UCIrvine | The Henry Samueli School of Engineering

Network topology; Distributed memory algorithms (contd)

EECS 221: Intro to High-Performance Computing

Aparna Chandramowlishwaran

April 25, 2017

SUMMA: 2D distributed matrix multiply (whiteboard)

Properties of a network

- **Analogy**: Networks are like streets
  - Link = street
  - Switch = intersection
  - Hops = number of blocks
  - Routing algorithm = directions

- **Latency**: Time to move data between nodes in the network
- **Bandwidth**: Rate at which date can be moved

Properties of a network

- **Analogy**: Networks are like streets
  - Link = street
  - Switch = intersection
  - Hops = number of blocks
  - Routing algorithm = directions

- **Latency**: Time to move data between nodes in the network
- **Bandwidth**: Rate at which date can be moved
  - Limitation — number of wires and rate at which each wire can accept data

Bisection bandwidth

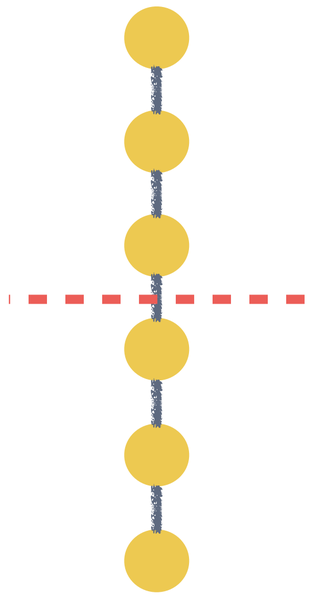
- **Definition**: bandwidth across smallest cut that divides the network into two equal parts

Bisection bandwidth

- **Definition**: bandwidth across smallest cut that divides the network into two equal parts

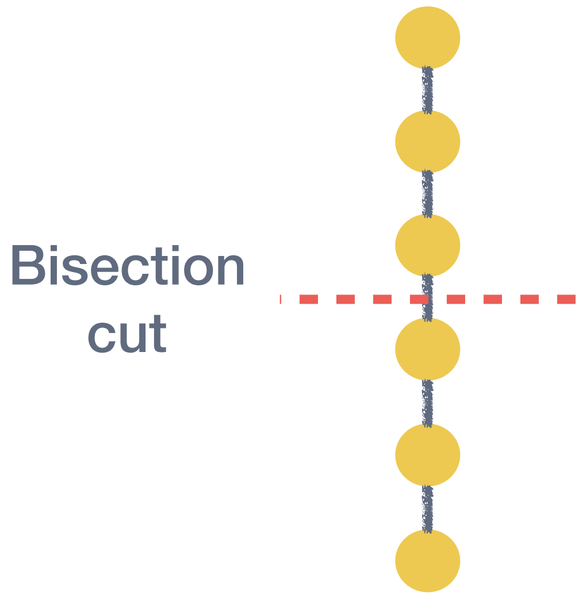
Bisection bandwidth

- **Definition**: bandwidth across smallest cut that divides the network into two equal parts



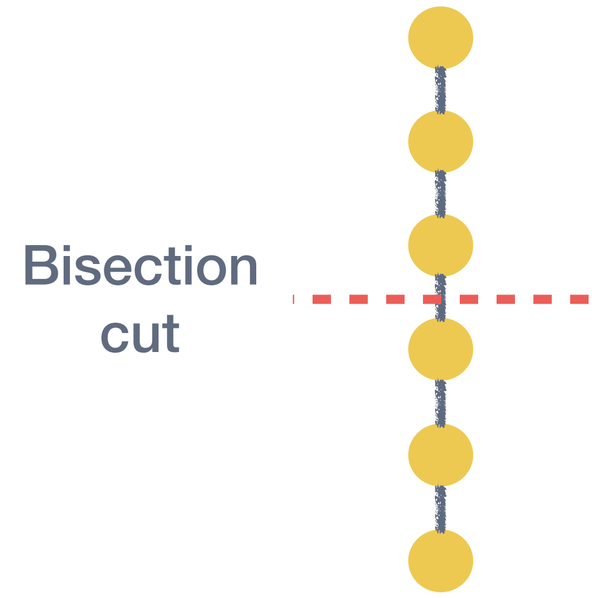
Bisection bandwidth

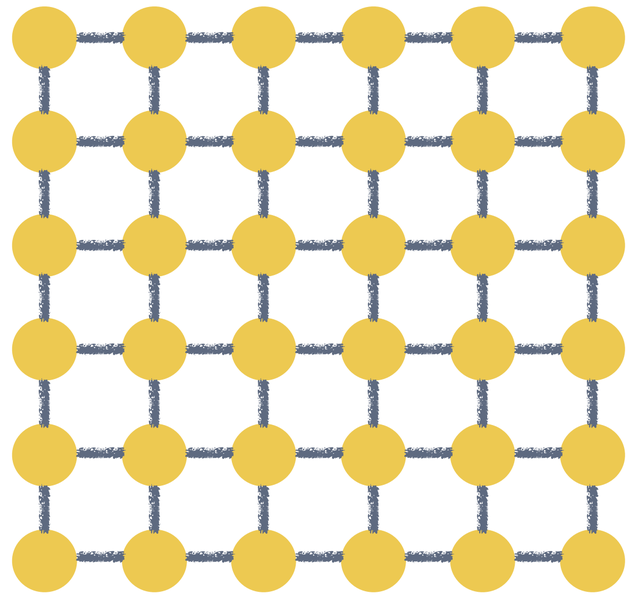
- **Definition**: bandwidth across smallest cut that divides the network into two equal parts



Bisection bandwidth

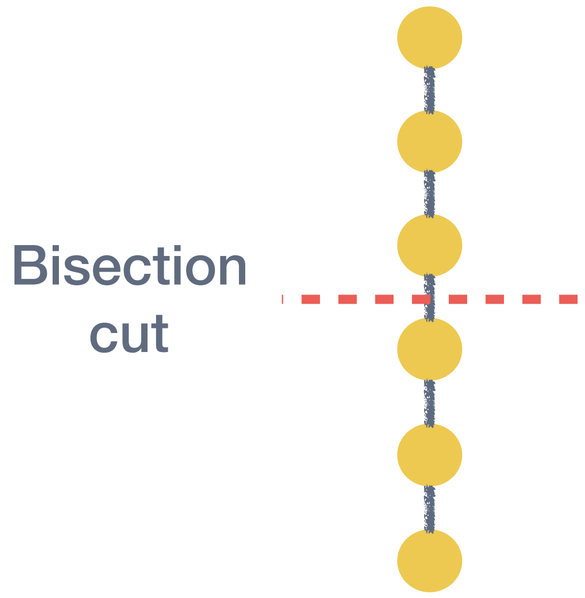
- **Definition**: bandwidth across smallest cut that divides the network into two equal parts

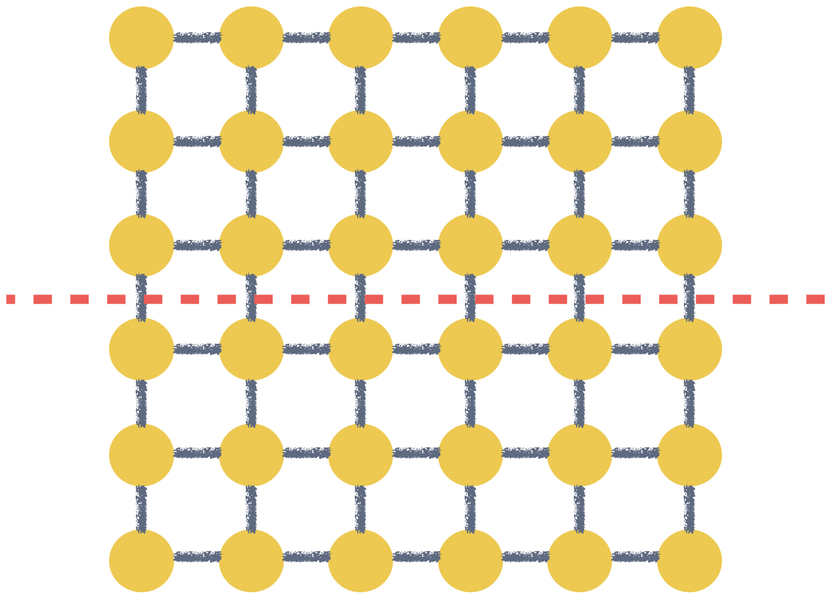




Bisection bandwidth

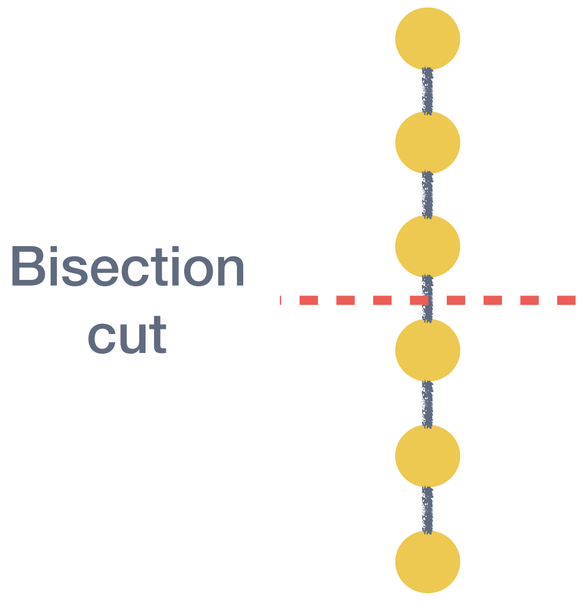
- **Definition**: bandwidth across smallest cut that divides the network into two equal parts

