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

On-Chip Network / Network-on-Chip

Memory Controller

System/Storage Area Network
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 $\sim 3 \times$ (zero-load latency)

- **Zero load latency** (topology + routing + flow control)
- **Min latency given by routing algorithm**
- **Min latency given by topology**

- **Throughput given by flow control**
- **Throughput given by routing**
- **Throughput given by topology**

Offered Traffic (bits/sec)
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

Network Topology Graph

Vertices - cores
Edges - links

Best topology?

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

Topology is fixed at design-time.
Benefits to being regular and flexible
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
  - \((\text{Min \# channels crossing two halves}) \times \text{(BW of each channel)}\)
  - 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*
Can you suggest a regular topology (each router with same degree) with smallest possible diameter?

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

---

Application’s Task Communication Graph

Vertices - tasks
Edges - communication
Can you suggest a regular topology (each router with same degree) with **diameter = 1**?

**Application’s Task Communication Graph**

Vertices - tasks
Edges - communication

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

**Fully Connected**
Degree = ?
Bisection BW = ?

Degree = 5
Bisection BW = 9
Bus

**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

Diameter = ?
Degree = ?
Bisection BW = ?

January 11, 2017
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: On-Chip Interconnection Networks

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

Abstract

Using on-chip interconnection networks in place of all-bus 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 minimize timing variations 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%. 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 interconnections network. As shown in Figure 1, a chip employing an on-chip network is composed of a number of network clients: processors, DRAMs, 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 permits 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%) in each tile and requires 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 wiring so that their electrical properties are optimized and well-controlled. These controlled electrical parameters, in particular low and predictable crosstalk, enable the use of aggressive signaling techniques that can reduce power dissipation by a factor of ten and increase propagation velocity by three times [3]. Sharing the wiring resources among many communication paths makes more efficient use of the wires when one client is idle, other clients continue to make use of the network resources.

As on-chip interconnection network facilitates modularity by defining a standard interface in each the same manner as a backplane bus. For the past three decades systems have been composed of plug-and-play hardware. In this network, we have provided that clients are not made or distributed for profit or commercial advantage and that it bears no notice or the full citation in the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

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

- **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

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diameter?</td>
<td>(N/2)</td>
</tr>
<tr>
<td>Avg Distance?</td>
<td>(N/4)</td>
</tr>
<tr>
<td>Bisection BW?</td>
<td>2</td>
</tr>
<tr>
<td>Degree?</td>
<td>2</td>
</tr>
</tbody>
</table>
### Torus

![Torus diagram]

#### 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

**Diameter?** $\sqrt{N}$

**Bisection BW?** $2\sqrt{N}$

**Degree?** 4

#### Cons
- Unequal link lengths
- Harder to layout
Mesh

- **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)

Diameter? 2($\sqrt{N}$-1)
Bisection BW? $\sqrt{N}$
Degree? 4
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

Max = 4
Avg = 2.22

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

3-ary 2 cube
Max = 2
Avg = 1.33

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

3-ary 2 mesh
Max = 4
Avg = 1.77

9-ary 1 cube

$k$-ary $n$ cube

$k$-ary $n$ mesh
Network Latency

\[ T = H \cdot t_r + T_w + T_s + T_c \]
- \( H \): number of hops
- \( t_r \): router delay
- \( T_w \): wire delay
- \( T_s \): serialization delay
- \( T_c \): contention delay

\[ T = H \cdot t_r + \frac{D}{v} + \frac{L}{b} + T_c \]
- \( D \): 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

- 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 $\sim \log N$

- “`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