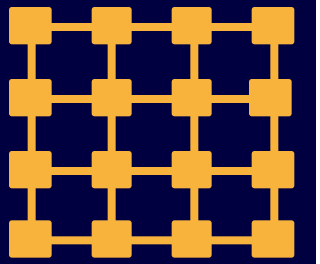
ECE 8823 A / CS 8803 - ICN
Interconnection Networks
Spring 2017

http://tusharkrishna.ece.gatech.edu/teaching/icn_s17/



Lecture 2: Topology - I

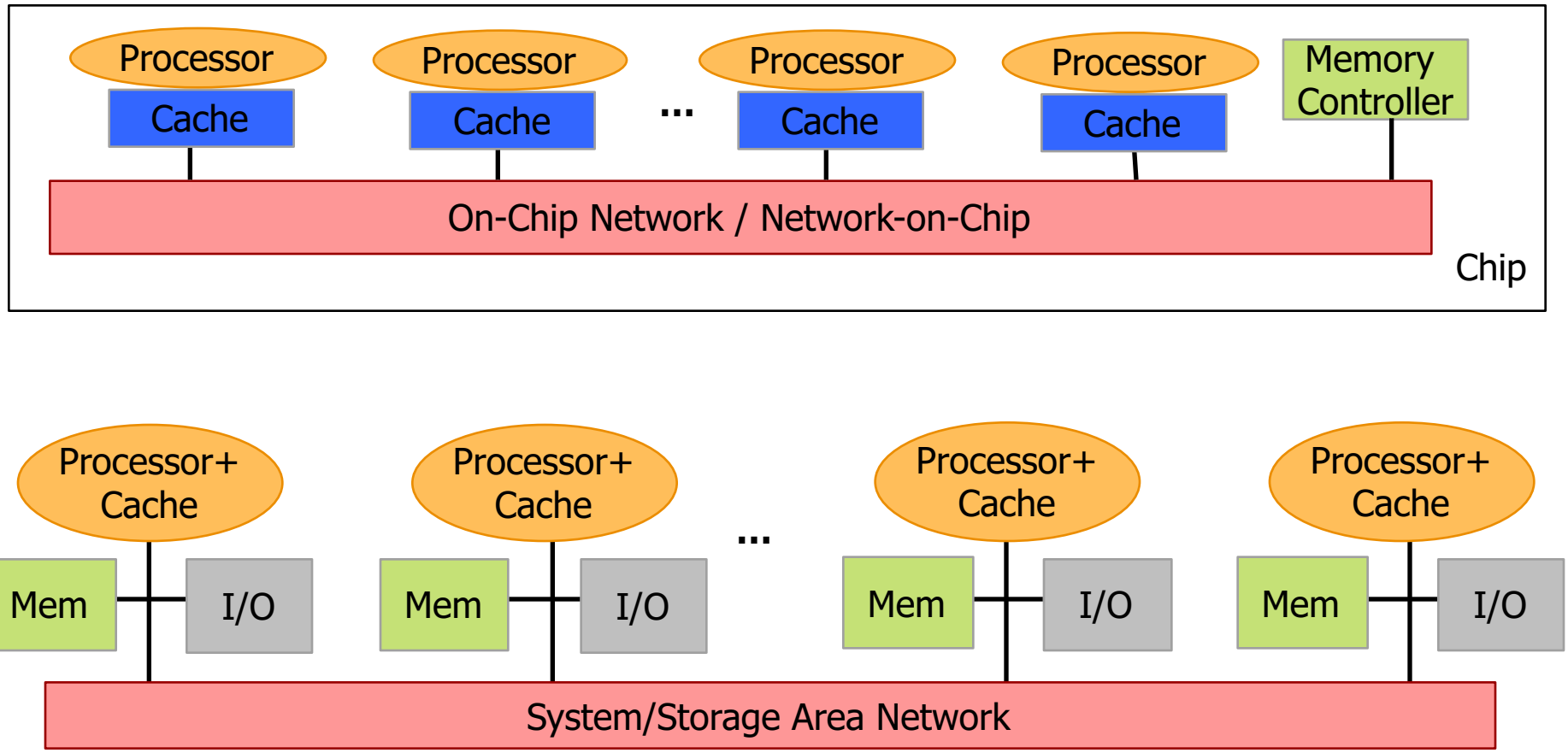
Tushar Krishna

Assistant Professor
School of Electrical and Computer Engineering
Georgia Institute of Technology