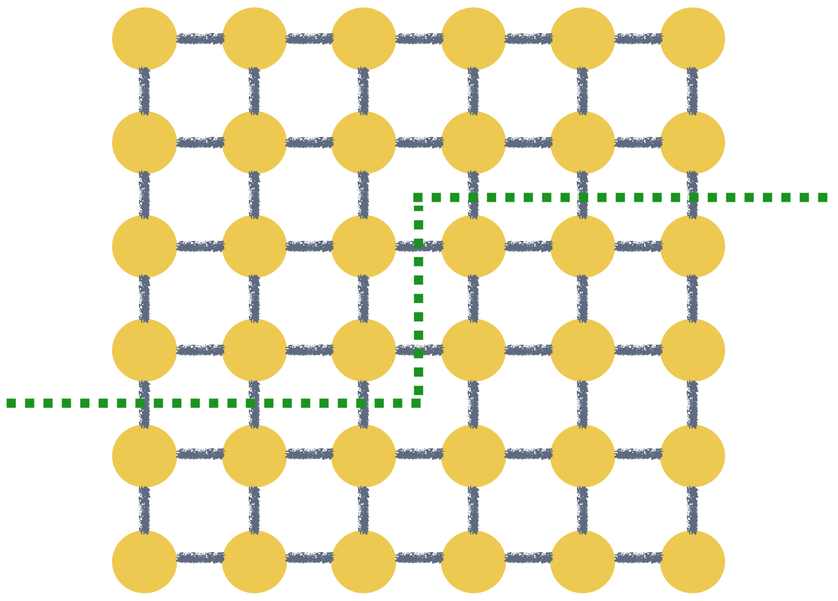




Bisection bandwidth

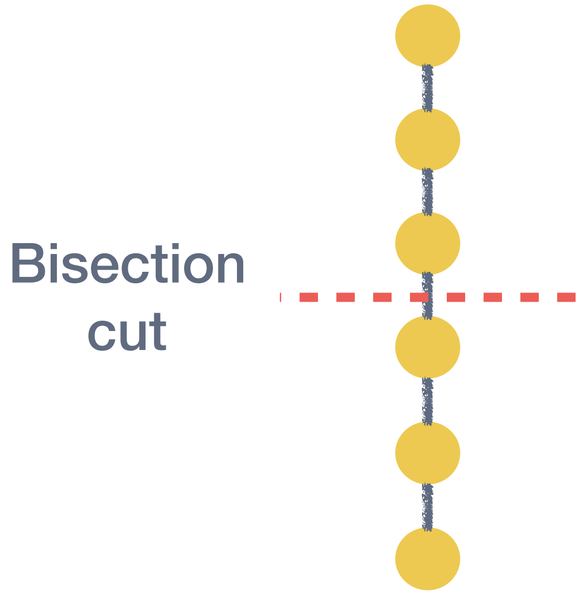
- **Definition**: bandwidth across smallest cut that divides the network into two equal parts

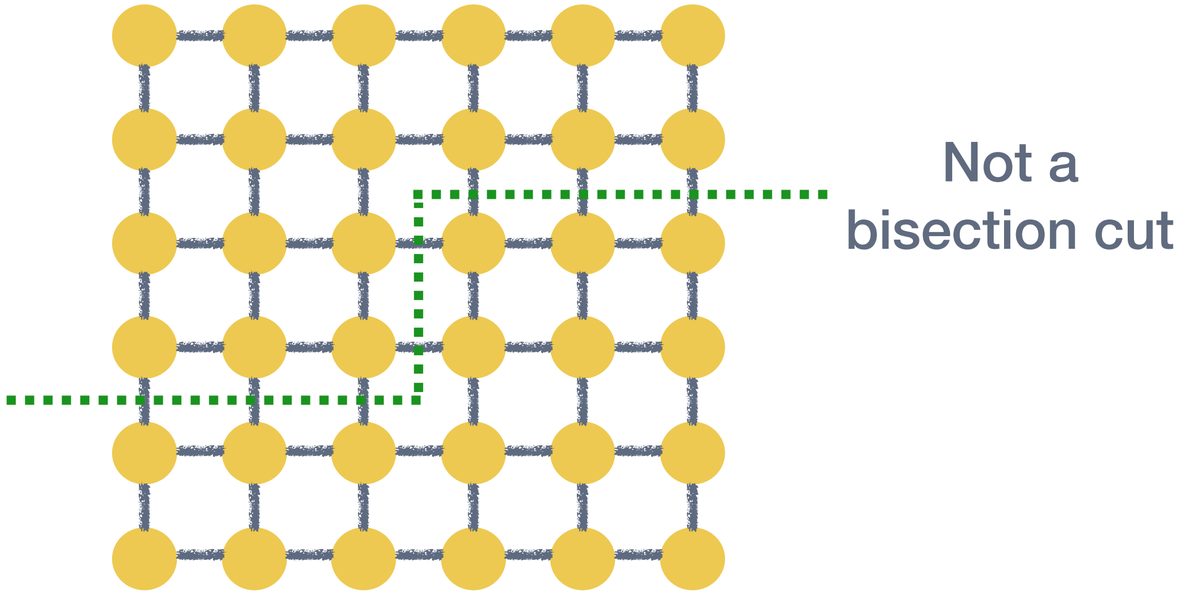




Bisection bandwidth

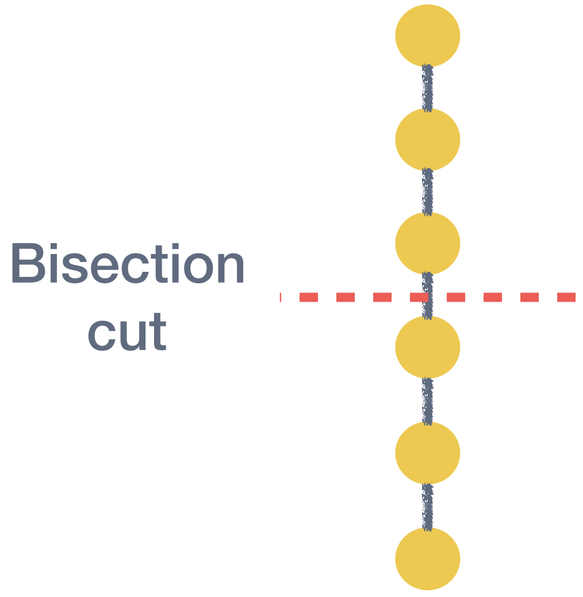
- **Definition**: bandwidth across smallest cut that divides the network into two equal parts



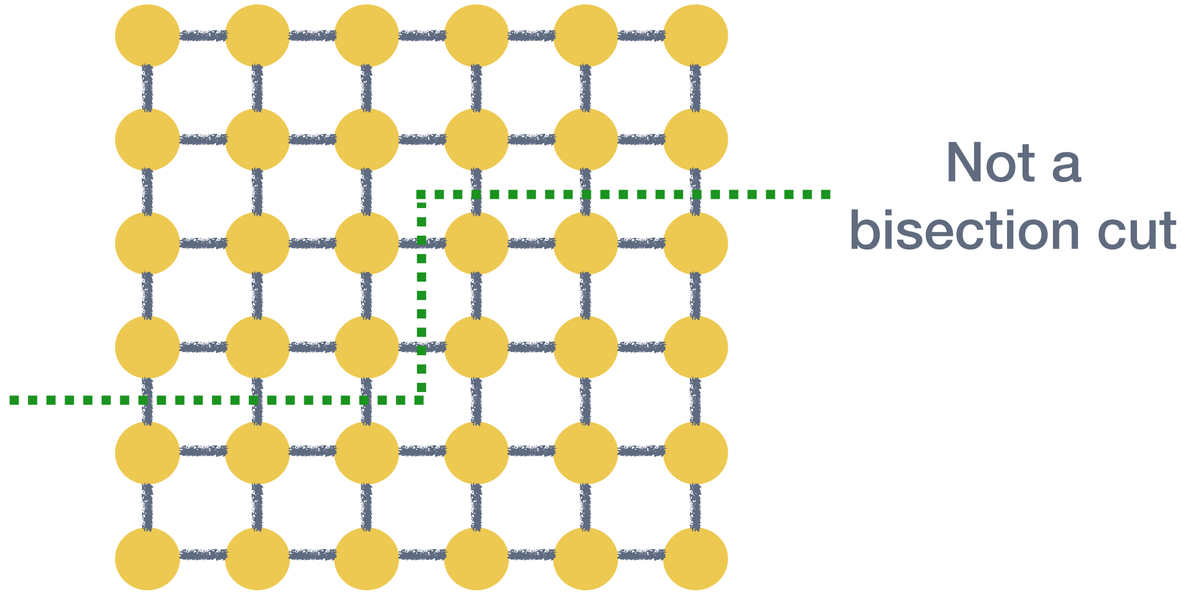


Bisection bandwidth

- **Definition**: bandwidth across smallest cut that divides the network into two equal parts

