NoC-Topology Exploration

The LNM Institute of Information Technology

Sunil Kumar

sunil@lnmiit.ac.in

Topology Overview

- **Definition**: determines arrangement of links/channels and nodes in network

- Often first step in network design
  - It specify both type of network and associated details

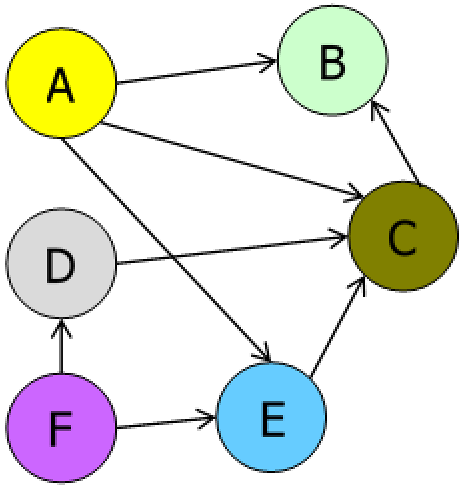
- Selection of a good topology consists in fitting the requirements in the available packaging technology

Topology Overview

- Significant impact on network cost-performance
  - Determines implementation complexity, i.e., **cost**
    - number of routers and links
    - router degree (i.e., ports)
    - ease of layout
  - Determines application **performance**
    - number of hops -> latency and energy consumption
    - maximum throughput

How to Select a Topology?

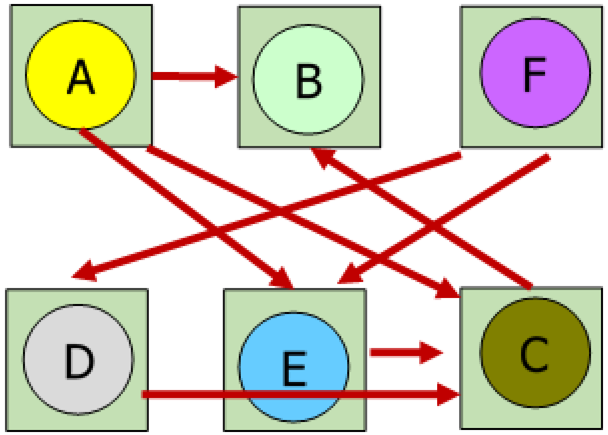
LNMIIT
The LNM Institute of Information Technology



Application's Task Communication Graph

Vertices - tasks

Edges - communication



Network Topology Graph

Vertices - cores

Edges – links

- Topology is fixed at design-time.
- Benefits to being regular and flexible

Problems?

- Cannot change algorithm
- Cannot change mapping
- Cannot adapt to data-dependent load
- imbalance in application
- Layout/packaging issue with long wires
- and high-node degree

Design Time Metrics

- **Switch Degree** – number of ports at a node
  - Proxy for area / energy cost
    - Higher degree requires more links and port counts at each router

- **Bisection Bandwidth** – bandwidth crosssing a minimal cut that divides the network in half
  - (Min # channels crossing two halves) * (BW of each channel)
  - Proxy for peak bandwidth
    - Can be misleading as it does not account for routing and flow control efficiency
    - At this stage, we assume **ideal routing** (perfect load balancing) and **ideal flow control** (no idle cycles on any channel)

- **Network Diameter** – maximum of minimum hop count/ routing distance between two nodes (number of links in *shortest* route)
  - Proxy for latency

Some Run-Time Metrics

- **Hop count (or routing distance)**
  - Number of hops between a communicating pair
  - Depends on application and mapping
  - *Average hop count* or *Average distance*: average hops across all valid routes

- **Channel load**
  - Number of flows passing through a particular link
  - Depends on application, mapping, routing and flow control as much as on the topology
  - **Maximum Channel Load**: Estimated max bandwidth the network can support / Max bits per second (bps) that can be injected by every node before it saturates
  - Maximum channel load determines throughput
    - The *throughput* is the data rate in bits per second that the network accept per input port

- **Path diversity**
  - Number of shortest paths between a communicating pair
  - Can be exploited by routing algorithm
  - Provides fault tolerance

Network Latency

- Time required for a packet to traverse the network
  - Start: head arrives at input port
  - End: tail departs output port

- Latency = Head latency + serialization latency
  - Serialization latency: time for packet with Length L to cross channel with bandwidth b (L/b)

- Approximate with hop count
  - Other design choices (routing, flow control) impact latency
    - Unknown at this stage

Network Latency

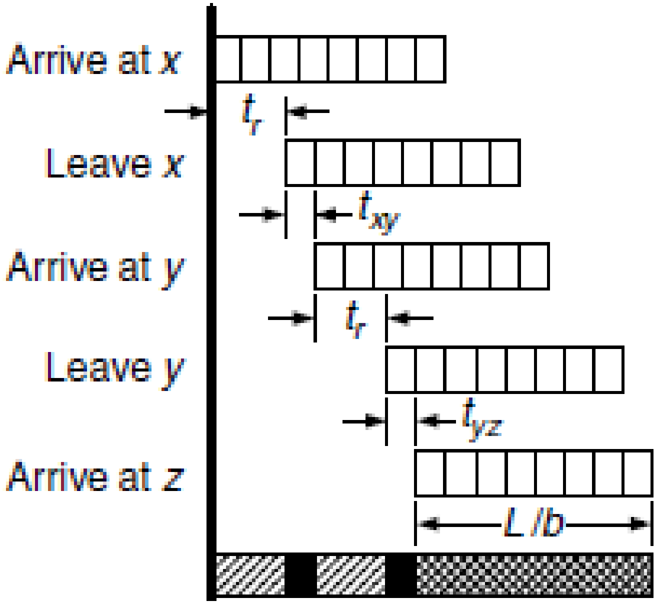
- T = H.tr + Tw + Ts + Tc
  - H = number of hops
  - tr = router delay
  - Tw = wire delay
  - Ts = serialization delay
  - Tc = contention delay

- T = H.tr + D / v + L / b + Tc
  - D = wire distance
  - v = propagation velocity
  - L = packet length
  - b = channel bandwidth

Example

- Packet propagating on a *two-hop* route from node *x* to node *z*, via node *y*
  - First row: each phit of the packet arriving at node *x*
  - Second row: leaving *x* (routing delay *tr)*
  - Third row: arriving at *y* (link latency *txy)*
  - Fourth row: leaving y (second routing delay *tr)*
  - Fifth row: arriving z (link latency *tyz)*
  - At this head latency the serialization latency should be added (L/b)

- 64-node network with *Havg*=4 hops and 16-bit wide channel
  - The frequency *fc* =1GHz, *tc*=5ns and *tr*=8ns
  - Total routing delay 32ns (8*4)
  - Total wire delay is 20ns (5*4)
  - If L=64bytes, and b=2Gbytes/s, serialization delay is equal to 32ns
  - Total latency is 84ns

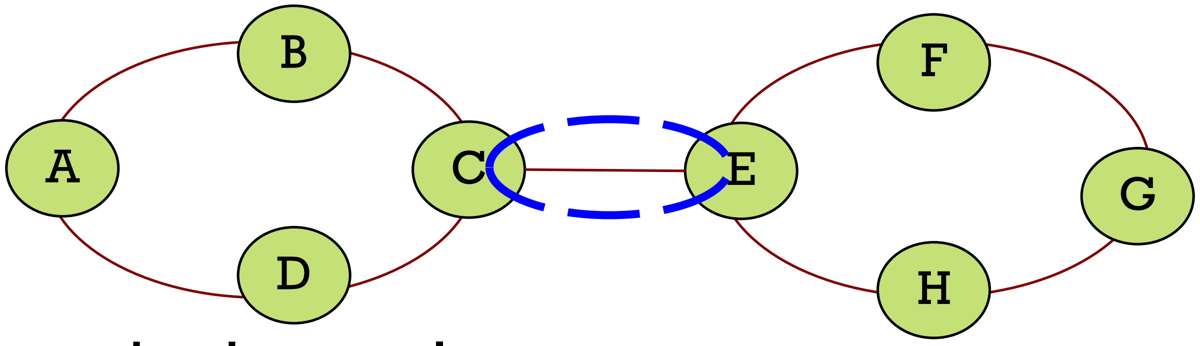


Maximum Channel Load

- Identify channel with maximum traffic
  - Count total flows through it

- Maximum Throughput = 1 / (max channel load)

