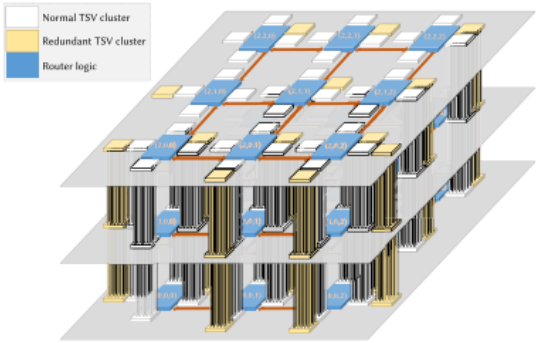
Intensive Lectures on

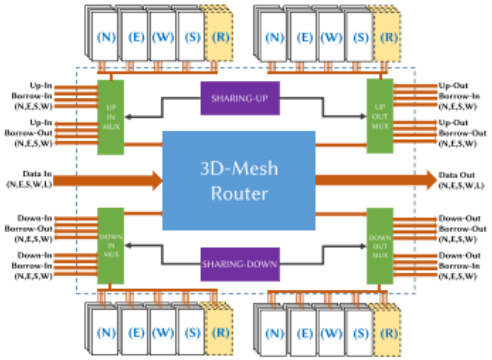
Network-on-Chip

Abderazek Ben Abdallah

The University of Aizu

E-mail: benab@u-aizu.ac.jp





Hong Kong University of Science and Technology, Hong Kong, March 2013

Part 1

Application requirements

NoC: A paradigm shift in VLSI Design

Critical problems addressed by NoC

Traffic abstractions

Data abstraction

Network delay modeling

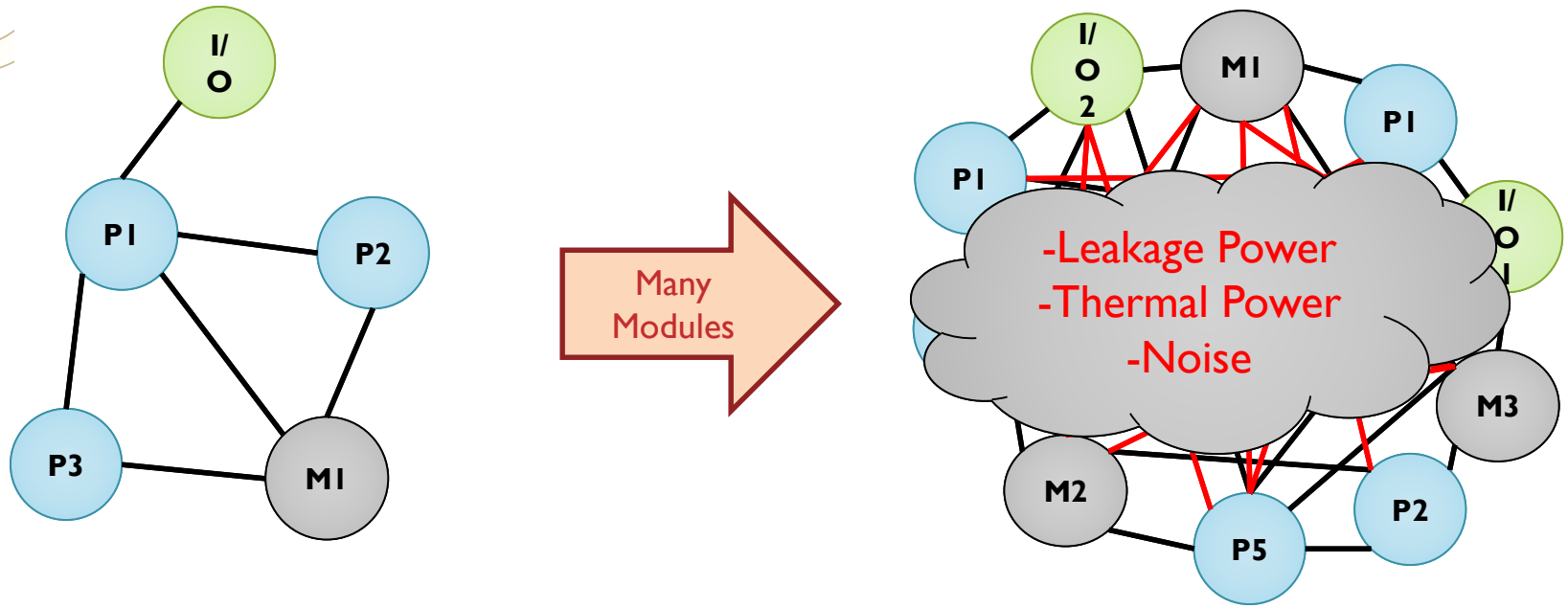
Wire Delay vs. Logic Delay

Operation	Delay (.13mico)	Delay (.05micro)
32-bit ALU Operation	650ps	250ps
32-bit Register read	325ps	125ps
Read 32-bit from 8KB RAM	780ps	300ps
Transfer 32-bit across chip (10mm)	**1400ps**	**2300ps**
Transfer 32-bit across chip (200mm)	2800ps	4600ps

2:1 global on-chip communication to operation delay 9:1 in 2010

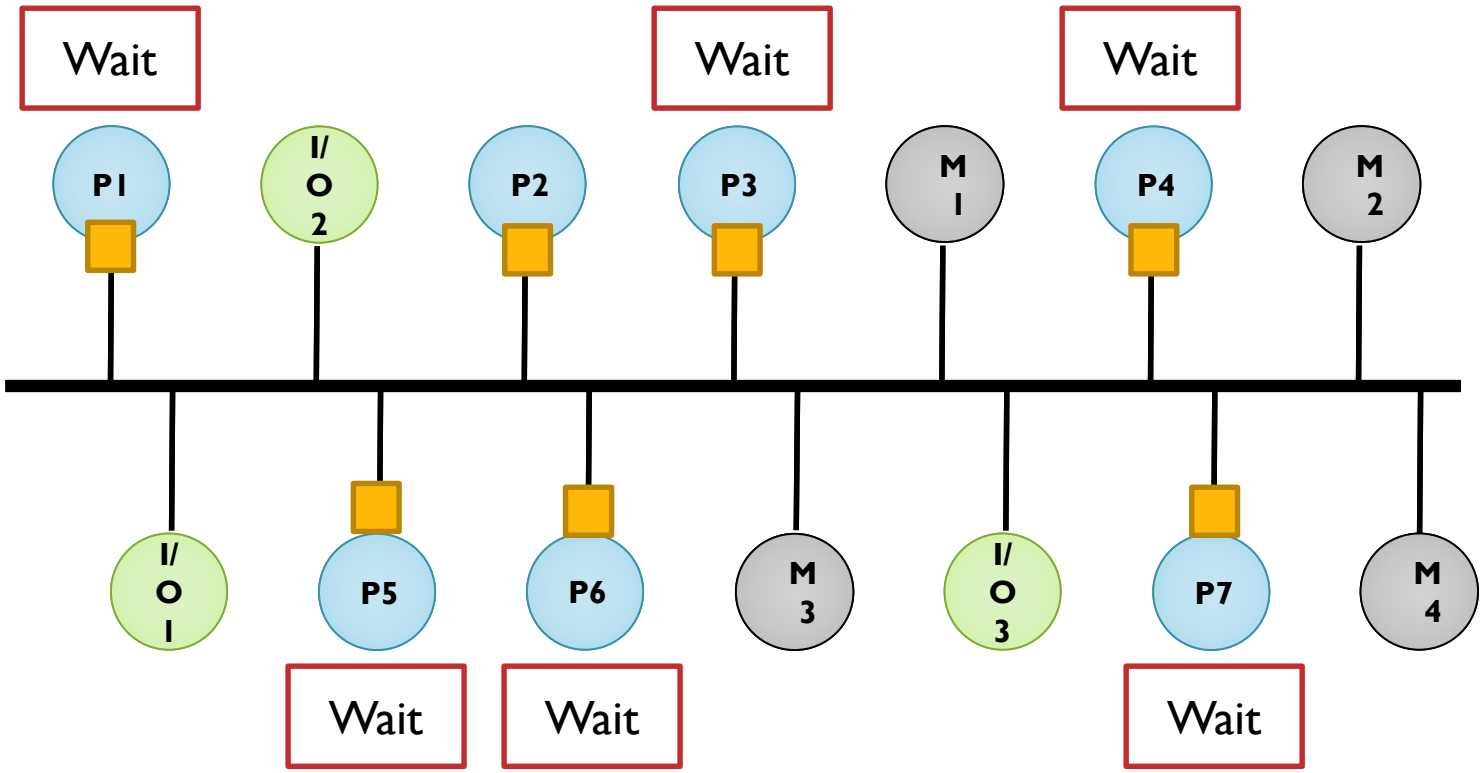
Ref: W.J. Dally HPCA Panel presentation 2002

On-chip Interconnection Types



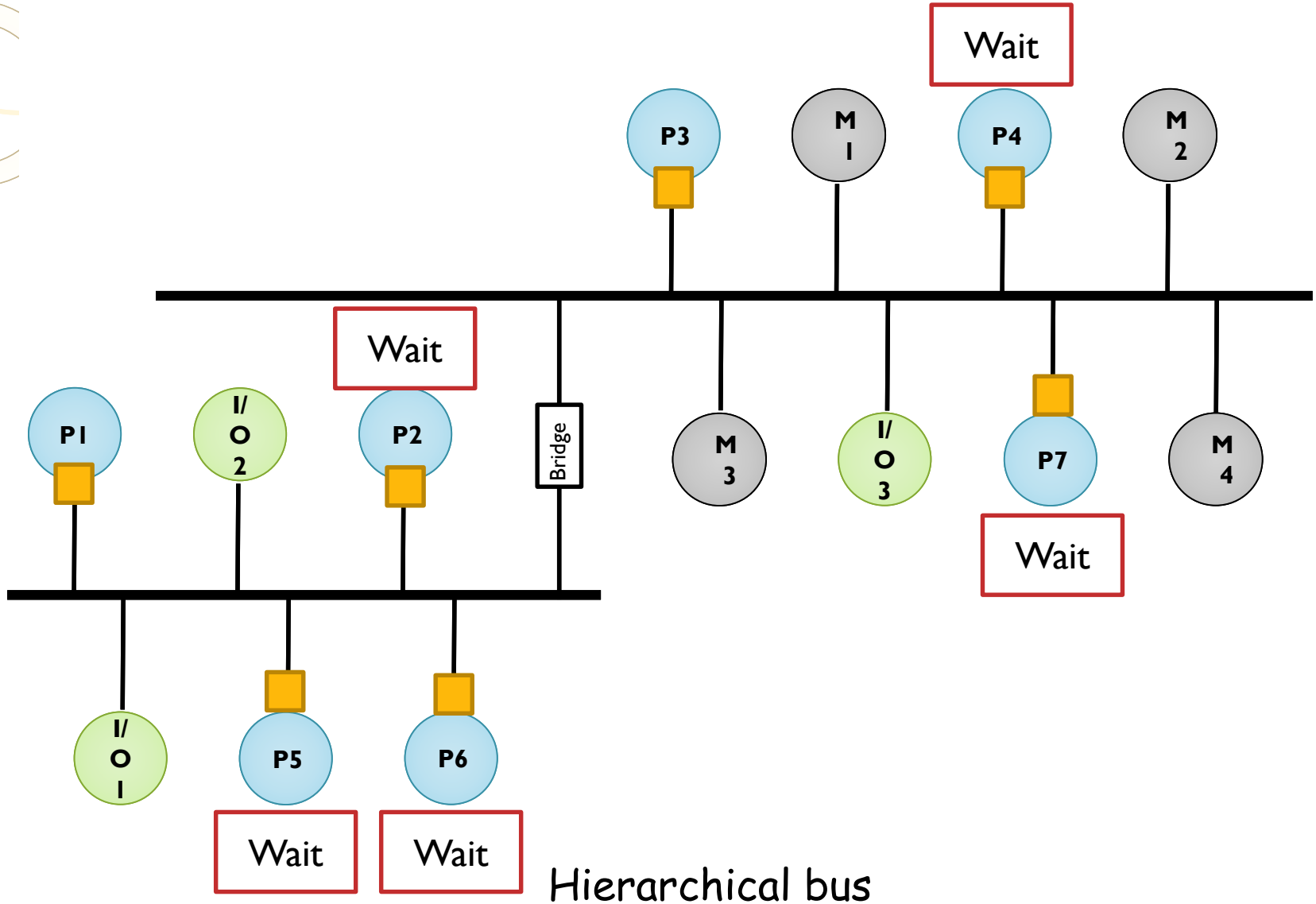
Point-to-Point

On-chip Interconnection Types



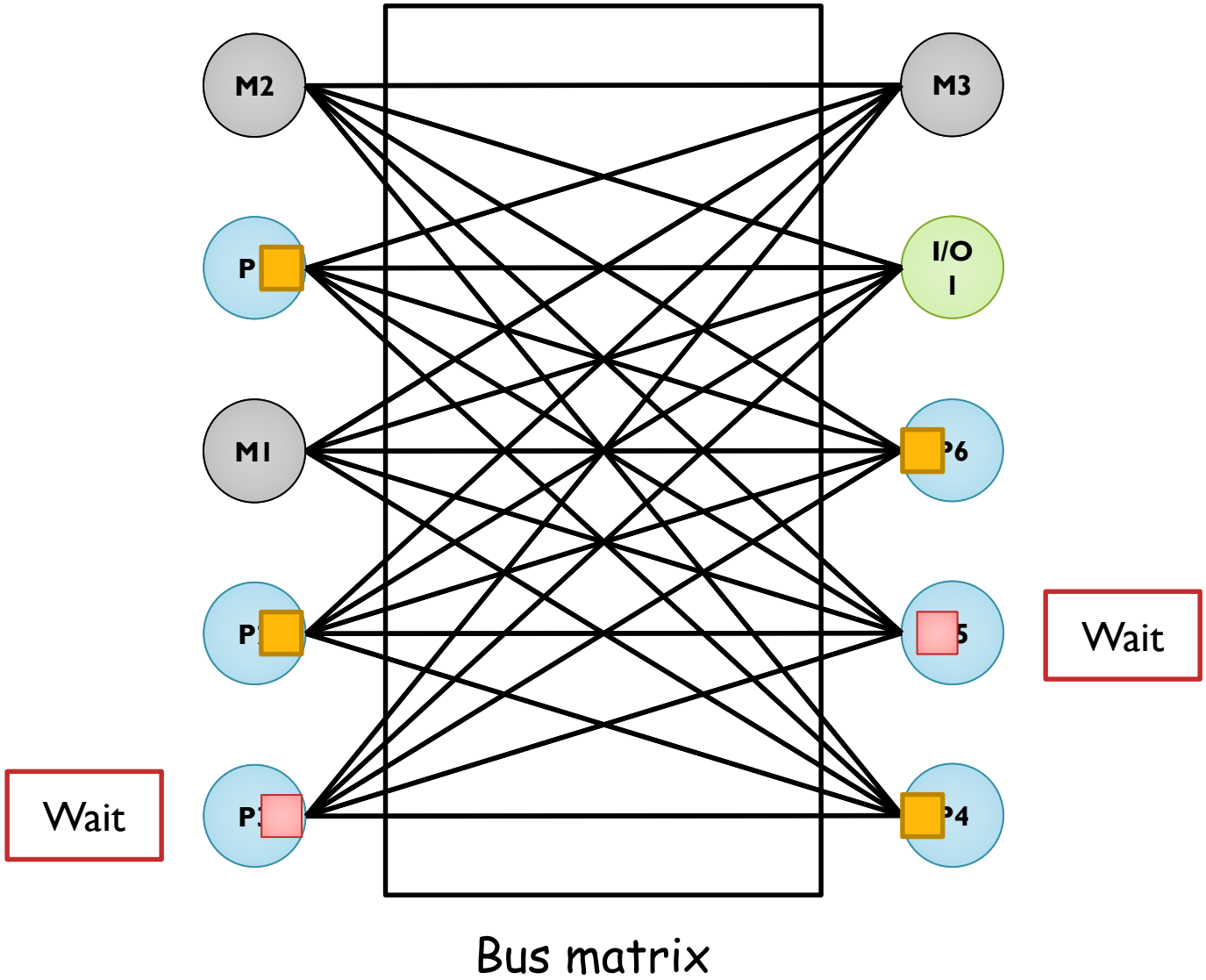
Shared bus

On-chip Interconnection Types

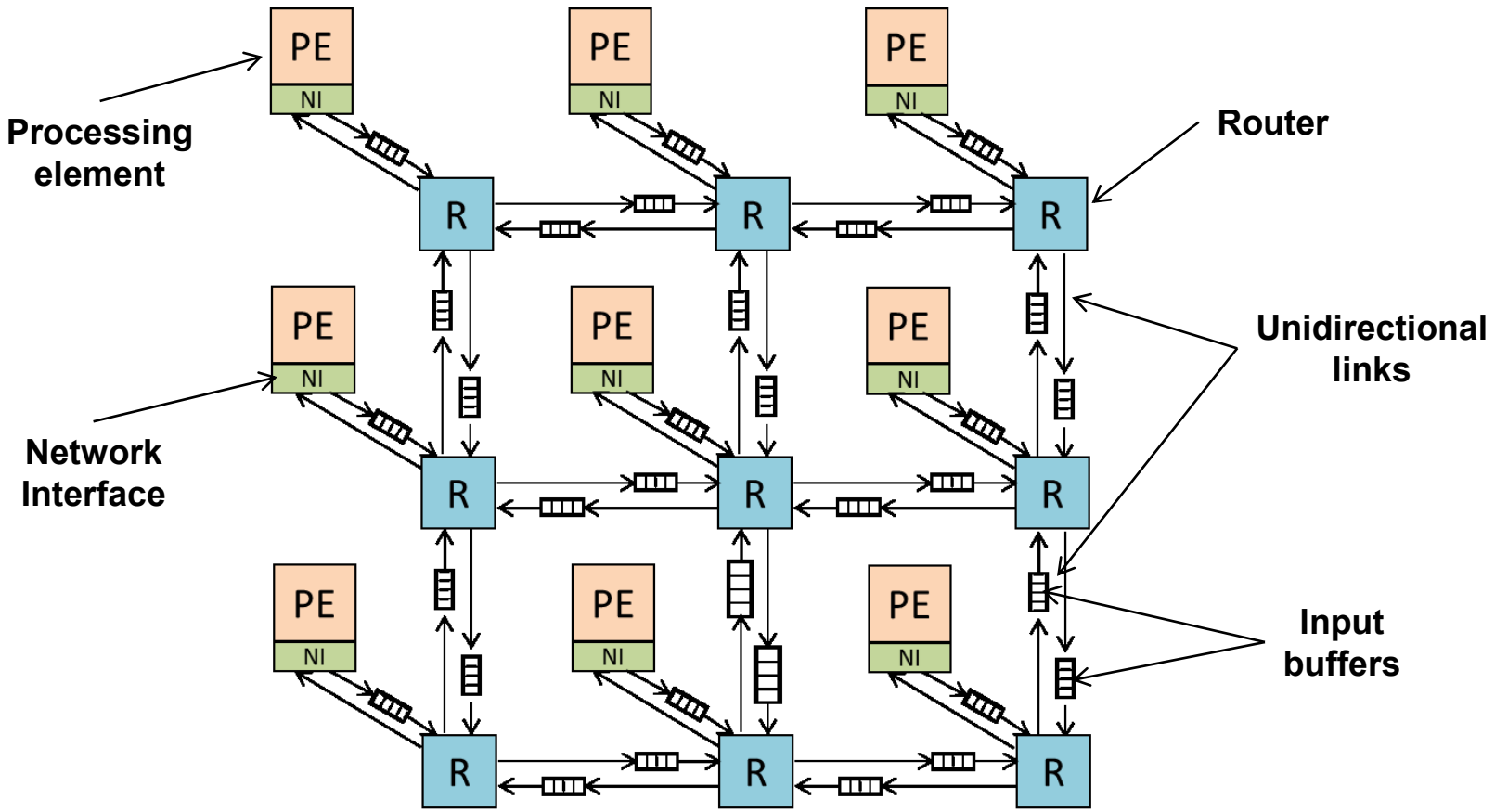


Hierarchical bus

On-chip Interconnection Types

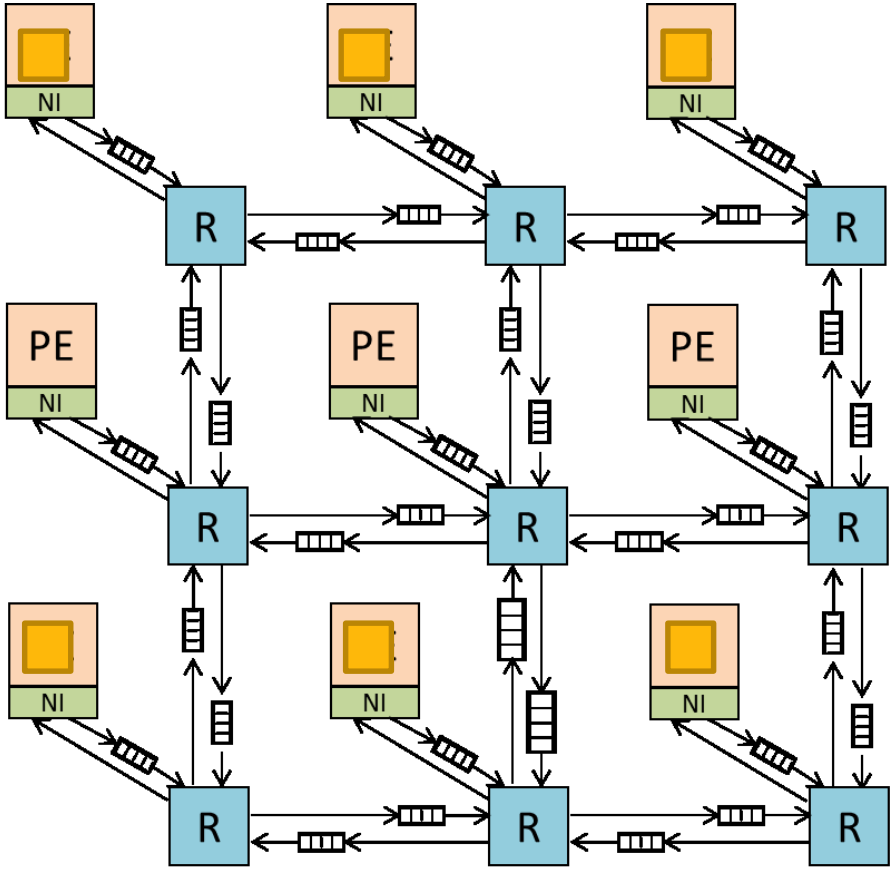


On-chip Interconnection Types



Network-on-Chip -> our main topic in this lecture.

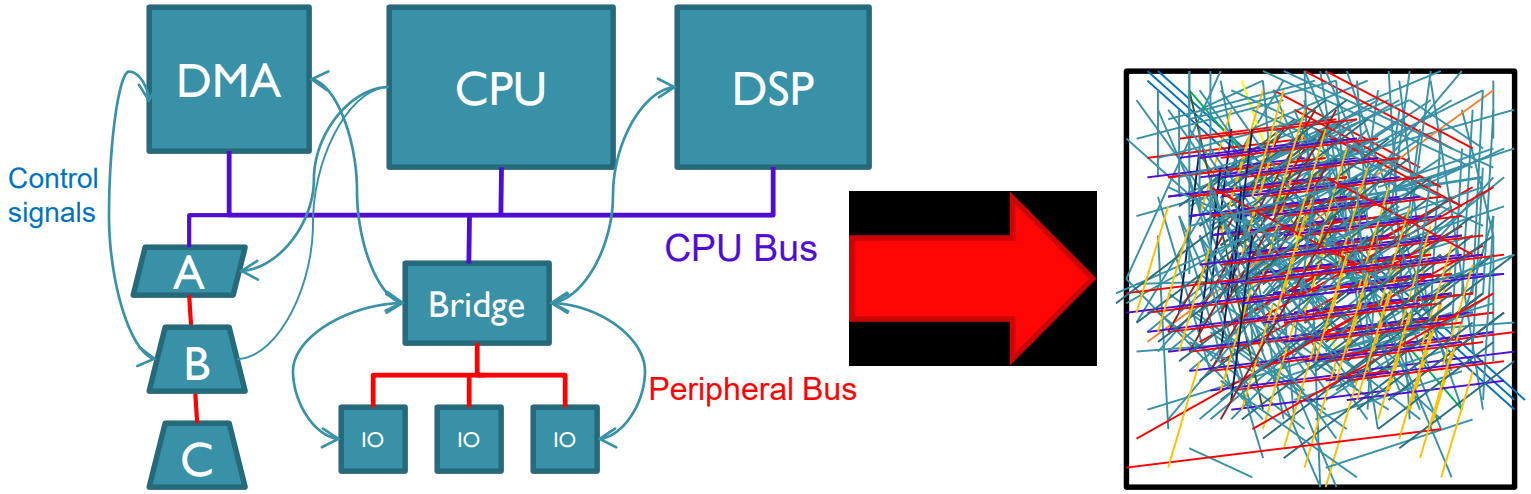
On-chip Interconnection Types



Network-on-Chip -> our main topic in this lecture

Traditional SoC Nightmare

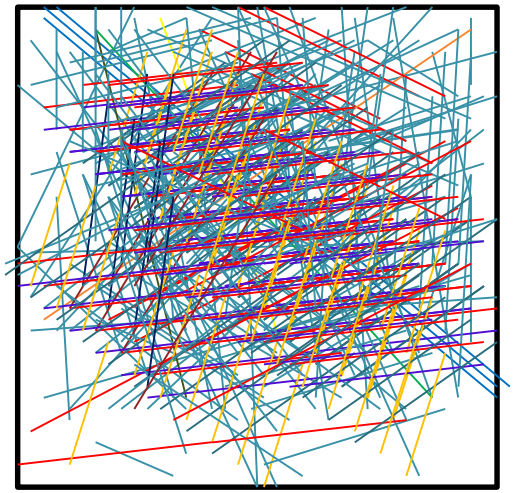
- Variety of dedicated interfaces
- Design and verification complexity
- Unpredictable performance
- Many underutilized wires

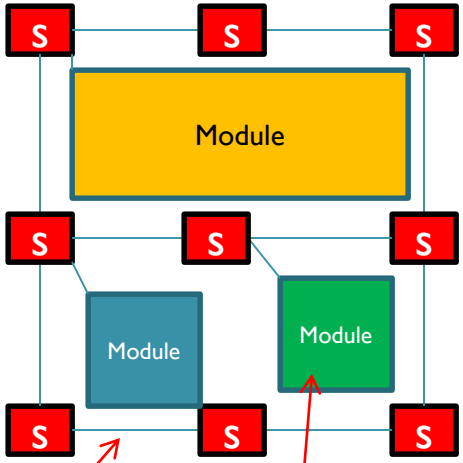


NoC: A paradigm Shift in VLSI

From: Dedicated signal wires

To: Shared network





Point-To-point Link

Computing Module

Network switch

Part II: NoC Building Blocks

Topology

Routing Algorithms

Routing Mechanisms

Control Flow

Network Interface

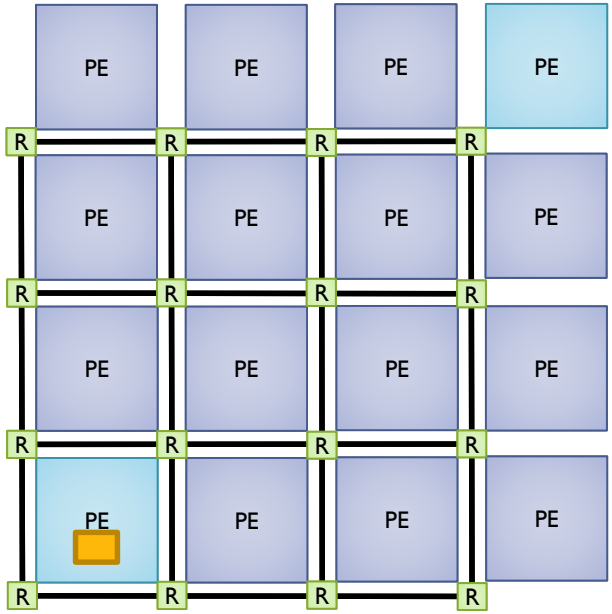
Router Architecture

NoC Topology

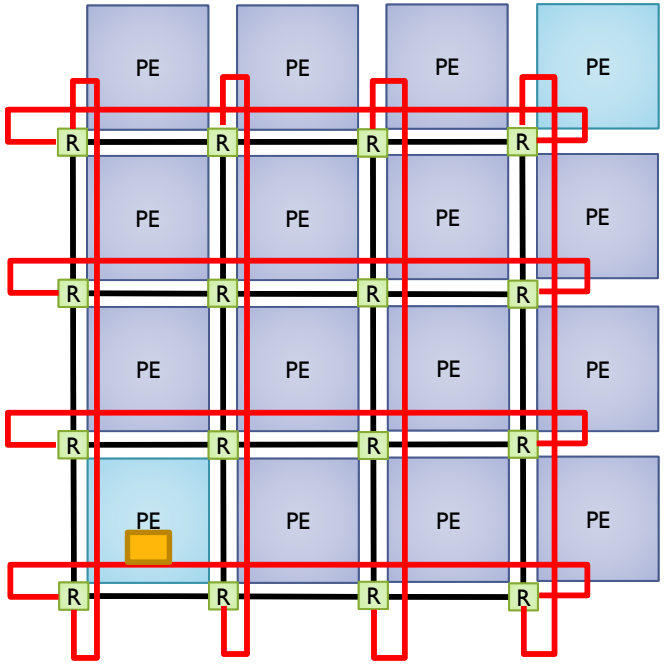
NoC topology is the connection map between PEs.

