

ECE 1749H:
Interconnection Networks for
Parallel Computer Architectures:
Routing

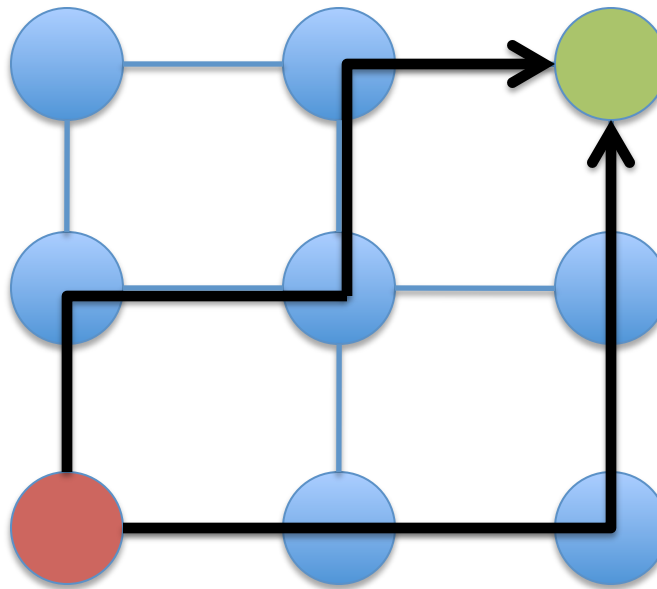
Prof. Natalie Enright Jerger

Routing Overview

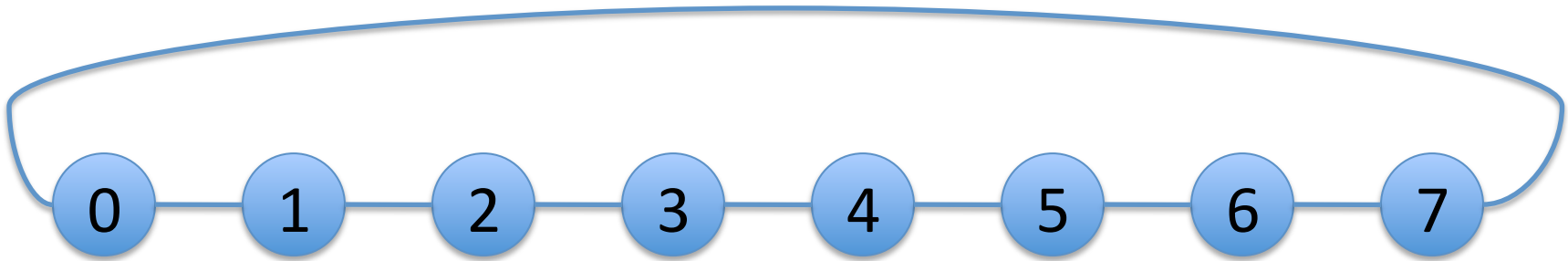
- Discussion of topologies assumed ideal routing
- In practice...
 - Routing algorithms are not ideal
- Goal: distribute traffic **evenly** among paths
 - Avoid hot spots, contention
 - More balanced → closer throughput is to ideal
- Keep complexity in mind

Routing Basics

- Once topology is fixed
- Routing algorithm determines path(s) from source to destination

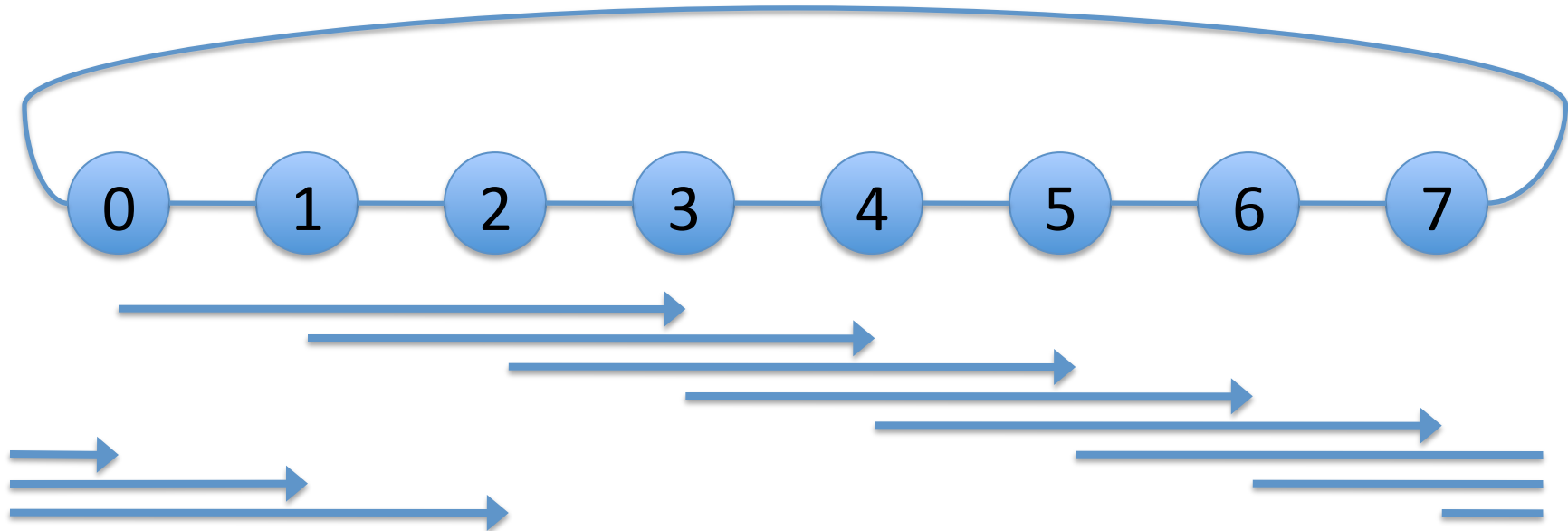


Routing Example



- Some routing options:
 - Greedy: shortest path
 - Uniform random: randomly pick direction
 - Adaptive: send packet in direction with lowest local channel load
- Which gives best worst-case throughput?