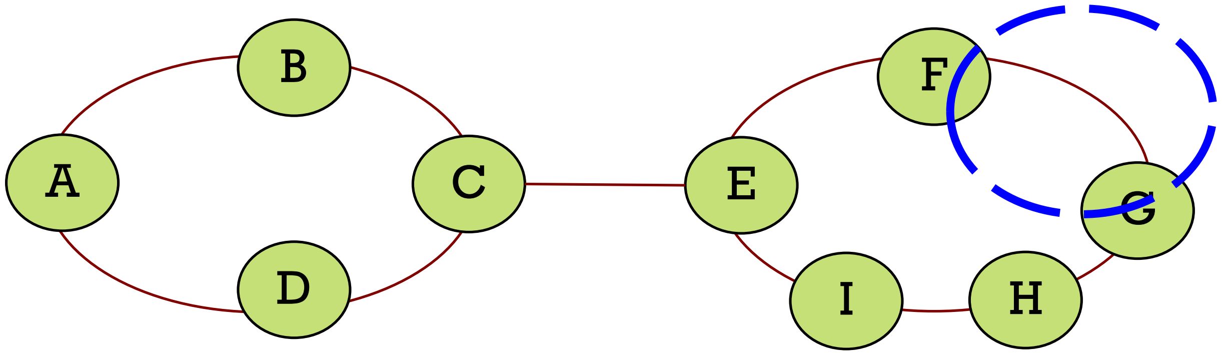
Maximum Channel Load



- Identify bottleneck channel
  - For uniform random traffic, is the bisection channel

- Suppose each node generates p messages per cycle
  - 4p messages per cycle in left ring
  - 2p message per cycle will cross to other ring
  - Link can handle one message per cycle
  - So maximum injection rate of p = ½

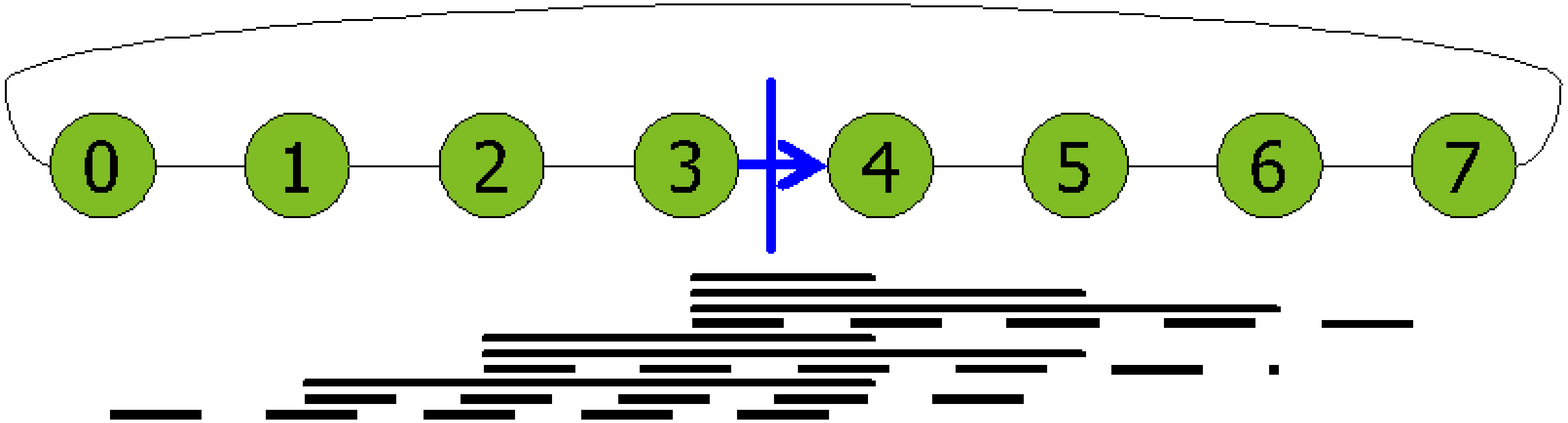
Maximum Channel Load



- What if Hot Spot Traffic?
  - Suppose every node sends to node G

- Which is the bottleneck channel?
  - Used by A, B, C, D, E, and F to send to G
  - Max Throughput = 1 / 6

Maximum Channel Load



- With uniform random traffic
  - 3 sends 1/8 of its traffic to 4,5,6
  - 3 sends 1/16 of its traffic to 7 (2 possible shortest paths)
  - 2 sends 1/8 of its traffic to 4,5
  - Etc
- Max Channel load = 1

Path Diversity

- Multiple minimum length paths between source and destination pair
- Fault tolerance
- Better load balancing in network
- Routing algorithm should be able to exploit path diversity
- We'll see shortly
  - Butterfly has no path diversity
  - Torus can exploit path diversity

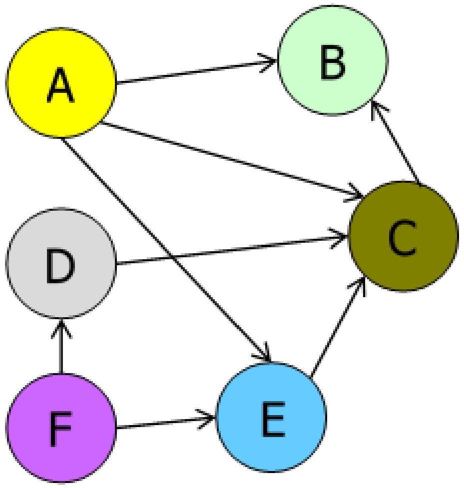
Path Diversity (2)

- Edge disjoint paths: no links in common
- Node disjoint paths: no nodes in common except source and destination
- If j = minimum number of edge/node disjoint paths between any source-destination pair
  - Network can tolerate j link/node failures

Symmetry

- Vertex symmetric
  - An automorphism exists that maps any node ‘a’ onto another node ‘b’
  - Topology same from point of view of all nodes
- Edge symmetric
  - An automorphism exists that maps any channel ‘a’ onto another channel ‘b’

Regular Topology

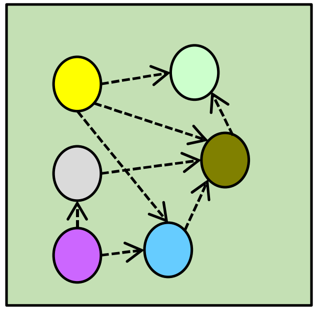


Application's Task Communication Graph

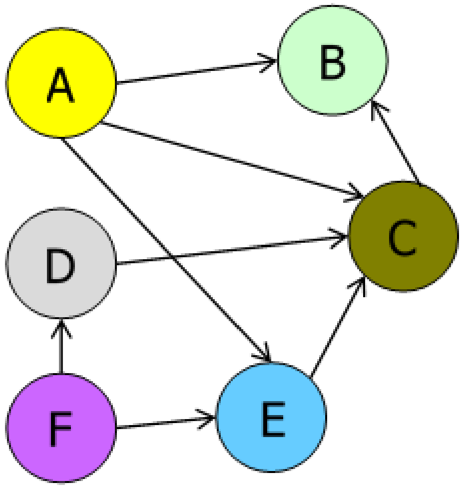
Vertices - tasks

Edges - communication

- Can you suggest a regular topology (each router with same degree) with smallest possible diameter?
- **Trick question :p**
  - One node. Degree = 0, Diameter = 0.