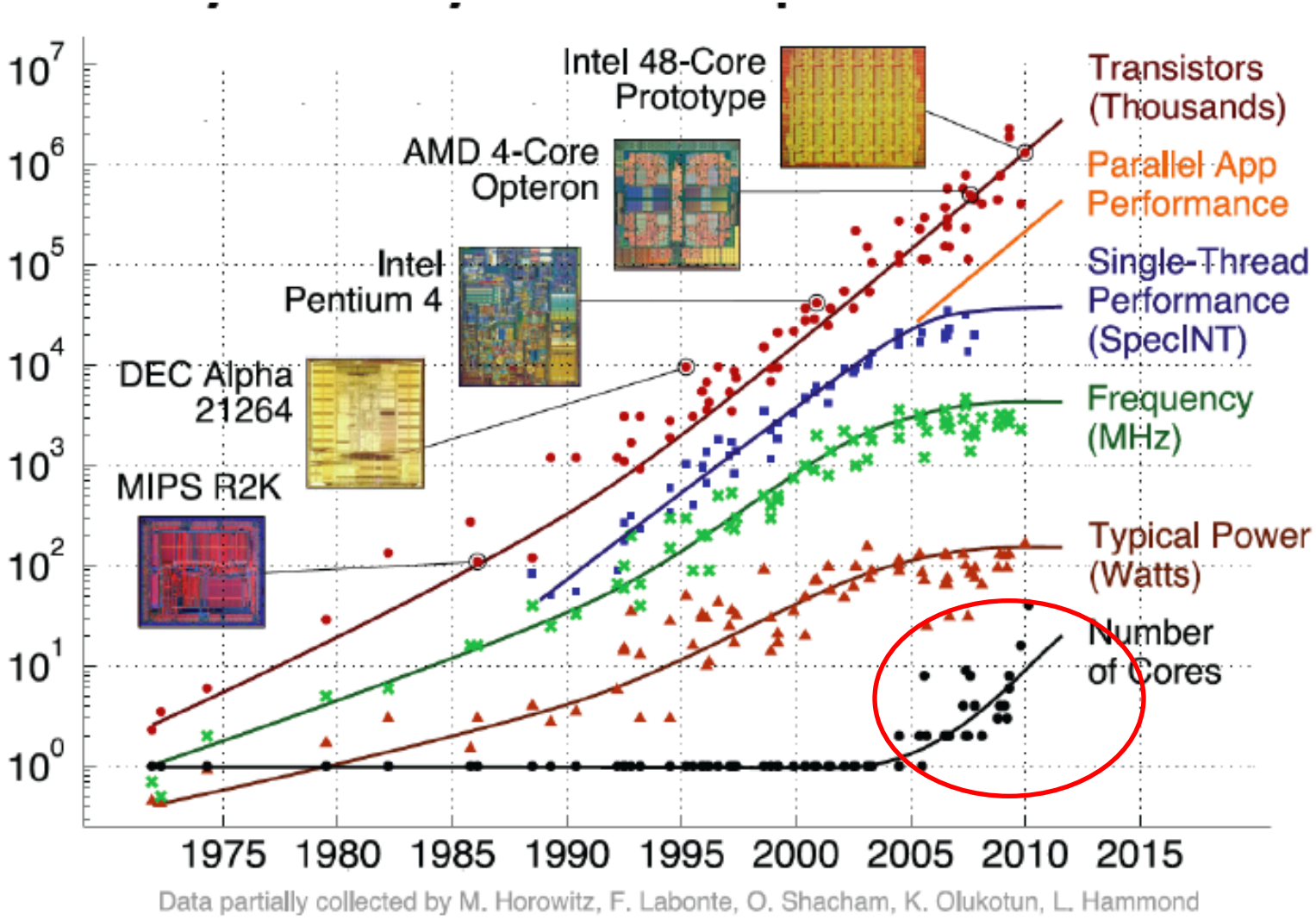
tushar@ece.gatech.edu

Acknowledgment: Slides adapted from Univ of Toronto ECE 1749 H (N Jerger) and MIT 6.883 (L-S Peh)

