COMP 633 - Parallel Computing

Lecture 20
November 4, 2021

Interconnection Networks

- Reading
  - Kumar et al., Basic Communication Operations
- PA2
  - Please choose your project by this Friday

Topics

- Interconnection networks for parallel processors
  - components
  - characteristics
  - network models
- Analysis of networks
  - diameter
  - bisection bandwidth
  - degree
  - cost
  - example networks
- Simple cost measures for communication
  - store-and-forward model
  - cut-through model

Kinds of networks

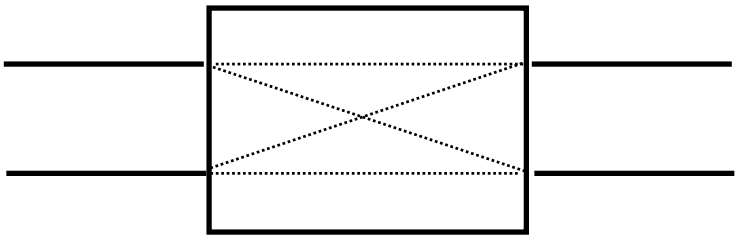
- Wide-area networks (WAN)
  - ~~telephone,~~ internet
- Local-area networks (LAN)
  - ethernet, wireless 802.11x
- System-level networks
  - processor to processor
  - (processor to memory)

These networks differ in sclability, assumptions, cost

  - Primary focus in this course is system-level networks

Components of a network

- clusters
  - each processor has a dedicated network interface
- switches
  - k inputs, m outputs, $m \geq k$
    - simplest: k = m = 2



- links
  - characteristic bandwidth

    (# parallel bits per link) • (signaling rate)

Four characteristics of networks

- Network topology
  - physical interconnection structure of network
    - analogy: Roadmap showing interstates
- Routing algorithm
  - rules that specify which routes a message may follow
    - analogy: To go from Durham to DC, take I-85N to I-95N to I-495
- Switching Strategy
  - determines how a message traverses a route
    - analogy: Presidential convoy reserves entire route in advance, while a group of travelers in separate cars make individual switching decisions
- Flow control
  - determines when a message makes progress
    - analogy: Traffic signals and rules: two cars cannot occupy the same location at the same time

Network topology

- Connected undirected graph G = (N, C)
  - N = set of nodes
  - C = set of channels (bidirectional links)

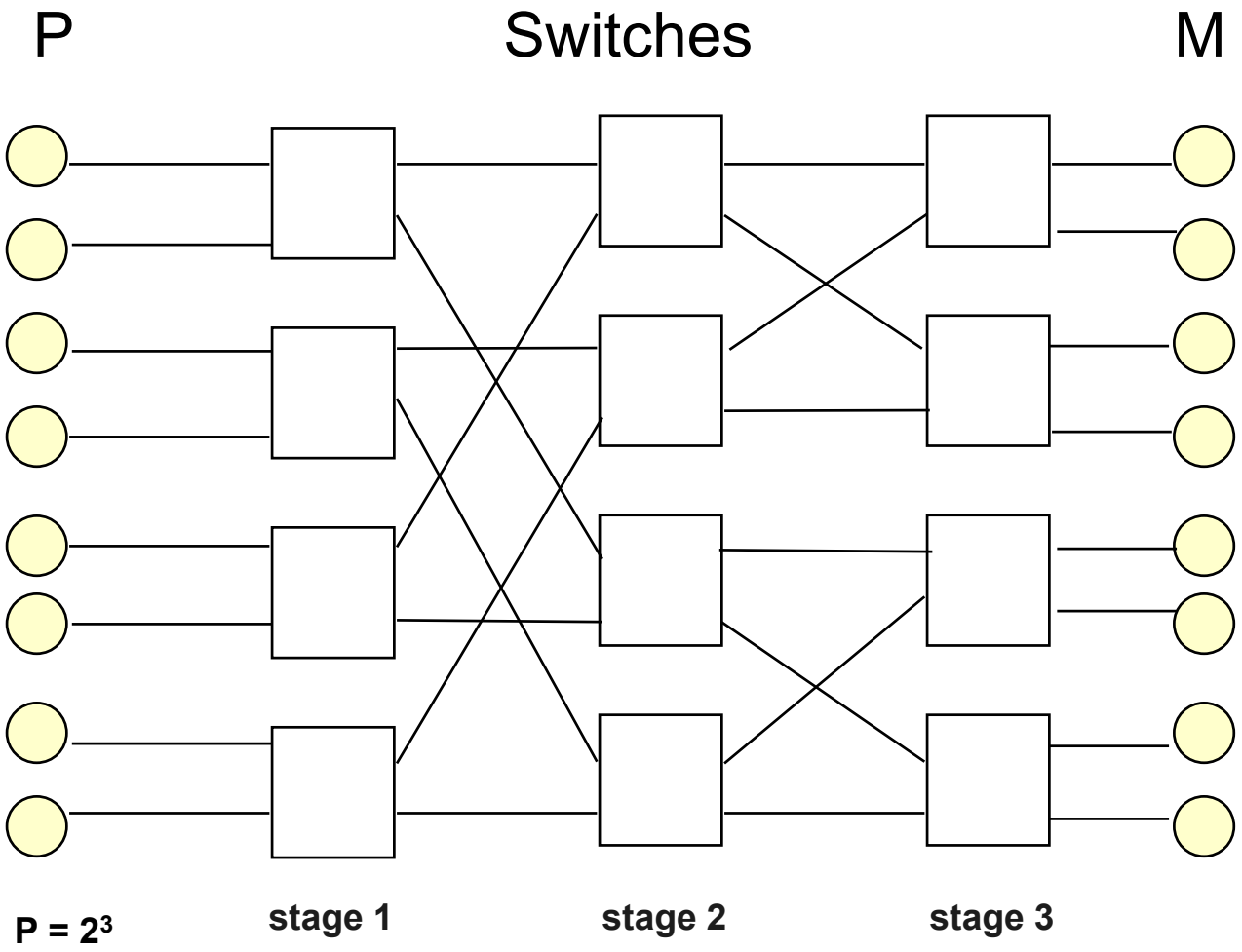
- Indirect network (switching fabric)
  - contains switch nodes without an attached processor or memory
  - switching nodes do not generate traffic
  - typical case in modern networks

