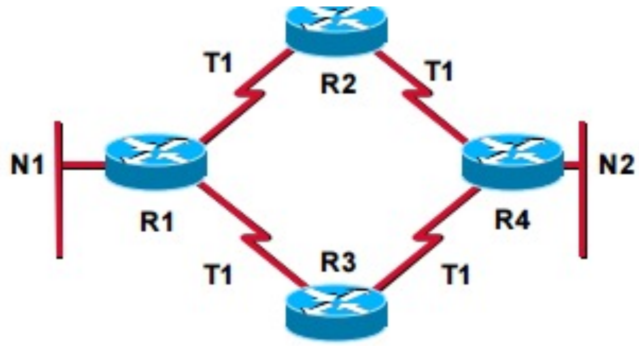
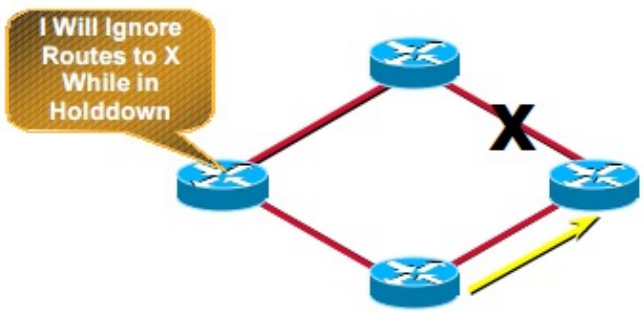
University of BRISTOL

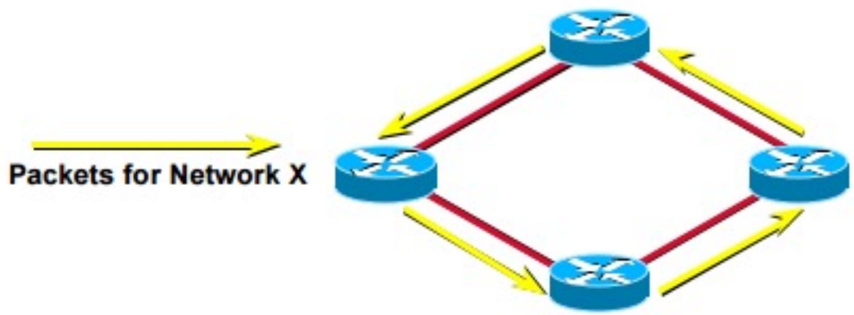
Lecture 6
Part 1: Routing

Introduction of Routing

- Convergence: time required for router to identify and use an alternate path
  - Dependent on timer values and algorithm
  - Difficult to predict precisely
- Load balancing: process of distributing network traffic across multiple servers.
  - Rapid failover
  - Achieve maximum throughput
- Holddown: Set minimum convergence time
  - Prevents routing loops







Routing Loop: A Routing Disagreement

- Metrics: Numeric value used to choose among paths

Metrics are cost values used by routers to determine the best path to a destination network. The most common metric values are hop, bandwidth, delay, reliability, load, and cost.

  - RIP/RIPv2 is hop count and ticks (IPX)

    The metric field in the RIP IP packet will tick by one for every router that it the packet hits. Once it goes to 15 hops (ticks) it is declared as unreachable

  - OSPF/ISIS is interface cost (bandwidth)
  - (E)IGRP is compound

- Path determination depends on metric

Routing protocol goals (What's a good routing algorithm?)

- Optimal path selection: short path lengths
  - Reduce hops and the overall latency
  - A trade-off exists in oblivious routing for load balance and average path length
- Loop-free routing
  - Routing interacts with the flow control of the network to avoid deadlocks and/or livelocks.
- Adapts to changes easily and quickly
  - Ability to work in the presence of faults in the network
- Routing algorithm must ensure freedom from Starvation
  - under scenarios where certain packets are prioritized during routing, some of the low priority packets never reach their intended destination
  - can be avoided by using a fair routing algorithm, or reserving some bandwidth for low priority data packets

Routing protocol goals (What's a good routing algorithm?))

