EE/CSCI 451: Parallel and Distributed Computation

Lecture #8

9/10/2020

Viktor Prasanna

prasanna@usc.edu

ceng.usc.edu/~prasanna

University of Southern California

Announcement

- Midterm 1 date: 9/25
  - Discussion session - 2hours

- HW2 due 9/10 (upcoming Thu.)
- PHW2 due 9/14 (+3 free late-days)
- HW1 and PHW1 grades are out

PHW1 Statistics	
Average	93.51
Median	98.00
Standard Deviation	18.86

HW1 Statistics	
Average	91.10
Median	96.00
Standard Deviation	15.39

Announcement: Midterm1 Logistics

- **Online Proctored** Exam:
  - Time: Week 6 discussion session – 2 hours: 3:30-5:30PM (Los Angeles time)
  - Format: Open-book, open-notes **[Attendance is required, no make-up given]**
  - Proctoring: 2 proctors watching different subgroups of students in separate Zoom meetings, links will be sent to students in advance
    - Require **camera-enabled** device
- Receiving and returning your exam:
  - Exam will be released on Piazza under resource page at/around 3:26 PM
  - You will submit the completed exam on Blackboard (a submission portal will be created in advance)
- Completing your exam:
  - Download the assignment pages (exam pages) as pdf files on to your tablet and annotate it with your answers. <u>Only hand-annotated pdf files are acceptable.</u>
    - Require a **writable tablet** device
- Coverage: Week 1-Week 5 contents (Week 6 contents - analytical modeling & communication primitives - not covered)
- Special note 1: Important - discussion **attendance is required** on Week 5 (Sept 18)!
  - 10-min midterm trial run to make sure all students are prepared for and comfortable with the exam process
- Special note 2:
  - We have created a Piazza poll [link] to collect info regarding your available resources/capabilities to complete the exam with writable tablet. Everyone is **required to participate in the poll**

Announcement

- Lecture slides will be released **before** each class for your reference

- Occasionally, lecture slides will be **updated** after the lecture
  - The updated slides will over raid the slides released before the class

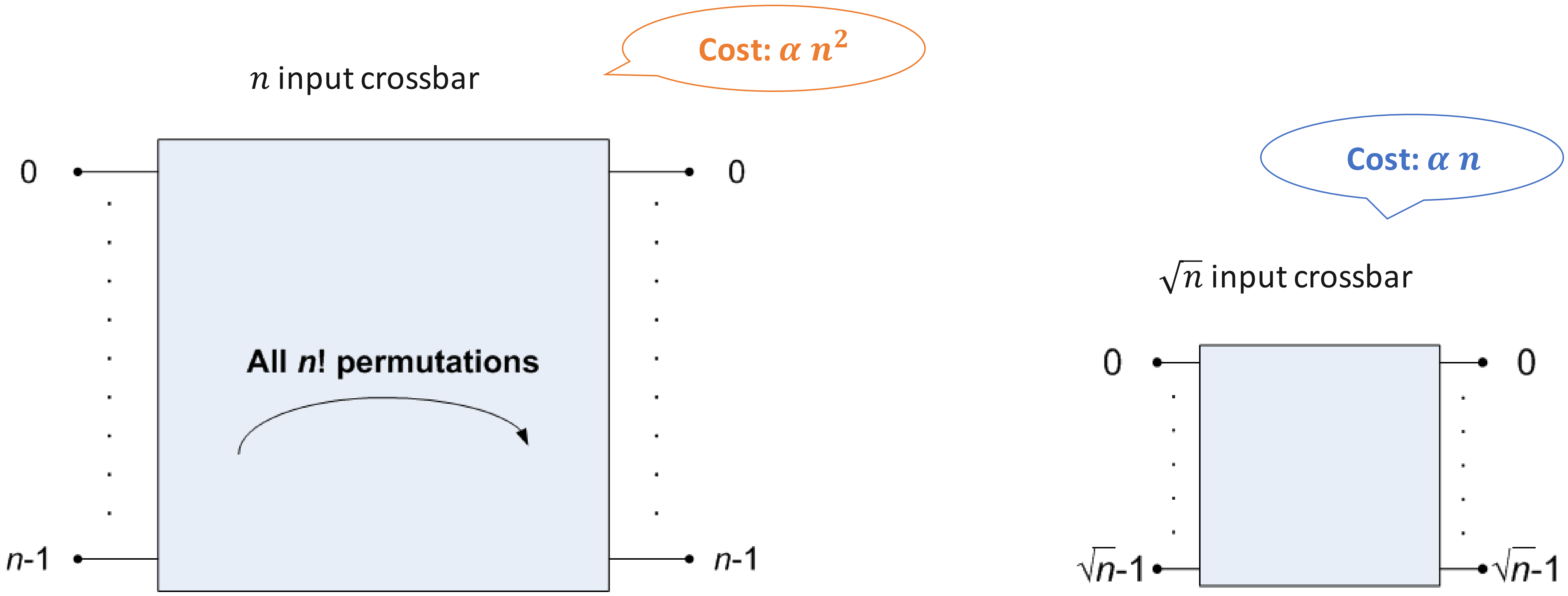
- Lecture slides are available at https://piazza.com/usc/fall2020/eecsci451/resources

Outline

- From last class
  - Static and Dynamic networks
  - Shuffle exchange network
  - Multistage network
  - Crossbar network
  - Performance metrics
- Today
  - CLOS network
  - Butterfly network
  - Hypercube network
  - Tree-based network
  - Performance metrics

