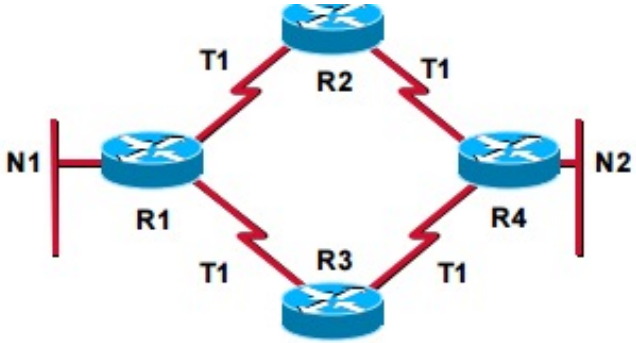
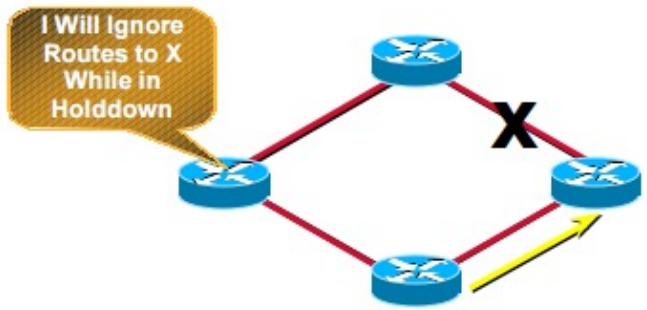
University of BRISTOL

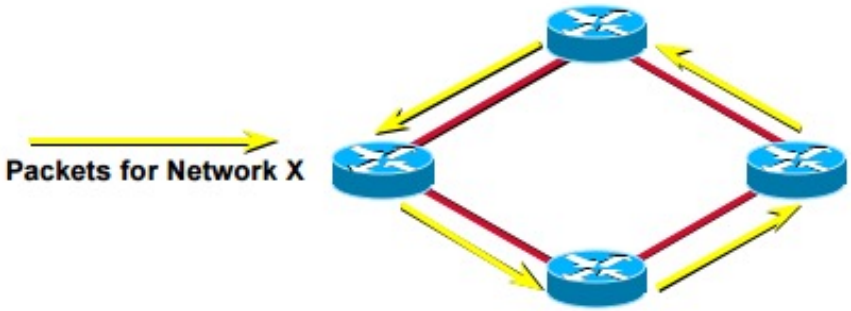
Lecture 6
Part 1: Routing

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

Introduction of Routing

University of BRISTOL

- 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

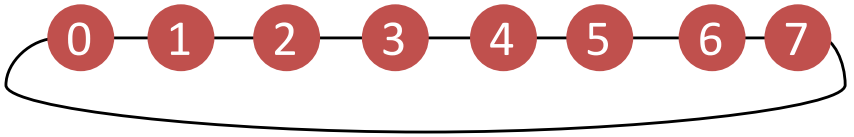
- Permutation traffic

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	$d_i = s_{b-i-1}$
Bit rotation	$d_i = s_{i+1} \bmod b$
Shuffle	$d_i = s_{i-1 \bmod b}$
Transpose	$d_i = s_{i+b/2 \bmod b}$
Digit permutations	$d_x = f(s_{g(x)})$
Tornado	$d_x = s_x + (\lceil k/2 \rceil - 1) \bmod k$
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 $1-\Delta/8$ (Δ is the minimum distance)
- Adaptive
  - Send packet in the direction for which the local channel has the lowest load

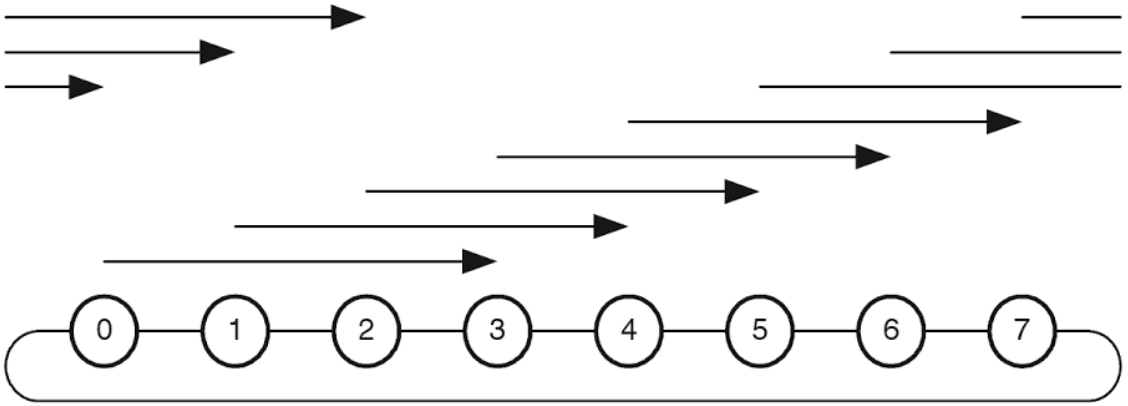
Discussion: Which algorithm gives the best worst-case throughput?

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

University of BRISTOL

- 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

University of BRISTOL

- 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

University of BRISTOL

- 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

University of BRISTOL

- 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)

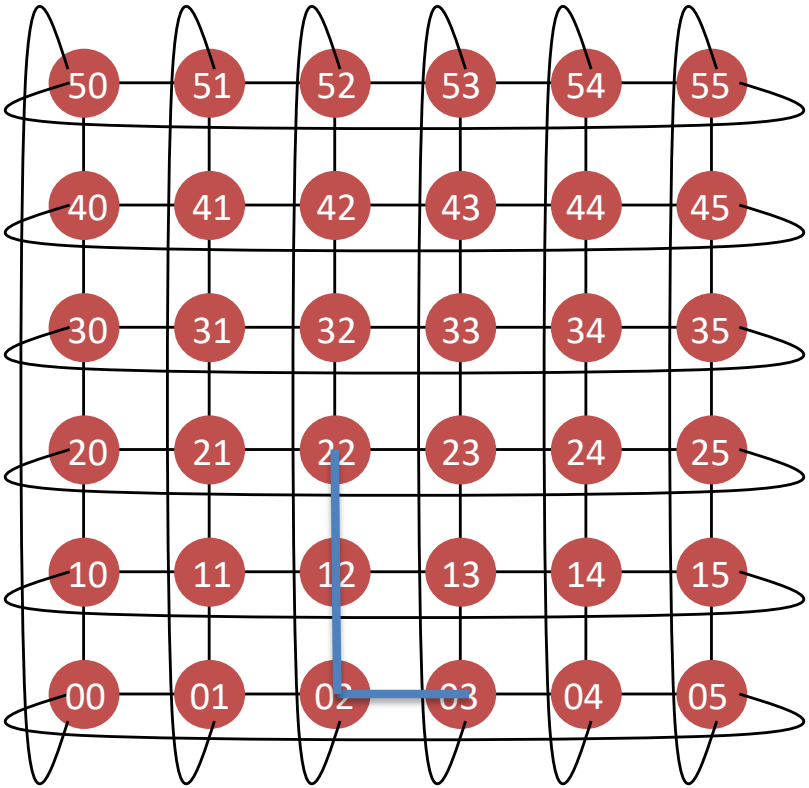
- Step 1: Choose preferred directions

$$m_i = d_i - s_i \; mod \; k$$

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

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

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



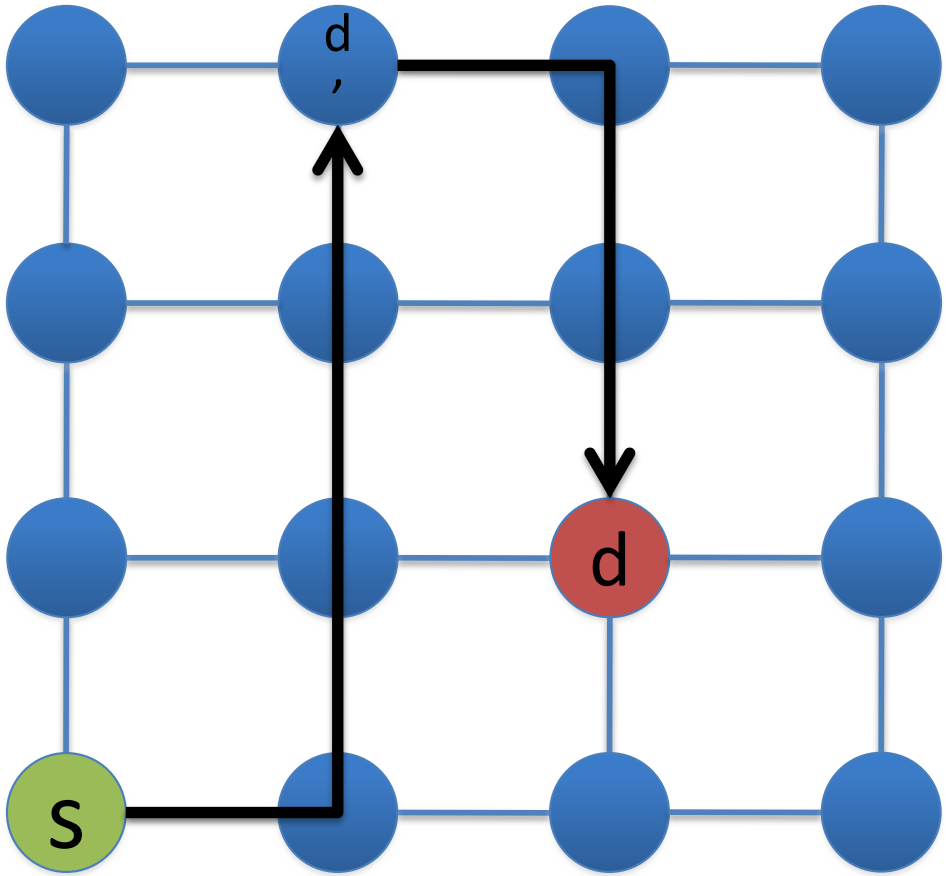
A 6-ary 2-cube (k-ary n-cube) network (2D torus network)