Good load balances across the network channels even in the presence of non-uniform traffic pattern

- Permutation traffic

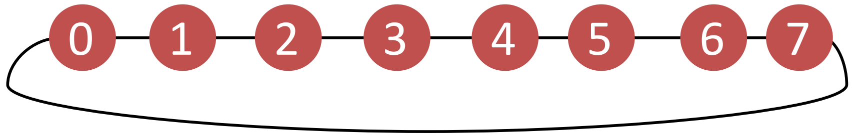
All traffic from each source can be represented by a permutation function, that maps source to destination.

Name	Pattern
Random	$\lambda_{sd} = 1/N$
Permutation	$d = \pi(s)$
Bit permutation	$d_i = s_{f(i)} \oplus g(i)$
Bit complement	$d_i = \neg s_i$
Bit reverse	$d_i = s_{b\text{-}i\text{-}1}$
Bit rotation	$d_i = s_{i+1} \bmod b$
Shuffle	$d_i = s_{i\text{-}1} \bmod b$
Transpose	$d_i = s_{i+b/2} \bmod b$
Digit permutations	$d_x = f(s_{g(x)})$
Tornado	$d_x = s_x + (\lceil k/2 \rceil\text{-}1) \bmod k$
Neighbor	$d_x = s_x + 1 \bmod k$

- Large load misbalances will cause suboptimal throughput

Routing example for an 8-node ring network

- Greedy
  - Shortest path and direction
- Uniform random
  - Randomly pick a direction for each packet with equal probability
- Weighted random
  - Randomly pock a direction for each packet, but weight the short direction with probability 1-Δ/8 (Δ is the minimum distance)
- Adaptive
  - Send packet in the direction for which the local channel has the lowest load



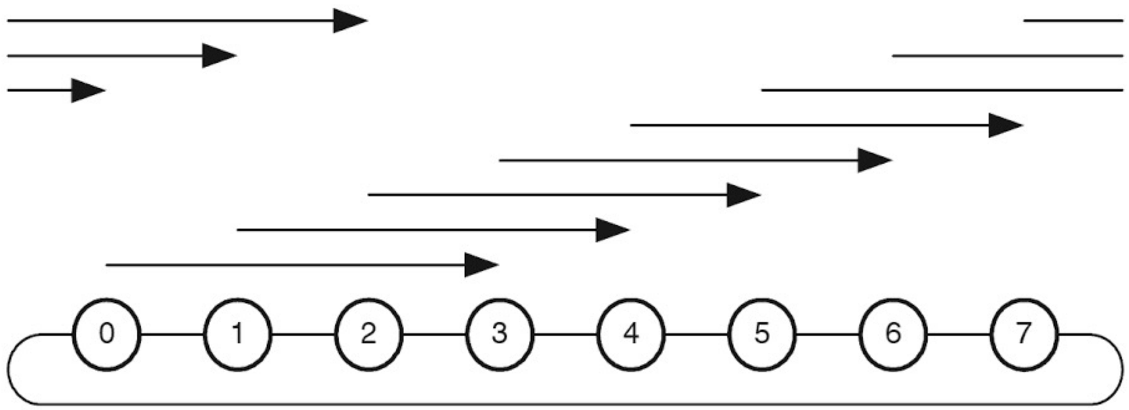
Discussion: Which algorithm gives the best worst-case throughput?

Tornado traffic on an 8-node ring network

- Tornado traffic

$$d_x = s_x + \left(\frac{k}{2} - 1\right) mod\ k$$

$$d_x = s_x + 3$$



- Throughput of several example routing algorithms

Algorithm	*Throughput on Tornado*
Greedy	0.33
Random	0.40
Weighted random	0.53
Adaptive	0.53

Routing algorithms Classification: Adaptivity

- Oblivious algorithms
  - Choose a route without considering any information about the network present state. (Deterministic algorithms are a subset of oblivious algorithms)
  - Deterministic algorithms

    Always choose same path (ignore path diversity)

- Adaptive algorithms
  - The state of the network is incorporated in making the routing decision to adapt to network state such as network congestion.
  - Adapt to the state of the network: node, link, length of queues, historical channel load information.

Routing algorithms Classification: Routing Decision

- Source routing: The routing path is determined at the source and the path computation only needs to be done once for each packet.
  - routing information is looked up at the source and routing information is added to the header of the packet (increasing packet size)
  - when a packet arrives at a router, the routing information is extracted from the routing field in the packet header
  - does not require a destination address in a packet, any intermediate routing tables, or functions needed to calculate the route
- √ Per-hop routing: At each hop en route to the destination, the packet goes through routing computation to determine the next productive hop.
  - e.g. XY co-ordinates or number identifying destination node/router
  - routing decisions are made in each router by looking up the destination addresses in a routing table or by executing a hardware function

Routing algorithms Classification: Hop Count

- Minimal Routing:

  Minimal number of hop count between source and destination is traversed. Depending on the topology and the adaptivity of the routing algorithm, there might be multiple minimal paths.