- Direct network
  - every node can be a producer and/or consumer of messages
  - no pure switching nodes

Indirect networks

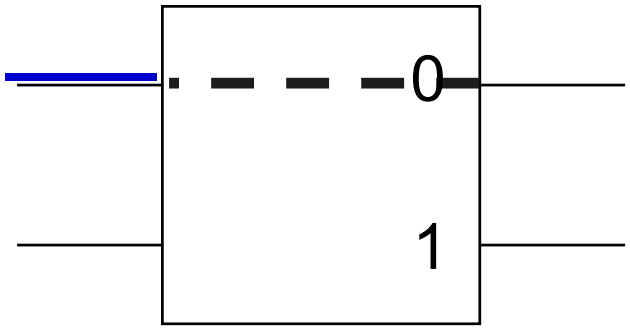
- Processor to memory interconnect in shared-memory machines
- Connect p processors to p memory banks
  - Example: bus
    - $\Theta(p)$ switches
    - simultaneous references always serialize
  - Example: crossbar
    - $\Theta(p^2)$ switches
    - simultaneous references in disjoint banks serviced in parallel
  - Example: multistage network
    - $\Theta(p \lg p)$ switches and links
      - $\Theta(\lg p)$ stages of $\Theta(p)$ switches each
    - simultaneous reference of disjoint memories may be serialized
      - contention within the network

Multistage Butterfly indirect network ($p = 8$)

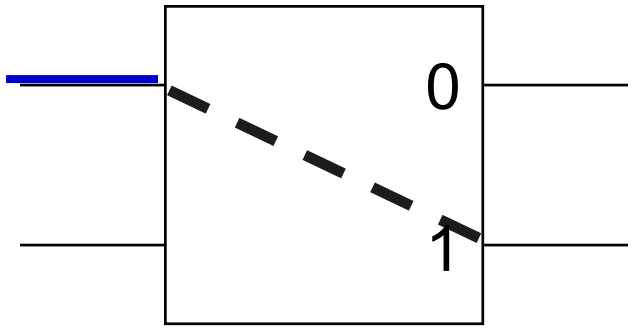


Routing in butterfly networks

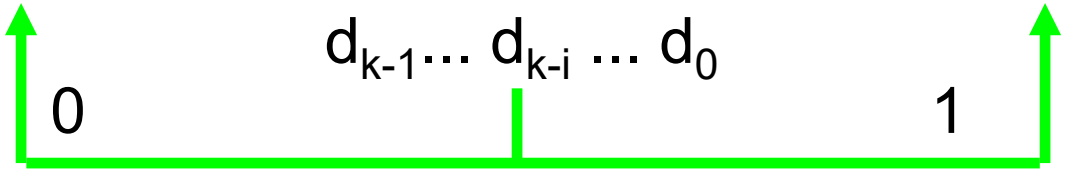
- based on destination address
  - destination address d_{k-1} ….. d_0
  - in stage i, switch setting is determined by d_{k-i}
    - switch to top or bottom