Routing Example (2)

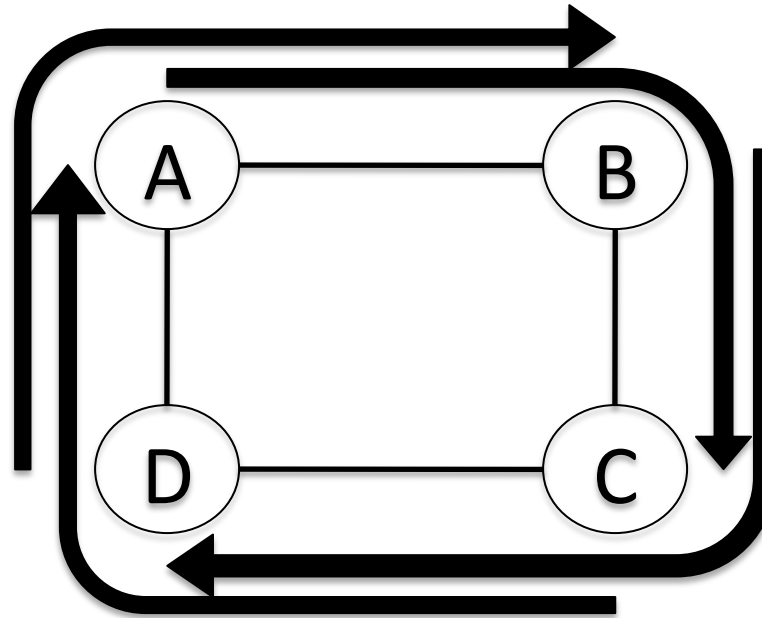


- Consider tornado traffic
 - node i sends to $i+3 \bmod 8$
- Greedy:
 - All traffic moves clockwise
 - Counterclockwise channels are idle
- Adaptive is best

Routing Algorithm Attributes

- Types
 - Deterministic, Oblivious, Adaptive
- Number of destinations
 - Unicast, Multicast, Broadcast?
- Adaptivity
 - Oblivious or Adaptive? Local or Global knowledge?
 - Minimal or non-minimal?
- Implementation
 - Source or node routing?
 - Table or circuit?

Routing Deadlock

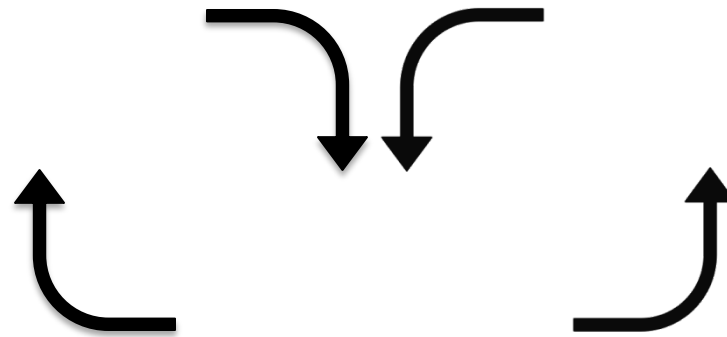


- Each packet is occupying a link and waiting for a link
- Without routing restrictions, a **resource cycle** can occur
 - Leads to deadlock

Deterministic

- All messages from *Source* to *Destination* traverse the same path
- Common example: Dimension Order Routing (DOR)
 - Message traverses network dimension by dimension
 - Aka XY routing
- Cons:
 - Eliminates any path diversity provided by topology
 - **Poor load balancing**
- Pros:
 - **Simple** and inexpensive to implement
 - **Deadlock-free**