Regular Topology

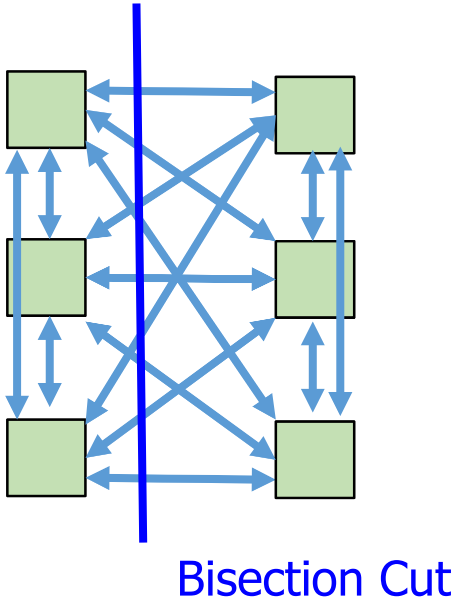


Application's Task Communication Graph

Vertices - tasks

Edges - communication

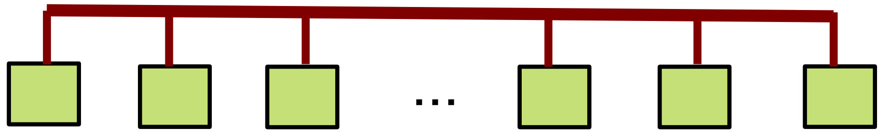
- Can you suggest a regular topology (each router with same degree) with diameter = 1?



Challenge?
Not scalable!!
Cannot layout more than 4-6 cores in this manner for area and power reasons

Fully Connected
Degree = ? 5
Bisection BW = ? 9

Bus Topology



- Diameter = ? 1
- Degree = ? 1
- Bisection BW = ? 1

- **Pros**
  - Cost-effective for small number of nodes
  - Easy to implement snoopy coherence
  - Most multicores with 4-6 cores use Buses
- **Cons**
  - Bandwidth! -> Not scalable

Popular Bus Protocols

- ARM AMBA Bus
  - AHB
  - AXI
  - ACE
  - CHI
- IBM Core Connect
- ST Microelectronics STBus

- How to increase bus bandwidth?
  - Hierarchical Buses
  - Split-buses

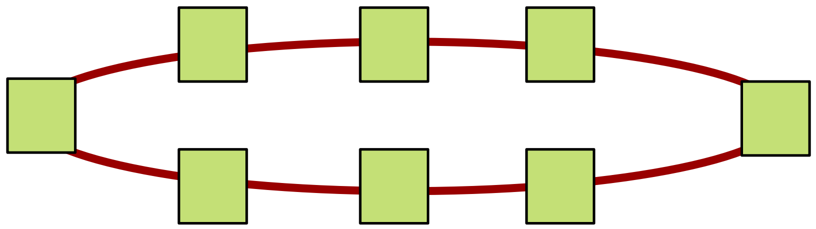
Topology Classification

- **Direct**
  - Every switch/router also network end point
  - All routers are sources and destinations of traffic
  - Example: Ring, Mesh, Torus
  - Most on-chip networks use direct topologies

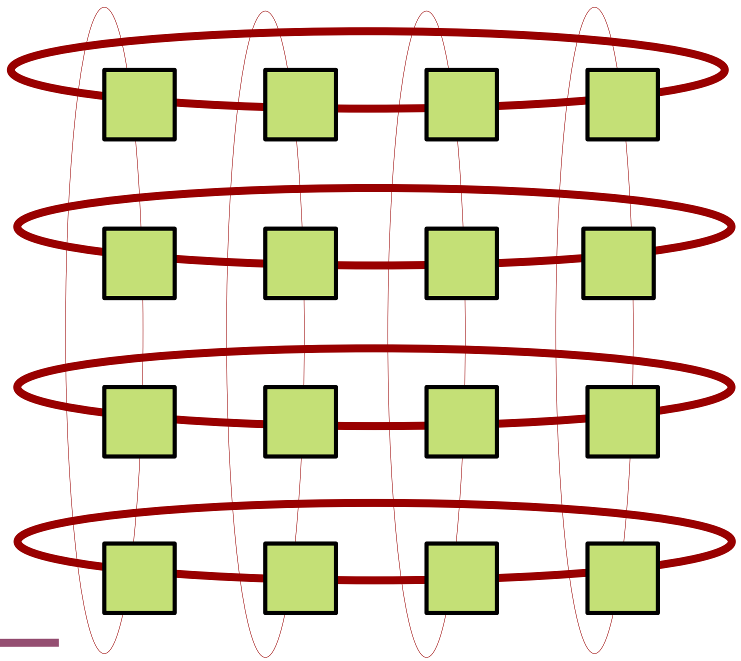
- **Indirect**
  - Not all switches/routers are end points
  - Terminal nodes can source / sink traffic
  - Intermediate nodes switch traffic
  - Examples: Crossbar, Butterfly, Clos, Omega, Benes, …

Torus

- Formally: k-ary n-cube
  - K^n network nodes
  - n-dimensional grid with k nodes in each dimension



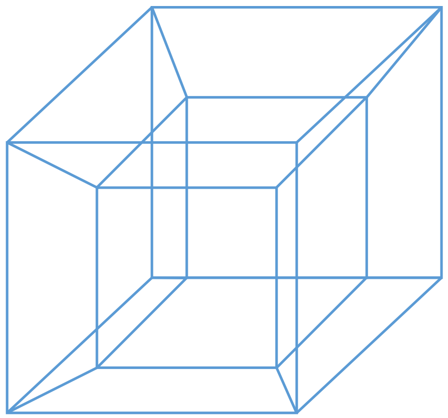
8-ary 1-cube



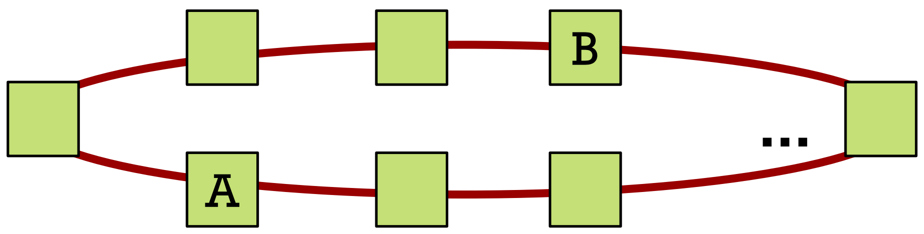
4-ary 2-cube

Torus

- Topologies in Torus Family
  - Ring k-ary 1-cube
  - Hypercubes 2-ary n-cube
- Edge Symmetric
  - Good for load balancing
  - Removing wrap-around links for mesh loses edge symmetry
    - More traffic concentrated on center channels
- Good path diversity
- Exploit locality for near-neighbor traffic



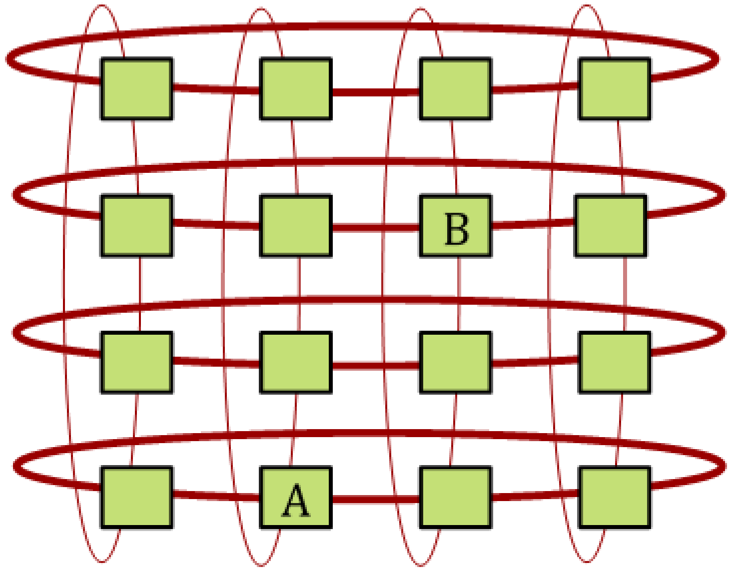
Ring



- Diameter? N/2
- Avg Distance? N/4
- Bisection BW? 2
- Degree? 2

- **Pros**
  - Cheap: O(N) cost
  - Used in most multicores today
- **Cons**
  - High latency
  - Difficult to scale - bisection bandwidth remains constant
  - No path diversity
  - 1 shortest path from A to B