- Mainly adopted from large-scale networks and parallel computing
- A good topology allows to fulfill the requirements of the traffic at reasonable costs
- Topology classifications:
  1. Direct topologies
  2. Indirect topologies

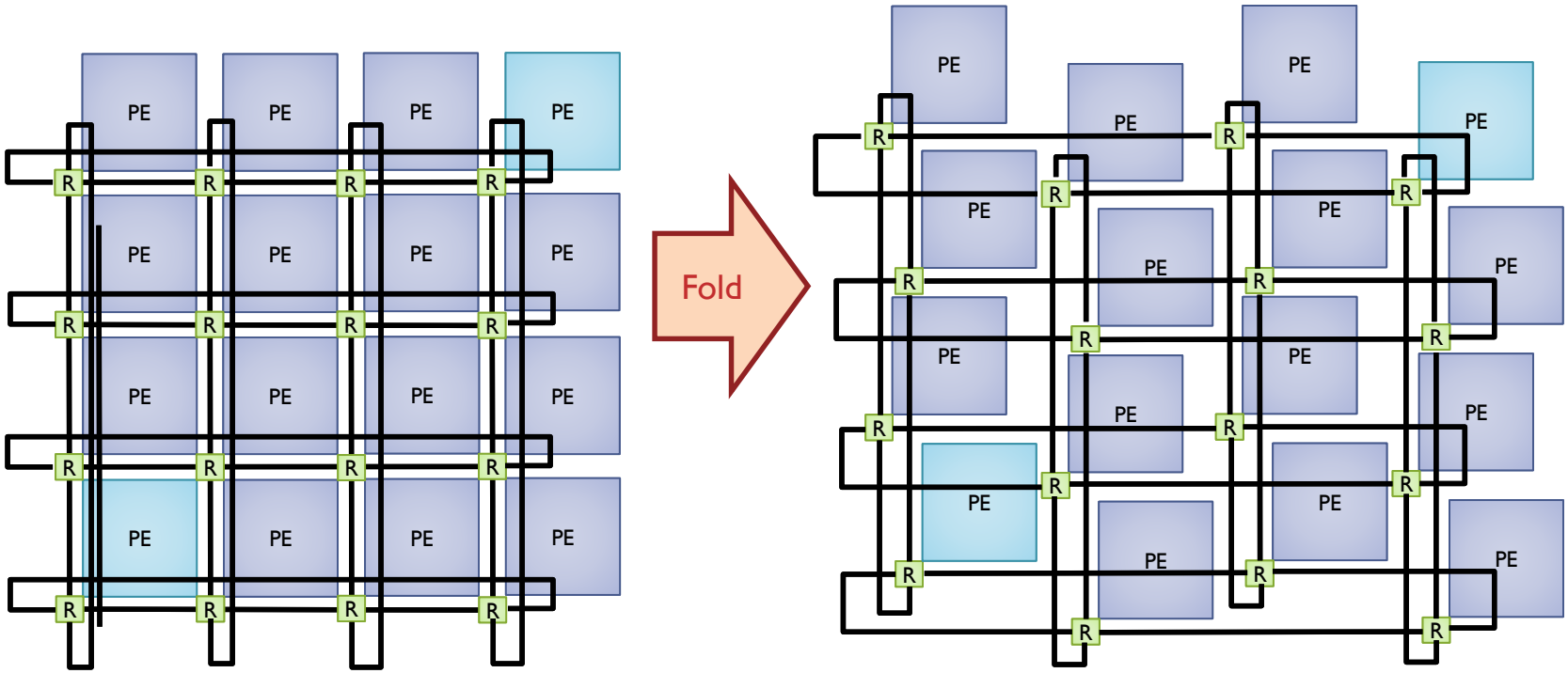
Direct Topology: Mesh



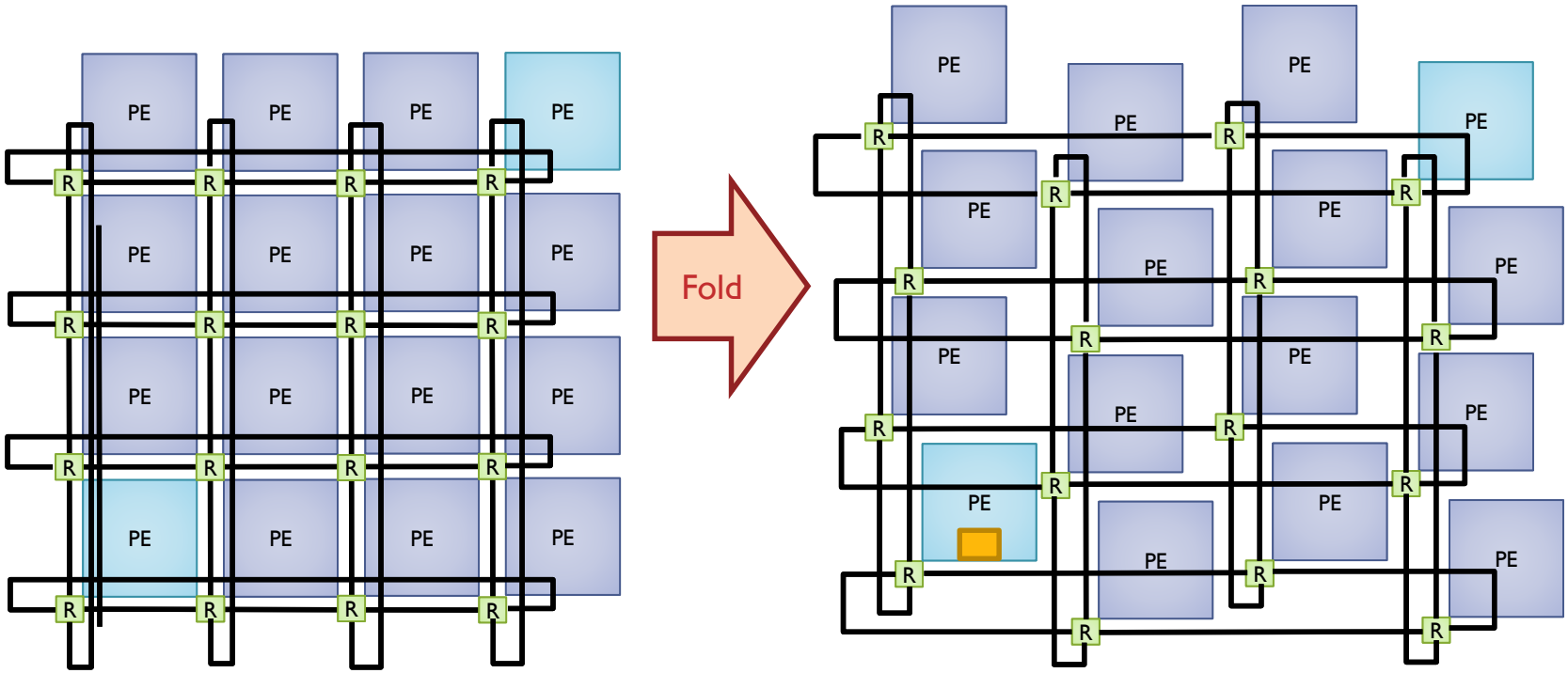
Direct Topology: Torus



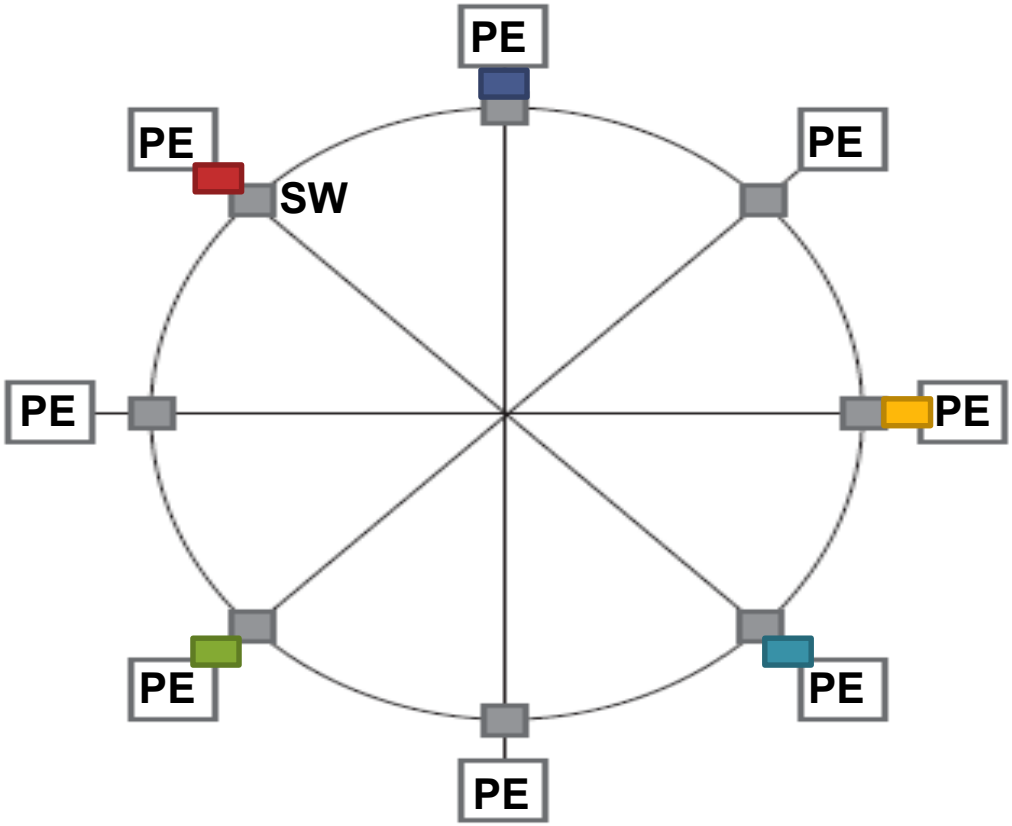
Direct Topology: Folded Torus



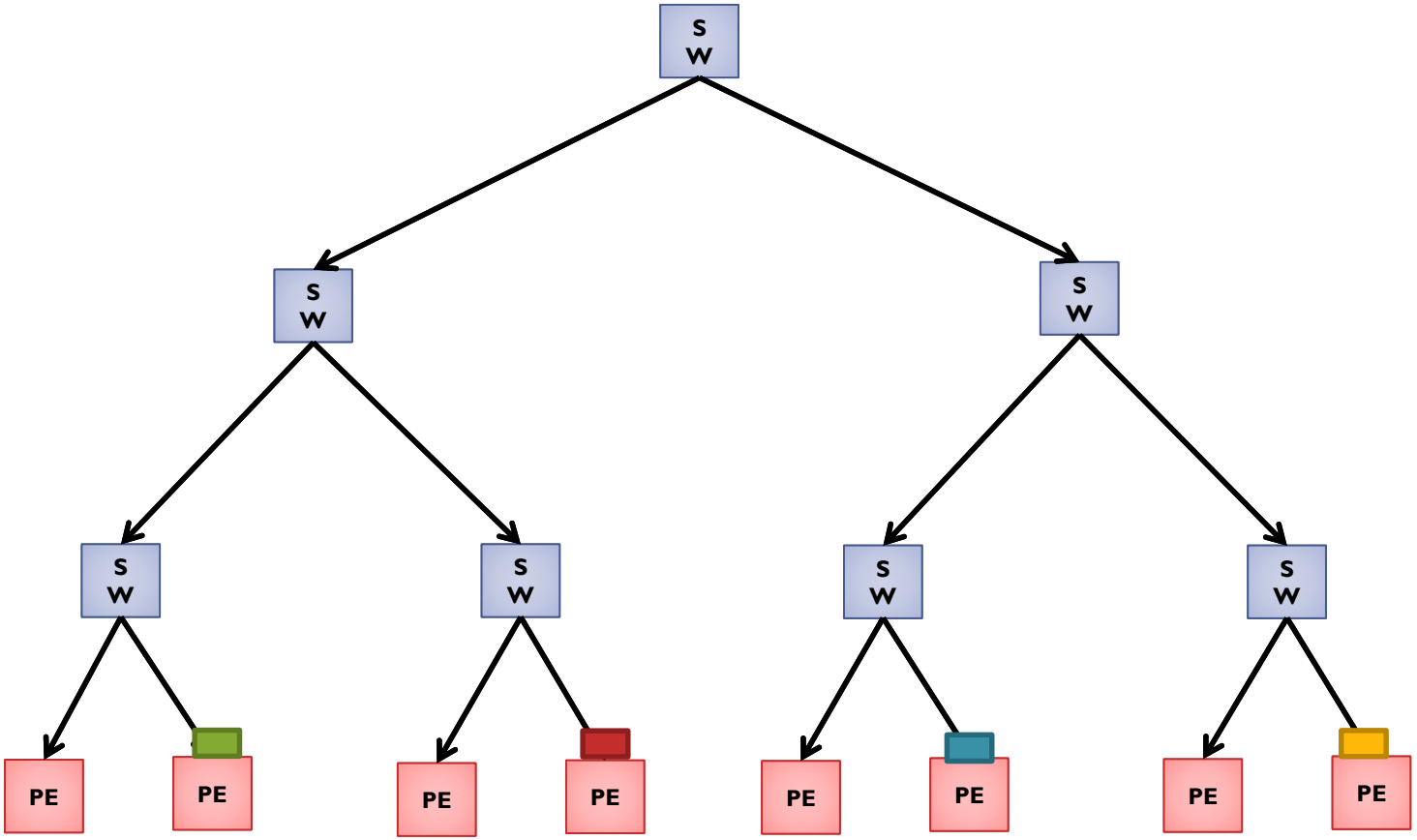
Direct Topology: Folded Torus



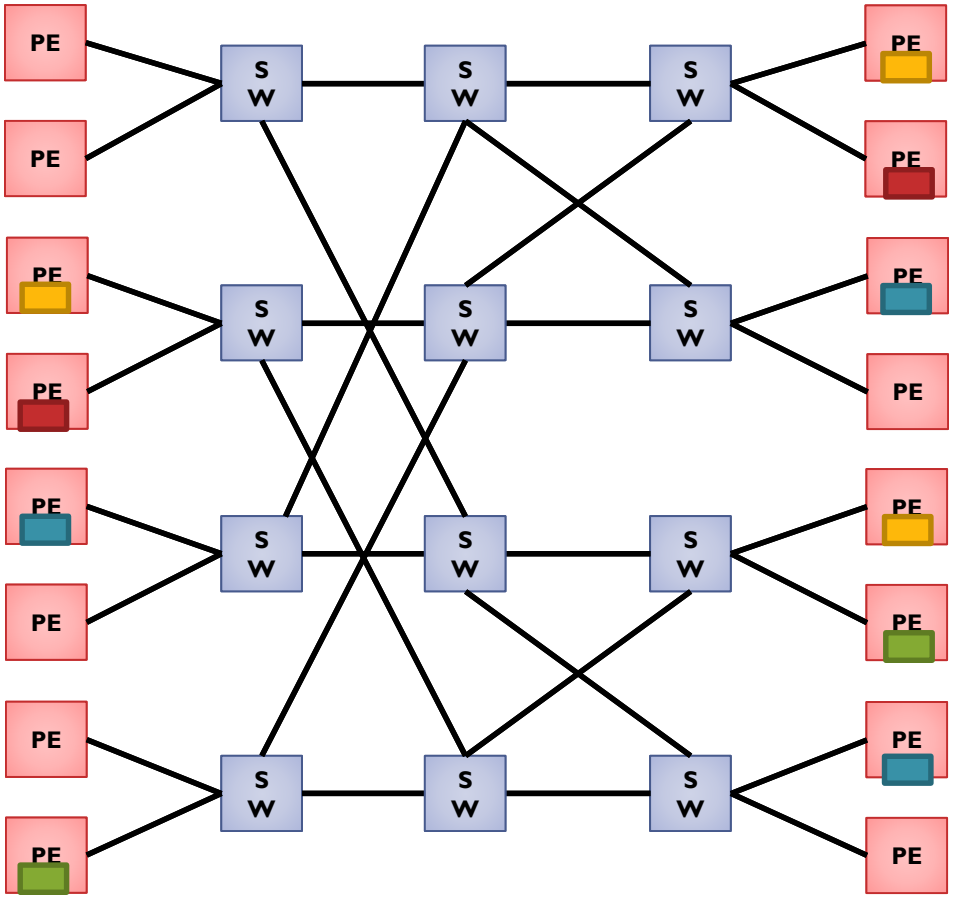
Direct Topology: Octagon



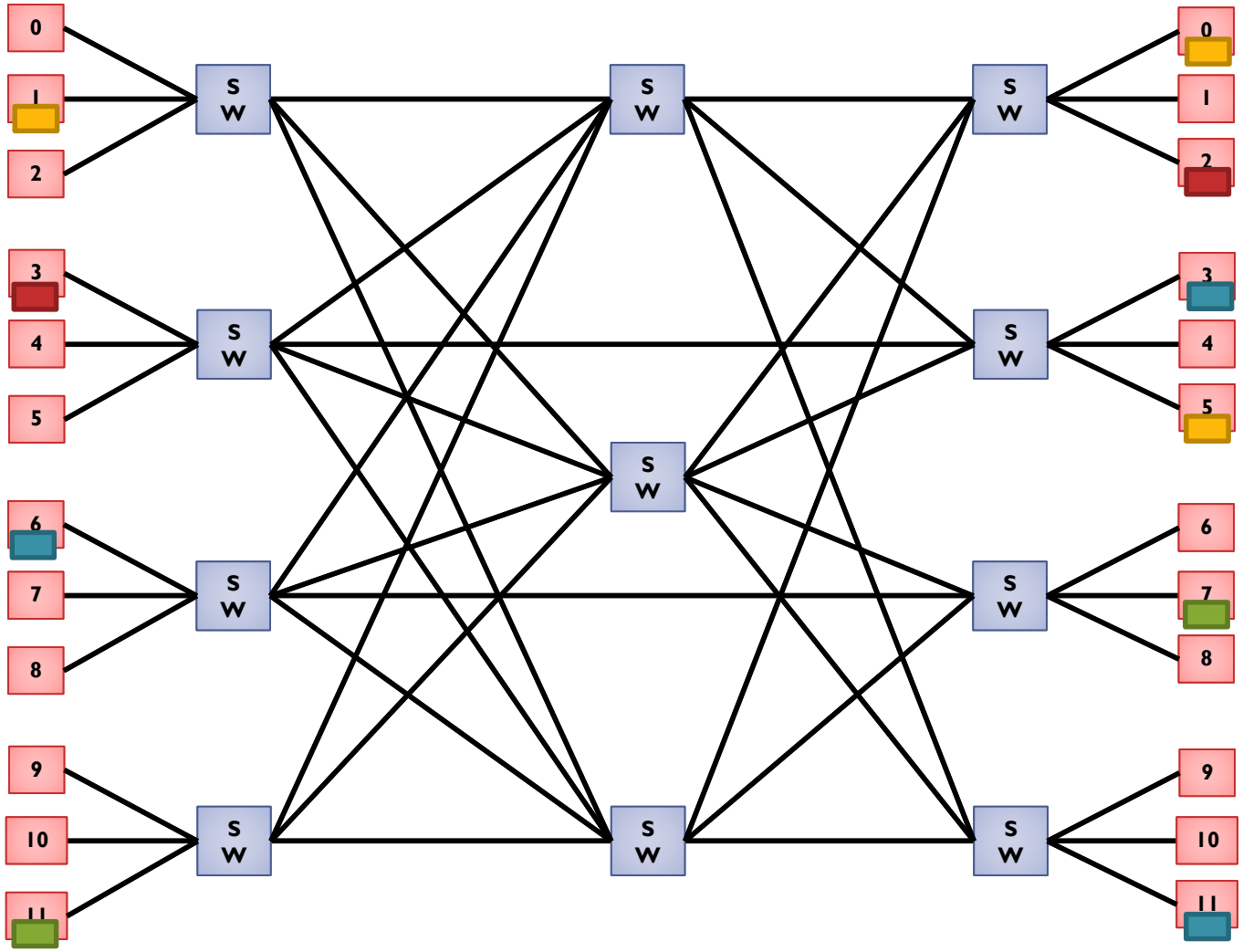
Indirect Topology: Fat Tree



Indirect Topology: k-ary n-fly butterfly network



Indirect Topology: (m, n, r) symmetric Clos network



How to Select a Topology ?

- **Application decides the topology type**
  - if *PEs = few tens* → Mesh is recommended
  - if *PEs = 100 or more* → Hierarchical star is recommended
- Some topologies are better for certain designs than others

Part II: NoC Building Blocks

Topology

Routing Algorithms

Routing Mechanisms

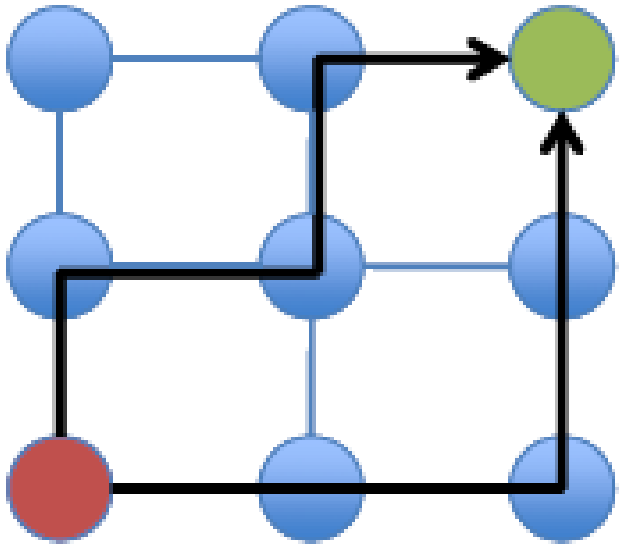
Control Flow

Network Interface

Router Architecture

NoC Routing

Routing algorithm determine path(s) from source to destination. Routing must prevent deadlock, livelock , and starvation.



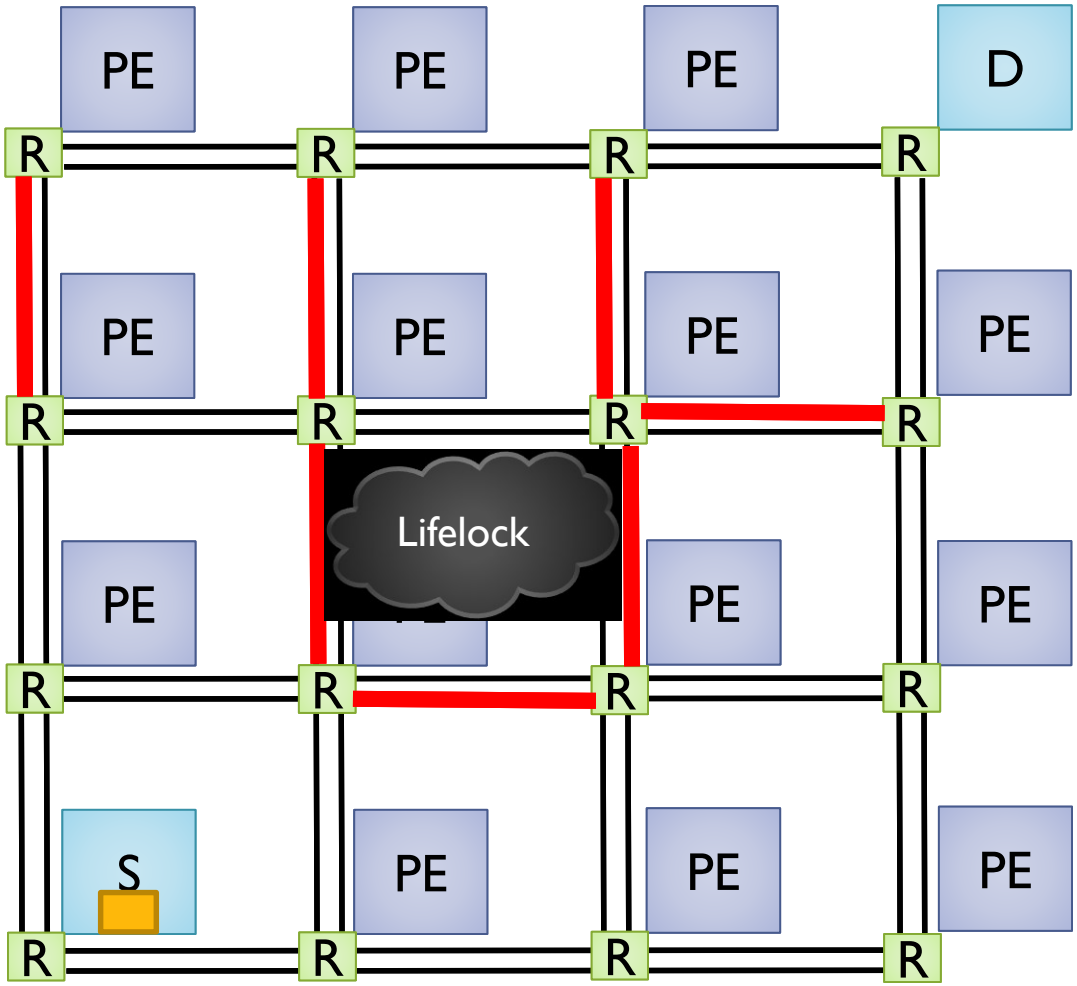
Deadlock, Livelock, and Starvation

Deadlock: A packet does not reach its destination, because it is blocked at some intermediate resource.

Livelock: A packet does not reach its destination, because it enters a cyclic path.

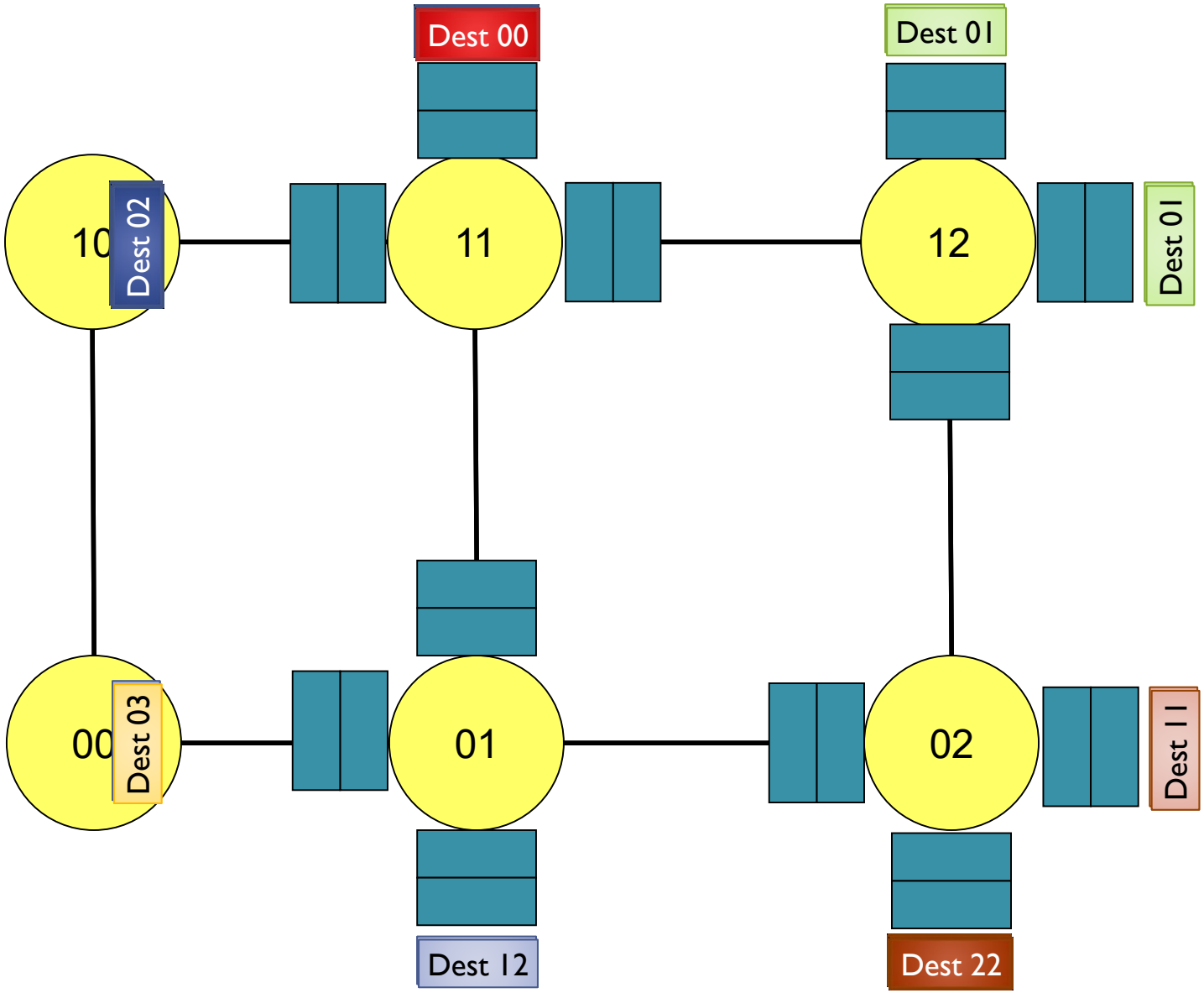
Starvation: A packet does not reach its destination, because some resource does not grant access (wile it grants access to other packets).

Lifelock

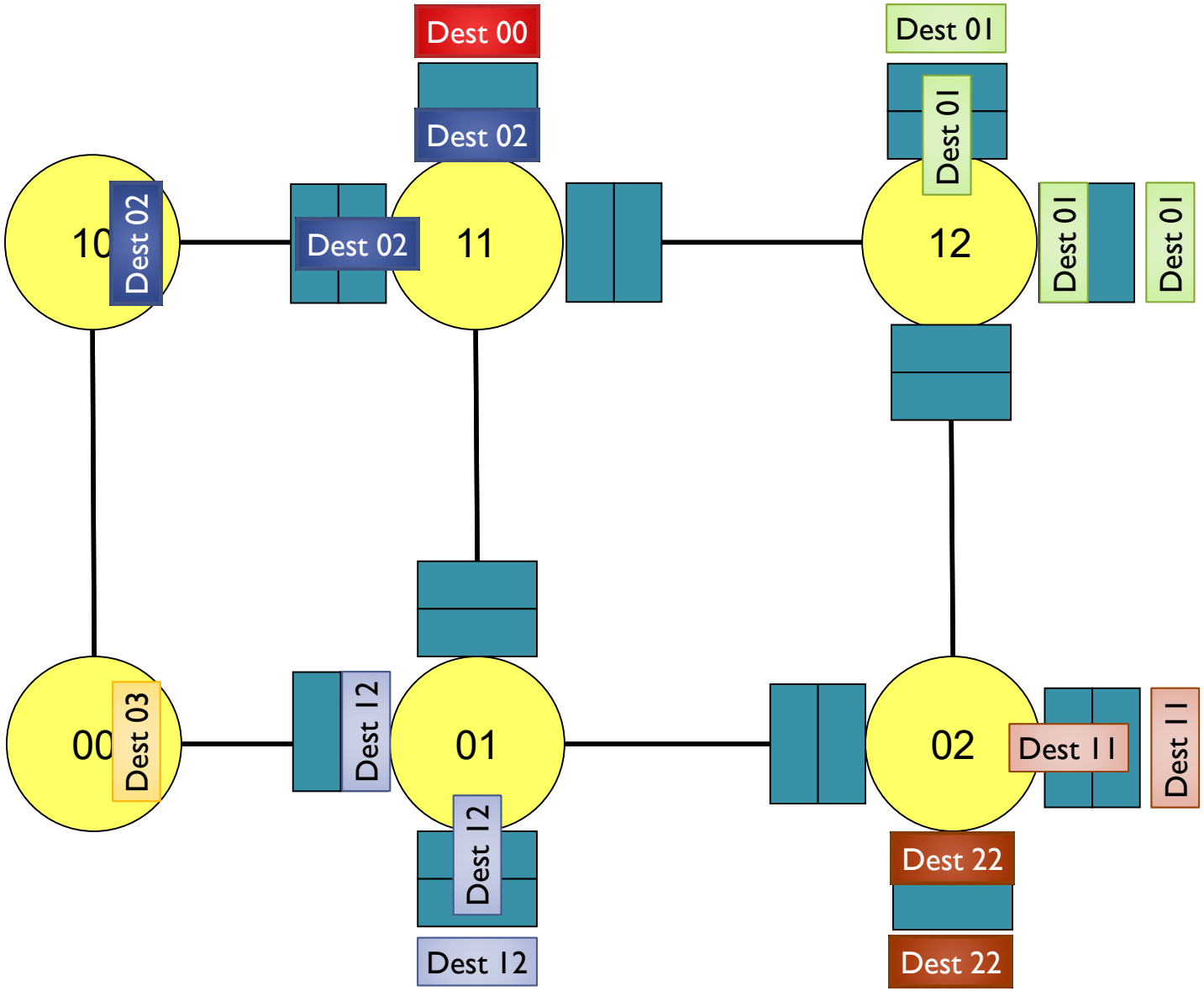


Congested channel

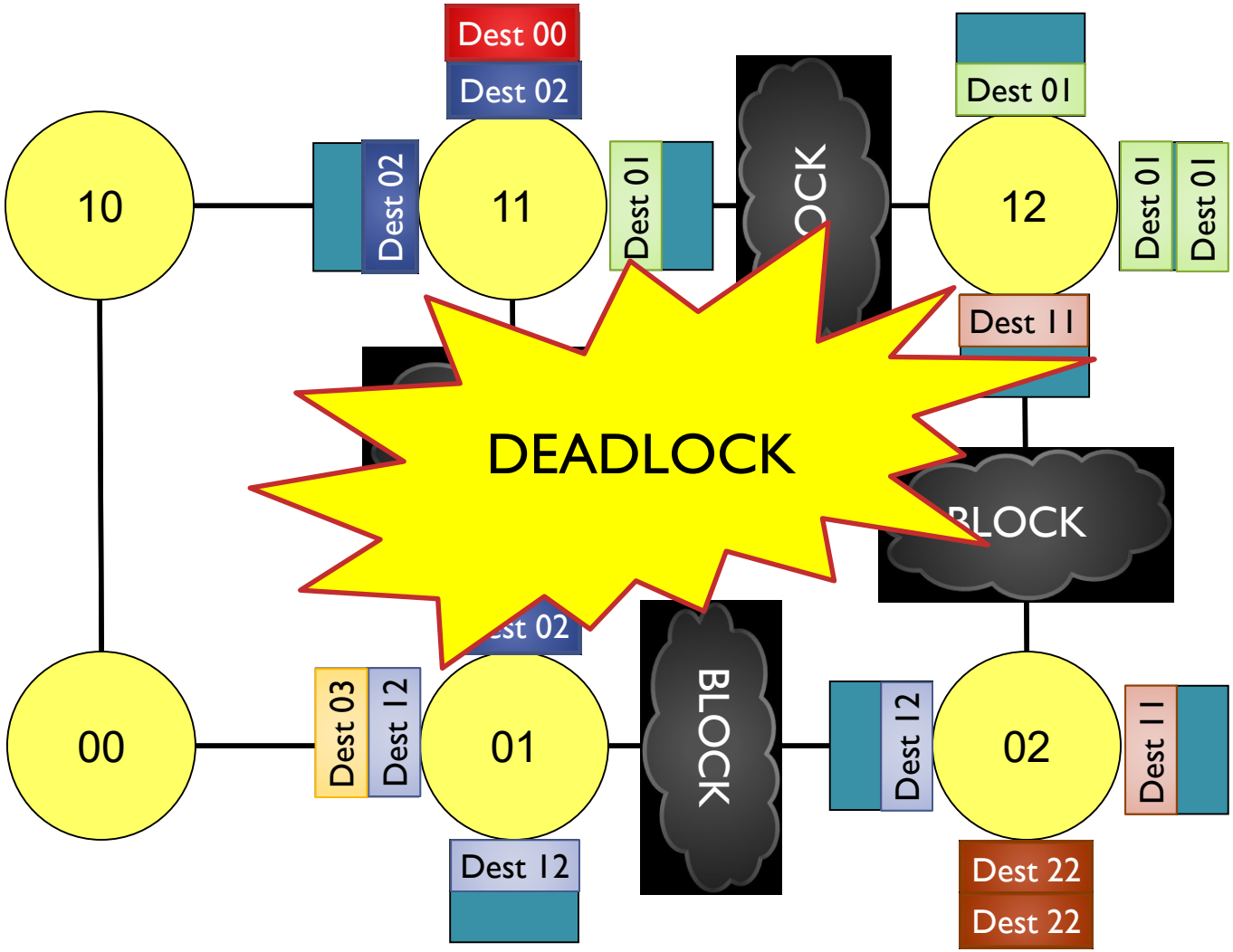
Deadlock


Dest 00
Dest 01
10
Dest 02
11
12
Dest 01
00
Dest 03
01
02
Dest 11
Dest 12
Dest 22

