# ECE 4750 Computer Architecture, Fall 2016
# T06 Fundamental Network Concepts

School of Electrical and Computer Engineering
Cornell University

revision: 2016-10-16-22-06

**1 Network/Roadway Analogy** **3**

1.1. Running Errands . . . . . . . . . . . . . . . . . . . . . . . . 4

1.2. Network Technology . . . . . . . . . . . . . . . . . . . . . . 5

1.3. Networks in Computer Architecture . . . . . . . . . . . . . . . 6

**2 Network Topology** **8**

2.1. Single-Stage Bus Topology . . . . . . . . . . . . . . . . . . . 8

2.2. Single-Stage Crossbar Topology . . . . . . . . . . . . . . . . . 9

2.3. Multi-Stage Butterfly Topology . . . . . . . . . . . . . . . . . 10

2.4. Multi-Stage Torus Topology . . . . . . . . . . . . . . . . . . . 12

2.5. Terminology . . . . . . . . . . . . . . . . . . . . . . . . . . 14

**3 Network Routing** **18**

3.1. Oblivious Deterministic Routing . . . . . . . . . . . . . . . . . 18

3.2. Oblivious Non-Deterministic Routing . . . . . . . . . . . . . . 18

3.3. Adaptive Routing . . . . . . . . . . . . . . . . . . . . . . . . 19

3.4. Deadlock . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

**4 Analyzing Network Performance 21**

4.1. Traffic Patterns . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

4.2. Ideal Throughput . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

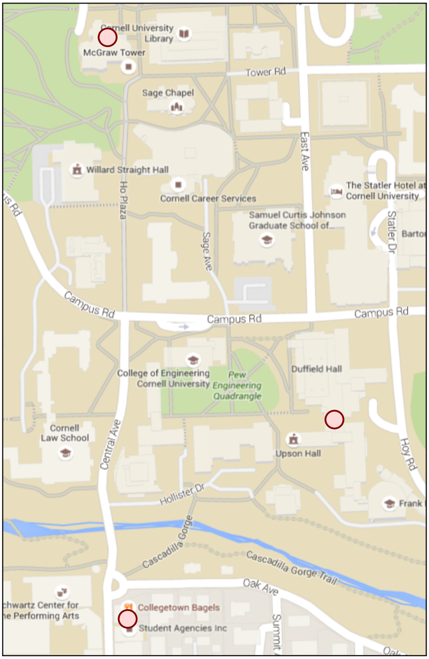
4.3. Zero-Load Latency . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

4.4. Comparing Topologies . . . . . . . . . . . . . . . . . . . . . . . . . . 28

4.5. Comparing Routing Algorithms . . . . . . . . . . . . . . . . . . . . . 29

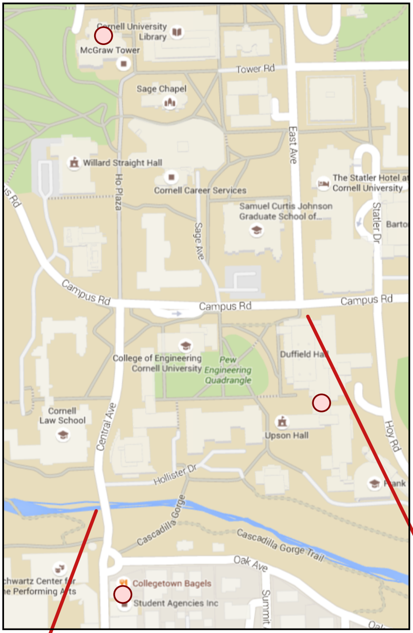
# 1. Network/Roadway Analogy

Our goal is to run some errands around town using our bike. Assume we are studying in the Duffield Atrium and we need to: (1) do some laundry in Collegetown (maybe pickup some coffee?), and (2) pick up some books about computer architecture from the library.



- Duffield Atrium
- Coffee/Laundry
- Library

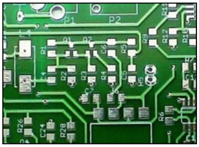
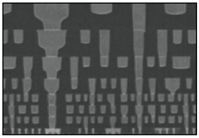
## 1.1. Running Errands



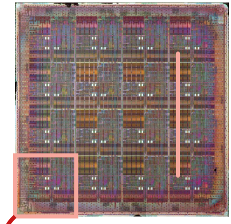
Holliste
Cascadilla Creek

- Network Topology
  - Arrangement of roads and intersections to interconnect sources and destinations
  - Wide vs. narrow roads
  - Long vs. short roads
  - Small vs. large intersections
- Network Routing
  - Path from source to destination along roads and intersections
  - Short vs. long paths
  - Common vs. rare paths
- Network Microarchitecture
  - Managing long line of bikes and cars on road
  - Managing many cars and bikes at same intersection

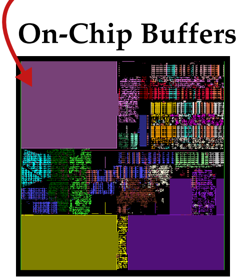
## 1.2. Network Technology

| | **AWG24 Twisted Pair** | **PCB Trace** | **On-Chip M6 Wire in 0.18µm** |
|---|---|---|---|
| | |  |  |
| **Resistance** | 0.08 Ω/m | 5 Ω/m | 40 kΩ/m |
| **Inductance** | 400 nH/m | 300 nH/m | 4 µH/m |
| **Capacitance** | 40 pF/m | 100 pF/m | 300 pF/m |
| **Data Rate** | ≈Gb/s | ≈Gb/s | ≈Gb/s |
| **Critical Length** | 1m | 10cm | <1cm |
| **Pitch** | ≈mm | <mm | <µm |

