Introductions!

- **Instructor:** Tushar Krishna
  - Assistant Professor, School of ECE and CS
  - **Background**
    - PhD from MIT in EECS in Oct 2013
    - Intel (MA) from Nov 2014 – Jan 2015
  - **Research Interests**
    - Computer Architecture
      - Multi/many core, Heterogeneous, Reconfigurable
    - Communication/Interconnection Fabric
      - Cache Coherence Protocols
      - Networks-on-Chip (NoCs), HPC networks
    - CAD
      - Design automation for low-cost chip design
  - **Contact**
    - Email: tushar@ece.gatech.edu
    - Office: Klaus 2318
    - Website: http://tusharkrishna.ece.gatech.edu
What is an Interconnection Network?

- Networks *within* a system (collaboration)
- Not the Internet: network *between* systems
Why Interconnection Networks matter?

Chinese 260-core processors ShenWei SW26010 enabled supercomputer Sunway TaihuLight be the most productive in the world

IBM reveals 'brain-like' chip with 4,096 cores

Meet KiloCore, a 1,000-core processor so efficient it could run on a AA battery

This monstrous CPU is 100 times more power-efficient than today's laptops.

IBM pushes silicon photonics with on-chip optics

Big Blue researchers have figured out how to use standard manufacturing processes to make chips with built-in optical links that can transfer 25 gigabits of data per second.
What is an Interconnection Network?
What is an Interconnection Network?

- Interconnection Networks connect processors and memory elements within and across computers
What is an Interconnection Network?

- **Application:** Ideally wants low-latency, high-bandwidth, dedicated channels between processors and memory

- **Technology:** Dedicated channels too expensive in terms of area and power
What is an Interconnection Network?

- **Interconnection Network**: A programmable system that transports data between terminals over a set of shared physical channels.
Interconnection Networks

Key Design Principles

- Transfer maximum amount of information (high bandwidth) within the least amount of time (low latency) so as to not bottleneck the system
- Efficiently utilize shared but scarce resources (buffers, links, logic) to reduce area and power
Interconnection Networks can be grouped into four domains:

- Depending on the type and proximity of devices to be connected:
  - On-Chip Networks (OCNs or NoCs)
  - System/Storage Area Networks (SANs)
  - Local Area Networks (LANs)
  - Wide Area Networks (WANs)
On-Chip Network (OCN or NoC)

- Networks on multicore/MPSoc chips
  - Devices include micro architectural functional units, register files, processor/IP cores, caches, directories, memory controllers

- Current/Future Systems: tens to hundreds of devices
  - Intel Single-Chip Cloud Computer – 48 Cores
  - Tilera TILE64 – 64 Cores

- Tightly coupled with on-chip links
  - Proximity: milli meters
  - Delay: pico seconds
System/Storage area Network (SAN)

- Multi-processor and multi-computer Networks
  - Inter-processor and processor-memory interactions

- Server and Datacenter Networks
  - Storage and I/O components

- Hundreds to thousands of devices interconnected
  - IBM Blue Gene/L Supercomputer (64K nodes, each with 2 processors)

- Tightly-coupled with proprietary interconnects
  - Proximity: tens of meters (typical) to a few hundred meters
  - Link Delay: nano seconds
  - Examples: Infiniiband, Myrinet, Quadrics, Advanced Switching Interconnect
Local Area Network (LAN)

- Networks between autonomous computer systems
  - Example: Machine room or throughout a building or campus
  - “Clusters”

- Hundreds of devices
  - thousands with bridging

- Loosely coupled with commodity interconnects (e.g., Ethernet)
  - Proximity: few kilometers to few tens of kilometers
  - Delay: micro seconds
Wide Area Network (WAN)

- Networks between LANs and autonomous computer systems distributed across the world
- Millions of devices
- Loosely coupled with electrical and optical interconnects
  - Proximity: many thousands of kilometers
  - Delay: milli seconds
- Largest WAN is the Internet
Interconnection Network Domains

Source: Hennessy and Patterson, 5th Edition, Appendix F
Why study NoCs now?
Why study NoCs now?
NoCs are critical for performance

Early designs were buses and point-to-point

DOES NOT SCALE!!!
Differences between Off-Chip (SANs) and On-Chip Networks

- Significant research in multi-chassis interconnection networks (off-chip) since the 90s
  - Supercomputers
  - Clusters of Workstations
  - Internet Routers

- We can leverage research insights, but ...
  - constraints are different
  - new opportunities
Off-chip vs. On-Chip

- **Off-Chip Networks**
  - Bandwidth limited by chip pin-bandwidth
  - Latency limited by long off-chip cables