Deadlock


Dest 00
Dest 01
Dest 02
Dest 01
10
Dest 02
Dest 02
11
12
Dest 01
Dest 01
00
Dest 03
Dest 12
01
02
Dest 11
Dest 11
Dest 12
Dest 22
Dest 12
Dest 22

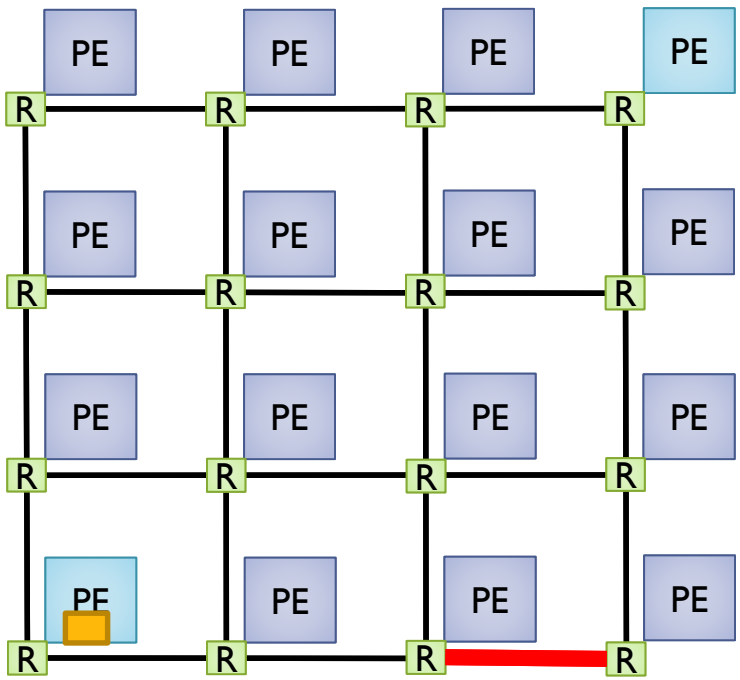
Deadlock


Dest 00
Dest 02
Dest 01
10
Dest 02
11
Dest 01
12
Dest 01
Dest 01
Dest 11
DEADLOCK
BLOCK
00
Dest 03
Dest 12
01
BLOCK
Dest 12
02
Dest 11
Dest 12
Dest 22
Dest 22

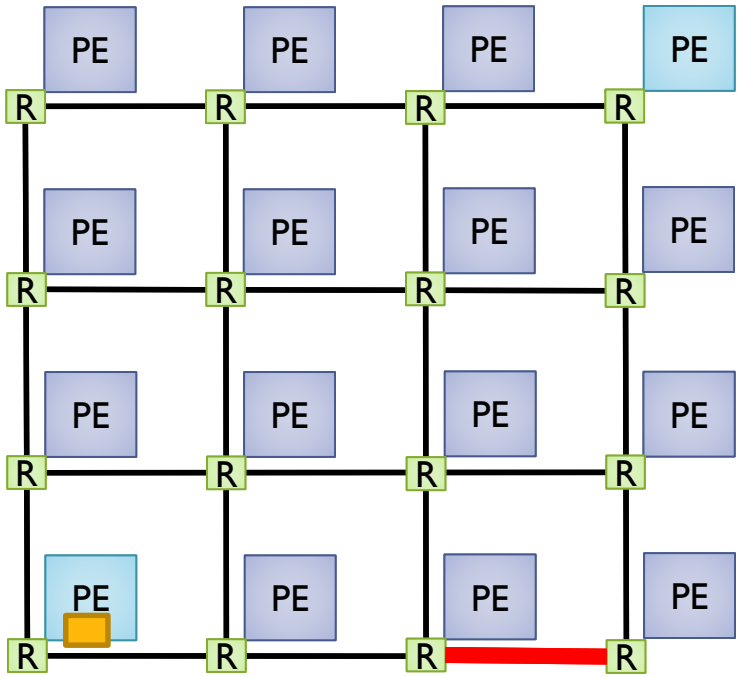
Routing Algorithm Attributes

- Number of destinations
  - Unicast, Multicast, Broadcast?
- Adaptivity
  - Deterministic, Oblivious or Adaptive
- Implementation (Mechanisms)
  - Source or node routing?
  - Table or circuit?