Recap



Why NoCs are important

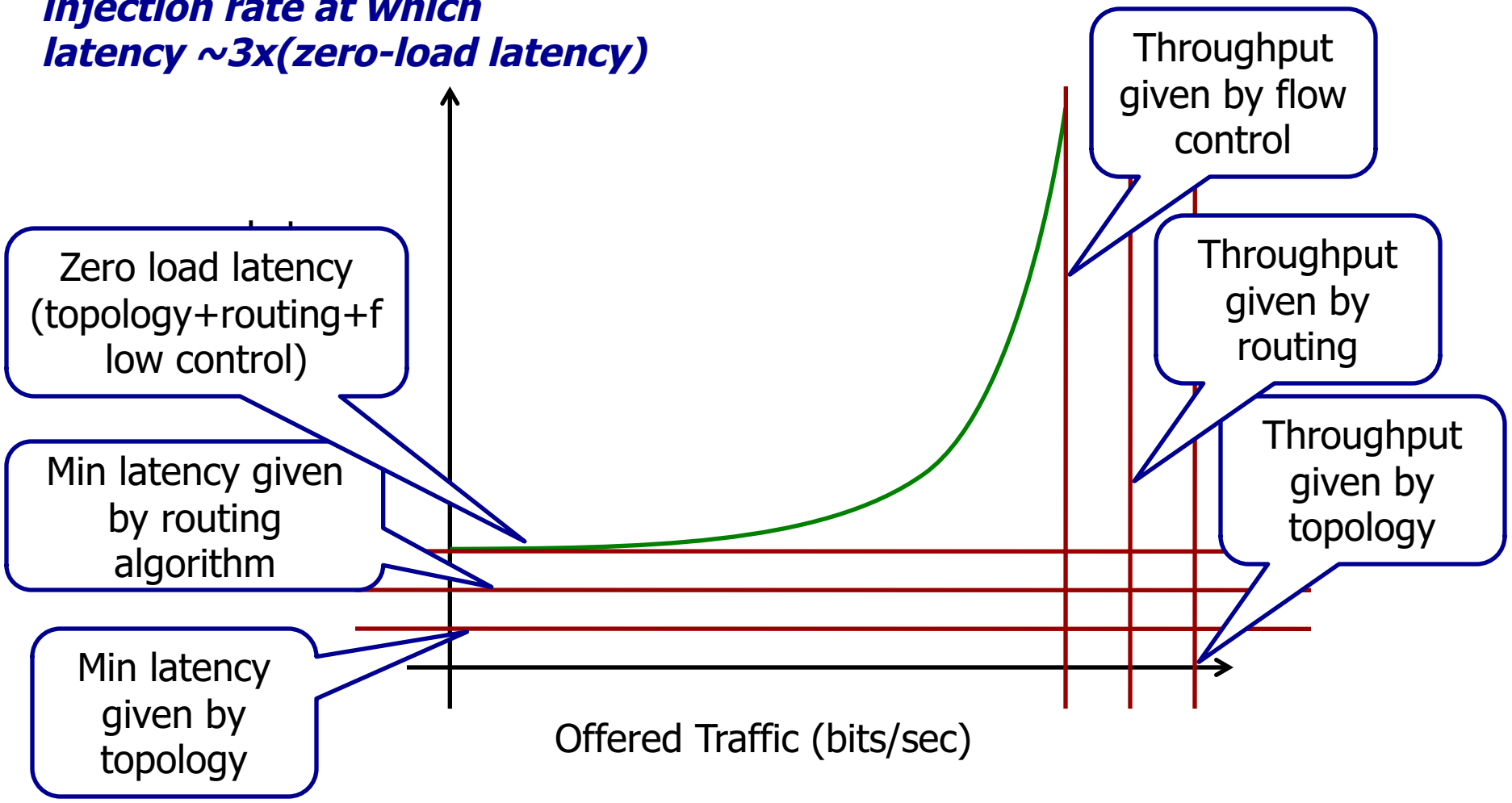


Network Architecture

- **Topology**
  - How to connect the nodes
  - ~Road Network
- **Routing**
  - Which path should a message take
  - ~Series of road segments from source to destination
- **Flow Control**
  - When does the message have to stop/proceed
  - ~Traffic signals at end of each road segment
- **Router Microarchitecture**
  - How to build the routers
  - ~Design of traffic intersection (number of lanes, algorithm for turning red/green)

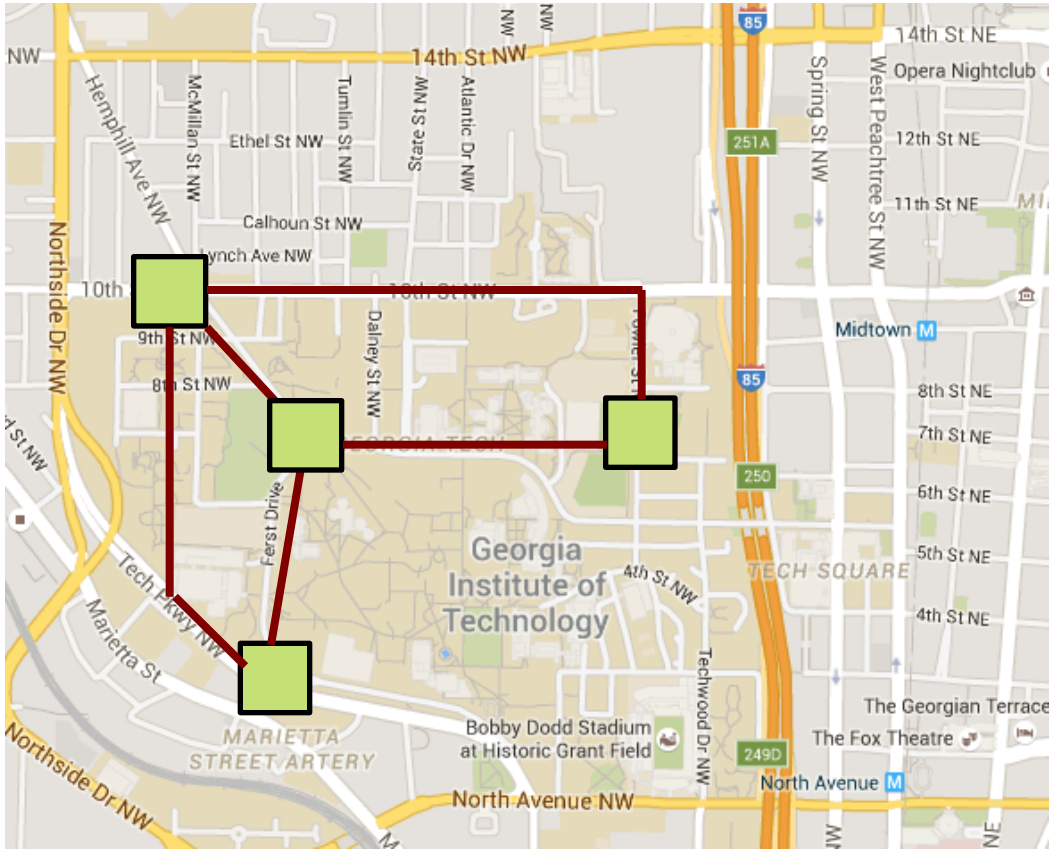
Network Performance

Saturation Throughput: the injection rate at which latency ~3x(zero-load latency)