Switch to top

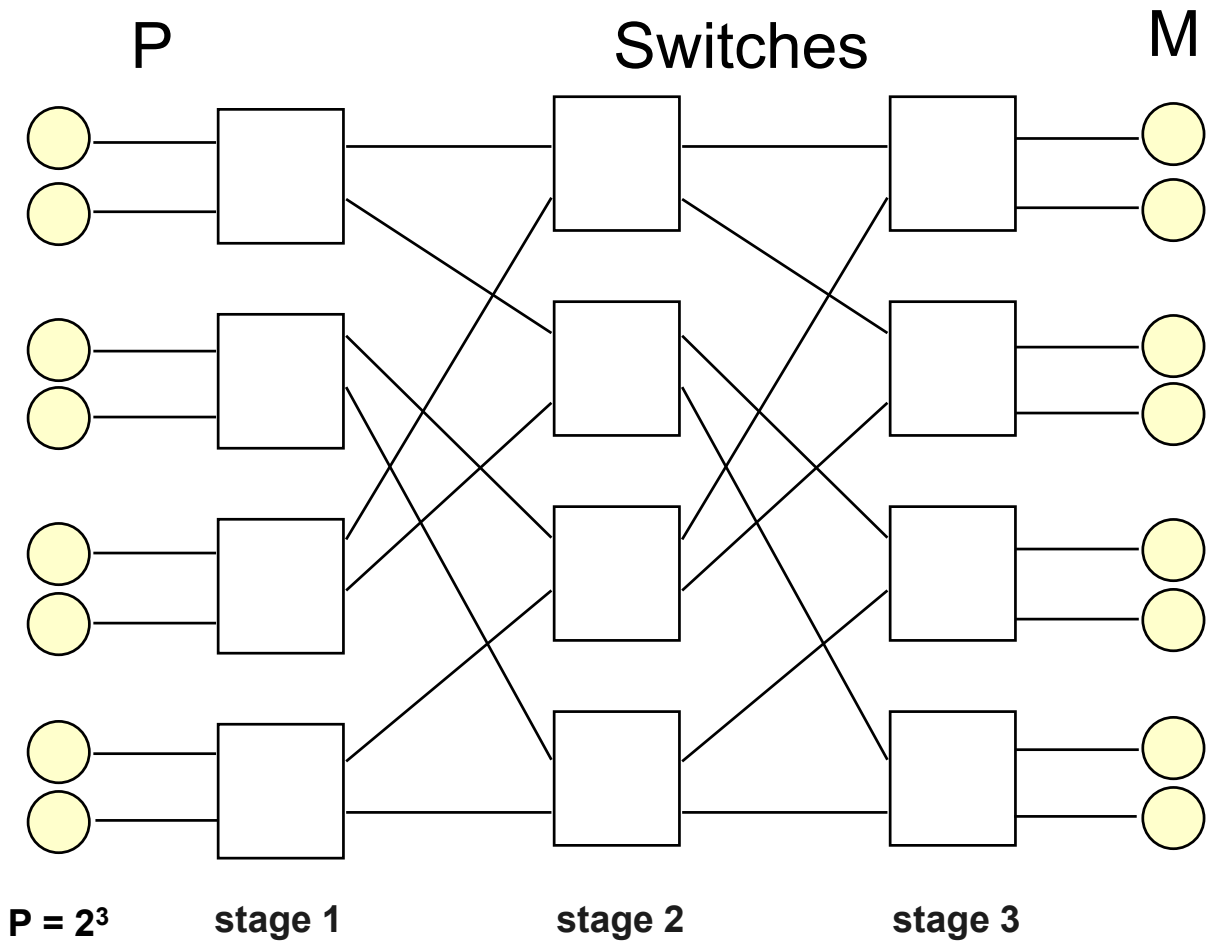


Switch to bottom



Multistage Omega network (p = 8)

- Isomorphic to butterfly network
  - same "perfect shuffle" connection pattern between successive stages



Network Topology: Graph-theoretic measures

- Diameter: Maximum length of shortest path between any pair of nodes

$$\max_{u,v \in N} \left(\min_{u \to v \in C^*} |u \to v| \right)$$

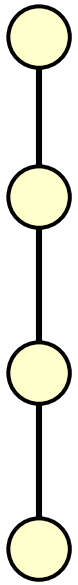
  - i.e. distance between maximally separated nodes - related to latency