- Nonminimal Routing:

  The number of hop count traversed en route to the destination node exceeds the minimal hop count. Nonminimal routing increases path diversity and can improve network throughput.

Deterministic

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

Dimension-order routing (s:03, d:22)

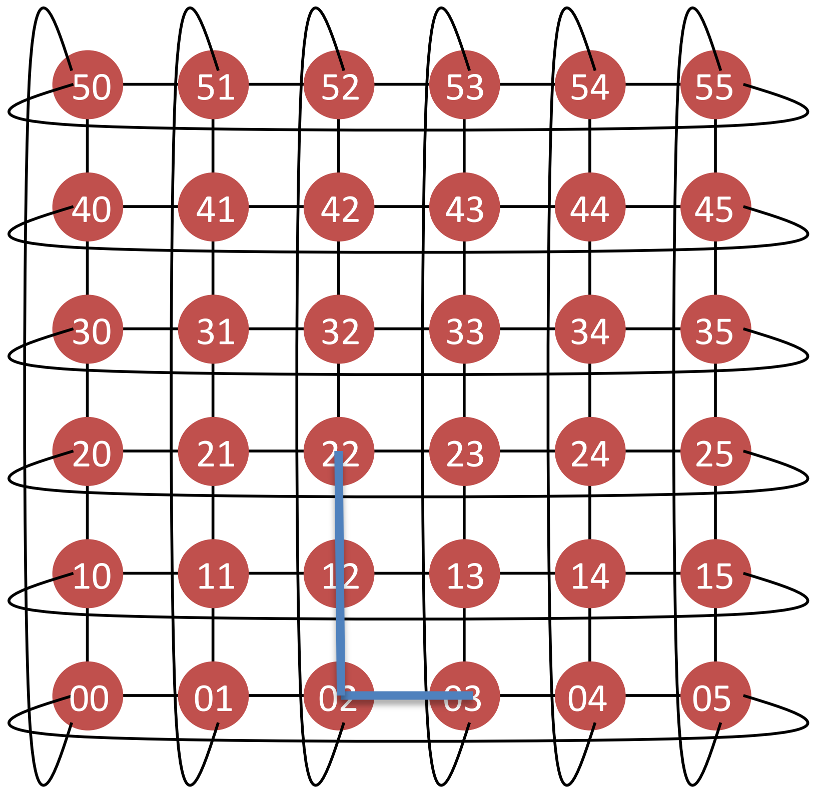
- Step 1: Choose preferred directions

$$m_i = d_i - s_i \bmod k$$

$$\Delta_i = m_i - \begin{cases} 0 & \text{if } m_i \le k/2 \\ k & \text{otherwise} \end{cases}$$

$$D_i = \begin{cases} 0 & \text{if } \Delta_i = k/2 \\ sign(\Delta_i) & \text{otherwise} \end{cases}$$

In our case: D=(+1, -1)