Topology:
How to connect the nodes with links

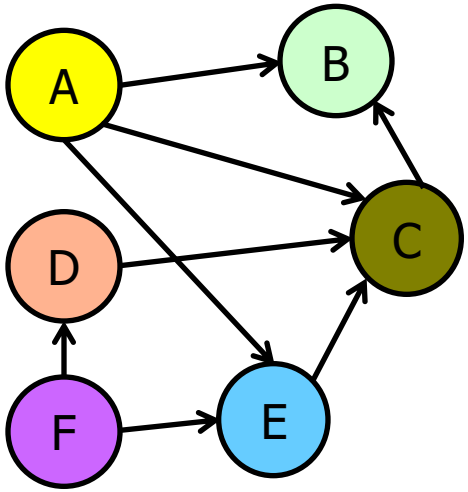
~Road Network



Topology Overview

- Often the first step in network design
- 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?

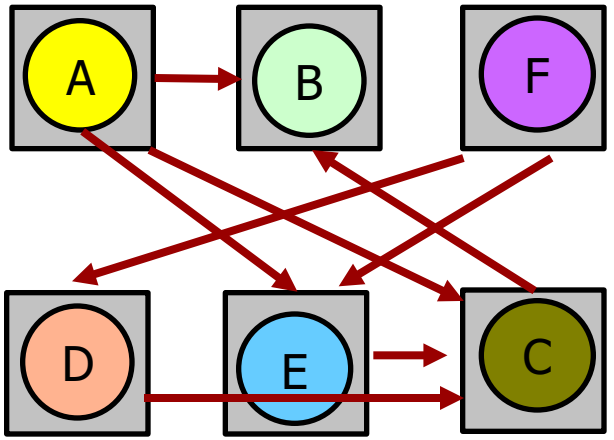


Application's Task Communication Graph

Vertices - tasks
Edges - communication

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

Best topology?



Network Topology Graph

Vertices - cores
Edges - links

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

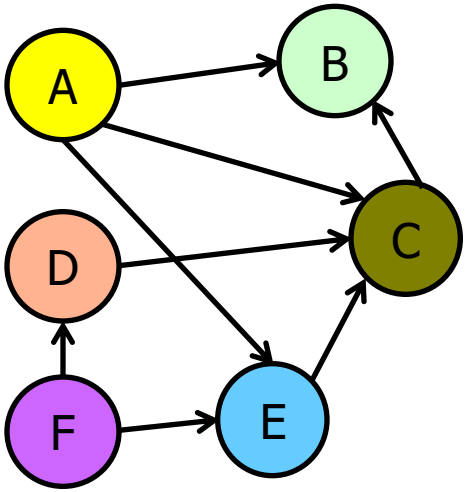
Let us define some *design-time* metrics

- **Degree** – number of ports at a node
  - Proxy for area/energy cost
- **Bisection Bandwidth** - bandwidth crossing 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)
- **Diameter** – maximum routing distance (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 and mapping
  - *Maximum channel load determines throughput*
- **Path diversity**
  - Number of shortest paths between a communicating pair
    - Can be exploited by routing algorithm
    - Provides **fault tolerance**

Regular topologies

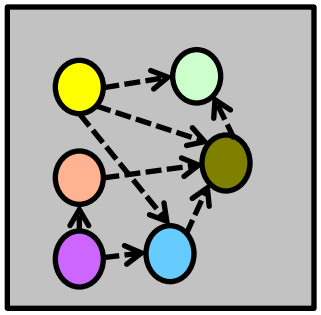


Application's Task Communication Graph

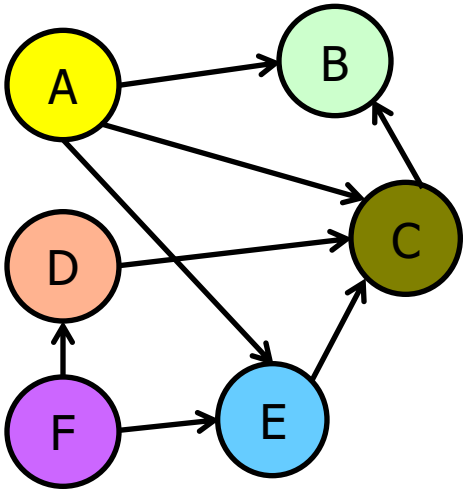
Vertices - tasks
Edges - communication

Can you suggest a regular topology (each router with same degree) with smallest possible diameter?