Torus



- Diameter? $\sqrt{N}$
- Bisection BW? $2\sqrt{N}$
- Degree? 4

- Pros
  - O(N) cost
  - Exploit locality for near neighbor traffic
  - High path diversity
    - 6 shortest paths from A to B
  - Edge symmetric
    - good for load balancing
    - Same router degree
- Cons
  - Unequal link lengths
  - Harder to layout

Channel Load for Torus

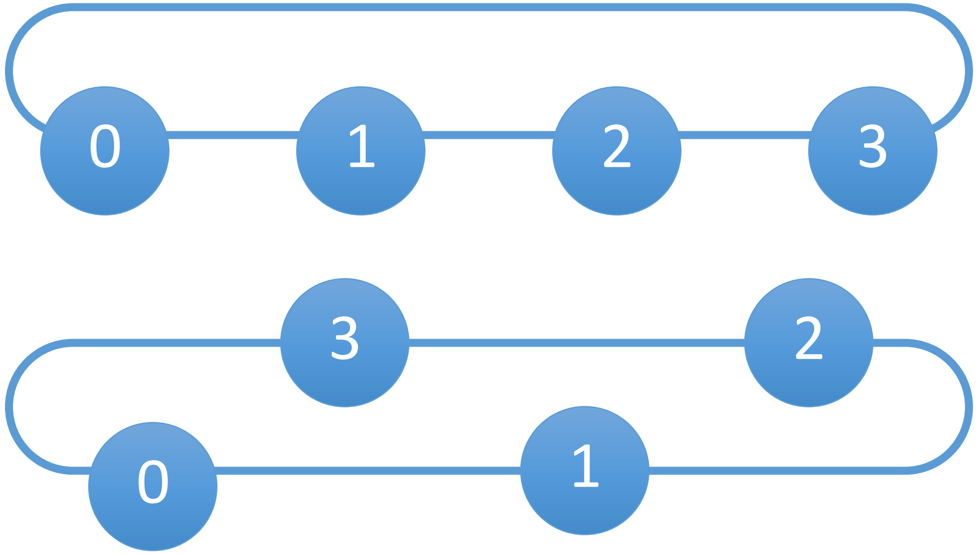
- Even number of k-ary (n-1)-cubes in outer dimension
- Dividing these k-ary (n-1)-cubes gives a 2 sets of k^{n-1} bidirectional channels or $4k^{n-1}$
- ½ Traffic from each node cross bisection

$$channelload = \frac{N}{2} \times \frac{k}{4N} = \frac{k}{8}$$

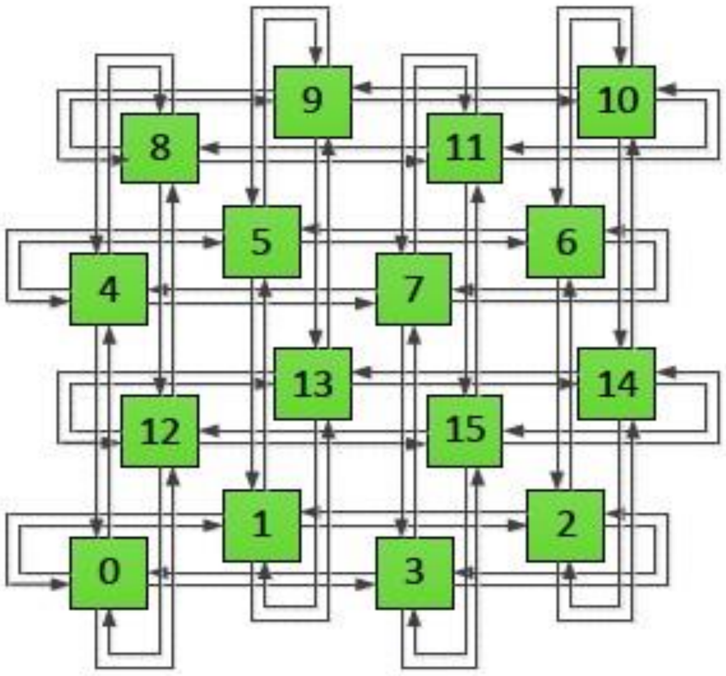
- Mesh has ½ the bisection bandwidth of torus

Implementation

- Folding
  - Equalize path lengths
    - Reduces max link length
    - Increases length of other links

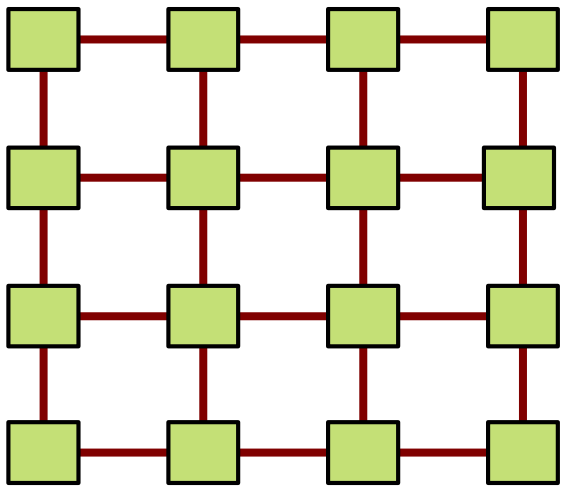


Folded Torus



- Easier to layout
- Is there any con compared to the mesh?
  - All channels have double the length

Mesh



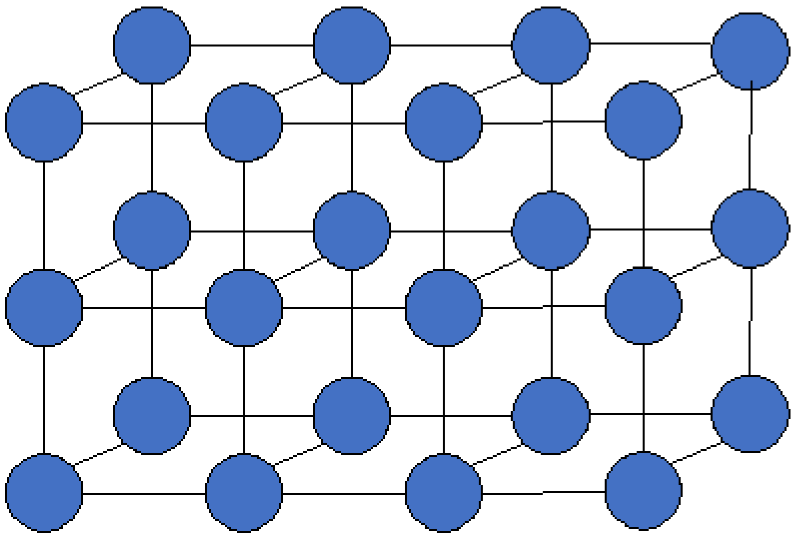
- Diameter? 2(√N-1)
- Bisection BW? √N
- Degree? 4

- Pros
  - O(N) cost
  - Easy to layout on-chip: regular and equal-length links
  - Path diversity
  - 3 shortest paths from A to B

- Cons
  - Not symmetric on edges
    - Performance sensitive to placement on edge vs. middle
    - Different degrees for edge vs. middle routers
  - Blocking, i.e., certain paths can block others (unlike crossbar)

