Topologies

Maurizio Palesi

Network Topology

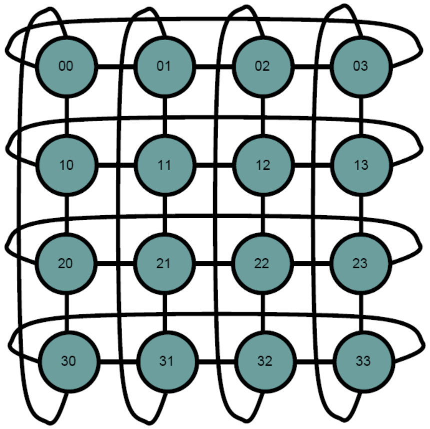
- Static arrangement of channels and nodes in an interconnection network
- The roads over which packets travel
- Topology chosen based on cost and performance
  - Cost and performance determided by many factors (flow control, routing, traffic)
  - Measures to evaluate just the topology
    - ✓ Bisection bandwidth
    - ✓ Channel load
    - ✓ Path delay

Factors Affecting Perfomance

- Factors that influence the performance of a NoC are
  - *Topology* (static arrangement of channels and nodes)
  - *Routing Technique* (selection of a path through the network)
  - *Flow Control* (how are network resources allocated, if packets traverse the network)
  - *Router Architecture* (buffers, switches, …)
  - *Traffic Pattern*

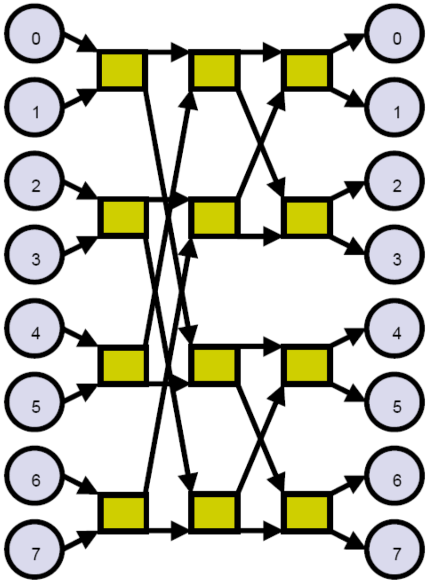
Direct and Indirect Networks

- Direct Network
  - Every Node in the network is both a terminal and a switch



Direct Network

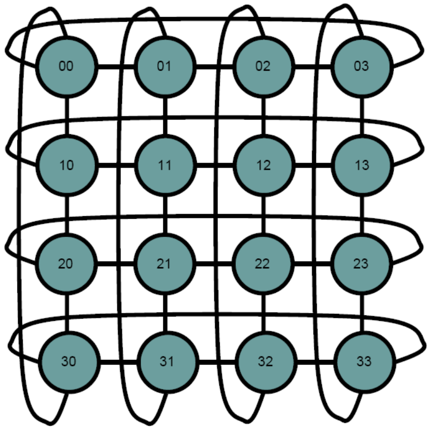
- Indirect Network
  - Nodes are either switches or terminal



Indirect Network

Direct Networks

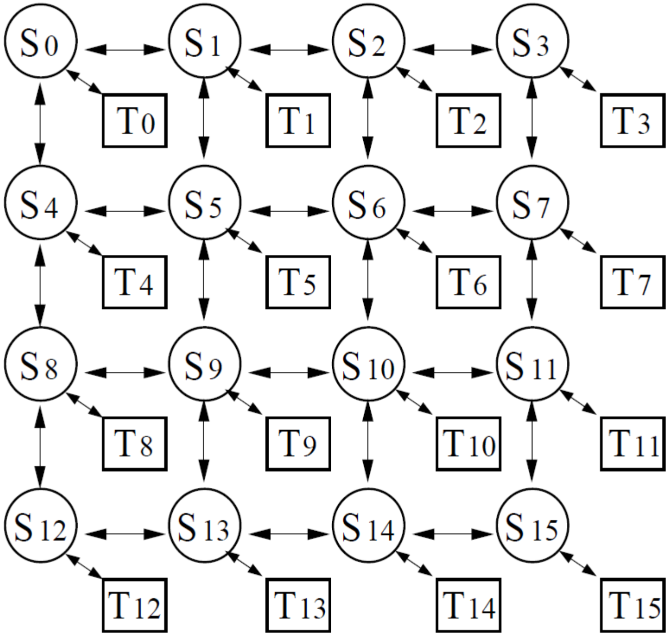
- aka point-to-point network
- Consists of a set of nodes, each one being directly connected to a (usually small) subset of other nodes in the network
  - ➔ These nodes may have different functional capabilities
    - ✓ *E.g.*, vector processors, graphics processors, I/O processors, *etc.*



Direct Network

Direct Networks - Router

- A common component of the node is *the router*
  - It handles message communication among nodes
    - For this reason, direct networks are also known as router-based networks
  - Each router has direct connections to the router of its neighbors



Direct Networks - Links

- Two neighboring nodes are connected by a pair of unidirectional channels in opposite directions
- A bidirectional channel may also be used to connect two neighboring nodes

Direct Networks - Scalability

- As the number of nodes in the system increases, the total communication bandwidth also increase
  - ➔Thus, direct networks have been a popular interconnection architecture for constructing large-scale parallel computers

Direct Networks - Topologies

- Many network topologies have been proposed in terms of their graph-theoretical properties
  - ➔Very few of them have ever been implemented
  - ➔Most of the implemented networks have an *orthogonal topology*

DN – Orthogonal Topology

- A network topology is orthogonal if and only if nodes can be arranged in an orthogonal *n*-dimensional space, and every link can be arranged in such a way that it produces a displacement in a single dimension
- Orthogonal Topologies
  - ➔ Strictly orthogonal topology
    - ✓ Every node has at least one link crossing each dimension
  - ➔ Weakly orthogonal topology
    - ✓ Some nodes may not have any link in some dimensions

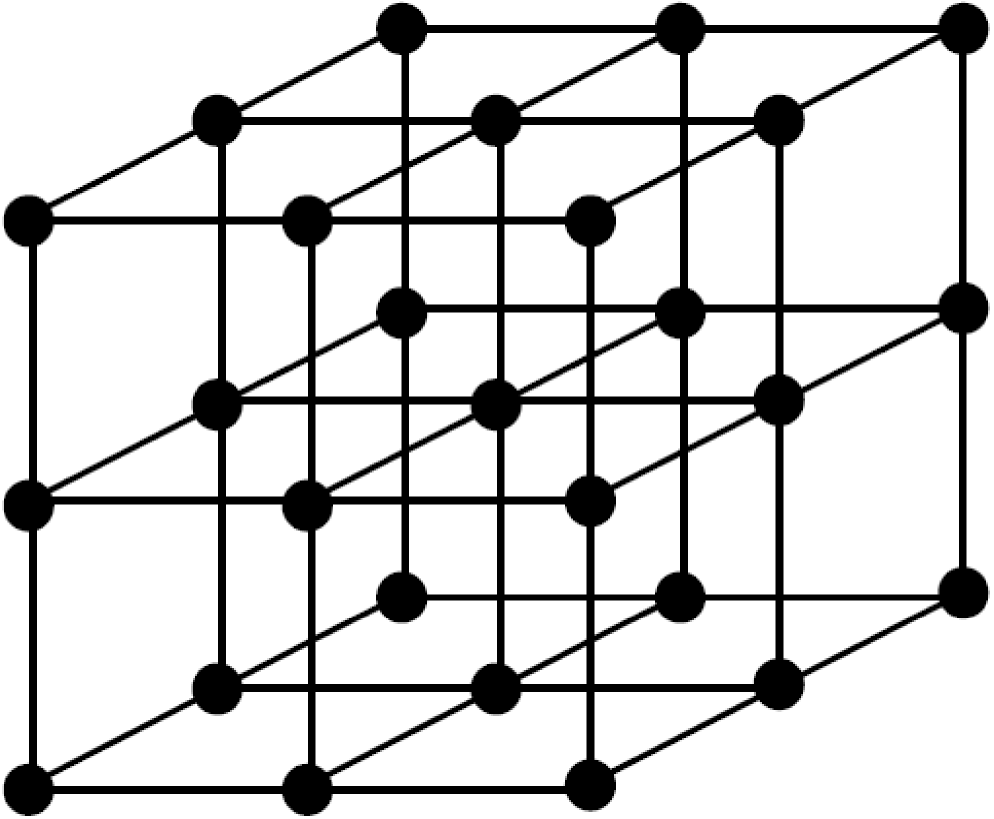
DN – Strictly Orthogonal Topologies

- Routing is very simple
  - Can be efficiently implemented in hardware
- Most popular strictly orthogonal direct network topologies
  - *n*-dimensional mesh
  - *k*-ary *n*-cube (torus)
  - Hypercube