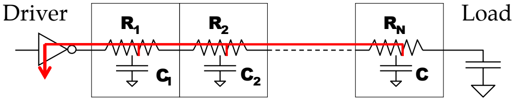
### On-Chip Wires



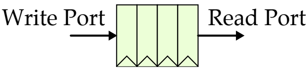
### On-Chip Buffers



Distributed Wire Resistance and Capacitance



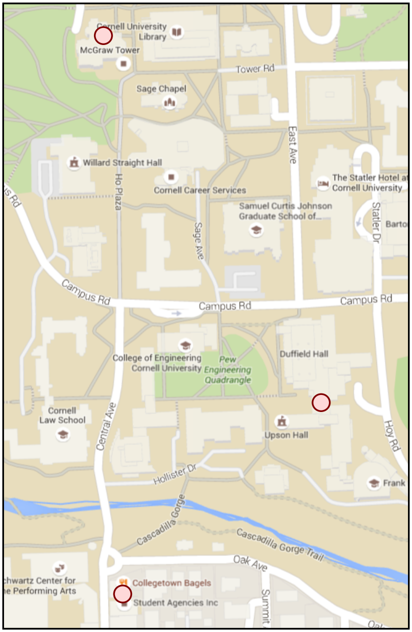
Because both wire resistance and wire capacitance increase with length, wire delay grows quadratically with length



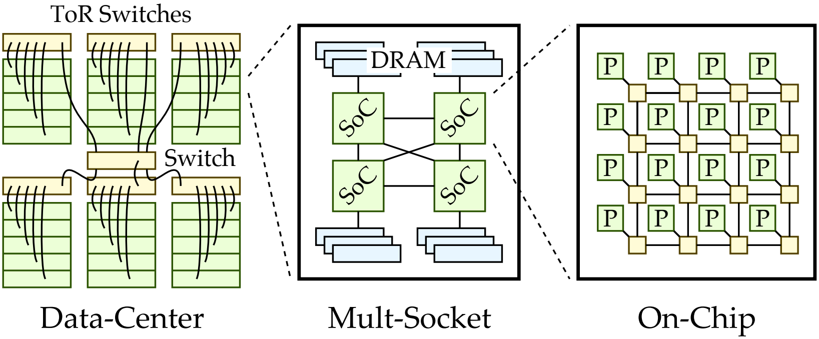
On-chip buffers are 1r1 FIFOs implemented using either a register file or SRAM

On-chip network technology constraints very different from off-chip technology constraints

## 1.3. Networks in Computer Architecture

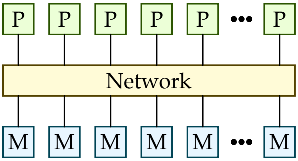


- Network Topology
  - Arrangement of channels and routers to interconnect sources and destinations
  - Roads = channels (implemented with cables, traces, wires)
  - Width of road = channel bw
  - Intersections = routers
  - Size intersection = router radix
- Network Routing
  - Path from source to destination along channels and routers
  - Short vs. long paths = minimal vs. non-minimal paths
  - Common vs. rare paths = channel congestion
- Topology is constrained by packaging (geography)

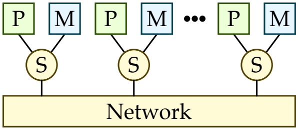


### Network Transactions

We will use processor-memory networks as a driving example throughout the course. Processor-memory networks allow many processors to perform memory read/write transactions on many memories.



Dance-Hall Organization



Integrated-Node Organization

- Message = Logical unit of data transfer provided by network interface
- Packet = Unit of routing within a network
- Flit = Smallest unit of resource allocation in channel/router (flow-control digit)
- Phit = Smallest unit of data processed by a channel/router (physical digit)

For the processor-memory networks we will primarily study:

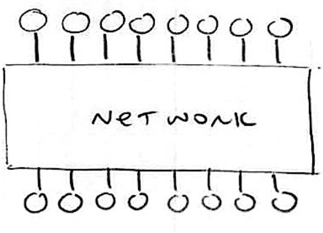
Message = Packet = Flit

The processor will send memory request messages/packets over the network, and the memories will send memory response messages/packets back to the processor. In this context, we consider the network messages/packets to be the "transactions".

| | | |
|---|---|---|
| Processors | : | Instructions |
| Memories | : | Memory accesses |
| Networks | : | Network packets |

# 2. Network Topology

Network topology is the arrangement of channels and routers to interconnect sources and destinations. We will explore four different topology classes:



- Single-stage bus topologies
- Single-stage crossbar topologies
- Multi-stage butterfly topologies
- Mult-stage torus topologies

## 2.1. Single-Stage Bus Topology

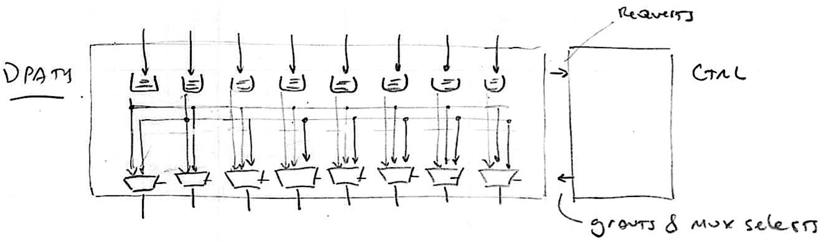
**SWMR Bus**

**MWSR Bus**

**MWMR Bus**

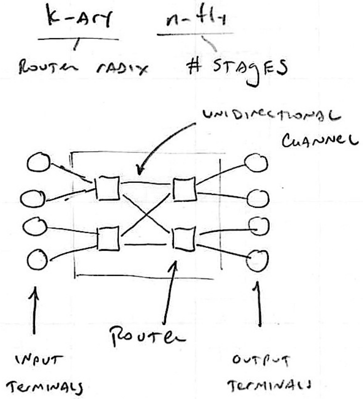
### Integrating Multiple Buses

## 2.2. Single-Stage Crossbar Topology



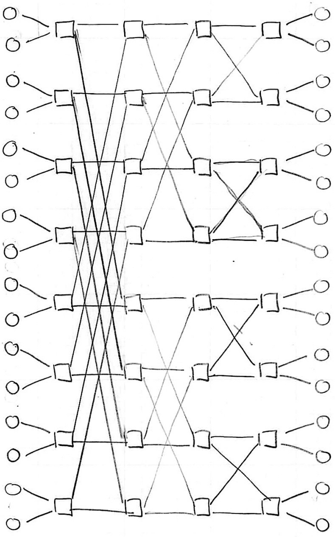
- Single-stage topologies are difficult to scale in terms of cycle time, energy, and area
- Multi-stage topologies improve scalability but raise many other interesting challenges