- **On-Chip Networks**
  - Very high on-chip bandwidth
    - Abundant metal layers and wiring
  - Much lower latency due to short wires
Off-chip vs. On-Chip

- The key concepts remain the same across SANs and NoCs
  - the constraints are different → the design decisions are different
    - E.g., an off-chip topology may not be feasible on-chip due to on-chip layout constraints
    - Or an on-chip link circuit may not be feasible off-chip due to technology constraints

- We will mostly focus on on-chip networks when discussing concepts, but will periodically look at implications in the off-chip (SAN/datacenter) space
Agenda

- Course Motivation
- **Course Logistics**
- Introduction to NoCs
- Simulation Infrastructure
What’s unique about this course

- Not taught as a standard course in any university
  - Sits at the intersection of Computer Architecture, Parallel and Distributed Architectures, Distributed Systems, Computer Networks, and VLSI Design

- Traditional Course Structure
  - **Computer Architecture** courses focus on processor pipeline and memory hierarchy, skimming through the system interconnection
  - **Networking** courses focus on protocols, ignoring router hardware
  - **Communications** courses focus on signal processing algorithms, skimming through hardware implementations
  - **Optics/electronic link** courses focus on link circuitry, skimming through how these links are composed as networks
What’s unique about this course

- **Sister courses**
  - **Active:** University of Toronto ECE 1749H (N Jerger)
  - **Inactive:** Past offerings in MIT (Peh), Stanford (Dally), Penn State (Das), Utah (Balasubramonian), Cornell (Batten)

- Handful of top researchers in NoCs across academia and industry
  - opportunity to gain expertise in a niche area
  - Influence manycore architectures (Intel, AMD, IBM), supercomputers (IBM, NVIDIA, Cray), datacenters (Google, Amazon, Facebook), internet routers (Cisco, Juniper) …

- projects will be open-research questions
  - **Very high chance of publication**
  - **Last Year:**
    - One to be presented at HPCA 2017 at Austin in February
    - 3 more under submission
Course Structure

- **Phase I [Jan-Feb]**
  - ~7-8 Instructor Lectures
  - 4 Programming Lab Assignments

- **Phase II [Mar-Apr]**
  - Paper Readings, Critiques, and Discussions
  - Each student presents one paper in class and leads its discussion

- **Phase III [End of Feb – Apr]**
  - Research Project
Phase I

- Lectures
  - ~7-8 weeks of instructor lectures

- Required Textbook
    - Available for free download (within Georgia Tech)
    - I will send out the link for download
    - Supplemental reading material/papers will be posted

- Optional Textbooks:
Phase I

- **Lab Assignments**
  - 4 Lab Assignments to familiarize yourself with the **Garnet2.0** NoC simulator
    - Part of gem5 (www.gem5.org), one of the leading open-source simulators for computer architecture research in industry and academia [Over 1000 citations in 4 years]
    - [http://tusharkrishna.ece.gatech.edu/garnet2.0](http://tusharkrishna.ece.gatech.edu/garnet2.0) has instructions on setting it up on the ECE/CS/local machines
      - *Wait for my announcement on T-Square before setting up gem5*
  - The 4th lab assignment can be replaced by a concrete deliverable from your project
Phase I

- **Lab 1 is due this Friday at 1 pm [Hard Deadline]!**
  - **Goal:** run synthetic traffic simulations for 4 traffic patterns and report the latency vs. injection rate results
    - I will send announcement via T-Square on what to submit
  - Setup Garnet right away and get started!
  - Gives you time to drop course if you don’t have right background
Phase II

- Writing, Presenting, and Discussing research ideas will be key thrusts

- In every class, we will discuss 2-3 papers
  - State-of-the-art research across domains
    - Datacenters, HPC, On-Chip, Circuits, Novel Technologies, Application-specific, ...
  - Everyone (including presenter) has to submit a 1 page write-up before the beginning of class on one of the papers
    - Short Summary + 2 strengths + 2 weaknesses + Suggested improvement (optional)

- One student will present on one of the papers/topics for 15-20 minutes and lead the class discussion
  - You can get the actual slides from the original presenter by emailing them, or make your own slides. You could also present on the white board if you wish.
  - The presenter should create a Piazza post for the paper when it is assigned.
  - Might have 2-3 presentations per class on different papers taking opposing views, leading to a healthy debate

- The night before the class, everyone should add one question/topic they would like to see discussed on the paper’s piazza post. Presenter should collect these and lead discussion accordingly
Phase III