- 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

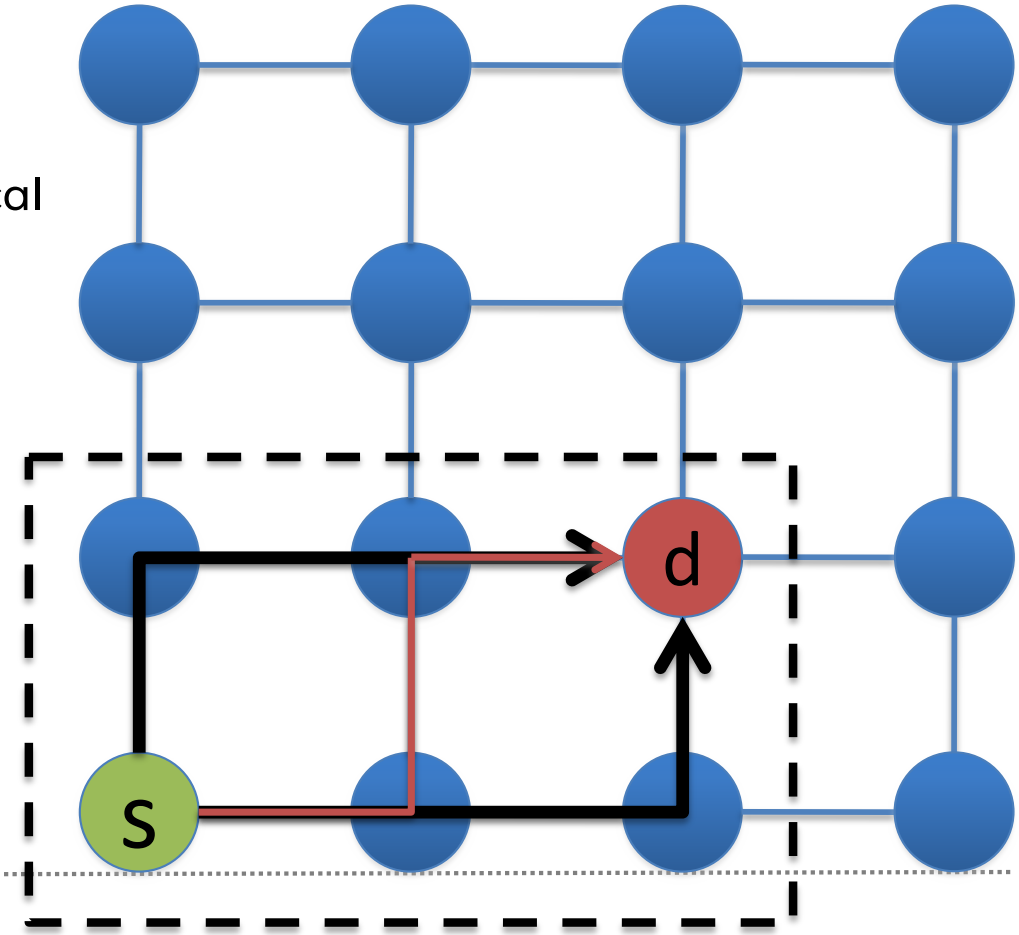
University of BRISTOL

- 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



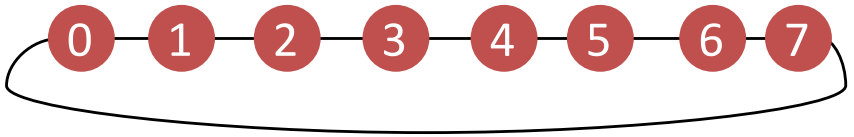
Oblivious Routing: Minimal Oblivious

- 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

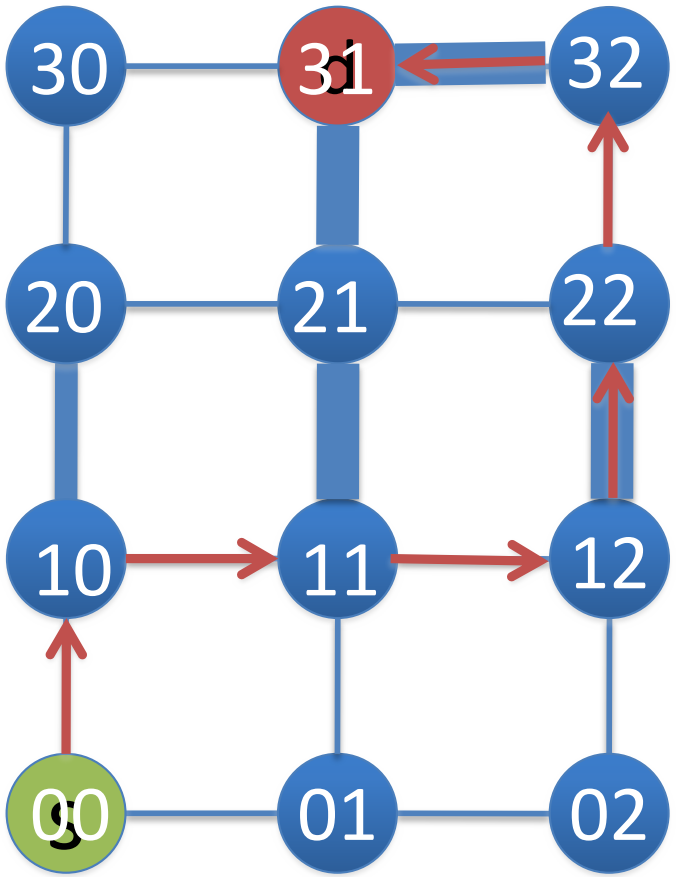
- Uses network state to make routing decisions
  - Buffer occupancies often used
  - Couple with flow control mechanism (no coupling in deterministic and oblivious routing)
- Local information readily available
  - Flit or packet queues at the present node to estimate congestion
  - Backpressure enable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



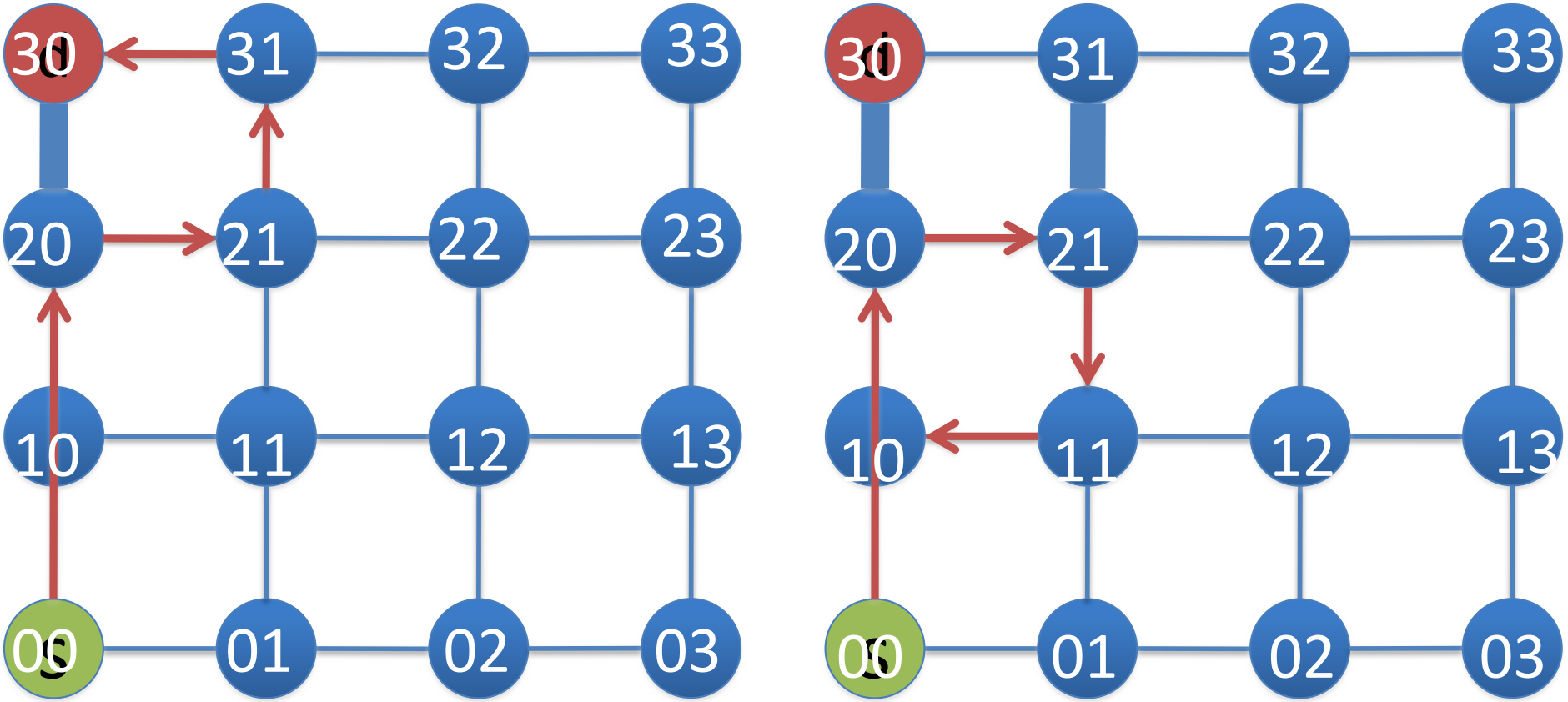
Non-minimal adaptive (Fully adaptive)