n-Dimensional Mesh

- It has $K_0 x K_1 x \ldots x K_{n-1}$ nodes, K_i nodes along each dimension *i*
- Two nodes *X* and *Y* are neighbors if and only if $y_i = x_i$ for all *i*, $0 \le i \le n-1$, except one, *j*, where $y_j = x_j \pm 1$
  - ➔ Thus, nodes have from *n* to 2*n* neighbors, depending on their location in the mesh
    - ✓ Therefore, this topology is not regular

n-Dimensional Mesh

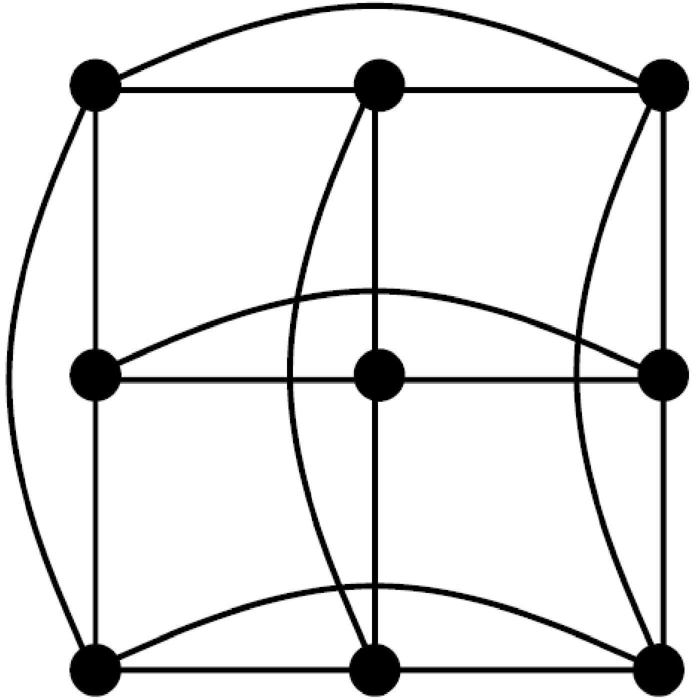


3-dimensional mesh

k-ary *n*-cube

- All nodes have the same number of neighbors
- It has K^n nodes
- Two nodes *X* and *Y* are neighbors if and only if $y_i = x_i$ for all i, $0 \le i \le n-1$, except one, *j*, where $y_j = (x_j \pm 1) \bmod K$
  - ➔ Modular arithmetic adds wraparound channels
    - ✓ Therefore, this topology is regular

k-ary *n*-cube

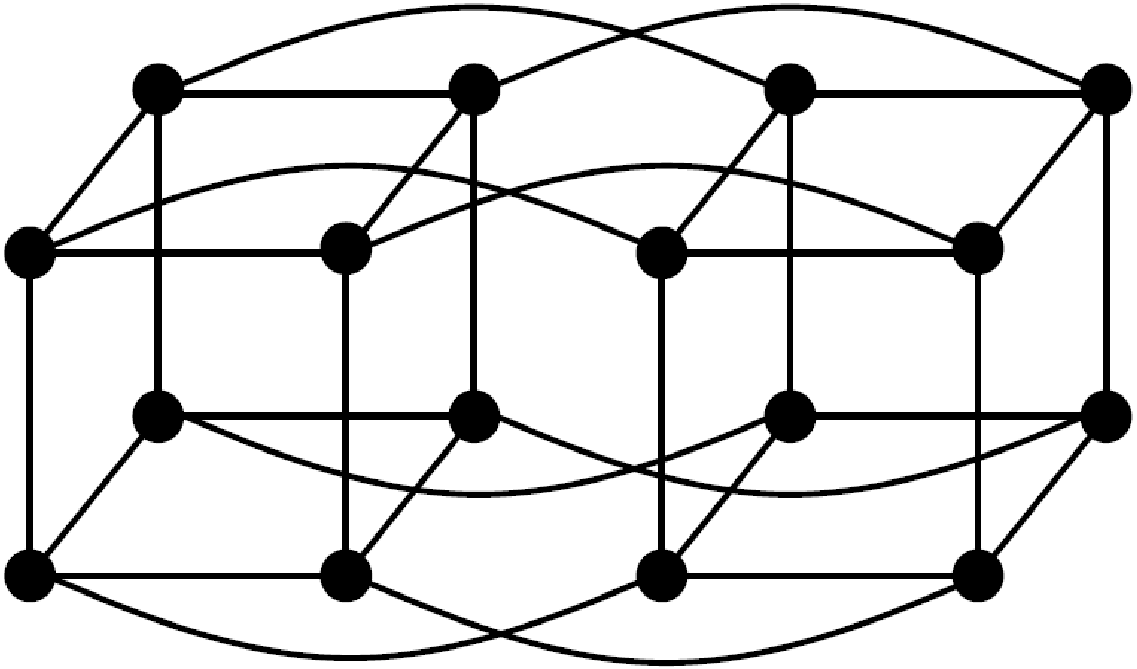


3-ary 2-cube

Hypercube

- It is a special case of both *n*-dimensional meshes and *k*-ary *n*-cubes
- A hypercube is an n-dimensional mesh in which $K_i = 2$ for $0 \leq i \leq n-1$, or a 2-ary *n*-cube
  - ➔ This topology is regular

Hypercube

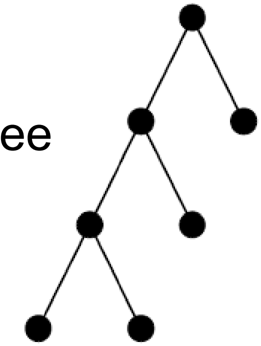


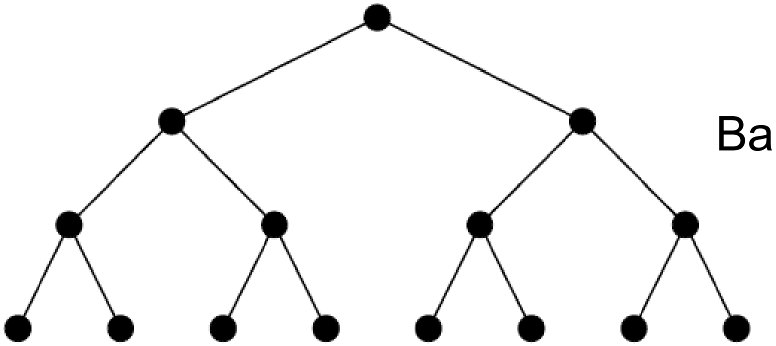
2-ary 4-cube (hypercube)

Other Direct Network Topologies

- Aimed at minimizing the network diameter
- Every node but the root has a single parent node
  - ➔Trees contain no cycles
- *k*-ary tree
  - ➔A tree in which every node but the leaves has a fixed number *k* of descendants
- Balanced tree
  - ➔The distance from every leaf node to the root is the same

Ubalanced tree





Balanced tree

Drawbacks of Trees

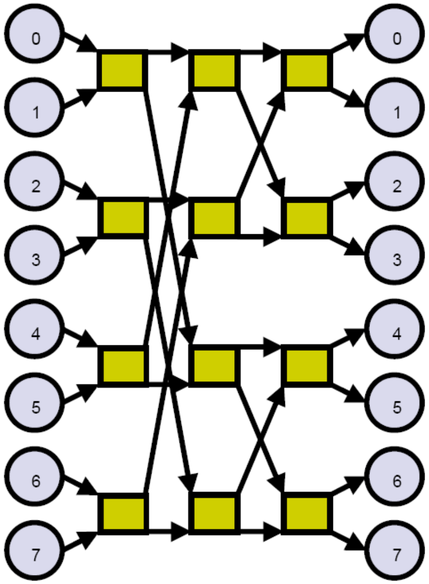
- Root node and the nodes close to it become a bottleneck
  - Allocating a higher channel bandwidth to channels located close to the root node
    - ✓ Using channels with different bandwidths is not practical, especially when message transmission is pipelined
- There are no alternative paths between any pair of nodes

Properties of Trees

- For any connected graph, it is possible to define a tree that spans the complete graph
  - ➔ Thus, For any connected network, it is possible to build an acyclic network connecting all the nodes by removing some links
  - ➔ This property can be used to define a routing algorithm for any irregular topology
    - ✓ However, that routing algorithm may be inefficient due to the concentration of traffic across the root node

Indirect Networks

- The communication between any two nodes is carried through some switches
- Each node has a network adapter that connects to a network switch
- The interconnection of those switches defines various network topologies