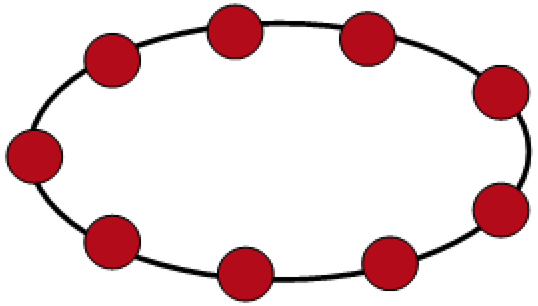
Multi-Dimensional Topologies

- Used in Supercomputers, Datacenters, and other off-chip System Area Networks

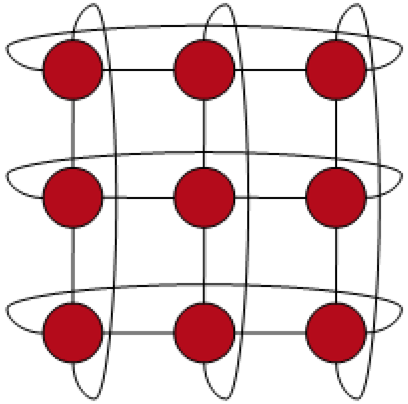


2,3,4-ary 3 Mesh

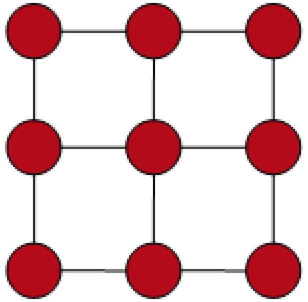
Hop Count



9-ary 1 cube



3-ary 2 cube



3-ary 2 mesh

	9-ary 1 cube	3-ary 2 cube	3-ary 2 mesh
Max =	4	2	4
Avg =	2.22	1.33	1.77

k-ary n cube

$$H_{avg} = \begin{cases} \frac{nk}{4} & k \text{ even} \\ n\left(\frac{k}{4} - \frac{1}{4k}\right) & k \text{ odd} \end{cases}$$

k-ary n mesh

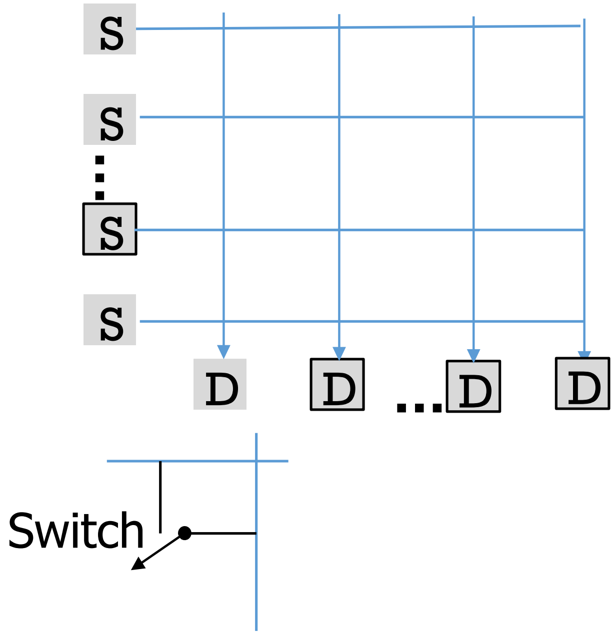
$$H_{avg} = \begin{cases} \frac{nk}{3} & k \text{ even} \\ n\left(\frac{k}{3} - \frac{1}{3k}\right) & k \text{ odd} \end{cases}$$

Topology Classification

- **Direct**
  - Every switch/router also network end point
  - All routers are sources and destinations of traffic
  - Example: Ring, Mesh, Torus
  - Most on-chip networks use direct topologies

- **Indirect**
  - Not all switches/routers are end points
  - Terminal nodes can source / sink traffic
  - Intermediate nodes switch traffic
  - Examples: Crossbar, Butterfly, Clos, Omega, Benes, …

Crossbar

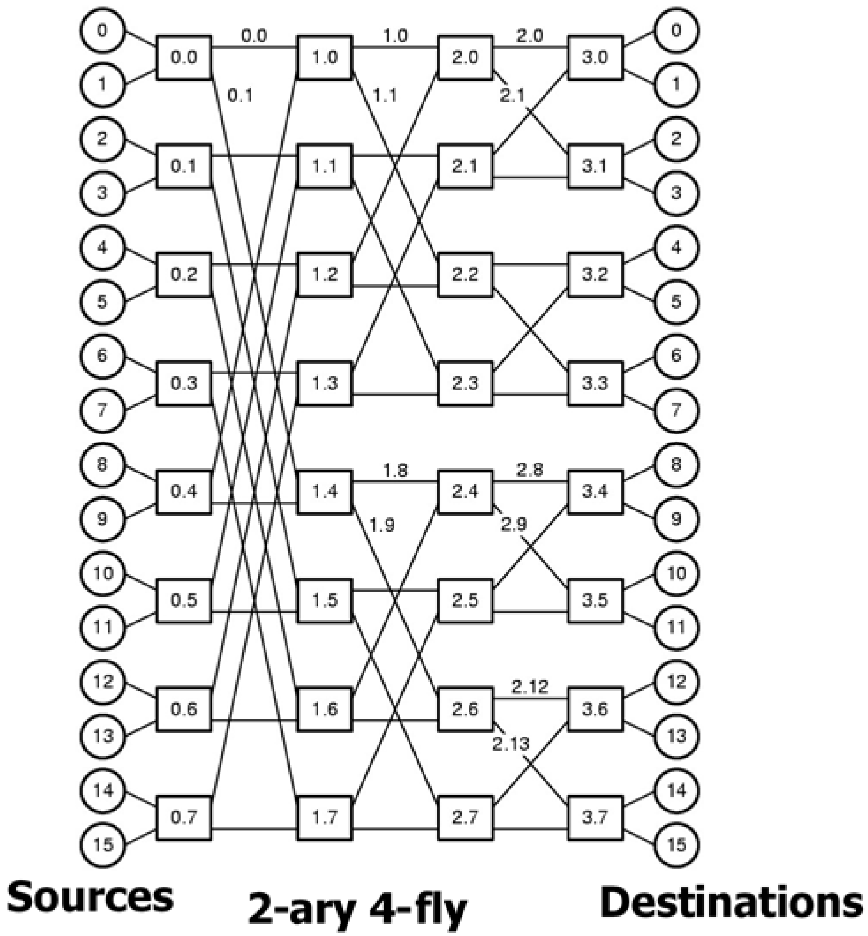


- Diameter = ? 1
- Degree = ? 1
- Bisection BW = ? N

- Pros
  - Every node connected to all others (**non-blocking**)
  - Low latency and high bandwidth
  - Used by GPUs

- Cons
  - Area and Power goes up quadratically (O(N2) cost)
  - Expensive to layout
  - Difficult to arbitrate

Butterfly (k-ary n-fly)



As a convention, source and destination nodes drawn logically separate on the left and right, though physically the two 0s, two 1s, etc are often the same physical node.

Radix of each switch = k

(i.e., k inputs and k outputs)