- Research Project
  - Propose a solution for a research problem in the Network-on-Chip/System-Area Area Network space; implement and evaluate it
    - Implement an idea from a paper and propose an extension, or implement a completely novel idea
    - Start thinking of project ideas as I present topics in class
      - I will also periodically provide a list of potential ideas
Project Ideas

- **Microarchitecture**
  - SMART NoC emulating any Network Topology
  - NoCs for heterogeneous CPU-GPU systems
  - Approximation-aware NoC
  - Resilient NoCs
  - Hard and soft NoCs for FPGAs

- **Open-source Hardware (in RTL/Chisel)**
  - Network-on-Chip generator in Chisel
  - Connect RISC-V CPU + Harmonica GPU (Sudha Yalamanchilli’s group) with a NoC
  - Connect multiple RISC-V cores with a NoC and an open-source coherence protocol
  - Secure on-chip networks

- **Datacenter Networks**
  - Networks on a Rack
Phase III

- Research Project
  - Infrastructure
    - The project can be implemented in Garnet, or any other tool (C++/RTL)
      - Garnet/gem5 is useful since the plumbing related to other parts of the system (and even real apps) are provided.
    - We will introduce you to Chisel and Bluespec System Verilog (C-like abstractions for generating Verilog)
  - Projects related to your own Special Problem / MS / PhD research work will be encouraged as long as they have a networks component
  - Projects can be done individually (preferred) or in groups of two (if scope is larger, clearly defining each member’s role)
Phase III

- Research Project
  - Milestone 1: Project Proposals [Week 7 or 8]
    - By week 7 or 8, you will submit a project proposal to me which should be a 0.5-1 page description of what you are planning to do. You will meet me after class or during office hours to discuss it.
  - Milestone 2: Project Proposal Presentation [Week 10]
    - We will have a lightning session in class where you will present your project idea to class in 90-120 sec (1-2 slides).
  - Milestone 3: Preliminary Report [Week 13]
    - You will submit a preliminary report of your project
      - This report will not be evaluated – is for feedback for your final report
    - Your report will be assigned to 2-3 other class members who will review it and give you feedback
Phase III

- Research Project
  - **Milestone 4: Peer Report Reviews [Week 14]**
    - Each person will be assigned 2 reports of other teams to be reviewed and constructive comments/feedback given for their final reports
      - I will read through the reports and reviews, and add my comments too if necessary
  - **Milestone 5: Final Report [Week 16]**
    - The final reports are due the night before the final presentations; all reports will be compiled into a *Proceedings* and given to everyone
  - **Milestone 6: Final Project Presentation [Last Week]**
    - 10-20 minute conference style presentation
How Projects will be evaluated

- Looking for thorough understanding, implementation, evaluation, and presenting of idea
  - the idea might not lead to any improvement over the state-of-the-art
    - that is OK
    - rather that is research!

- If the idea leads to novel insights and there is a chance for publication, I am excited to work with you on polishing the work over summer and submit it to a conference or journal
  - I will fund your travel for presenting at the conference if it gets accepted 😊

Potential venues:
- WDDD: Workshop on Duplicating, Deconstructing, and Debunking
- NOCS: Network-on-Chip Symposium
- Computer Architecture Venues: ISCA, MICRO, HPCA, ASPLOS
- System Design Venues: DAC, DATE, ICCAD
- ...

- If you want to work on a novel publishable idea or an extension to your own MS/PhD research, contact me – part of it can be used for the course
## Schedule (Tentative)

<table>
<thead>
<tr>
<th>Week</th>
<th>Dates</th>
<th>Monday</th>
<th>Wednesday</th>
<th>Due [Friday]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(Jan 9 - )</td>
<td>Introduction</td>
<td></td>
<td>Lab 1</td>
</tr>
<tr>
<td>2</td>
<td>(Jan 16 - )</td>
<td>MLK Day</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(Jan 23 - )</td>
<td></td>
<td></td>
<td>Lab 2</td>
</tr>
<tr>
<td>4</td>
<td>(Jan 30 - )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(Feb 6 - )</td>
<td>HPCA 2017</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(Feb 13 - )</td>
<td></td>
<td></td>
<td>Lab 3</td>
</tr>
<tr>
<td>7</td>
<td>(Feb 20 - )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(Feb 27 - )</td>
<td></td>
<td></td>
<td>Project Proposals</td>
</tr>
<tr>
<td>9</td>
<td>(Mar 6 - )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(Mar 13 - )</td>
<td></td>
<td>Proposal</td>
<td>Lab 4</td>
</tr>
<tr>
<td>11</td>
<td>(Mar 20 - )</td>
<td>Spring Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>(Mar 27 - )</td>
<td>DATE 2017</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>(Apr 3 - )</td>
<td></td>
<td></td>
<td>Preliminary Report</td>
</tr>
<tr>
<td>14</td>
<td>(Apr 10 - )</td>
<td></td>
<td></td>
<td>Peer Reviews</td>
</tr>
<tr>
<td>15</td>
<td>(Apr 17 - )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(Apr 24 - )</td>
<td></td>
<td>Reading</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>(May 1 - )</td>
<td>Final Presentations</td>
<td>Final Re</td>
<td></td>
</tr>
</tbody>
</table>