- Bisection width: Minimum number of edges crossing approximately equal bipartition of nodes
  - related to bandwidth with full applied load
  - a *scalable* network has bisection width $\Omega(p)$

- Degree: number of edges (links) per node (switch)
  - related to cost and switch complexity
  - fixed degree is simpler and more scalable

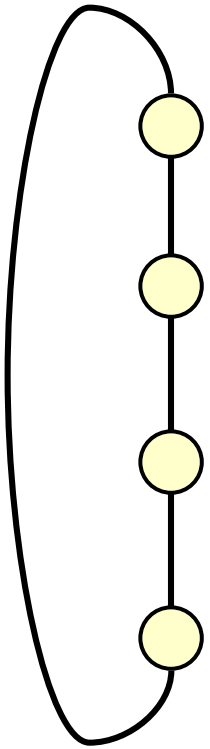
- Cost: number of wires
  - length of wires and wiring regularity is also an issue

Linear array



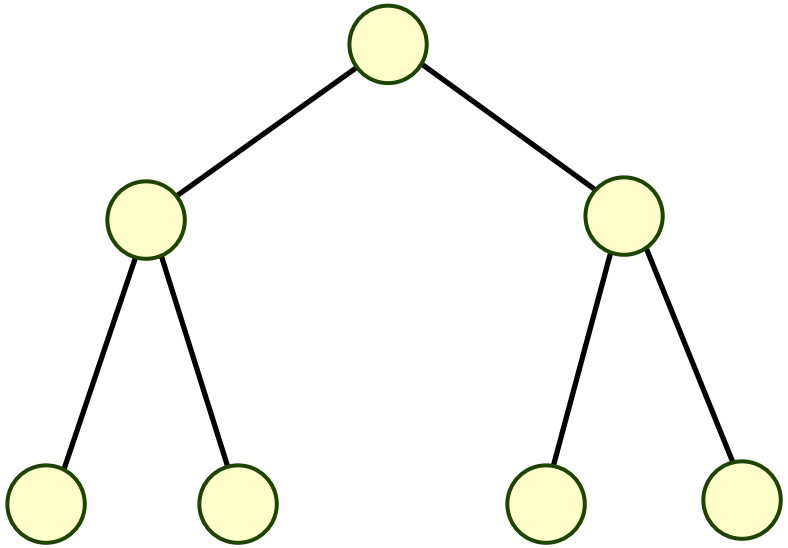
- $|C| = p-1$
- Diameter = p-1
- Degree ≤ 2
- Bisection width = 1

Ring



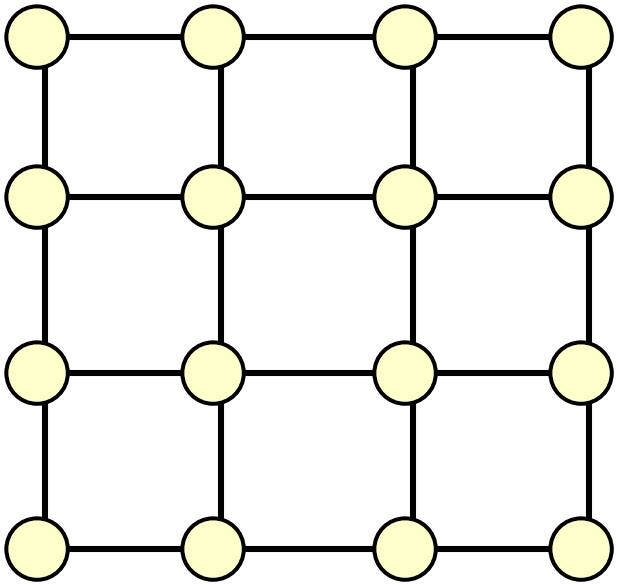
- |C| = p
- Diameter = p/2
- Degree = 2
- Bisection width = 2