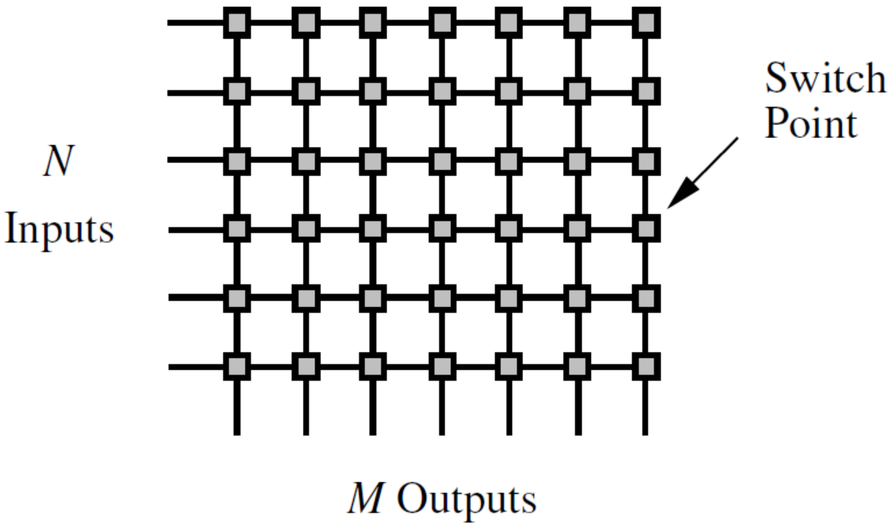
Indirect Network

IN - Crossbar Networks

- Allow any processor in the system to connect to any other processor (or memory unit) so that many processors can communicate simultaneously without contention
- Used in
  - The design of high-performance small-scale multiprocessors
  - The design of routers for direct networks
  - As basic components in the design of large-scale indirect networks

Crossbar Networks (cnt'd)

- A crossbar can be defined as a switching network with *N* inputs and *M* outputs
  - ➔Which allows up to min{*N*,*M*} one-to-one interconnections without contention
  - ➔The cost of such a network is *O*(*NM*), which is prohibitively high with large *N* and *M*

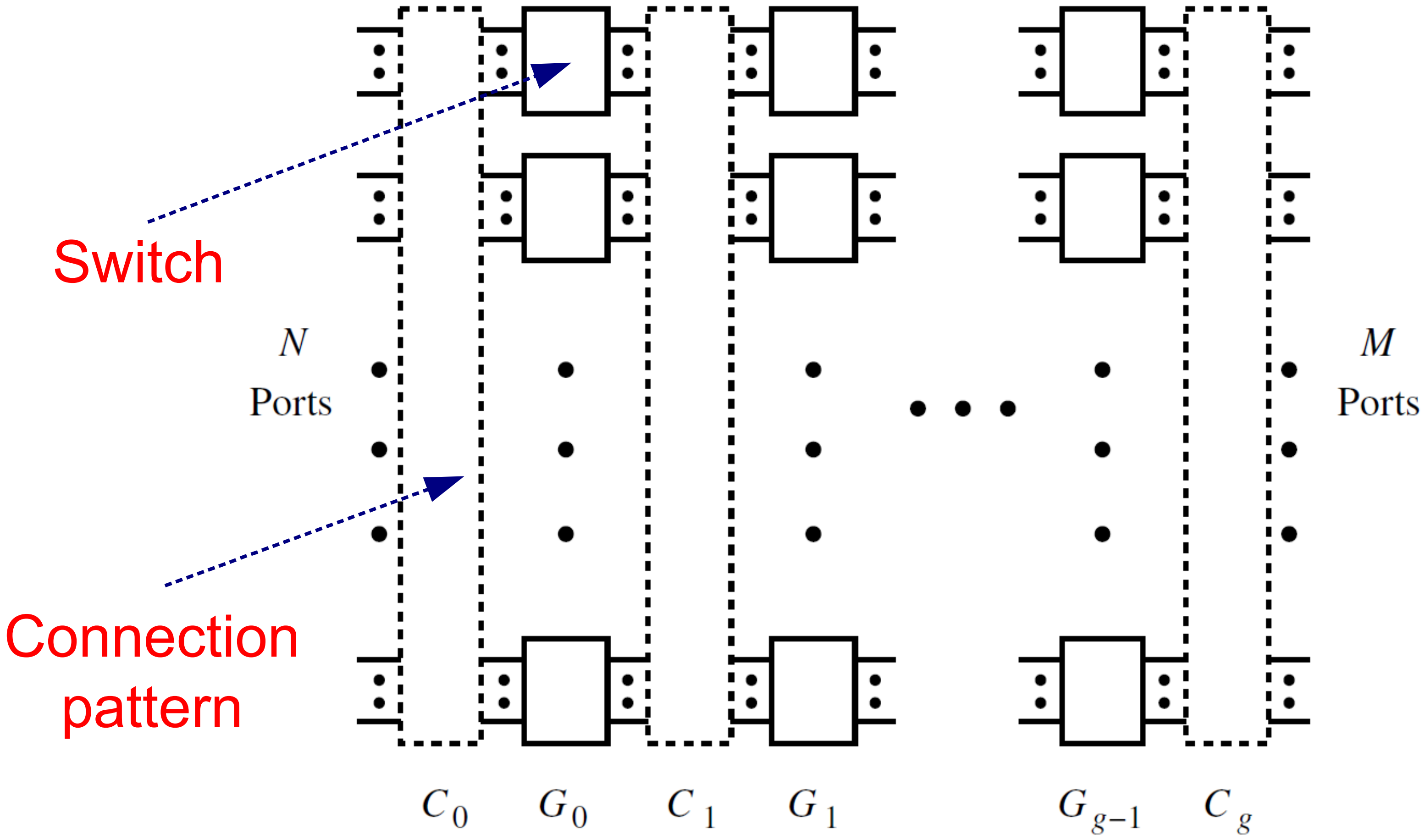


IN - Multistage Networks

- MINs: Multistage Interconnection Networks
  - Connect input devices to output devices through a number of switch stages
    - ✓ Where each switch is a crossbar network
  - The number of stages and the connection patterns between stages determine the routing capability of the networks
- MINs are good for constructing parallel computers with hundreds of processors and have been used in some commercial machines

Generalized MIN Model

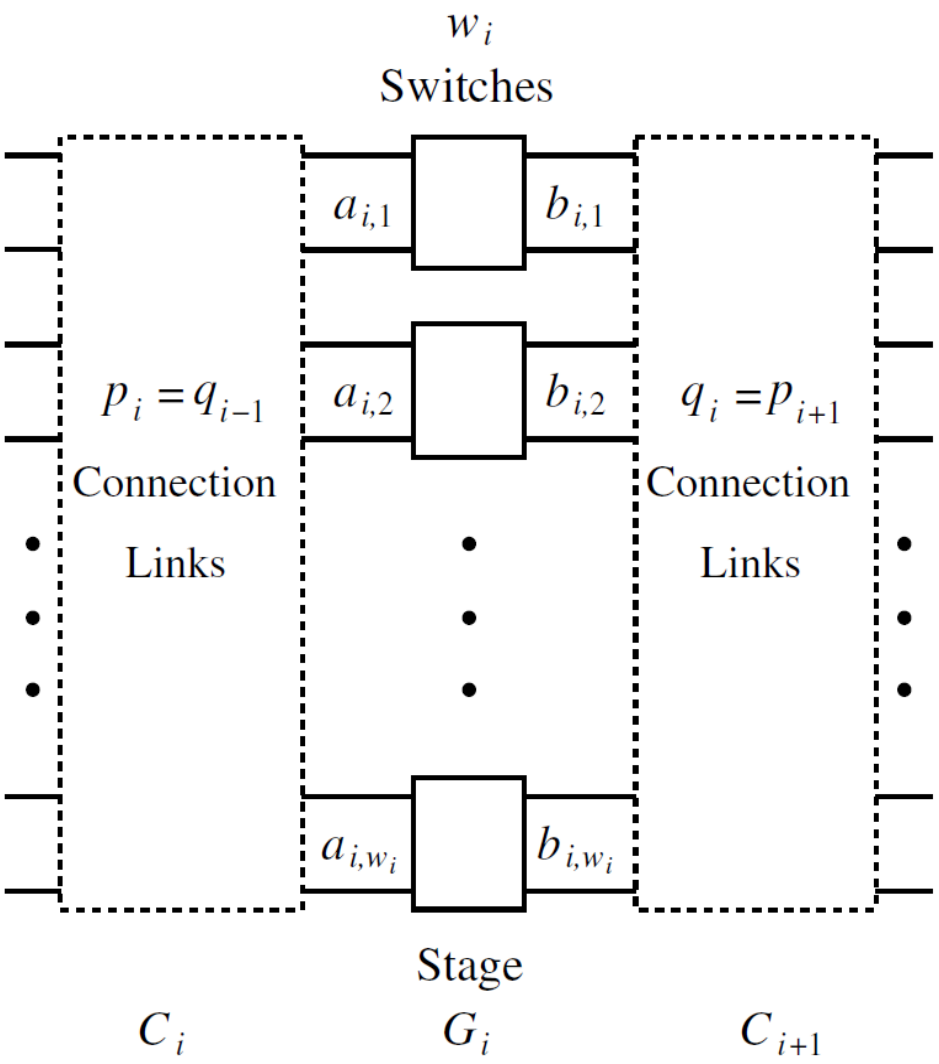
A generalized MIN with *N* inputs, *M* outputs, and *g* stages



