Interconnection Networks II: Topology, Routing, and Flow Control
Recap: Last time we covered some basics on interconnection networks.

Today: We will talk about topologies, routing, and flow-control in more depth.

Next week: Read and discuss research papers.

Today's Reading Assignment: 10.4 10.8
Next week: Papers to be announced.

Advanced Computer Architecture II: Multiprocessor Design

Interconnection Networks II: Topology, Routing, and Flow Control

Professor Russ Joseph
Department of Electrical and Computer Engineering
Northwestern University

February 22, 2005

Administrative Stuff

Announcements
No class on Thursday.
We will meet again next Tuesday.
If I didn't see you last week, please drop by my office this week.
Exam 2 will be on Thursday, 3/3/05

Projects
Let me know if there are problems,
Don't forget weekly progress reports,
Contact me if you have problems with SimOS,
If you are ready to give up on SimOS, try Simics (http://www.simics.net).

Outline

Review of ICN Basics
Standard Topologies
Routing Issues
Flow-control strategies
Review: Interconnection Networks Basics

Networks can be categorized by these basic criteria:

- topology – where the network elements are connected
- routing algorithm – what path messages take through the system
- switching strategy – how message advances along its path
- flow control – when the message (or portions of it) move along the path

Fully Connected

Structure: All inputs connected to all outputs,

What's an example of a fully connected network? Hint: We have studied it extensively this quarter.

What are the problems with fully connected networks?

Topology

You can think of network as a graph \( G = (V,E) \).

Basic network structure has obvious impact on several important issues:

- Performance (latency as well as bandwidth)
- Cost (in terms of $$$ and complexity)
- Fault-Tolerance

We will keep these in mind as we look at the following topological structures.

Linear Array and Rings

Structure: Input of one node connected to output of previous node in a chained fashion.
**Meshes and Tori**

Structure: Each node has a d-vector coordinate \((i_1, ..., i_d)\) where \(0 \leq i_j \leq k_j\). Nodes which differ in precisely one coordinate by one share a link.

Mesh: No wrap around. What is node degree?
Tori: Wrap around applies. What is node degree?

![2D Grid](image1)
![3D Cube](image2)

**Butterflies**

Structure: Well, I'm not going to describe it in words, so pay attention to the pictures. Basically, it's like a tree with many roots.

![16 node butterfly](image3)
![Building block](image4)

**Trees**

Structure: Each node has 0-k children which are linked.
How does this scale?
What are the fault tolerance issues with this approach?

![Tree](image5)

**Hypercubes**

Structure: A d-dimensional binary torus.
Very important historically.

![Hypercube](image6)
Routing
Routing determines the path that communications take through the network.

Usually, the routing algorithm restricts the absolute number of paths presented by topology to a subset of possible paths.

Key concerns include performance (obviously), complexity of the algorithm, deadlock avoidance, and fault-tolerance.

Routing Mechanisms
There are pros/cons to each of the following common routing approaches:

Arithmetic – Decisions made on simple comparisons and calculations based on packet header,

<table>
<thead>
<tr>
<th>Direction</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>West (-x)</td>
<td>( \Delta x &lt; 0 )</td>
</tr>
<tr>
<td>East (+x)</td>
<td>( \Delta x &gt; 0 )</td>
</tr>
<tr>
<td>South (-y)</td>
<td>( \Delta x = 0, \Delta y &lt; 0 )</td>
</tr>
<tr>
<td>North (+y)</td>
<td>( \Delta x = 0, \Delta y &gt; 0 )</td>
</tr>
<tr>
<td>Stop</td>
<td>( \Delta x = 0, \Delta y = 0 )</td>
</tr>
</tbody>
</table>

Source-based – Source determines and specifies output ports to be used and embeds in package header,

Table-lookup – Each switch has routing table which is indexed using header info.

Deterministic and Adaptive Routing
Routing algorithm is deterministic if the final route is based entirely on source/destination regardless of other traffic.

Conversely, adaptive algorithms are influenced by network traffic.

Deadlock Freedom
We spoke before about protocol level deadlock, now we address this issue at the network level.

Channels and buffers are resources that messages must share fairly and productively.
Review: Store-and-Forward vs. Cut-Through Routing
For packet switching, two main choices are store-and-forward and cut-through routing.

Virtual Channels
Common deadlock free approach for wormhole routing is use of virtual channel.
Virtual Channel
- Provide multiple buffers for each physical channel.
- Break dependence by assigning packets to different virtual channels based on source/destination.

Virtual Cut-Through and Wormhole Routing
Under cut-through packet switching, what happens when a packet is blocked?

Virtual cut-through
- Spool the blocked incoming packet into buffer.
- In worst case, behaves like store-and-forward.

Wormhole
- Only buffer a few flits of packet.
- Leave remainder on the network.

Turn-Model Routing
Restrict the allowed routes by specifying when turns may be made.
Can break cycles in dependence graph.

There are many possible ways to apply turn-model routing.
Adaptive Routing
Clearly, adaptation is necessary for fault-tolerance.
Also useful for improving link utilization and avoiding congestion.
However, adaptive routing adds complexity to the switch.

Link-Level Control
Destination provides feedback to source indicating when it is ready to receive (e.g., REQ/ACK).
There are many ways this can be done.
Today we look at two alternatives:
Banking/Credits
Low/High Water Marks

Flow Control
Flow control governs when messages (or packets) move through links in the network.

Basic Goals:
Deal with link/buffer contention fairly.
Do not let input/output buffers overflow at switches.

End-to-End Control
There are important macro problems that even the best link-level control cannot address (e.g., back-pressure and hotspots).

At the source, we can introduce some control to reduce the delay due to congestion/contention.

At a high-level, these mechanisms introduce gaps in traffic flow which decouple processor-to-processor interaction.

Examples:
Node must wait to send N bytes of data until it has received N bytes of data.
Node waits for chunks of its data to be acknowledged before sending more.
Summary

Interconnection network design includes careful selection of structure and policies:

- Choice of topology has very broad implications on performance, cost, and reliability of network.
- Routing algorithms affect both performance and correctness.
- Flow-control governs hops between nodes usually at link-level, but also possibly via end-to-end strategies.

Next Time: Readings from research papers. Check webpage.