## 2.3. Multi-Stage Butterfly Topology



2-ARY 2-FLY

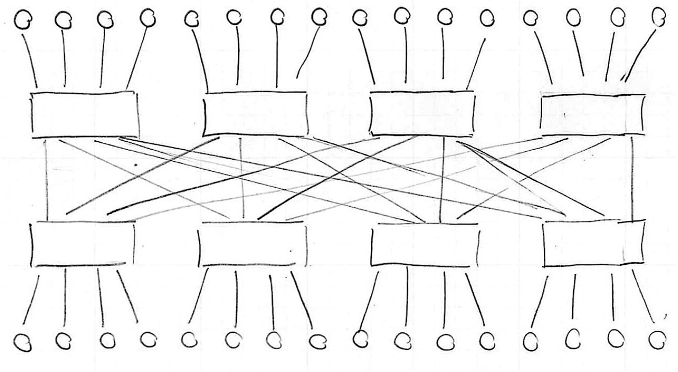
EACH ROUTER IS SIMILAR TO A 2x2 CROSSBAR



2-ARY
4-FLY

$2^4$ = NODES

### Example Butterfly Topology: 4-ary 2-fly



### Example Butterfly Topology: 3-ary 2-fly

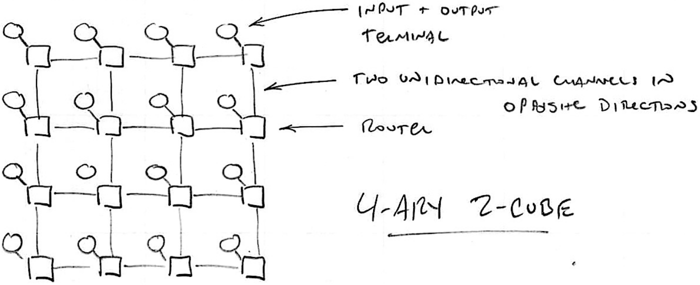
Sketch a 3-ary 2-fly. Use circles for the terminals, squares for the routers, and lines for one uni-directional channel.

## 2.4. Multi-Stage Torus Topology

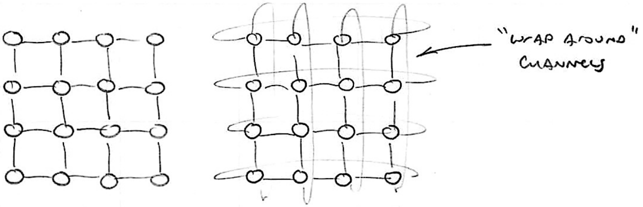
K-ARY N-CUBE

NODES IN EACH DIMENSION

n DIM grid

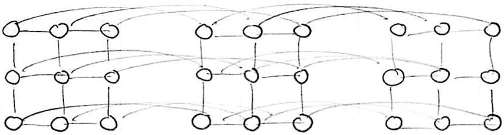


4-ARY 2-CUBE



4-ARY 2-CUBE MESH

4-ARY 2-CUBE TORUS

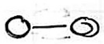


3-ARY 3-CUBE MESH

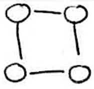
### Example Torus Topology: 8-ary 1-cube torus

Sketch a 8-ary 1-cube torus. Use circles for a terminal+router and lines for the two uni-directional channels in opposite directions.

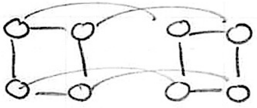
### Constructing k=ary n-cube from k k-ary (n-1) cubes



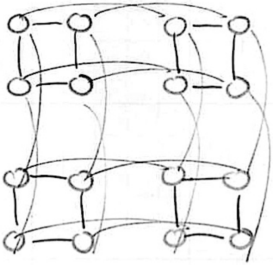
2 ARY 1 cube



2 ARY 2cube



2 ARY 3 CUBE

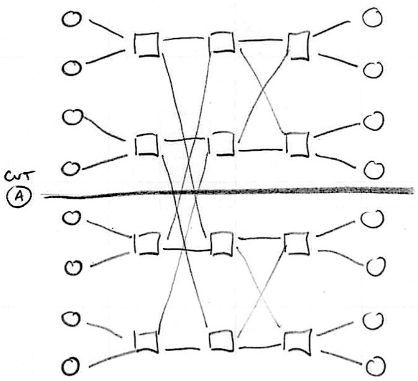


2 ARY 4 CUBE

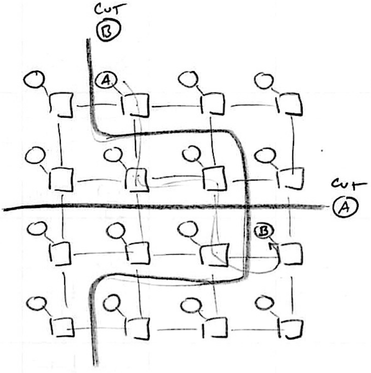
HYPER CUBES

binary 1-cube

## 2.5. Terminology



2-ARY 3FLY



4-ARY 2-CUBE

MESH

### Nodes and Channels

- Uni-directional channels in butterfly
- Bi-directional channels in mesh
- Indirect Network: node is either a terminal or router (butterfly)
- Direct Network: node combines a terminal and router (mesh)
- Channel parameters
  - $w_c$ = channel width (number of pins/wires)
  - $f_c$ = channel frequency
  - $t_c$ = channel latency
  - $l_c$ = channel length
  - $b_c$ = channel bandwidth ($w_c \times f_c$)