University of BRISTOL

- 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

University of BRISTOL

Lecture 6

Part 2: Flow Control

Switching/Flow Control Overview

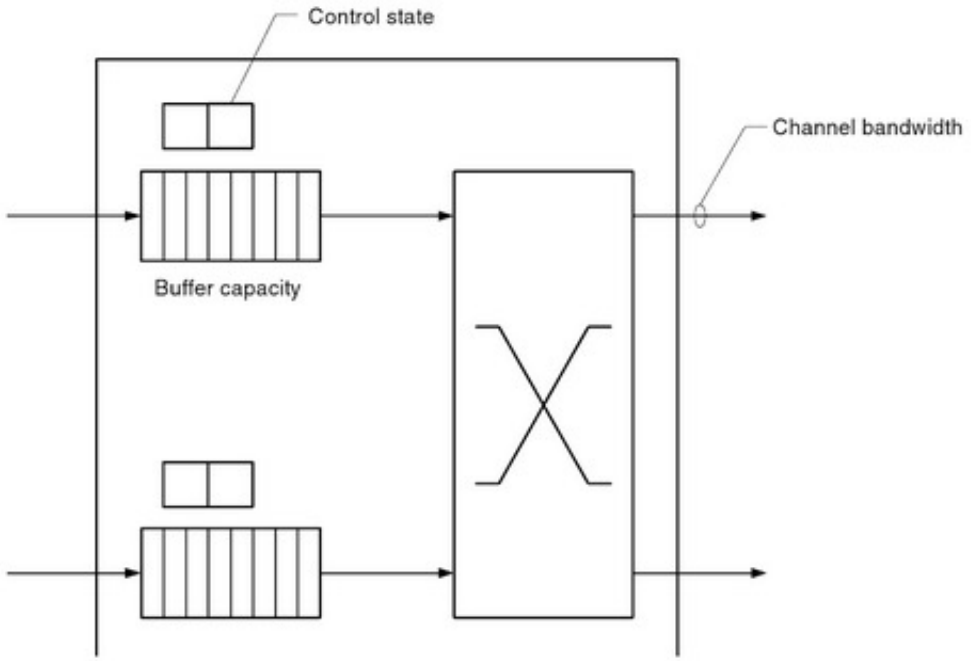
University of BRISTOL

- Topology: determines connectivity of network
- Routing: determines paths through network
- Flow Control: determine allocation of resources to messages as they traverse network
  - Channel bandwidth
  - Buffer capacity
  - Control state

- Significant impact on throughput and latency of network

Flow control

- Resource allocation:
  - The routing of a message requires allocation of various resources: the channel (or link), buffers, control state
  - Efficiently allocate network resources:
    - Close to ideal bandwidth,
    - Low and predictable latency
- Contention resolving



Flow control for different switching strategies

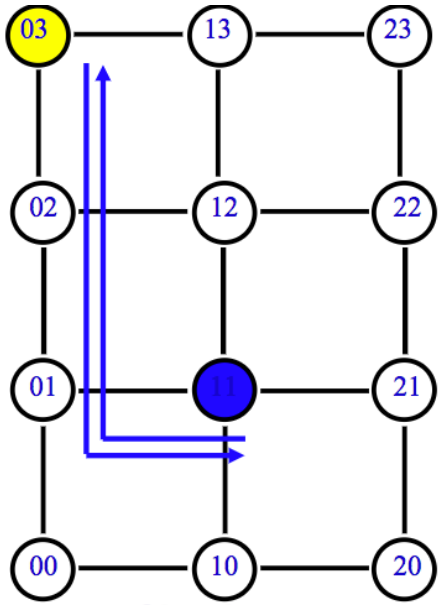
University of BRISTOL

- Different flow control techniques based on switching granularity
  - Circuit-switching:
    - operates at the granularity of messages
  - Packet-switching:
    - allocation made to whole packets
    - Flit-based: allocation made on a flit-by-flit basis

Basis Switching strategies

- Circuit Switching: a bufferless flow control
  - Advantage: Easier to make latency guarantees (after circuit reservation)
  - Disadvantage: doesn't scale well with network size

    - Several links are occupied for the duration of the transmitted data, even when no data is being transmitted.