Binary Tree



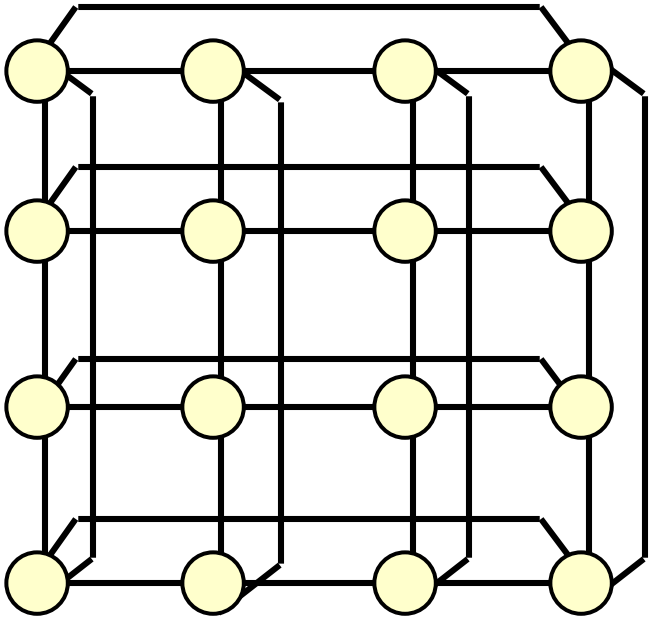
- $|C| = p - 1$
- Diameter = $2 \lg p$
- Degree ≤ 3
- Bisection width = 1

d-dimensional mesh



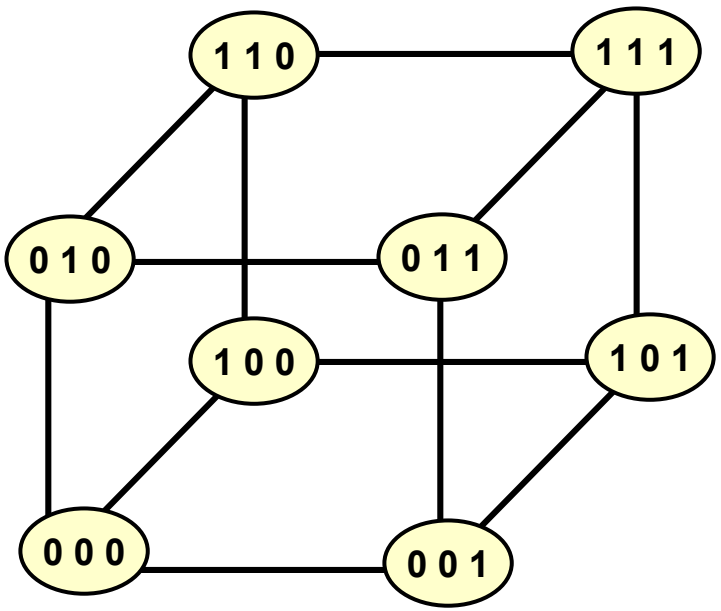
- $p = k^d$
  - Cartesian product of d linear arrays with $k = p^{1/d}$ nodes each
- $|C| < 2dp$
  - short wires when d ≤ 3
- Diameter = $dp^{1/d}$
- d ≤ Degree ≤ 2d
- Bisection width = $p^{(1-1/d)}$
  - 2-D mesh, $d = 2$ $\sqrt{p} \times \sqrt{p}$

k-ary *d*-cubes



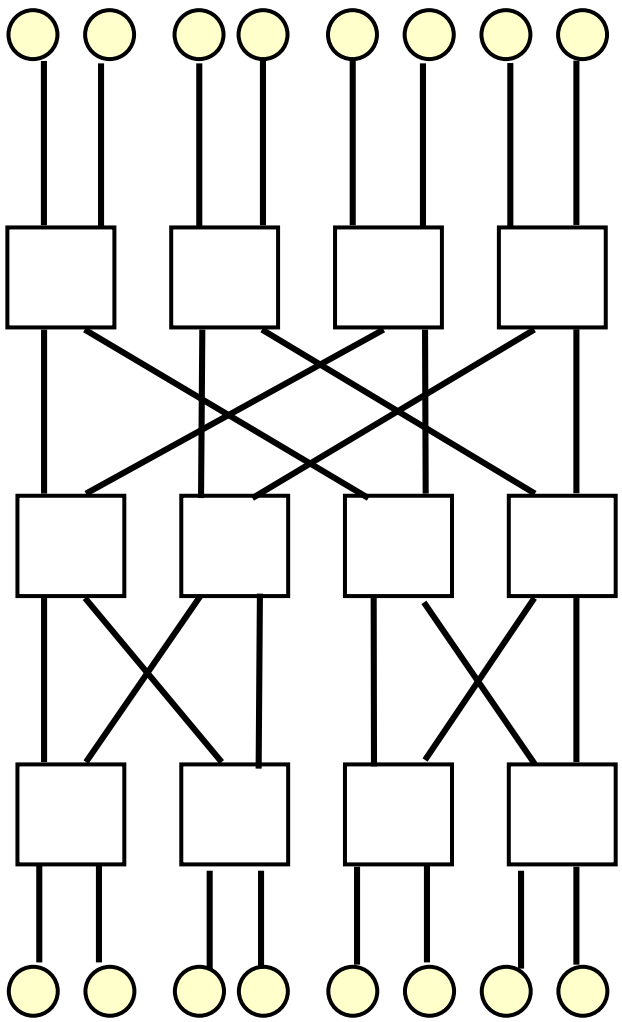
- $p = k^d$
  - Cartesian product of *d* rings with k = $p^{1/d}$ nodes each
- $|C| = 2dp = 2dk^d$
- Diameter = $dp^{1/d} / 2$
- Degree = $2d$
- Bisection width = $2\ p^{(1-1/d)} = 2k^{d-1}$
  - Ring: p-ary 1-cube
  - 2-D Torus: $\sqrt{p}-\text{ary}\quad 2-\text{cube}$
  - 3-D Torus: $\sqrt[3]{p}-\text{ary}\quad 3-\text{cube}$
  - Hypercube: 2-ary (lg p)-cube

