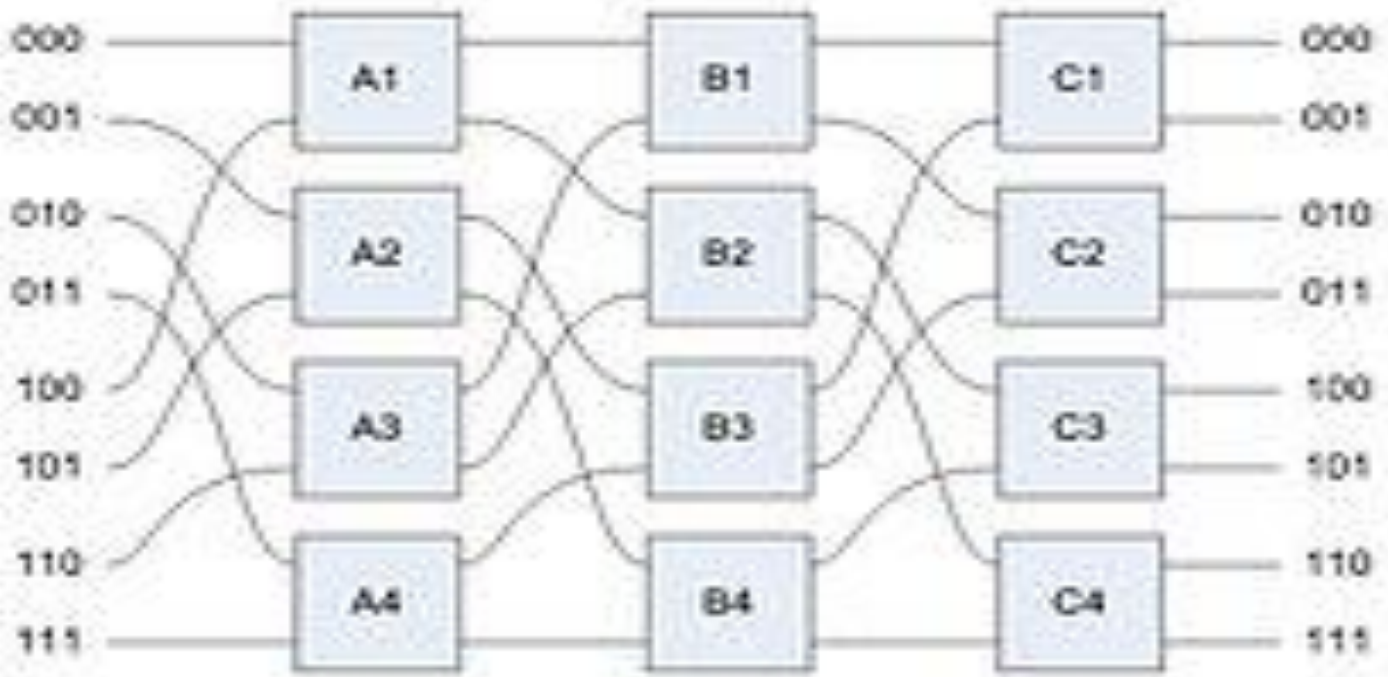
Multistage Network

- A message may pass through more than one switching node
- Reduces the number of switches required
- Originally designed for the PSTN (Public Switched Telephone Network) by Bell Labs

- The Omega Network is an example of a multistage network
  - Has the same interconnection pattern at each stage
  - Interconnection pattern is based on a perfect shuffle network