Number of stages = n

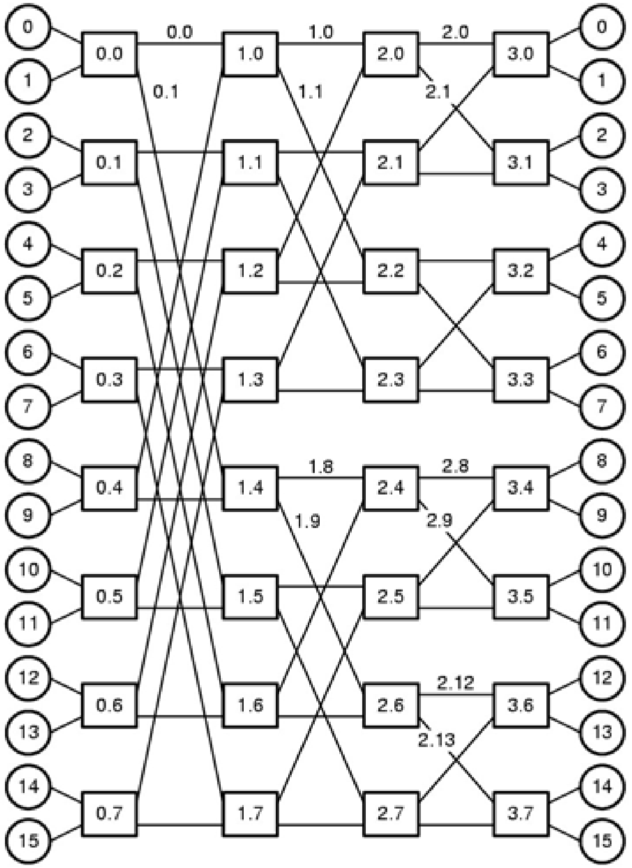
Total Source/Destination

Terminal Nodes = k^n

In each stage, k^{n-1} switches

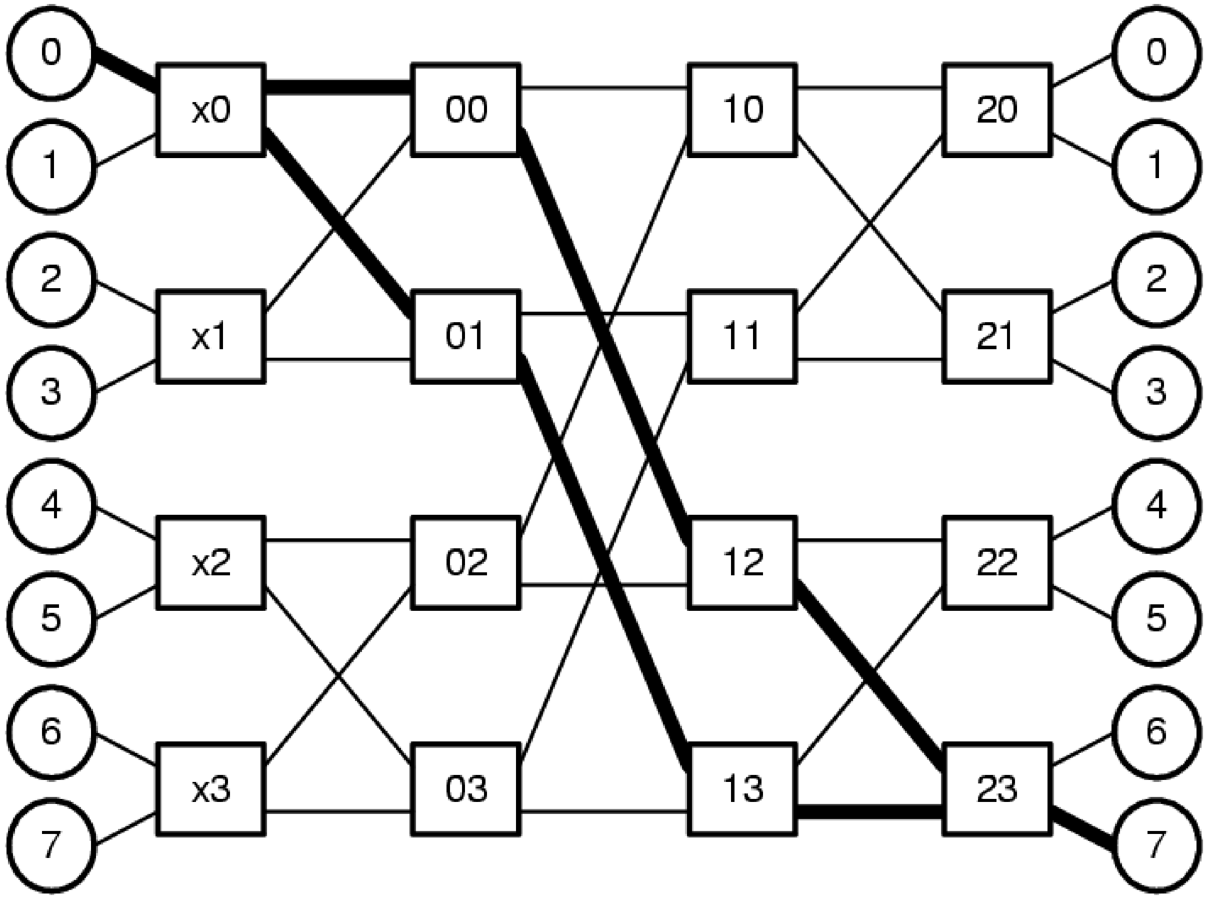
Each switch is a k x k crossbar

Butterfly (k-ary n-fly)

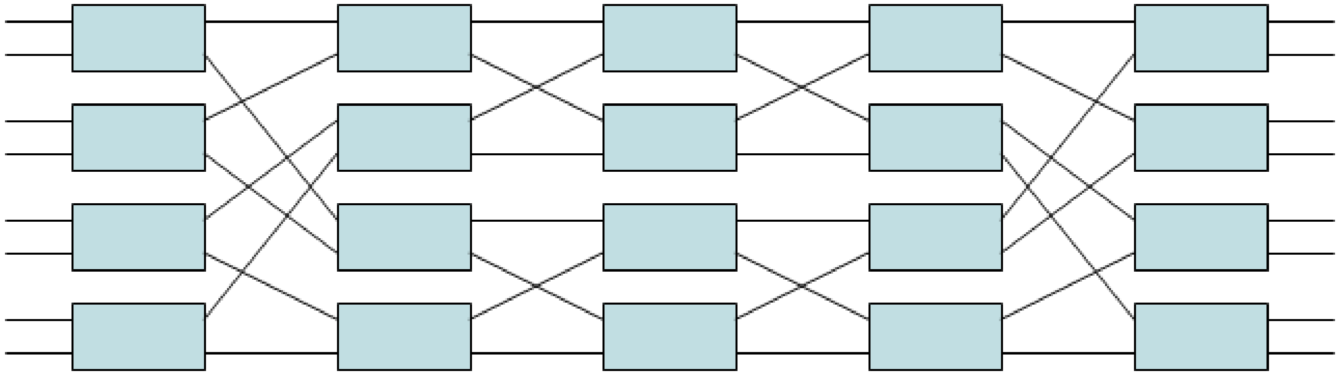


Degree?	k
Diameter?	n+1
Bisection Bandwidth?	N/4 where N = k^n
Hop Count?	n+1
Channel Load? (for uniform traffic)	1
Path Diversity?	None only one route between any pair