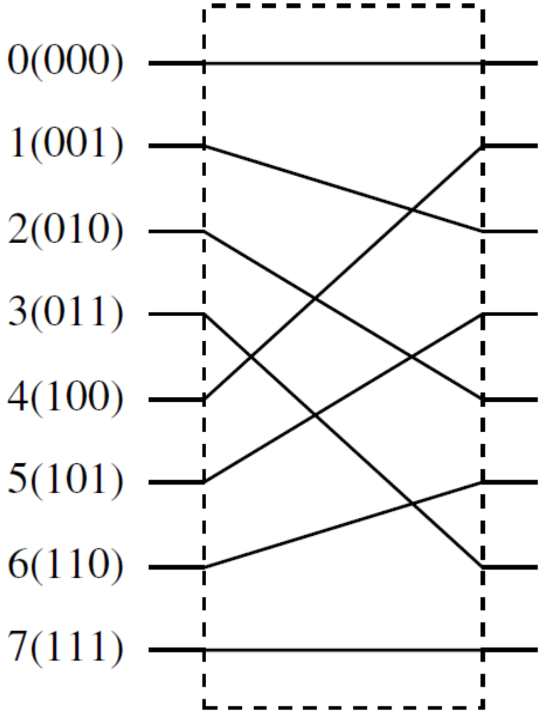
Generalized MIN Model

- All the switches will be identical
  - ➔ Thus amortizing the design cost
- Banyan networks are a class of MINs with the property that there is a unique path between any pair of source and destination
  - ➔ An *N*-node ($N = K^n$) Delta network is a subclass of Banyan networks, which is constructed from identical K×K switches in *n* stages, where each stage contains *N*/*K* switches
    - ✓ Omega, flip, cube, butterfly, ...

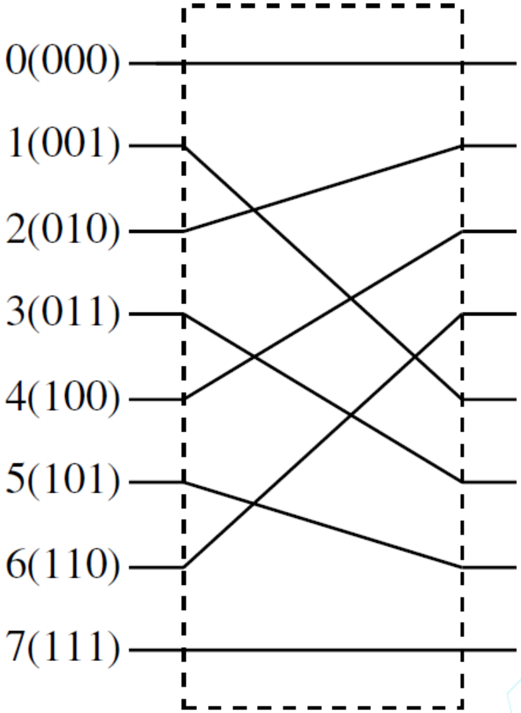
A Closer View of Stage G_i



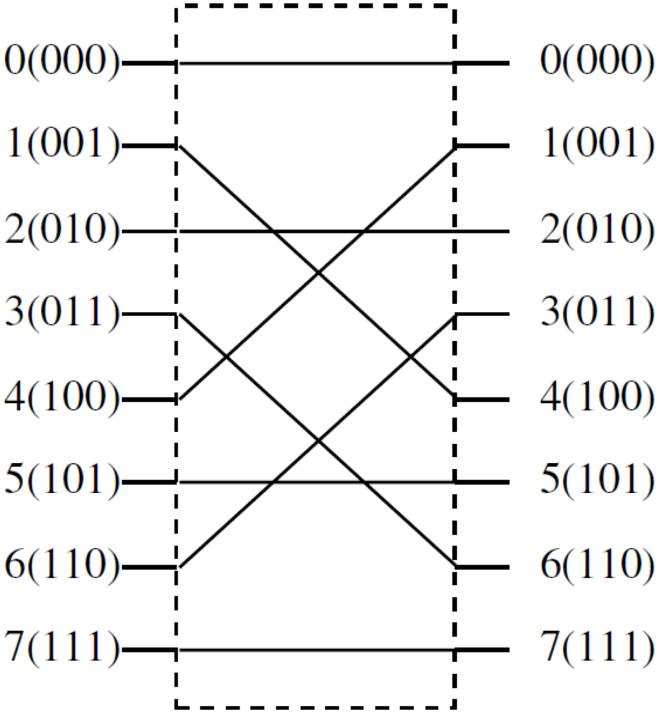
Connection Patterns



Perfect shuffle

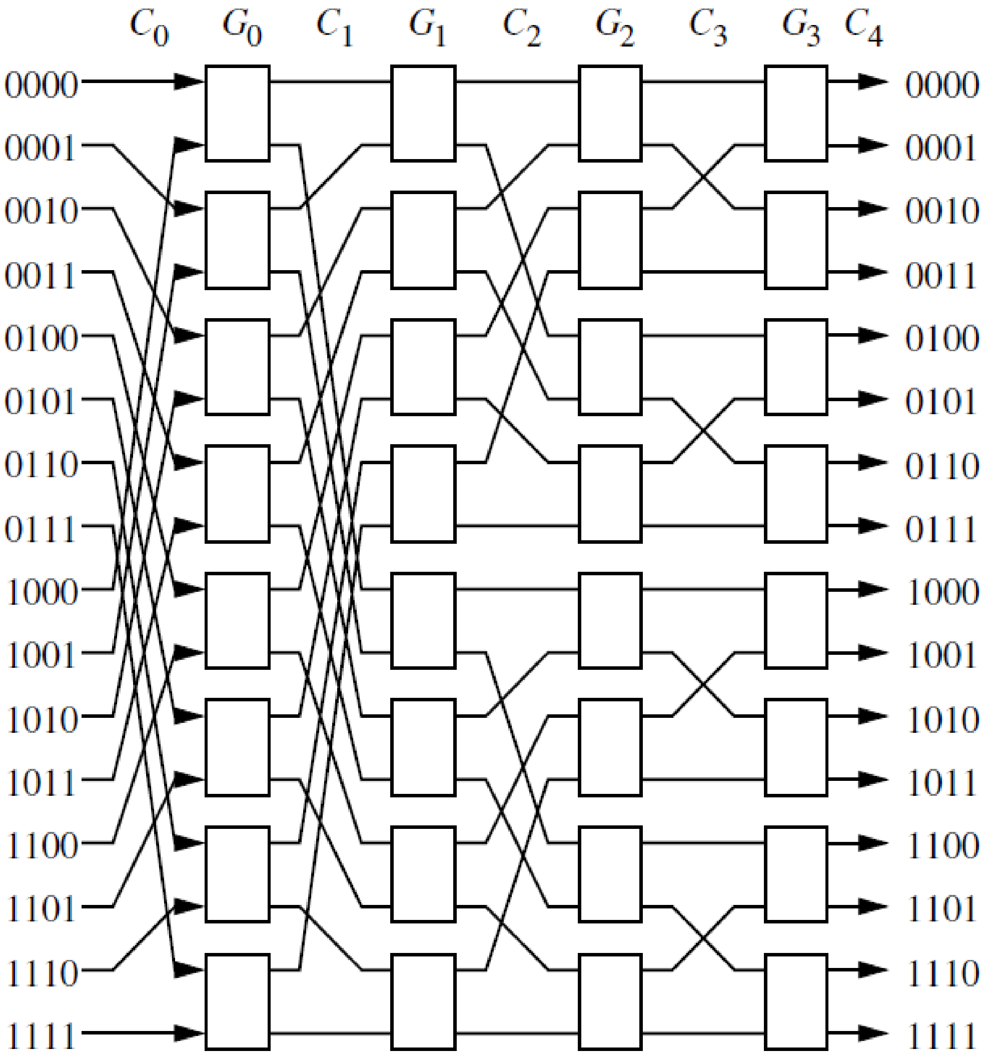


Inverse perfect shuffle



Bit reversal

16x16 MIN Example



MINs Classification

- Blocking
- Nonblocking

MINs - Blocking

- A connection between a free input/output pair is not always possible
  - ➔Because of conflicts with the existing connections
- Typically, there is a unique path between every input/output pair
  - ➔Minimizing the number of switches and stages
  - ➔It is also possible to provide multiple paths to reduce conflicts and increase fault tolerance
    - ✓Multipath networks

MINs - Nonblocking

- Any input port can be connected to any free output port without affecting the existing connections
  - ➔ The same functionality as a crossbar
  - ➔ Require multiple paths between every input and output
    - ✓ Leads to extra stages