Static Vs. Adaptive Routing



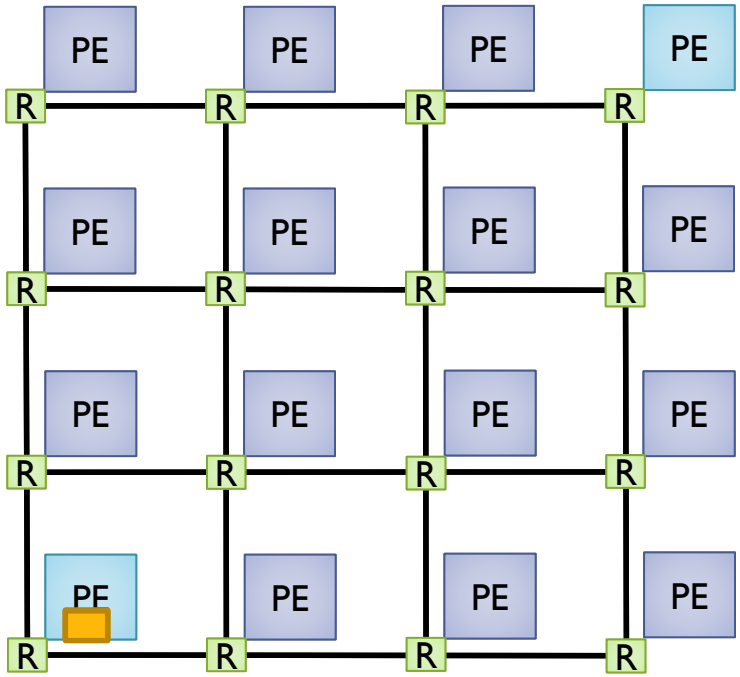
Static



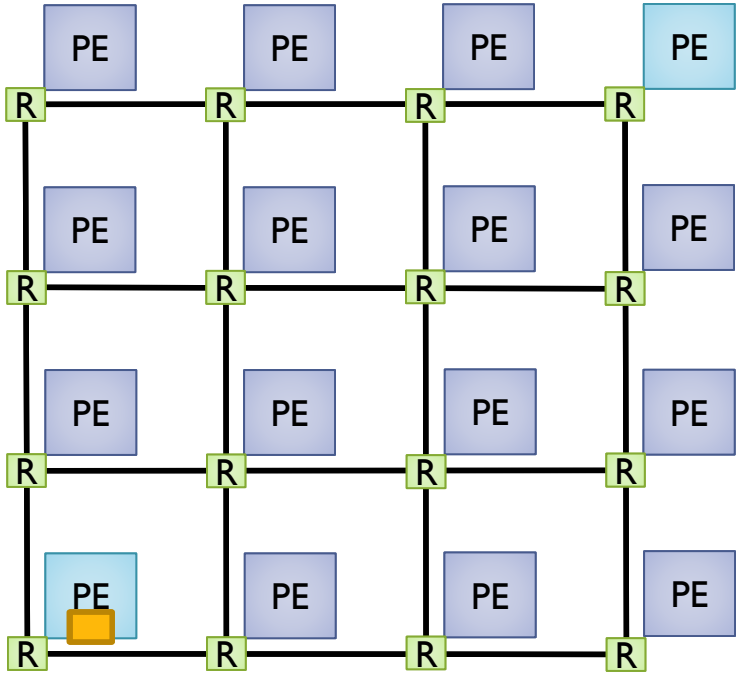
Adaptive

Congested channel

Minimal Vs. Non-Minimal

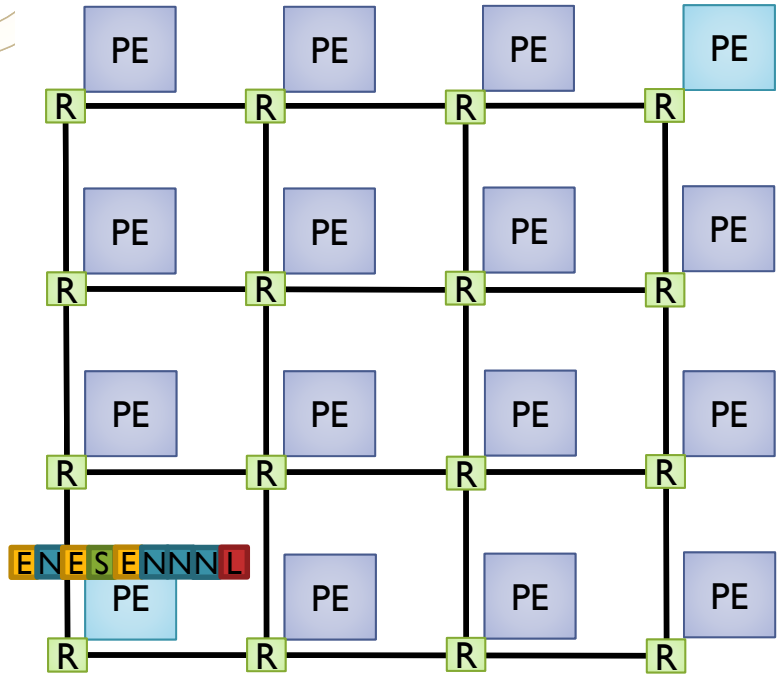


Minimal

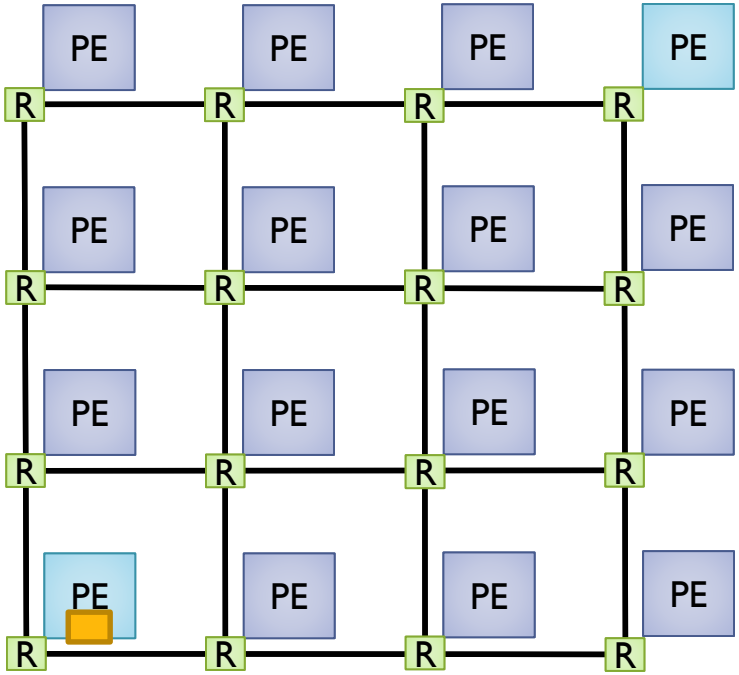


Non-Minimal

Source Vs. Distributed

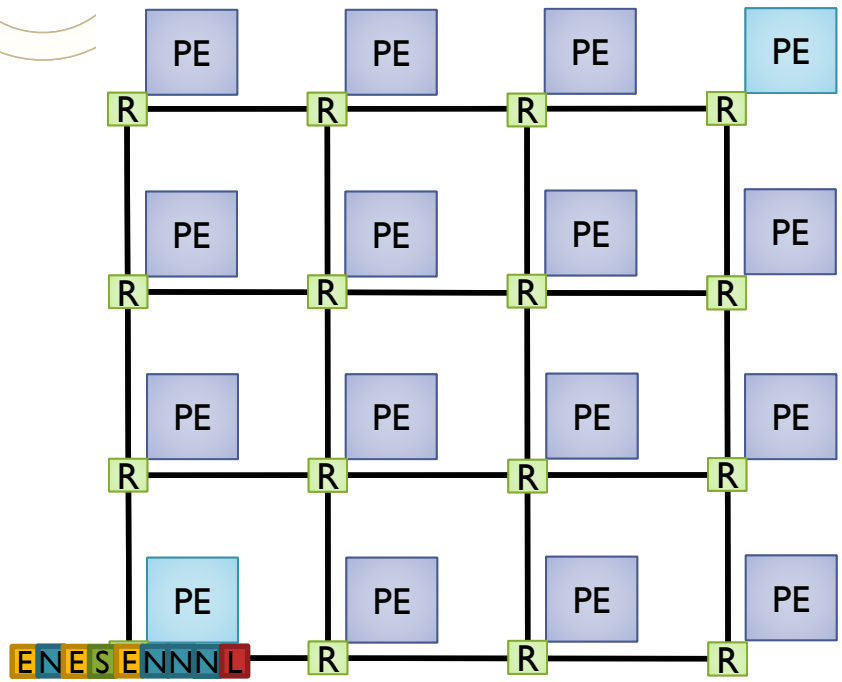


Source

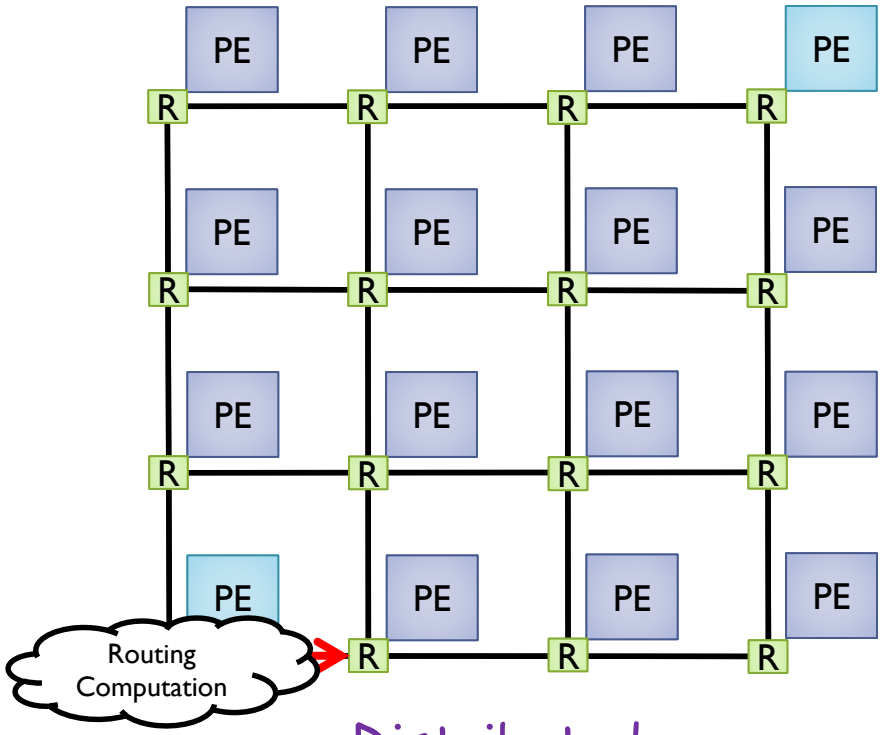


Distributed

Source Vs. Distributed

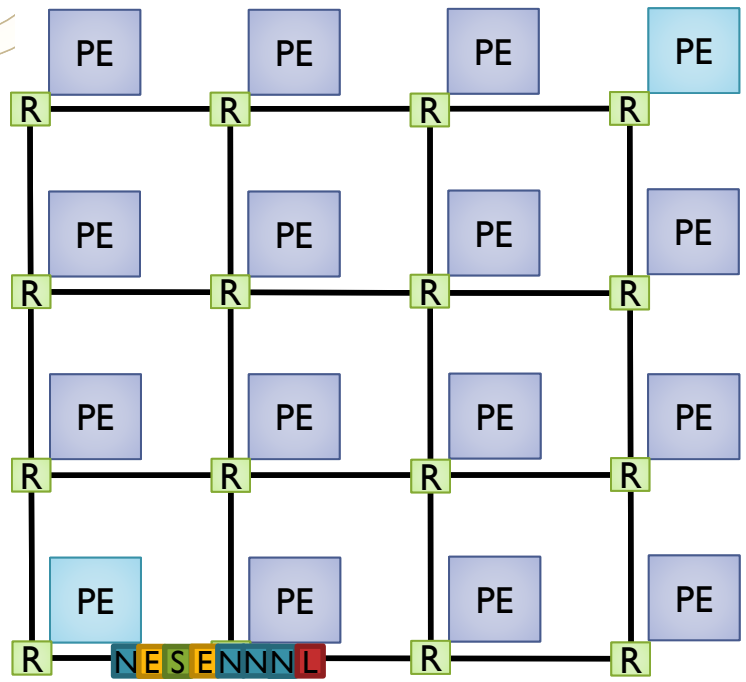


Source

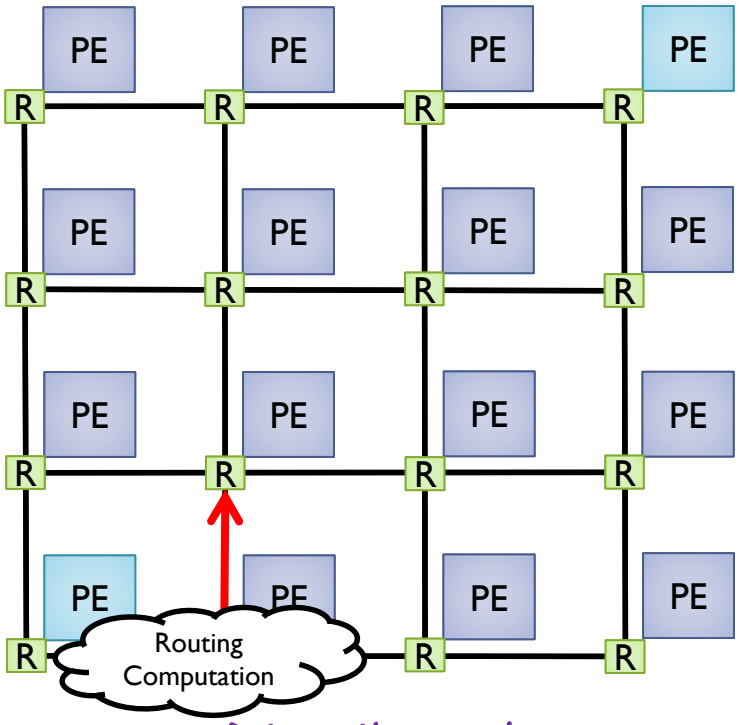


Distributed

Source Vs. Distributed

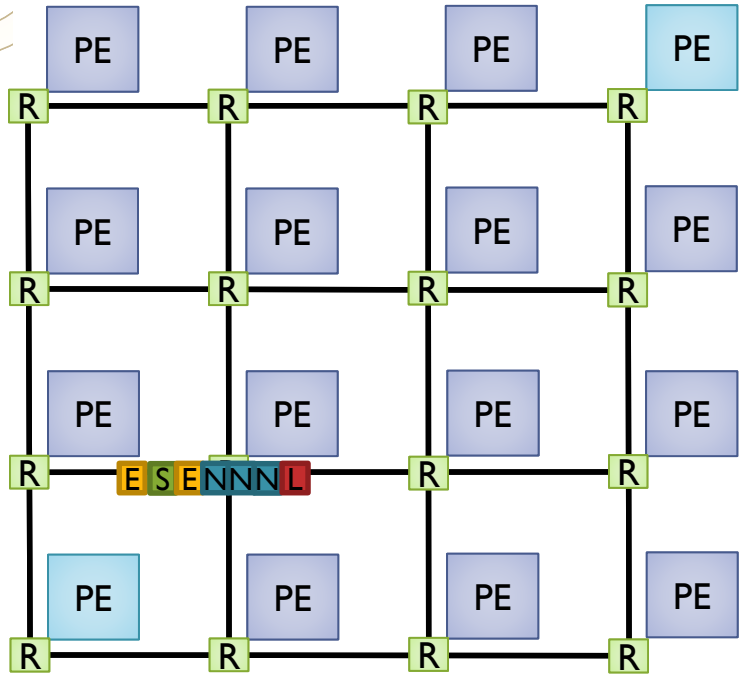


Source

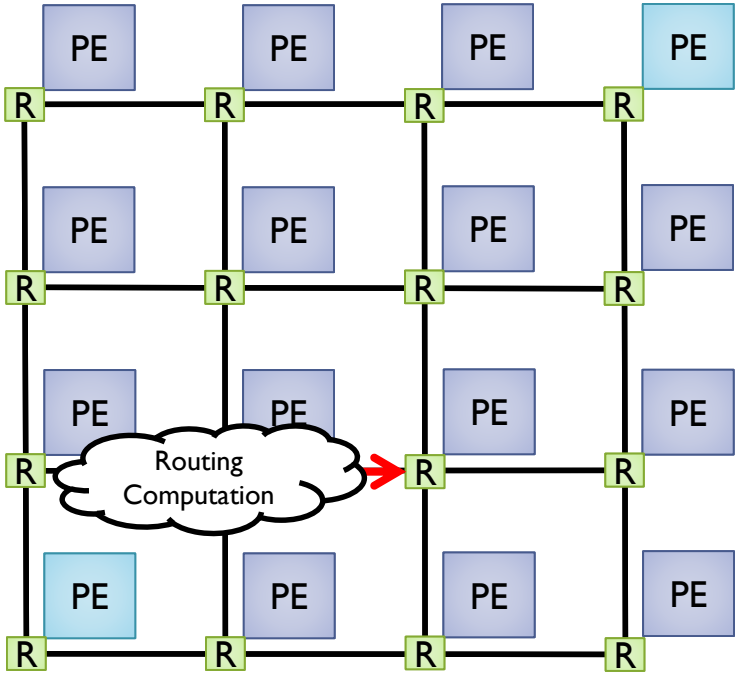


Distributed

Source Vs. Distributed

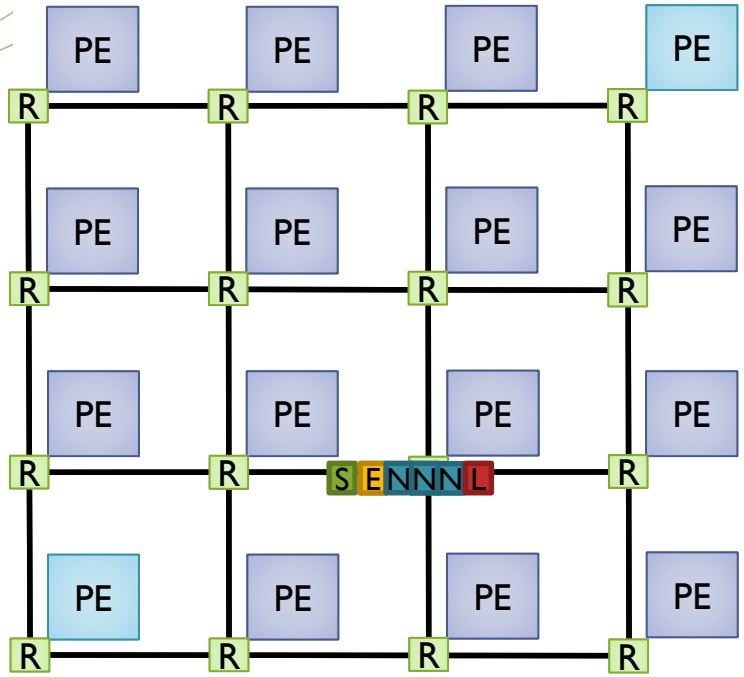


Source

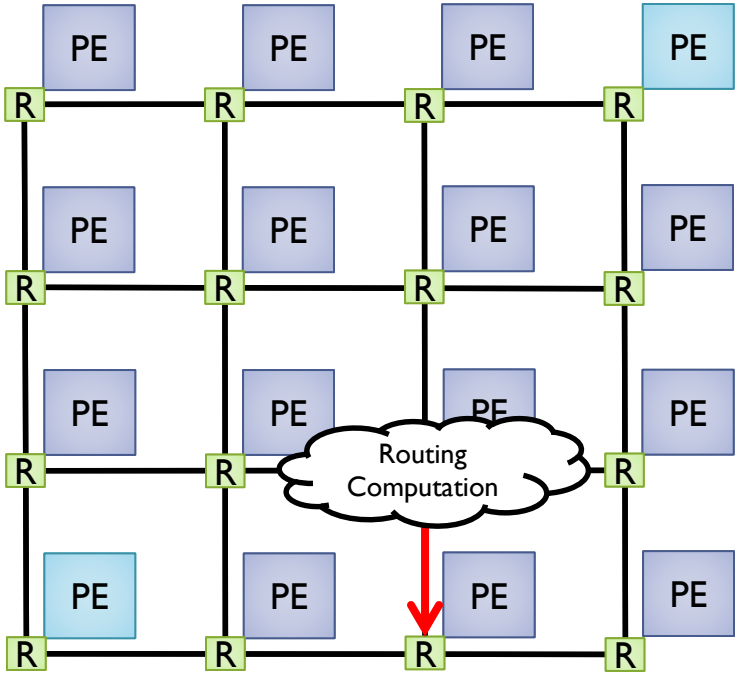


Distributed

Source Vs. Distributed

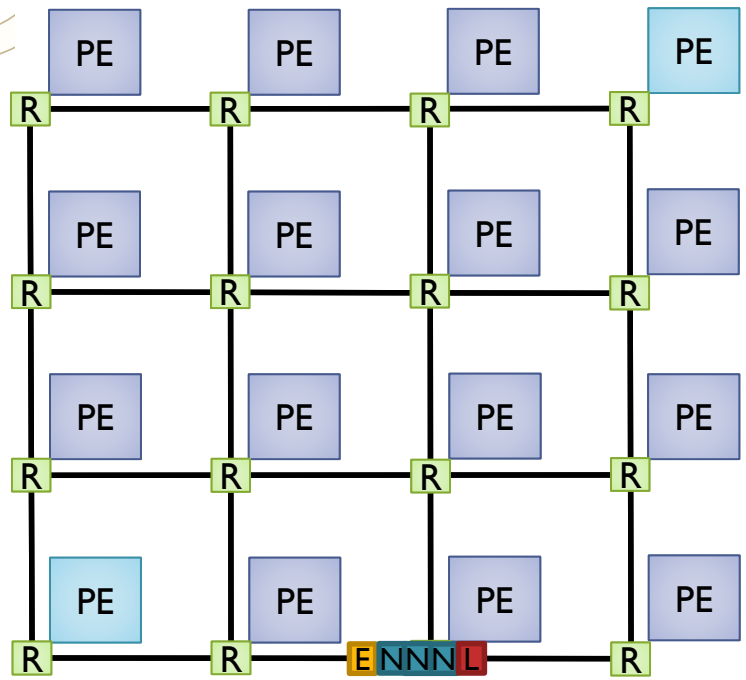


Source

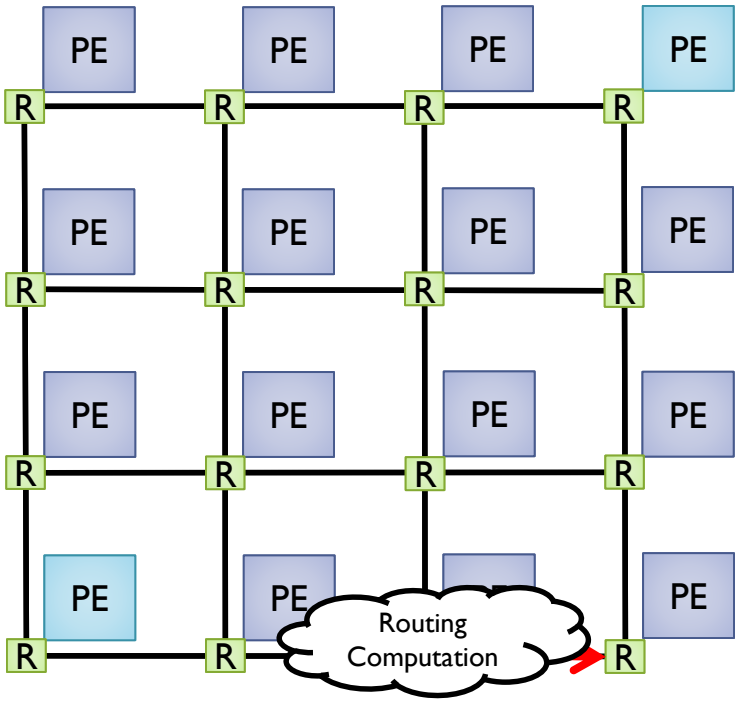


Distributed

Source Vs. Distributed

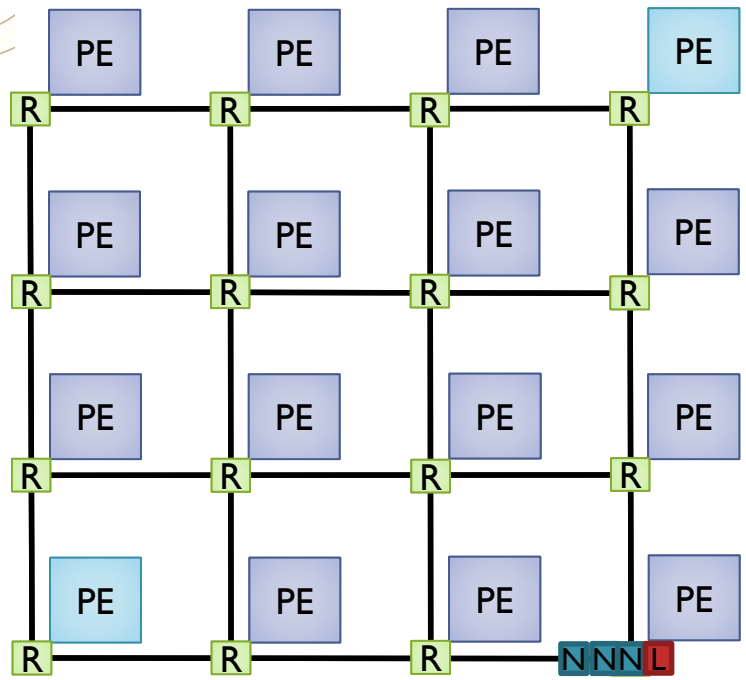


Source

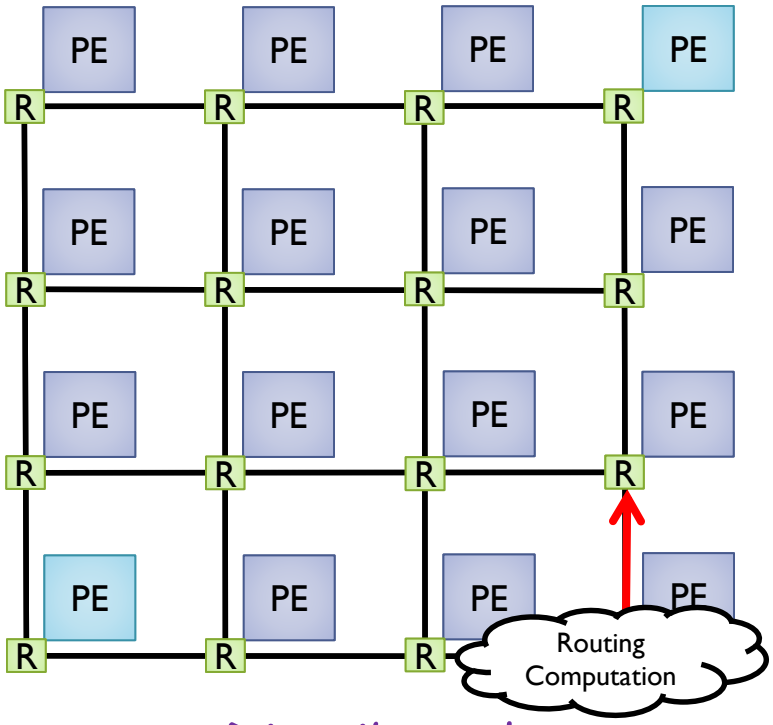


Distributed

Source Vs. Distributed

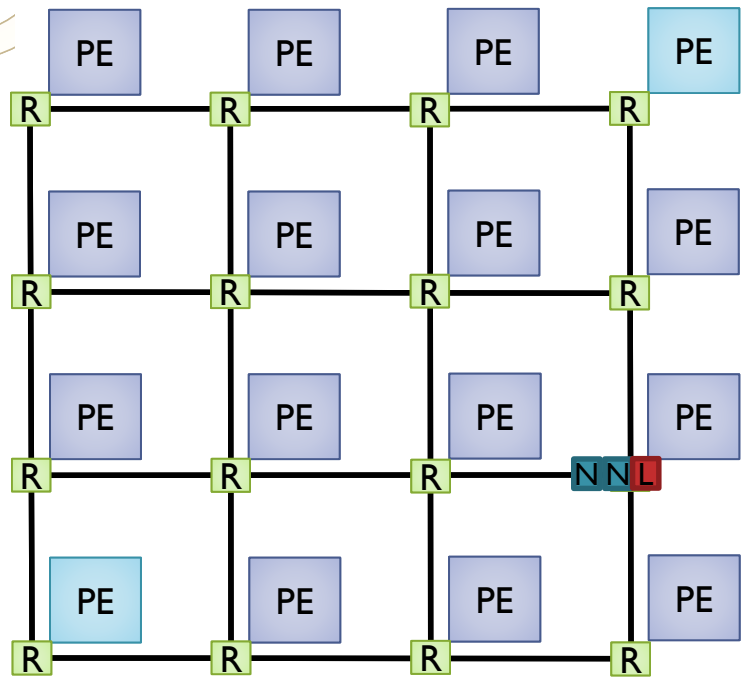


Source

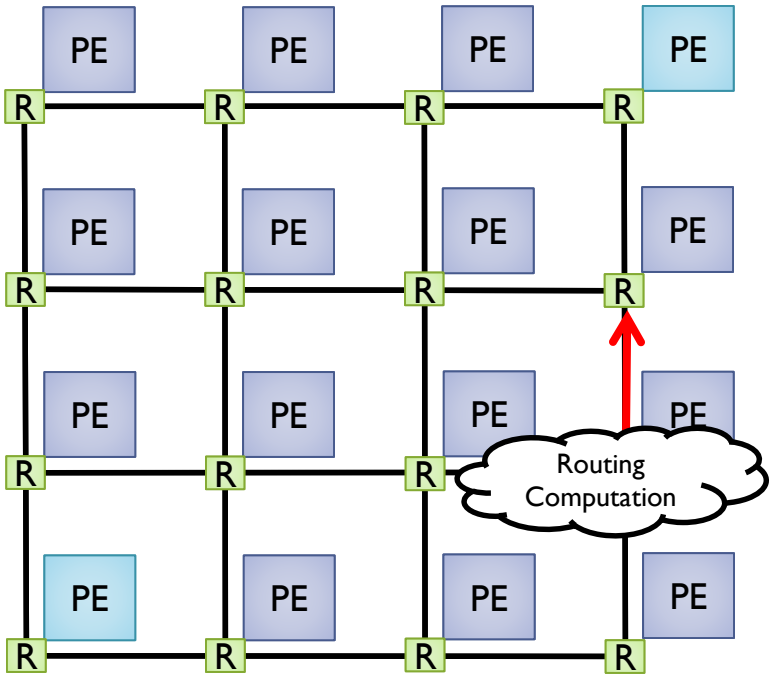


Distributed

Source Vs. Distributed

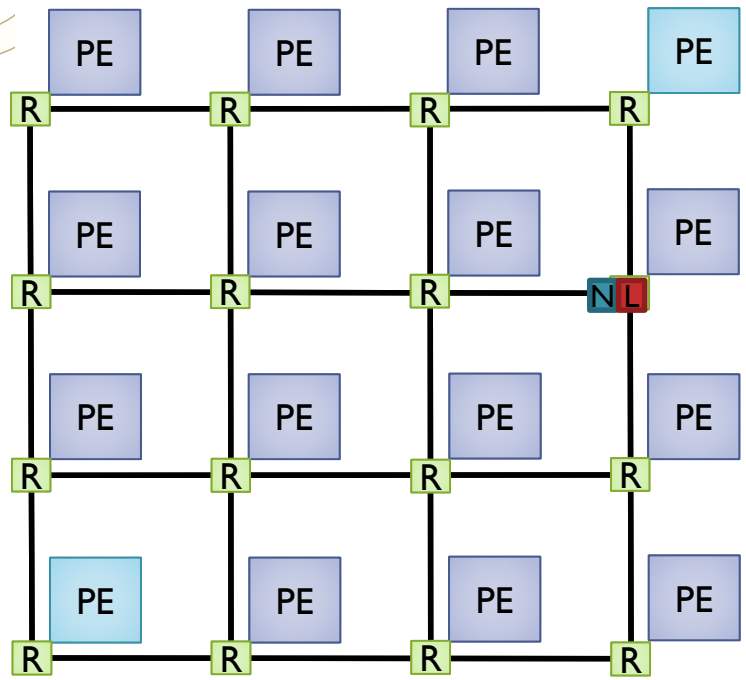


Source

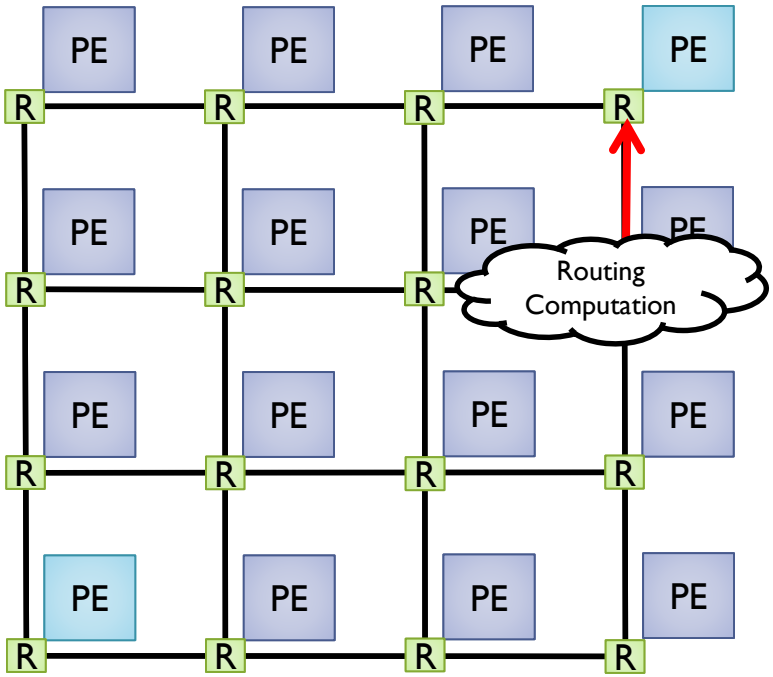


Distributed

Source Vs. Distributed

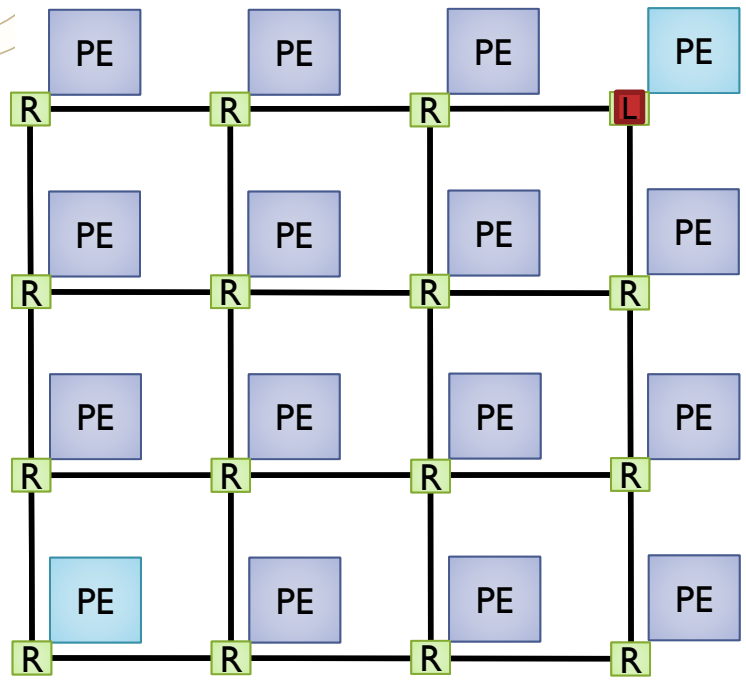


Source

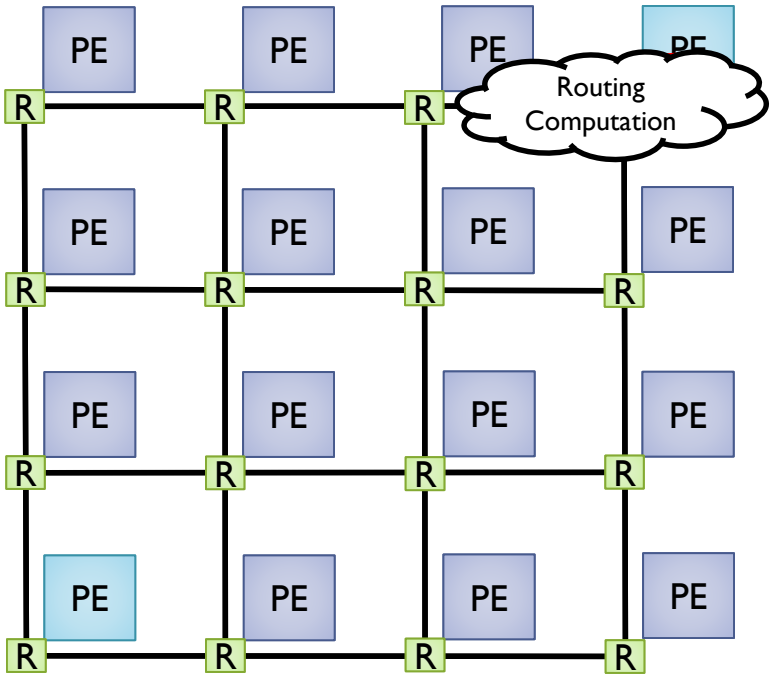


Distributed

Source Vs. Distributed

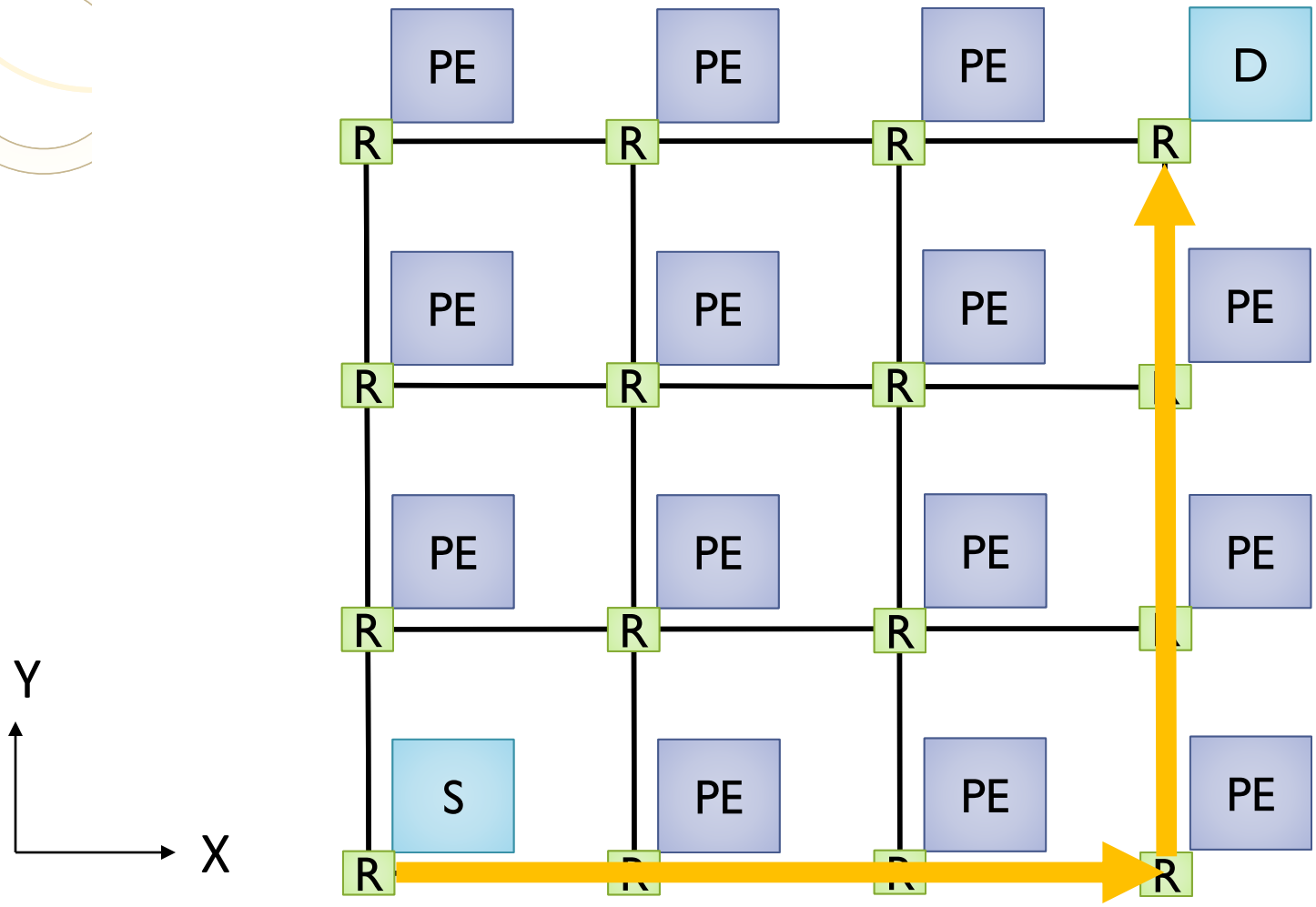


Source



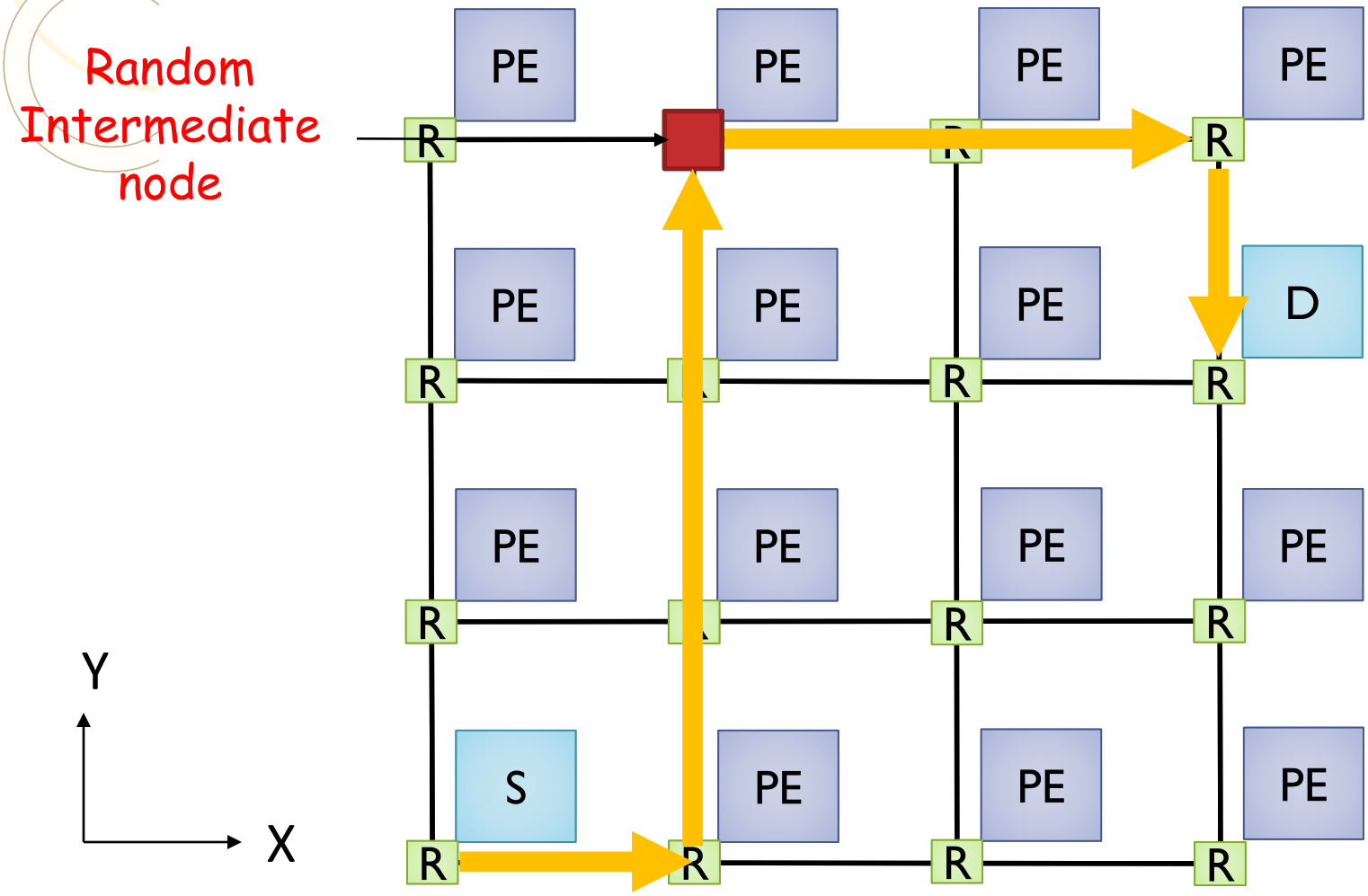
Distributed

Routing examples



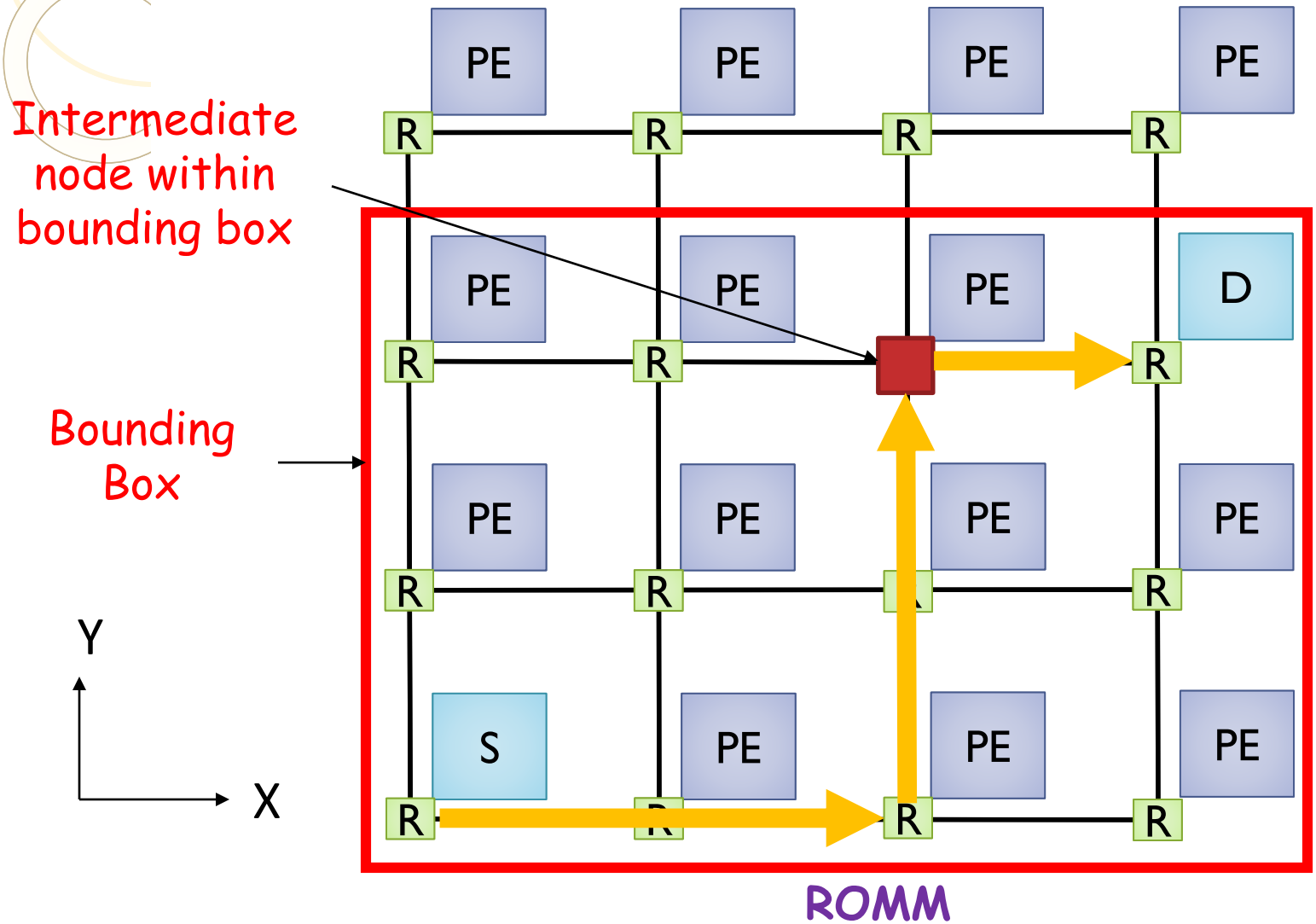
Dimension Ordered Routing
(XY Routing)

Routing examples

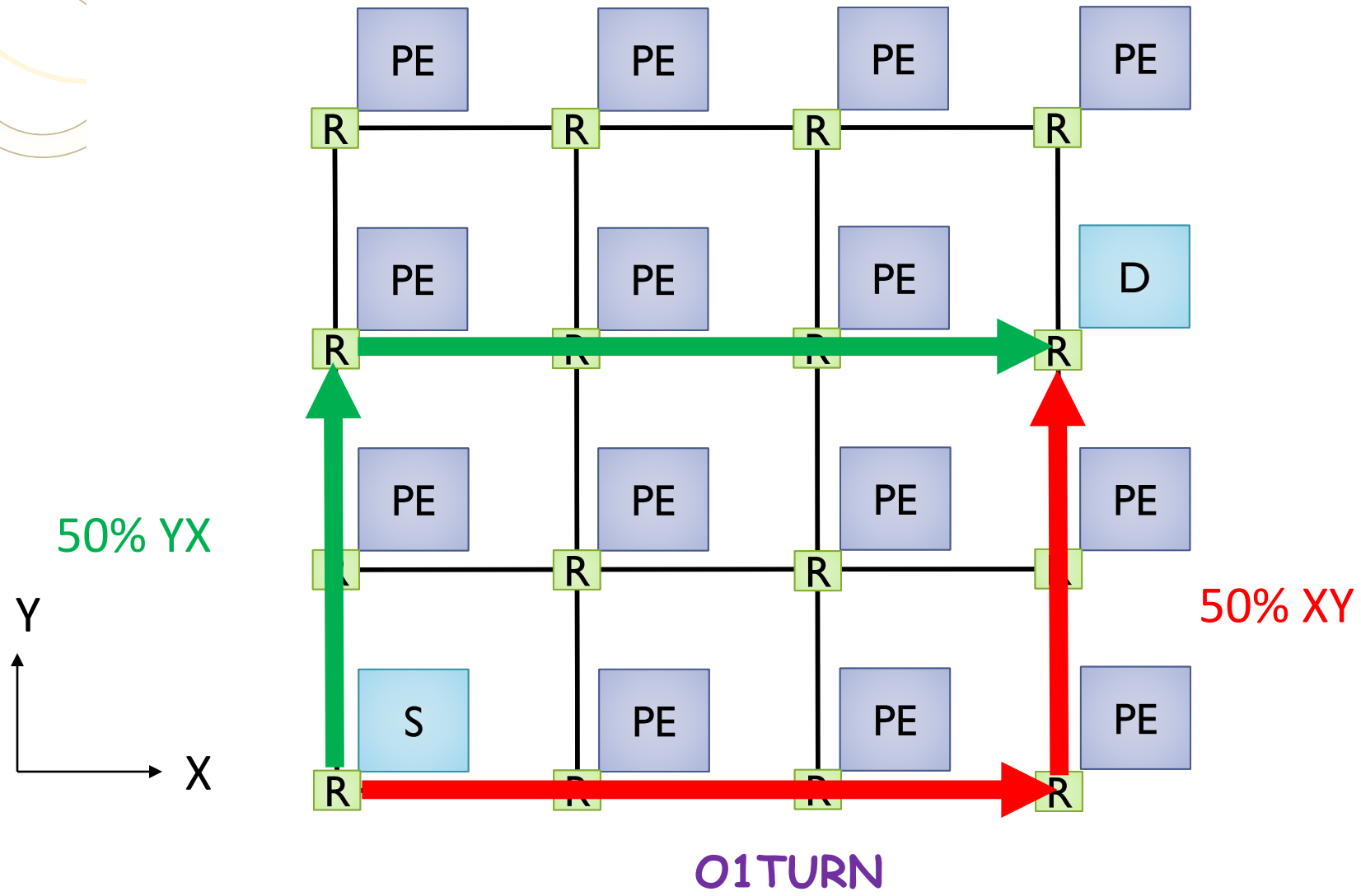


Valiant routing algorithm (VAL)

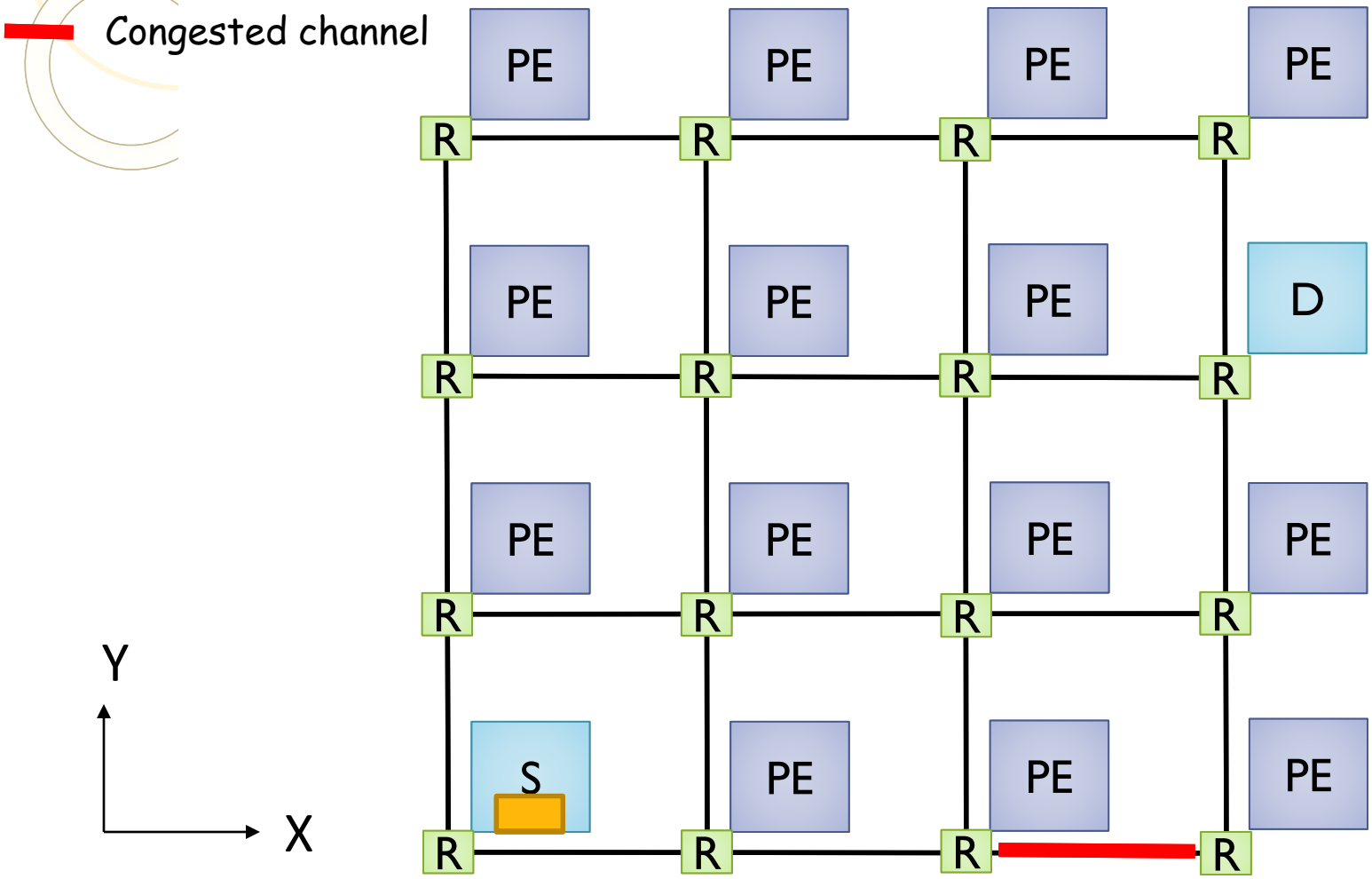
Routing examples



Routing examples



Routing examples



Dynamic XY
(DyXY)

Summary of Routing Algorithms

- Deterministic algorithms are simple and inexpensive but they do utilize path diversity and thus are weak on load balancing
- Oblivious algorithms give often good results since they allow good load balancing and their effects are easy to analyse
- Adaptive algorithms although in theory superior, are complex and power hungry

Summary of Routing Algorithms

- Latency paramount concern
  - Minimal routing most common for NoC
  - Non – minimal can avoid congestion and deliver low latency
- NoC researchers favor DOR for simplicity and deadlock freedom
- Here we only cover <u>unicast routing</u>

Part II: NoC Building Blocks

Topology

Routing Algorithms

Routing Mechanisms

Control Flow

Network Interface

Router Architecture

Routing Mechanism

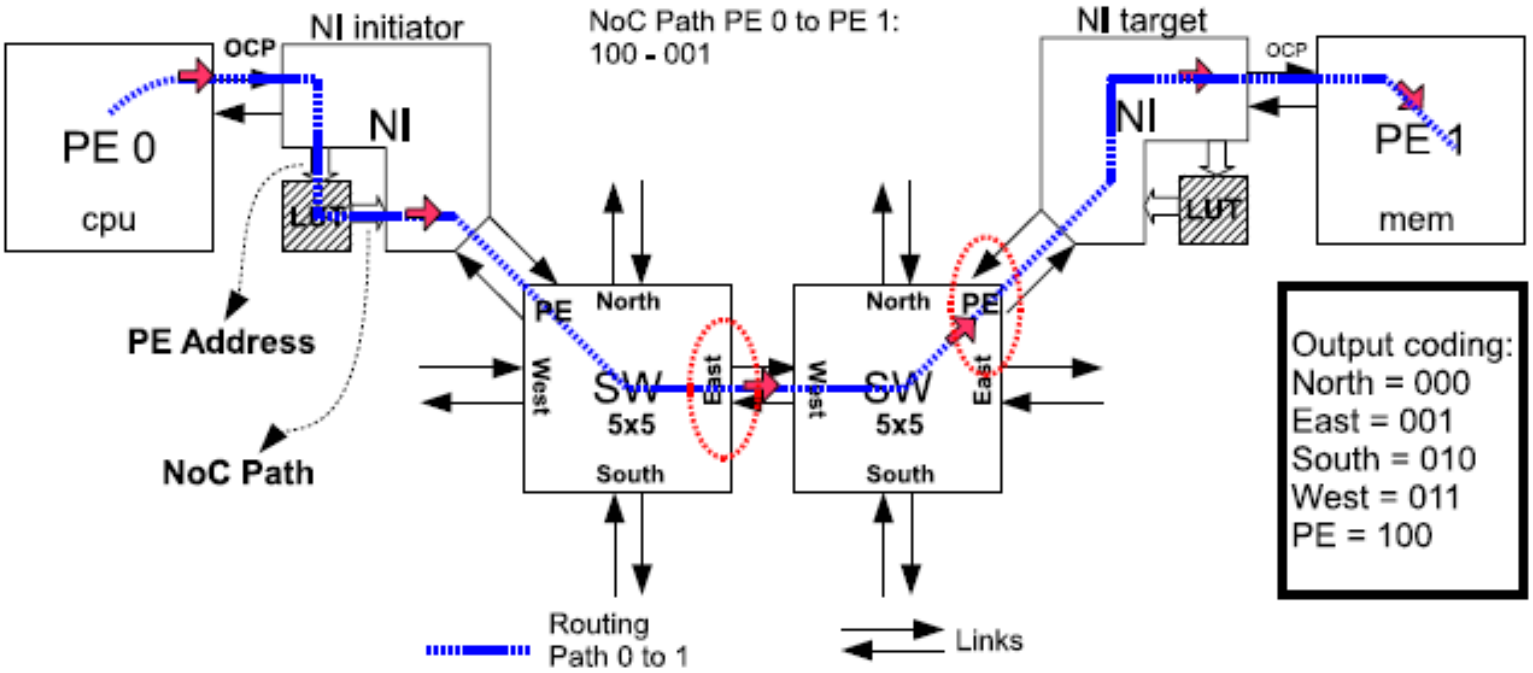
The term routing mechanics refers to the mechanism that is used to implement any routing algorithm.

