18-447 Lecture 26: Interconnects

James C. Hoe
Department of ECE
Carnegie Mellon University
Housekeeping

• Your goal today
  – get an overview of parallel processing interconnect topics—whether it is on-a-chip or around-the-world

• Notices
  – HW 5 past due, Lab 4 due Friday 5/1
  – Midterm 3, Thursday, 5/7, 5:30pm~6:25pm

• Readings
  – P&H Ch 6
  – *The CONNECT Network-on-Chip Generator, 2015* (optional)
Connecting Things “Systematically”
Broadcast Bus

- Simple and cheap
- Everyone sees everyone else’s transactions (good for ordering and cache coherence)
- But
  - bandwidth cannot scale with system size, \( N \)
  - latency suffer terribly under load
  - electrically challenging as speed and \( N \) grow

Physical extent by itself is not necessarily an issue, e.g., IEEE 802.3 CSMA/CD and ALOHAnet
Other Extreme: All-to-All Point-to-Point

- ideal case when cost no object
- not scalable in cost: # of links and # of connections per node
Crossbar Switch

• Concurrent sends to non-conflicting destinations

• Still expensive to scale, $O(N)$ wires but $O(N^2)$ $X$s
Multistage Circuit Switched

- More restrictions on concurrent Tx-Rx pairs
- More scalable, e.g., \(O(N \log N)\) cost for Butterfly

2-by-2 crossbar
Packet Switched

- Packetized send and forget operation
- Packets “hop” from router to router, pending availability of the next-required switch and buffer

2-by-2 router
From a Distance: Performance Characteristics
A network is a network: 

functionally I just want to send from A to B

bandwidth, latency, scalability, “features”
Bandwidth

Bisection Bandwidth
• which cut?
• best vs. worst

Endpoint Bandwidth:
• to whom?
• 1-to-1, 1-to-many
• who else is sending?

Aggregate Bandwidth
• which traffic pattern?
Latency

End-to-End Latency
- between whom?
- average/best/worst case
- who else is sending?

Latency Measures
- diameter
- hops
- cycle or sec (includes buffer and contention delays)
Test Traffic Patterns

• Ideally, know the traffic and perf. requirement
• If not, resort to “test traffic patterns”
  – capture average, best, worst case scenarios
  – stress and highlight hotspots and weaknesses
  – like “benchmarks” for CPUs
• Random: non/uniform, {all-to-all, 1-to-all, all-to-1}
• Bit permutations
  – each source has 1 destination
  – dest ID is a bit permutation of source ID
  – e.g. transpose, shuffle, complement, reverse, ...
• Other synthetic: tornado, nearest neighbor, ...
• Playback of real/synthetic workload traces
Load-Delay Curve

Desiderata
- high saturation throughput
- low latency at low load
- well behaved approaching saturation throughput

latency

zero-load latency

offered load

saturation throughput
A Little Closer Now:
Different Topologies to Meet Different Requirements
Unidirectional Ring

- Simplest topology and implementation
  - $O(N)$ cost
  - $O(1)$ worst-case bisection BW (left-right halves), but $O(N)$ best-case bisection BW (odd-even halves)
  - $N/2$ average hops; latency depends on utilization

Simplicity allows very high-freq router and link
1D Mesh

- Bi-directional links; travel left or right to go from src to dest; $N/3$ average hops
- “Torus” wraps around nodes 0 and ($N-1$) for $N/4$ avg hops; physically interleaved to avoid long links
• 2D layout scales easily as system-area network or network-on-chip; $O(N^{0.5})$ bisection bandwidth
• Dimensional routing: first route to col in fewest hops then route in 2nd dimension
• Generalizable to higher dimensional mesh networks
Higher Dimensional Topologies: e.g., Butterfly & Hypercube

- Fewer hops; higher bisection bandwidth
- Hard to physically place wires in high dimensions
- Hypercube switch complexity grows as $\log(N)$
Fat Tree

- Like a tree, $2\log(n)$ hops for a neighborhood of $n$ nodes; $2\log(N)$ worst-case hops across a system.
- Unlike a simple tree, fat-tree adds an alternate up-route at each router at each level: $O(N)$ bisection BW.
- Random-up, deterministic-down routing.

$\frac{N}{2}$ links at each level.
Of all things, why a lowly ring?

Traffic, Scale & Cost Dictates

Datacenter hw acceleration plane

Traditional sw (CPU) server plane

Microsoft Catapult [MICRO 2016, Caulfield, et al.]
Up Close and Personal: Packets and Routers
Network Packets

CM-5 Packets

<table>
<thead>
<tr>
<th>route</th>
<th>T</th>
<th>L</th>
<th>data (4-20 bytes)</th>
<th>sum</th>
</tr>
</thead>
</table>

Ethernet Packets

<table>
<thead>
<tr>
<th>preamble</th>
<th>dest.</th>
<th>source</th>
<th>length</th>
<th>data (0-1500 bytes)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>pad (0-46)</th>
<th>checksum</th>
</tr>
</thead>
</table>

- **Header**
  - dest ID or route bits
  - src ID, priority, packet type, etc.
- **Data payload**
  - large vs. small
  - fixed vs. variable

- **Checksum**
  - redundancy coding (e.g., CRC)
  - most cases only for detection not correction
A Basic Router

- Packet enters on an Rx-link and choose a Tx-link to exit
  - route table maps dest-ID to Tx-link; OR
  - a fixed fxn of dest-ID or route-bits; OR
  - adaptive for congestion or fault
- Packets wait in buffer until
  - next router has buffer space; AND
  - Tx-link/crossbar is free
Packets vs. Flits

- A “packet” is made up of 1 or more fixed-size “flits”
  - route packets
  - flow-control flits
- Credit-based flow control
  - Tx logic hold credits for downstream Rx buffer
  - Tx logic deduct 1 credit when sending 1 flit; stop when out of credit
  - Rx logic return a credit token when a flit advances out of its buf
Virtual Networks

- Time-multiplex same physical links over multiple sets of packet buffers
- Effectively multiple independent networks
  - to provide different priority packet classes
  - to get around blockage
  - to avoid deadlocks