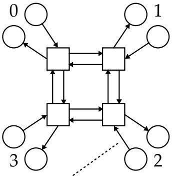
### Bisection Cuts

- Cut: set of channels that partitions terminals into two sets
- Bisection Cut: cut that partitions terminals in two equal halves
- Min Bisection Cut ($B_c$): min channel count over all bisection cuts
- Bisection Bandwidth ($B_B$): min bandwidth over all bisection cuts
- For networks with uniform channel bandwidth: $B_B = B_c \times b_c$
- Bisection bandwidth is a good way to estimate global wiring resources (i.e., technology constraints)
- Example from previous butterfly/mesh topologies
  - Cut A on butterfly is a minimum bisection cut
  - Both cut A and B on mesh are bisection cuts
  - Cut A on mesh is a minimum bisection cut

### Paths

- Channel Hop Count ($H_c$): number of channels on a path
- Channels from terminal to first router may or may not be included
- Router Hop Count ($H_r$): number of routers on a path
- Minimal Path: smallest hop count between two terminals
- Diameter ($H_{max}$): largest minimal path between all terminal pairs
- Average min hop count ($H_{min}$): average over all terminal pairs
- Example from previous butterfly/mesh topologies
  - Path from A to B on mesh: $H_r = 5$, $H_c = 4$
  - $H_{max,bfly} = 4$, $H_{max,mesh} = 8$
  - $H_{min,bfly} = 4$

Calculating $H_{min}$ usually requires enumerating the minimal hop count for every source/destination pair. The book states that $H_{min}$ for a torus with even $k$ is $nk/4$ and for a mesh with even $k$ is $nk/3$. Where does this come from?



"terminal channels"
circles = terminals
squares = routers

CALCULATING $H_{MIN,R}$

$$\begin{array}{cc|cccc|} & & & \text{src} & & \\ & & 0 & 1 & 2 & 3 \\ & 0 & 1 & 2 & 3 & 2 \\ & 1 & 2 & 1 & 2 & 3 \\ \text{Dest} & 2 & 3 & 2 & 1 & 2 \\ & 3 & 2 & 3 & 2 & 1 \end{array}$$

$1 \times 4 = 4$
$2 \times 8 = 16$
$3 \times 4 = 12$

including $i \to i$ $32/16 = 2$
excluding $i \to i$ $28/12 = 2.3$

CALCULATING $H_{MIN,C}$ WITH TERMINAL CHANNELS

$$\begin{array}{cc|cccc|} & & & \text{src} & & \\ & & 0 & 1 & 2 & 3 \\ & 0 & 2 & 3 & 4 & 3 \\ & 1 & 3 & 2 & 3 & 4 \\ \text{Dest} & 2 & 4 & 3 & 2 & 3 \\ & 3 & 3 & 4 & 3 & 2 \end{array}$$

$2 \times 4 = 8$
$3 \times 8 = 24$
$4 \times 4 = 16$

including $i \to i$ $40/16 = 2.5$
excluding $i \to i$ $40/12 = 3.3$

CALCULATING $H_{MIN,C}$ WITHOUT TERMINAL CHANNELS

$$\begin{array}{cc|cccc|} & & & \text{src} & & \\ & & 0 & 1 & 2 & 3 \\ & 0 & 0 & 1 & 2 & 1 \\ & 1 & 1 & 0 & 1 & 2 \\ \text{Dest} & 2 & 2 & 1 & 0 & 1 \\ & 3 & 1 & 2 & 1 & 0 \end{array}$$

$1 \times 8 = 8$
$2 \times 4 = 8$

including $i \to i$ $16/16 = 1$
excluding $i \to i$ $16/12 = 1.3$ <

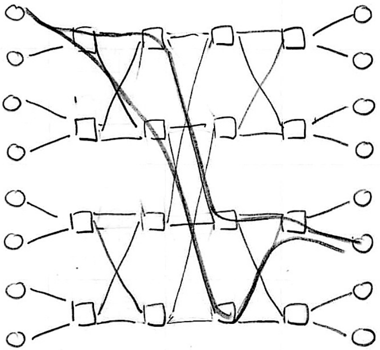
EQUATION FROM BOOK IS $\frac{nk}{3}$ for even n

$$\frac{nk}{3} = \frac{2 \cdot 2}{3} = 1.33$$

THIS IS FOR $H_{MIN,C}$
WITHOUT TERMINAL CHANNELS
+ IGNORING $i$ SENDING TO ITSELF

### Path Diversity

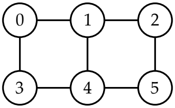
- Path diversity is number of paths between any two terminals
- Mesh has high path diversity, butterfly has no path diversity

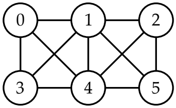


ADDING EXTRA
BFLY STAGES
↑ PATH DIVERSITY

### Example Topologies

Mesh





SMesh

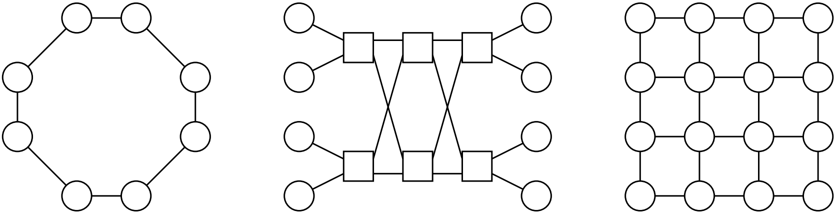
Assume $b_c$ for mesh is 32 bits/cycle, while $b_c$ for smesh is 16 bits/cycle. smesh has reduced channel bandwidth due to increased number of channels, assumes specific technology constraint. Calculate the following parameters for each topology.

| | $B_C$ | $B_B$ | $H_{max}$ | $H_{min}$ |
|---|---|---|---|---|
| mesh | | | | |
| smesh | | | | |

# 3. Network Routing

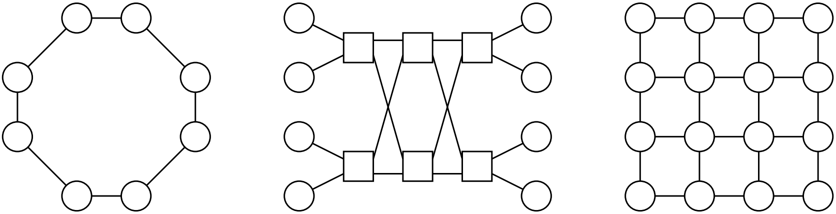
Network routing is the choice of path from source to destination along channels and routers. Routing algorithms can be oblivious (not factor in network state) or adaptive (factor in network state), and can also be deterministic or non-deterministic.