Topology & Physical Constraints

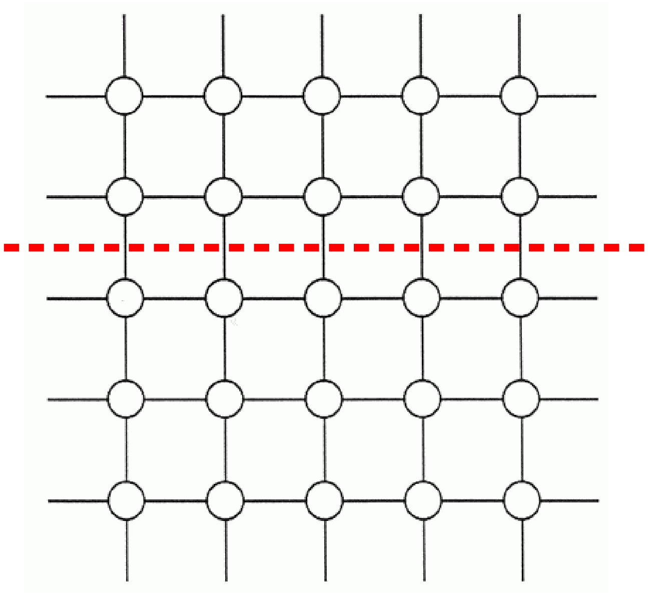
- It is important to model the relationships between physical constraints and topology
  - ➔ And the resulting impact on performance
- Network optimization is the process of utilizing these models
  - ➔ For selecting topologies that best match the physical constraints of the implementation
- For a given implementation technology, physical constraints determine architectural features
  - ➔ Channel widths
    - ✓ Impact on zero-load latency

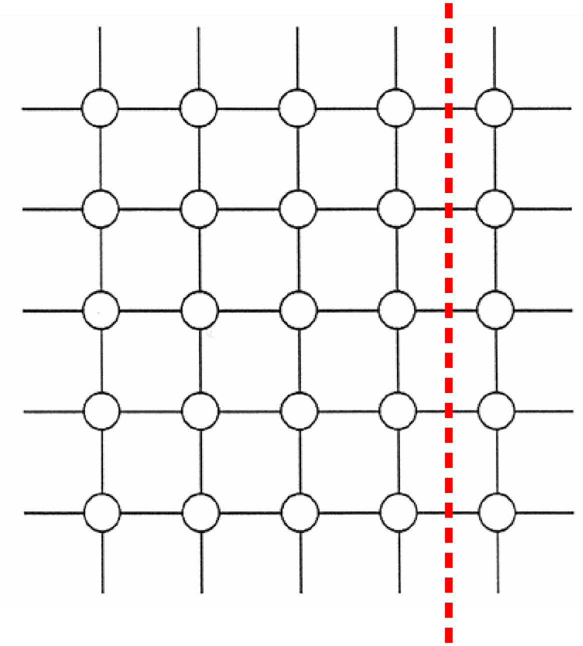
Bisection Width/Bandwidth

- One of the physical constraints facing the implementation of interconnection networks is the available wiring area
- The available wiring area is determined by the packaging technology
  - Whether the network resides on a chip, multichip module, or printed circuit board
- VLSI systems are generally wire limited
  - The silicon area required by these systems is determined by the interconnect area, and the performance is limited by the delay of these interconnections
- The choice of network dimension is influenced by how well the resulting topology makes use of the available wiring area
  - One such performance measure is the *bisection width*

Cuts

- A ***cut*** of a network, $C(N_1,N_2)$, is a set of channels that partitions the set of all nodes into two disjoint sets, N_1 and N_2
  - ➔Each element in $C(N_1,N_2)$ is a channel with a source in N_1 and destination in N_2 or vice versa





Bandwidth of the Cut

- Total ***bandwidth of the cut*** *C*(N_1,N_2)

$$B(N_1,N_2)=\sum_{c\in C(N_1,N_2)} b_c$$

Bisection

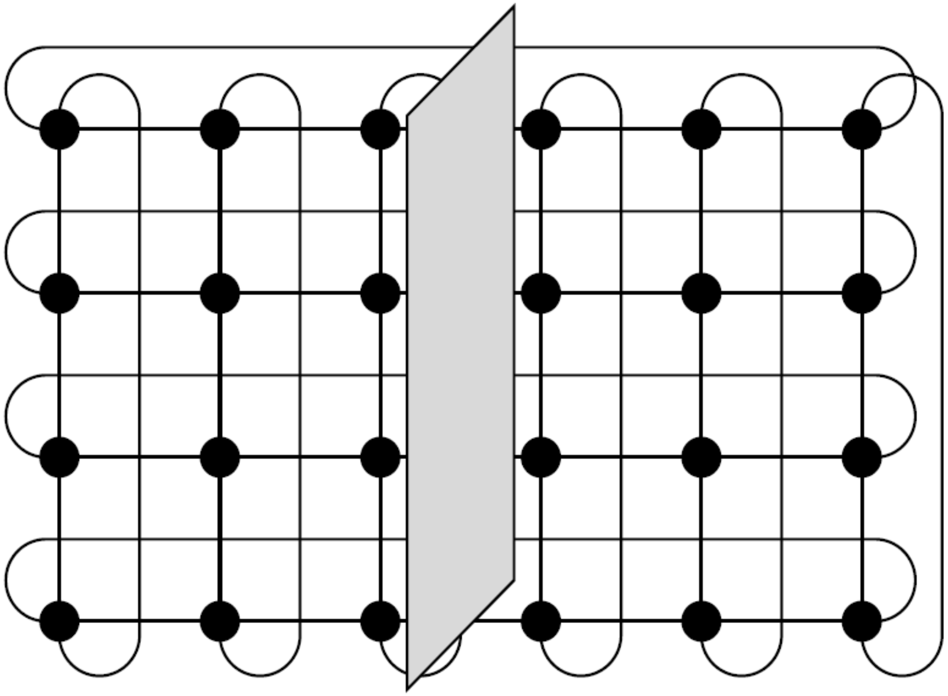
- The ***bisection*** is a cut that partitions the entire network nearly in half
- The ***channel bisection*** of a network, B_C, is the minimum channel count over all bisections

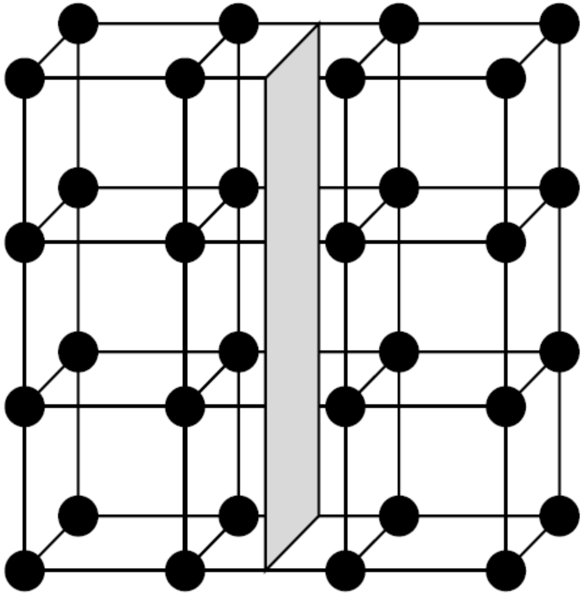
$$B_C = \min_{\text{bisections}} \left| C(N_1, N_2) \right|$$

- The ***bisection bandwidth*** of a network, B_B, is the minimum bandwidth over all bisections

$$B_B = \min_{\text{bisections}} \left| B(N_1, N_2) \right|$$

Bisection Examples





Diameter

- The ***diameter*** of a network, H_{max}, is the largest, minimal hop count over all pairs of terminal nodes

$$H_{\max} = \max_{x,y \in N} \left| H(x, y) \right|$$

For a fully connected network with N terminals built from switches with out degree δ_O, H_{max} is bounded by

$$H_{\max} \geq \log_{\delta_O} N \qquad (1)$$

Each terminal can reach at most δ_O other terminals after one hop

At most δ_O^2 after two hops, and at most δ_O^H after H hops

If we set $\delta_O^H = N$ and solve for H, we get (1)

Average Minimum Hop count

- The ***average minimum hop count*** of a network, H_{min}, is defined as the average hop count over all sources and destinations

$$H_{\min} = \frac{1}{N^2} \sum_{x,y \in N} H(x,y)$$

Physical Distance and Delay

- The ***physical distance*** of a path is

$$D(P) = \sum_{c \in P} l_c$$

- The ***delay*** of a path is

$$t(P) = D(P)/v$$

Performance

- Throughput
  - Data rate in bits/s that the network accepts per input port
  - It is a property of the entire network
  - It depends on
    - ✓Routing
    - ✓Flow control
    - ✓**Topology**

Ideal Throughput

- ***Ideal throughput*** of a topology
  - ➔Throughput that the network could carry with perfect flow control (no contention) and routing (load balanced over alternative paths)
- Maximum throughput
  - ➔It occurs when some channel in the network becomes saturated