- Two approaches:
  1. Fixed routing tables at the source or at each hop
  2. Algorithmic routing uses specialized hardware to compute the route or next hop at run-time

Table-based Routing

- Two approaches:
  - Source-table routing implements all-at-once routing by looking up the entire route at the source
  - Node-table routing performs incremental routing by looking up the hop-by-hop routing relation at each node along the route
- Major advantage:
  - A routing table can support any routing relation on any topology.

Table-based Routing



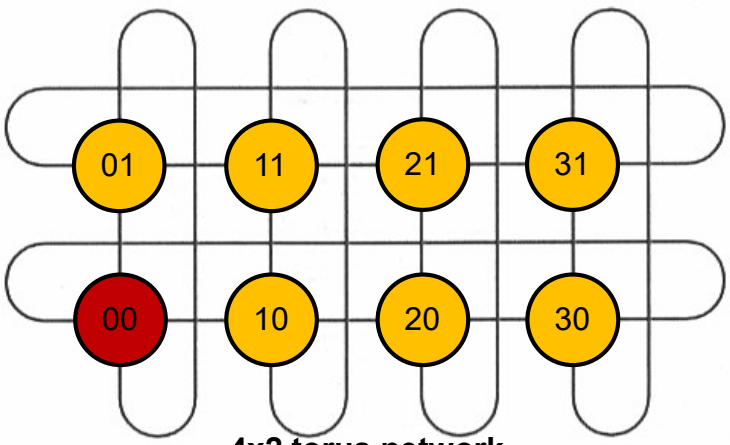
Example routing mechanism for deterministic source routing NoCs. The NI uses a LUT to store the route map.

Source Routing

- All routing decisions are made at the source terminal
- To route a packet we need:
  1) the table is indexed using the packet destination
  2) a route or a set of routes are returned, one route is selected
  3) the route is prepended and embedded in the packet
- Because of its speed, simplicity and scalability source routing is very often used for <u>deterministic and oblivious routing</u>

Source Routing - Example

- The example shows a routing table for a 4x2 torus network
- In this example there are <u>two</u> alternative routes for each destination
- Each node has its own routing table



4x2 torus network

In this example the order of XY should be the opposite, i.e. 21->12

Source routing table for node 00 of 4x2 torus network

Destination	Route 0	Route 1
00	X	X
10	EX	WWWX
20	EEX	WWX
30	WX	EEEX
01	NX	SX
11	NEX	ENX
21	NEEX select	WWNX
31	NWX	WNX

index → 21

Example:

-Routing from 00 to 21

-Table is indexed with 21

-Two routes:
NEEX and WWNX

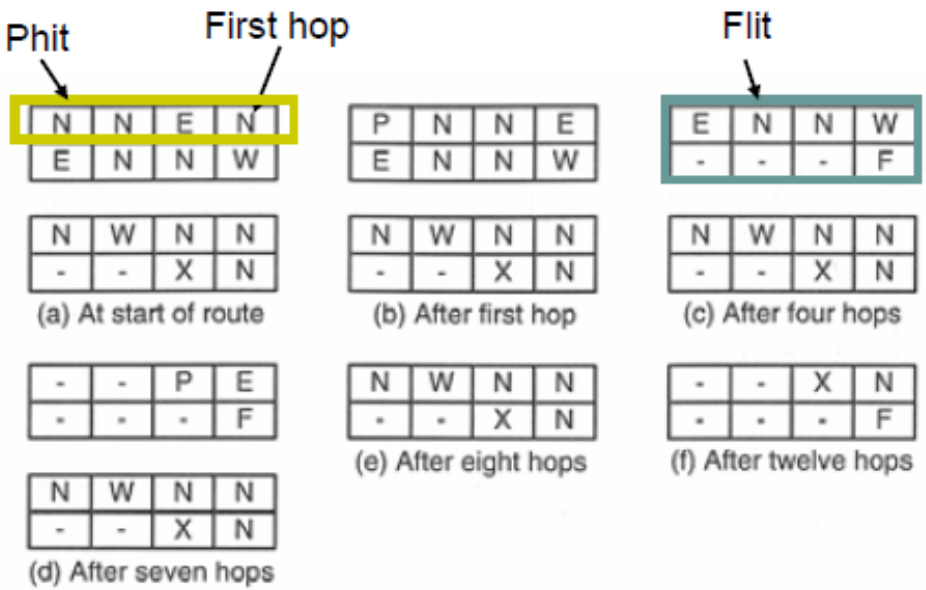
-The source arbitrarily selects NEEX

Arbitrary Length Encoding of Source Routes

- **Advantage:**
  - It can be used for arbitrary-sized networks
- The complexity of routing is moved from the network nodes to the terminal nodes
- But routers must be able to handle arbitrary length routes

Arbitrary Length-Encoding

- Router has
  - 16-bit phits
  - 32-bit flits
  - Route has 13 hops: NENNWNNENNWNN
- Extra symbols:
  - P: Phit continuation selector
  - F: Flit continuation Phit
- The tables entries in the terminals must be of arbitrary length



Node-Table Routing

- Table-based routing can also be performed by placing the routing table in the **routing nodes** rather than in the terminals

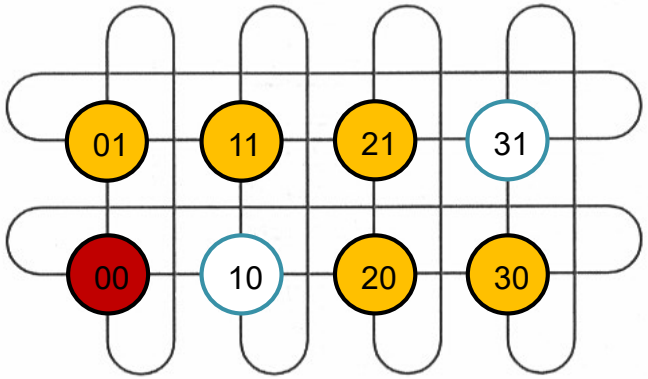
- Node-table routing is appropriate for **adaptive routing** algorithms, since it can use state information at each node

Node-Table Routing

- A table lookup is required, when a packet arrives at a router, which takes additional time compared to source routing
- Scalability is sacrificed, since different nodes need tables of varying size
- Difficult to give two packets arriving from a different node a different way through the network without expanding the tables

Example of Node-Table Routing

- Table shows a set of routing tables
- There are two choices from a source to a destination



Routing Table for Node 00

To	From 00		01		02		03		10		11		12		13	
00	X	X	W	**N**	W	E	E	**N**	S	N	S	W	S	W	S	E
01	E	**N**	X	X	W	**S**	E	W	S	**W**	S	N	S	W	S	W
02	E	W	E	**N**	X	X	W	**N**	S	W	S	E	S	N	S	W
03	W	**N**	E	W	E	**N**	X	X	S	**E**	S	E	S	E	S	N
10	N	S	N	W	N	W	N	E	X	X	W	**N**	W	E	E	**N**
11	N	E	N	S	N	W	N	W	E	**S**	X	X	W	**N**	E	W
12	N	E	N	E	N	S	N	W	E	W	E	**N**	X	X	W	**N**
13	N	W	N	E	N	E	N	S	W	**S**	E	W	E	**N**	X	X

Note: Bold font ports are misroutes

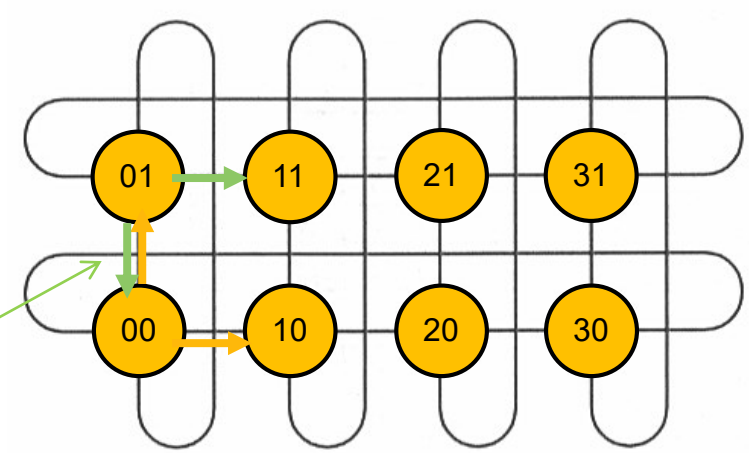
Example of Node-Table Routing

Livelock can occur

A packet passing through node 00 destined for node 11.

If the entry for (00->11) is N , go to 10 and (10-> 11) is S

=> 00 <-> 10 (livelock)



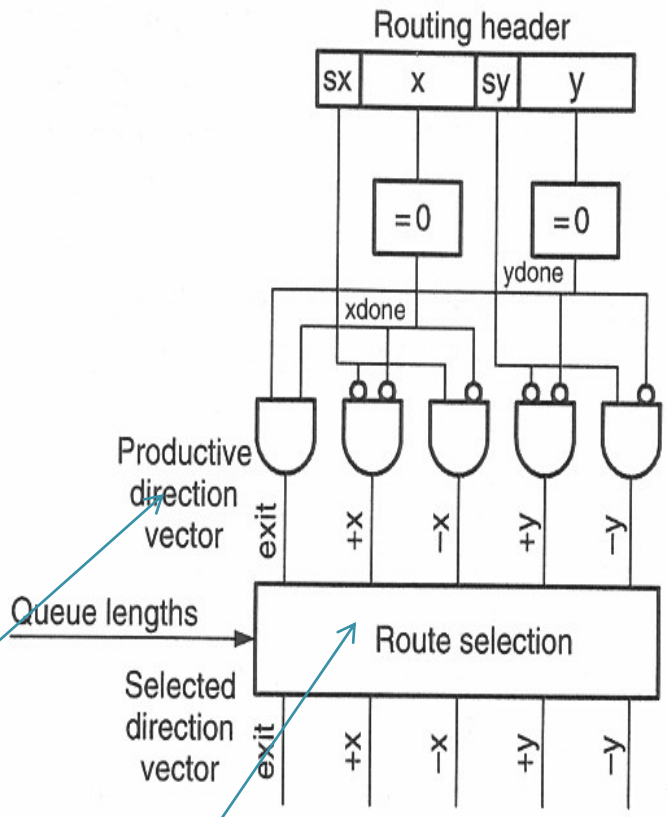
	From															
To	00		01		02		03		10		11		12		13	
00	X	X	W	**N**	W	E	E	**N**	S	N	S	W	S	W	S	E
01	E	**N**	X	X	W	**S**	E	W	S	**W**	S	N	S	W	S	W
02	E	W	E	**N**	X	X	W	**N**	S	W	S	E	S	N	S	W
03	W	**N**	E	W	E	**N**	X	X	S	**E**	S	E	S	E	S	N
10	N	S	N	W	N	W	N	E	X	X	W	**N**	W	E	E	**N**
11	N	E	N	S	N	W	N	W	E	**S**	X	X	W	**N**	E	W
12	N	E	N	E	N	S	N	W	E	W	E	**N**	X	X	W	**N**
13	N	W	N	E	N	E	N	S	W	**S**	E	W	E	**N**	X	X

Algorithmic Routing

- Instead of using a table, algorithms can be used to compute the next route
- In order to be fast, algorithms are usually not very complicated and implemented in hardware

Example: Algorithmic Routing

- Dimension-Order Routing
  - sx and sy indicated the preferred directions
    - sx=0, +x; sx=1, -x
    - sy=0, +y; sy=1, -y
  - x and y represent the number of hops in x an y direction
  - The PDV is used as an input for selection of a route



Determines the type of the routing

Indicates which channels advance the packet

Example: Algorithmic Routing

- A minimal oblivious router - Implemented by randomly selecting one of the active bits of the PDV as the selected direction
- Minimal adaptive router - Achieved by making selection based on the length of the respective output Qs.
- Fully adaptive router – Implemented by picking up unproductive direction if Qs > threshold results

Exercise

Compression of source routes. In the source routes, each port selector symbol [N,S,W,E, and X] was encoded with three bits. Suggest an alternative encoding to reduce the average length (in bits) required to represent a source route. Justify your encoding in terms of typical routes that might occur on a torus. Also compare the original three bits per symbol with your encoding on the following routes:

(a) NNNNNEEX

(b) WNEENWWWWWNX

Next lecture

Part II: NoC Building Blocks

Topology

Routing Algorithms

Routing Mechanisms

Switching

Control Flow

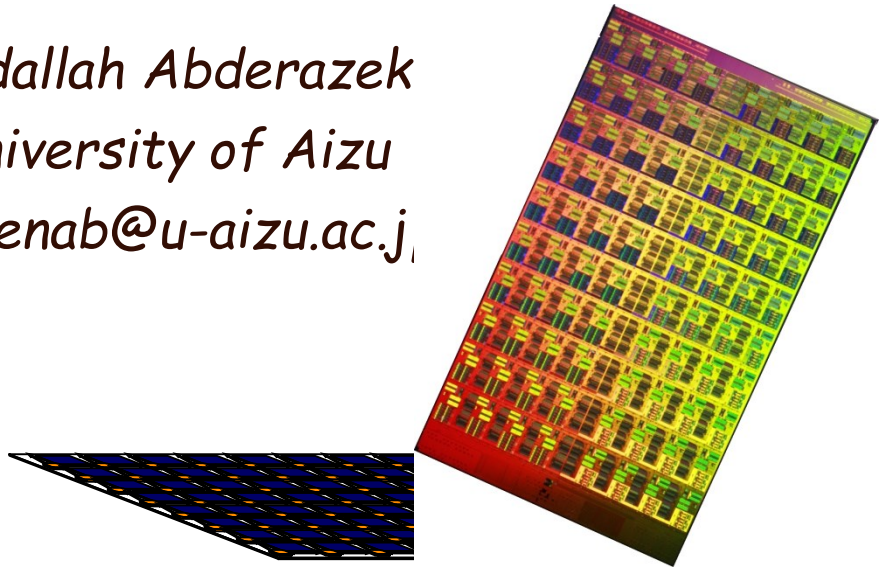
Router Architecture

Network Interface

Network-on-Chip

(2/2)

Ben Abdallah Abderazek
The University of Aizu
E-mail: benab@u-aizu.ac.j



Part II: NoC Building Blocks

Topology

Routing Algorithms

Routing Mechanisms

Switching

Flow Control

Router Architecture

Network Interface

Part II: NoC Building Blocks

Topology

Routing Algorithms

Routing Mechanisms

Switching

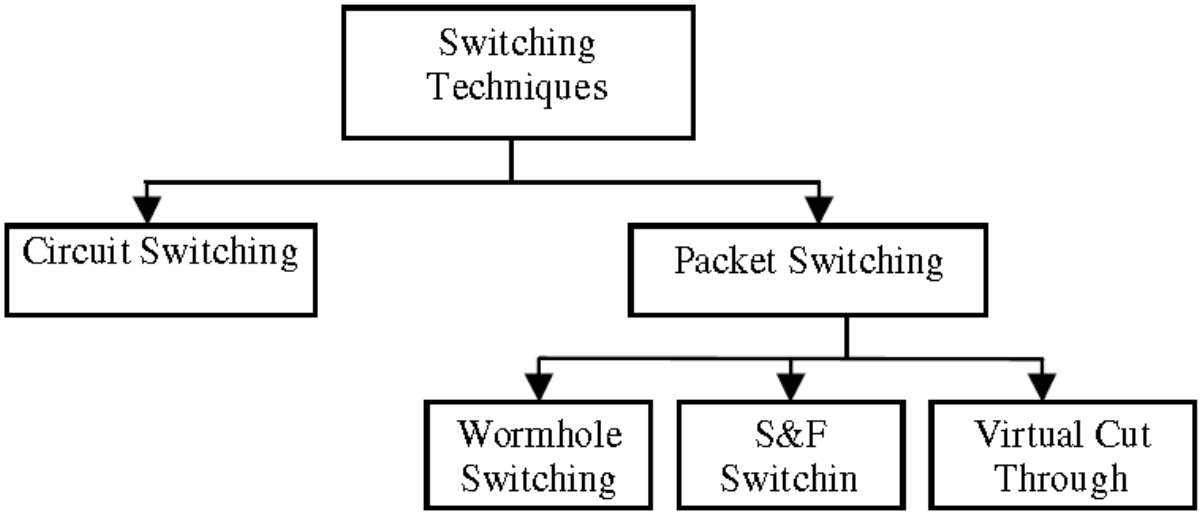
Flow Control

Router Architecture

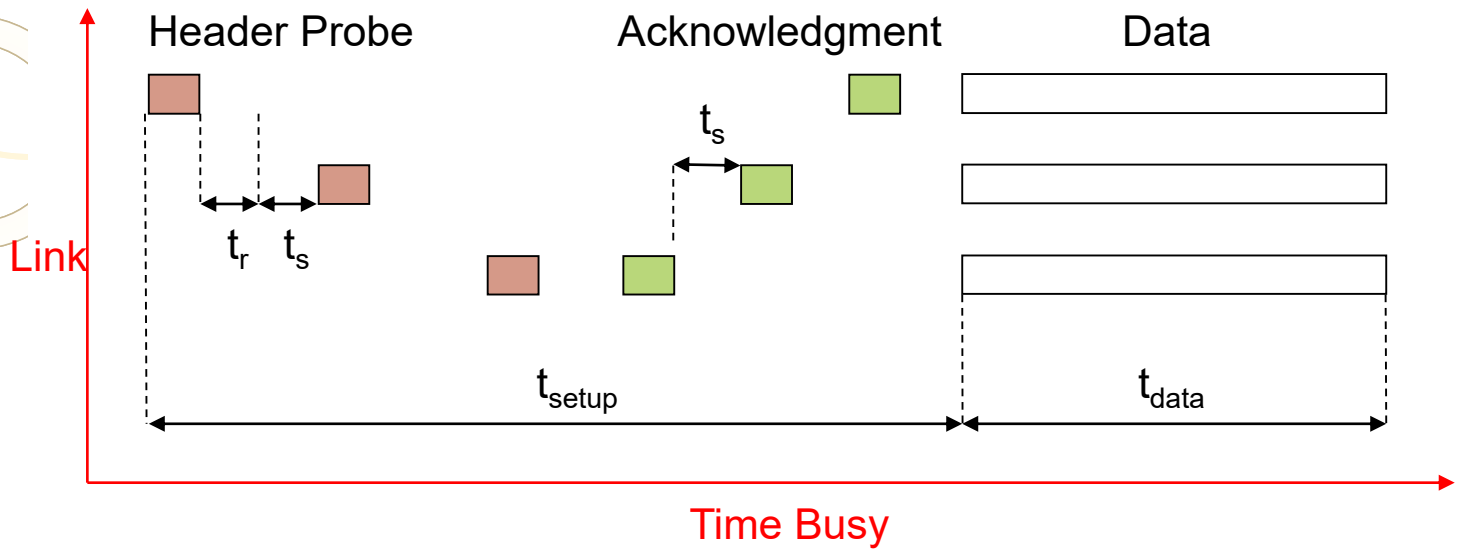
Network Interface

NoC Switching

- Switching techniques define the way and time of connections between input and output ports inside a switch.
- Circuit switched networks reserve a physical path before transmitting the data packets
- Packet switched networks transmit the packets without reserving the entire path.

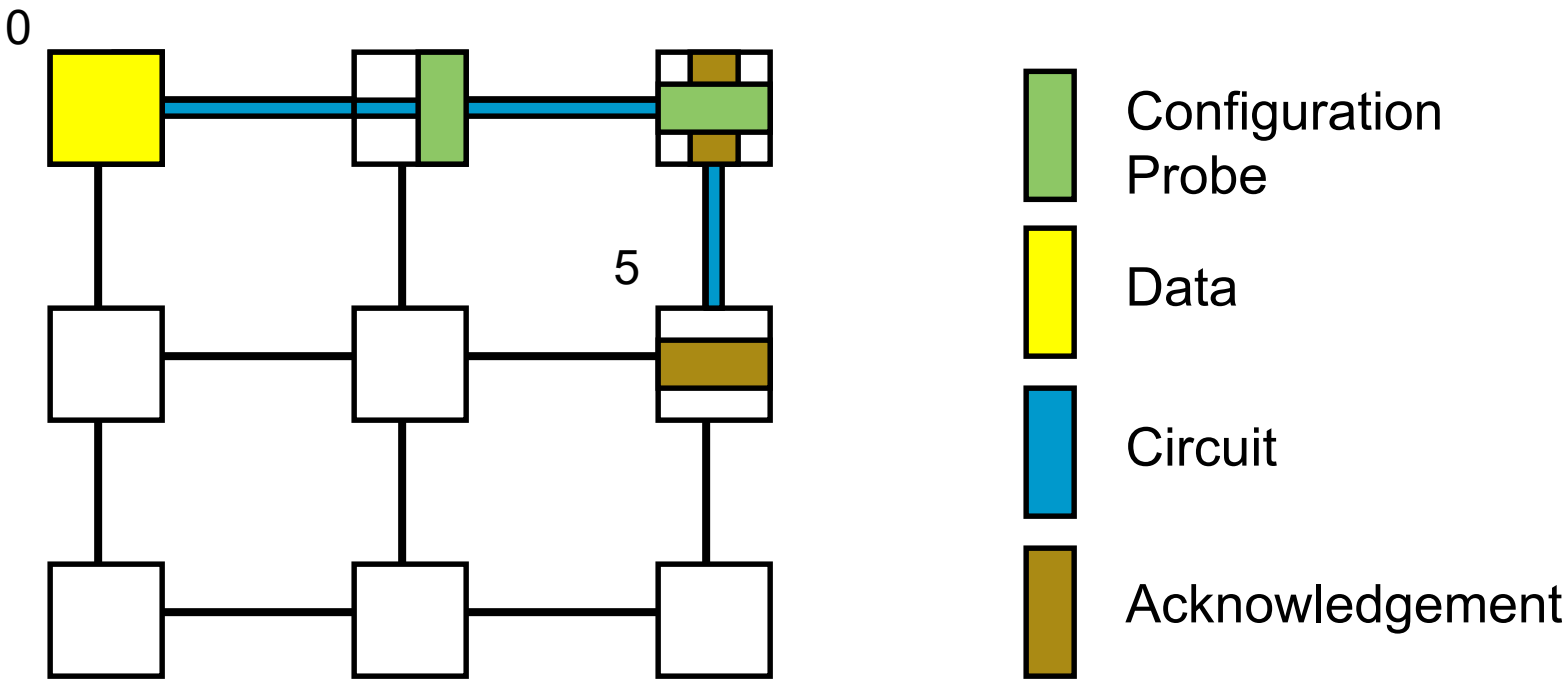


Circuit Switching



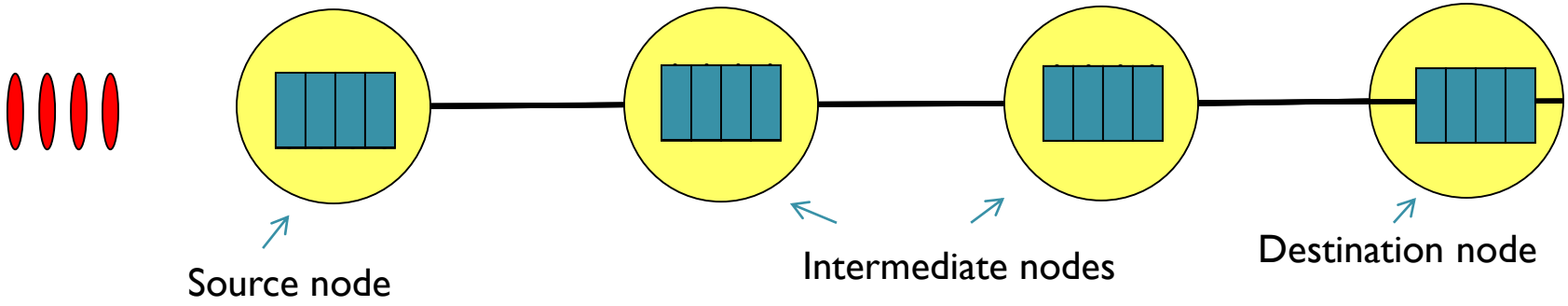
- Hardware path setup by a *routing header* or *probe*
- End-to-end acknowledgment initiates transfer at full hardware bandwidth

Circuit Switching Example



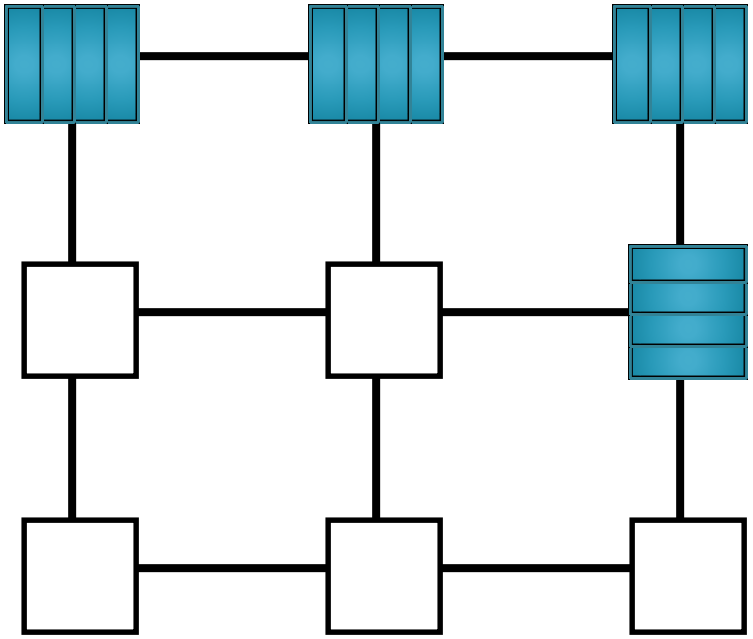
- Significant latency overhead prior to data transfer
- Other requests forced to wait for resources

Store & Forward Switching



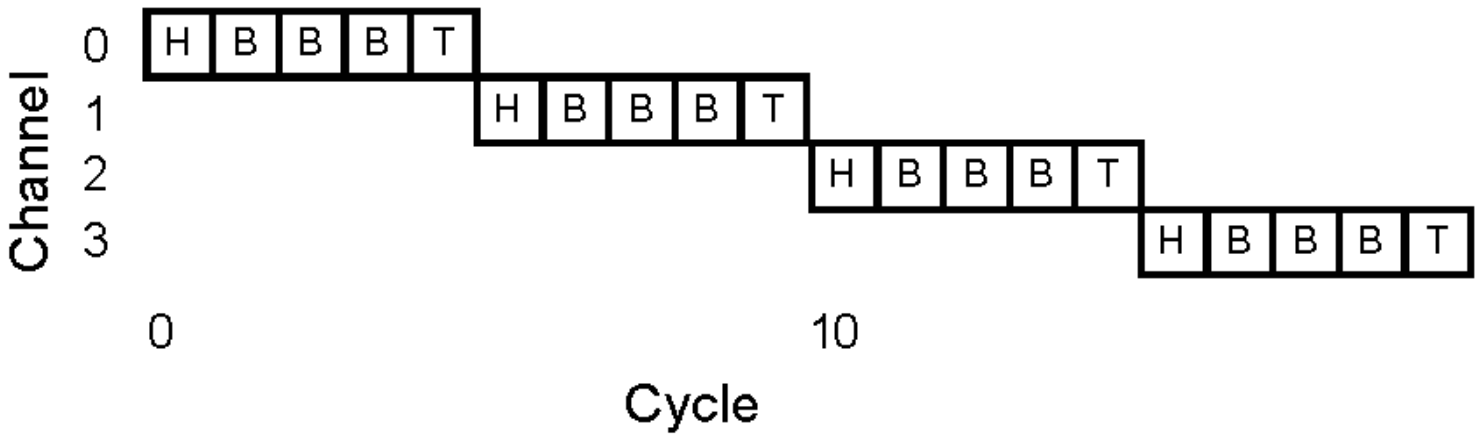
- Each node along a route waits until a packet is completely received (stored) and then the packet is forwarded to the next node
- Two resources are needed
  - Packet-sized buffer in the switch
  - Exclusive use of the outgoing channel

Store & Forward Switching Example



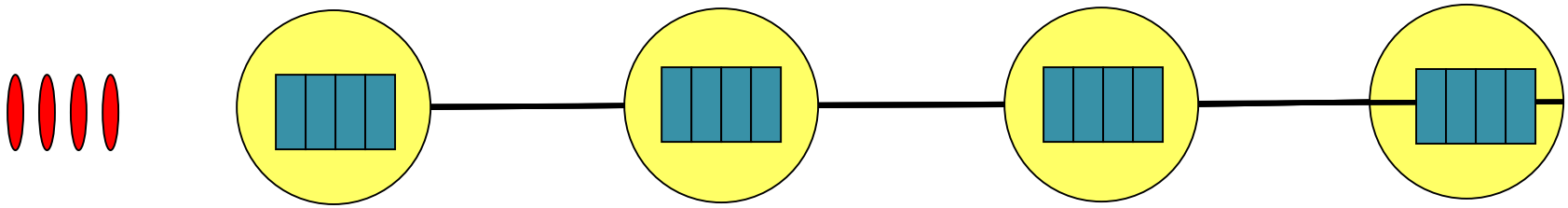
- High per-hop latency
- Larger buffering required

Store & Forward Switching

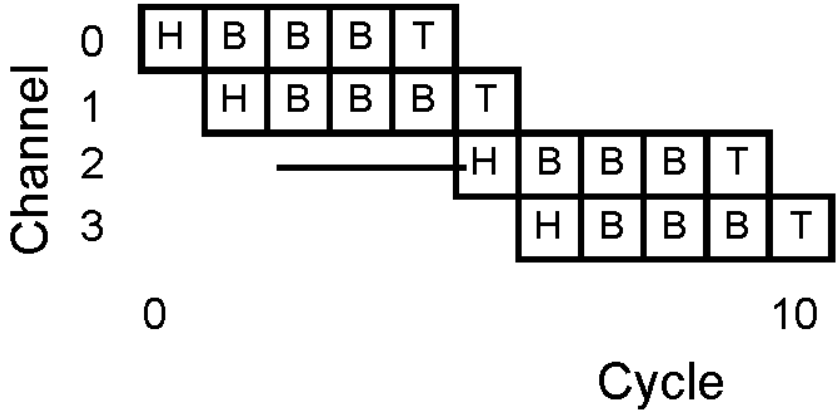


- Advantage
  - While waiting to acquire resources, no channels are being held idle
- Disadvantage
  - Requires a large amount of buffer space at each node
  - Very high latency