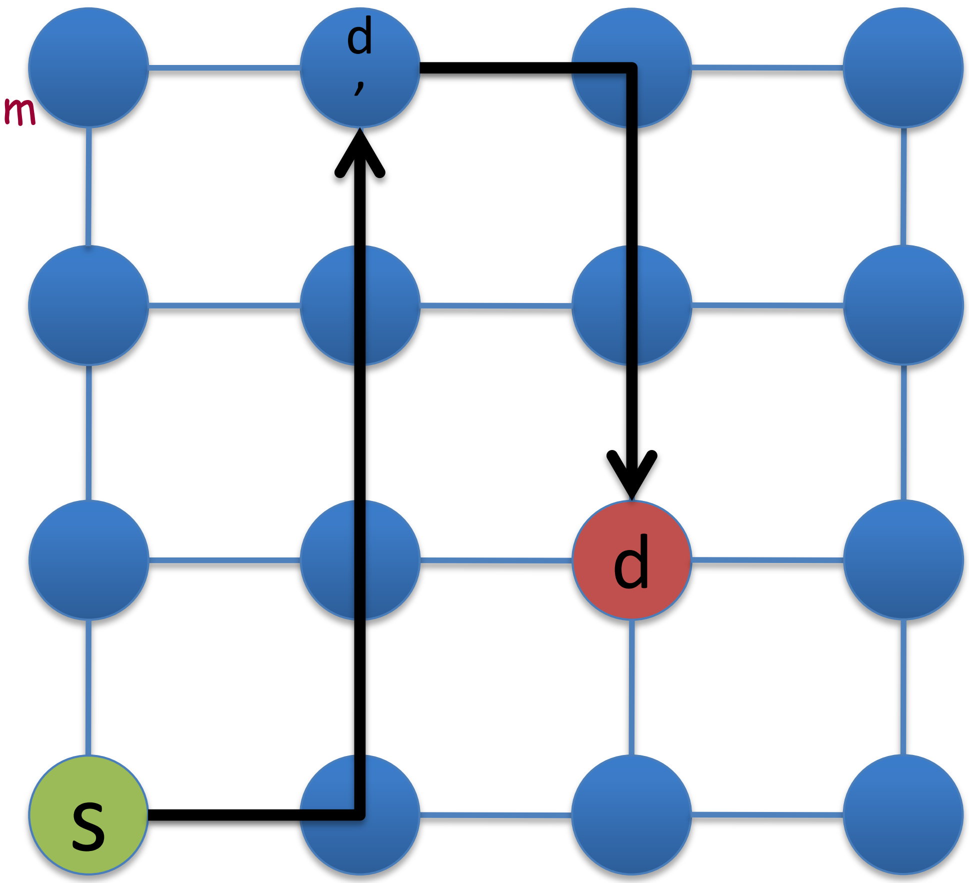


A 6-ary 2-cube (k-ary n-cube) network (2D torus network)

- Step 2: Routing
- Pros & Cons
  - Simple to implement
  - Simplify the problem of deadlock avoidance by preventing any cycles of channel dependency between dimensions
  - Poor load balancing

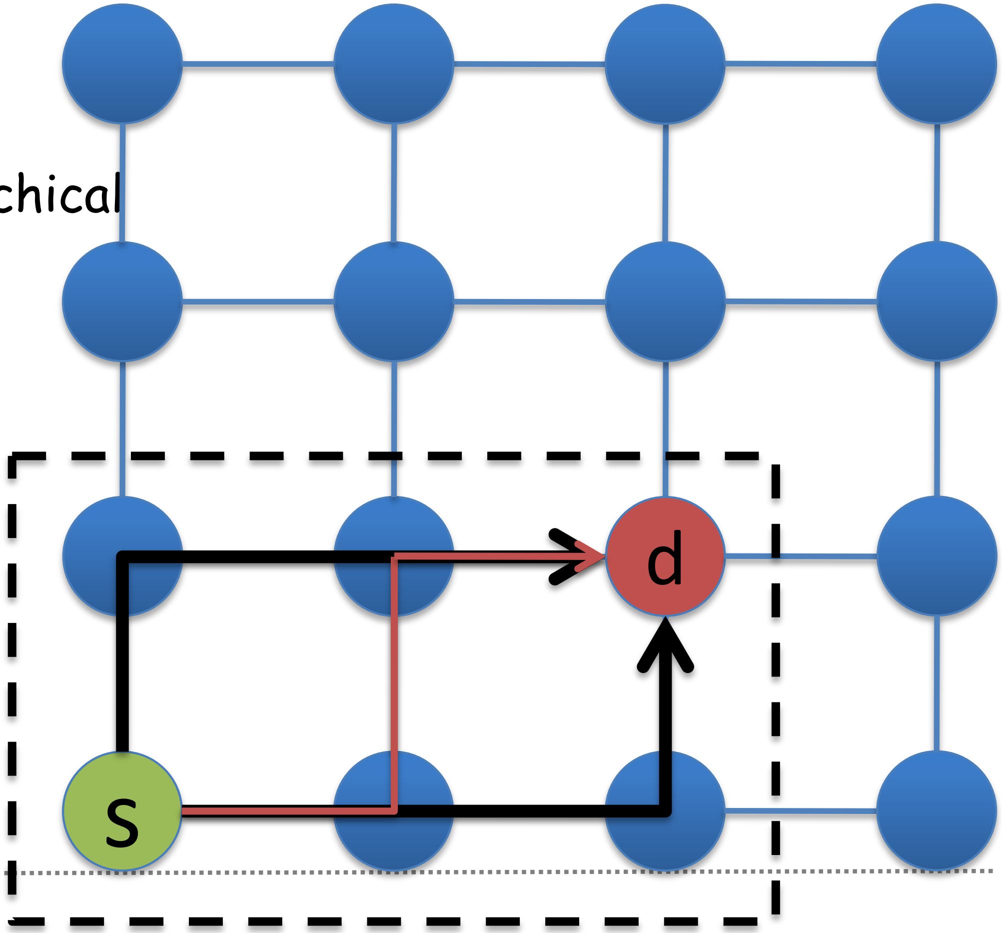
Oblivious Routing: Valiant's Routing Algorithm