- **GT Holiday**
- **Instructor Travel**
- **Phase I**
- **Phase II**
- **Phase III**
# Grading

<table>
<thead>
<tr>
<th>Item</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab1</td>
<td>5%</td>
</tr>
<tr>
<td>Lab2</td>
<td>10%</td>
</tr>
<tr>
<td>Lab3</td>
<td>10%</td>
</tr>
<tr>
<td>Lab4</td>
<td>10%</td>
</tr>
<tr>
<td>Paper Critiques</td>
<td>10% [Best of 10]</td>
</tr>
<tr>
<td>Paper Presentation</td>
<td>10%</td>
</tr>
<tr>
<td>Peer Reviews</td>
<td>4% [2% each]</td>
</tr>
<tr>
<td>Proposal Presentation</td>
<td>2%</td>
</tr>
<tr>
<td>Project Presentation</td>
<td>15%</td>
</tr>
<tr>
<td>Final Report</td>
<td>24%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>
What is expected from you?

- **Required Background**
  - This is an **advanced** graduate-level class
  - Graduate-level Computer Architecture (ECE 6100 / CS 6290) background expected
  - Moderate expertise in C++ Programming

- **Willingness to do research!**
  - Heavy paper reading [~15 papers]
  - Lot of Writing
    - Critiques, Reviews, and Report
  - Identify, Implement, Evaluate and Present a new idea
  - 3 presentations (Paper + Proposal + Final Report)
Self-Assessment Pre-Requisite Check

- List of terms you should already be familiar with
  - Mesh
  - Ring
  - XY Routing
  - Turn Model
  - Deadlock
  - Credit-based
  - Virtual Channel
  - Dateline
  - Arbiter
  - Crossbar
  - Wormhole
  - Private L2
  - LLC
  - Snoopy Bus
  - Directory Protocol

For each term, give yourself a self-assessment score:

2: Familiar with it
1: Heard of it and confident of looking it up
0: Never heard of it

If you have a score of less than 20, you may not have the right background. Come talk to me!
Course Information

- **Course Website**
  - [http://tusharkrishna.ece.gatech.edu/teaching/icn_s17/](http://tusharkrishna.ece.gatech.edu/teaching/icn_s17/)
  - All readings will be posted here!

- **T-Square**
  - Lecture Slides will be posted here
  - Lab Assignments will be posted and submitted here
  - Project Reports will be submitted here

- **Piazza**
  - Access via T-Square
  - Use for questions related to the lab assignments
    - Try to answer each other’s questions
  - **There are no TAs in this course**
  - Do not upload code!
  - If no one is able to answer within a day I will respond
  - Discussions on paper readings also encouraged
How to contact me

- Piazza

- Email for questions not suitable for Piazza
  - Why did I lose points?
  - Is this an acceptable project proposal

- Office Hours
  - Monday and Wednesday after class
  - Tuesdays 3-4 pm
Agenda

- Course Motivation
- Course Logistics
- Introduction to NoCs
- Simulation Infrastructure
Introduction to NoCs

On-Chip Network

Core + L1$
Core + L1$
Core + L1$
Core + L1$
Core + L1$
Core + L1$

L2$
L2$
L2$
L2$
L2$
L2$

Role of the Network-on-Chip

- “Shared Memory” Systems
  - Transport Cache Lines and Cache Coherence Messages between caches and memory controller(s) in shared memory CMPs

- “Message Passing” Systems
  - Transfer data between IP blocks in MPSoCs (Multi-Processor System on Chip)
Example of Traffic Flows

On-Chip Network

LD (A) R1

Core + L1$

Core + L1$

Core + L1$

Core + L1$

Core + L1$

Core + L1$

Req

Fwd

Rsp

L2$

L2$

L2$

L2$

L2$

L2$

If network delay = 10 cycles, controller delay = 5 cycles, how many cycles before data arrives?
Modern NoCs

Core will not be shown explicitly in the most of the slides. Only the routers will be.
Network Architecture