Tackling Path Diversity in a butterfly



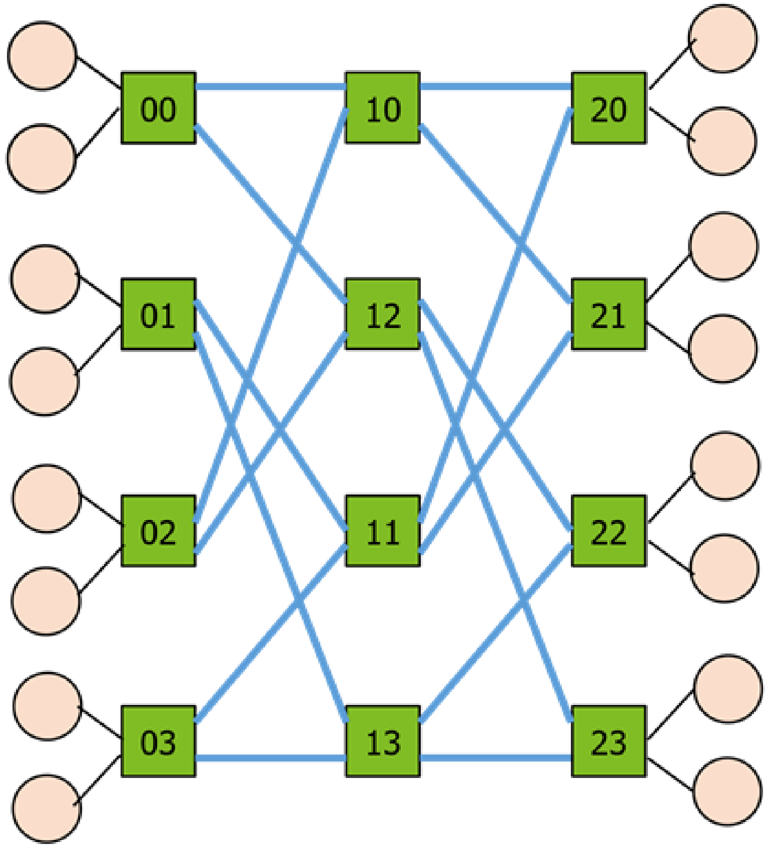
Benes Network



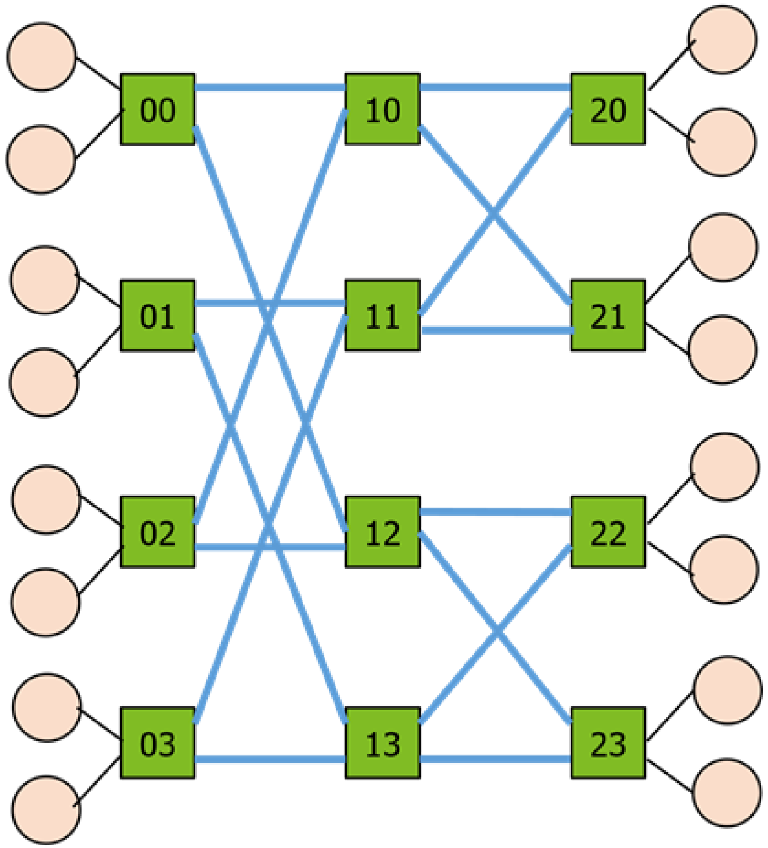
- Back to back butterflies
- N-alternate paths between any pair
- Is non-blocking

Shuffle/ Omega Network

Isomorphic Butterfly



Shuffle Network



2-ary 3-fly

Butterfly (2)

- No path diversity $\left|R_{xy}\right| = 1$
- Hop Count
  - $\mathrm{Log}_k n + 1$
  - Does not exploit locality
    - Hop count same regardless of location
- Switch Degree = 2k
- Channel Load → uniform traffic

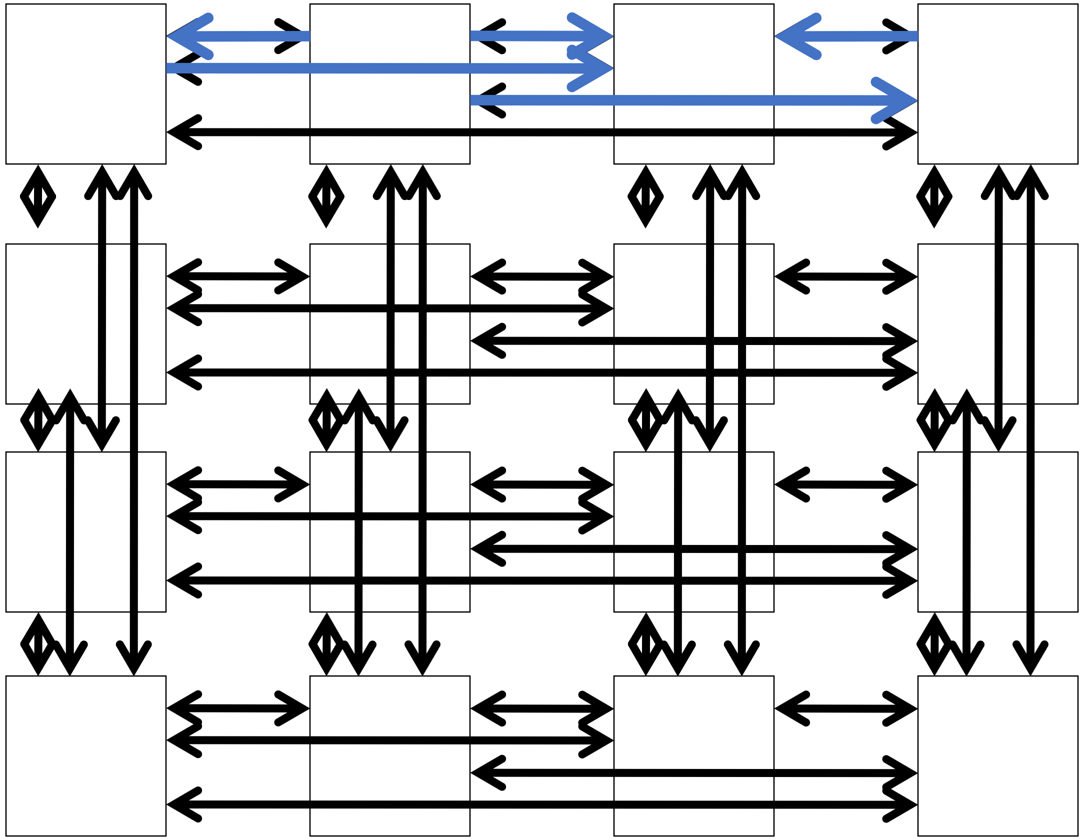
$$\frac{NH_{\min}}{C} = \frac{k^n(n+1)}{k^n(n+1)} = 1$$

  - Increases for adversarial traffic

Flattened Butterfly

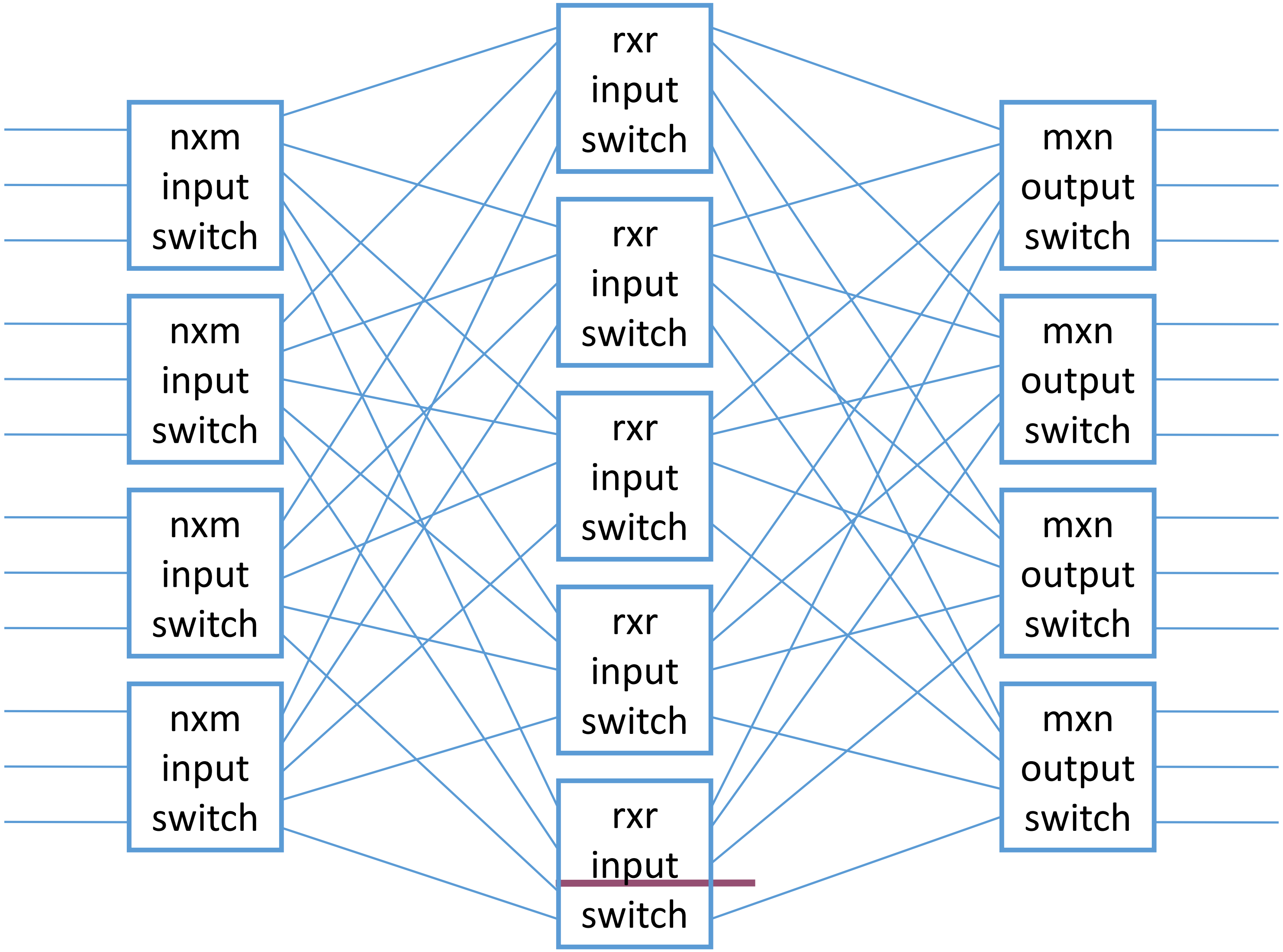
- Proposed by Kim et al (ISCA 2007)
  - Adapted for on-chip (MICRO 2007)
- Advantages
  - Max distance between nodes = 2 hops
  - Lower latency and improved throughput compared to mesh
- Disadvantages
  - Requires higher port count on switches (than mesh, torus)
  - Long global wires
  - Need non-minimal routing to balance load