(Boolean) Hypercube



- $|C| = p \lg p$
- Diameter = $\lg p$
- Degree = $\lg p$
- Bisection width = $\Theta(p)$

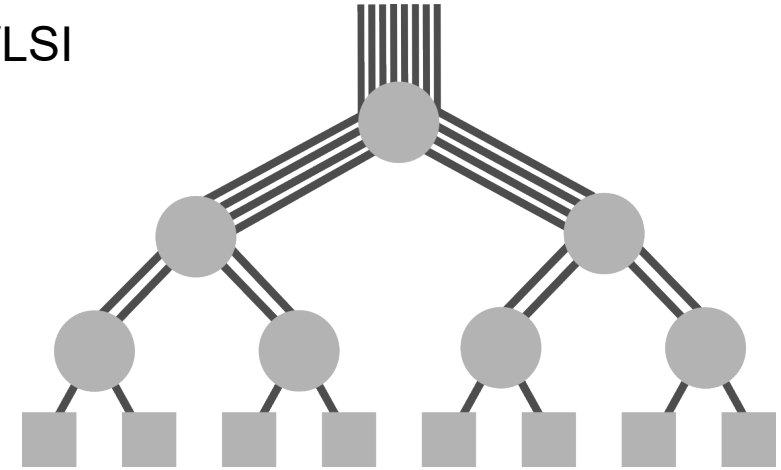
Butterfly (Indirect)



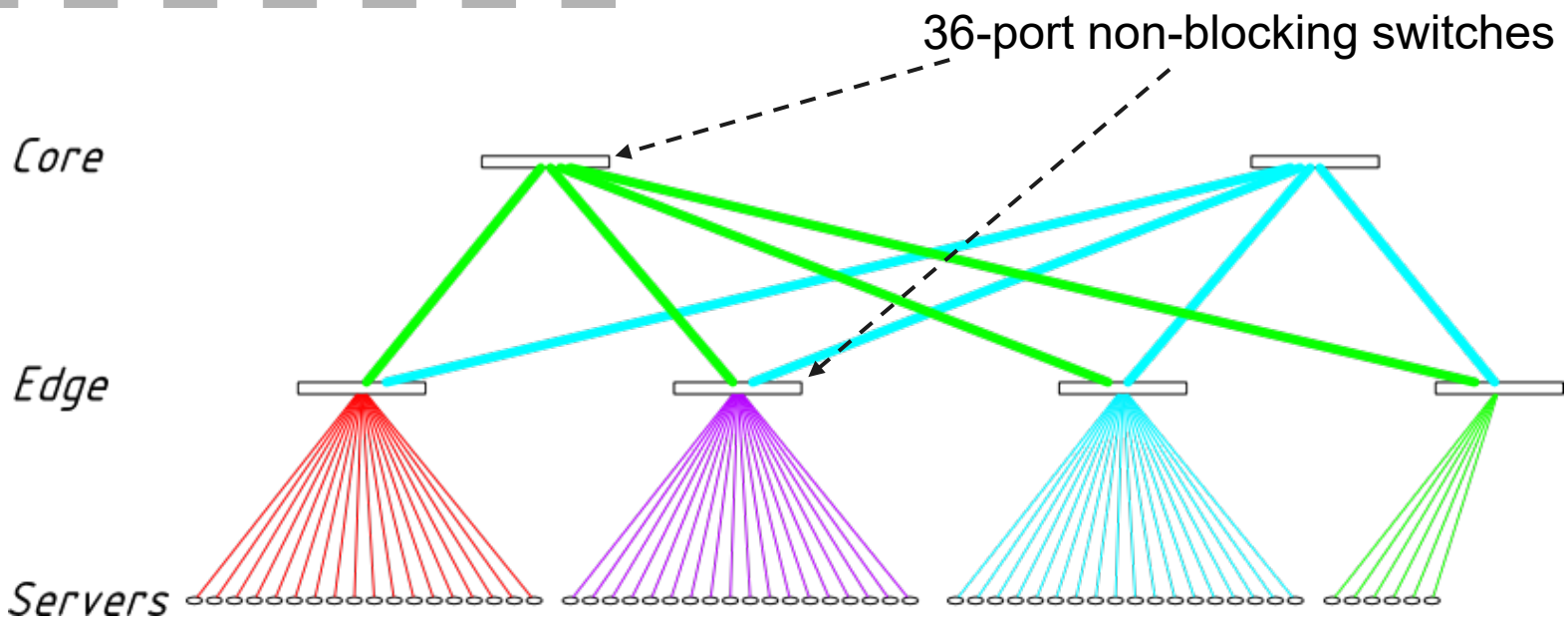
- $|C| = p \lg p$
- Diameter = $\lg p$
- Degree = 2
- "Bisection" width (congestion)
  - There are some bad permutations $\Theta(p^{1/2})$
  - Overwhelming majority have bisection of $\Theta(p)$