Dimension Order Routing



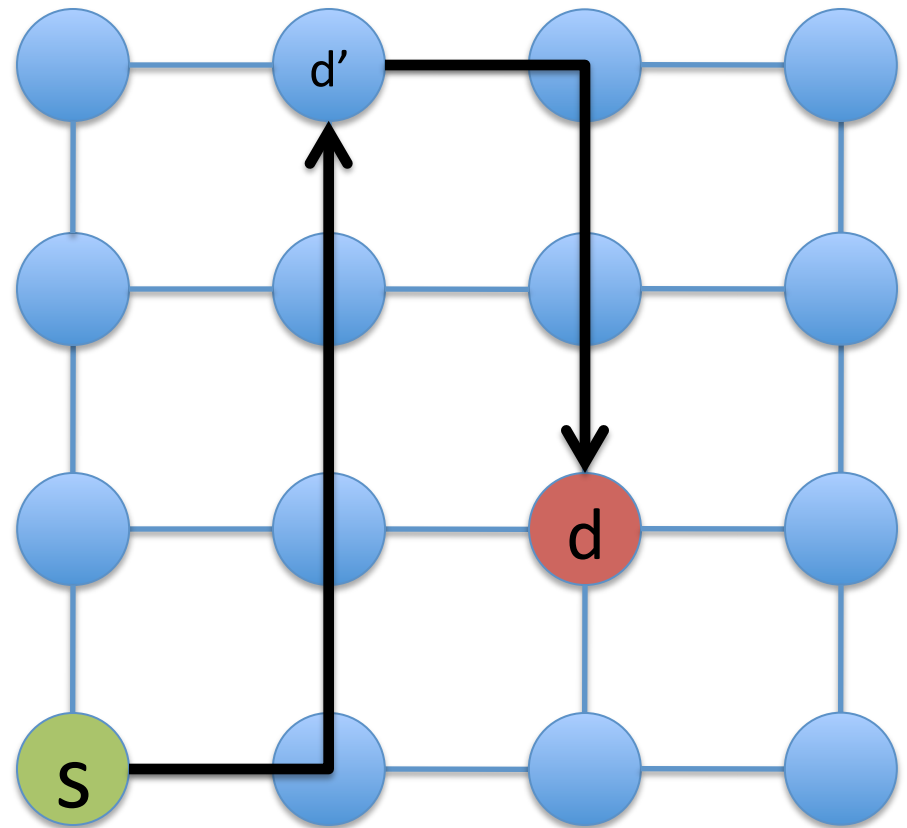
- a.k.a X-Y Routing
 - Traverse network dimension by dimension
 - Can only turn to Y dimension after finished X

Oblivious

- Routing decisions are made without regard to network state
 - Keeps algorithms simple
 - Unable to adapt
- Deterministic algorithms are a subset of oblivious

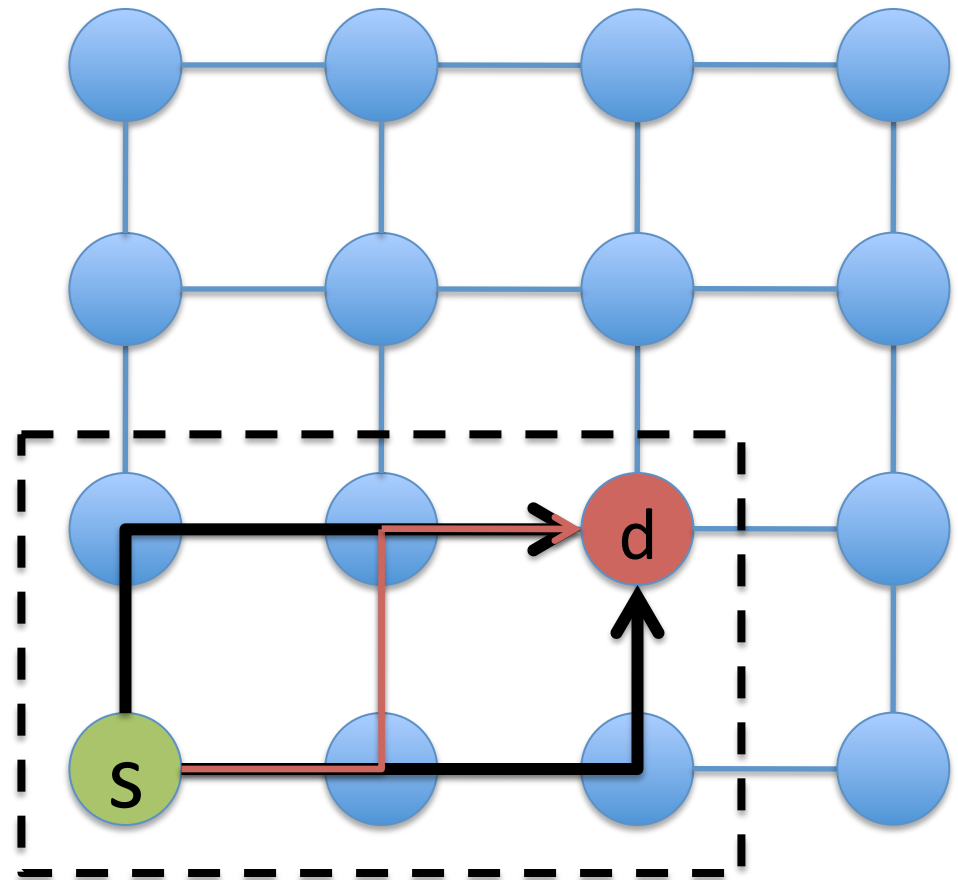
Valiant's Routing Algorithm

- To route from s to d
 - Randomly choose intermediate node d'
 - Route from s to d' and from d' to d .
- Randomizes any traffic pattern
 - All patterns appear uniform random
 - Balances network load
- Non-minimal
- Destroys locality



Minimal Oblivious

- Valiant's: Load balancing but significant increase in hop count
- Minimal Oblivious: some load balancing, but use shortest paths
 - d' must lie within min quadrant
 - 6 options for d'
 - Only 3 different paths