Trick question :p
One node. Degree = 0, Diameter = 0.



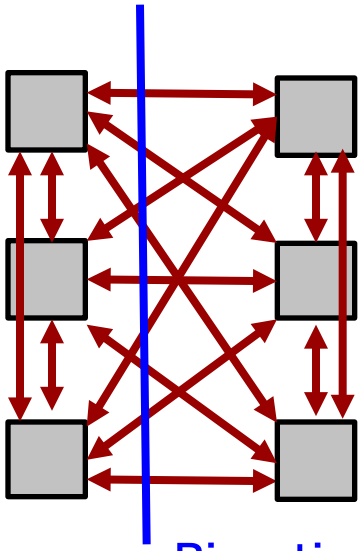
Regular topologies



Application's Task Communication Graph

Vertices - tasks
Edges - communication

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



Bisection Cut

Challenge?

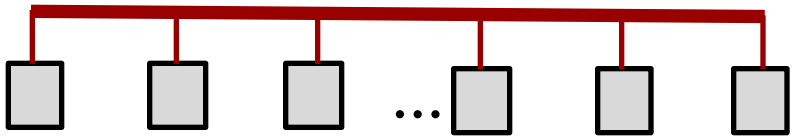
Not scalable!! Cannot layout more than 4-6 cores in this manner for area and power reasons

Fully Connected

Degree = ? 5

Bisection BW = ? 9

Bus



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

Route Packets, Not Wires!

Route Packets, Not Wires: On-Chip Interconnection Networks

William J. Dally and Brian Towles
Computer Systems Laboratory
Stanford University
Stanford, CA 94305
{billd,btowles}@cva.stanford.edu

Abstract

Using on-chip interconnection networks in place of ad-hoc global wiring structures the top level wires on a chip and facilitates modular design. With this approach, system modules (processors, memories, peripherals, etc...) communicate by sending packets to one another over the network. The structured network wiring gives well-controlled electrical parameters that eliminate timing iterations and enable the use of high-performance circuits to reduce latency and increase bandwidth. The area overhead required to implement an on-chip network is modest, we estimate 6.6%. This paper introduces the concept of on-chip networks, sketches a simple network, and discusses some challenges in the architecture and design of these networks.

1 Introduction

We propose replacing design-specific global on-chip wiring with a general-purpose on-chip interconnection network. As shown in Figure 1, a chip employing an on-chip network is composed of a number of network clients: processors, DSPs, memories, peripheral controllers, gateways to networks on other chips, and custom logic. Instead of connecting these top-level modules by routing dedicated wires, they are connected to a network that routes packets between them. Each client is placed in a rectangular tile on the chip and communicates with all other clients, not just its neighbors, via the network. The network logic occupies a small amount of area (we estimate 6.6%) in each tile and makes use of a portion of the upper two wiring layers.

Using a network to replace global wiring has advantages of structure, performance, and modularity. The on-chip network structures the global wires so that their electrical properties are optimized and well controlled. These controlled electrical parameters, in particular low and predictable cross-talk, enable the use of aggressive signaling circuits that can reduce power dissipation by a factor of ten and increase propagation velocity by three times [3]. Sharing the wiring resources between many communication flows makes more efficient use of the wires: when one client is idle, other clients continue to make use of the network resources.

An on-chip interconnection network facilitates modularity by defining a standard interface in much the same manner as a backplane bus. For the past three decades systems have been constructed by plugging modules into standard backplane buses such as VME or PCI. The definition of a standard interface facilitates reusability and interoperability of the modules. Also, standard interfaces allow shared interconnect to be highly optimized since its development cost can be amortized across many systems.

Of course, these modularity advantages are also realized by on-chip buses [1][5][8], a degenerate form of a network. Networks are generally preferable to such buses because they have higher bandwidth and support multiple concurrent communications. Some of our motivation for intra-chip networks stems from the use of inter-chip networks to provide general system-level interconnect [7].

The remainder of this paper describes our initial thoughts on the design of on-chip interconnection networks. To provide a baseline, we start in Section 2 by sketching the design of a simple on-chip network. Section 3 revisits the design choices made in this simple network and discusses the challenges and open research issues in the design of such networks. Section 4 discusses the advantages and disadvantages of on-chip networks.

2 Example on-chip interconnection network

To give a flavor for on-chip interconnection networks this section sketches the design of a simple network. Consider a 12mm x 12mm chip in 0.1µm CMOS technology with an 0.5µm minimum wire pitch. As shown in Figure 1, we divide this chip into 16 3mm x 3mm tiles. A system is composed by placing client logic (e.g., processors, DSPs, peripheral controllers, memory subsystems, etc.) into the tiles. The client logic blocks communicate with one another only over the network. There are no top-level connections other than the network wires.

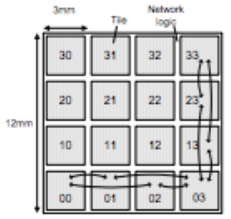


Figure 1 Partitioning the die into module tiles and network logic

DAC 2001, June 18-22, 2001, Las Vegas, Nevada, USA.