- The Butterfly Network is another example of a multistage network

Omega Network



Omega Network Routing

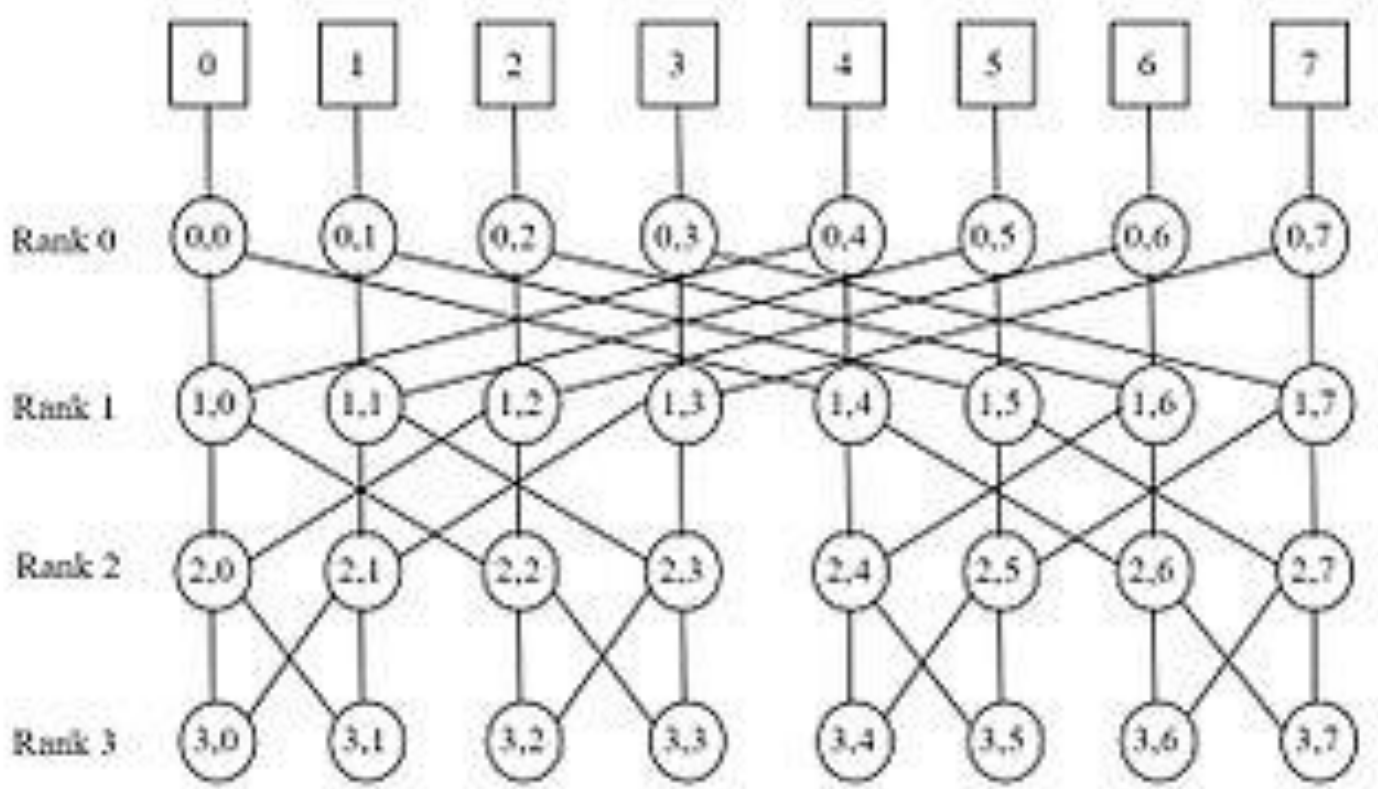
- Destination-tag routing
  - Forwards message through a switch based on a destination address bit
    - 0 means upper output
    - 1 means lower output
- XOR-tag routing
  - Sets each switch to be either pass-through or swapped
    - For upper input,
      - 0 means pass-through
      - 1 means swapped
    - For lower input,
      - 0 means swapped
      - 1 means pass-through
- Example shows a 2x2 switch, but higher degrees are possible

Analysis of Omega Network

- For 2x2 switches and *n* sources and *n* destinations (for *n* a power of two),
  - Number of switches per stage = $n/2$
  - Number of stages = $\log_2 n$
  - Total number of switches = $\frac{n}{2} \log_2 n$ = O($n \log n$)

- For *m* x *m* switches (for *m* a power of two) and *n* sources and *n* destinations (for *n* a power of two and for *n* a multiple of *m*),
  - Number of switches per stage = n/m
  - Number of stages = $\log_m n$
  - Total number of switches = $\frac{n}{m} \log_m n$ = O($n \log n$) for *n* >> *m*