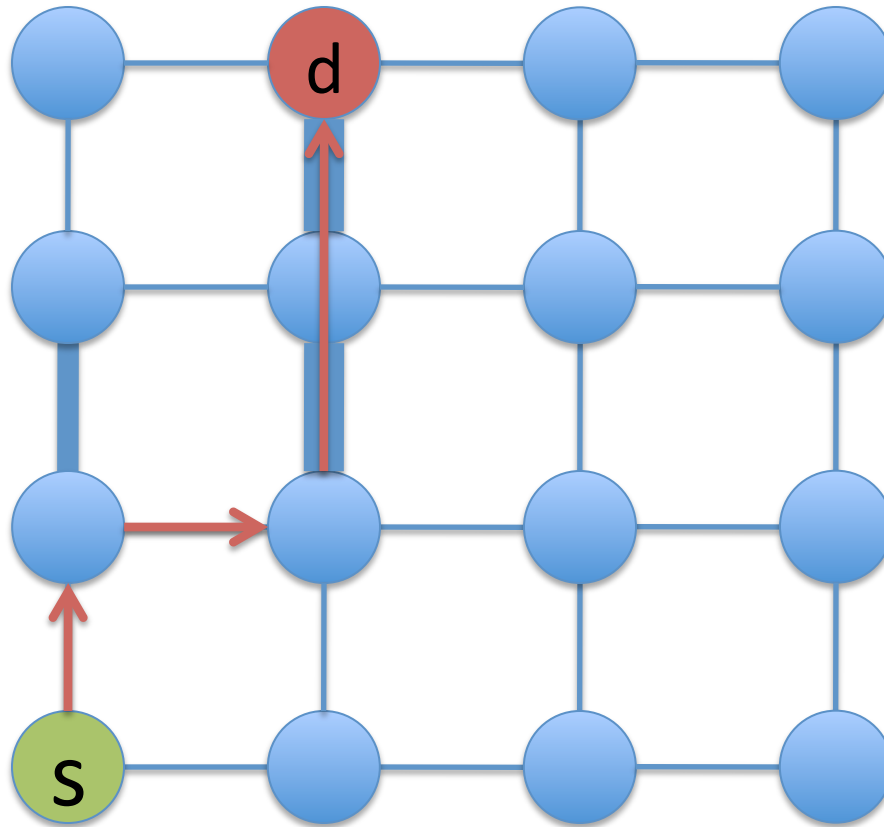
Oblivious Routing

- Valiant's and Minimal Adaptive
 - Deadlock free
 - When used in conjunction with X-Y routing
- Randomly choose between X-Y and Y-X routes
 - Oblivious but not deadlock free!

Adaptive

- Exploits path diversity
- Uses network state to make routing decisions
 - Buffer occupancies often used
 - Coupled with flow control mechanism
- Local information readily available
 - Global information more costly to obtain
 - Network state can change rapidly
 - Use of local information can lead to non-optimal choices
- Can be minimal or non-minimal