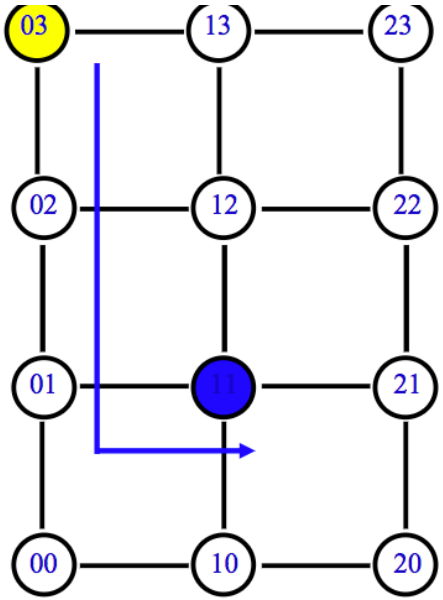


Circuit set-up

Two traversals – latency overhead

Waste of bandwidth

Request packet can be buffered



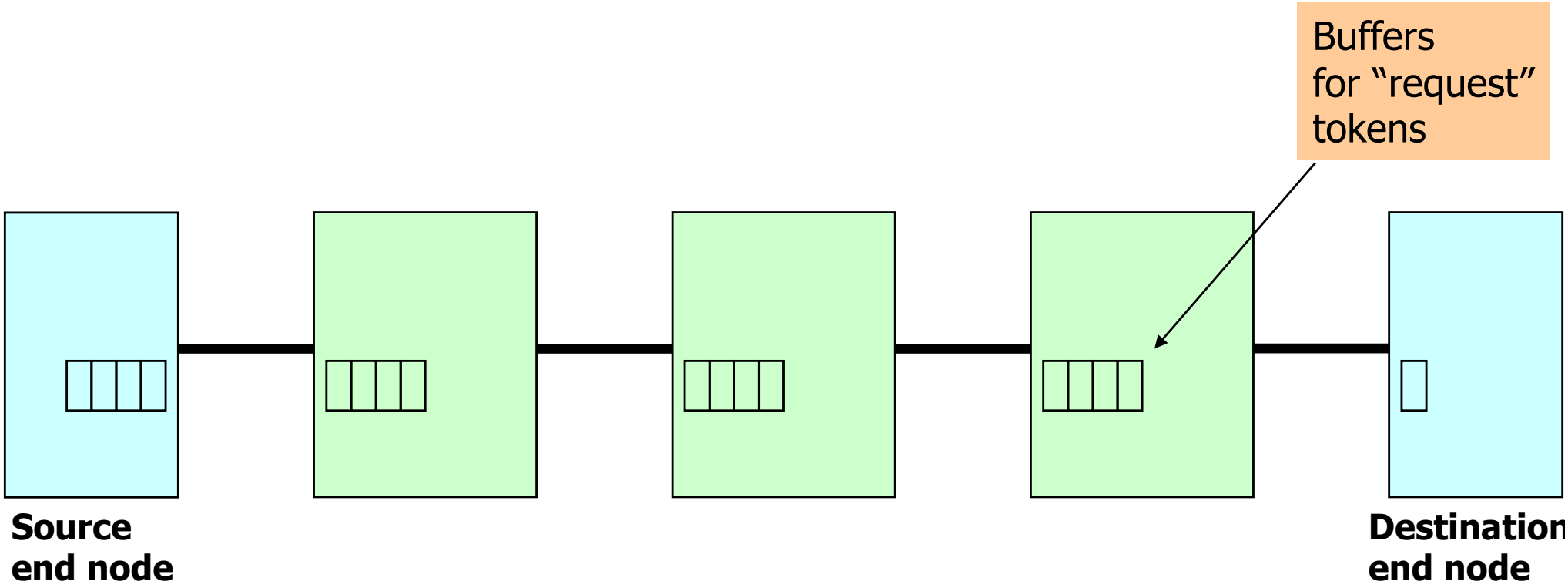
Circuit utilization

Third traversal – latency overhead

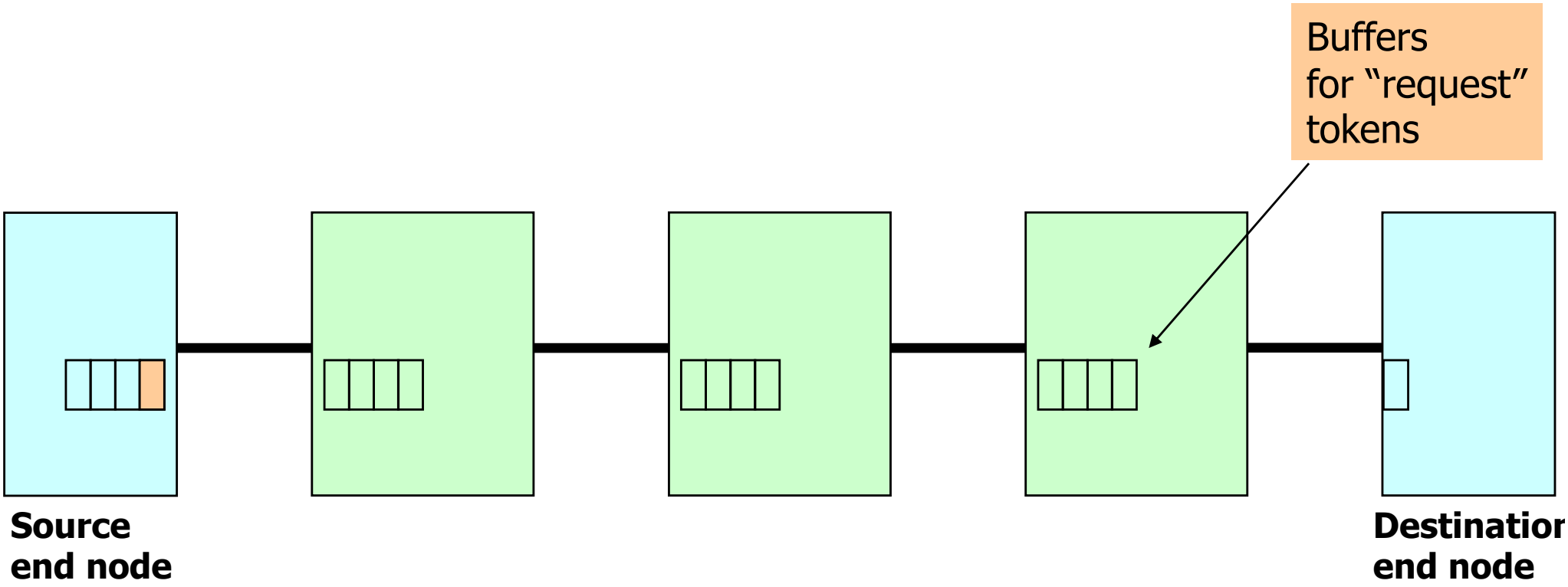
Contention-free transmission

Poor resource utilization

Circuit Switching

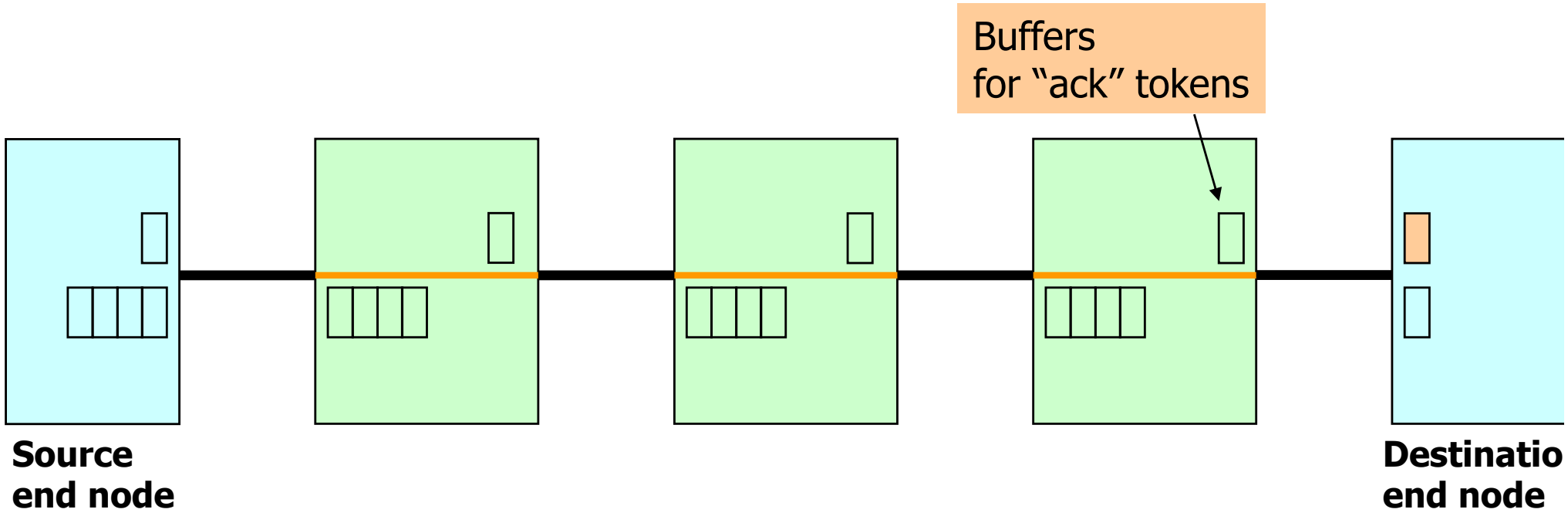


Circuit Switching



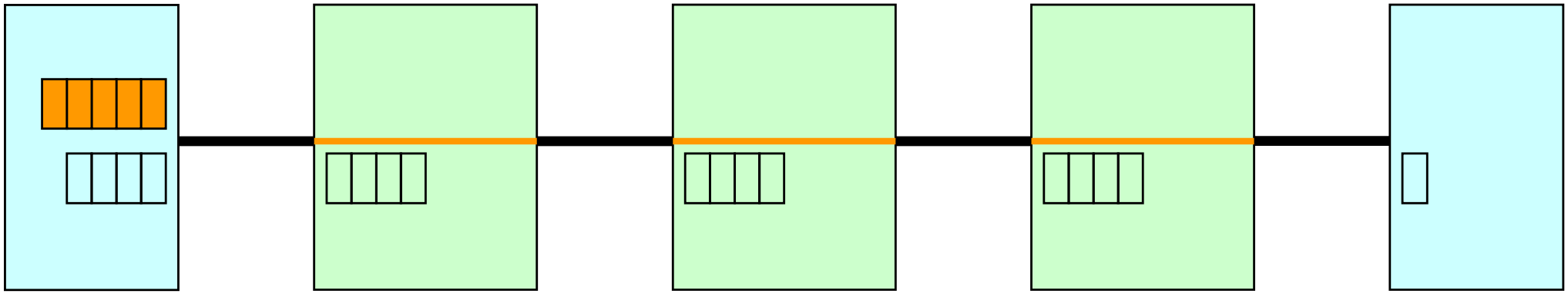
Request for circuit establishment
(routing and arbitration is performed during this step)

Circuit Switching



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

Circuit Switching

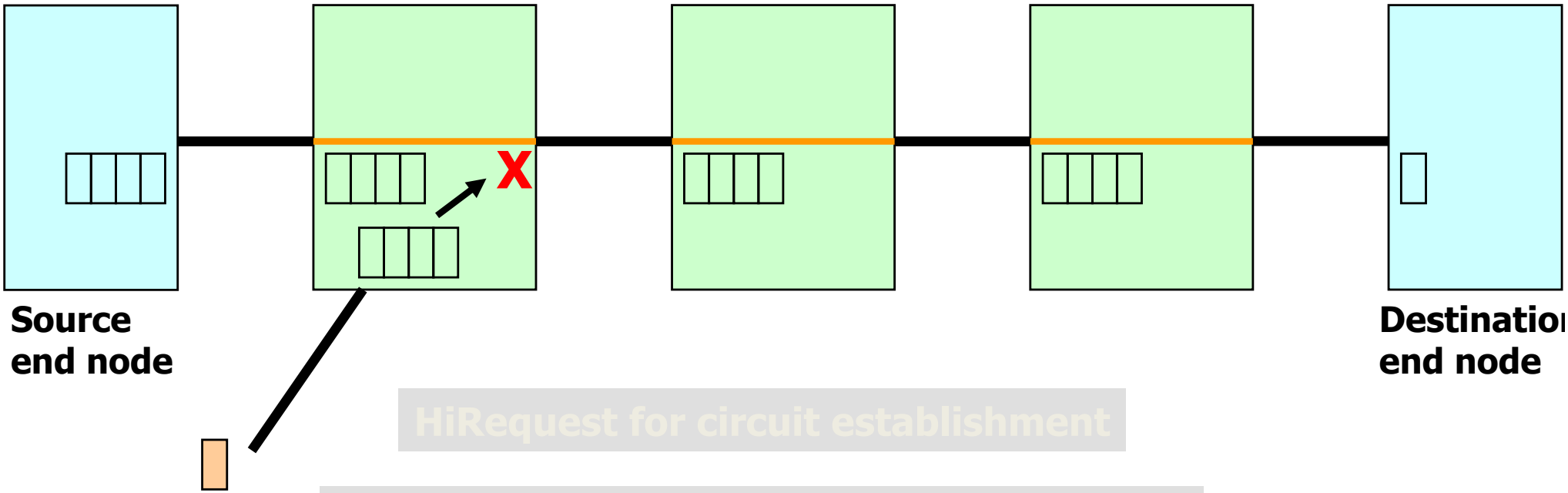


Request for circuit establishment

Acknowledgment and circuit establishment

Packet transport
(neither routing nor arbitration is required)

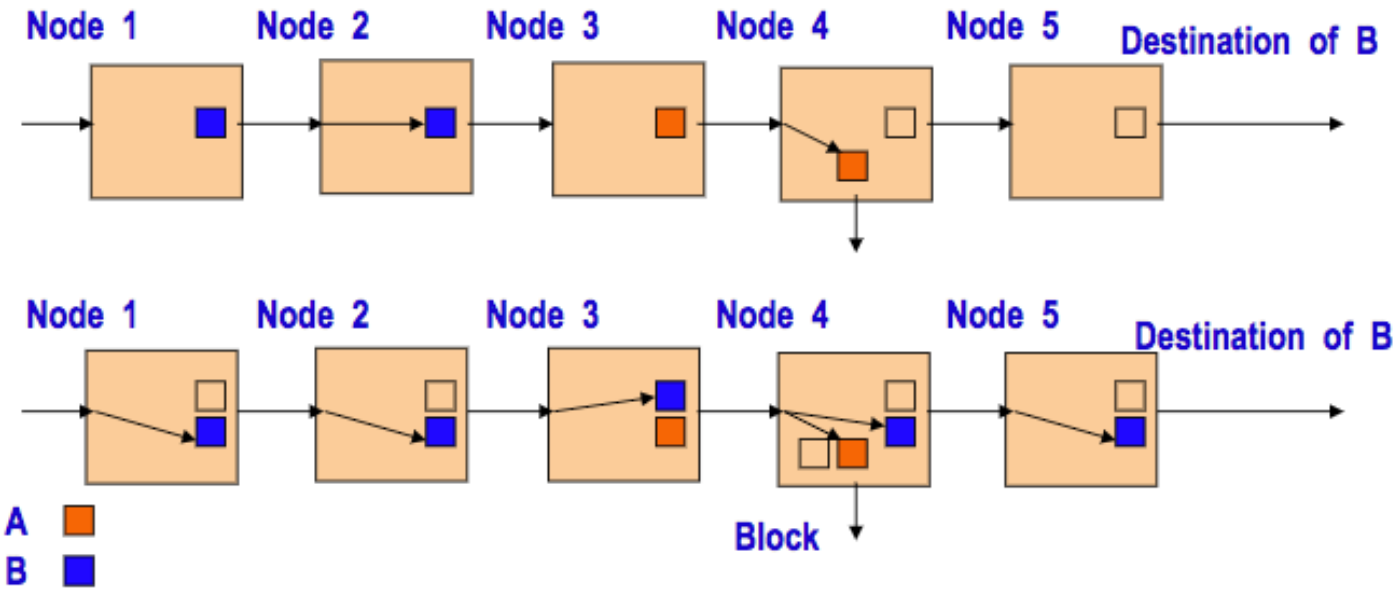
Circuit Switching



High contention, low utilization (ρ) → *low throughput*

Virtual circuit switching

- Multiple virtual circuits (channels) multiplexed on a single physical link.
- Virtual-channel flow control decouples the allocation of channel state from channel bandwidth.
  - Allocate one buffer per virtual link
    - can be expensive due to the large number of shared buffers
  - Allocate one buffer per physical link
    - uses time division multiplexing (TDM) to statically schedule usage
    - less expensive routers