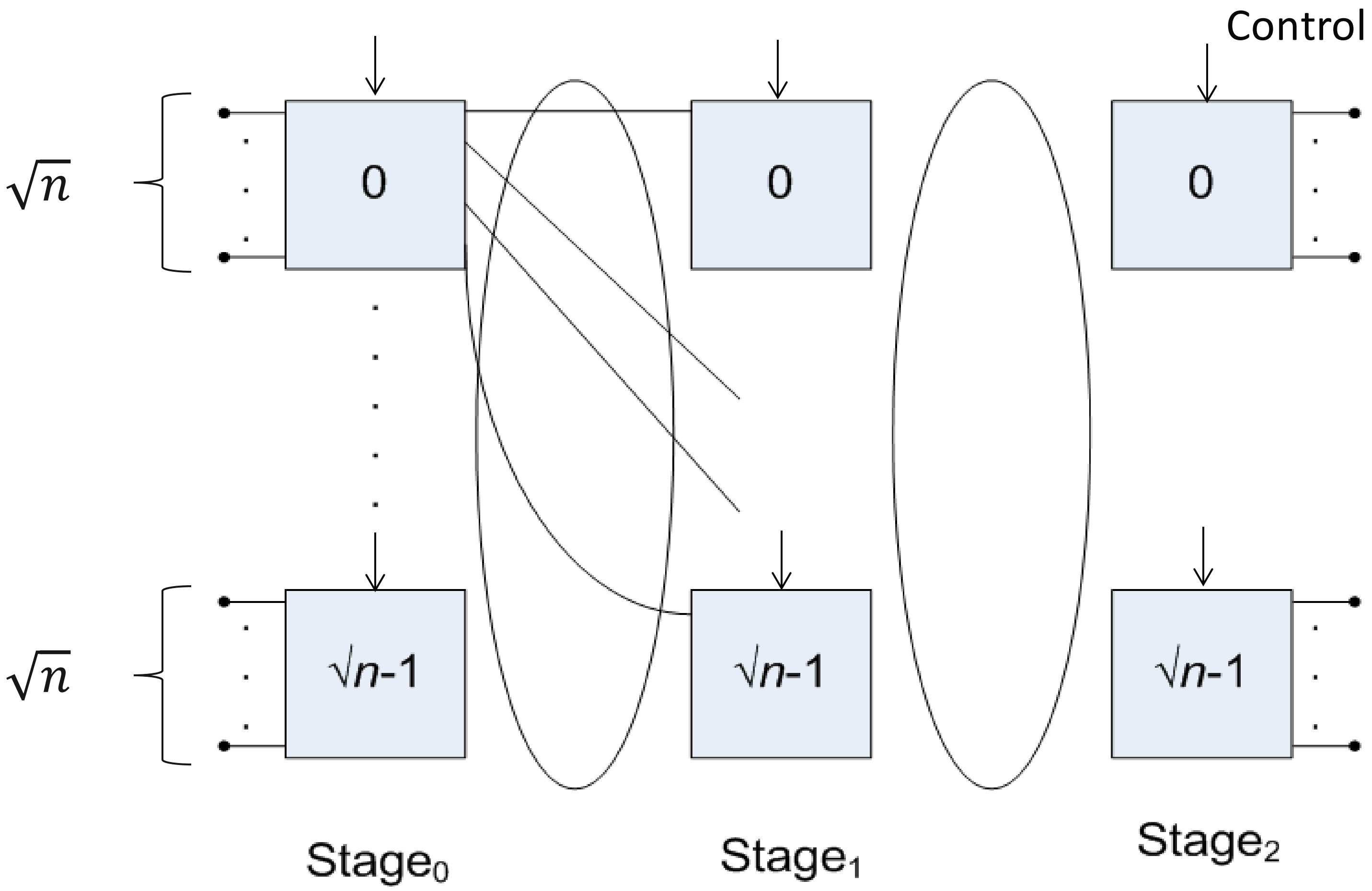
CLOS network (1)

Multistage network



CLOS network (2)

Structure of CLOS network



CLOS network (3)

Stage $\boldsymbol{i}$ → Stage $\boldsymbol{i+1}$ connections, $\boldsymbol{i=0,1}$

Any Box in stage i is connected to all boxes in the next stage

- Box - $\sqrt{n}$ output ports
- Stage - $\sqrt{n}$ boxes

CLOS network (4)

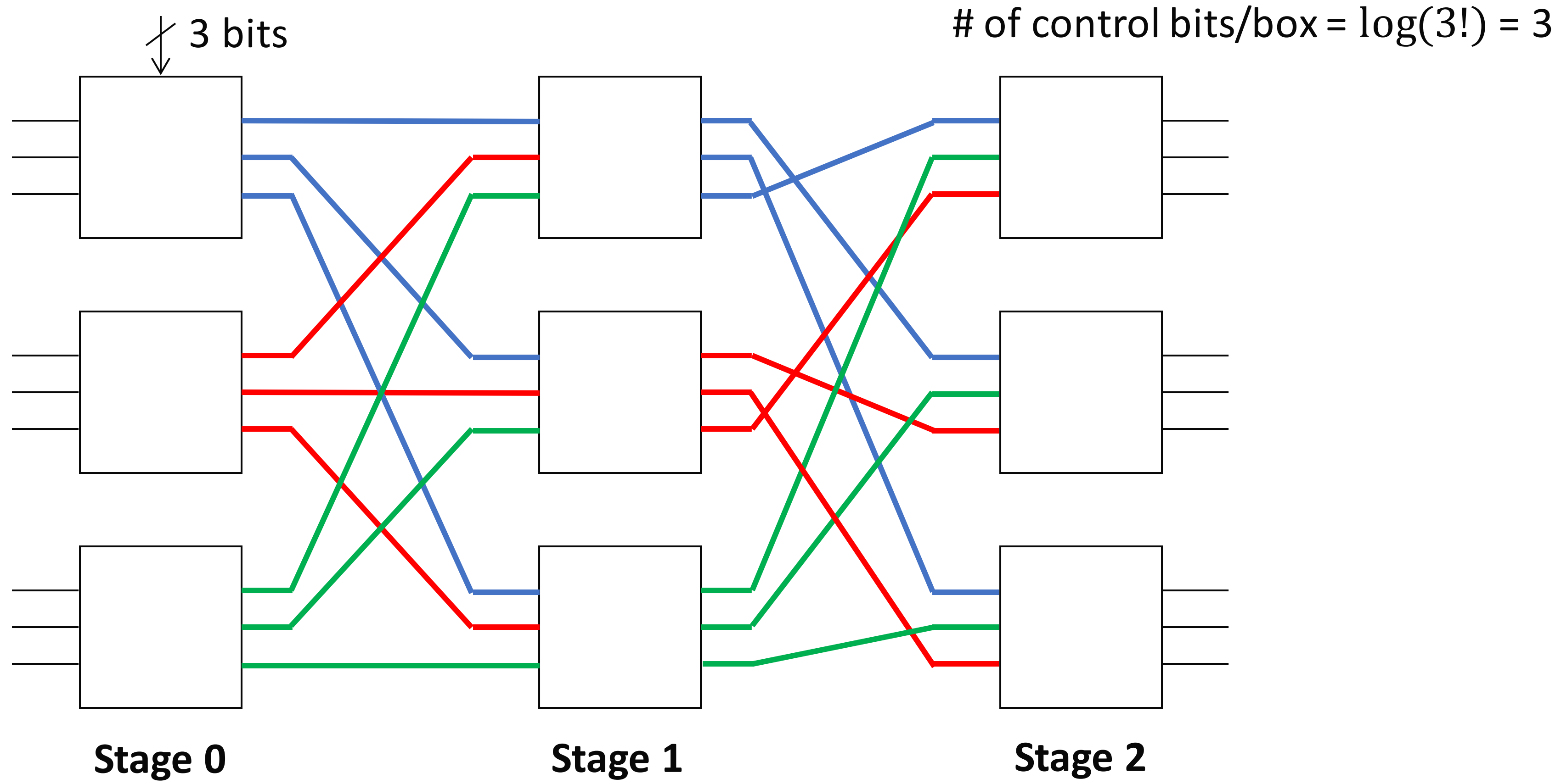
- 3 stage network
- Each box → $\sqrt{n} \times \sqrt{n}$ crossbar
- Number of switches (boxes) = $3\sqrt{n}$
- Cost of $\sqrt{n} \times \sqrt{n}$ crossbar = $O\left((\sqrt{n})^2\right)$
- Total Cost = $O(\,n\sqrt{n}\,)$ = $O(n^{\frac{3}{2}})$