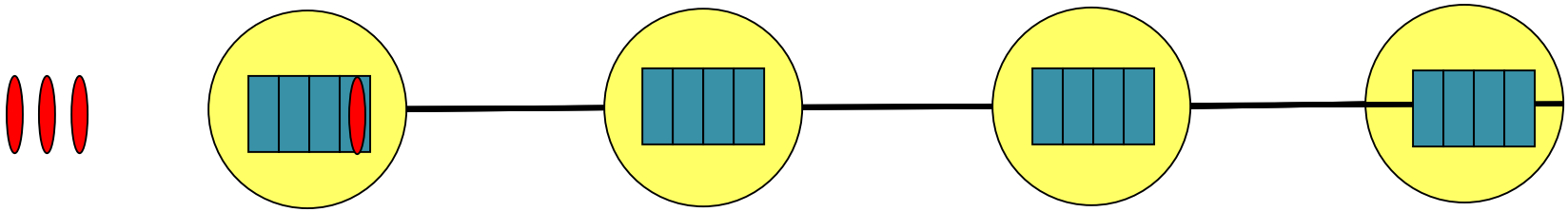
Virtual Cut-through Switching



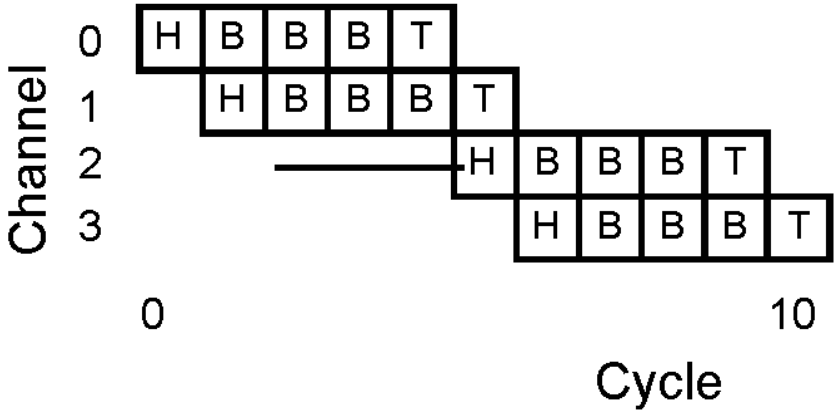
- Transmission on the next channel starts directly when the new header flit is received
- Channel is released after tail flit



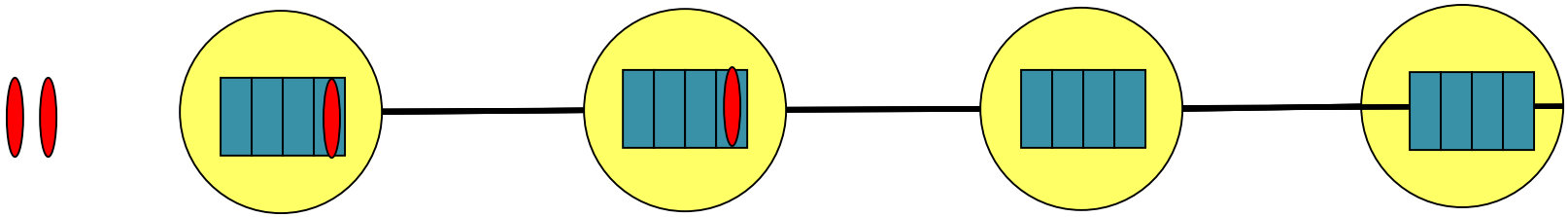
Virtual Cut-through Switching



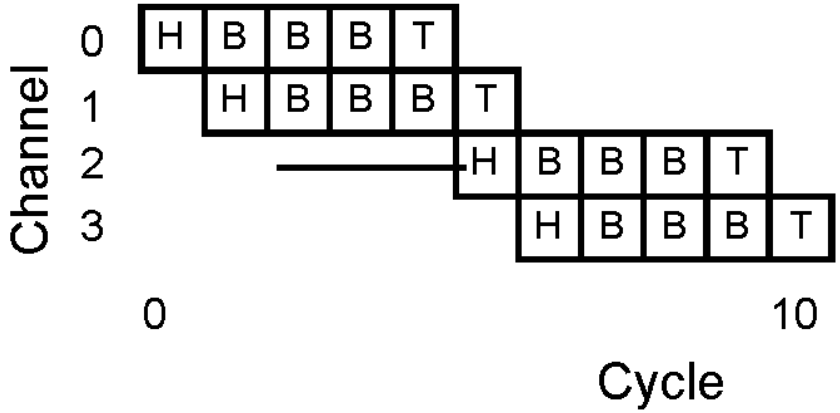
- Transmission on the next channel starts directly when the new header flit is received
- Channel is released after tail flit



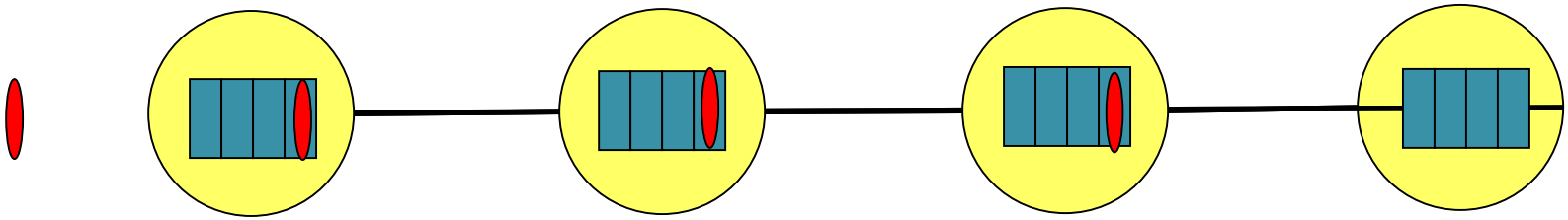
Virtual Cut-through Switching



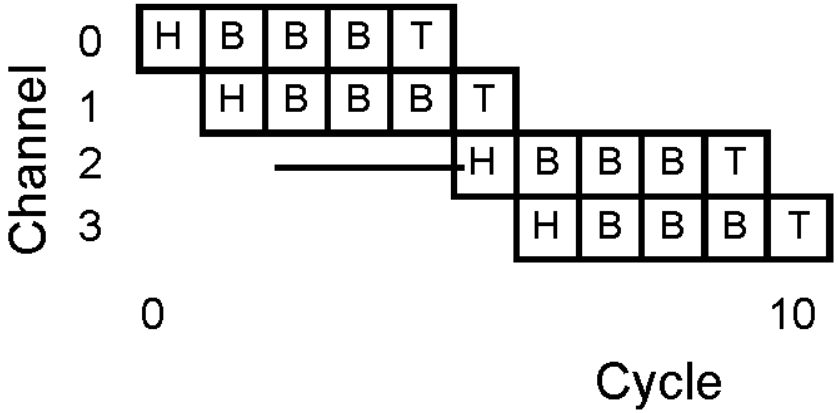
- Transmission on the next channel starts directly when the new header flit is received
- Channel is released after tail flit



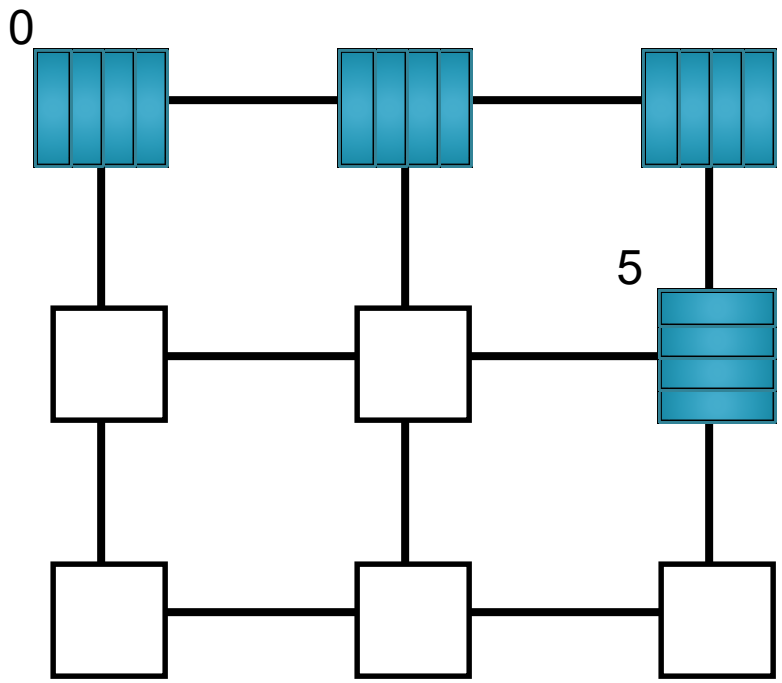
Virtual Cut-through Switching



- Transmission on the next channel starts directly when the new header flit is received
- Channel is released after tail flit

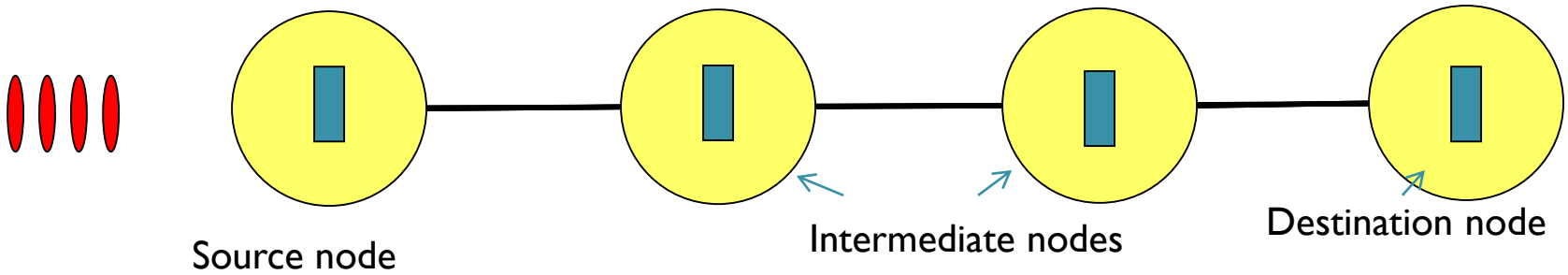


Virtual Cut-through Switching Example



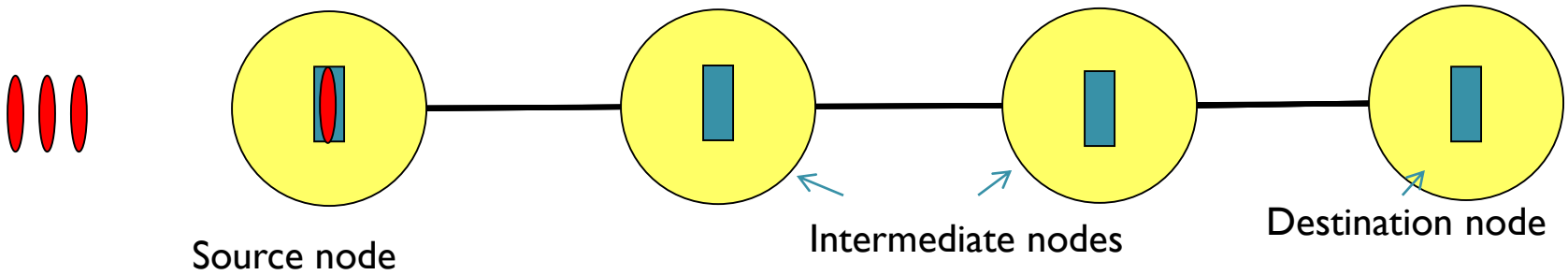
- Lower per-hop latency
- Larger buffering required

Wormhole Switching



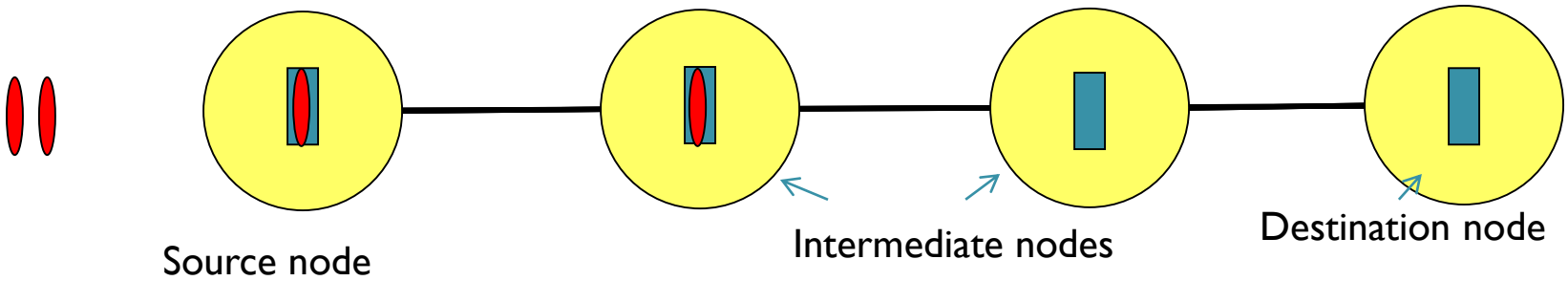
- Large packets are divided into small flits
- An entire packet need not be buffered to move on to the next node, increasing throughput.
- More efficient use of buffers than virtual cut-through
- Bandwidth and Channel allocation are decoupled

Wormhole Switching



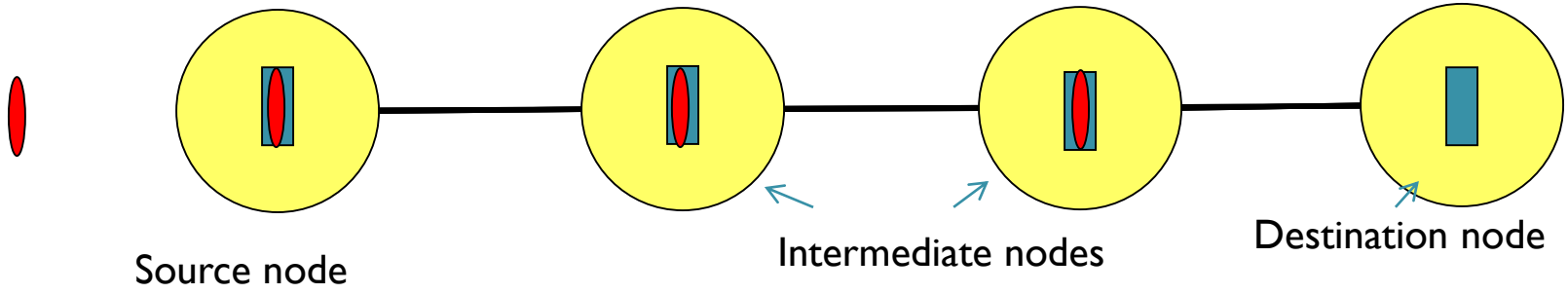
- Large packets are divided into small flits
- An entire packet need not be buffered to move on to the next node, increasing throughput.
- More efficient use of buffers than virtual cut-through
- Bandwidth and Channel allocation are decoupled

Wormhole Switching



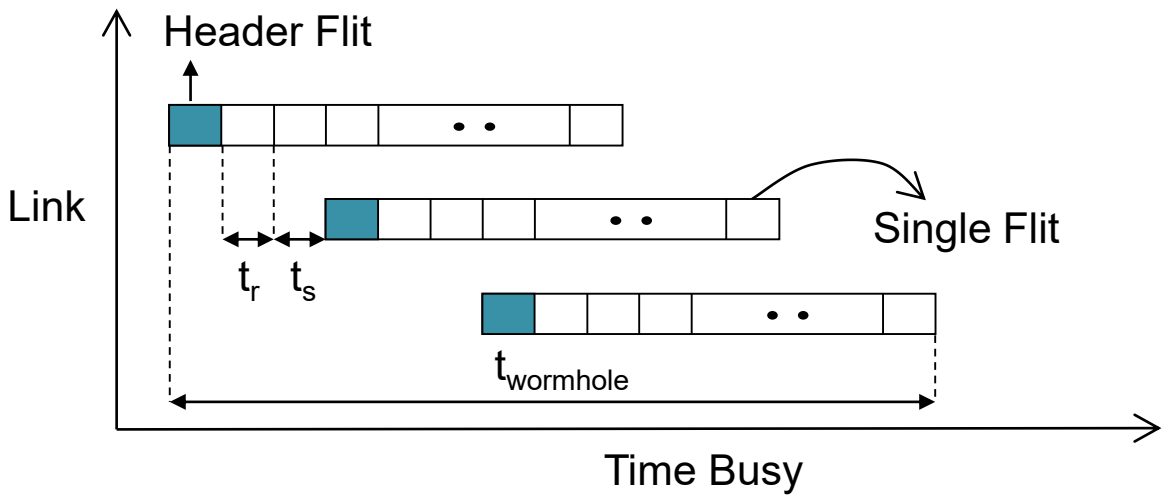
- Large packets are divided into small flits
- An entire packet need not be buffered to move on to the next node, increasing throughput.
- More efficient use of buffers than virtual cut-through
- Bandwidth and Channel allocation are decoupled

Wormhole Switching



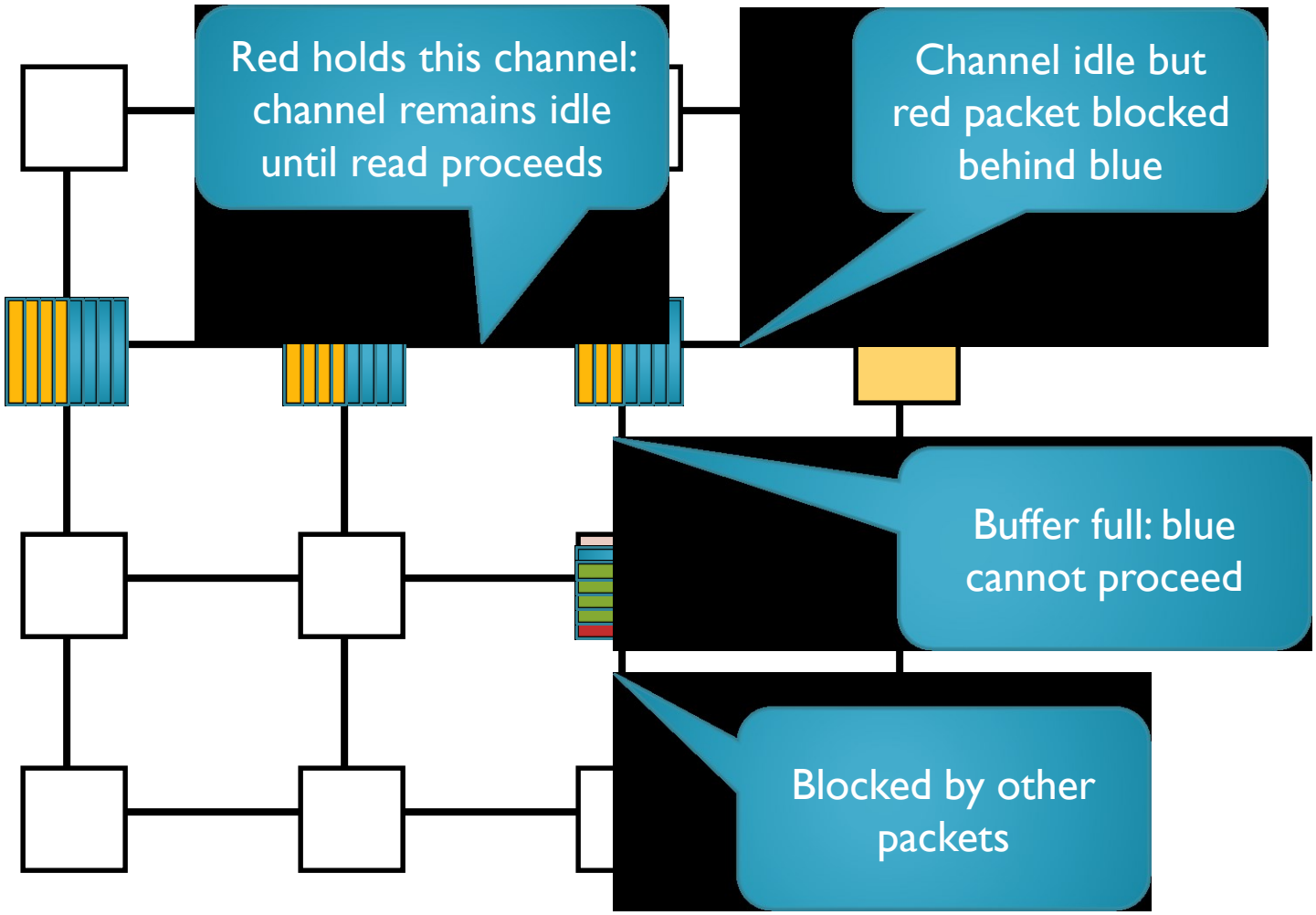
- Large packets are divided into small flits
- An entire packet need not be buffered to move on to the next node, increasing throughput.
- More efficient use of buffers than virtual cut-through
- Bandwidth and Channel allocation are decoupled

Wormhole Switching



- Message are pipelined, but buffer space is on the order of a few flits
- Small buffers + message pipelining → small compact switches/routers
- Messages cannot be interleaved over a channel: routing information is only associated with the header

Wormhole Example

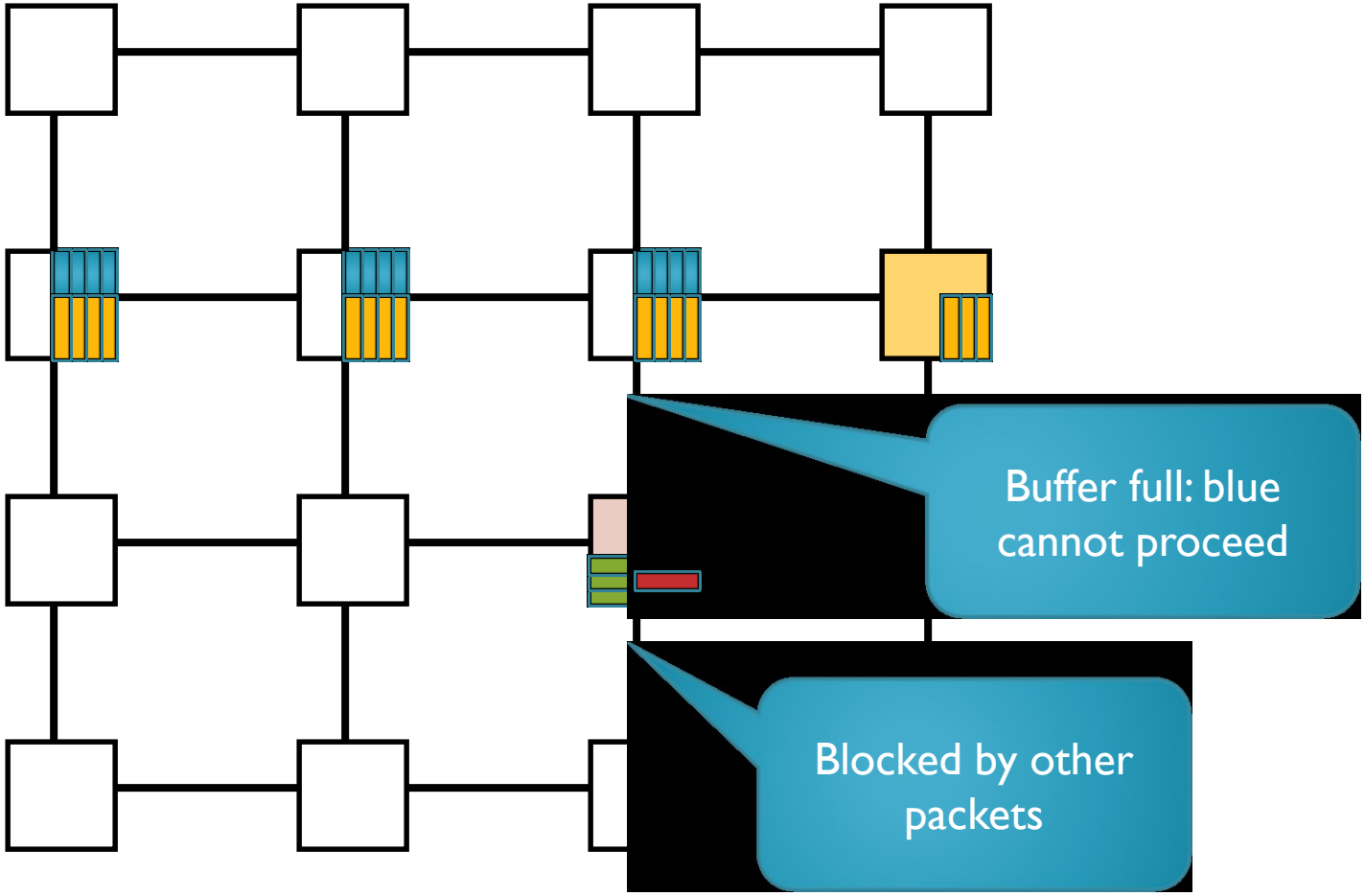


- 6 flit buffers/input port

Virtual Channel

- Virtual channels used to combat HOL block in wormhole
- Virtual channels: multiple flit queues per input port
  - Share same physical link (channel)
- Link utilization improved
  - Flits on different VC can pass blocked packet

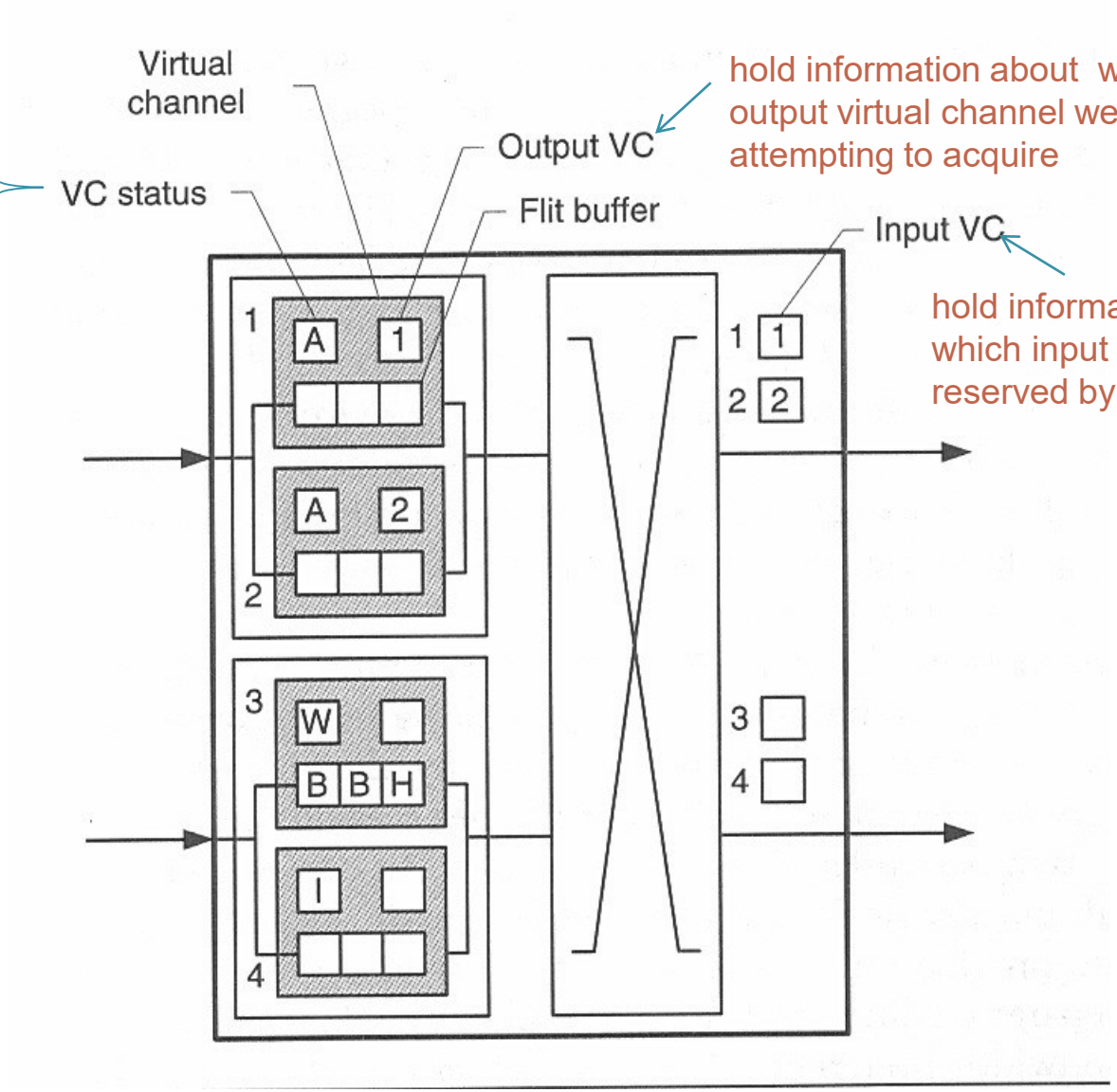
Virtual Channel Example



- 6 flit buffers/input port
- 3 flit buffers/VC

Virtual Channel Example

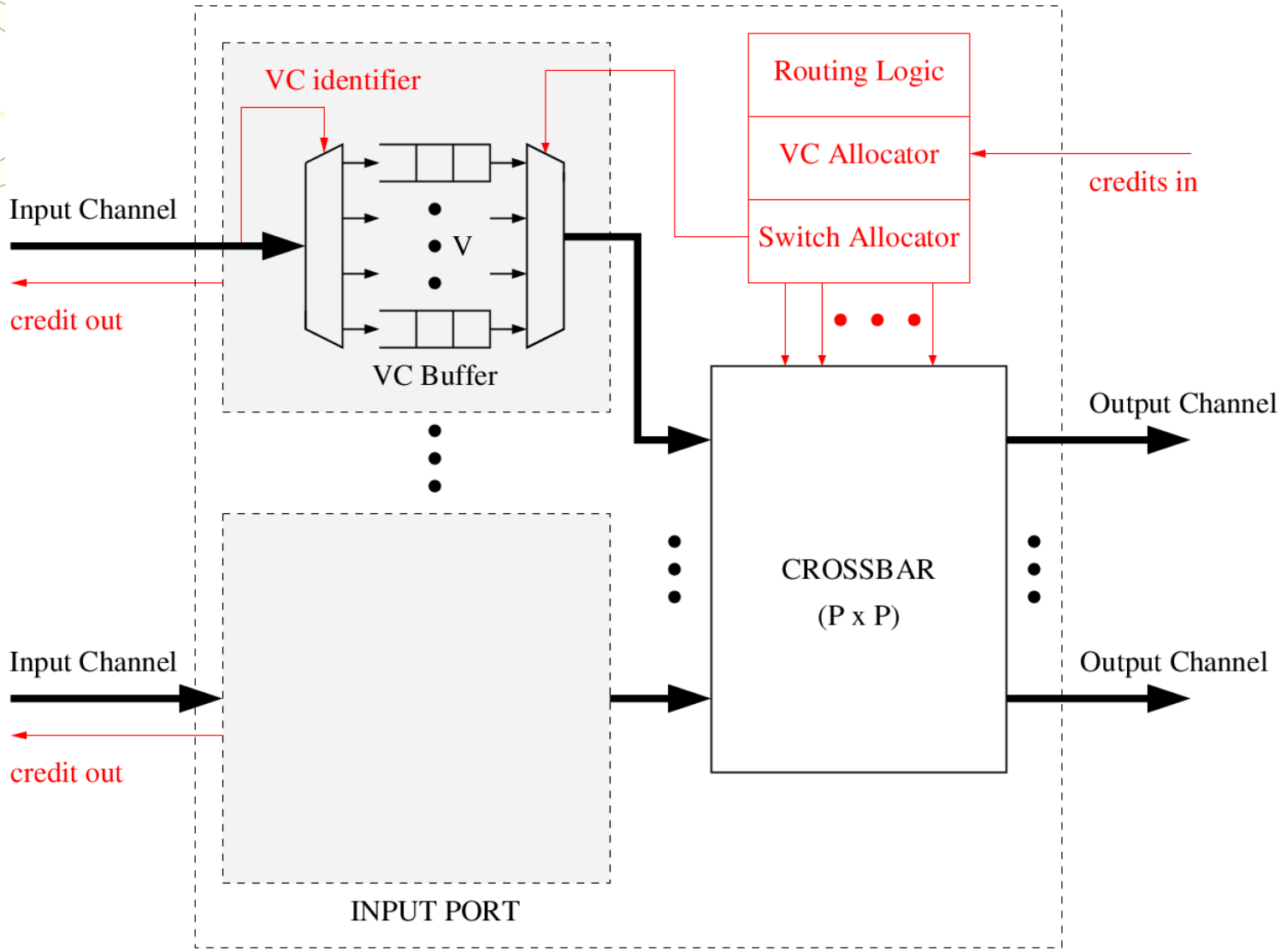
A: active
W: waiting
I: idle



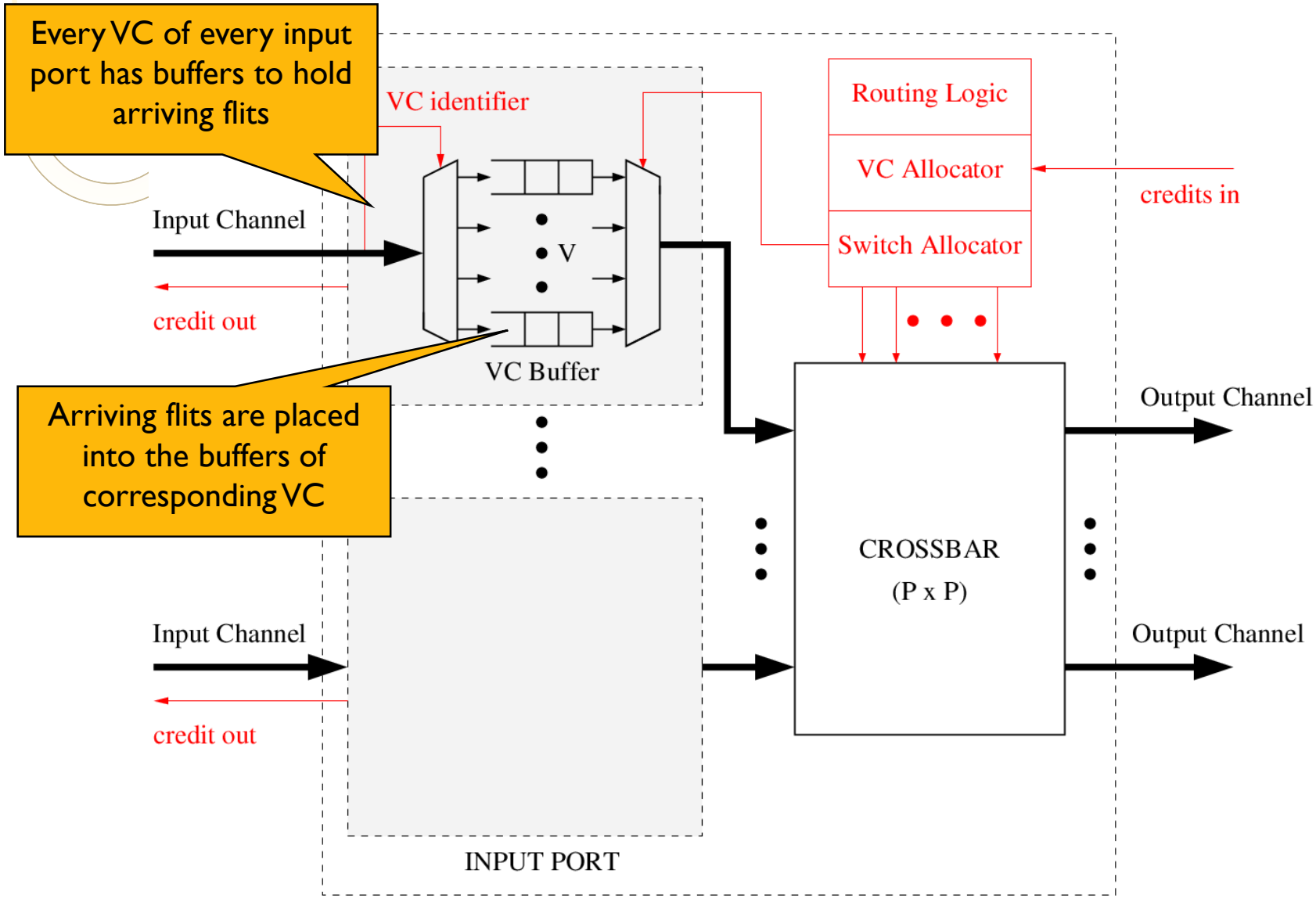
hold information about which output virtual channel we are attempting to acquire