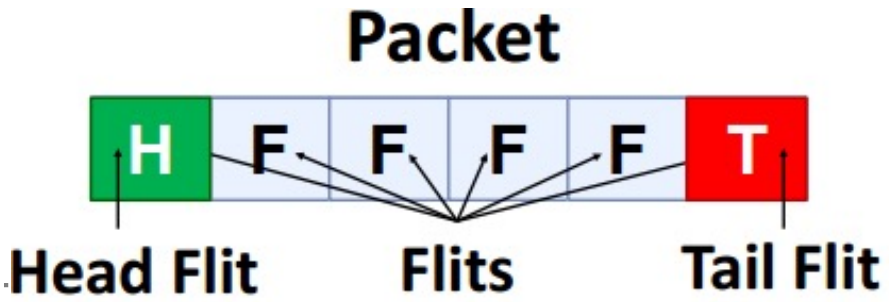
Packet switching

- Routing, arbitration, switching is performed on a per-packet basis
- Sharing of network link bandwidth is done on a per-packet basis

- Buffered flow control
  - Packets are transmitted from source and make their way independently to receivers
    - possibly along different routes and with different delays
    - QoS guarantees are harder to make
  - Zero start up time, followed by a variable delay due to contention in routers along packet path

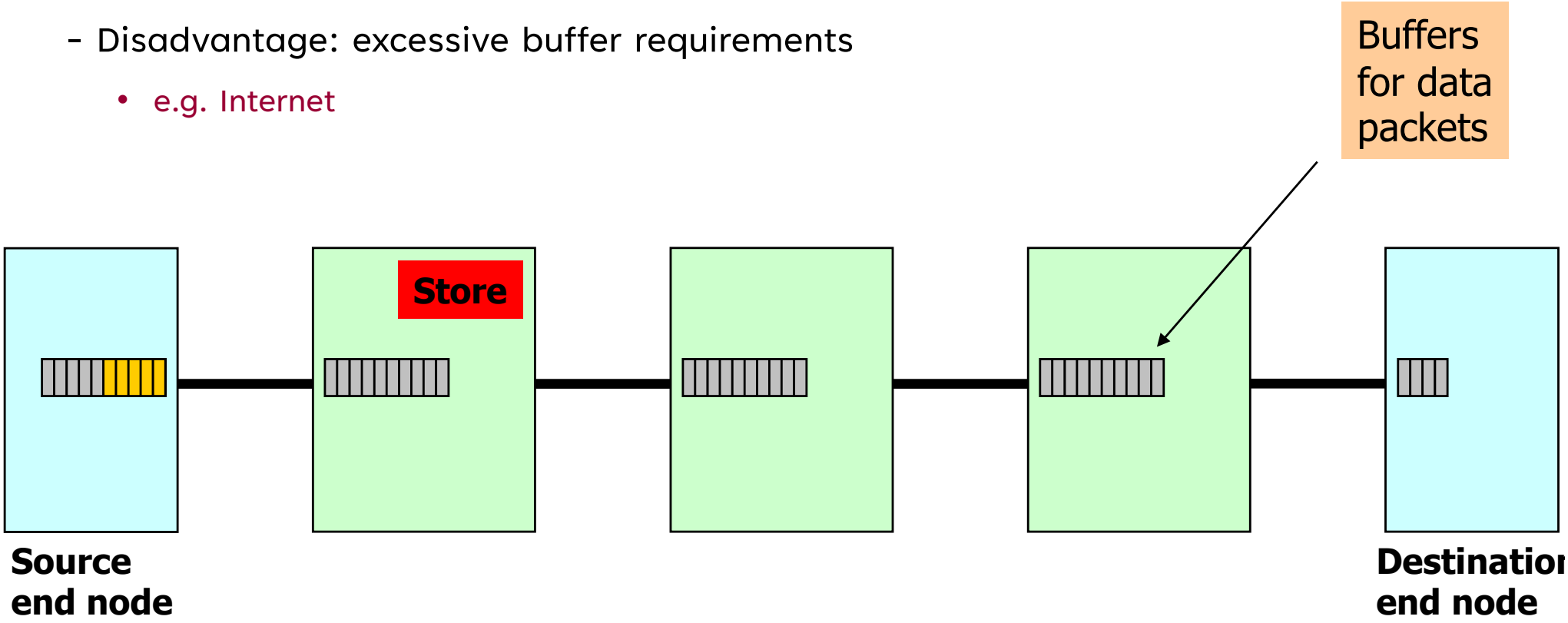
Packet switching: Packets/Flits

- A message is broken into multiple packets (each packet has header information that allows the receiver to re-construct the original message)
- A packet may itself be broken into flits – flits do not contain additional headers
- Two packets can follow different paths to the destination Flits are always ordered and follow the same path
- Such an architecture allows the use of a large packet size (low header overhead) and yet allows fine-grained resource allocation on a per-flit basis



Store and Forward (SF) flow control

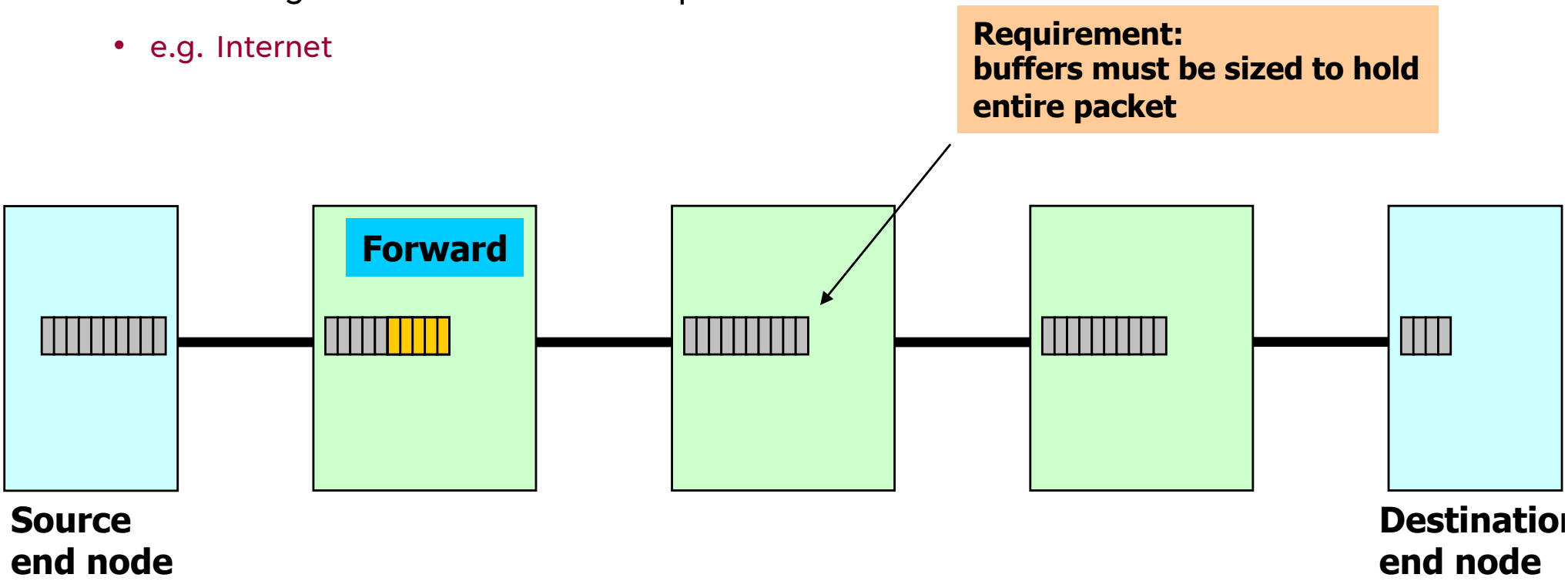
- Packet is sent from one router to the next only if the receiving router has buffer space for entire packet
- Buffer size in the router is at least equal to the size of a packet
- Disadvantage: excessive buffer requirements
  - e.g. Internet