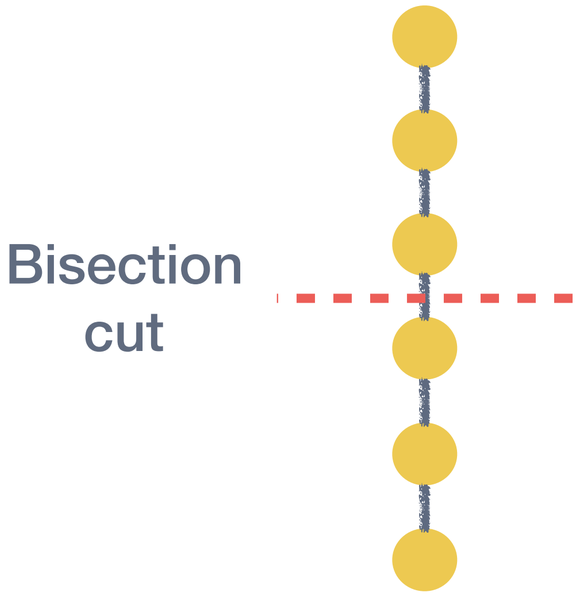


bisection bw = link bw

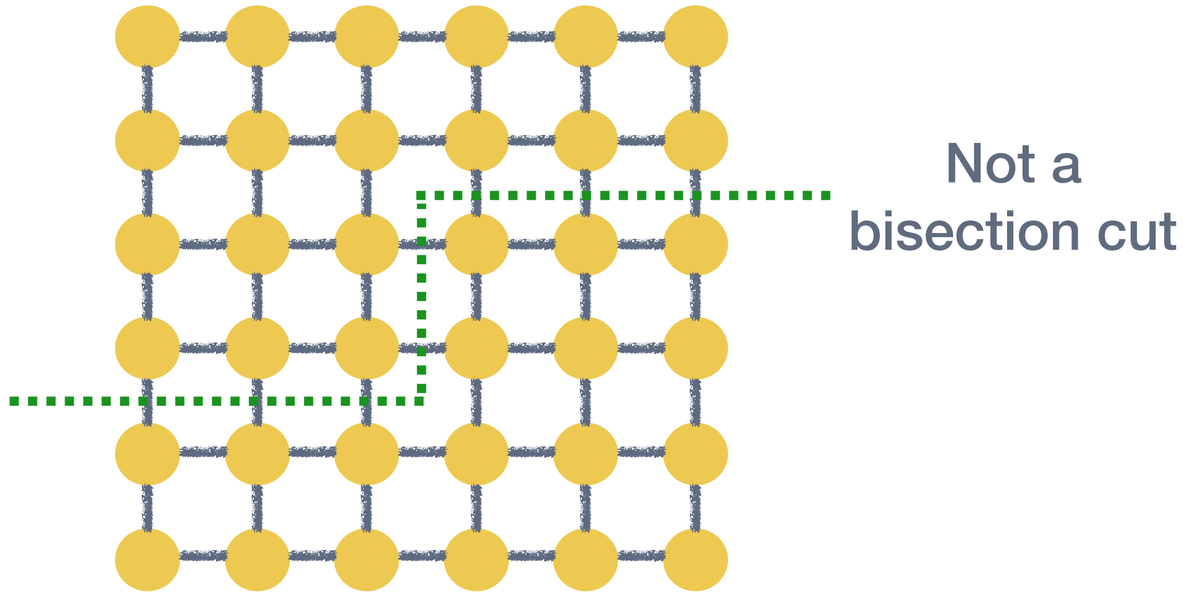


Bisection bandwidth

- **Definition**: bandwidth across smallest cut that divides the network into two equal parts



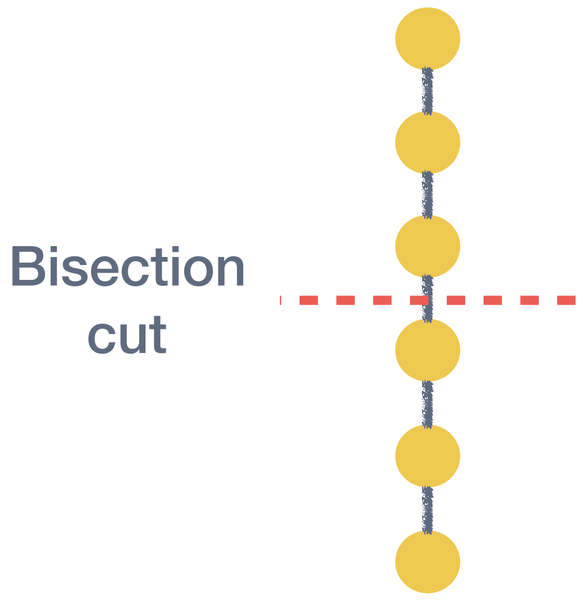
bisection bw = link bw



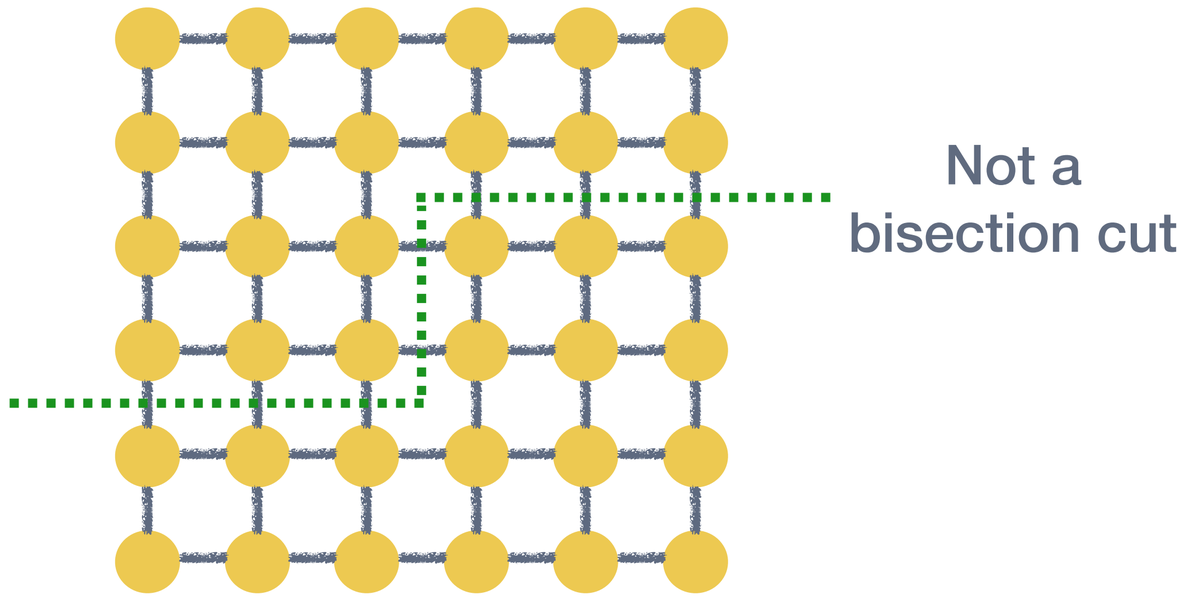
bisection bw = √p * link bw

Bisection bandwidth

- **Definition**: bandwidth across smallest cut that divides the network into two equal parts



bisection bw = link bw



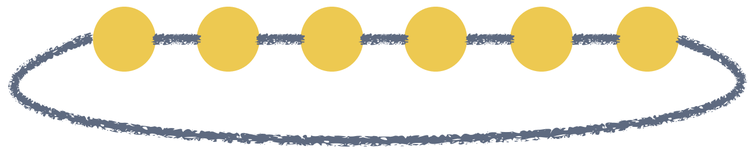
bisection bw = √p * link bw

- When is bisection bandwidth important?
  - All processors need to communicate with all other processors (or) all-to-all communication

Linear and ring networks

- **Diameter** = length of the shortest path between furthest pair of nodes
- **Bisection bandwidth** = bandwidth across smallest cut that bisects the network
- **Linear**: P-1 links

- **Ring/Torus**: P links

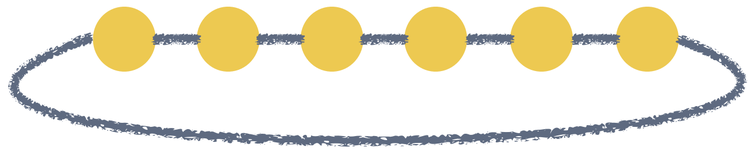


Linear and ring networks

- **Diameter** = length of the shortest path between furthest pair of nodes
- **Bisection bandwidth** = bandwidth across smallest cut that bisects the network
- **Linear**: P-1 links

Diameter = ?
Average distance = ?
Bisection = ?

- **Ring/Torus**: P links

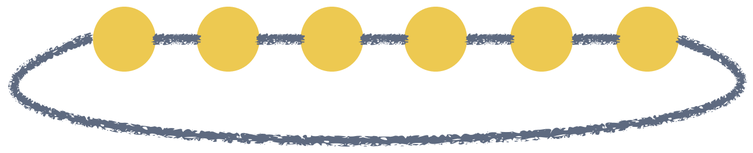


Linear and ring networks

- **Diameter** = length of the shortest path between furthest pair of nodes
- **Bisection bandwidth** = bandwidth across smallest cut that bisects the network
- **Linear**: P-1 links

Diameter = ?
Average distance = ?
Bisection = ?

- **Ring/Torus**: P links



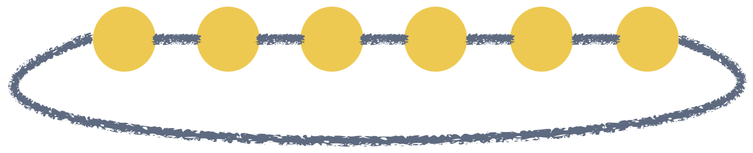
Diameter = ?
Average distance = ?
Bisection = ?