Note: CLOS network can realize all $n!$ permutations

(For **any** permutation from input to output, each of the $3\sqrt{n}$ boxes can be configured such that the connection specified by the permutation can be realized)

Example CLOS network

Example using 3×3 switches (n = 9)

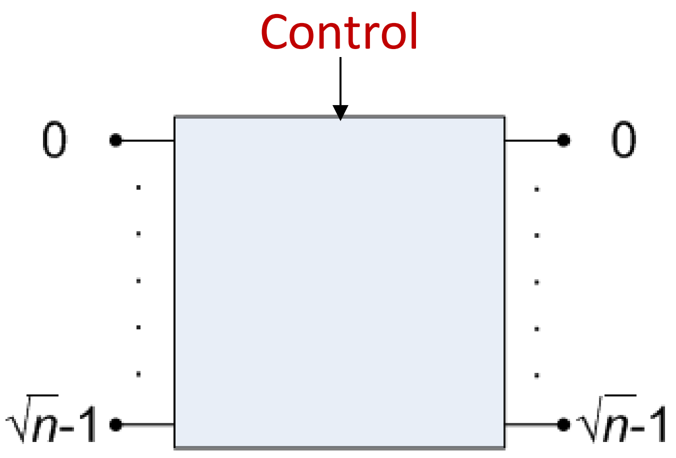


CLOS network (5)

Realizing a permutation

Choose the control setting for each of the $3\sqrt{n}$ boxes so the desired connection is realized

# of control bits/box = $\log(\sqrt{n}!)$

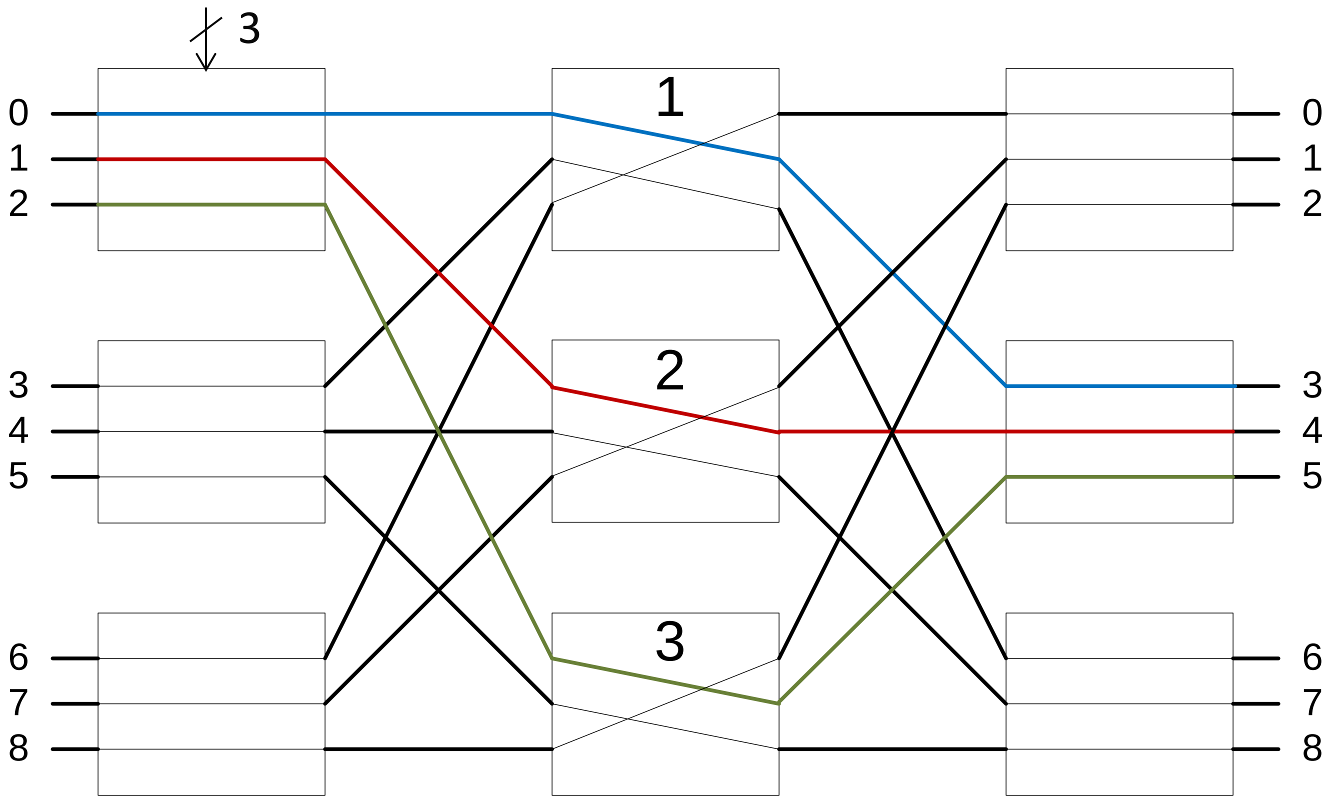


Any permutation on $\sqrt{n}$ inputs

can be specified by $\sqrt{n}\,\log\sqrt{n}$ bits

CLOS network (6)

Example: $i \rightarrow (i+3) \bmod 9$



CLOS network (7)