Packets are completely stored before any portion is forwarded

Store and Forward (SF) Flow Control

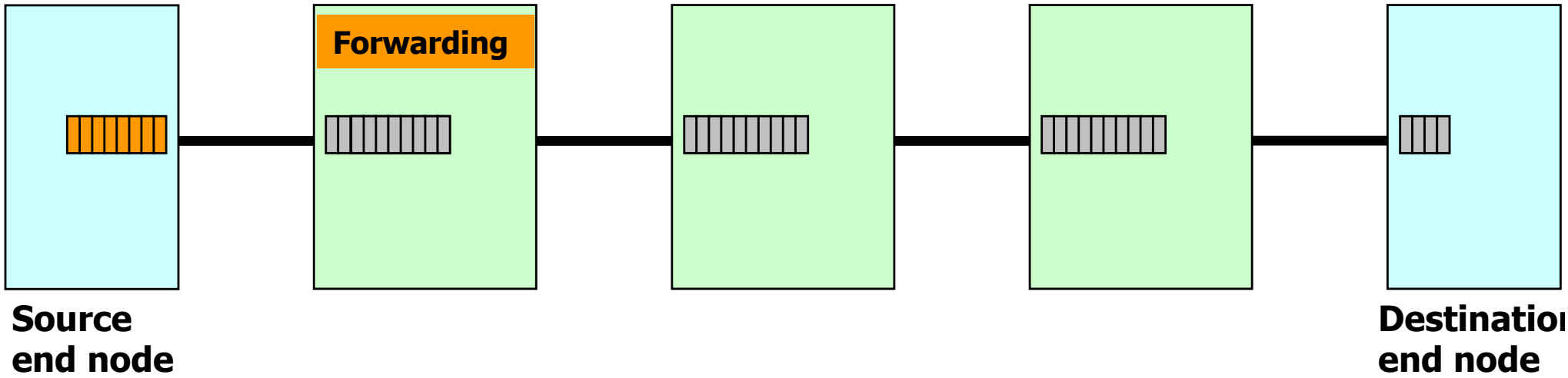
- Packet is sent from one router to the next only if the receiving router has buffer space for entire packet
- Buffer size in the router is at least equal to the size of a packet
- Disadvantage: excessive buffer requirements
  - e.g. Internet



Packets are completely stored before any portion is forwarded

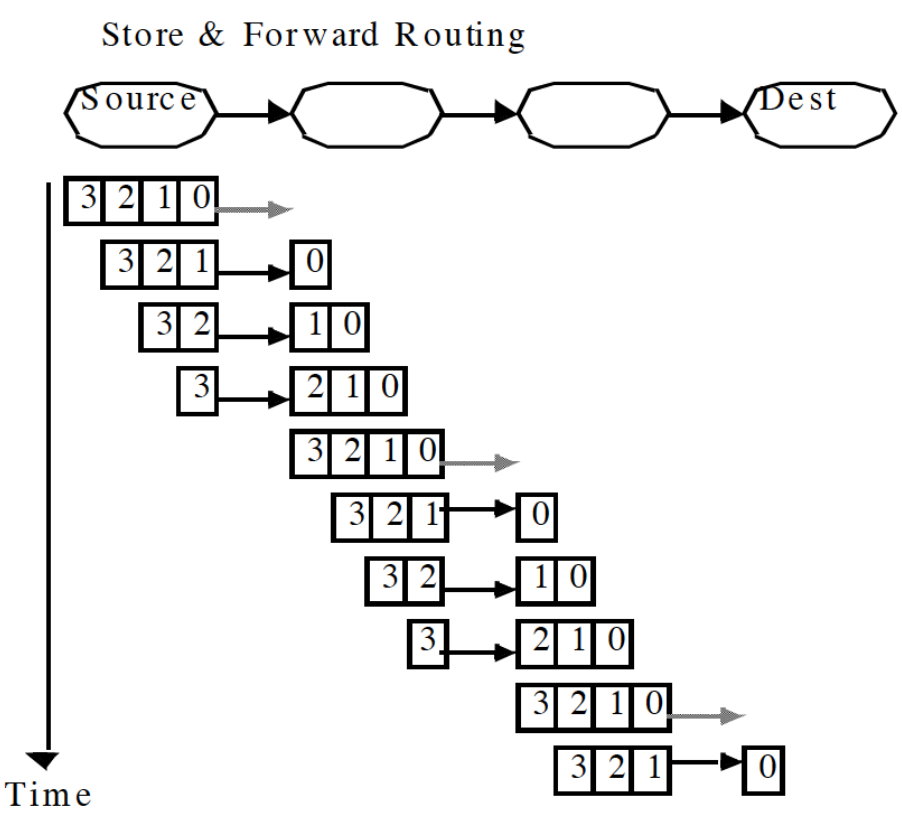
Cut-Through Flow Control

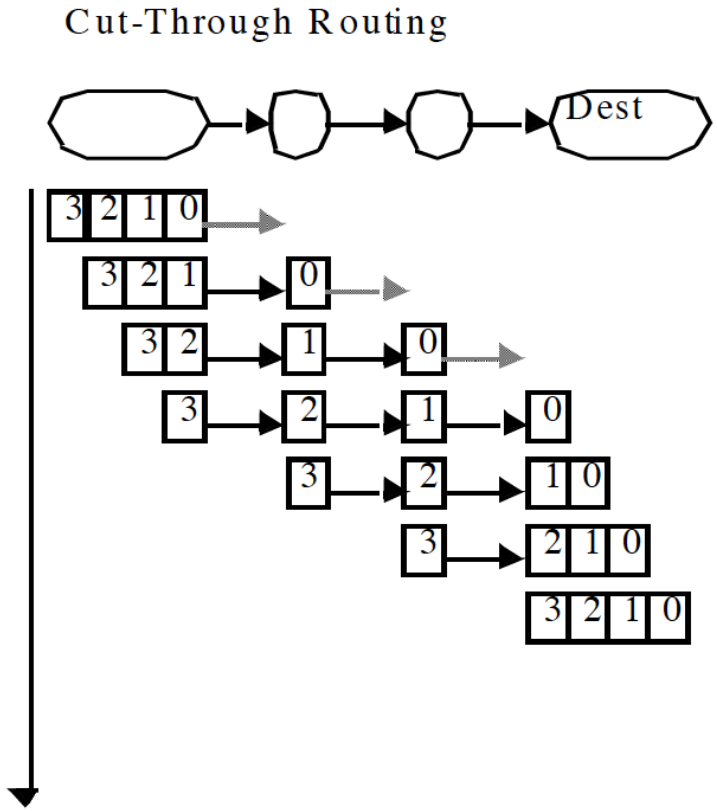
- Forwards first flit of a packet as soon as space for the entire packet is available in the next router
- Reduces router latency over SF switching
- Same buffering requirements as SF switching



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

SF vs. Cut-Through Switching

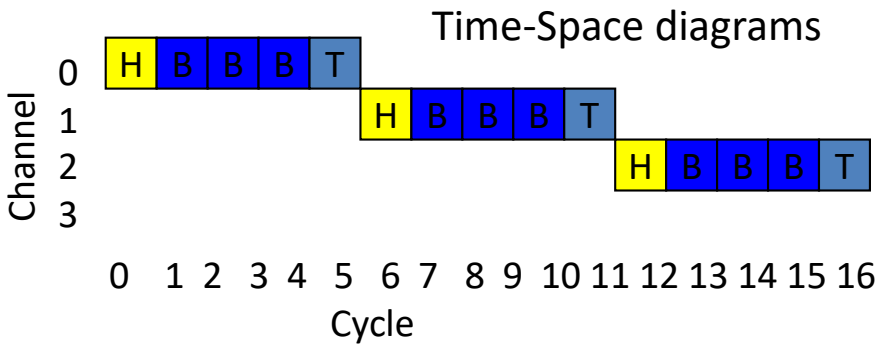




Latency of Packet-buffer flow control

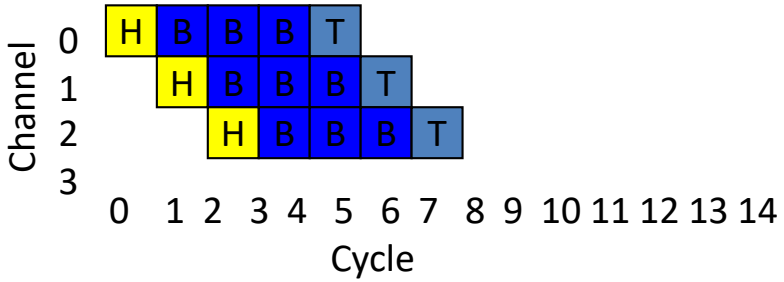
- Store and forward
  - Serialization latency is experienced at each hop

$$T_0 = H(t_r + \frac{L}{b})$$



- Cut-through flow control
  - One serialization latency

$$T_0 = Ht_r + \frac{L}{b}$$



10 GbE Store-and-forward switch: 5-35 microseconds
10 GbE cut-through switch: 300-500 nanoseconds

Characteristic of cut-through flow control

University of BRISTOL

Advantages

- High channel utilisation using buffers to decouple channel allocation
- Very low latency