hold information about which input VC it is reserved by

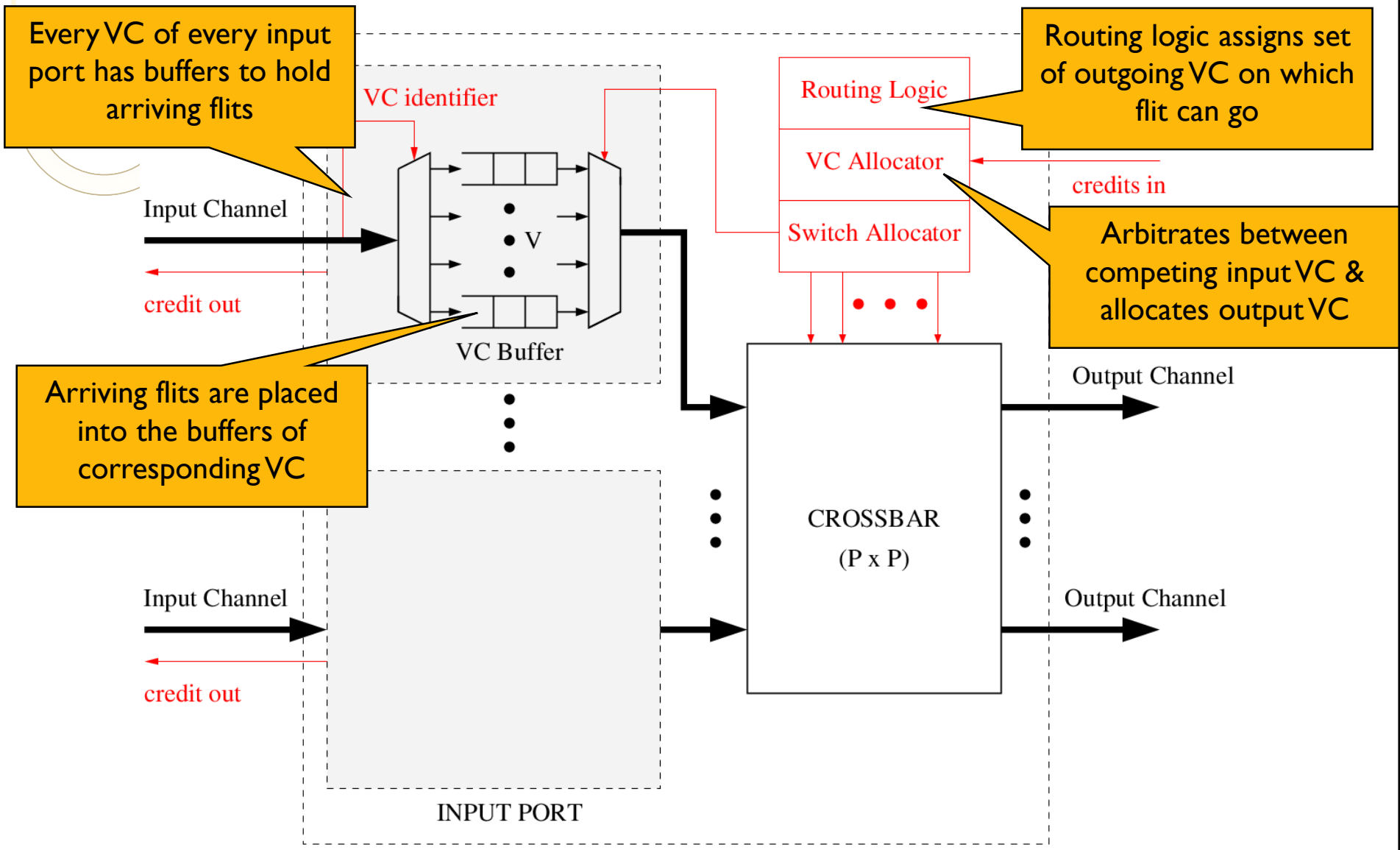
A Virtual Channel Router



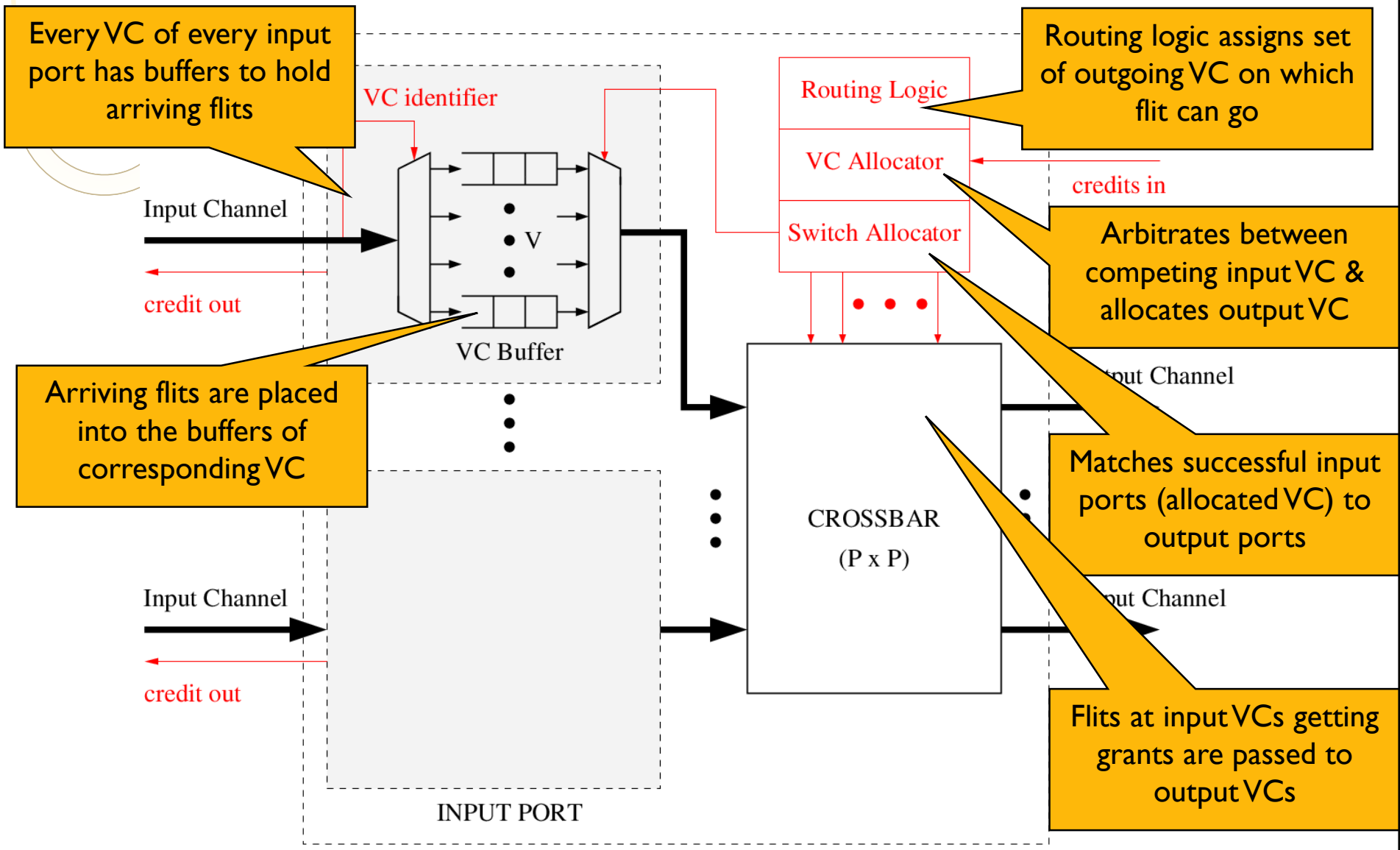
A Virtual Channel Router



A Virtual Channel Router

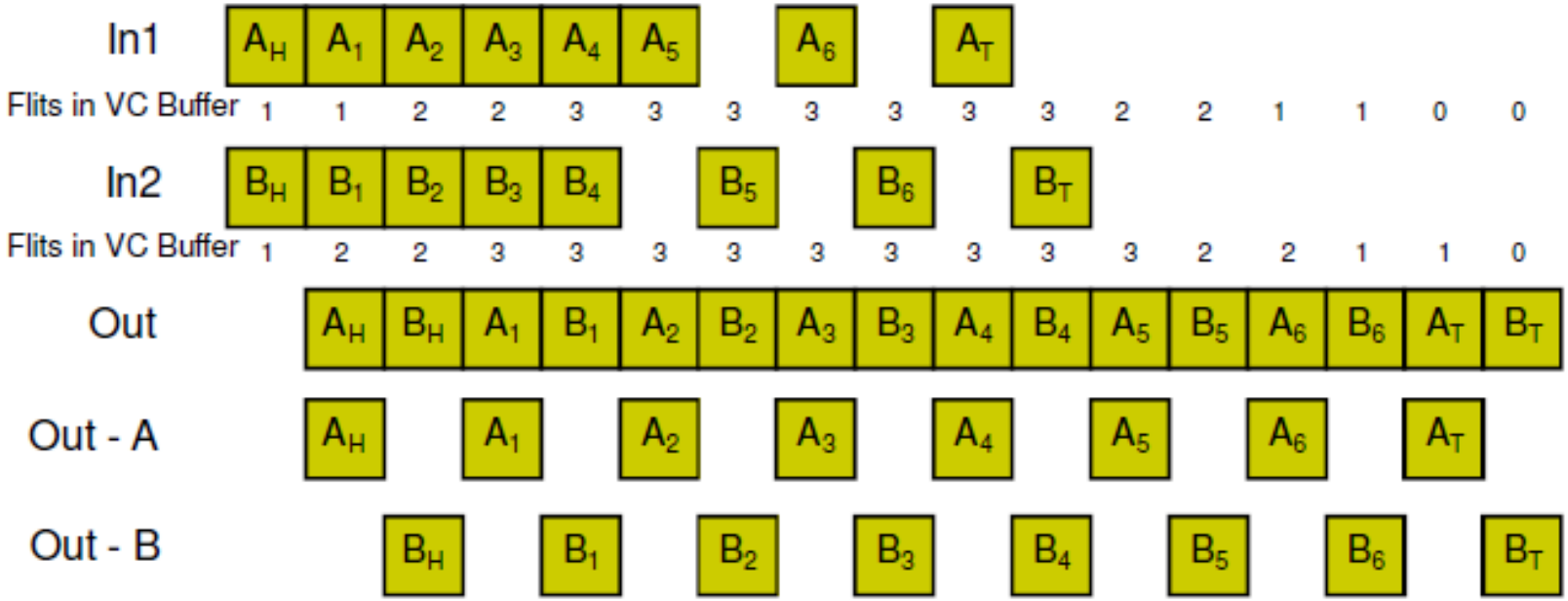


A Virtual Channel Router



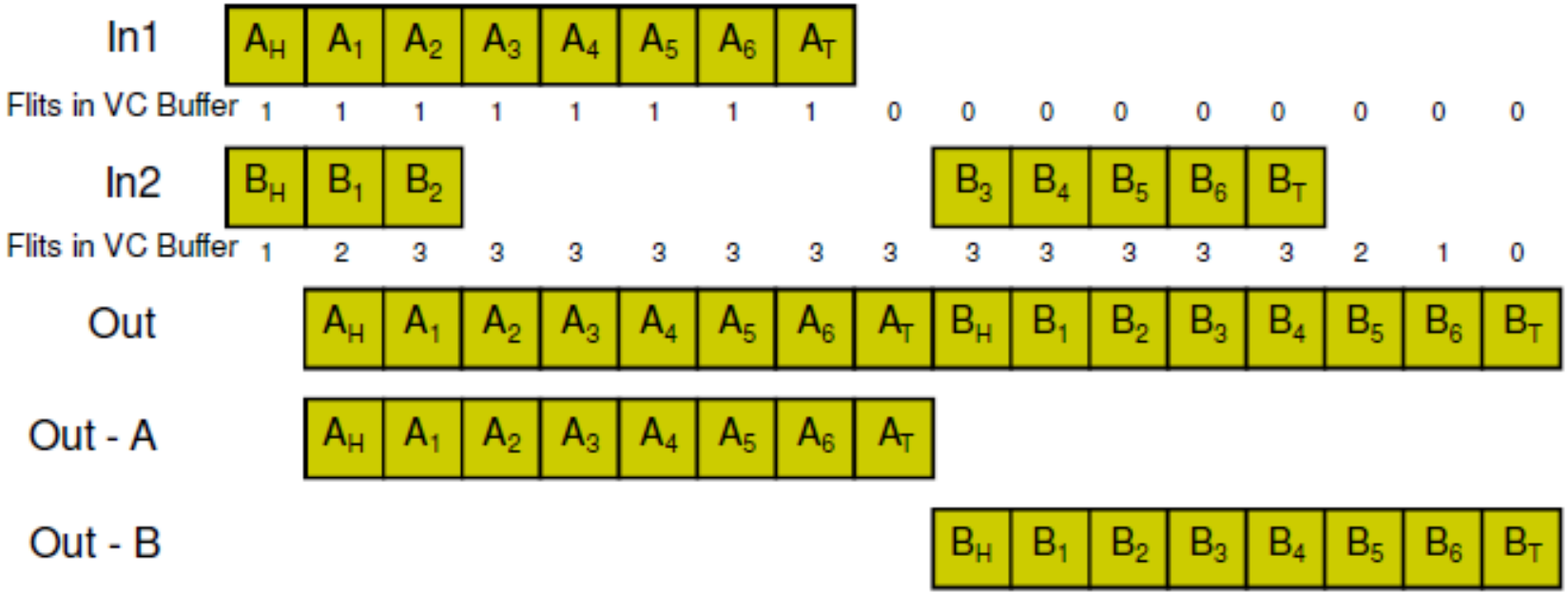
VC Arbitration: Fair Bandwidth

- The virtual channels interleave their flits
- This results in a high average latency



VC Arbitration: Winner-Take-All

- A winner-take all arbitration reduces the average latency with no throughput penalty



Summary of Switching Techniques

Switching Technique	Communication Entity	Path Reservation	Buffer Size	Resource Utilization
Circuit Switching	Flit	Yes	Small	Poor
SAF Switching	Packer	No	Large	Good
VCT Switching	Packet	No	Large	Good
Wormhole Switching	Flit	Yes	Small	Moderate

Summary of switching techniques

Break + Qs

Part II: NoC Building Blocks

Topology

Routing Algorithms

Routing Mechanisms

Switching

Flow Control

Router Architecture

Network Interface

Flow Control (FC)

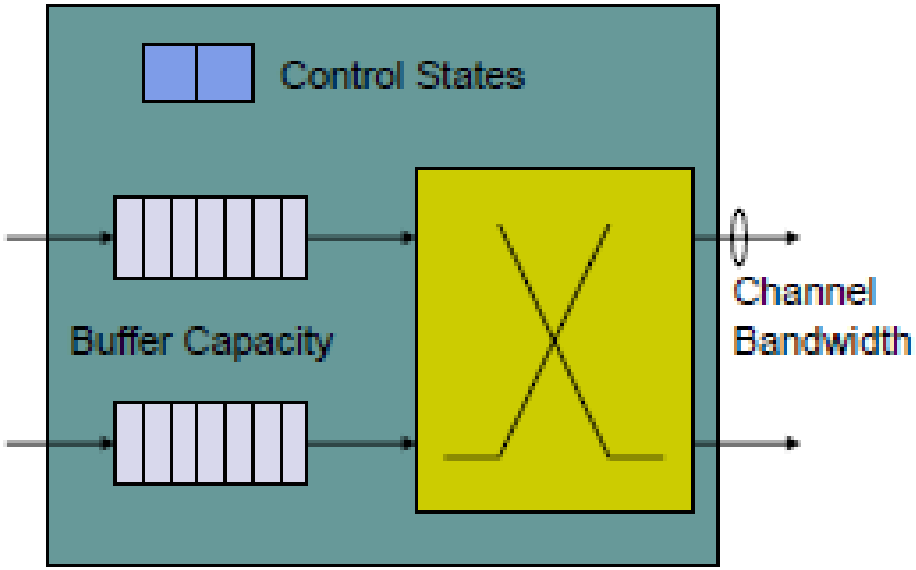
FC determines (1) how resources (Buffers and channel bandwidth) are allocated and (2) how packet collisions over resources are resolved.

- Goal is to use resources as efficient as possible to allow a <u>high throughput</u>

- A <u>resource collision</u> occurs when a packet P is unable to proceed because some resource it needs is held by another packet.

Node Resources

1. **Control State**
   - Tracks the resources allocated to the packet in the node and the state of the packet
2. **Buffer**
   - Packet is stored in a buffer before it is send to next node



3. **Bandwidth**
   - To travel to the next node bandwidth has to be allocated for the packet

Flow Control

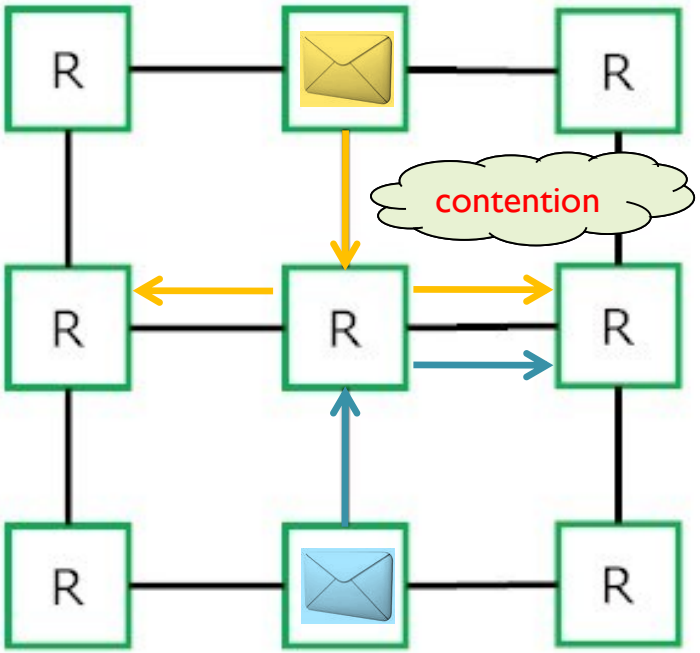
NoC Flow Control can be divided into:

1. Bufferless flow control
   - Packets are either *dropped* or *misrouted*

2. Buffered flow control (covered here)
   - Packets that cannot be routed via the desired channel are stored in buffers
     - Stop-Go,
     - ACK/NACK,
     - Credit-Based

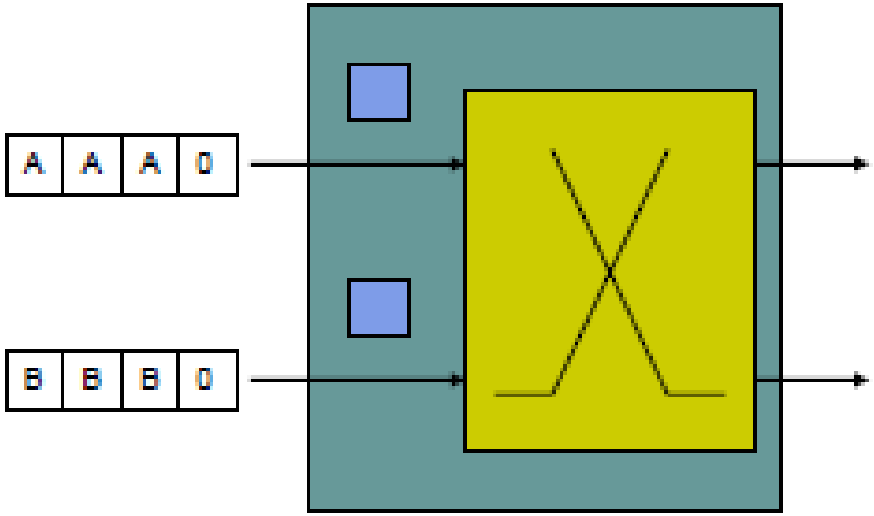
Bufferless flow Control

- Flits can't wait in routers.
- Contention is handled by:
  - Dropping and retransmitting from the source.
  - Deflecting to a free output.



Bufferless Flow Control

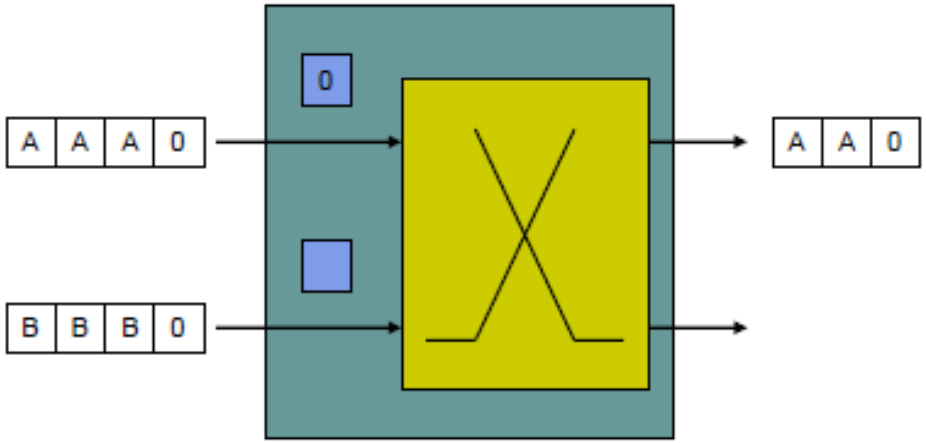
- No buffers mean **less implementation cost**
- If more than one packet shall be routed to the same output, one has to be
  - Misrouted or
  - Dropped



Example: 2 packets A and B (consisting of several flits) arrive at a network node

Bufferless Flow Control

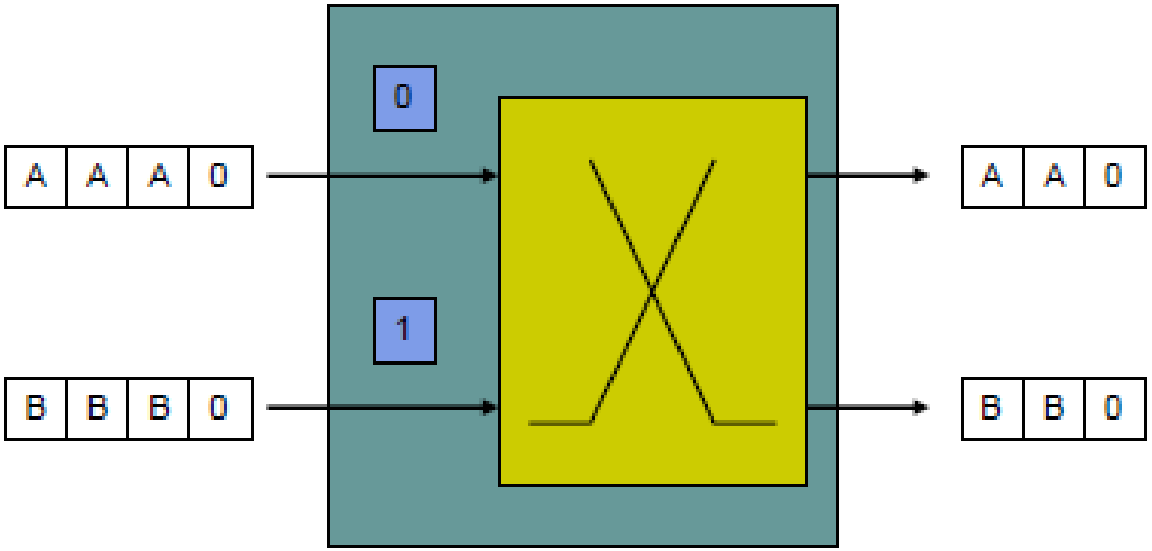
- Packet B is **dropped** and must be resended



- But, there must be a **protocol** that informs the sending node that the packet has been dropped
  - Example: Resend after no ACK has been received within a given time

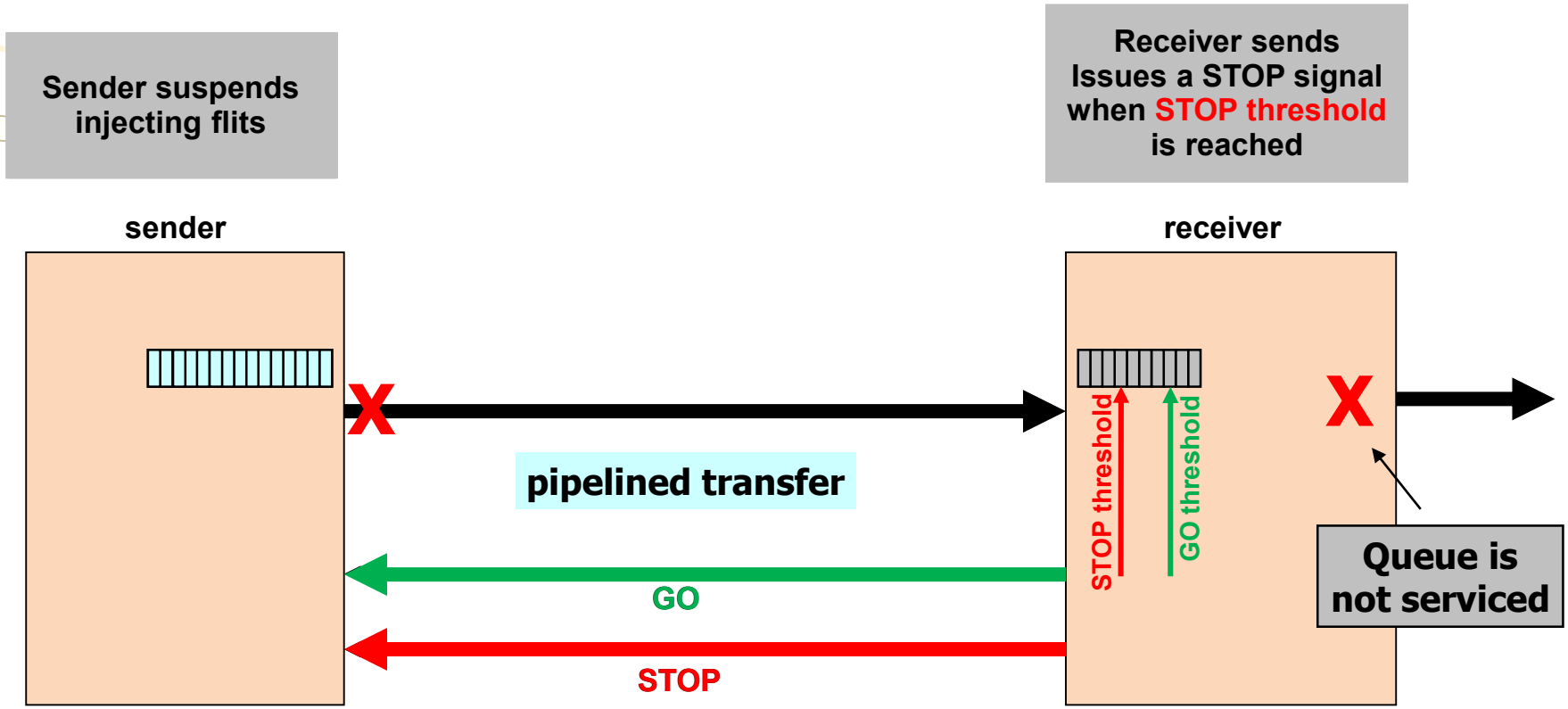
Bufferless Flow Control

- Packet B is **misrouted**

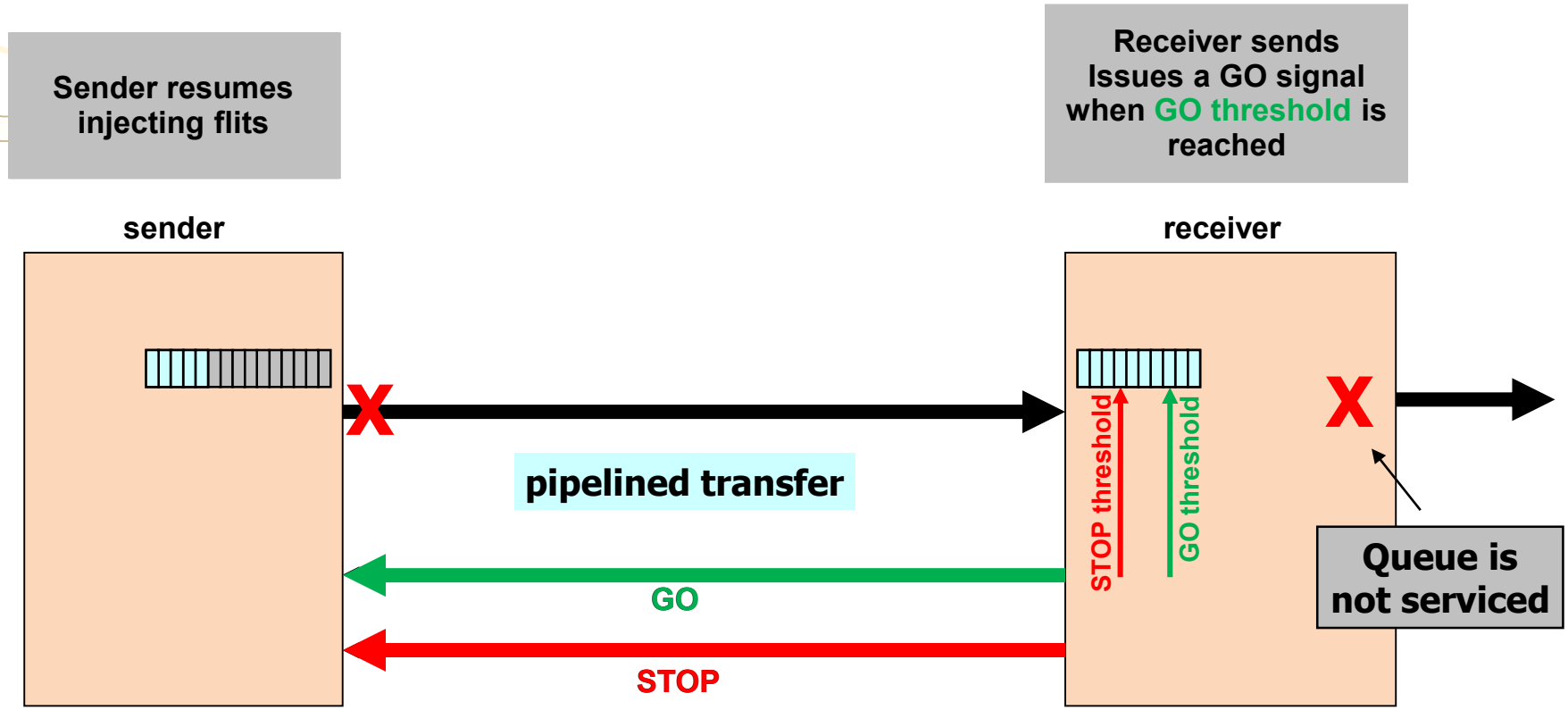


- No further action is required here, but at the receiving node packets have to be **sorted** into original order

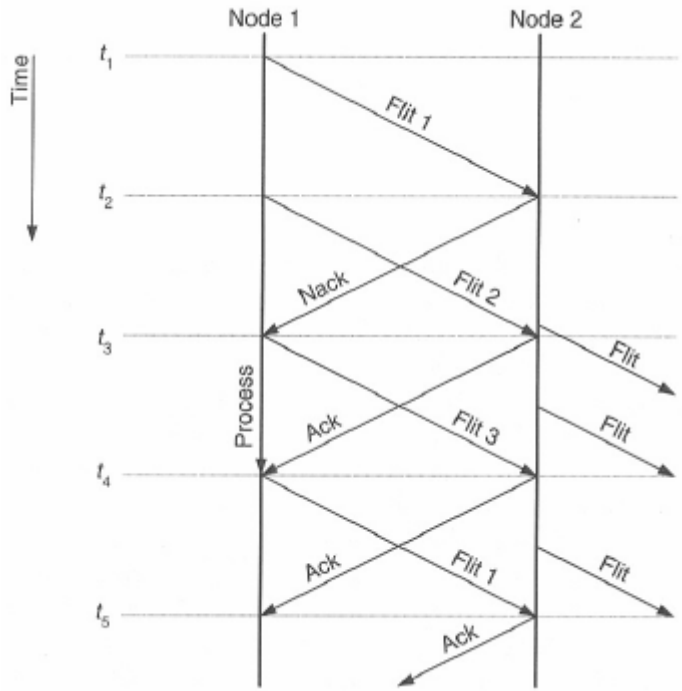
Stop-Go Flow Control



Stop-Go Flow Control



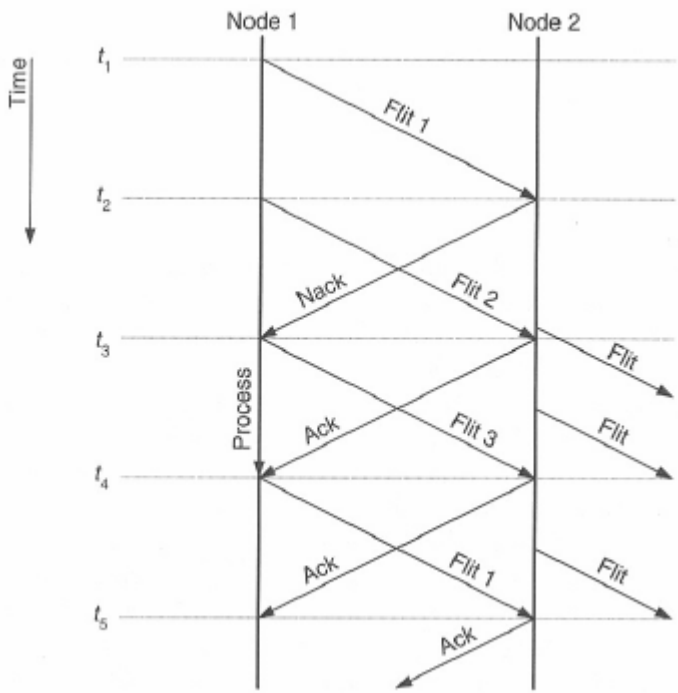
Ack/Nack Flow Control



- Upstream node sends packets without knowing, if there are free buffers in the downstream node.