Dally and Towles,
DAC 2001

The birth of "on-chip" ***switched*** networks

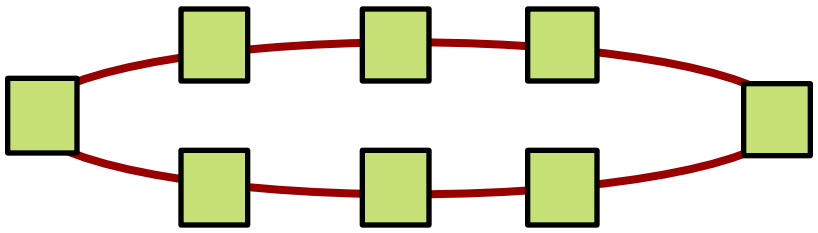
Topology Classification

- **Direct**
  - Each router (switch) is associated with a terminal node
  - All routers are sources and destinations of traffic
  - Example: Ring, Mesh, Torus
    - Most on-chip networks use direct topologies

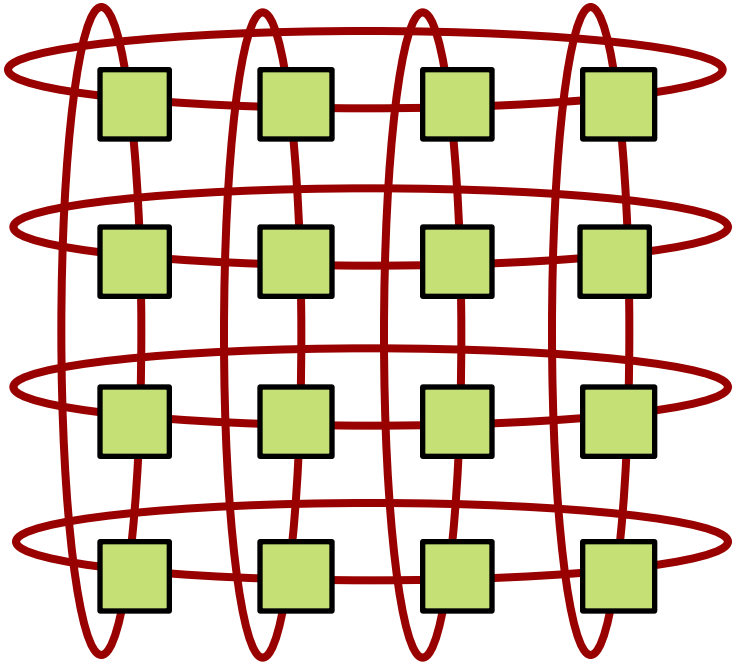
- **Indirect**
  - Routers (switches) are distinct from terminal nodes
  - Terminal nodes can source / sink traffic
  - Intermediate nodes switch traffic
  - Examples: Crossbar, Butterfly, Clos, Omega, Benes, …
    - Next lecture