## 3.1. Oblivious Deterministic Routing



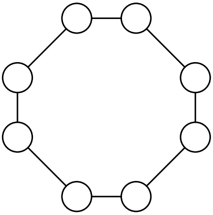
- Ring: always use minimal path, equidistant choose CW
- Butterfly: use destination to choose middle router
- Mesh: route in X first, then route in Y (Dimension-Ordered-Routing)

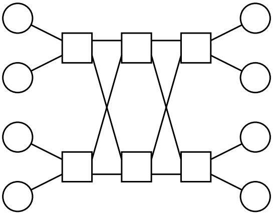
## 3.2. Oblivious Non-Deterministic Routing

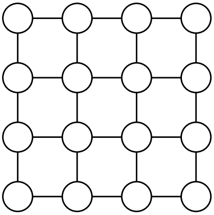


- Ring: randomly choose CW vs CCW
- Butterfly: randomly choose middle router
- Mesh: randomly choose between XY-DOR and YX-DOR (O1-TURN)

## 3.3. Adaptive Routing





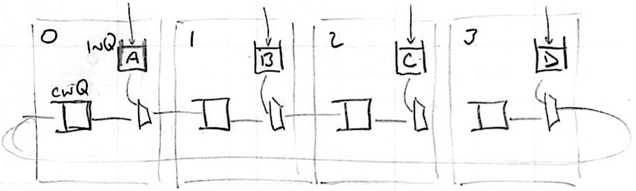


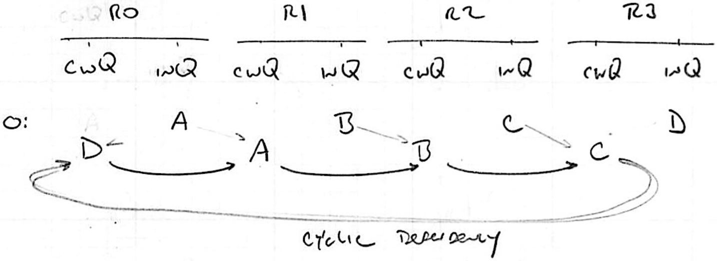
- Ring: look at queues, choose direction with least congestion
- Butterfly: look at queues, choose middle router with least congestion
- Mesh: look at queues during each hop to choose X vs Y

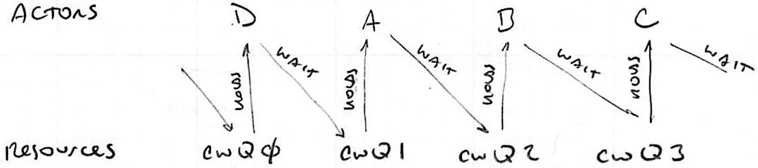
## 3.4. Deadlock

CYCLIC DEPENDENCY IN "WAITS FOR" AND "HOLDS" RELATION

ASSUME TORNADO $i \rightarrow i+2$







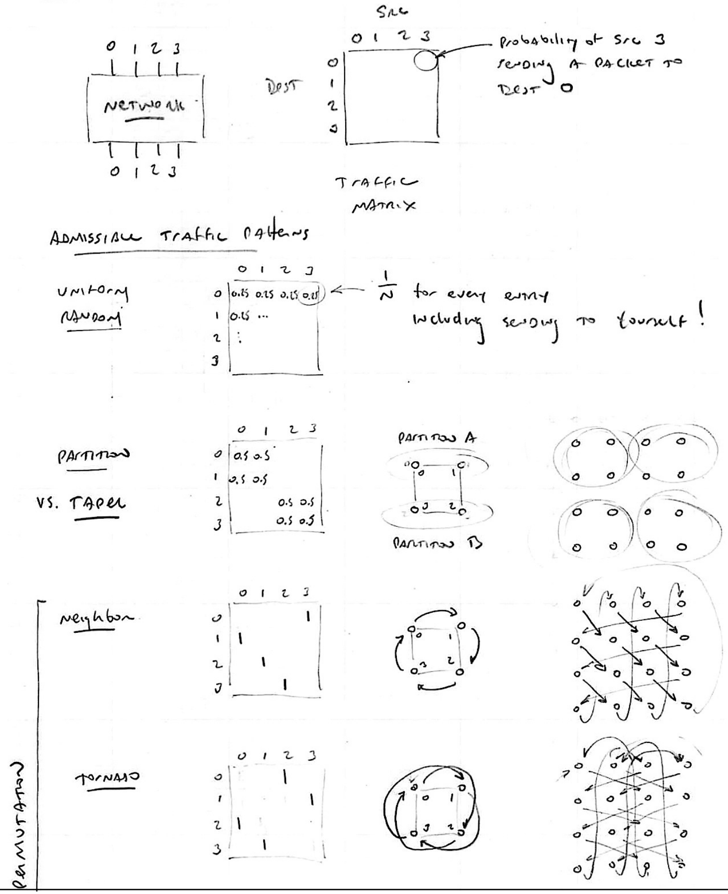
- DEADLOCK AVOIDANCE VS DEADLOCK DETECTION / RECOVERY

CONTRAST TO LIVELOCK
CHAOS ROUTING ALGORITHM

# 4. Analyzing Network Performance

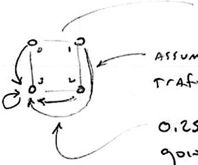
Similar to analyzing processors and memories, we will use simple first-order equations to help build intuition about throughput and latency trade-offs in networks.