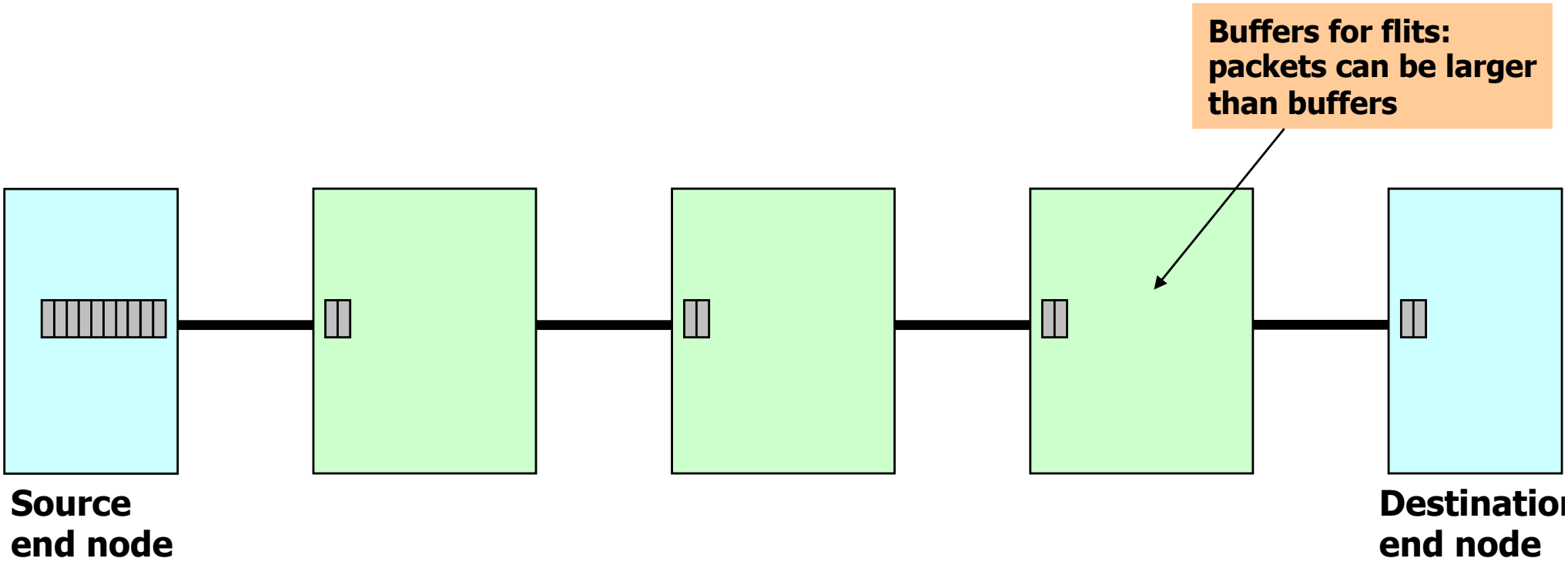
Disadvantages

- Inefficient use of buffer storage
- Contention latency is increased due to allocate channels in packets

Open questions: why does Internet choose SF flow control?

Wormhole flow control for packet switching

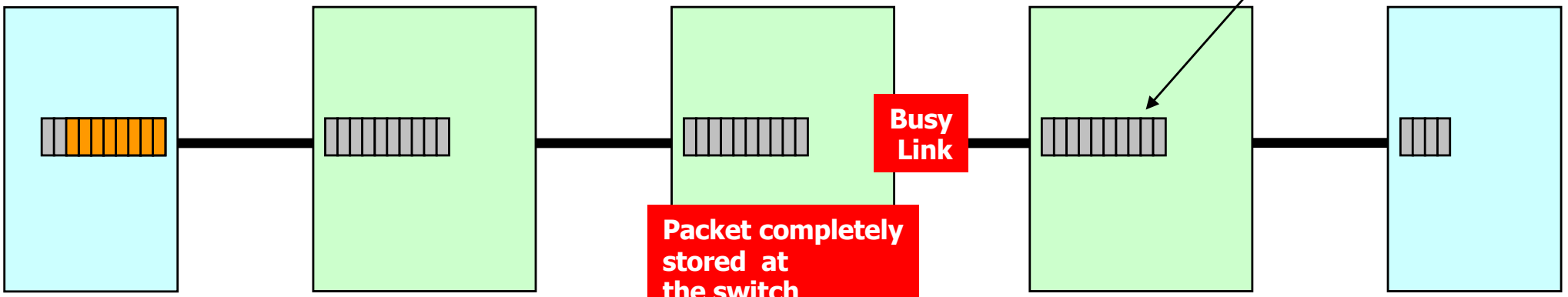
- flit is forwarded to receiving router if space exists for that flit
- pipeline the hops: switch examines the header, decides where to send the message, and then starts forwarding it immediately



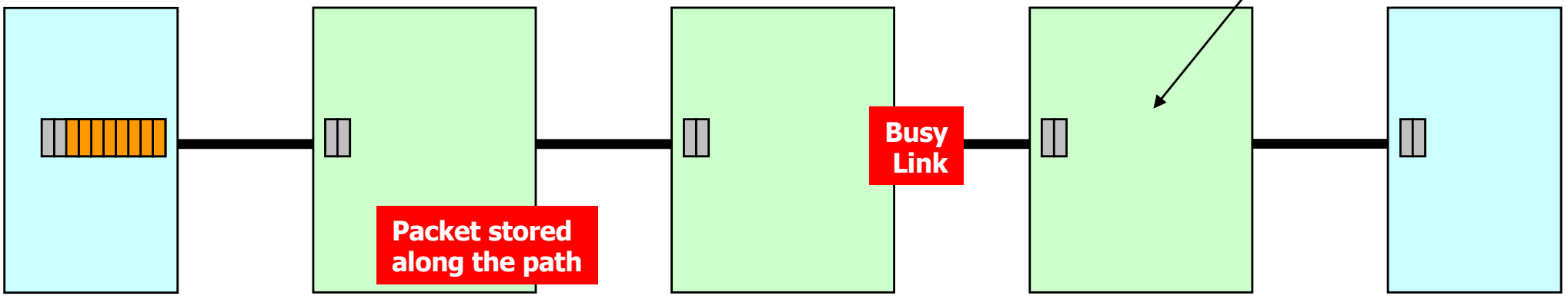
Cut-through vs. Wormhole

University of BRISTOL

Cut-through



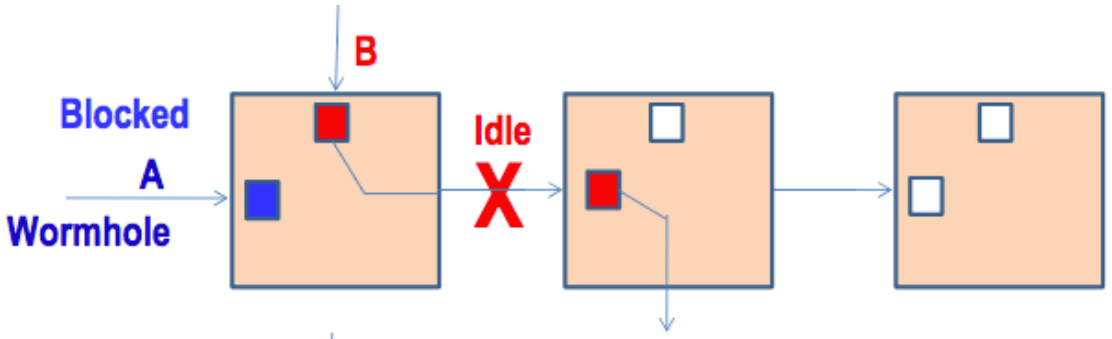
Wormhole



Maximizing sharing of link BW increases ρ (i.e., ρ_s)

Flit-Buffer Flow Control 1: Wormhole Flow Control

- Wormhole Flow Control: just like cut-through, but with buffers allocated per flit (not packet)
- A head flit must acquire three resources at the next switch before being forwarded:
  - channel control state (virtual channel, one per input port)
  - one flit buffer
  - one flit of channel bandwidth
- The other flits adopt the same virtual channel as the head
- Consumes much less buff space than cut-through routing – does not improve channel utilization as another packet cannot cut in (only one VC per input port)
- However, ?



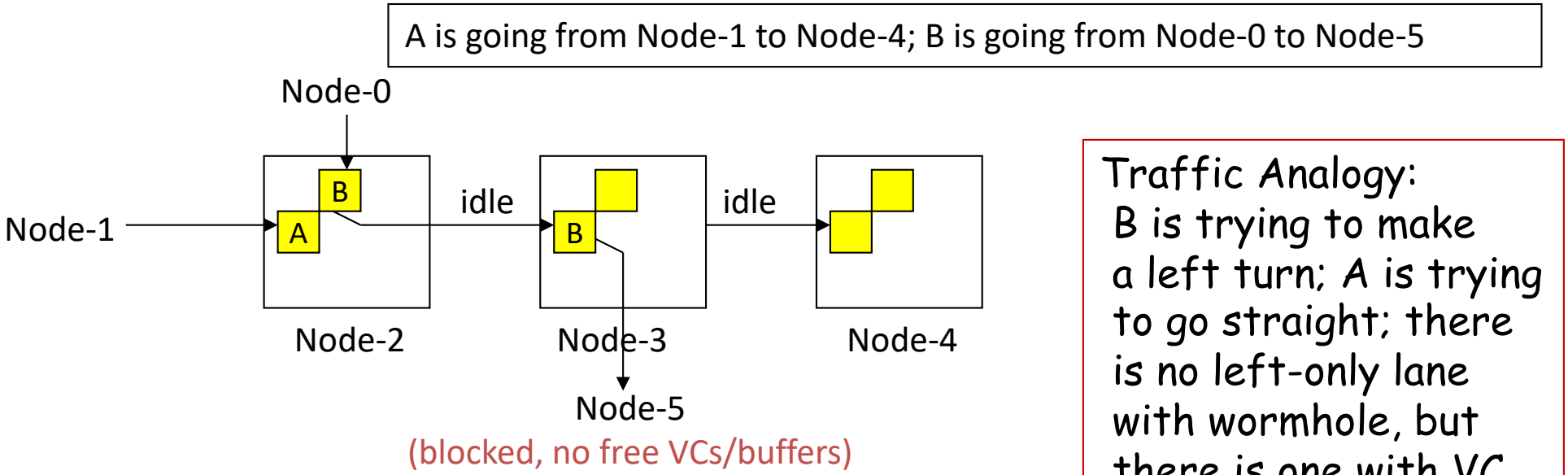
Virtual Channel Flow Control

University of BRISTOL

- Each switch has multiple virtual channels per phys. channel
- Each virtual channel keeps track of the output channel assigned to the head, and pointers to buffered packets
- A head flit must allocate the same three resources in the next switch before being forwarded
- By having multiple virtual channels per physical channel, two different packets are allowed to utilize the channel and not waste the resource when one packet is idle