Minimal Adaptive Routing

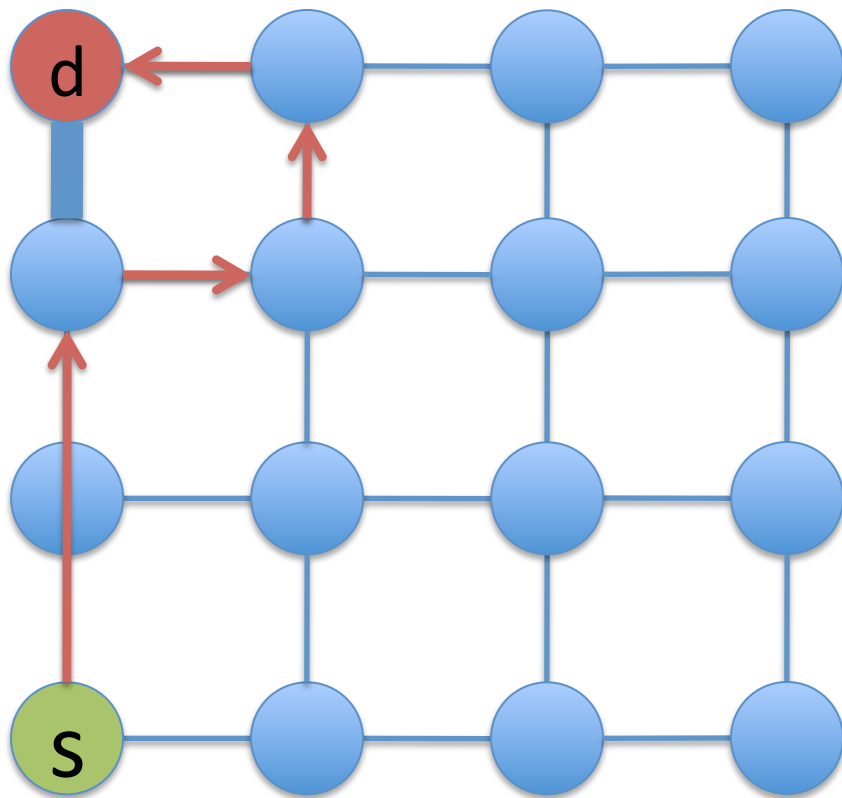


- Local info can result in sub-optimal choices

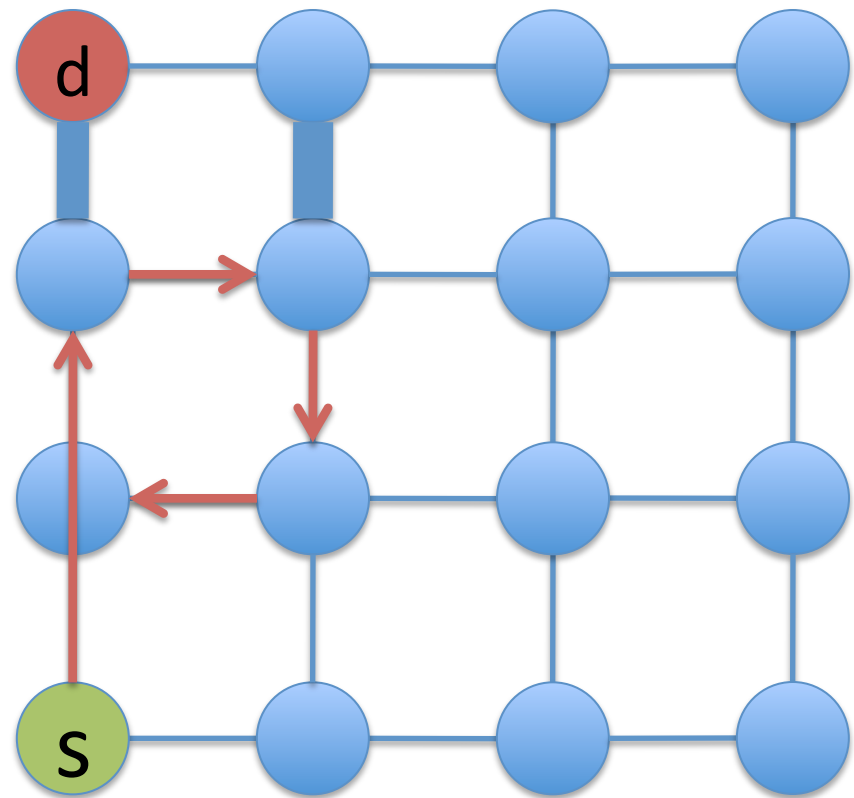
Non-minimal adaptive

- Fully adaptive
- Not restricted to take shortest path
- Misrouting: directing packet along non-productive channel
 - Priority given to productive output
 - Some algorithms forbid U-turns
- Livelock potential: traversing network without ever reaching destination
 - Mechanism to guarantee forward progress
 - Limit number of misroutings