## 4.1. Traffic Patterns



## Hot spot

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & & & & \\ 1 & & & & \\ 2 & & & & \\ 3 & 0.25 & 0.25 & 0.25 & 0.25 \end{array}$$



Assume $b_c$

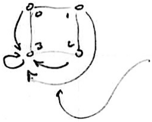
Assume b/cycle traffic injected

0.25b + 0.25b = 0.5b going over 2→3 channel

## Inadmissible traffic patterns

Oversubscribed hotspot

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & & & & \\ 1 & & & & \\ 2 & & & & \\ 3 & 1 & 1 & 1 & 1 \end{array}$$

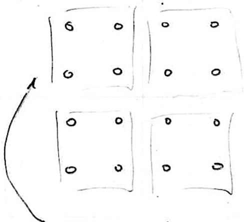


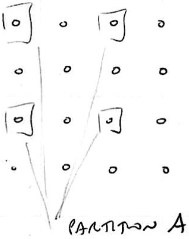
1b + 1b = 2b going over this 2→3 channel

Channel 2→3 oversubscribed but so is ejection channel to output terminal at node 3

## Logical to physical mapping

- Assume partition traffic pattern. How are logical src/dest IDs in traffic pattern mapped to physical terminal IDs?

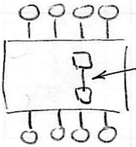




Mapping can turn any logical permutation pattern into any other pattern.

Given permutation pattern usually assumes a specific mapping

## 4.2. Ideal Throughput



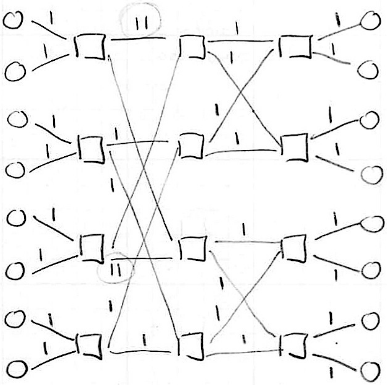
Bottleneck channel will limit aggregate throughput

Channel load ($\gamma_c$) is amount of traffic that crosses channel c if each input injects one unit of traffic according to given traffic pattern

Ex traffic pattern

src $i \longrightarrow$ dest $i+1$

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 |   |   |   |   |   |   |   | 1 |
| 1 | 1 |   |   |   |   |   |   |   |
| 2 |   | 1 |   |   |   |   |   |   |
| 3 |   |   | 1 |   |   |   |   |   |
| 4 |   |   |   | 1 |   |   |   |   |
| 5 |   |   |   |   | 1 |   |   |   |
| 6 |   |   |   |   |   | 1 |   |   |
| 7 |   |   |   |   |   |   | 1 |   |



Channel load ranges from 0-2

Max channel load ($\gamma_{max}$) is 2. These are the bottleneck channels that will limit throughput

Alternative way to think about $\gamma$.

Channel load is ratio of BW demanded from channel to input BW injected by one terminal

IDEAL THROUGHPUT $\Theta_{TERM} = \frac{b_c}{\gamma_{MAX}}$

BW of bottleneck channel

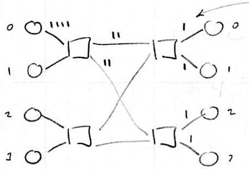
max channel load

In previous example

$\Theta_{TERM} = \frac{b}{\gamma_{MAX}} = \frac{b}{2}$ $\Theta_{TOT} = N \cdot \frac{b}{\gamma_{MAX}} = 8 \cdot \frac{b}{2} = 4b$

SMALL BFLY EXAMPLES

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0.25 & 0.25 & 0.25 & 0.25 \\ 1 & 0.25 & 0.25 & 0.25 & 0.25 \\ 2 & 0.25 & 0.25 & 0.25 & 0.25 \\ 3 & 0.25 & 0.25 & 0.25 & 0.25 \end{array}$$

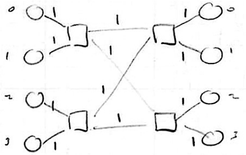


each tick = 1/4 unit

$\gamma_{MAX} = 1$

$\Theta_{TERM} = b$

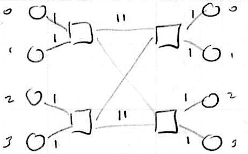
$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & & & 1 & \\ 1 & 1 & & & \\ 2 & & & & 1 \\ 3 & & 1 & & \end{array}$$



$\gamma_{MAX} = 1$

$\Theta_{TERM} = b$

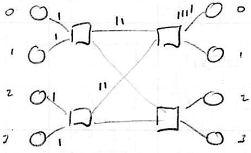
$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & & 1 & & \\ 1 & 1 & & & \\ 2 & & & & 1 \\ 3 & & & 1 & \end{array}$$



$\gamma_{MAX} = 2$

$\Theta_{TERM} = b/2$

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 1 & 1 & 1 \\ 1 & & & & \\ 2 & & & & \\ 3 & & & & \end{array}$$



