Interconnection Networks

Reading

- Appendix E
Goals

- Coverage of basic concepts for high performance multiprocessor interconnection networks
  - Primarily link & data layer communication protocols
- Understand established and emerging micro-architecture concepts and implementations

Technology Trends

Source: W. J. Dally, "Enabling Technology for On-Chip Networks," NOCS-1, May 2007
Where is the Demand?

- Cables, connectors, transceivers
- Throughput
- Area, power
- Performance
- Cost
- Latency, power

Blurring the Boundary

- Use heterogeneous multi-core chips for embedded devices
  - IBM Cell → gaming
  - Intel IXP → network processors
  - Graphics processors → NVIDIA

- Use large numbers of multicore processors to build supercomputers
  - IBM Cell
  - Intel Tera-ops processor

- Interconnection networks are central all across the spectrum!
Cray XT3

- 3D Torus interconnect
- HyperTransport + Proprietary link/switch technology
- 45.6GB/s switching capacity per switch

Blue Gene/L

Five Steps to a Petaflop Computer

1. Chip
2. Processor
3. Board
4. Tower
5. 1 Petaflop

1 Gigaflap
32 Gigaflops
2 Teraflops
16 Teraflops
1 Petaflop
Blue Gene

From http://i.n.com.com/i/ne/p/photo/BlueGeneL_03_550x366.jpg

Intel TeraOp Die

- 2D Mesh
- Really a test chip
- Aggressive speed – multi-GHz links?
On-Chip Networks

- Why are they different?
  - Abundant bandwidth
  - Power
  - Wire length distribution

- Different functions
  - Operand networks
  - Cache memory
  - Message passing
Some Industry Standards

- On-Chip
  - Open Core Protocol
    - Really an interface standard
  - AXI
- PCI Express
- AMD HyperTransport (HT)
- Intel QuickPath Interconnect (QPI)
- Infiniband
- Ethernet

Basic Concepts

- Link Level
- Switch level
- Topology
- Routing
- End-to-End latency
Link Level Concepts

Messaging Hierarchy

- Where?: Destination decisions, i.e., which output port
- When?: When is data forwarded
- How?: Synchronization of data transfer

- Lossless transmission (unlike Ethernet)
- Performance rather than interoperability is key
**System View**

![System View Diagram]

From http://www.psc.edu/publications/tech_reports/PDIO/CrayXT3-ScalableArchitecture.jpg

---

**Messaging Units**

- Data is transmitted based on a hierarchical data structuring mechanism
  - Messages $\rightarrow$ packets $\rightarrow$ flits $\rightarrow$ phits
  - While flits and phits are fixed size, packets and data may be variable sized

---

ECE 4100/6100 (17)
Link Level Flow Control

- Unit of synchronized communication
  - Smallest unit whose transfer is requested by the sender and acknowledged by the receiver
  - No restriction on the relative timing of control vs. data transfers

- Flow control occurs at two levels
  - Level of buffer management (flits/packets)
  - Level of physical transfers (phits)
  - Relationship between flits and phits is machine & technology specific

- Briefly: Bufferless flow control

---

Physical Channel Flow Control

Asynchronous Flow Control
- What is the limiting factor on link throughput?

Synchronous Flow Control
- How is buffer space availability indicated?
Flow Control Mechanisms

- Credit Based flow control
- On/off flow control
- Optimistic Flow control (also Ack/Nack)
- Virtual Channel Flow Control

Message Flow Control

- Basic Network Structure and Functions
  - Credit-based flow control

Sender sends packets whenever credit counter is not zero.

Queue is not serviced.
Message Flow Control

- Basic Network Structure and Functions
  - Credit-based flow control

Sender resumes injection
Receiver sends credits after they become available

Credit counter

Queue is not serviced

Timeline*

Round trip credit time equivalently expressed in number of flow control buffer units - $t_{rt}$

*From W. J. Dally & B. Towles, "Principles and Practices of Interconnection Networks," Morgan Kaufmann, 2004
Performance of Credit Based Schemes

- The control bandwidth can be reduced by submitting block credits

- Buffers must be sized to maximize link utilization
  - Large enough to host packets in transit

\[
F \geq \frac{t_{rt} \times b}{L_f}
\]

*From W. J. Dally & B. Towles, "Principles and Practices of Interconnection Networks," Morgan Kaufmann, 2004

Optimistic Flow Control

- Transmit flits when available
  - De-allocate when reception is acknowledged
  - Re-transmit if flit is dropped (and negative ack received)

- Issues
  - Inefficient buffer usage
    - Messages held at source
    - Re-ordering may be required due to out of order reception

- Also known as Ack/Nack flow control
Virtual Channel Flow Control

- Physical channels are idle when messages block
- Virtual channels decouple physical channel from the message
- Flow control (buffer allocation) is between corresponding virtual channels