- Topology
- Routing
- Flow Control
- Router Microarchitecture
Topology:
How to connect the nodes with links

~Road Network
Many potential topologies

Topology often constrained by area and packaging
Routing: Which path should a message take

~Series of road segments from source to destination
Flow Control:
When does the message stop/proceed

~Traffic Signals / Stop signs at end of each road
Router Microarchitecture: How to build the routers

~Design of traffic intersection (number of lanes, algorithm for turning red/green)
Examples

Oracle SPARC T5 (2013)
16 multi-threaded cores and 8 L2 banks connected by a Crossbar NoC

IBM Cell (2005)
1 general purpose and 8 special purpose engines connected by a Ring NoC

Intel SCC (2009)
24 tiles with 2 cores each connected by a Mesh NoC
What makes network design challenging (and interesting)

- Network resources are distributed with almost no centralized control
- Traffic is (often) unpredictable
  - Why?
    - Mapping of tasks to core
    - Memory layout of data
    - Cache sizes, policies
    - Data sharing
    - ...
- Surprisingly easy for the network to become the bottleneck
NoC Metrics

- Performance
  - Latency
  - Bandwidth

- Power
  - Energy Consumption in Links
  - Energy Consumption in Routers

- Area
How to evaluate a NoC?

Latency

Offered Traffic (bits/sec)
Agenda

- Course Motivation
- Course Logistics
- Introduction to NoCs
- Simulation Infrastructure
The gem5 full-system simulator

http://www.gem5.org  Join the mailing list!

http://tusharkrishna.ece.gatech.edu/teaching/garnet_gt/
has instructions on setup for this class

<table>
<thead>
<tr>
<th>Workload</th>
<th>Simple (Fast)</th>
<th>Detailed (Slow)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>System Emulation</td>
<td>Full-System</td>
</tr>
<tr>
<td>ISA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>AtomicSimple TimingSimple</td>
<td>InOrder OutOfOrder</td>
</tr>
<tr>
<td>L1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2+Dir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2+Dir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2+Dir</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network-on-Chip</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Memory System

- Classic
- Ruby
  * Caches
  * Coherence protocols

NoC

- Simple
- Garnet2.0
Garnet in standalone mode

User can inject any traffic pattern

**Sources:** CPUs

**Destinations:** Directories

Traffic Pattern determines which CPU sends to which Directory.

**Injection Rate is user specified**

Directories consume any packet that is received.

---

### Source (binary coordinates):

\[
(y_{k-1}, y_{k-2}, \ldots, y_1, y_0, x_k-1, x_{k-2}, \ldots, x_1, x_0)
\]

<table>
<thead>
<tr>
<th>Traffic Pattern</th>
<th>Destination (binary coordinates)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit-Complement</td>
<td>((\bar{y}<em>{k-1}, \bar{y}</em>{k-2}, \ldots, \bar{y}<em>1, \bar{y}<em>0, \bar{x}</em>{k-1}, \bar{x}</em>{k-2}, \ldots, \bar{x}_1, \bar{x}_0))</td>
</tr>
<tr>
<td>Bit-Reverse</td>
<td>((x_0, x_1, \ldots, x_{k-2}, x_{k-1}, y_0, y_1, \ldots, y_{k-2}, y_{k-1}))</td>
</tr>
<tr>
<td>Shuffle</td>
<td>((y_{k-2}, y_{k-3}, \ldots, y_0, x_{k-1}, x_{k-2}, x_{k-3}, \ldots, x_0, y_{k-1}))</td>
</tr>
<tr>
<td>Tornado</td>
<td>((y_{k-1}, y_{k-2}, \ldots, y_1, y_0, x_{k-1} + \left[\frac{k}{n}\right] - 1, \ldots, x_k - \left[\frac{k}{n}\right] - 1))</td>
</tr>
<tr>
<td>Transpose</td>
<td>((x_{k-1}, x_{k-2}, \ldots, x_1, x_0, y_{k-1}, y_{k-2}, \ldots, y_1, y_0))</td>
</tr>
<tr>
<td>Uniform Random</td>
<td>(\text{random}())</td>
</tr>
</tbody>
</table>
Example

```bash
./build/Garnet_standalone/gem5.debug \configs/example/garnet_synth_traffic.py \--num-cpus=16 \--num-dirs=16 \--network=garnet2.0 \--topology=Mesh_XY \--mesh-rows=4 \--sim-cycles=1000 \--injectionrate=0.01 \--synthetic=uniform_random
```
That’s all for today!

If you have random traffic, how many hops does each message take on average on this 4x4 mesh topology?

Can you generalize this to a $k \times k$ mesh?