$\gamma_{MAX} = 4$

$\Theta_{TERM} = b/4$

More generally for uniform random traffic

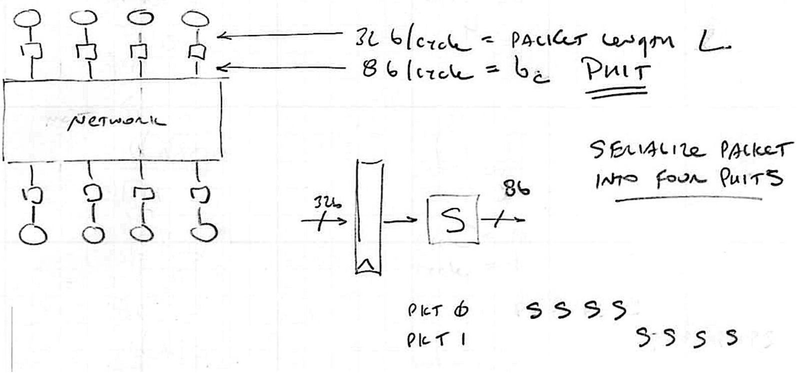
- on average with uniform random traffic, half the traffic crosses the bisection
- N total units of traffic, N/2 cross bisection
- ideal routing will evenly balance load across bisection channels (this was an issue in ring example)
- so

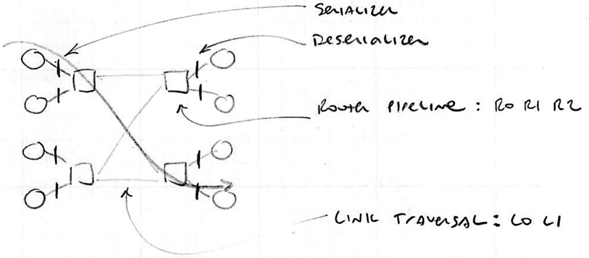
$$\gamma_{MAX} = \frac{N/2}{B_c} = \frac{N}{2B_C}$$

$$\Theta_{TERM} = \frac{b}{N/2B_C} = \frac{2bB_c}{N} = \frac{2B_B}{N}$$

$$\Theta_{TOT} = N\frac{2B_B}{N} = 2B_B$$

## 4.3. Zero-Load Latency





```
PKT 1   HEAD FLIT → S R0 R1 R2 L0 L1 R0 R1 R2 D
        BODY FLIT     S R0 R1 R2 L0 L1 R0 R1 R2 D
        BODY FLIT       S R0 R1 R2 L0 L1 R0 R1 R2 D
        TAIL FLIT         S R0 R1 R2 L0 L1 R0 R1 R2 D →
```

SERIALIZATION LATENCY $L/b$

CHANNEL LATENCY $t_c$

ROUTER PIPELINE LATENCY $t_R$

$$T = \underbrace{T_{HEAD}}_{} + \underbrace{\frac{L}{b}}_{}$$

→ serialization latency

↳ HEAD FLIT LATENCY INCLUDES $t_c$, $t_r$, HOP COUNT, + <u>CONTENTION</u>

$$\underbrace{T_\phi}_{} = \underbrace{H_R t_r}_{} + \underbrace{H_c t_c}_{} + \underbrace{\frac{L}{b}}_{}$$

LATENCY DUE TO ROUTER HOPS

LATENCY DUE TO CHANNEL HOPS

SERIALIZATION LATENCY

<u>ZERO LOAD LATENCY</u> (NO CONTENTION)

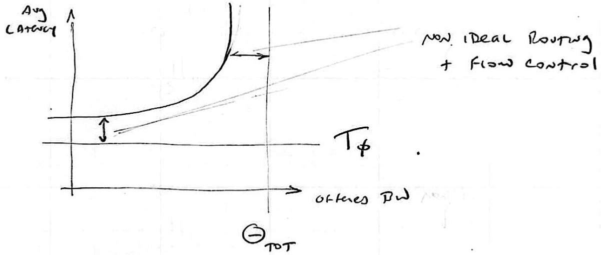
## FOUR WAYS TO IMPROVE LATENCY

$$T_\phi = H_R t_R + H_c t_c + \frac{L}{b}$$

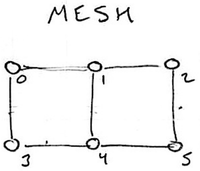
wider channels or shorter msgs

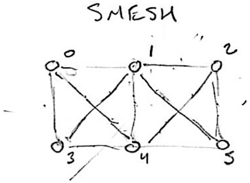
Shorter Routes Faster routers Faster channels

## AVG LATENCY VS OFFERED BW



## 4.4. Comparing Topologies





ASSUME :

$L = 128$ b

$b_c$ for mesh is 32 b/cycle

$b_c$ for smesh is 16 b/cycle ← reduces channel bandwidth due to increased # channels assuming constant resources

$t_r = 3$

$t_c = 1$.

WHICH TOPOLOGY CAN ACHIEVE HIGHER IDEAL THROUGHPUT UNDER UNIFORM RANDOM TRAFFIC?

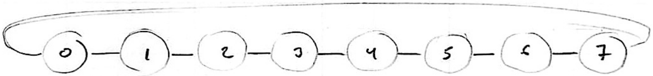
(a) $\Theta_{Term,\, mesh} > \Theta_{term,\, smesh}$

(b) $\Theta_{term,\, mesh} < \Theta_{term,\, smesh}$

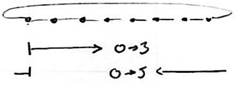
(c) $\Theta_{Term,\, mesh} == \Theta_{term,\, smesh}$

CALCULATE ZERO LOAD LATENCY UNDER UNIFORM RANDOM TRAFFIC

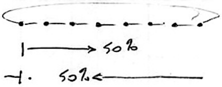
## 4.5. Comparing Routing Algorithms



- GREEDY : Always use minimal path, equidistant choose randomly

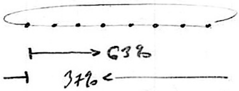


- UNIFORM RANDOM : Randomly pick direction



for 0 → 3
50% take 4 hops
50% take 6 hops

- WEIGHTED RANDOM : Randomly pick direction but weight probability by distance



for 0 → 3
5/8 time take 3 hop path
3/8 time take 5 hop path

Probability of taking short path is

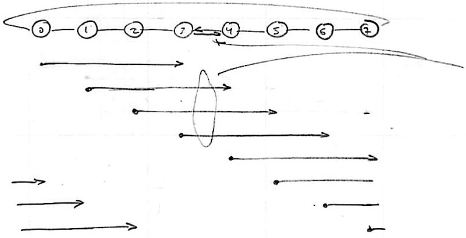
$$\frac{N - H_{MIN,R} + 1}{N}$$

- ADAPTIVE : Look at queues in either direction send in direction of queue that has most free entries.