Ring and Torus

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

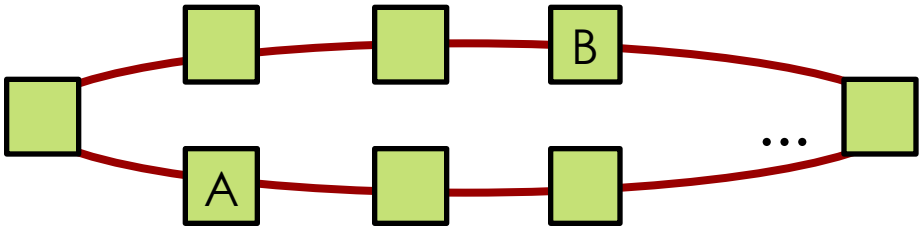


8-ary 1-cube



4-ary 2-cube

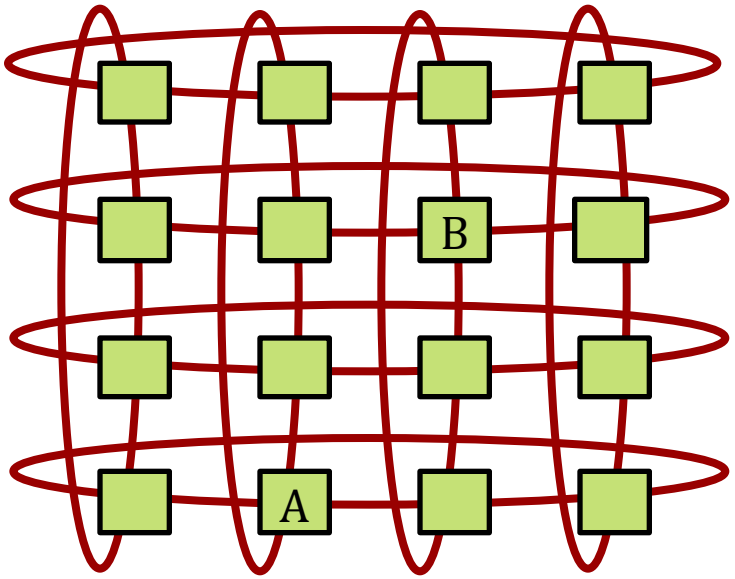
Ring



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

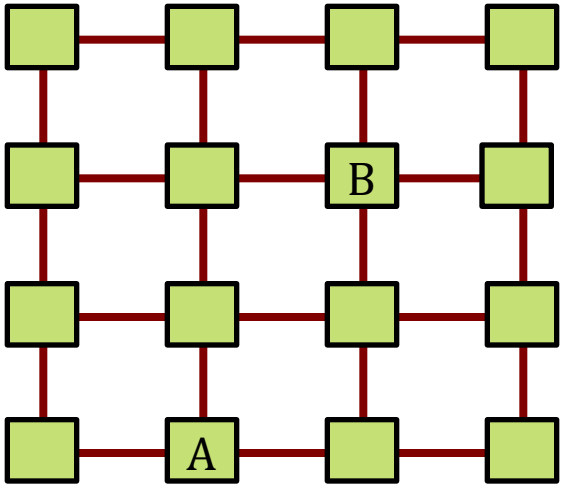
Torus



Diameter?	√N
Bisection BW?	2√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

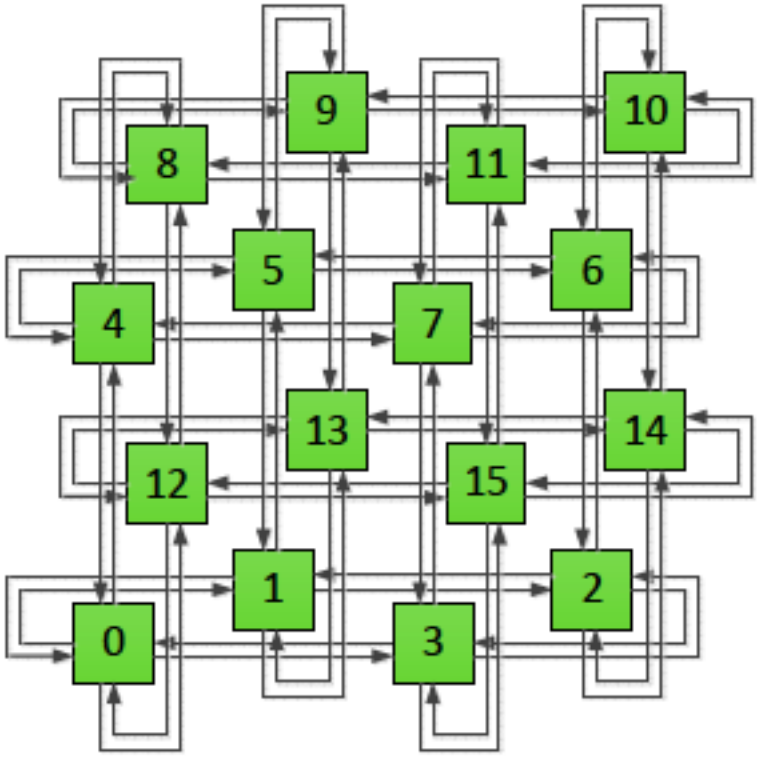
Mesh



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)

Folded Torus



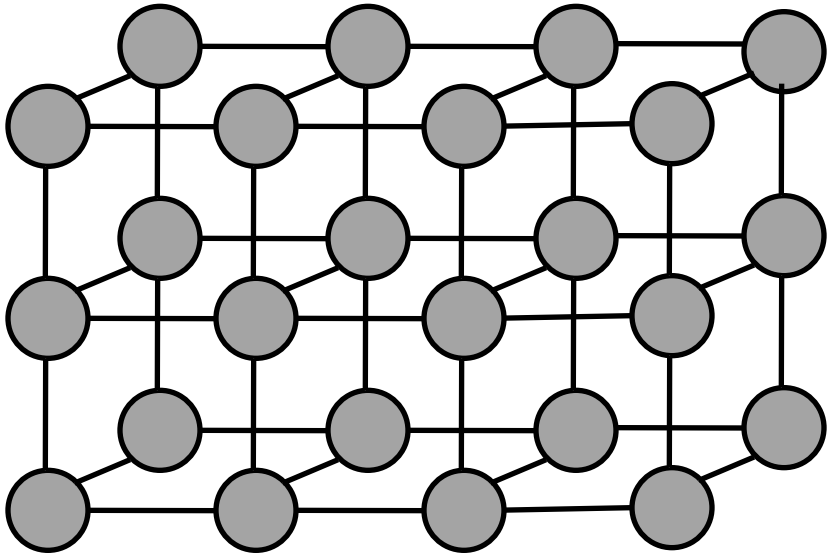
Easier to layout

Is there any con compared to the mesh?