Non-minimal routing example

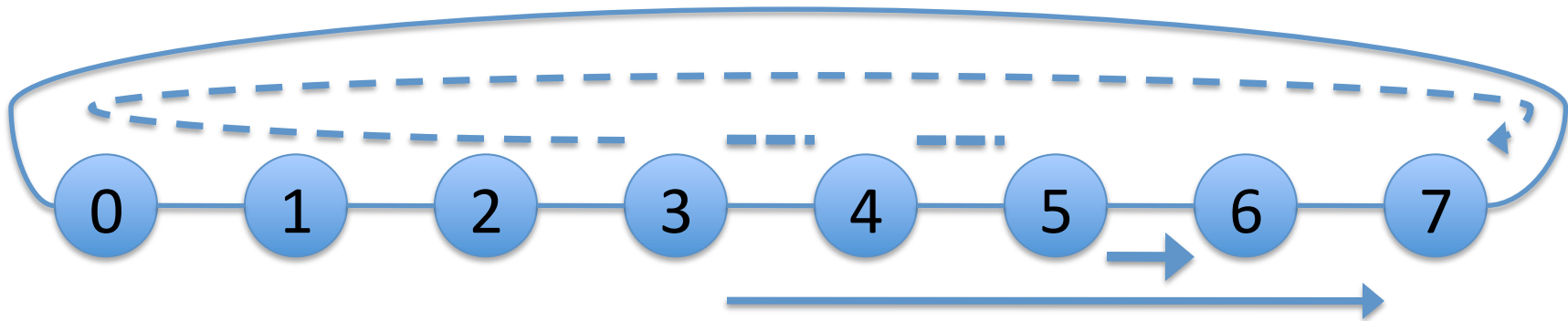


- Longer path with potentially lower latency



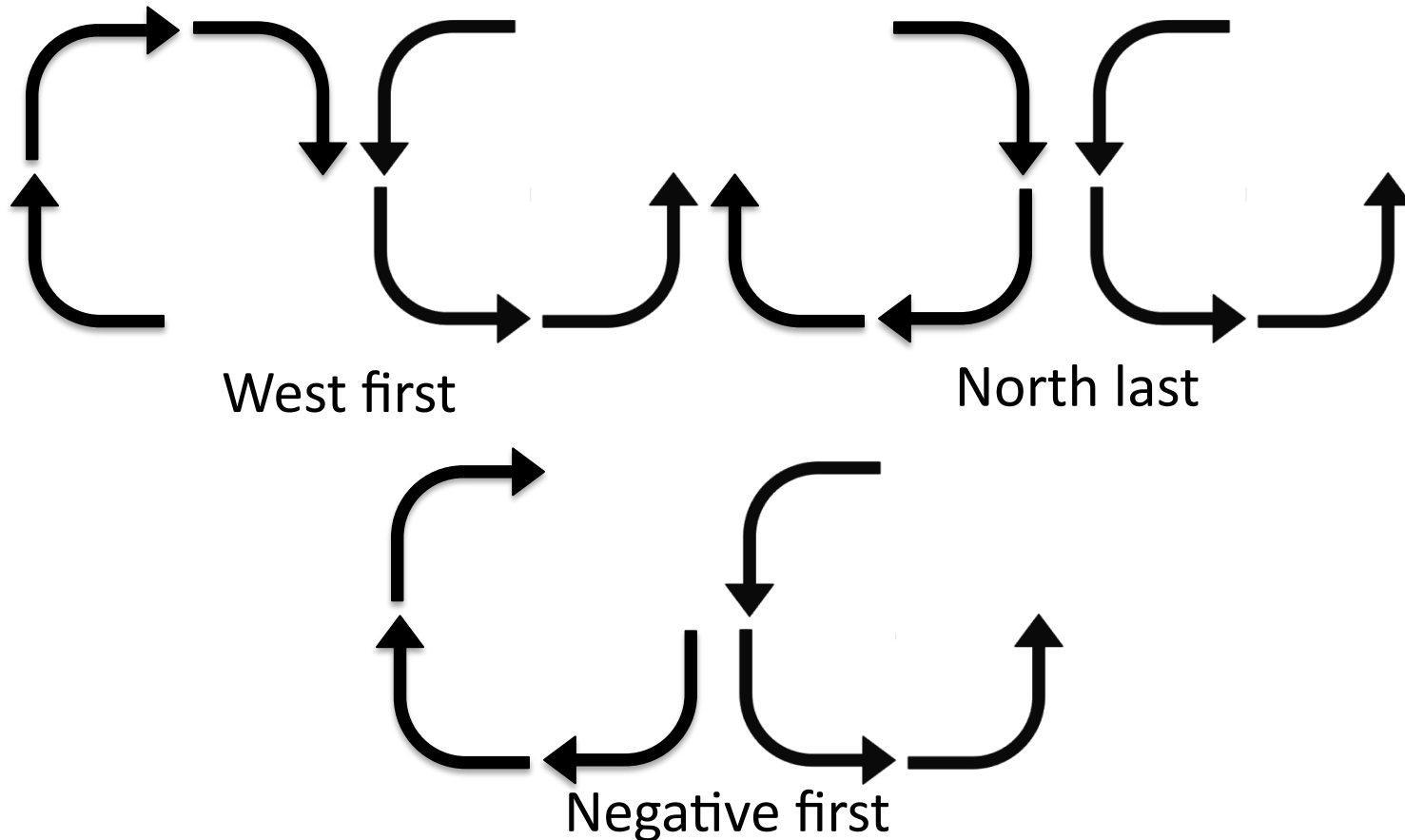
- Livelock: continue routing in cycle

Adaptive Routing Example



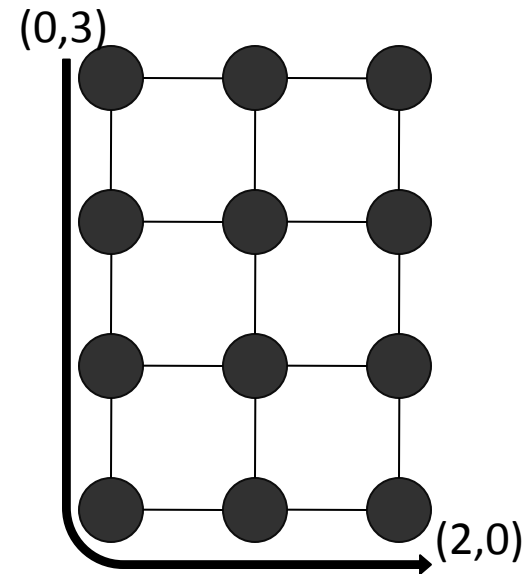
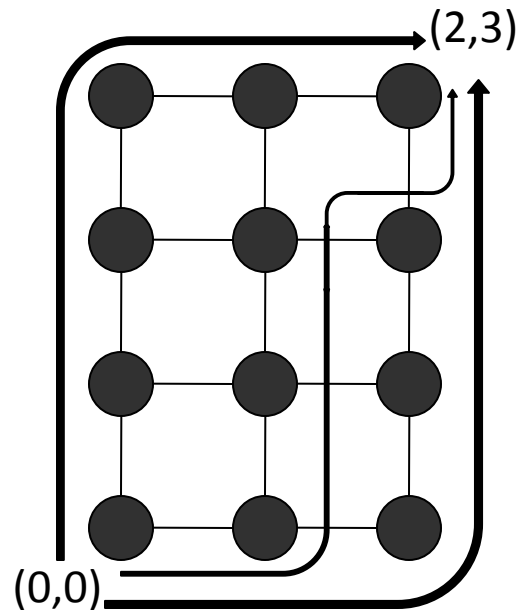
- Should 3 route clockwise or counterclockwise to 7?
 - 5 is using all the capacity of link $5 \rightarrow 6$
- Queue at node 5 will sense contention but not at node 3
- Backpressure: allows nodes to indirectly sense congestion
 - Queue in one node fills up, it will stop receiving flits
 - Previous queue will fill up
- If each queue holds 4 packets
 - 3 will send 8 packets before sensing congestion