Linear and ring networks

- **Diameter** = length of the shortest path between furthest pair of nodes
- **Bisection bandwidth** = bandwidth across smallest cut that bisects the network
- **Linear**: P-1 links

Diameter = P - 1
Average distance ~ P / 3
Bisection = 1

- **Ring/Torus**: P links

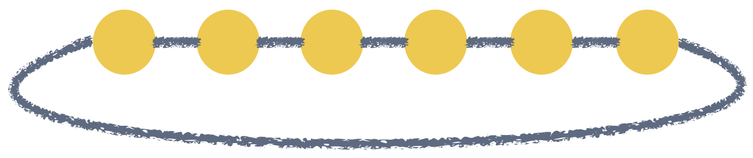


Linear and ring networks

- **Diameter** = length of the shortest path between furthest pair of nodes
- **Bisection bandwidth** = bandwidth across smallest cut that bisects the network
- **Linear**: P-1 links

Diameter = P - 1
Average distance ~ P / 3
Bisection = 1

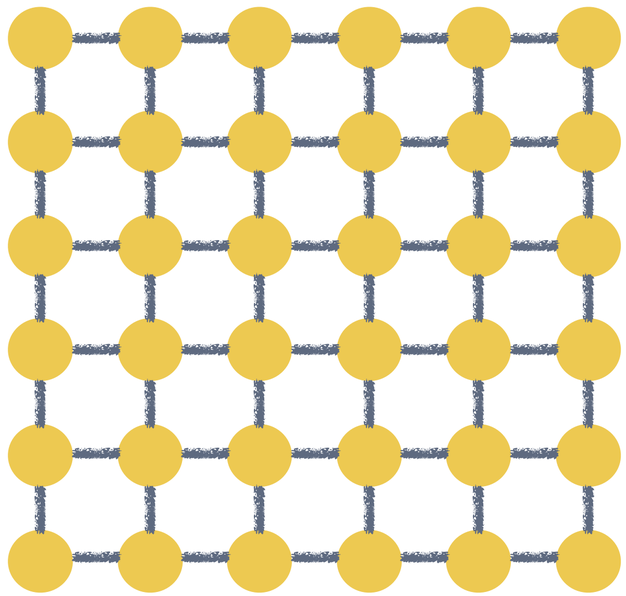
- **Ring/Torus**: P links



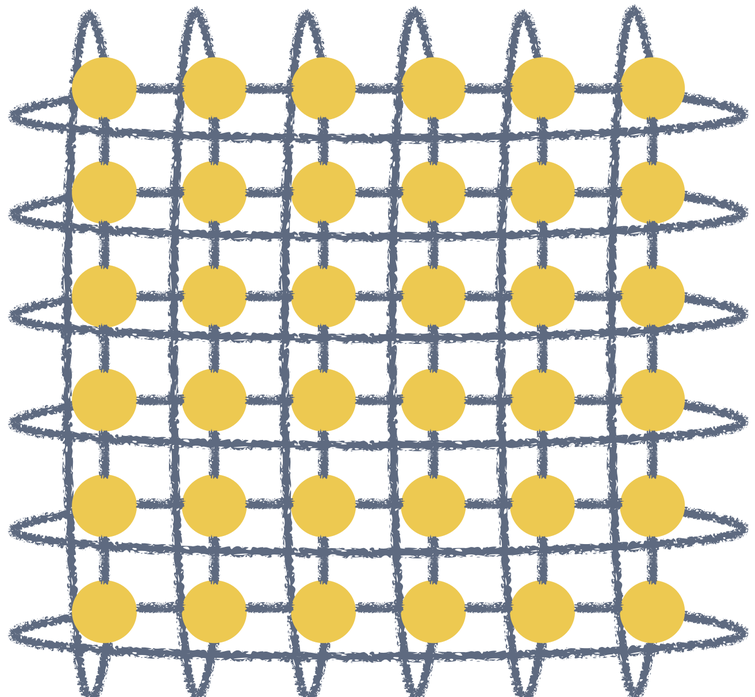
Diameter = P / 2
Average distance ~ P / 4
Bisection = = 2

2-dimensional mesh and torus

- **2-D mesh**: ~ 2*P links

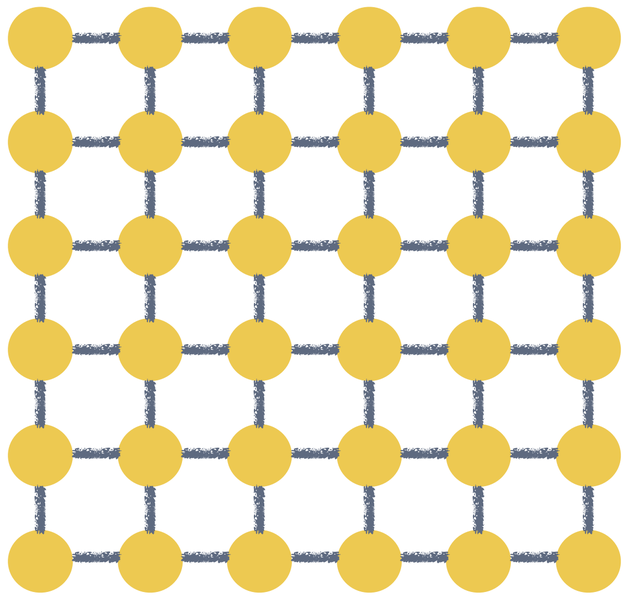


- **2-D Torus**: 2*P links



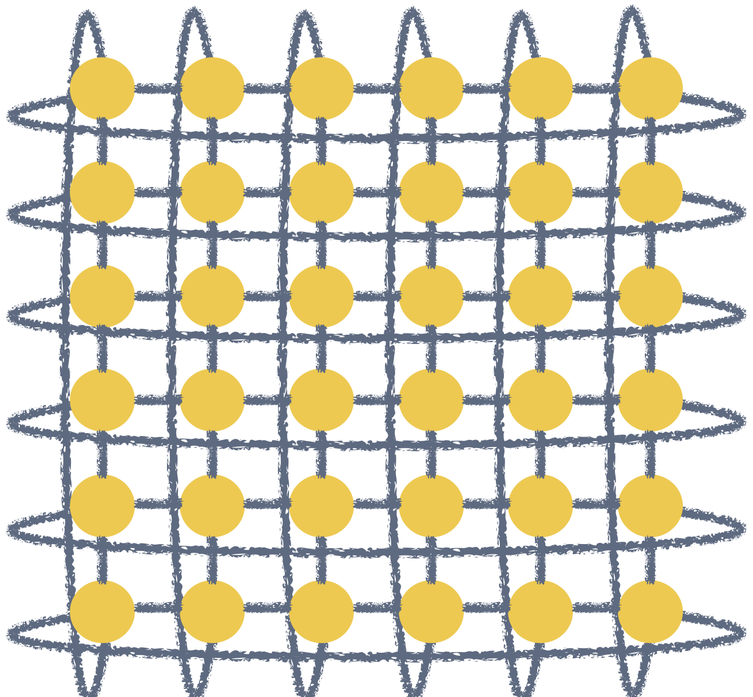
2-dimensional mesh and torus

- **2-D mesh**: ~ 2*P links



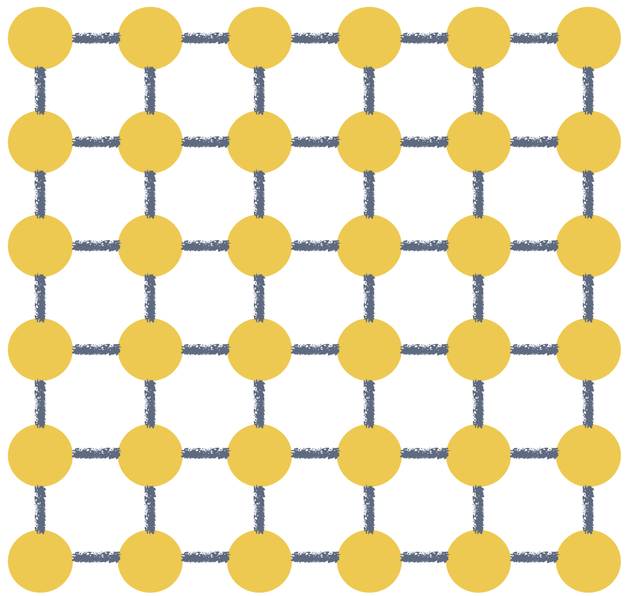
Diameter = ?
Bisection = ?