Virtual Channels

- Each virtual channel is a pair of unidirectional channels
  - Independently managed buffers multiplexed over the physical channel
- Improves performance through reduction of blocking delay
- Important in realizing deadlock freedom (later)
As the number of virtual channels increase, the increased channel multiplexing has multiple effects (more later)
- Overall performance
- Router complexity and critical path

Flits must now record VC information
- Or send VC information out of band

Flow control parameters are tuned based on link length, link width and processing overhead at the end-points
- Effective FC and buffer management is necessary for high link utilizations → network throughput
  - In-band vs. out of band flow control
- Links maybe non-uniform, e.g., lengths/widths on chips
  - Buffer sizing for long links
- Latency: overlapping FC, buffer management and switching → impacts end-to-end latency
- Some research issues
  - Reliability
  - QoS
  - Congestion management
Switching Techniques

- Determines “when” message units are moved

- Relationship with flow control has a major impact on performance
  - For example, overlapping switching and flow control
  - Overlapping routing and flow control

- Tightly coupled with buffer management
Circuit Switching

- Hardware path setup by a *routing header* or *probe*
- End-to-end acknowledgment initiates transfer at full hardware bandwidth
- System is limited by signaling rate along the circuits
- Routing, arbitration and switching overhead experienced once/message

---

Circuit Switching

- **Switching**
  - Circuit switching

Buffers for "request" tokens
Routing, Arbitration, and Switching

- Switching
  - Circuit switching

Buffers for “request” tokens

Request for circuit establishment
(routing and arbitration is performed during this step)

Buffers for “ack” tokens

Acknowledgment and circuit establishment
(as token travels back to the source, connections are established)
Switching

- Circuit switching

Source end node

Destination end node

Packet transport
(neither routing nor arbitration is required)

High contention, low utilization ($\rho$) $\rightarrow$ low throughput
Packet Switching (Store & Forward)

- Finer grained sharing of the link bandwidth
- Routing, arbitration, switching overheads experienced for each packet
- Increased storage requirements at the nodes
- Packetization and in-order delivery requirements
- Alternative buffering schemes
  - Use of local processor memory
  - Central (to the switch) queues

Routing, Arbitration, and Switching

- Switching
  - Store-and-forward switching

Buffers for data packets

Packets are completely stored before any portion is forwarded
### Switching

- **Store-and-forward switching**

![Diagram of Store-and-forward switching](image)

- **Requirement:** Buffers must be sized to hold entire packet (MTU)

- **Source end node**
- **Forward**
- **Destination end node**

Packets are completely stored before any portion is forwarded.

---

### Virtual Cut-Through

- Messages cut-through to the next router when feasible
- In the absence of blocking, messages are pipelined
  - Pipeline cycle time is the larger of intra-router and inter-router flow control delays
- When the header is blocked, the complete message is buffered at a switch
- High load behavior approaches that of packet switching

![Diagram of Virtual Cut-Through](image)
Routing, Arbitration, and Switching

- Switching
  - Cut-through switching

Portions of a packet may be forwarded ("cut-through") to the next switch before the entire packet is stored at the current switch.

Wormhole Switching

- Messages are pipelined, but buffer space is on the order of a few flits
- Small buffers + message pipelining $\rightarrow$ small compact switches/routers
- Supports variable sized messages
- Messages cannot be interleaved over a channel: routing information is only associated with the header
- Base Latency is equivalent to that of virtual cut-through
Routing, Arbitration, and Switching

- **Switching**
  - Virtual cut-through
    - Buffers for data packets
      - Requirement: buffers must be sized to hold entire packet (MTU)
    - Source end node
    - Destination end node
    - Buffers for flits: packets can be larger than buffers

- **Wormhole**
  - Packet stored along the path
  - Packet completely stored at the switch


© T.M. Pinkston, J. Duato, with major contributions by J. Filch
Comparison of Switching Techniques

• Packet switching and virtual cut-through
  - Consume network bandwidth proportional to network load
  - Predictable demands
  - VCT behaves like wormhole at low loads and like packet switching at high loads
  - Link level error control for packet switching

• Wormhole switching
  - Provides low (unloaded) latency
  - Lower saturation point
  - Higher variance of message latency than packet or VCT switching

Topologies
The Network Model

Metrics (for now): latency and bandwidth

Routing, switching, flow control, error control

Pipelined Switch Microarchitecture

Classification

- Shared medium networks
  - Example: backplane buses

- Direct networks
  - Example: $k$-ary $n$-cubes, meshes, and trees

- Indirect networks
  - Example: multistage interconnection networks

Direct Networks

- Buses do not scale, electrically or in bandwidth
- Full connectivity too expensive (not the same as Xbars)
- Network built on point-to-point transfers
System View

Cray XT3 Scalable Architecture

Processor
Memroy
Router

Injection channels
Injection channels

Performance critical

From http://www.psc.edu/publications/tech_reports/PDIO/CrayXT3-ScalableArchitecture.jpg