Adaptive Routing: Turn Model



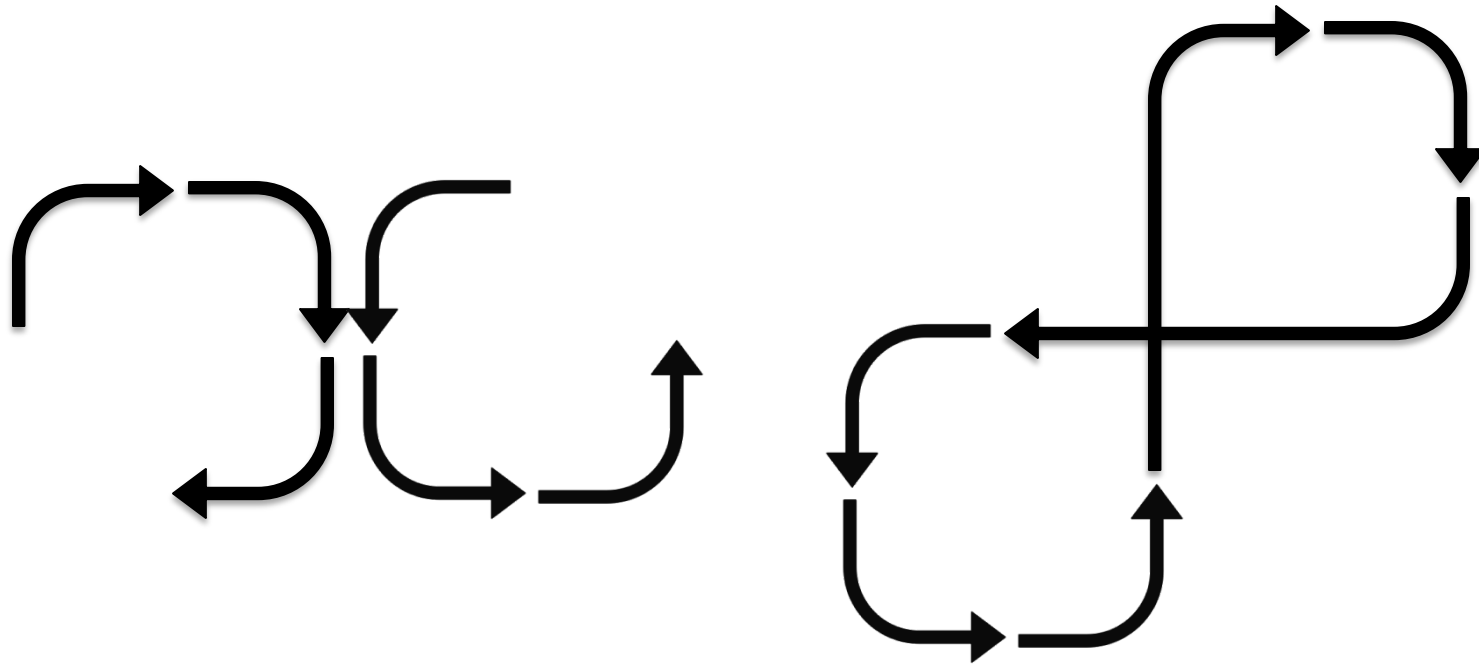
- Some adaptivity by removing 2 of 8 turns
 - Remains deadlock free (like DOR)

Negative-First Routing Example



- Limited or no adaptivity for certain source-destination pairs

Turn Model Routing Deadlock



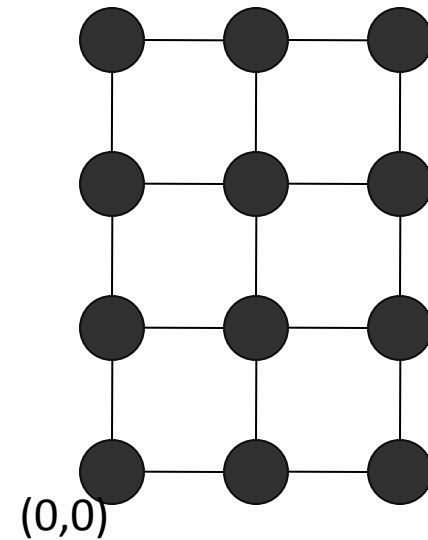
- Not a valid turn elimination
 - Resource cycle results

Routing Implementation

- Source tables
 - Entire route specified at source
 - Avoids per-hop routing latency
 - Unable to adapt dynamically to network conditions
 - Can specify multiple routes per destination
 - Give fault tolerance and load balance
 - Support reconfiguration (not specific to topology)

Source Table Routing

Destination	Route 1	Route 2
00	X	X
10	EX	EX
20	EEX	EEX
01	NX	NX
11	NEX	ENX
21	NEEX	ENEX
02	NNX	NNX
12	ENNX	NNEX
22	EENNX	NNEEX
03	NNNX	NNNX
13	NENNX	ENNNX
23	EENNNX	NNNEEX



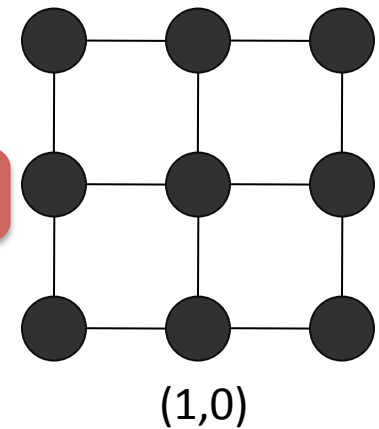
- Arbitrary length paths: storage overhead and packet overhead

Node Tables

- Store only next direction at each node
- Smaller tables than source routing
- Adds per-hop routing latency
- Can adapt to network conditions
 - Specify multiple possible outputs per destination