Do not change direction after initial choice

- Evaluate tornado traffic pattern on the 8-node ring
- Recall that in tornado, node $i$ sends to $i + ((N-1)/2) \bmod N$

GREEDY ALGO



3 PATHS ACROSS THIS CHANNEL

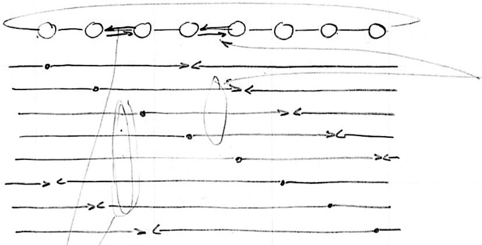
$\gamma_{MAX} = 3$

$\Theta_{TERM} = 0.33$

CLOCKWISE CHANNELS NOT USED AT ALL!
POOR LOAD BALANCING

$T_0 = 4$

URANDOM ALGO



STILL 3 PATHS CROSS THIS CHANNEL BUT EACH PATH ONLY CARRIES 0.5 UNITS OF TRAFFIC

$\gamma_{MAX,CCW} = 1.5$

WHAT ABOUT CLOCKWISE CHANNELS THOUGH?

5 PATHS CROSS EACH CW CHANNEL
EACH IS 0.5 UNIT OF TRAFFIC

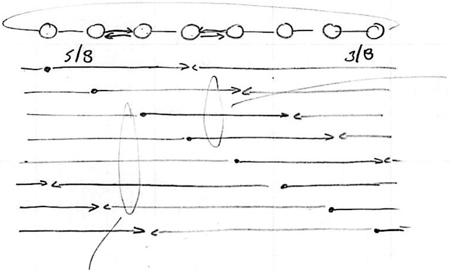
$\gamma_{MAX,CW} = 2.5 > \gamma_{MAX,CCW} = 1.5$ $\Theta_{TERM} = 0.4$

NOW THE CLOCKWISE CHANNELS ARE THE BOTTLENECK!

$T_0 = 0.5 \times 4 + 0.5 \times 6 = 5$

## Weighted Random Algo

Same number of paths cross CW and CCW channels as before except now with different amounts of traffic per path.



3 paths each with 5/8 traffic

$$\gamma_{max,ccw} = 3 \cdot \frac{5}{8} = 1.875$$

5 paths each with 3/8 traffic

$$\gamma_{max,cw} = 5 \cdot \frac{3}{8} = 1.875$$

$$\gamma_{max,cw} = \gamma_{max,ccw}$$ Balanced!

$$\Theta_{TEM} = \frac{1}{1.875} = 0.53$$

$$T_0 = \frac{5}{8} \times 4 + \frac{3}{8} 6 = 2.5 + 2.25 = 4.75$$

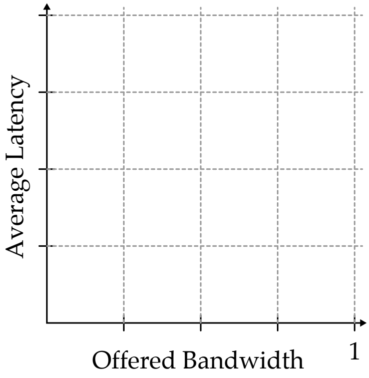
## Adaptive

Minimal latency at light load
Max throughput at high load

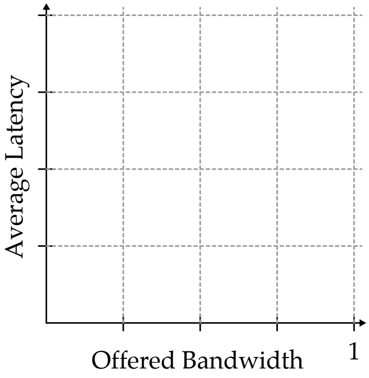
Let's fill in the following table to summarize the performance of the four routing algorithms on the 8-node given the tornado and uniform random traffic patterns.

| | **Tornado** | | **Uniform Random** | |
|---|---|---|---|---|
| **Routing Algo** | $\Theta_{term}$ | $T_0$ | $\Theta_{term}$ | $T_0$ |
| Greedy | | | 1.00 | 3.0 |
| Uniform Random | | | 0.57 | 4.5 |
| Weighted Random | | | 0.76 | 3.1 |
| Adaptive | | | 1.00 | 3.0 |

**Tornado Traffic Pattern**
Bandwidth vs. Latency



**Uniform Random Traffic Pattern**
Bandwidth vs. Latency



### Activity: Routing on butterfly with extra stage

Consider two routing algorithms for a 2-ary 2-fly with an extra stage:

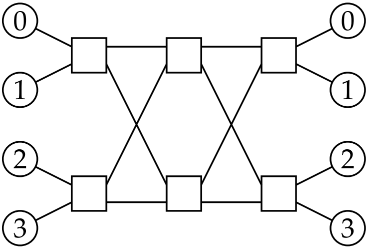
- Destination Tag Routing: use destination to choose middle router
- Random Middle Router: randomly choose middle router

Compare these two routing algorithms in terms of ideal terminal throughput ($\Theta_{term}$) and zero-load latency ($T_0$) for the following permutation traffic pattern.

- src 0 $\rightarrow$ dest 1
- src 1 $\rightarrow$ dest 0
- src 2 $\rightarrow$ dest 3
- src 3 $\rightarrow$ dest 2

*Hint: Use the tick-mark method to calculate the max channel load for each routing algorithm.*

**Destination-Tag Routing**



**Random Middle Routing**