- **2-D Torus**: 2*P links



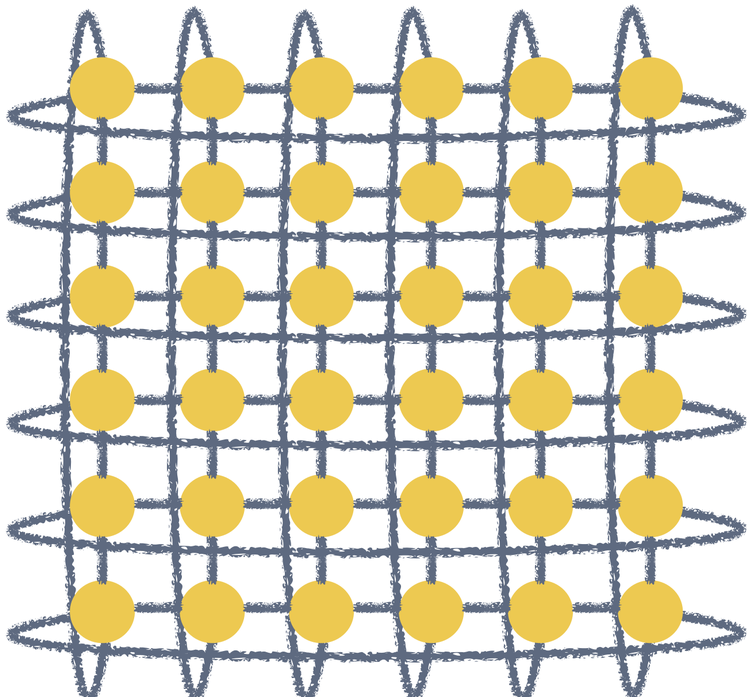
2-dimensional mesh and torus

- **2-D mesh**: ~ 2*P links



Diameter = ?
Bisection = ?

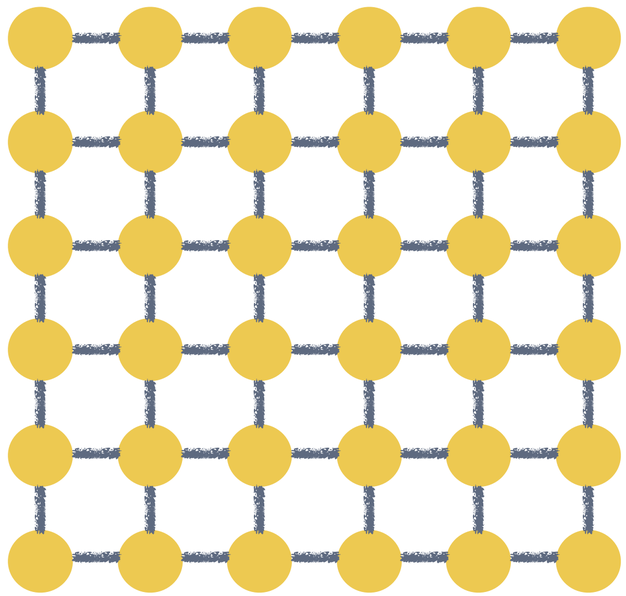
- **2-D Torus**: 2*P links



Diameter = ?
Bisection = ?

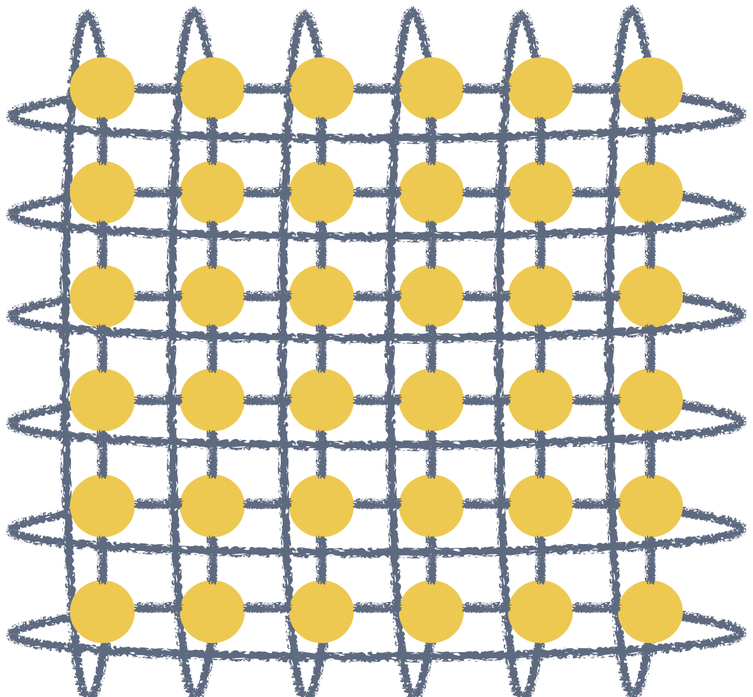
2-dimensional mesh and torus

- **2-D mesh**: ~ 2*P links



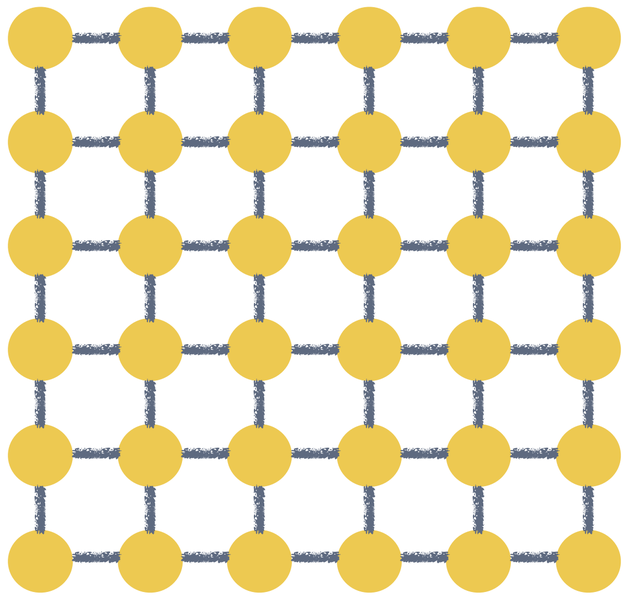
Diameter = 2 * (√P - 1)
Bisection = √P

- **2-D Torus**: 2*P links



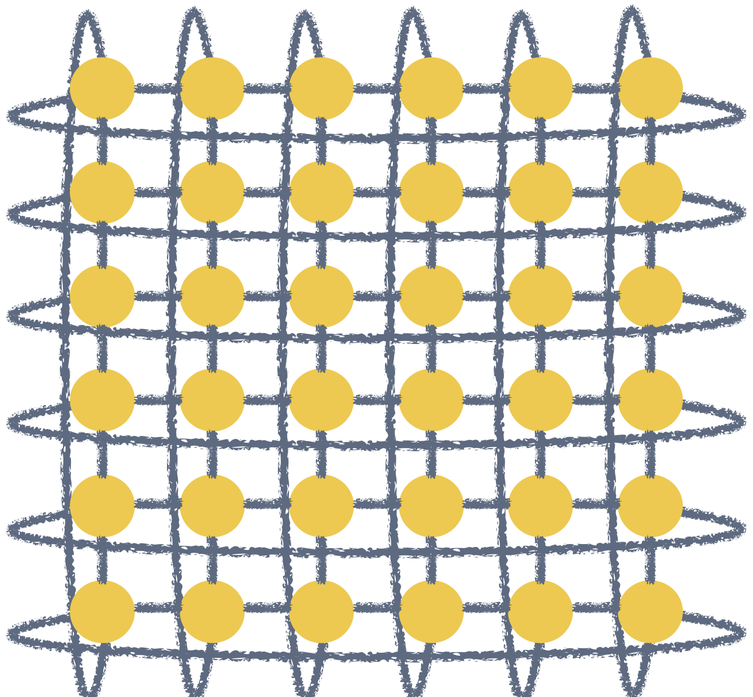
2-dimensional mesh and torus

- **2-D mesh**: ~ 2*P links



Diameter = 2 * (√P - 1)
Bisection = √P

- **2-D Torus**: 2*P links



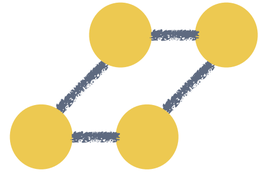
Diameter ~ √P
Bisection = 2 * √P

Hypercubes

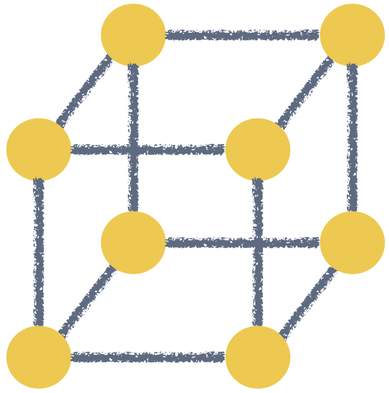
- **Number of nodes** = 2^d for dimension d

d = 0

d = 1



d = 2



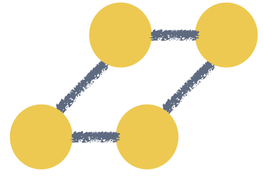
d = 3

Hypercubes

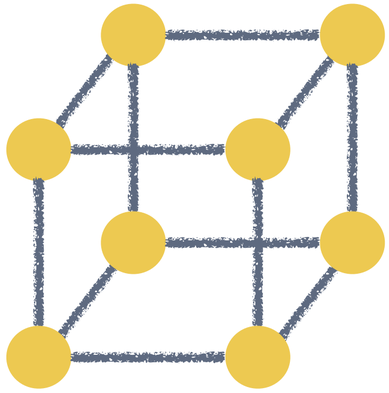
- **Number of nodes** = 2^d for dimension d

d = 0

d = 1



d = 2



d = 3

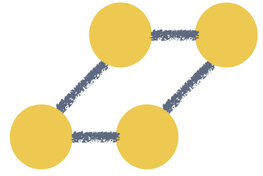
d = 4

Hypercubes

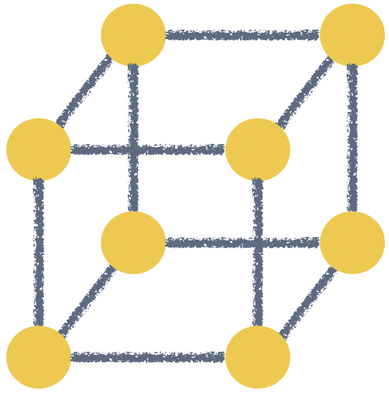
- **Number of nodes** = 2^d for dimension d