All channels have double the length

Multi-dimensional topologies

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

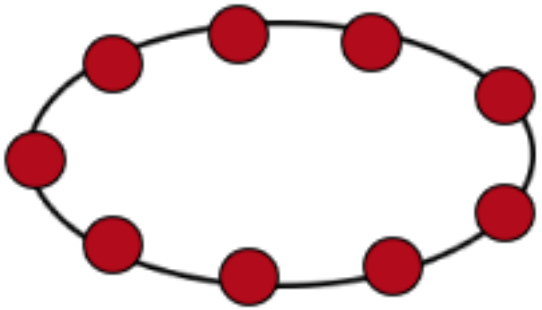


2,3,4-ary 3 Mesh

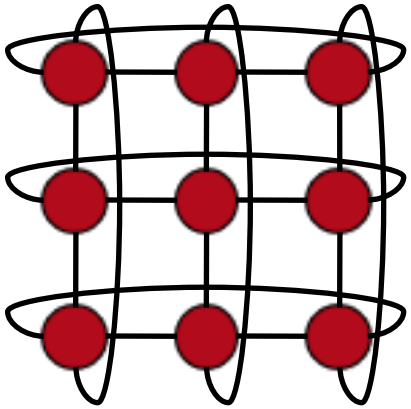
Run-Time Metrics

- Hop Count
  - Latency
- Maximum Channel Load
  - Throughput
- We will consider Uniform Random Traffic

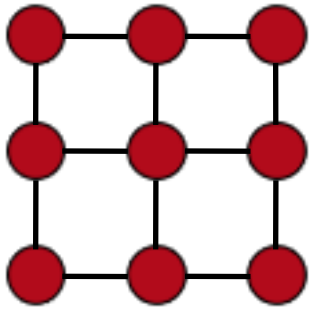
Hop Count



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

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

k-ary n mesh

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

Network Latency

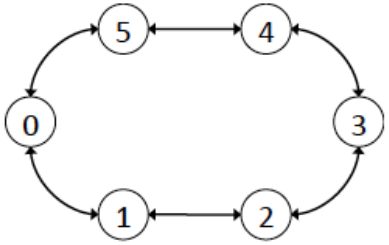
- $T = H.t_r + T_w + T_s + T_c$
  - H = number of hops
  - t_r = router delay
  - T_w = wire delay
  - T_s = serialization delay
  - T_c = contention delay
- $T = H.t_r + D / v + L / b + T_c$
  - D = wire distance
  - v = propagation velocity
  - L = packet length
  - b = channel bandwidth

How to reduce hop count?

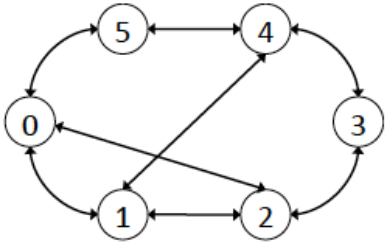
- Low-diameter topology
  - Challenge?
    - high-radix of each switch

- Some dedicated long-range links
  - High-radix for *few* switches
  - How to decide where to add long links?

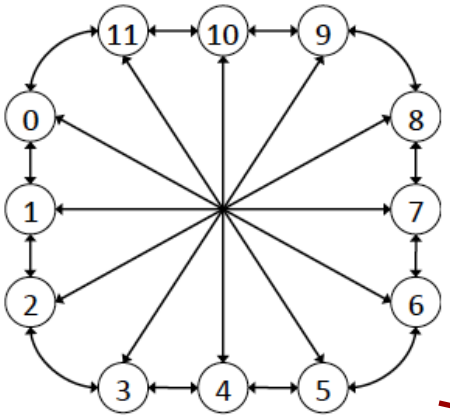
ST Spidergon



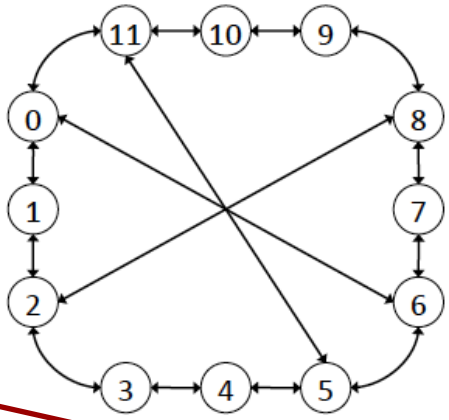
(a) 6-node Spidergon



(b) 6-node Spidergon with extra links

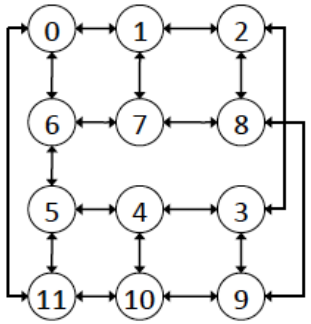


(c) 12-node Spidergon with all links



(d) 12-node Spidergon with links removed

- Proprietary NoC from ST Microelectronics
- Pseudo-regular topology
  - All routers have 2 or 3 ports
  - Depending on application BW needs, links can be added removed
    - e.g, (a) vs. (b), and (c) vs. (d)
  - Easy to layout



Small World Networks

- Milgram's Experiment and "Six degrees of separation"
  - Common across neurons, WWW, electrical power grid, …
- Add few long-distance links to a mesh randomly reduces average distance ~ logN
  - "`It's a Small World After All': NoC Performance Optimization Via Long-Range Link Insertion", VLSI 2006

Run-Time Metrics

- Hop Count
  - Latency
- Maximum Channel Load
  - Throughput
- We will consider Uniform Random Traffic