Flattened Butterfly



- Path diversity through non-minimal routes

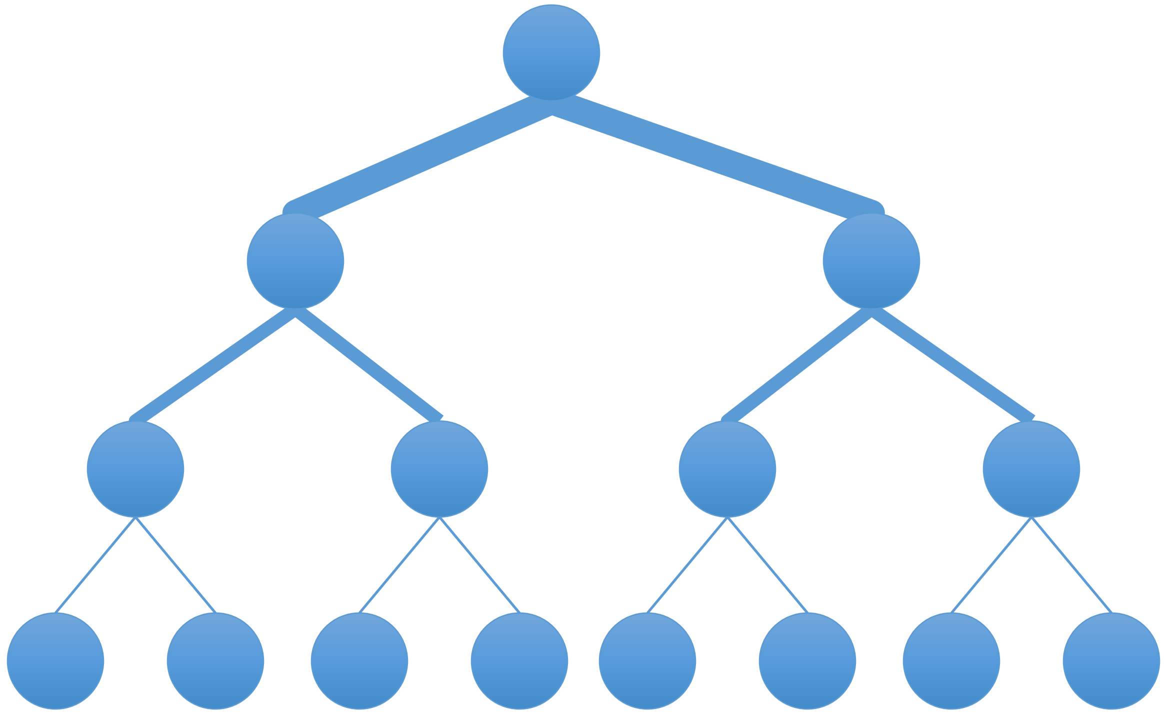
Clos Network



Clos Network

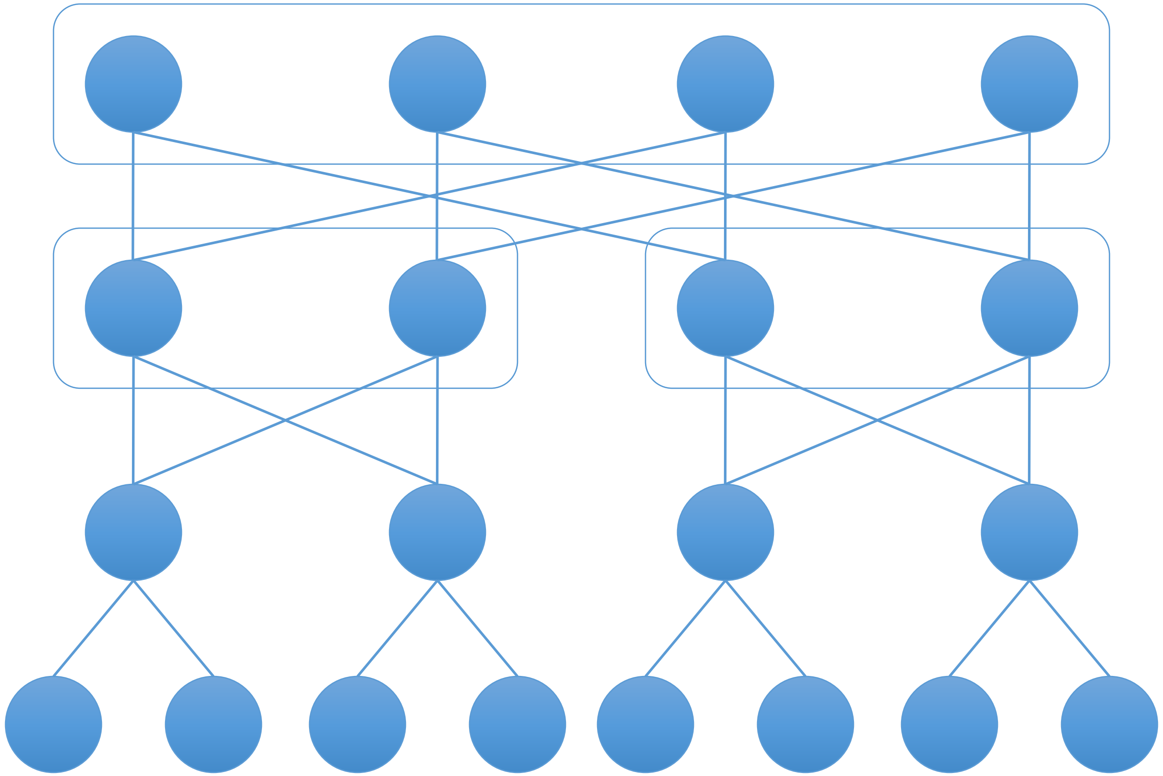
- 3-stage indirect network
- Characterized by triple (m, n, r)
  - M: # of middle stage switches
  - N: # of input/output ports on input/output switches
  - R: # of input/output switching
- Hop Count = 4

Folded Clos (Fat Tree)



- Bandwidth remains constant at each level
- Regular Tree: Bandwidth decreases closer to root

Fat Tree (2)



- Provides path diversity

Common On-Chip Topologies

- Torus family: mesh, concentrated mesh, ring
  - Extending to 3D stacked architectures
  - Favored for low port count switches
- Butterfly family: Flattened butterfly

Topology Summary

- First network design decision
- Critical impact on network latency and throughput
  - Hop count provides first order approximation of message latency
  - Bottleneck channels determine saturation throughput