Example: 4 input / output CLOS network

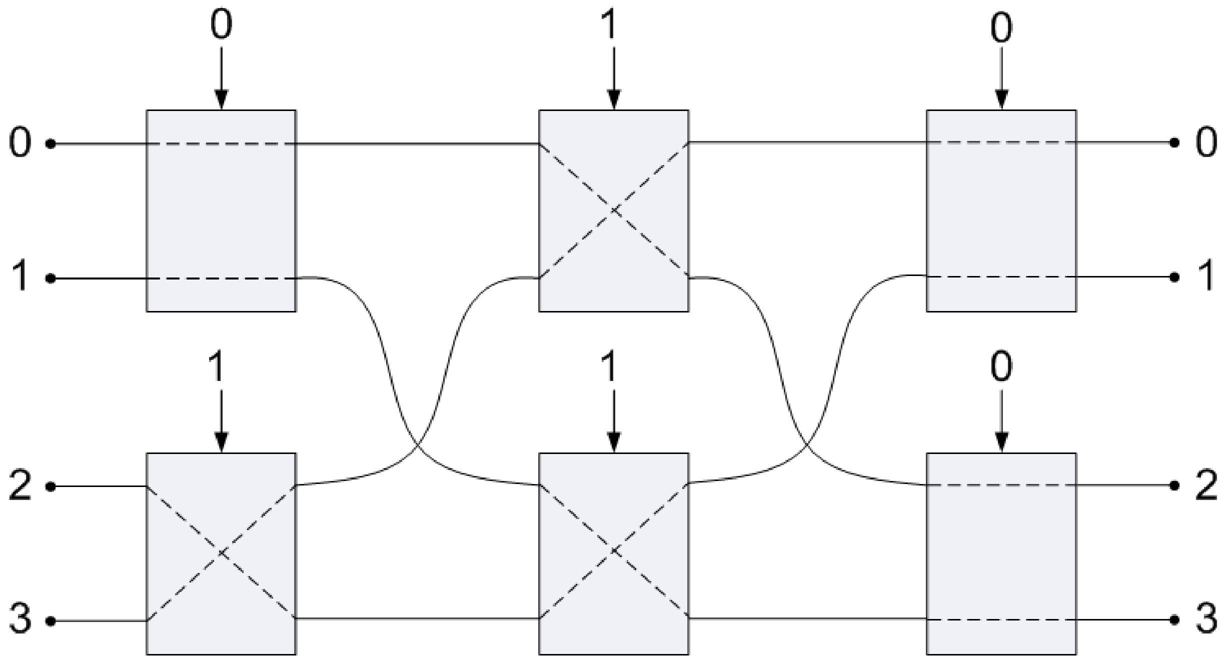
Example permutation:

$0 \rightarrow 2$

$1 \rightarrow 3$

$2 \rightarrow 1$

$3 \rightarrow 0$



Note: All 4! permutations can be realized by above network

CLOS network (8)

Another example: 4 input / output CLOS network

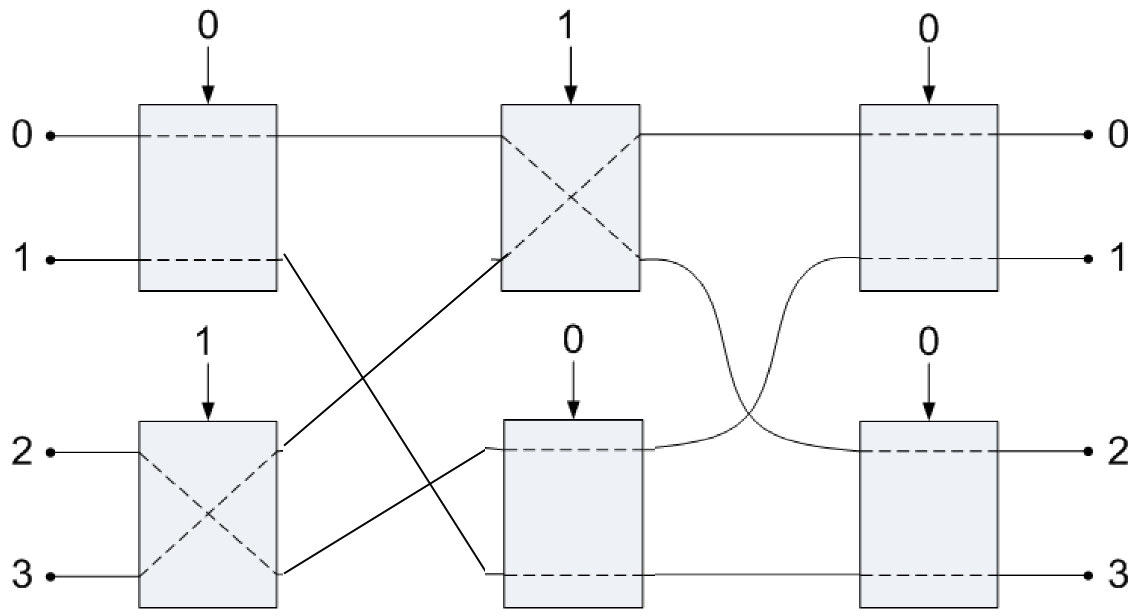
Example permutation:

$0 \rightarrow 2$

$1 \rightarrow 3$

$2 \rightarrow 1$

$3 \rightarrow 0$



CLOS network (9)

- Routing problem
  - Given a permutation p
  - Specify the switch settings such that all connections in p are realized

Each $3\sqrt{n}$ box $\rightarrow$ permutation on $\sqrt{n}$ inputs

Multistage network (1)

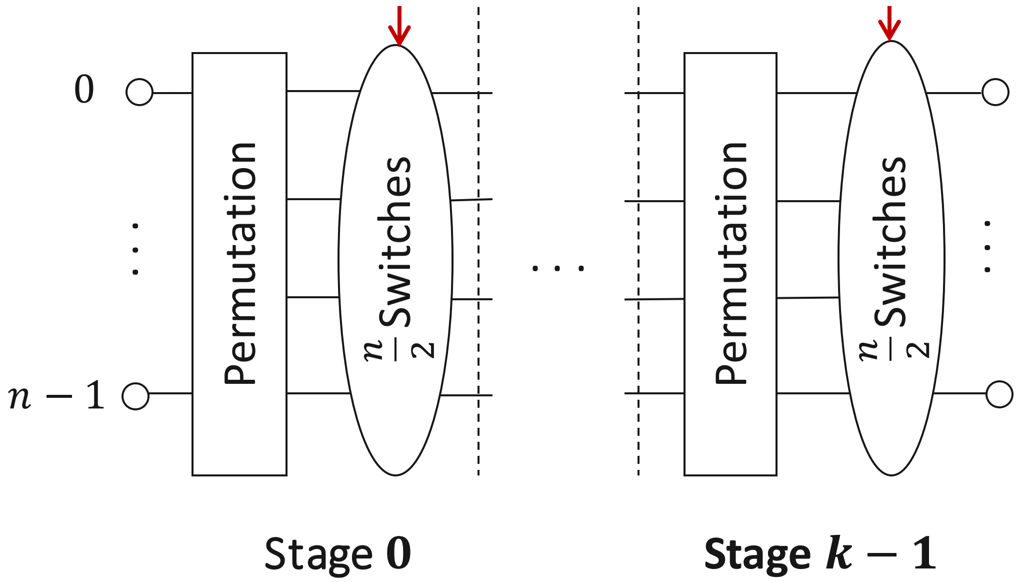
Non blocking network

Any connection request from input to output can be routed at **any** time **without rearranging** the existing set of connections at that time

Multistage network (2)

Rearrangeable network (1)