Channel Load

- We define the ***load of a channel c***, γ_c, as

$$\gamma_c = \frac{\text{bandwidth demanded from channel } c}{\text{bandwidth of the input ports}}$$

- Equivalently
  - Amount of traffic that must cross ***c*** if each input injects one unit of traffic
- Of course, it depends on the traffic pattern considered
  - We will assume uniform traffic

Maximum Channel Load

- Under a particular traffic pattern, the channel that carries the largest fraction of traffic determines the ***maximum channel load*** $\gamma_{\max}$ of the topology

$$\gamma_{\max} = \max_{c \in C} \gamma_c$$

Ideal Throughput

- When the offered traffic reaches the throughput of the network, the load on the ***bottleneck channel*** will be equal to the channel bandwidth ***b***
  - ➔ Any additional traffic would overload this channel
- The ***ideal throughput*** Θ_{ideal} is the input bandwidth that saturates the bottleneck channel

$$\gamma_c = \frac{\text{bandwidth demanded from channel } c}{\text{bandwidth of the input ports}}$$

$$\gamma_c = \gamma_{\max} = \frac{b}{\Theta_{\text{ideal}}}$$

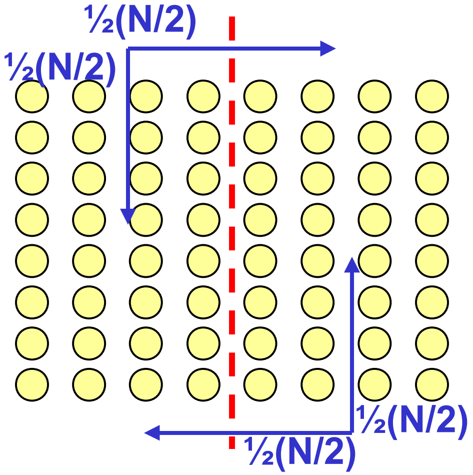
$$\boxed{\Theta_{\text{ideal}} = \frac{b}{\gamma_{\max}}}$$

Bounds for γ_{max}

- γ_{max} is very hard to compute for the general case (arbitrary topology and arbitrary traffic pattern)
- For uniform traffic some upper and lower bounds can be computed with much less effort

Lower Bound on γ_{max}

- The load on the bisection channels gives a lower bound on o_{max}
- Let us assume uniform traffic
  - ➔ On average, half of the traffic (***N*/2** packets) must cross the ***B*** $_C$ bisection channels
  - ➔ The best throughput occurs when these packets are distributed evenly across the bisection channels
  - ➔ Thus, the load on each bisection channel o_B is **at least**



$$\boxed{\gamma_{\max} \geq \gamma_B = \frac{N}{2B_C}}$$

Upper Bound on Θ_{ideal}

- We found that

$$\Theta_{\text{ideal}} = \frac{b}{\gamma_{\max}} \qquad \text{and} \qquad \gamma_{\max} \geq \gamma_B = \frac{N}{2B_C}$$

- Combining the above equations we have

$$\boxed{\Theta_{\text{ideal}} \leq \frac{2bB_C}{N} = \frac{2B_B}{N}}$$

Another Lower Bound on γ_{max}

- The average number of channel traversals required to deliver a packet for a given traffic is ***$H_{min}N$***

- Let us assume the best case in which all channels are loaded equally
  - ➔ The load on every channel in the network is

$$\boxed{\gamma_{c,LB} = \gamma_{\max,LB} = \frac{H_{\min}N}{|C|}}$$

Simple Upper Bound on $\gamma_{\max}$

- Let us suppose a routing function that balances load across all minimal paths equally
  - *E.g.*, If there are $|R_{xy}|$ paths from *x* to *y*, $1/|R_{xy}|$ is credited to each channel of each path
  - The maximum load over a channel *c* is

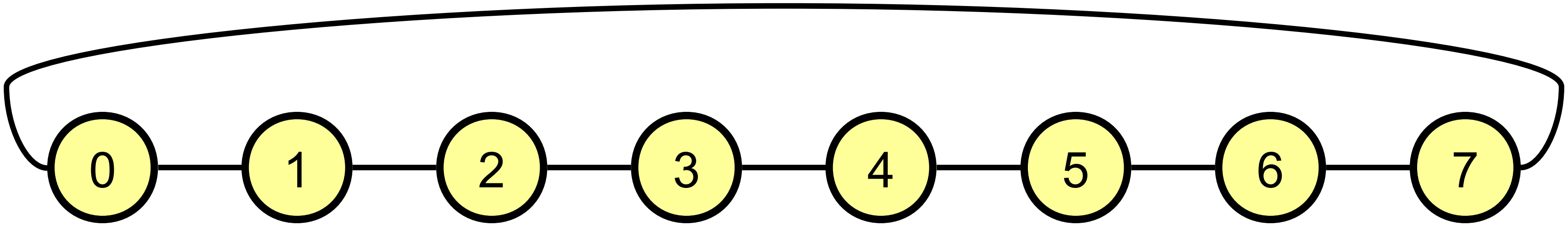
$$\gamma_{c,UB} = \frac{1}{N} \sum_{x \in N} \sum_{y \in N} \sum_{P \in xy} \begin{cases} 1/|R_{xy}| & \text{if } c \in P \\ 0 & \text{otherwise} \end{cases}$$

So, the maximum load $\gamma_{\max,UB}$ is the largest $\gamma_{c,UP}$ over all channels

$$\boxed{\gamma_{\max,UB} = \max_{c \in C} \gamma_{c,UB}}$$

Example (1/3)

- Let us consider a eight-node ring network



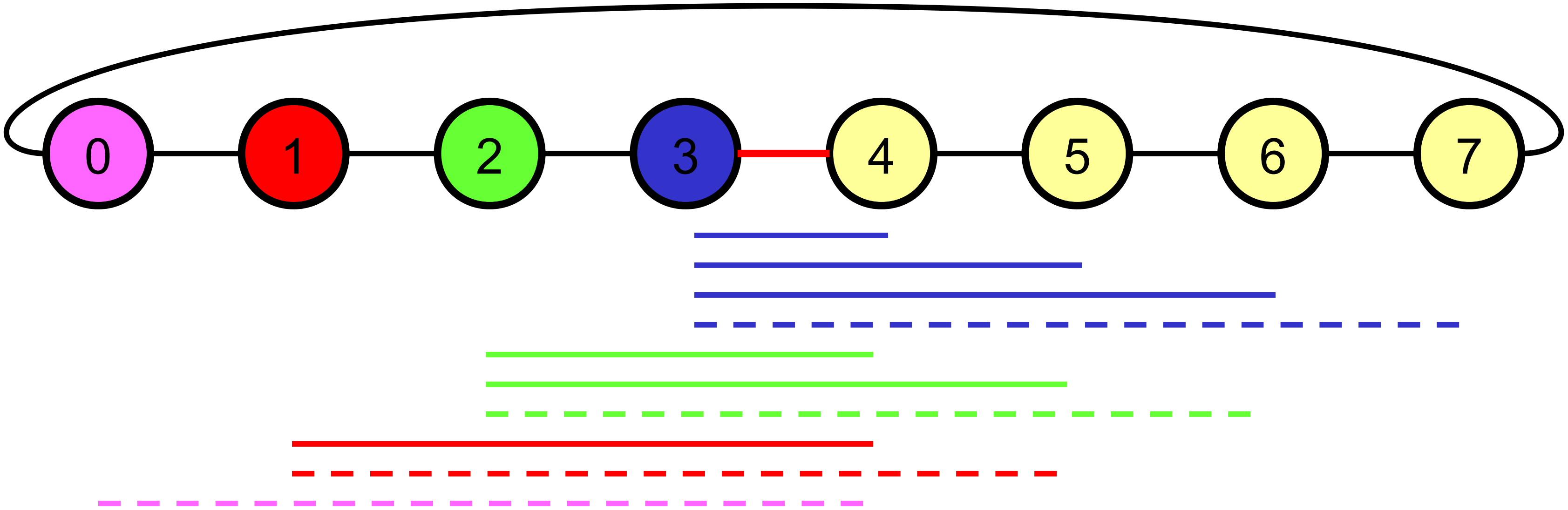
$$\gamma_{c,UB} = \frac{1}{N} \sum_{x \in N} \sum_{y \in N} \sum_{P \in xy} \begin{cases} 1/|R_{xy}| & \text{if } c \in P \\ 0 & \text{otherwise} \end{cases}$$

$$\gamma_{\max,UB} = \max_{c \in C} \gamma_{c,UB}$$

Example (2/3)

- Let us focus on channel (3,4)

$$\gamma_{(3,4),UB} = \frac{1}{N} \sum_{x \in N} \sum_{y \in N} \sum_{P \in xy} \begin{cases} 1/|R_{xy}| & \text{if } c \in P \\ 0 & \text{otherwise} \end{cases}$$



$$\gamma_{(3,4),UB} = \frac{1}{8}\left[6 \times 1 + 4 \times \frac{1}{2}\right] = 1$$