Butterfly Network



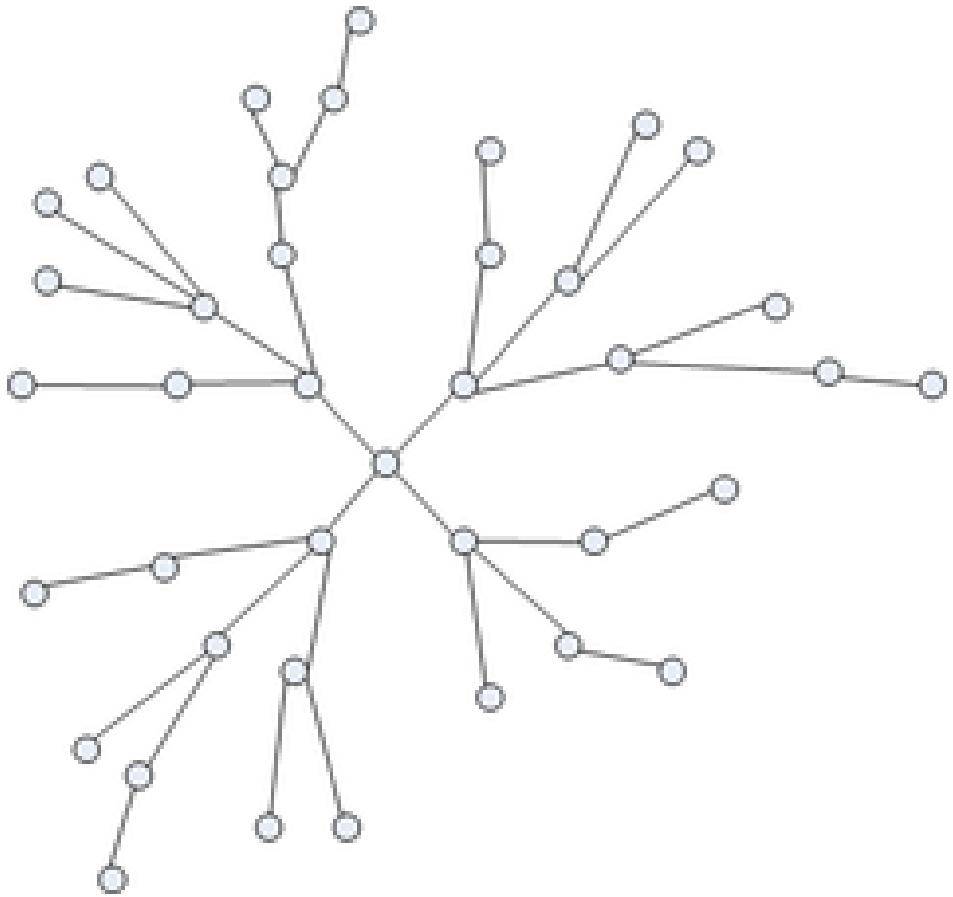
Butterfly Network Routing

- Each stage selects a sub-section of that portion of the network
  - Allows a large Butterfly Network to be built because of this successive stage decomposition
- Does ***not*** have the same interconnection pattern at each stage
- Example shows a 2x2 switch, but higher degrees are possible