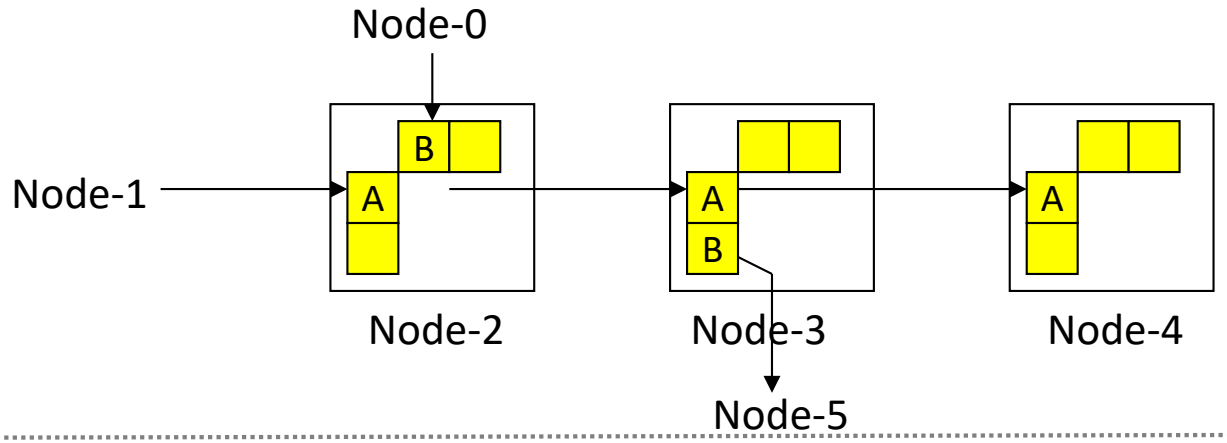
Example

- Wormhole:

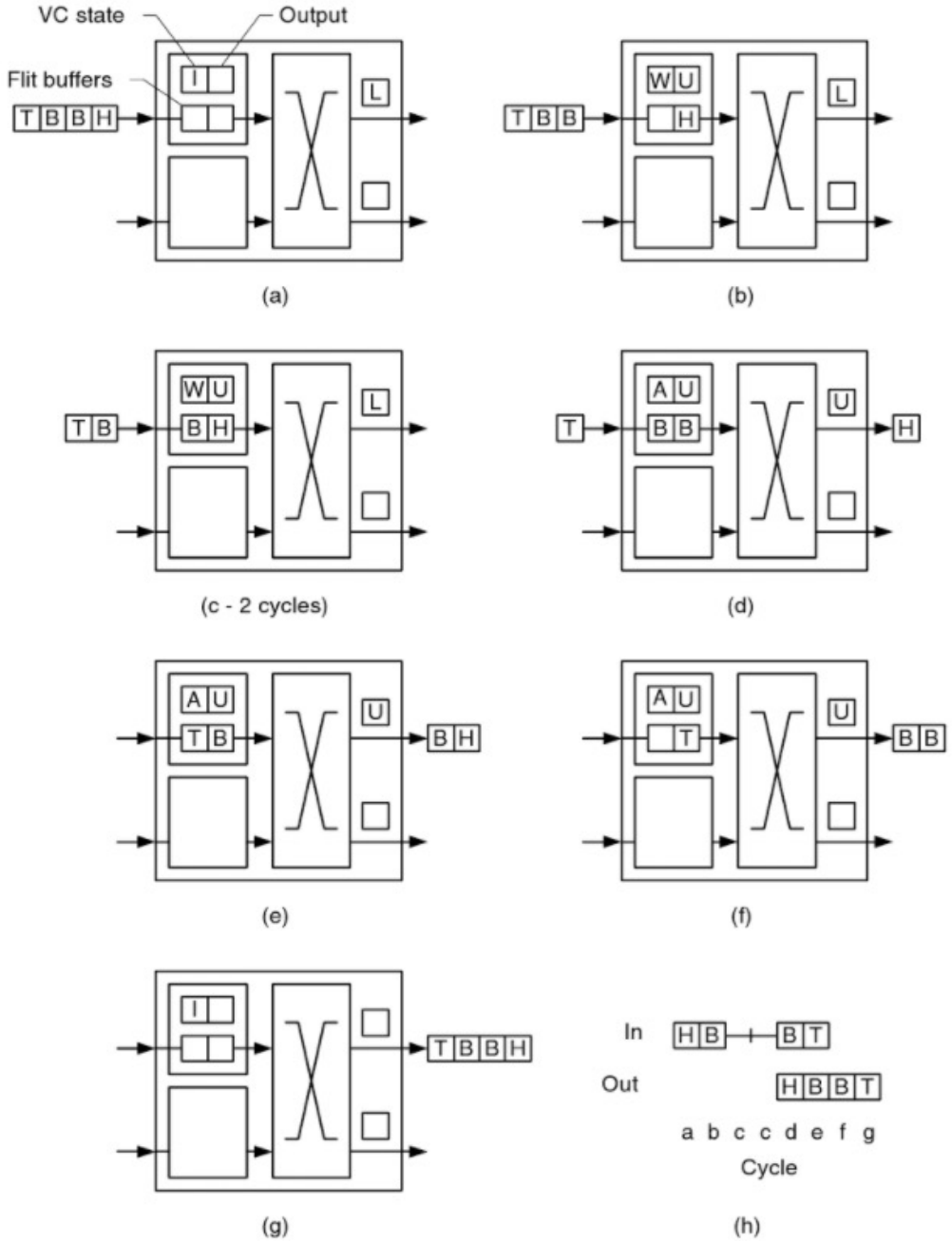


Traffic Analogy:
B is trying to make a left turn; A is trying to go straight; there is no left-only lane with wormhole, but there is one with VC

- Virtual channel:



Wormhole flow control



Summary

- Bufferless flow control for circuit switching
  - Virtual Channel
- Buffered flow control for packet switching
  - Storage and Forward (SF)
  - Cut-through
  - Wormhole

Lab Introduction

Introduction of Lab

- Aim and Goals
  - Understand the introduced concepts, including topologies, routing algorithms, and flow control in DCN
  - Explore a simple NoC simulator (BookSim) to evaluate different network topologies
- Lab Note:
  https://seis.bristol.ac.uk/~sy13201/DCN/Lab%20Note.html
- Assignment 1
  - Explore latency in design of interconnection networks
  - Compare Mesh and Torus network topologies
- Assignment 2
  - Implement your own network topologies and evaluate the performance

Deadline and Marking

Report with results and discussion: 20% for the unit mark

- Assignment 1 – Use BookSim to evaluate network topologies

- Assignment 2 – Design your own network topology and evaluate it.

Deadline: **May 5th, 2022, 13:00**, submit on BB.

Assignment 1

Use different traffic to evaluate 8-ary 2-cube Mesh and Torus network topologies.

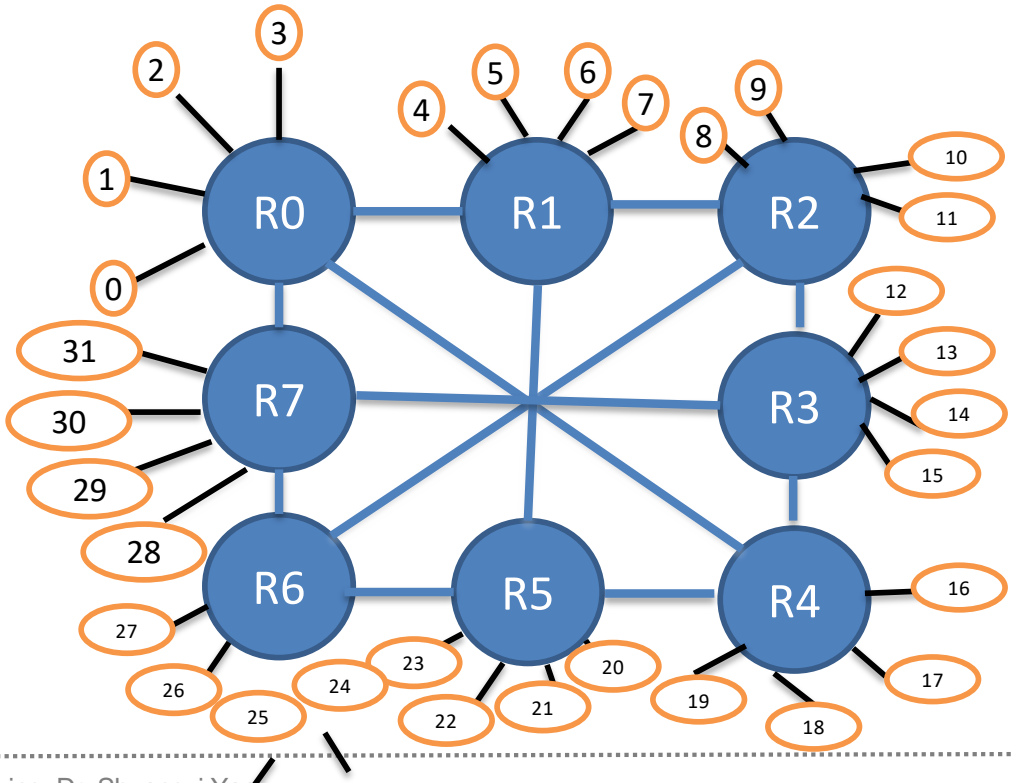
- Evaluate the latency and other parameters
- Discuss the possible reasons that cause the performance difference between two topologies

Assignment 2

Design your own network topology and evaluate it.

8 routers, 32 nodes, with bidirection links between all routers (**full-connected network**).

- Evaluate the latency and other parameters
- Discuss the possible reasons that cause the performance difference between two topologies



Reports

- Around 2-3 pages.
- Chapter 1: Assignment 1
  - Results and discussion.
  - Write in a clear and concise way.
- Chapter 2: Assignment 2
  - List the metrics you plan to evaluate the performance for your own topology
  - Report the performance and discuss it.