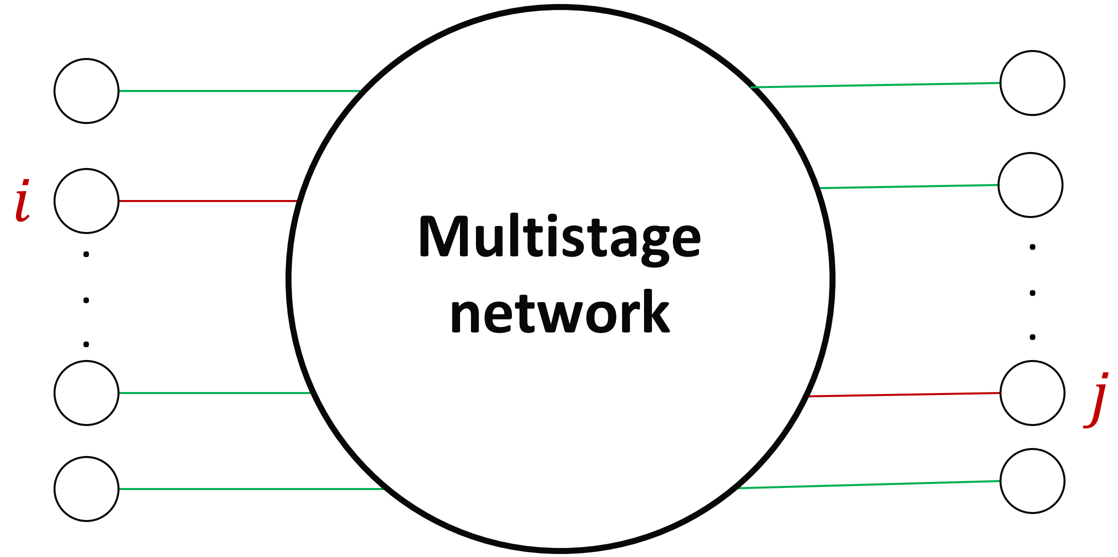
To route a connection, we may have to rearrange existing connections, i.e., change the control settings of some switches that are routing an existing connection



Multistage network (3)

Rearrangeable network (2)

A (multistage) network is rearrangeable if **any** connection request ($i \rightarrow j$) can be realized by (possibly) rearranging some existing connections at that time.

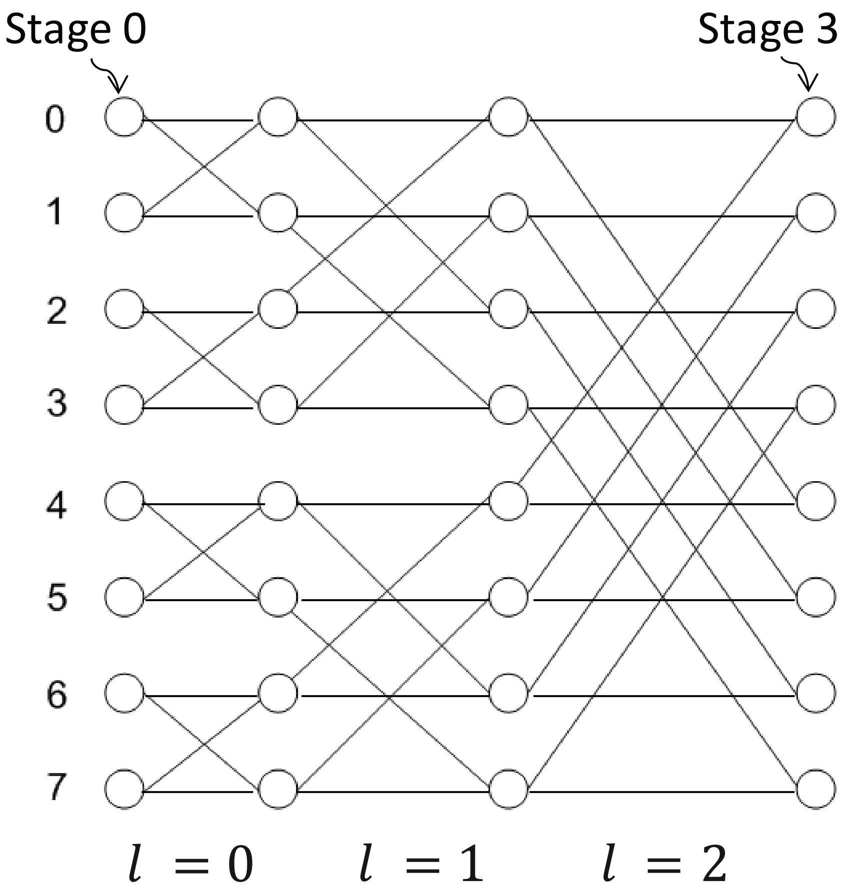


Green: existing
Red: new

Butterfly network (1)

- $n = 2^k$ for some k
- $\log_2 n + 1$ stages
- Power of 2 connections
- $2^l, l = 0,1, \dots \log_2 n - 1$
- **Stage l**, $0 \leq l < \log_2 n$
- $i \rightarrow i$ in Stage $l + 1$
- $i \rightarrow i$ complement bit l in Stage $l + 1$

8-input butterfly network



Butterfly network (2)