Node Table Routing

	To								
From	00	01	02	10	11	12	20	21	22
00	X -	N -	N -	E -	E N	E N	E -	E N	E N
01	S -	X -	N -	E S	E -	E N	E S	E -	E N
02	S -	S -	X -	E S	E S	E -	E S	E S	E -
10	W -	W -	W -	X -	N -	N -	E -	E N	E N
11	W -	W -	W -	S -	X -	N -	E S	E -	E N
12	W -	W -	W -	S -	S -	X -	E S	E S	E -
20	W -	W -	W -	W -	W -	W -	X -	N -	N -
21	W -	W -	W -	W -	W -	W -	S -	X -	N -
22	W -	W -	W -	W -	W -	W -	S -	S -	X -

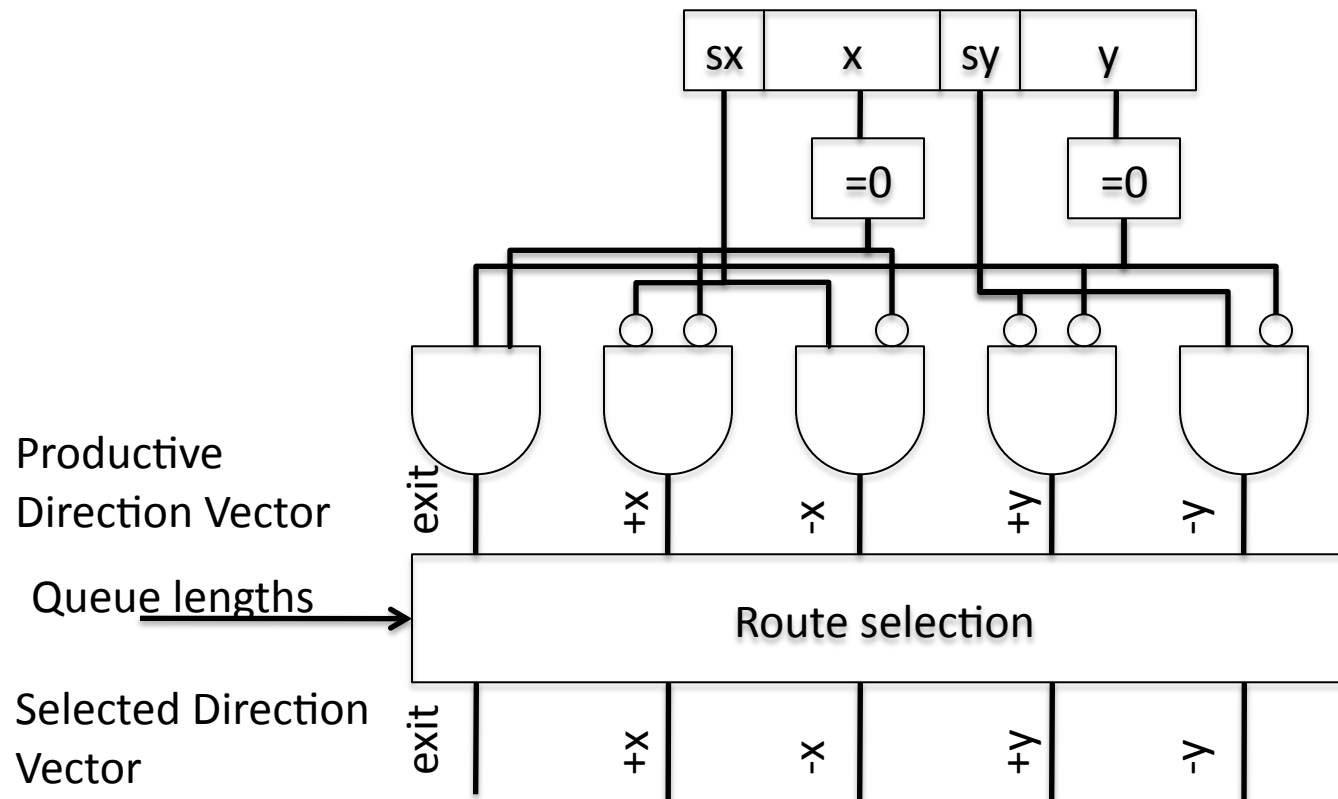


- Implements West-First Routing
- Each node would have 1 row of table
 - Max two possible output ports

Implementation

- Combinational circuits can be used
 - Simple (e.g. DOR): low router overhead
 - Specific to one topology and one routing algorithm
 - Limits fault tolerance
- Tables can be updated to reflect new configuration, network faults, etc

Circuit Based



- Next hop based on buffer occupancies
- Or could implement simple DOR
- Fixed w.r.t. topology

Routing Summary

- Latency paramount concern
 - Minimal routing most common for NoC
 - Non-minimal can avoid congestion and deliver low latency
- To date: NoC research favors DOR for simplicity and deadlock freedom
 - On-chip networks often lightly loaded
- Only covered unicast routing
 - Recent work on extending on-chip routing to support multicast

Seminar Announcement

- Title: Living in interesting times:
Disruptive trends in computer
architecture
- Where: GB405
- When: Wednesday, January 20, 2010,
2-3pm
- <http://www.eecg.utoronto.ca/cider/>
- Speaker: Bob Blainey, IBM Toronto