Processor

High latency region

PCIe

Common Topologies

Binary n-cube

Torus (3-ary 2-cube)

n-dimensional mesh

ECE 4100/6100  (53)

ECE 4100/6100  (54)
### Metrics

<table>
<thead>
<tr>
<th>Network</th>
<th>Bisection Width</th>
<th>Node Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-ary $n$-cube</td>
<td>$2Wk^{n-1}$</td>
<td>$2Wn$</td>
</tr>
<tr>
<td>Binary $n$-cube</td>
<td>$NW/2$</td>
<td>$nW$</td>
</tr>
<tr>
<td>$n$-dimensional mesh</td>
<td>$Wk^{n-1}$</td>
<td>$2Wn$</td>
</tr>
</tbody>
</table>

### Blocking vs. Non-blocking Networks

- Consider the permutation behavior
  - Model the input-output requests as permutations of the source addresses
Crossbar Network

Non-Blocking Clos Network

© T.M. Pinkston, J. Duato, with major contributions by J. Fitch
Clos Network Properties

- General 3 stage non-blocking network
  - Originally conceived for telephone networks

- Recursive decomposition
  - Produces the Benes network with 2x2 switches
Clos Network: Recursive Decomposition

16 port, 7 stage Clos network = Benes topology

Path Diversity

Alternative paths from 0 to 1. 16 port, 7 stage Clos network = Benes topology
Alternative paths from 4 to 0. 16 port, 7 stage Clos network = Benes topology

Contention free, paths 0 to 1 and 4 to 1. 16 port, 7 stage Clos network = Benes topology
Moving to Fat Trees

- Nodes at tree leaves
- Switches at tree vertices
- Total link bandwidth is constant across all tree levels, with full bisection bandwidth
- Equivalent to folded Benes topology
- Preferred topology in many system area networks

Folded Clos = Folded Benes = Fat tree network

Fat Trees: Another View

- Equivalent to the preceding multistage implementation
- Common topology in many supercomputer installations
Relevance of Fat Trees

- Popular interconnect for commodity supercomputing
- Active research problems
  - Efficient, low latency routing
  - Fault tolerant routing
  - Scalable protocols (coherence)
Generic Router Architecture

- Input queues (virtual channels)
- Output queues (virtual channels)
- Switch
- Address decoder
- From/to local processor
- Physical input channels
- Physical output channels

Routing Algorithms

- Deterministic Routing
- Oblivious Routing
- Source Routing
- Table-based routing
- Adaptive Routing
- Unicast vs. multicast
Deterministic Routing Algorithms

- Strictly increasing or decreasing order of dimension
- Acyclic channel dependencies

Deadlock
Overview: The Problem

- **Guaranteeing message delivery**
  - Fundamentally an issue of finite resources (buffers) and the manner in which they are allocated
    - How are distributed structural hazards handled

- Waiting for resources can lead to
  - Deadlock: configuration of messages that cannot make progress
  - Livelock: messages that never reach the destination
  - Starvation: messages that never received requested resources
    - Even though the requested resource does become available

Deadlocked Configuration of Messages

- Routing in a 2D mesh
- What can we infer from this figure?
  - Routing
  - Wait-for relationships
Example of Deadlock Freedom

- Strictly increasing or decreasing order of dimension
- Acyclic channel dependencies

Channel Dependencies

- When a packet holds a channel and requests another channel, there is a direct dependency between them
- Channel dependency graph $D = G(C,E)$
- For deterministic routing: single dependency at each node
- For adaptive routing: all requested channels produce dependencies, and dependency graph may contain cycles
• The configuration to the left can deadlock
• Add (virtual) channels
  - We can make the channel dependency graph acyclic via routing restrictions (via the routing function)
• Routing function is $c_{0i}$ when $j<i$, $c_{1i}$ when $j>i$

• Channels $c_{00}$ and $c_{13}$ are unused
• Routing function breaks cycles in the channel dependency graph
Summary

• Multiprocessor and multicore interconnection networks are different from traditional inter-processor communication networks
  ❖ Unique microarchitectures
  ❖ Distinct concepts

• Network-on-chip critical to multicore architectures

Glossary

• Adaptive routing
• Arbitration
• Bisection bandwidth
• Blocking networks
• Channel dependency
• Circuit switching
• Clos networks
• Credit-based flow control
• Crossbar network
• Deadlock and deadlock freedom
• Deterministic routing
• Direct networks
• Fat trees
• Flit
• Flow control
• Hypercubes
• Indirect networks
• K-ary n-cubes

• Oblivious routing
• Multistage networks
• Non-blocking networks
• Packet
• Packet switching
• Phit
• Routing
• Routing header
• Routing payload
• Router computation
• Switch allocation
• Virtual channels
• Virtual channel allocation (in a switch)
• Virtual channel flow control
• Virtual cut-through switching
• Wormhole switching