n input butterfly network

- Total number of nodes $= n \cdot (\log_2 n + 1)$

- Total number of edges $= n \cdot 2 \cdot \log_2 n$

# of stages

# of edges/node

Mesh-connected Network

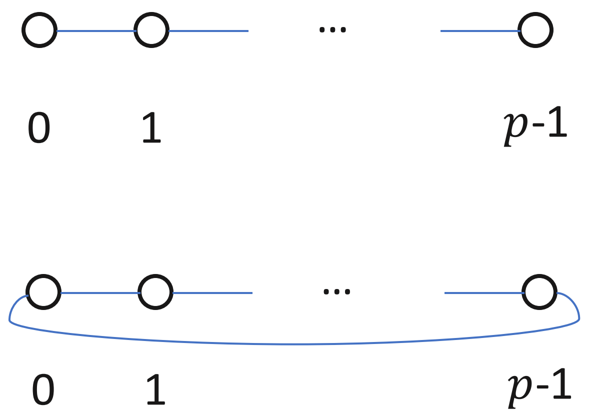
1-D mesh

Without wraparound

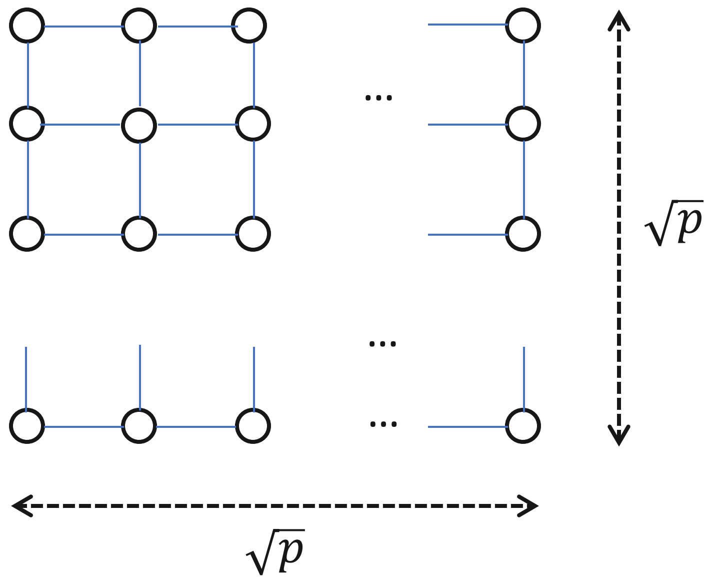
1-D torus

With wraparound

ring



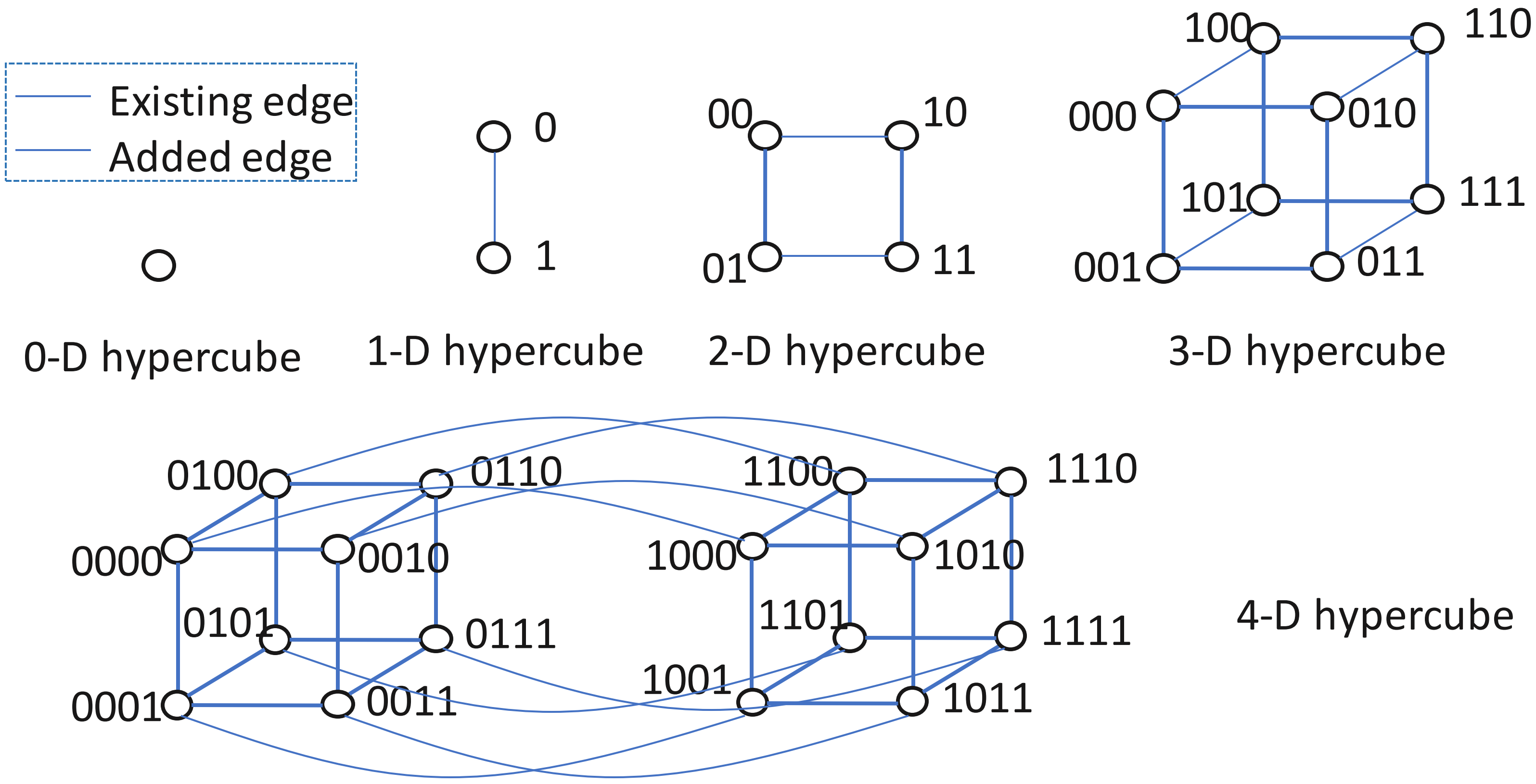
2-D mesh (without wraparound)



k-dimensional mesh: $\underbrace{p^{1/k} \times p^{1/k} \times \cdots \times p^{1/k}}_{k \text{ times}}$

Number of connections per node $\leq 2k$

Hypercube Network (1)



Construction of hypercube from hypercubes of lower dimension

Hypercube Network (2)

- In general, for k dimension hypercube:

  $p = \text{total number of nodes} = 2^k$

  Node $i = i_{k-1} i_{k-2} \dots i_0$

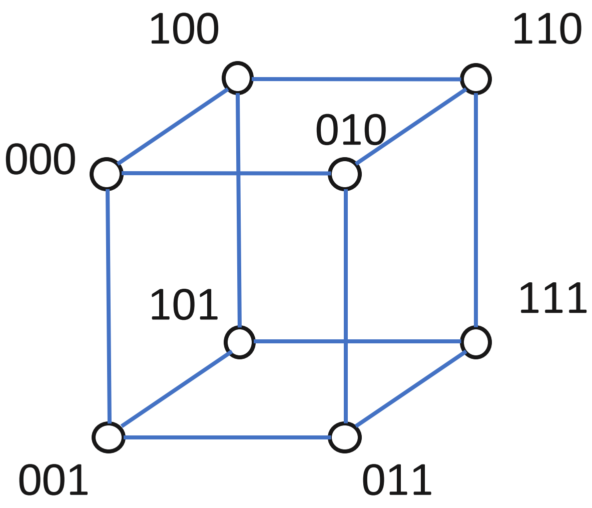
  k connections/node – complement a bit of i

- Example:

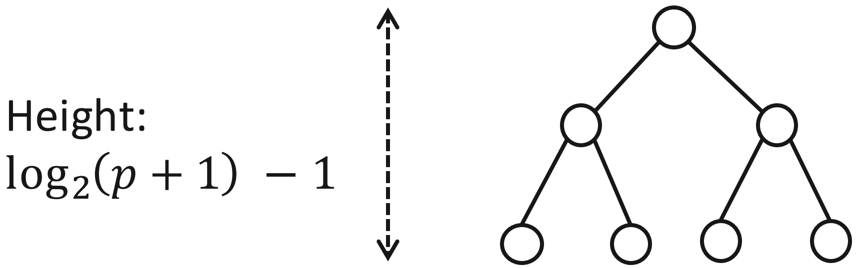
  000 — 001

  000 — 010

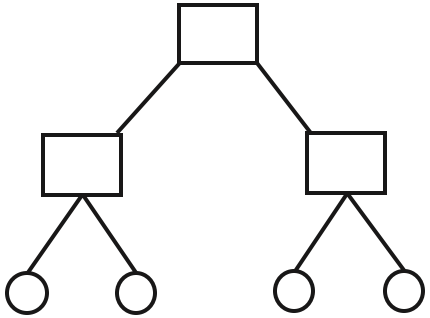
  000 — 100



Tree-based Network (1)



Static tree network

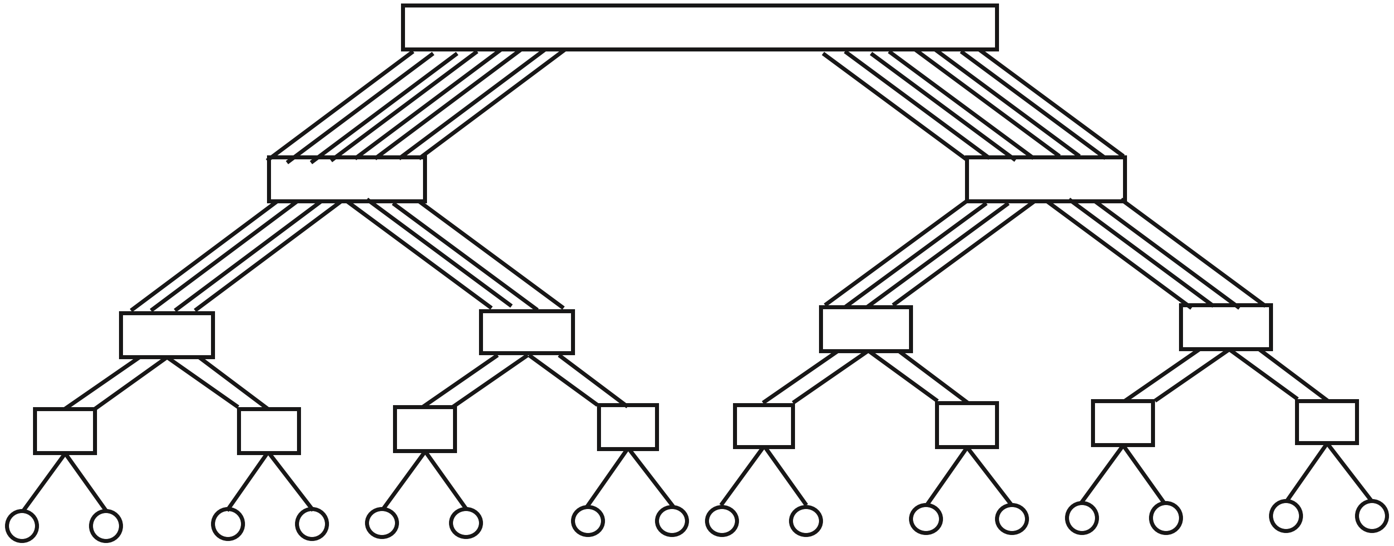


Dynamic tree network

p = total number of nodes

Tree-based Network (2)

Fat tree (16 processing nodes)



- In a fat tree, branches nearer the top of the hierarchy have higher bandwidth than branches further down the hierarchy.

Performance metrics (1)

Diameter

- Diameter $\triangleq$ Maximum distance between any two processing nodes in the network

$$diameter \triangleq \max_{(i,j)}\{distance(i,j)\}$$

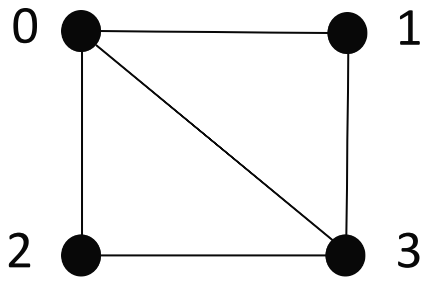
$$distance(i,j) \triangleq \min \text{path length between } i \text{ and } j$$

$$\triangleq \text{length of the shortest path between } i \text{ and } j$$

$$\text{length of a path} = \#\text{ of edges in the path}$$

Performance metrics (2)

Example:



	Distance
$0 \rightarrow 1$	1
$0 \rightarrow 2$	1
$0 \rightarrow 3$	1
$1 \rightarrow 0$	1
$1 \rightarrow 2$	2
$1 \rightarrow 3$	1
$2 \rightarrow 0$	1
$2 \rightarrow 1$	2
$2 \rightarrow 3$	1
$3 \rightarrow 0$	1
$3 \rightarrow 1$	1
$3 \rightarrow 2$	1

$\Rightarrow$ Diameter = 2

Performance metrics (3)

Diameter

- Example:
  - 1-D mesh (no wraparound): $p - 1\ \{0 \rightarrow p - 1\}$
  - 2-D mesh (no wraparound): $2(\sqrt{p} - 1)\ \{(0,0) \rightarrow (\sqrt{p} - 1, \sqrt{p} - 1)\}$
  - Hypercube : $\log_2 p\ \{0 \rightarrow p - 1\}$

Performance metrics (4)

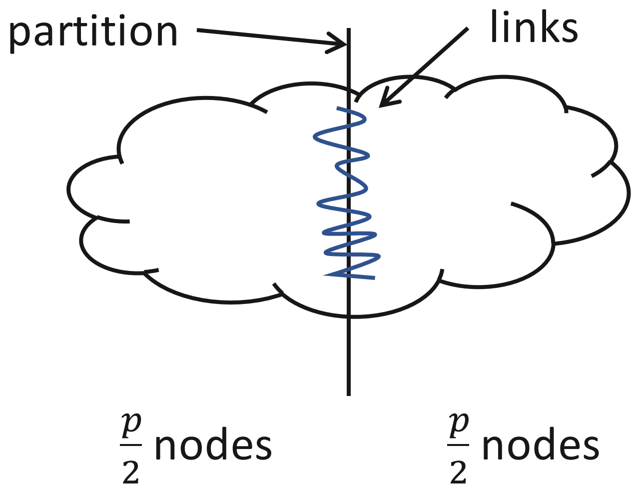
Diameter = d $\Rightarrow$ Routing can be performed in at most d steps (hops).