Fat-tree (Indirect)

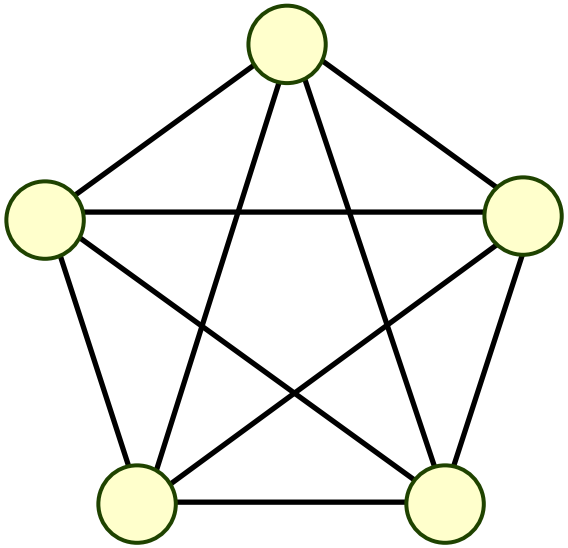
VLSI

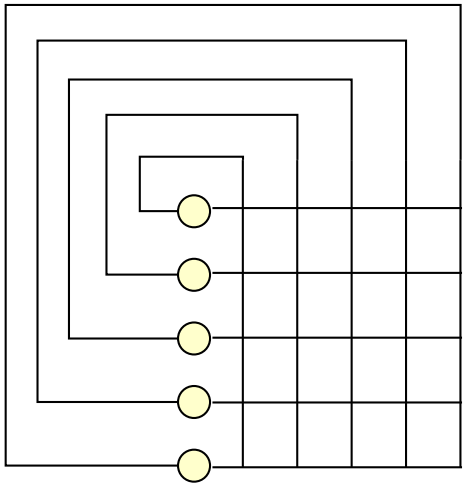


- |C| = p lg p
- Diameter = 2 lg p
- Degree = varying (2^i i ε 0..lg p)
- Bisection width = Θ(p)



Crossbar





- Complete graph on p nodes
- |C| = p(p-1)/2
- Diameter = 1
- Degree = p-1
- Bisection width = $p^2/4$

Networks in current parallel computers

- Modern interconnects are indirect
  - Hardware routing between source and destination

- Indirect networks
  - Cluster of commodity nodes
    - Fat-tree (assembled using 36 port non-blocking switches)
  - IBM Summit (ORNL)
    - Fat-tree Infiniband [4,608 nodes] (24,000 GPU, 202,752 cores)
  - Fujitsu Fugaku
    - 6D torus [160,000 nodes k-ary d-cube, ? k~7 d=6] (3M+ cores)

- Processor – memory interconnects (p procs, m memories)
  - Tera MTA
    - 3D torus (p = 256, m = 4,096)
  - NEC SX-9
    - crossbar (p = 16 procs * 16 channels/proc = 256, m = 8,192)