d = 0

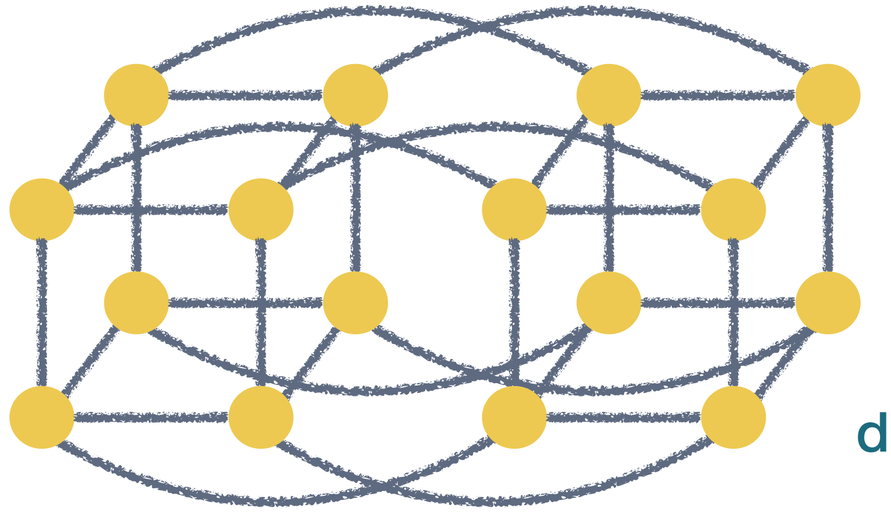
d = 1



d = 2



d = 3



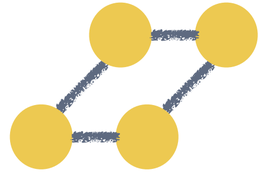
d = 4

Hypercubes

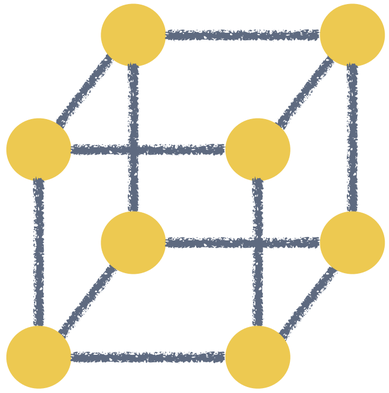
- **Number of nodes** = 2^d for dimension d

d = 0

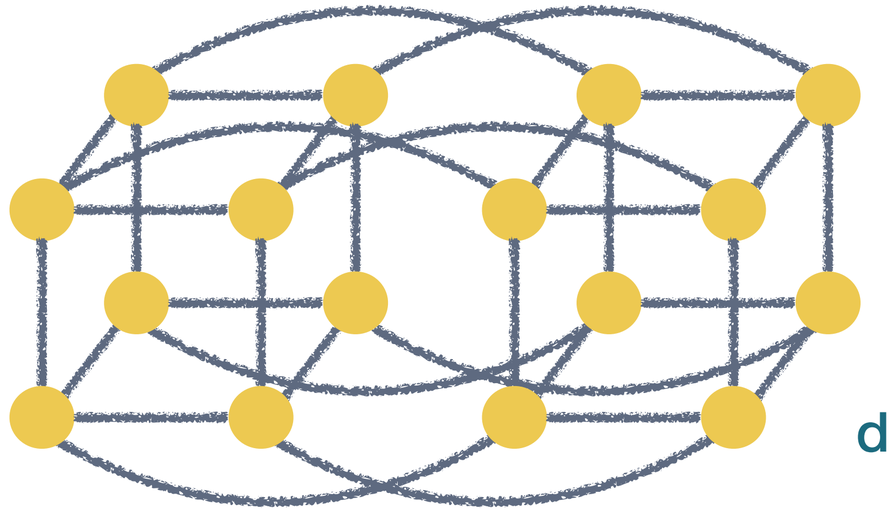
d = 1



d = 2



d = 3



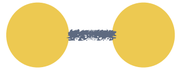
d = 4

Diameter = ?
Bisection = ?

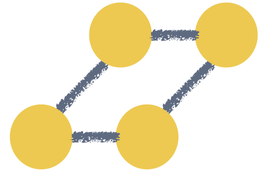
Hypercubes

- **Number of nodes** = 2^d for dimension d

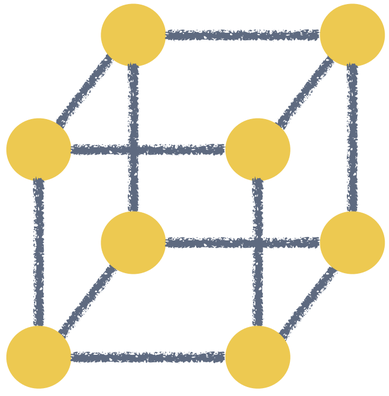
d = 0



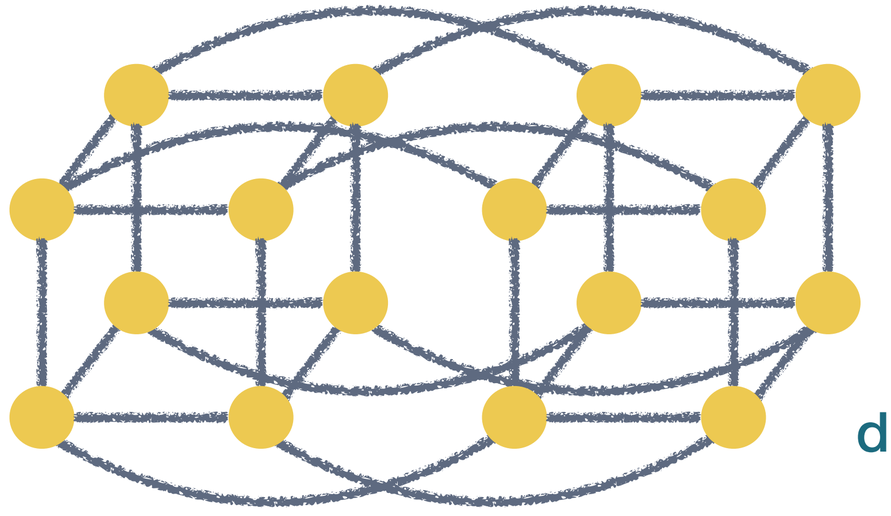
d = 1



d = 2



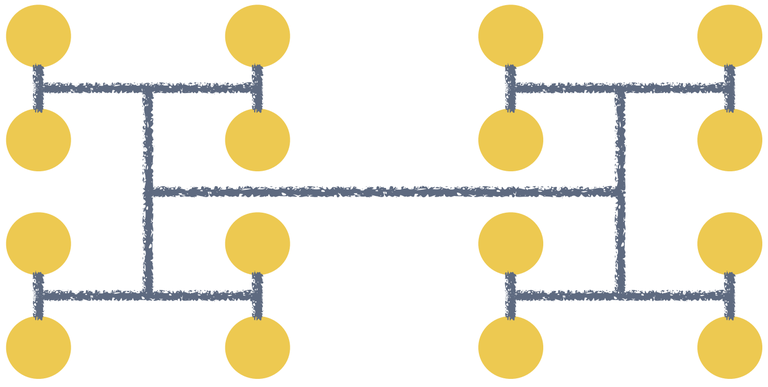
d = 3



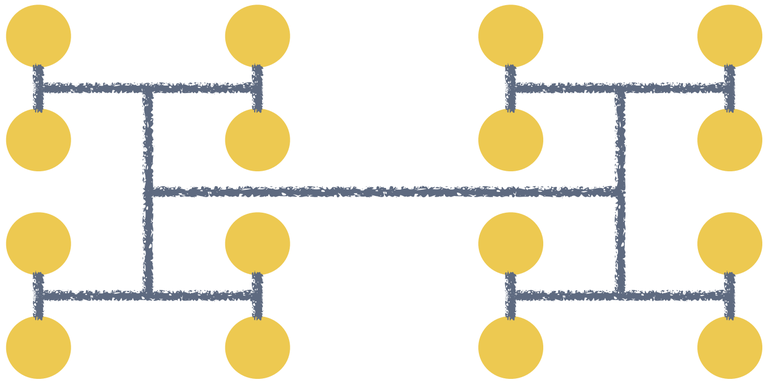
d = 4

Diameter = d
Bisection = P / 2

Trees

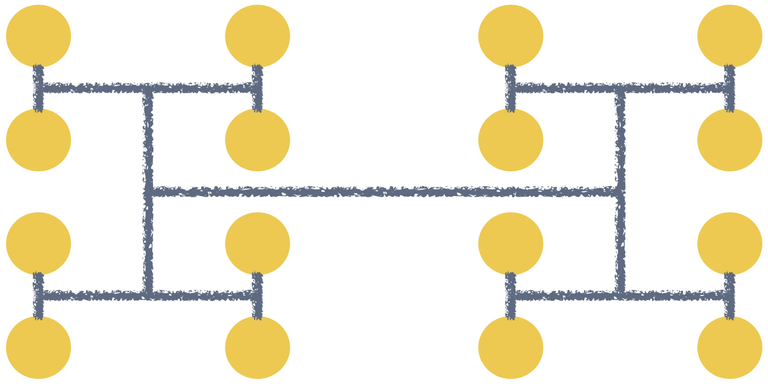


Trees



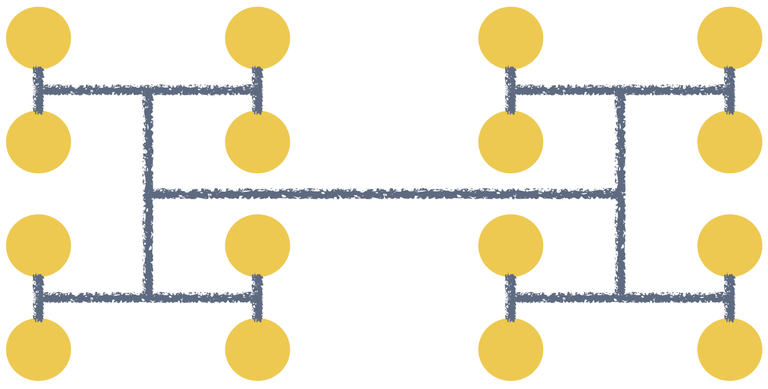
Diameter = ?
Bisection = ?