- 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 to be uniform random
  - Balances network load
- Tradeoff between locality and load balance
- Twice the load of random traffic
- Half the capacity of a network



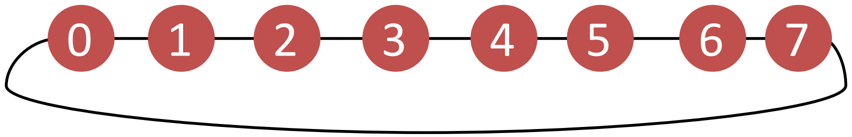
Oblivious Routing: Minimal Oblivious

- Valiant's: Load balancing comes at expense of significant hop count increase
  - Destroys locality
- Minimal Oblivious: achieve some load balancing, but use shortest paths
  - d' must lie within minimum quadrant
  - 6 options for d'
  - Only 3 different paths
- It works extremely well for hierarchical topologies, such as fat tee.



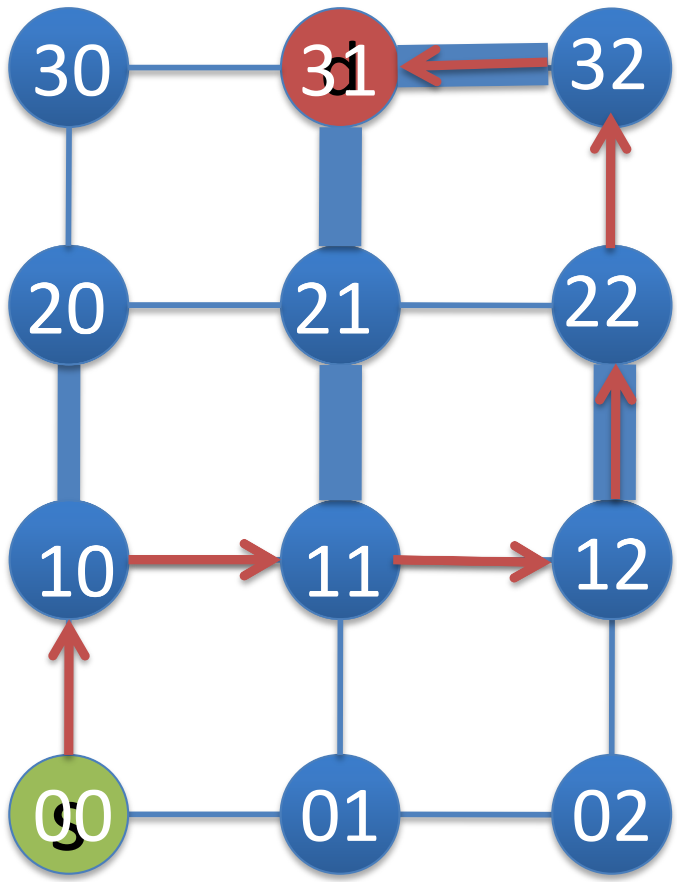
Adaptive Routing

- Uses network state to make routing decisions
  - Buffer occupancies often used
  - Couple with flow control mechanism (no coupling in deterministic and oblivious routing)
- Local information readily available
  - Flit or packet queues at the present node to estimate congestion
  - Backpressure enable routers sense congestion
- Challenges
  - 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

- Routing decisions are taken at each hop
  - At each hop a routing function generates a productive output vector
  - The vector identifies the output channel which will move the packet closer to the destination
  - Network state (usually queue length) is used to select the output channel
- Minimal adaptive routing is good at locally balancing, but poor at global load balance.
- Local info can result in sub-optimal choices

