EE382C
Lecture 9

Deadlock
4/26/11
Question of the day

• What is the minimum number of virtual channels needed to break deadlock in a mesh network using virtual-channel flow control with minimal adaptive routing?

• Give two approaches to allocating these VCs.
Deadlock Review

Deadlock

- A condition in which a set of agents waits indefinitely trying to acquire a set of resources

Deadlocked Configuration

- A configuration in which the wait-for/held-by graph between agents and resources has at least one cycle
- i.e., the configuration is deadlocked

Resource Dependence

- A resource A is dependent on a resource B if it is possible for A to be held-by an agent X and it is also possible for X to wait-for B
- This relationship is transitive
- If a resource dependence graph is acyclic, a deadlocked configuration cannot be reached
Example Zax

One day, making tracks In the prairie of Prax, Came a North-Going Zax And a South-Going Zax.

And it happened that both of them came to a place Where they bumped. 'There they stood. Foot to foot. Face to face.

"Look here, now!" the North-Going Zax said, "I say! You are blocking my path. You are right in my way. I'm a North-Going Zax and I always go north. Get out of my way, now, and let me go forth!"

"Who's in whose way?" snapped the South-Going Zax. "I always go south, making south-going tracks. So you're in MY way! And I ask you to move And let me go south in my south-going groove."

Then the North-Going Zax puffed his chest up with pride. "I never," he said, "take a step to one side. And I'll prove to you that I won't change my ways If I have to keep standing here fifty-nine days!"

"And I'll prove to YOU," yelled the South-Going Zax, "That I can stand here in the prairie of Prax For fifty-nine years! For I live by a rule That I learned as a boy back in South-Going School. Never budge! That's my rule. Never budge in the least! Not an inch to the west! Not an inch to the east! I'll stay here, not budging! I can and I will If it makes you and me and the whole world stand still!"

Well... Of course the world didn't stand still. The world grew. In a couple of years, the new highway came through And they built it right over those two stubborn Zax And left them there, standing un-budge in their tracks.
Zax Continued
Example

- Circuit-switched ring network
Example

• Consider the same 4-node ring network with virtual channel flow control
  – What is the dependence graph?
  – How can you avoid deadlock in this network?
Protocol Deadlock

network

wait for

node

request

server

reply
Dealing with Deadlock

- Two main approaches
  - Just say NO! (to deadlock)
    - AKA Deadlock Avoidance
    - Prevent a deadlock from ever occurring
      - e.g., by resource ordering
  - Fix it
    - AKA Deadlock Recovery
    - Detect a deadlock and correct it
      - e.g., by dropping packets or using “escape paths”
Resource Ordering

• Example – packet buffers in a ring
  – Nodes of dependence graph are buffers (on nodes)
  – Order by
    • Distance remaining
    • Dateline

• Example – virtual channels in a ring
  – Nodes of dependence graph are virtual channels, not router nodes
Order Buffers by Distance Remaining
Order Buffers by Dateline
Order Virtual Channels by Dateline
Last Thursday’s QOTD

• What is the minimum number of virtual channels needed to break deadlock in a mesh network using virtual-channel flow control?

• A – it depends on your routing algorithm
  – For DOR, you can get by with a single VC per PC
Ordering Channels
Dimension-order routing in a mesh
The Turn Model
(Glass and Ni ISCA '92)

• All cycles in a 2-D mesh require four turns
• Disallow one of these turns from the clockwise set
• And a “compatible” turn from the counterclockwise set
• This avoids deadlock
• At the expense of restricting routing
Turn Model

(a) West-first

(b) North-last

(c) Negative-first

(b) disallowed
Why is the last pattern disallowed?

(a)  

(b)  

(c)  

West-first  North-last  Negative-first  disallowed
It creates a cycle across the two sets.
Ordering Implied by West-First
Deadlock and Adaptive Routing

• With adaptive routing there are *alternatives*
• Can have cycles in the channel dependence graph as long as there are *escape paths* from each cycle
Example

• Consider a mesh with two VCs per node
• VC 0 is restricted to DOR (provides *escape path*)
• VC 1 can use arbitrary minimal routing

• This has cycles, but is deadlock free

• What if it were non-minimal?
Indirect Dependence
With non-minimal routing allowed

- Packets can traverse channels they already traversed
  - Therefore, two packets can wait on each other
- Only a concern with flit-buffer flow control. Why?

Class 0 is the DOR class
Duato’s Theorem

- Divide routing relation into $R_1$ and $R_c = R - R_1$
  - Routing subrelation and complement subrelation
- $R$ is deadlock free iff $R_1$ has no cycles in its *extended* channel dependency graph
- Indirect dependence from $c_a$ to $c_b$ (in $R_1$) if there exists a path from $c_a$ to $c_b$ over channels entirely in $R_c$
Basic Router Architecture

Diagram showing the basic architecture of a router with the following components:
- Router
- VC Allocator
- Switch Allocator
- Input Unit
- Output Unit
- Switch
Router Pipeline

<table>
<thead>
<tr>
<th>Cycle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head Flit</td>
<td>RC</td>
<td>VA</td>
<td>SA</td>
<td>ST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Body Flit 1</td>
<td></td>
<td>SA</td>
<td>ST</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Body Flit 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tail Flit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question of the day

- What is the minimum number of virtual channels needed to break deadlock in a mesh network using virtual-channel flow control with minimal adaptive routing?

- Give two approaches to allocating these VCs.
Summary

• Deadlock
  – Situation when agents wait indefinitely for resources
  – Occurs when there is a cycle in the wait-for/held-by graph
  – Cannot occur if resource dependence graph is acyclic

• Deadlock Avoidance
  – Typically achieved by resource (buffers or VCs) ordering
  – Accomplished by restricting resource allocation
    • Buffer allocation
    • VC allocation
    • Routing
  – Resource ordering
    • Distance based
    • Dateline

• Deadlock and Adaptive Routing
  – Escape channels, routing subrelation
  – Indirect dependence