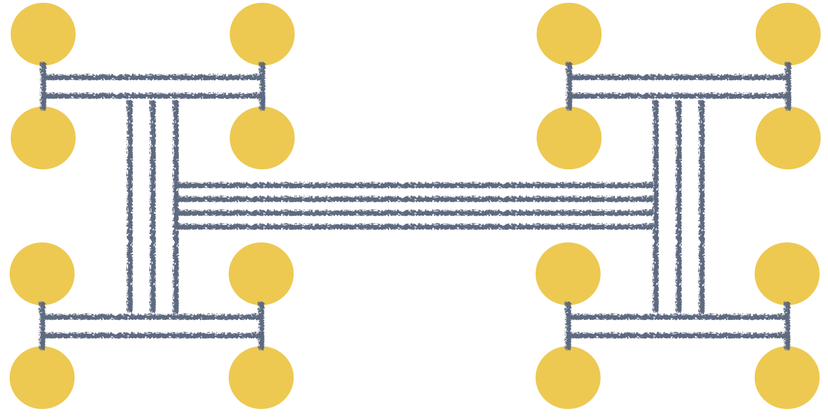
Trees



Diameter = log P
Bisection = 1

Trees

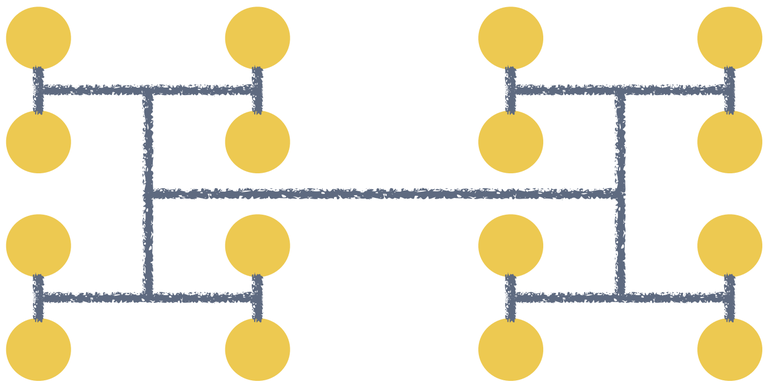


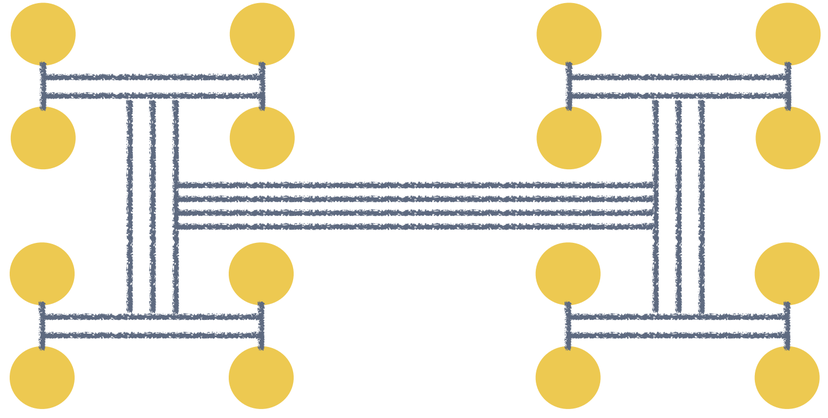


Diameter = log P
Bisection = 1

- **Fat trees**: more links at the top
- Avoids bisection bandwidth problem
- Natural for tree algorithms