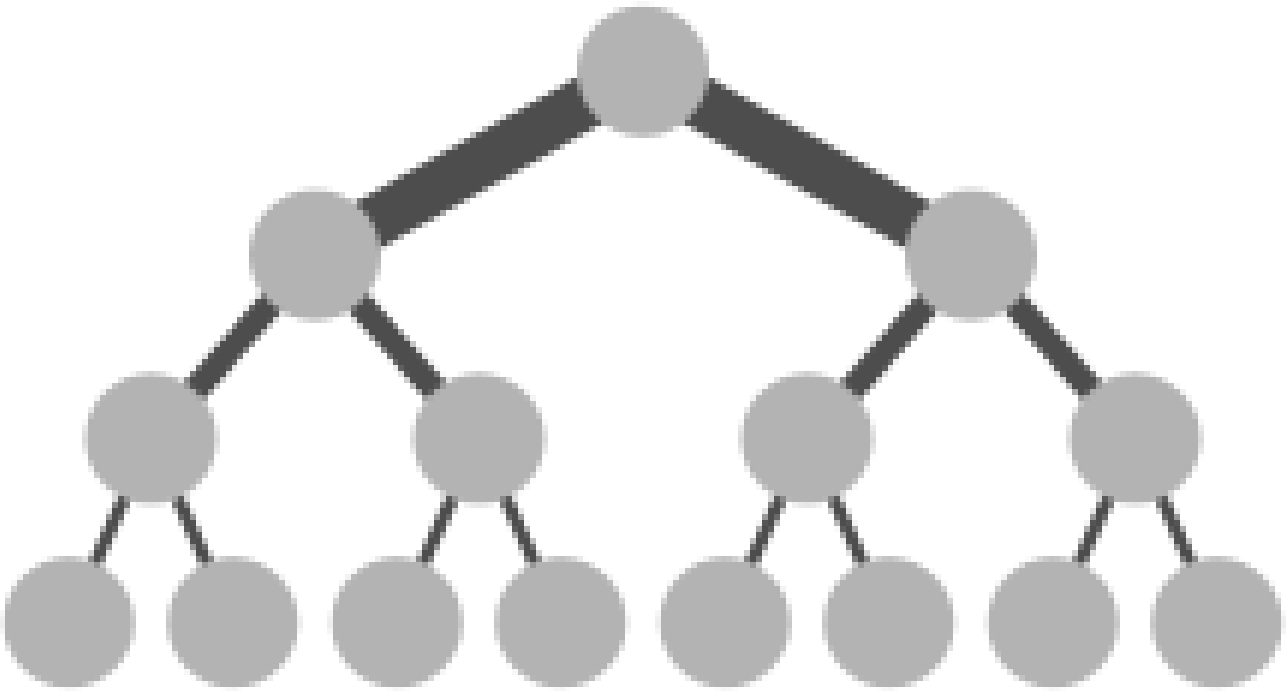
Analysis of Butterfly Network

- For 2x2 switches and *n* sources and *n* destinations (for *n* a power of two),
  - Number of switches per stage = n
  - Number of stages = $1 + \log_2 n$
    - First and last stages are composed of half switches
  - Total number of switches = n $(1 + \log_2 n)$ = O$(n \log n)$