Routing and flow control

- System-level networks
  - Tradeoffs are very different than WAN (TCP)
    - use flow control instead of dropping packets
    - mostly static routing instead of dynamic routing
  - Routing algorithm
    - prescribes a unique path from source to destination
      - e.g. dimension ordered routing on hypercube and lower dimensional d-cubes
      - some networks dynamically “misroute” if a needed link is unavailable
    - routing can be store-and-forward or cut-through
  - Flow control
    - contention for output links in a switch can block progress
    - generally low-latency per-link flow control is used
      - delay in access to a link rapidly propagates back to sender

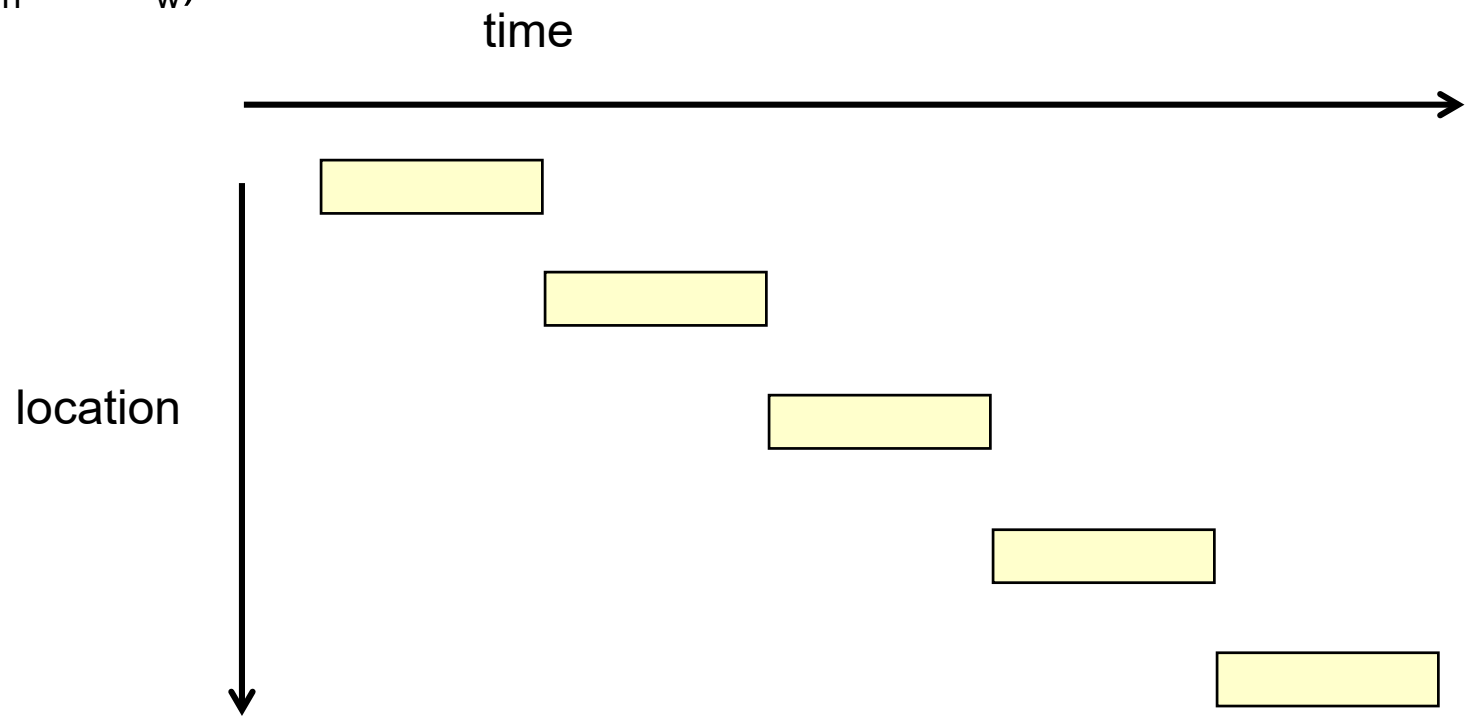
Communication cost model

- Message size m bits
- Number of hops (links) to travel h
- Channel width W and link cycle time t_c
  - Per-bit transfer time $t_w = t_c/W$
    - assuming m is sufficiently large
- Startup time t_s
  - overhead to insert message into network
- Node latency or per-hop time t_h
  - time taken by message header cross channel and be interpreted at destination

Store-and-forward routing

- flow-control mechanism at message or packet level
- packet s are transferred one link at a time
- large buffers, high latency
- cost

$$t_{SF} = t_s + (t_h + m\, t_w)\, h$$



Cut-through routing

- flow control is per-link and payload transmission is pipelined
- message spread out across multiple links in the network
- small buffers, low latency
- cost

$$t_{CT} = t_s + ht_h + mt_w$$