Example (3/3)

- It is simple to observe that the load on all channels is the same

  → $\gamma_{max,UB} = \gamma_{c,UB} = 1$

- Now, let us compute the lower bound

$$\gamma_{c,\,LB} = \gamma_{\max,LB} = \frac{H_{\min} N}{|C|}$$

  → H_{min} = 2 hops → $\gamma_{max,LB} = 2{}^{3}_{2}8/16 = 1$

- $\gamma_{max,LB} \frac{c}{\lambda} \gamma_{max} \frac{c}{\lambda} \gamma_{max,UB}$

  → $\gamma_{max} = 1$

Latency

- The ***latency*** of a network is the time required for a packet to traverse the network
  - ➔ From the time the head of the packet arrives at the input port to the time the tail of the packet departs the output port

Components of the Latency

- We separate latency, ***T***, into two components
  - ➔ ***Head latency*** (*T_h*): time required for the head to traverse the network
  - ➔ ***Serialization latency*** (*T_s*): time for a packet of length ***L*** to cross a channel with bandwidth ***b***

$$T = T_h + T_s = T_h + \frac{L}{b}$$

Contributions

- Like throughput, latency depends on
  - ➔ Routing
  - ➔ Flow control
  - ➔ Design of the router
  - ➔ ***Topology***

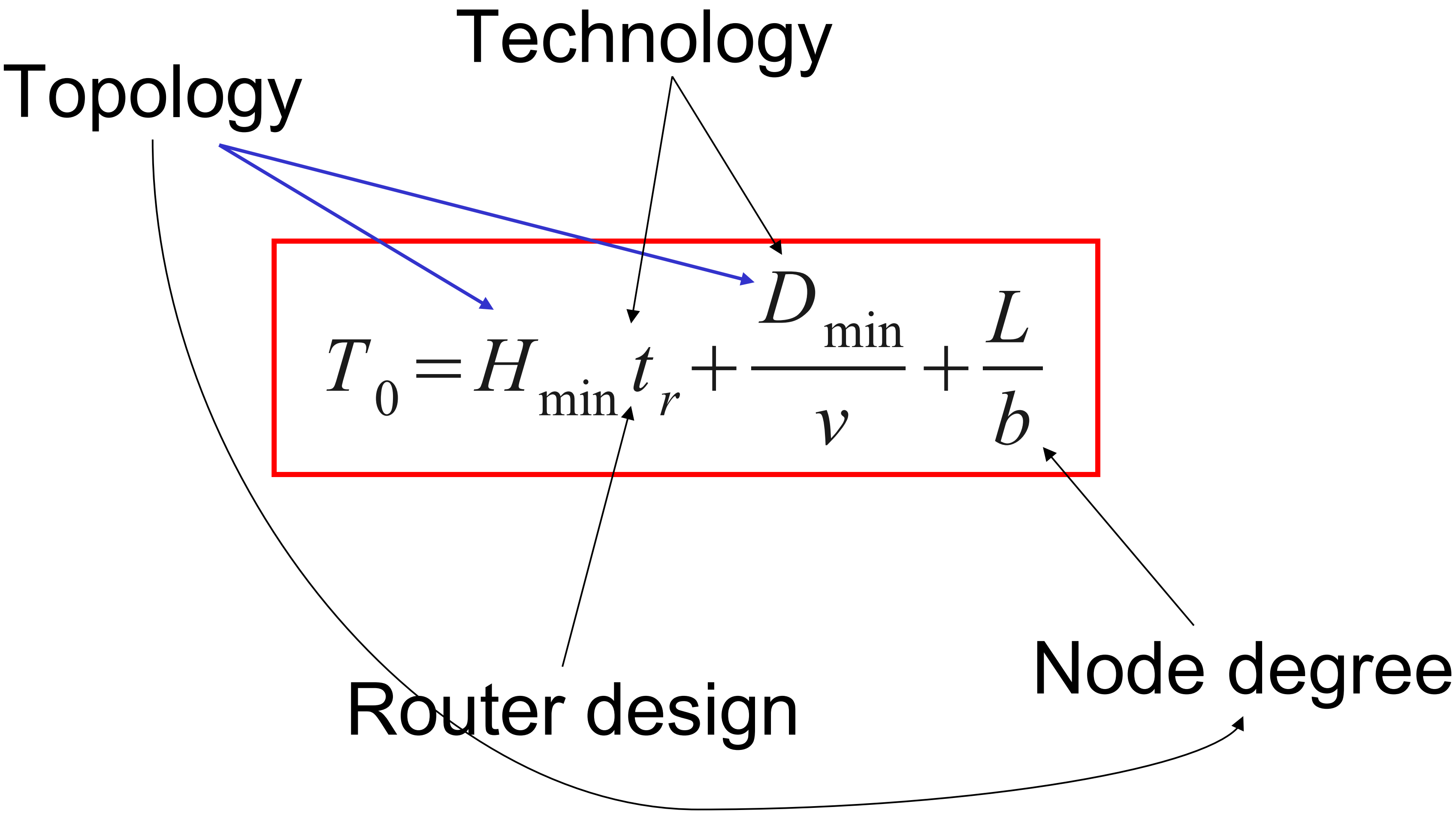
Latency at Zero Load

- We consider ***latency at zero load*, T_0**
  - ➔ Latency when no contention occurs
- T_h: sum of two factors determined by the topology
  - ➔ Router delay (T_r): time spent in the routers
  - ➔ Time of flight (T_w): time spent on the wires