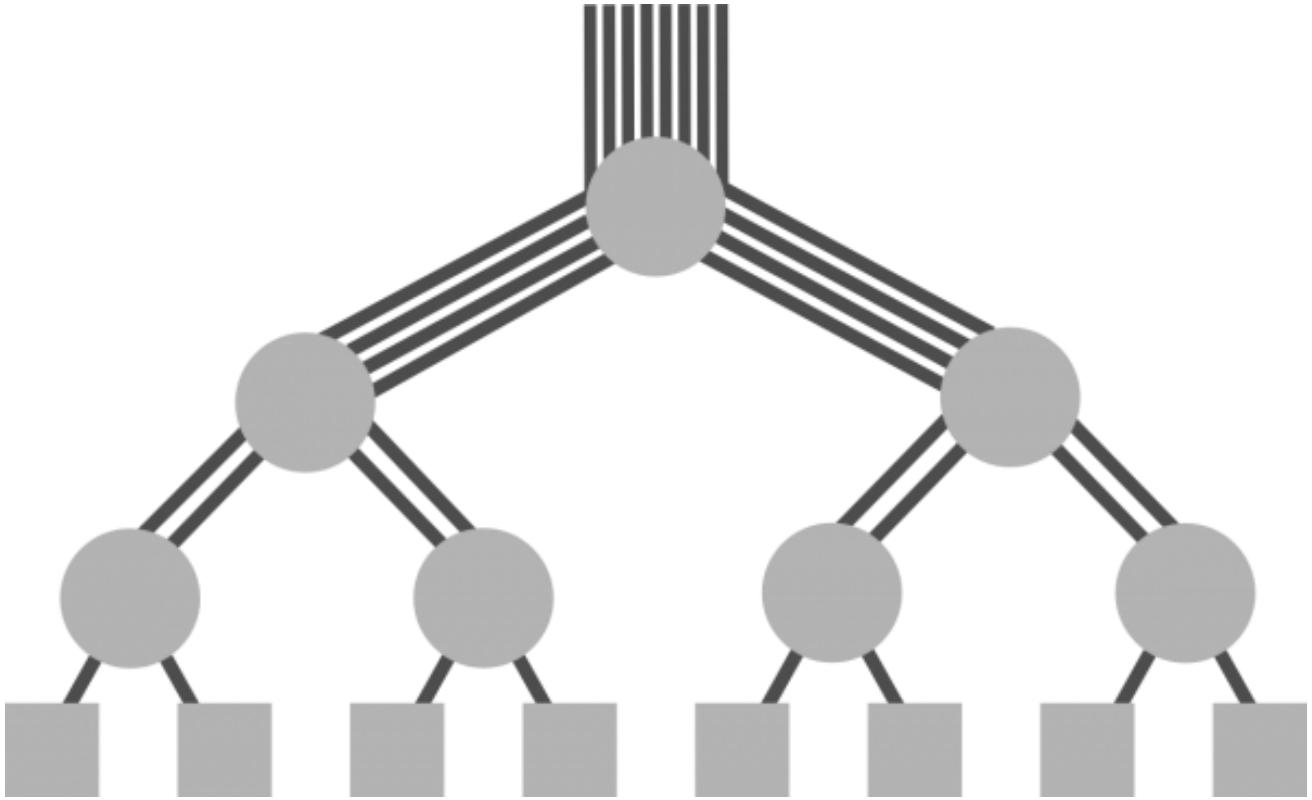
Tree



Fat Tree



Fat Tree Showing Multiple Data Paths Higher in the Tree

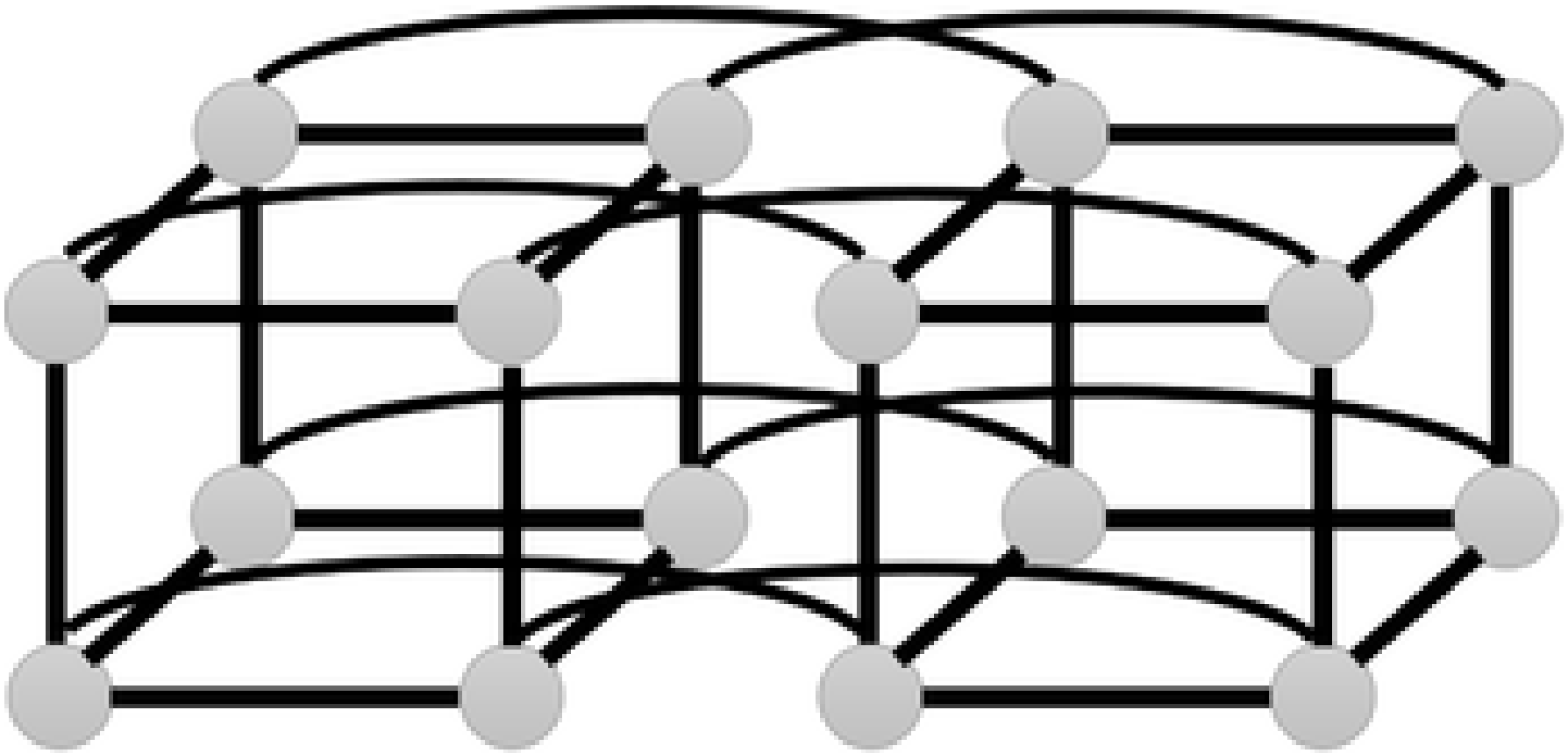


Fat Tree Analysis

- A Fat Tree alleviates the congestion that would occur at higher levels in the tree if messages are sent to uniformly distributed destinations
- A Fat Tree allows the network to be segmented among multiple subset trees

- If *viewed from the side*, a fat tree becomes bushier as you approach the root
  - Each message flows upward (*i.e.,* toward the root) only as far as necessary – that is, to the first node that it has in common with its destination
  - As messages are sent up the tree (*i.e.,* toward the root), a choice is made between multiple paths to alleviate congestion
  - Messages are routed to the destination as the message is sent down the tree (*i.e.,* toward the leaves)

Hypercube



Data parallelism

- C*

Communication

Effect on Operating Systems

- Gang scheduling
  - Communicating processes on different processors need to be running for best performance – and, since performance is the reason to run on a multiprocessor system, this is important
- Memory allocation
- Inter-processor communication
  - *n*-dimensional Nearest neighbor
  - Arbitrary
    - send
    - get
  - Reductions
  - Scans