Note: The routing algorithm may not do shortest path routing (for example, to reduce congestion)

Performance metrics (5)

Bisection width (1)

- Bisection width ≜ Minimum number of links to be removed to partition the network into two equal-sized (in number of nodes) subnetworks

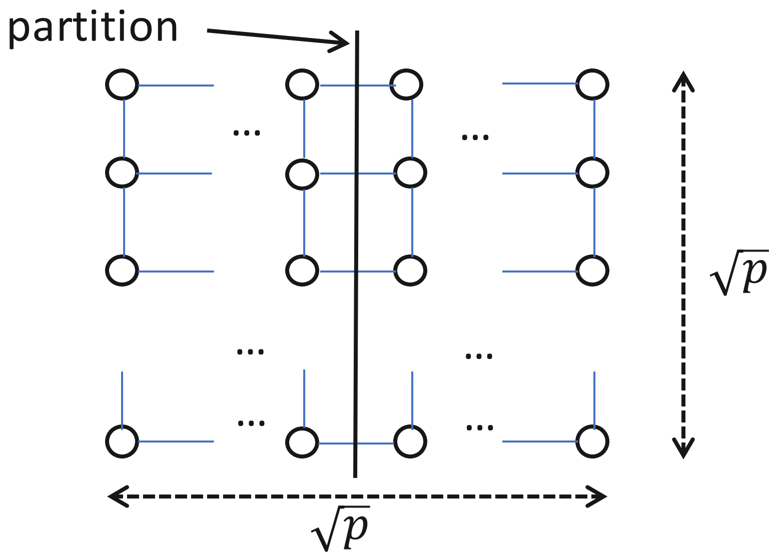


Minimum over **all** possible partitions

Performance metrics (6)

Bisection width (2)

- Example

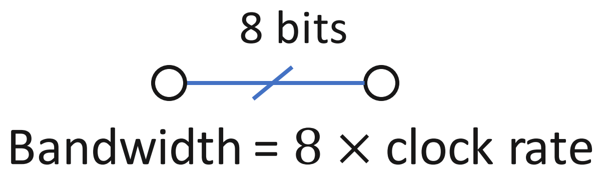


$\sqrt{p}$ for 2-D mesh (no wraparound)

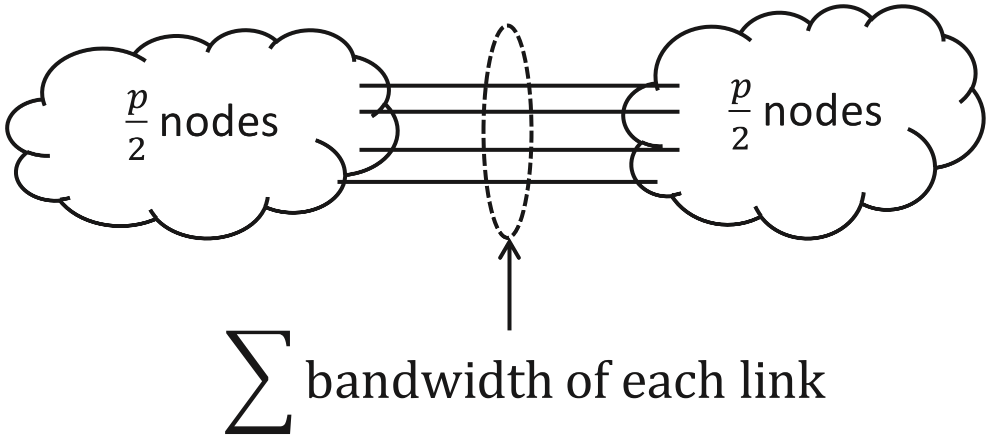
Performance metrics (7)

Bisection bandwidth (3)

- Bandwidth $\rightarrow$ number of bits/sec
  - Example:



- Bisection bandwidth $\triangleq$ Minimum bandwidth between any two equal halves (number of bits/sec that can be exchanged between 2 equal halves of the network)



Performance metrics (8)

Cost of a static network

- Cost $\triangleq$ number of communication links in the network
- Example:
  - Tree: $p - 1$
  - 1-D mesh (no wraparound): $p - 1$
  - d-dimensional wraparound mesh: $d \cdot p$
  - Hypercube: $\frac{p \cdot \log_2 p}{2}$
  - k-ary d-cube
    - A d-dimensional array with k elements in each dimension
    - Number of nodes $p = k^d$
    - Cost: dp

Summary

Network	Diameter	Bisection width	Cost (No. of links)
Completely connected	1	$\frac{p^2}{4}$	$\frac{p(p-1)}{2}$
Star	2	1	$p-1$
Complete binary tree	$2\log\left(\frac{p+1}{2}\right)$	1	$p-1$
1-D mesh, no wraparound	$p-1$	1	$p-1$
2-D mesh, no wraparound	$2(\sqrt{p}-1)$	$\sqrt{p}$	$2(p-\sqrt{p})$
2-D wraparound mesh	$2\lfloor\sqrt{p}/2\rfloor$	$2\sqrt{p}$	$2p$
Hypercube	$\log p$	$p/2$	$(p \cdot \log p)/2$
Wraparound k-ary d-cube $p = k^d$	$d\lfloor k/2\rfloor$	$2k^{d-1}$	dp

Interconnection networks

Desirable Features

- Low diameter (fast routing)
- High bisection bandwidth (large number of concurrent data transfers)
- Low cost (logic + wiring)
- Localized connections (high clock rate)
- Low congestion (low message latency)
- Simple distributed routing algorithm (low routing overhead)

Summary

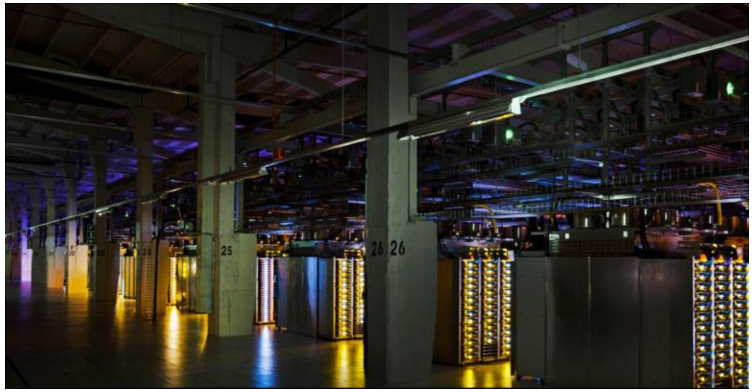
- Static network
- Multistage network
- Cost
- Performance
  - Routing
  - Diameter
  - Bisection bandwidth

Some Example Deployed Networks

Oak Ridge National Laboratory
Summit: Fastest computer in the world (2019)

- Network: Fat Tree
- Bandwidth: ~1 Petabits/s (entire network)

Google
Jupiter: A 40G Datacenter-scale Fabric



- Network: CLOS
- Bandwidth: ~1.3 Petabits/s (entire network)