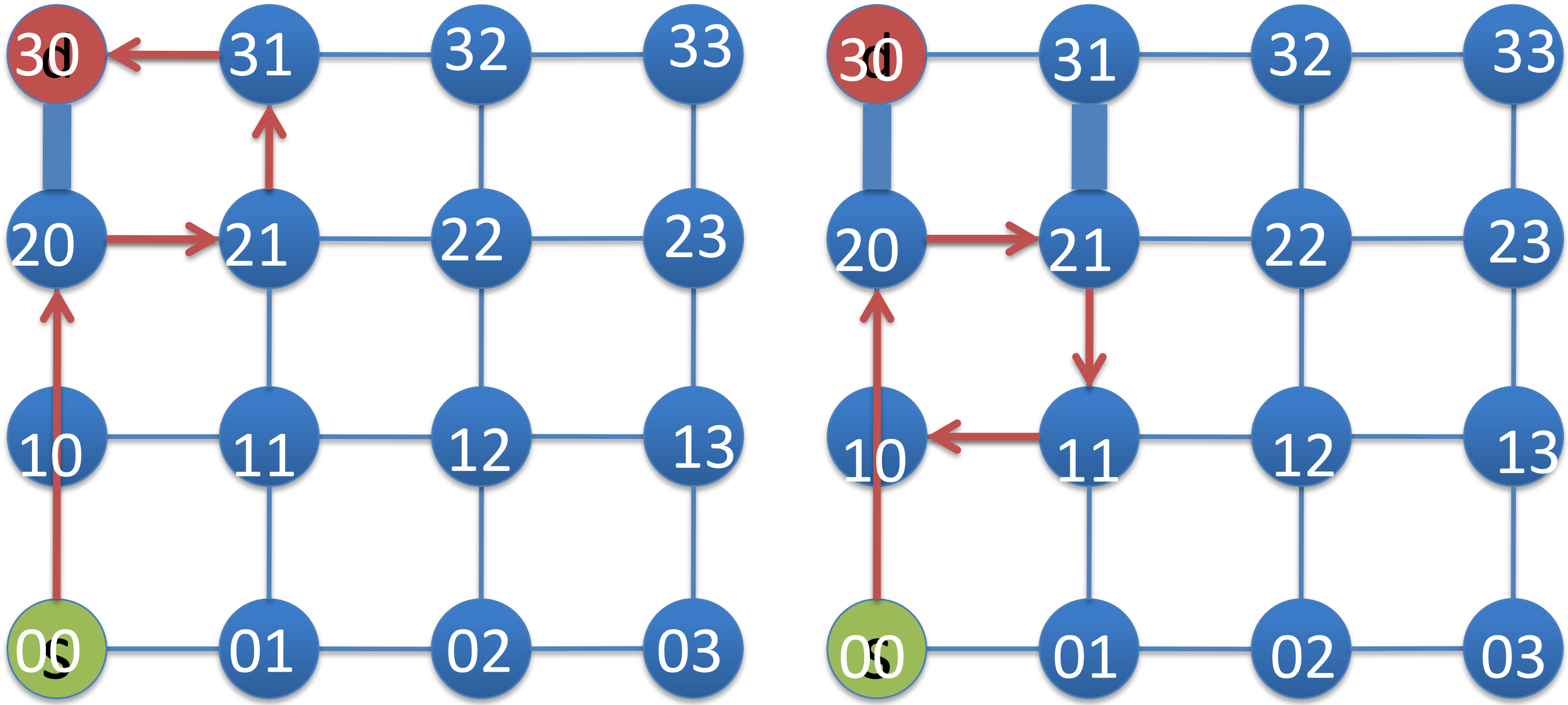


Non-minimal adaptive (Fully adaptive)

- 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 mis-routings

Non-minimal routing (Fully adaptive)

- Longer path with potentially lower latency



- Deadlock: continue routing in cycle
- Livelock: no productive move towards destination

Discussion

- Is adaptive routing algorithm better than other routing algorithms?
  - Poor worst-case performance
    - Local nature of most practical adaptive routing information
    - Balance local load but poor in global load balance
  - Delay responding to a change in traffic changes
  - Routing algorithm requirements