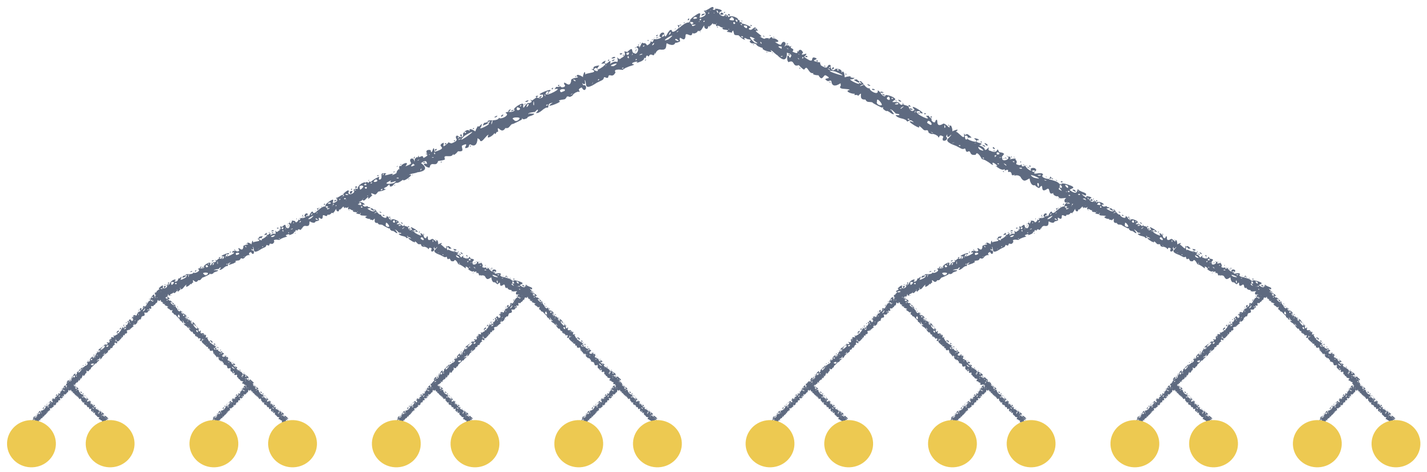
Trees



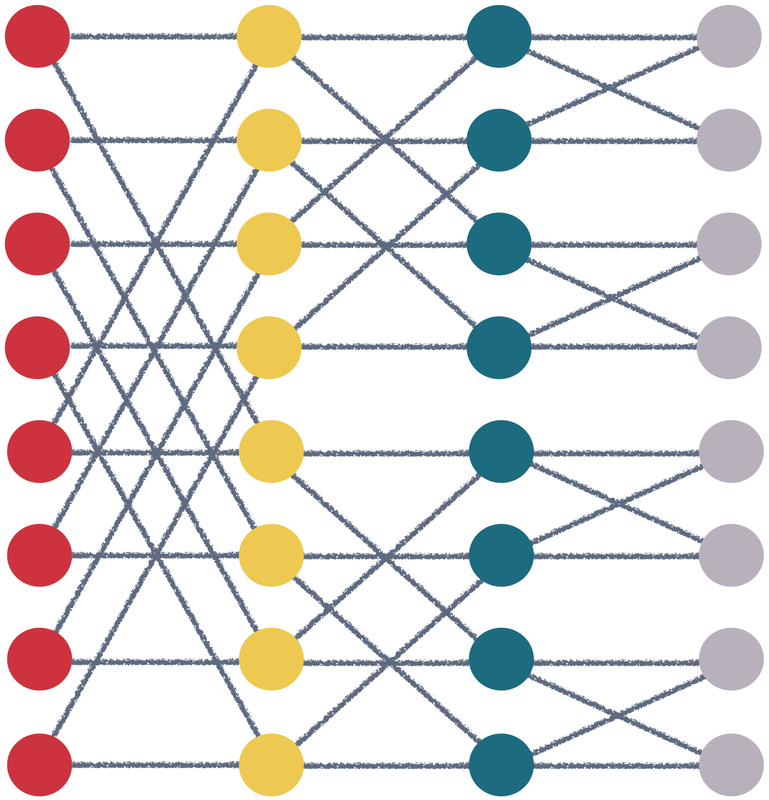


Diameter = log P
Bisection = 1

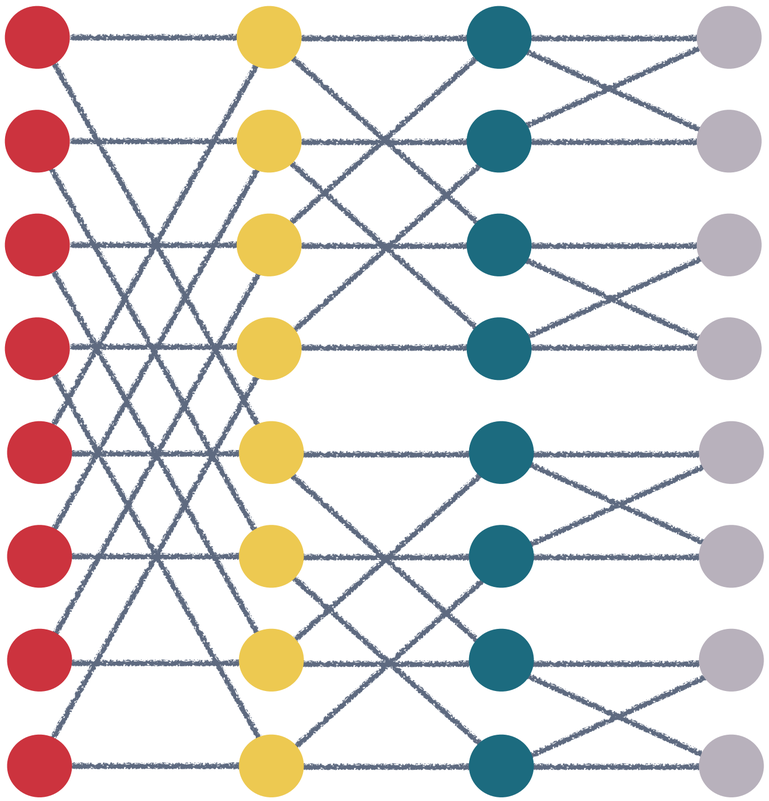
- **Fat trees**: more links at the top
- Avoids bisection bandwidth problem
- Natural for tree algorithms



Multistage butterfly network

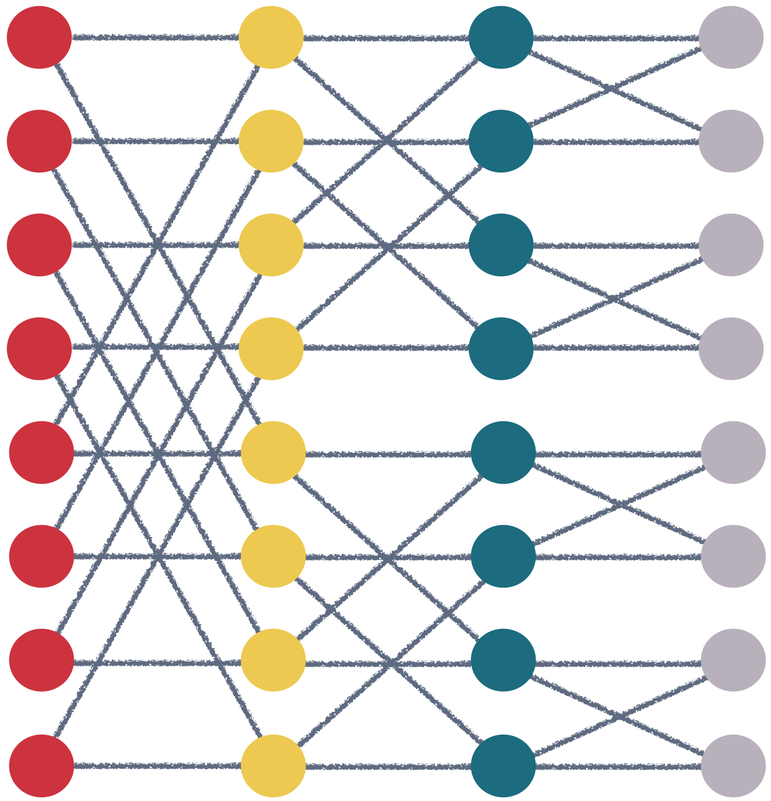


Multistage butterfly network



Diameter = ?
Bisection = ?

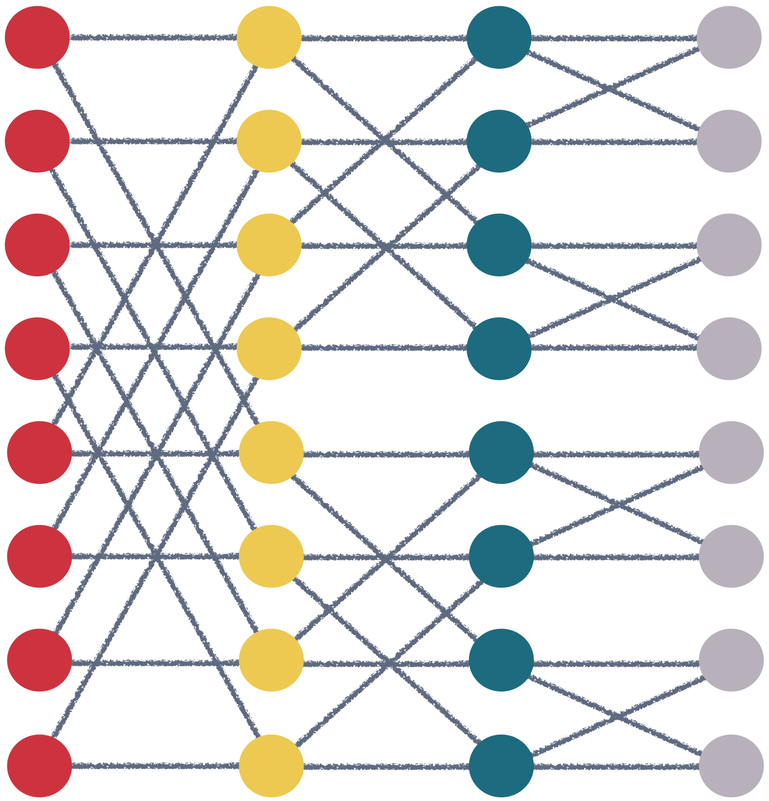
Multistage butterfly network



- Interior node is a switch
- Routing decision based on single bit of destination address
- Each stage looks at a different bit
- PxP network has log P stages of P/2 switches each

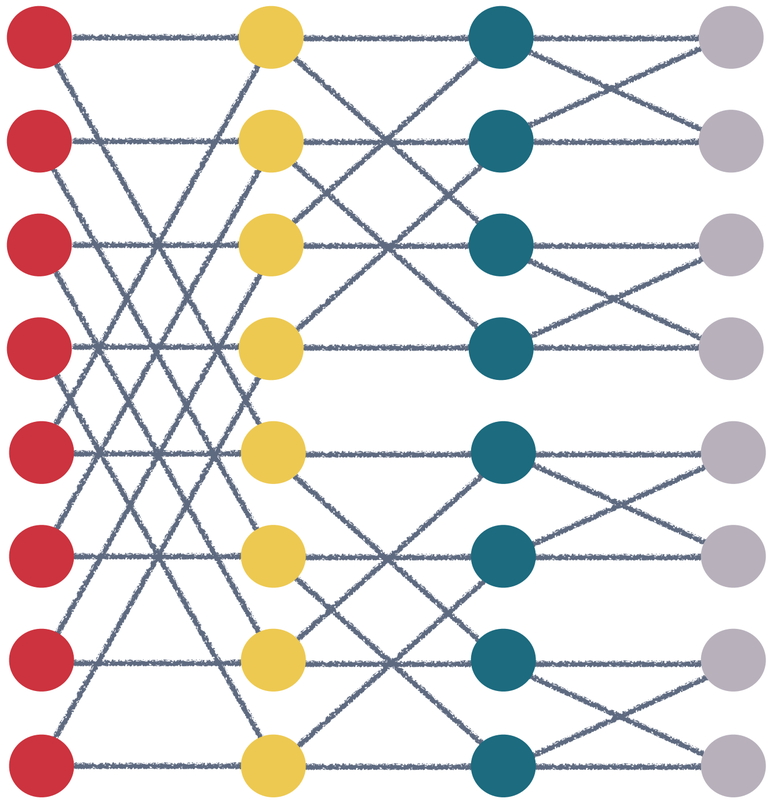
Diameter = ?
Bisection = ?

Multistage butterfly network



Diameter = log P
Bisection = P

Multistage butterfly network



Diameter = log P
Bisection = P

- Interior node is a switch
- Routing decision based on single bit of destination address
- Each stage looks at a different bit
- PxP network has log P stages of P/2 switches each
- **Cost**: lots of wires
- Natural for FFT

Summary of topology characteristics

Topology	Diameter	Bisection	Arc connectivity	# links
Linear	$p-1$	1	1	$p-1$
Ring	$p/2$	2	2	p
2-D mesh	$2(\sqrt{p}-1)$	$\sqrt{p}$	2	$2(p-\sqrt{p})$
2-D torus	$\approx \sqrt{p}$	$2\sqrt{p}$	4	$2p$
Hypercube	$\log p$	$p/2$	$\log p$	$p \log p/2$
Star	2	1	1	$p-1$
Butterfly	$\log p$	p	p	$\approx p \log p$
Completely connected	1	$p^2/4$	$p-1$	$p(p-1)/2$

Source: Grama, et al. (2003), Intro. to Parallel Computing

Topologies in practice

Newer

Older

Machine	Topology
IBM Blue Gene/Q	5D torus
K computer	6D torus
Tianhe-2	Fat tree
Tsubame	Fat tree
Cray XE6	3D torus
Cray XT3, XT4, XT5	3D torus
BG/L, BG/P	3D torus (+ others)
SGI Altix	Fat tree
Cray X1	4D hypercube
Millennium (UCB, Myricom)	Arbitrary
HP Alphaserver (Quadrics)	Fat tree
IBM SP	~ Fat tree
SGI Origin	Hypercube
Intel Paragon	2D mesh
BBN Butterfly	Butterfly