$$T_h = T_r + T_w = H_{\min} t_r + \frac{D_{\min}}{v}$$

$$\boxed{T_0 = H_{\min} t_r + \frac{D_{\min}}{v} + \frac{L}{b}}$$

Latency at Zero Load



Technology

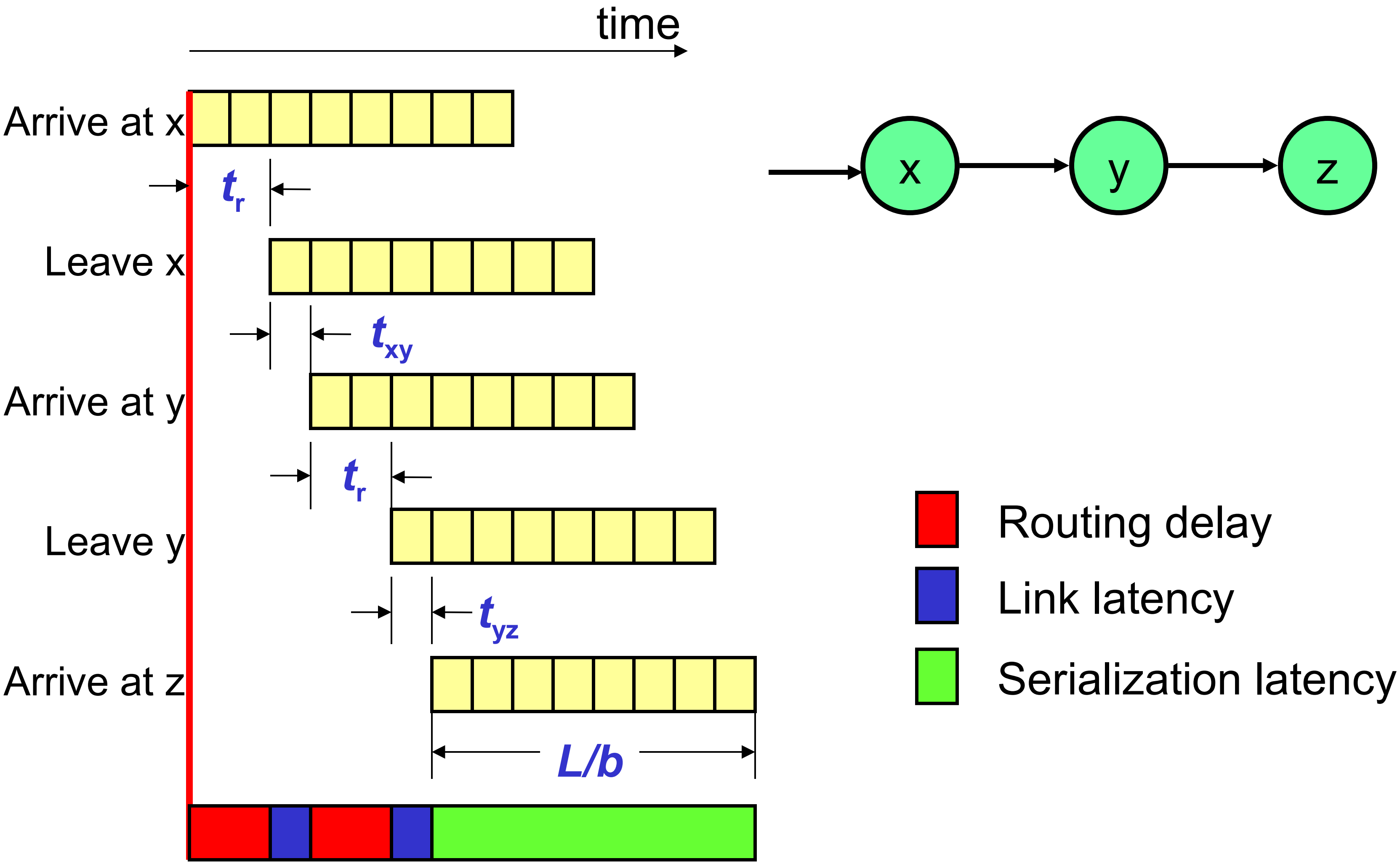
Topology

$$T_0 = H_{\min} t_r + \frac{D_{\min}}{v} + \frac{L}{b}$$

Router design

Node degree

Packet Propagation



time

Arrive at x

t_r

Leave x

t_{xy}

Arrive at y

t_r

Leave y

t_{yz}

Arrive at z

L/b

x

y

z

🟥 Routing delay

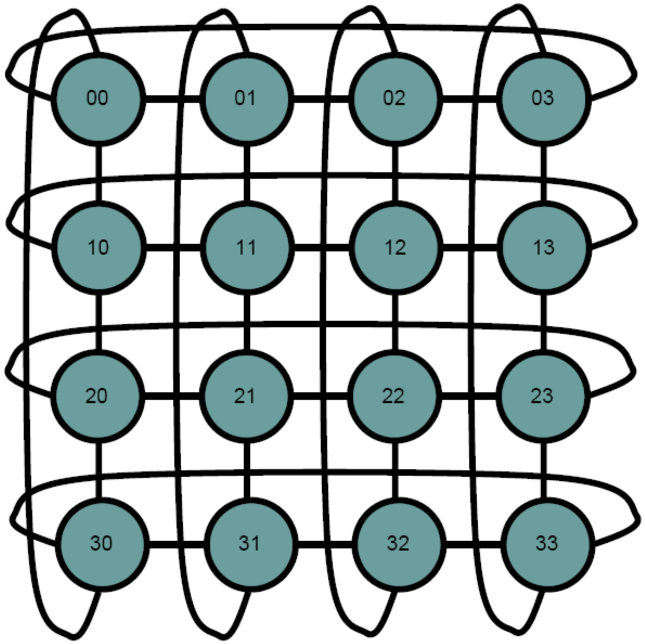
🟦 Link latency

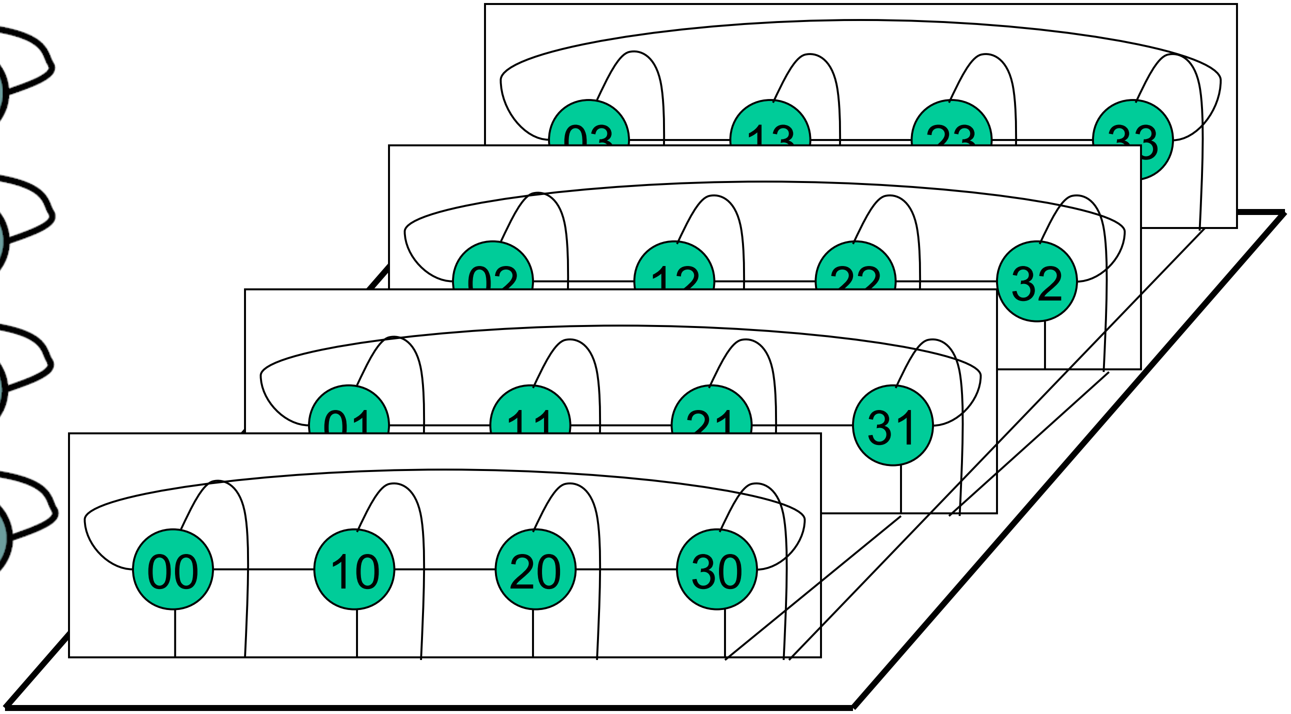
🟩 Serialization latency

Case Study

- A good topology exploits characteristics of the available packaging technology to meet *bandwidth* and *latency* requirements of the application
- To maximize bandwidth a topology should saturate the *bisection bandwidth*

Bandwidth Analysis (Torus)





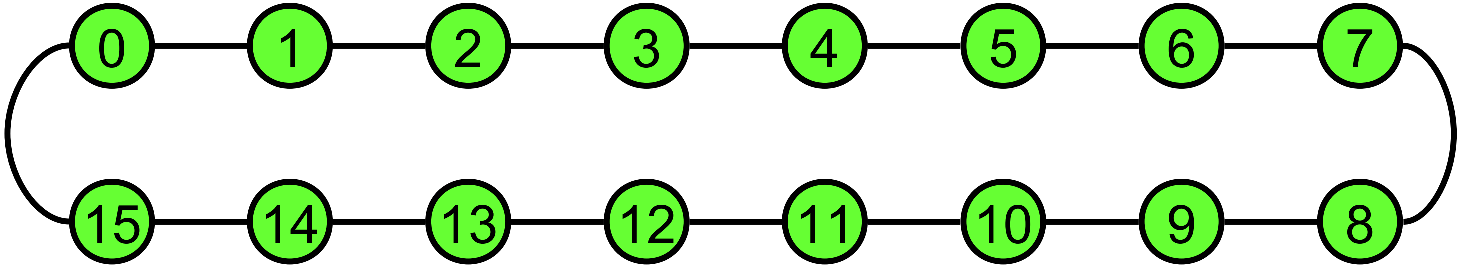
Assume: 256 signals @ 1Gbits/s

Bisection bandwidth 256 Gbits/s

Bandwidth Analysis (Torus)

- 16 unidirectional channels cross the mid-point of the topology
- To saturate the bisection of 256 signals
  - Each channel crossing the bisection should be 256/16 = 16 signals wide
- Constraints
  - Each node packaged on a IC
    - ✓ Limited number of I/O pins (*e.g.*, 128)
    - ✓ 8 channels per node → 8x16=128 pins → OK

Bandwidth Analysis (Ring)



- 4 unidirectional channels cross the mid-point of the topology
- To saturate the bisection of 256 signals
  - → Each channel crossing the bisection should be 256/4 = 64 signals wide
- Constraints
  - → Each node packaged on a IC
    - ✓ Limited number of I/O pins (*e.g.*, 128)
    - ✓ 4 channels per node → 4x64=256 pins → INVALID
  - → With identical technology constraints, the ring provides only half the bandwidth of the torus

Delay Analysis

- The application requires only 16Gbits/s
  - ➔ …but also minimum latency
- The application uses long 4,096-bit packets
- Suppose *random* traffic
  - ➔ Average hop count
    - ✓ Torus = 2
    - ✓ Ring = 4
- Channel size
  - ➔ Torus = 16 bits
  - ➔ Ring = 32 bits

Delay Analysis

- Serialization latency (channel speed 1GHz)
  - ➔Torus = 4,096/16 * 1ns = 256 ns
  - ➔Ring = 4,096/32 * 1ns = 128 ns
- Latency assuming 20ns hop delay
  - ➔Torus = 256 + 20*2 = 296 ns
  - ➔Ring = 128 + 20*4 = 208 ns
- No one topology is optimal for all applications
  - ➔Different topologies are appropriate for different constraints and requirements