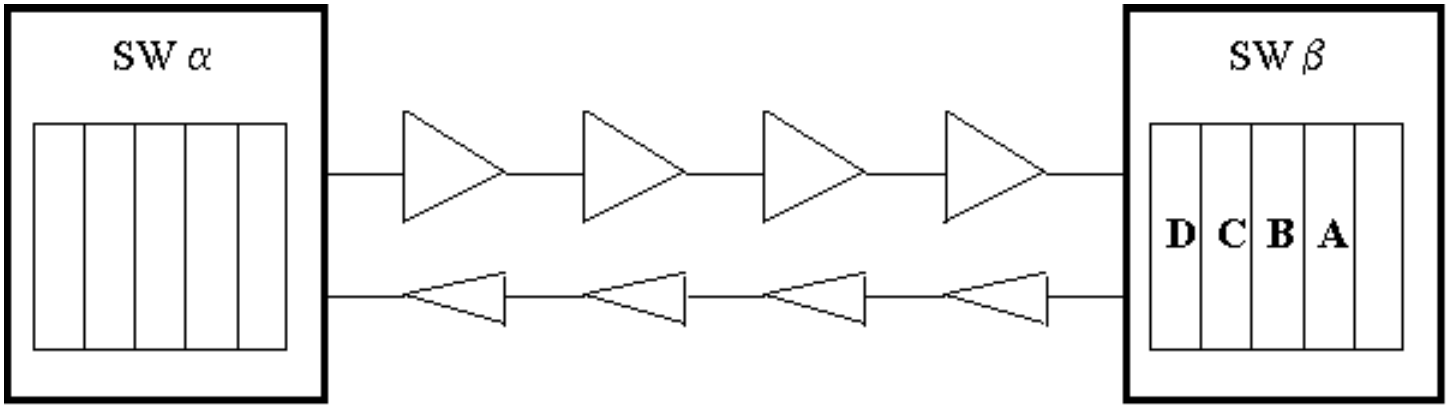
Ack/Nack Flow Control

- If there is no buffer available:
  - the downstream node sends ***Nack*** and drops the flit
  - the flit must be resent
  - flits must be reordered at the downstream node



- If there is a buffer available:
  - the downstream node sends ***Ack*** and stores the flit in a buffer

ACK/NACK



Transmission

ACK and buffering

NACK

ACK/NACK propagation

Memory deallocation

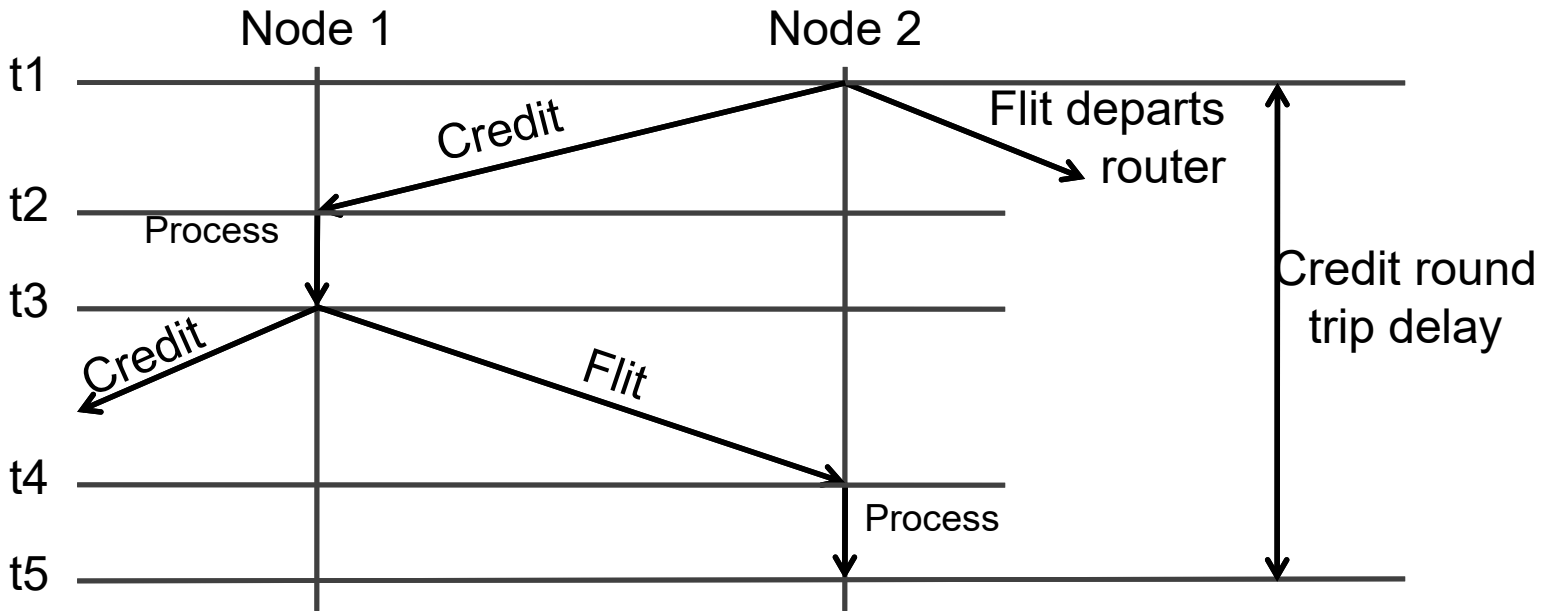
Retransmission

Go-back-N

Credit-Based Flow Control

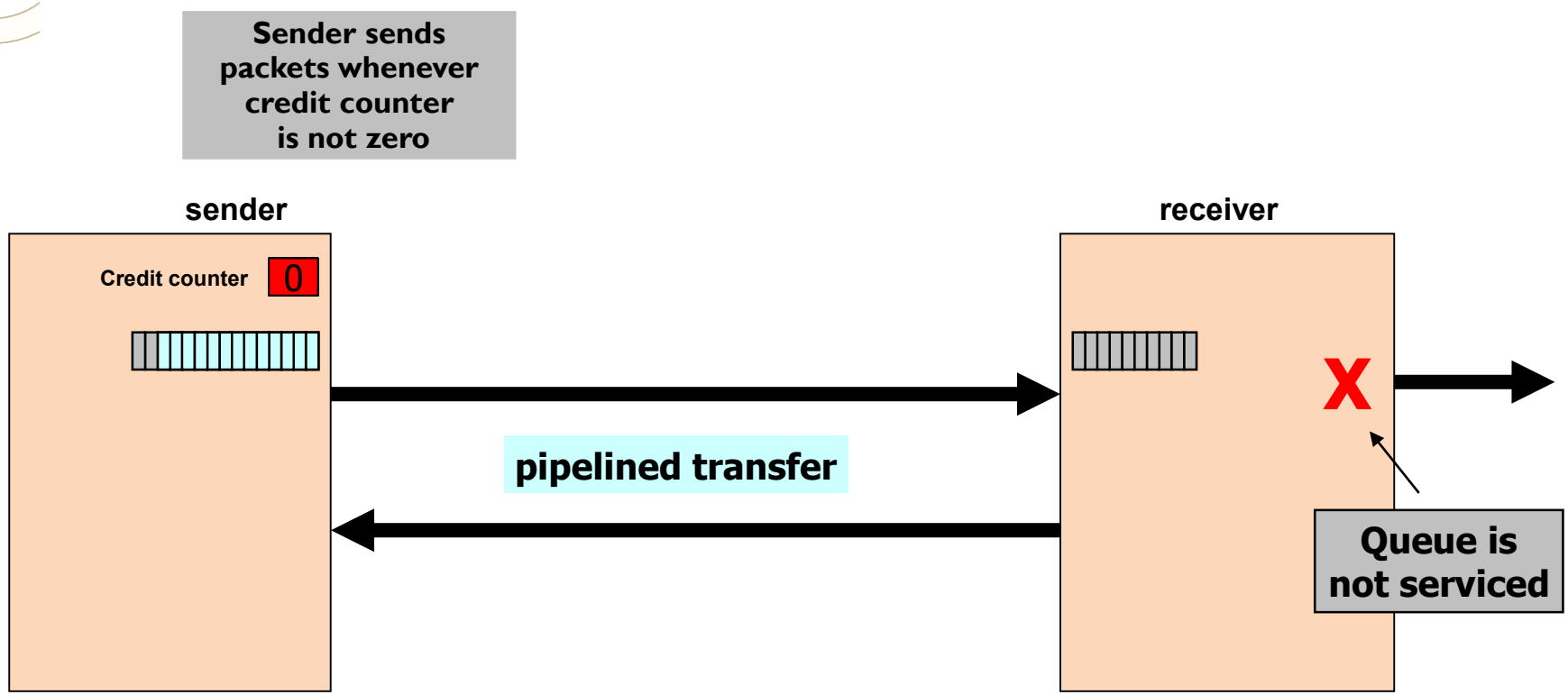
- Upstream router stores credit counts for each downstream VC
- Upstream router
  - When flit forwarded
    - Decrement credit count
  - Count == 0, buffer full, stop sending
- Downstream router
  - When flit forwarded and buffer freed
    - Send credit to upstream router
    - Upstream increments credit count

Credit Timeline

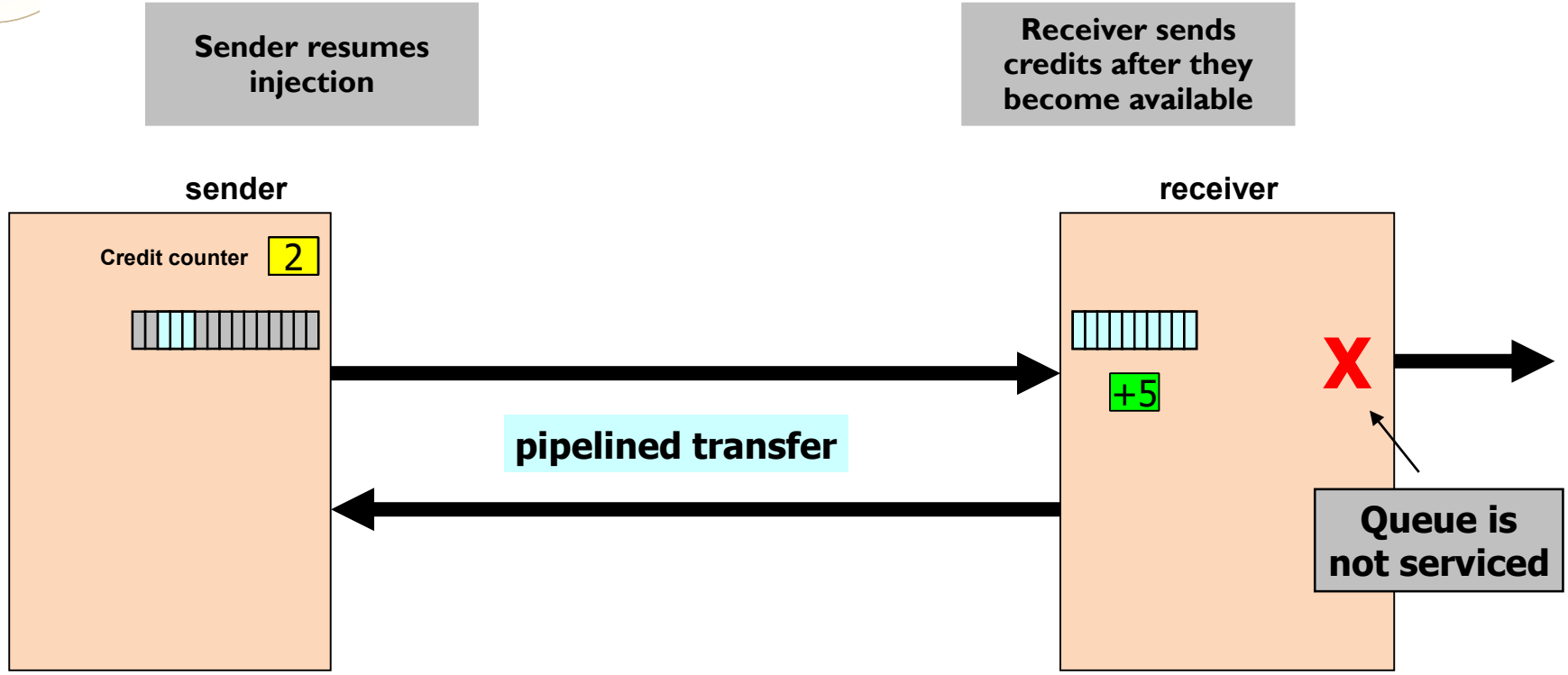


- Round-trip credit delay:
  - Time between when buffer empties and when next flit can be processed from that buffer entry
  - If only single entry buffer, would result in significant throughput degradation
  - Important to size buffers to tolerate credit turn-around

Credit-Based Flow Control in action



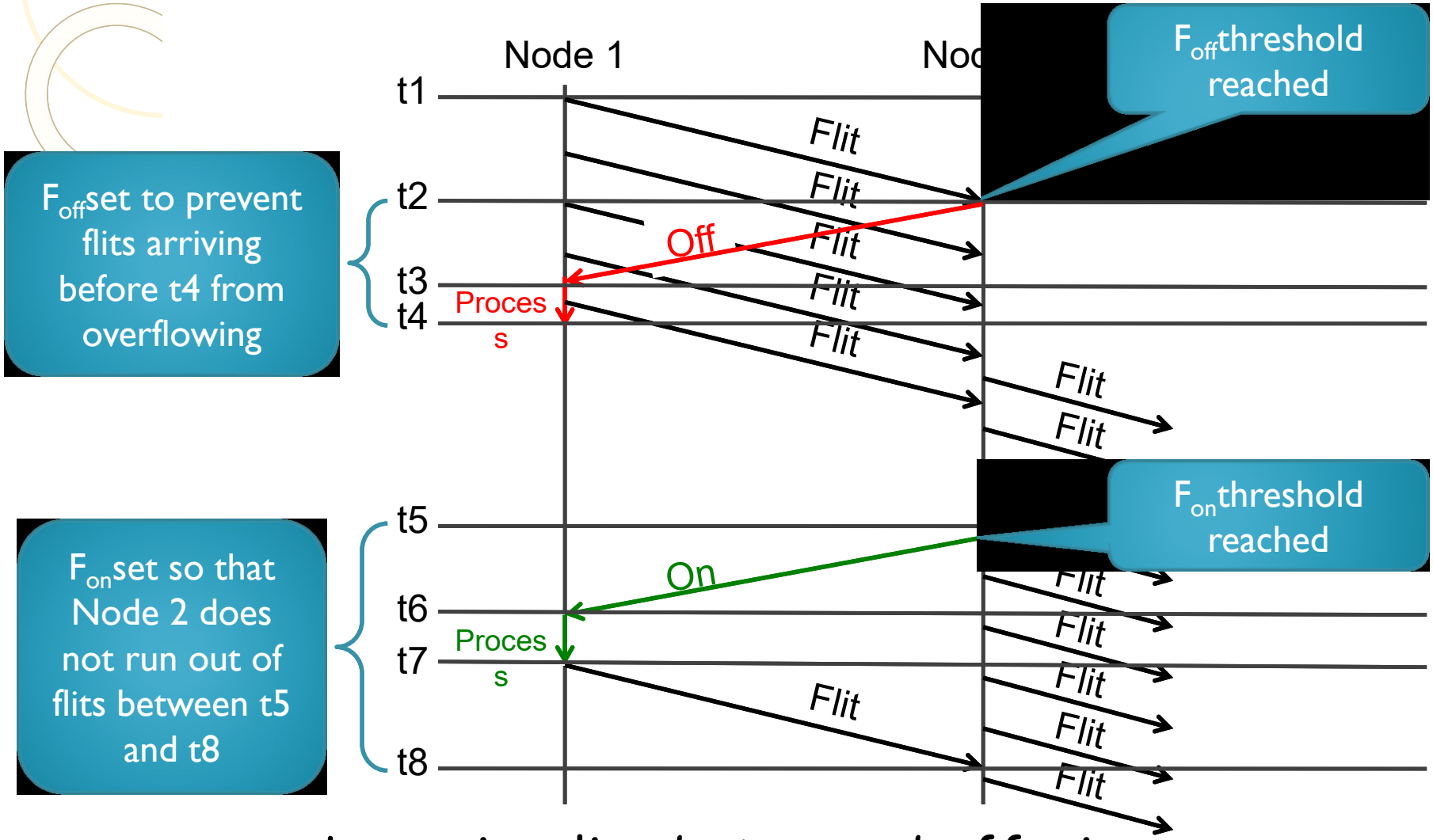
Credit-Based Flow Control in action



On-Off (stall-go) Flow Control

- Credit: requires upstream signaling for every flit
- On-off: decreases upstream signaling
- Off signal
  - Sent when number of free buffers falls below threshold F_{off}
- On signal
  - Send when number of free buffers rises above threshold F_{on}

On-Off Timeline



- Less signaling but more buffering
  - On-chip buffers more expensive than wires

Summary of FC

- On-chip networks require techniques with lower buffering requirements
  - Wormhole or Virtual Channel flow control
- Dropping packets unacceptable in on-chip environment
- Complexity of flow control impacts router microarchitecture

Summary of FC

- Ack/Nack: is rarely used because of its buffer and bandwidth inefficiency.
- Credit-based: Used in systems with small numbers of buffers.
- On/Off : Used in systems that have large numbers of flit buffers.

Part II: NoC Building Blocks

Topology

Routing

Switching

Virtual Channels

Flow Control

Router Architecture

Network Interface

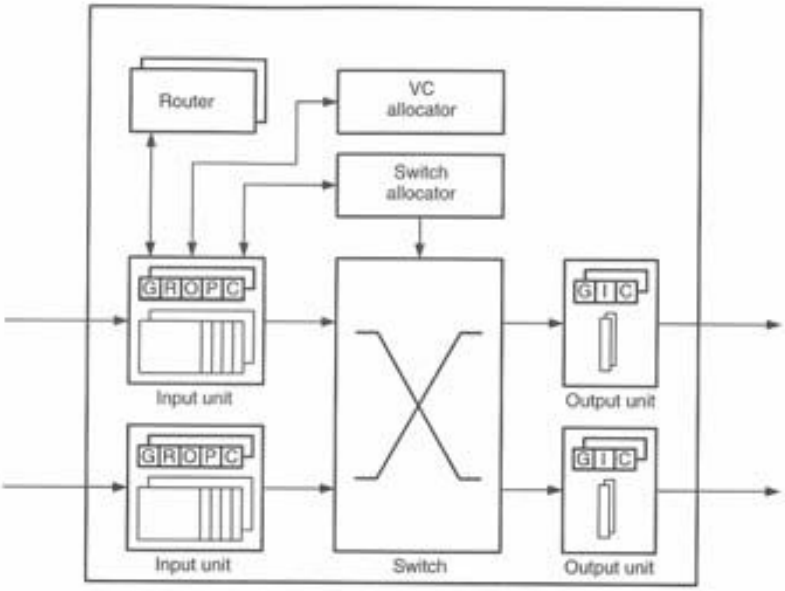
Typical Virtual Channel Router

A router functional blocks can be divided into:

1. <u>Data path</u>: handles storage and movement of a packets payload
   - Input buffers , Switch, Output buffers
2. <u>Control path</u>: coordinating the movements of the packets through the resources of the datapath
   - Route Computation, VC Allocator, Switch Allocator

Typical Virtual Channel Router

- The input unit contains a set of flit buffers
- Maintains the state for each virtual channel
  - G = Global State
  - R = Route
  - O = Output VC
  - P = Pointers
  - C = Credits



Virtual Channel State Fields (Input)

Virtual channel state fields, represented by a 5-vector: GROPC.

Field	*Name*	*Description*
G	Global state	Either idle (I), routing (R), waiting for an output VC (V), active (A), or waiting for credits (C).
R	Route	After routing is completed for a packet, this field holds the output port selected for the packet.
O	Output VC	After virtual-channel allocation is completed for a packet, this field holds the output virtual channel of port R assigned to the packet.
P	Pointers	Flit head and tail pointers into the input buffer. From these pointers, we can also get an implicit count on the number of flits in the buffer for this virtual channel.
C	Credit count	The number of credits (available downstream flit buffers) for output virtual channel O on output port R.

Virtual Channel State Fields (Output)

Output virtual channel state fields, represented by a 3-vector: GIC.

Field	*Name*	*Description*
G	Global state	Either idle (I), active (A), or waiting for credits (C).
I	Input VC	Input port and virtual channel that are forwarding flits to this output virtual channel.
C	Credit count	Number of free buffers available to hold flits from this virtual channel at the downstream node.

Packet Rate and Flit Rate

- The control of the router operates at two distinct frequencies
  - Packet Rate (performed once per packet)
    - Route computation
    - Virtual-channel allocation
  - Flit Rate (performed once per flit)
    - Switch allocation
    - Pointer and credit count update

The Router Pipeline

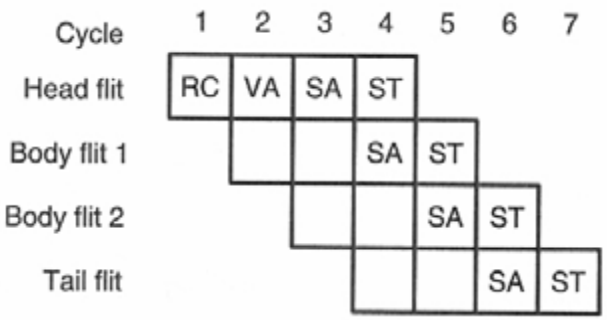
Cycle	1	2	3	4	5	6	7
Head flit	RC	VA	SA	ST			
Body flit 1				SA	ST		
Body flit 2					SA	ST	
Tail flit						SA	ST

No pipeline stalls

The Router Pipeline

- A typical router pipeline includes the following stages:
  - **RC** (Routing Computation)
  - **VC** (Virtual Channel Allocation)
  - **SA** (Switch Allocation)
  - **ST** (Switch Traversal

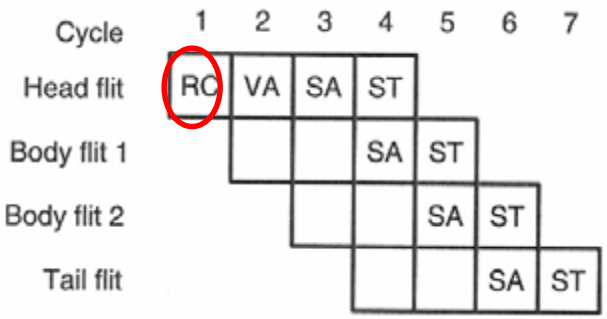
The Router Pipeline



no pipeline stalls

- Cycle 0
  - Head flit arrives and the packet is directed to an virtual channel of the input port (G = I)

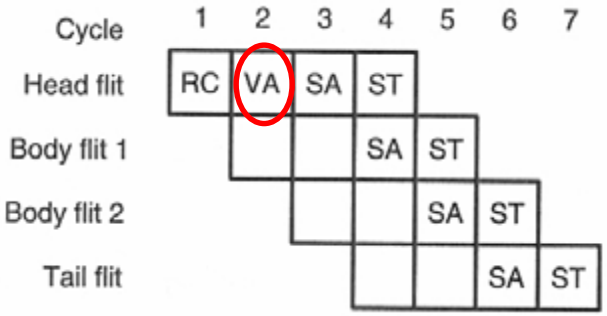
The Router Pipeline



no pipeline stalls

- Cycle 1
  - Routing computation
  - Virtual channel state changes to routing (G = R)
  - Head flit enters RC-stage
  - First body flit arrives at router

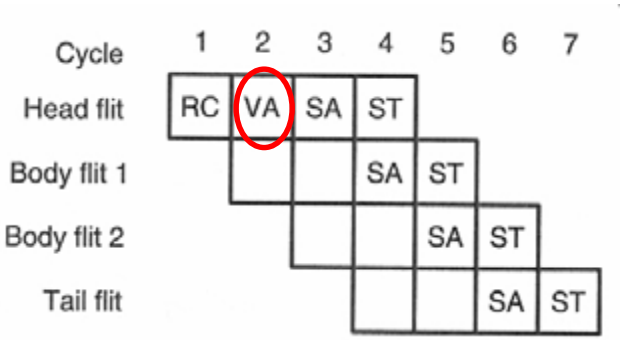
The Router Pipeline



no pipeline stalls

- Cycle 2: Virtual Channel Allocation
  - Route field (R) of virtual channel is updated
  - Virtual channel state is set to "waiting for output virtual channel" (G = V)
  - Head flit enters VA state
  - First body flit enters RC stage
  - Second body flit arrives at router

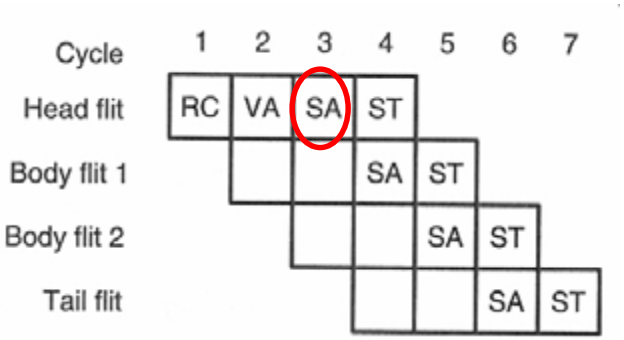
The Router Pipeline



no pipeline stalls

- Cycle 2: Virtual Channel Allocation
  - The result of the routing computation is input to the virtual channel allocator
  - If successful, the allocator assigns a single output virtual channel
  - The state of the virtual channel is set to active (G = A

The Router Pipeline

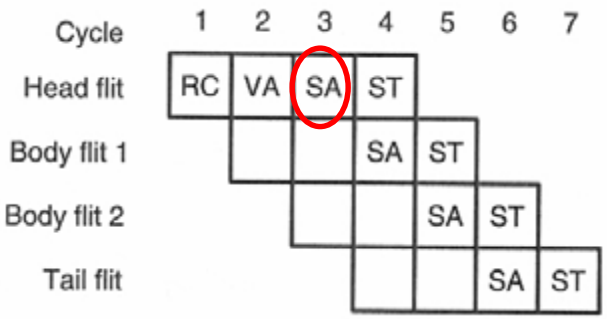


no pipeline stalls

- Cycle 3: Switch Allocation
  - All further processing is done on a flit base
  - Head flit enters SA stage
  - Any active VA (G = A) that contains buffered flits (indicated by P) and has downstream buffers available (C > 0) bids for a single-flit time slot through the switch from its input VC to the output VC

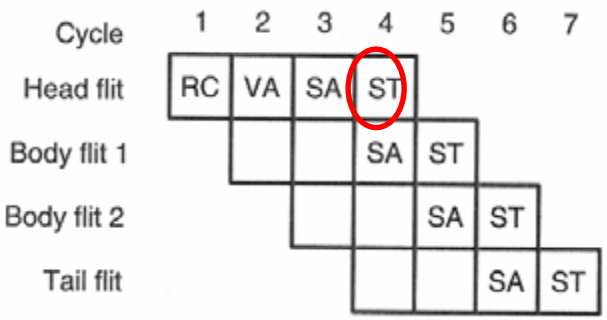
The Router Pipeline

- Cycle 3: Switch Allocation
  - If successful, pointer field is updated
  - Credit field is decremented



no pipeline stalls

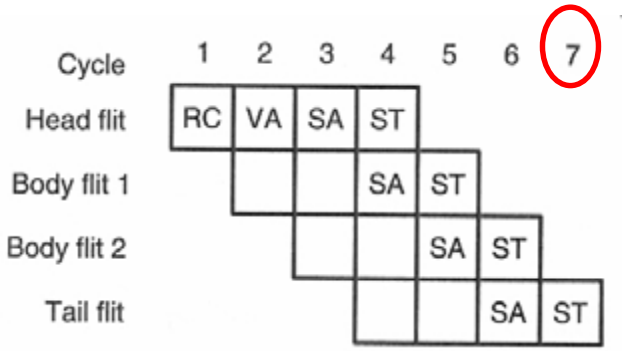
The Router Pipeline



no pipeline stalls

- Cycle 4: Switch Traversal
  - Head flit traverses the switch
- Cycle 5:
  - Head flit starts traversing the channel to the next router

The Router Pipeline

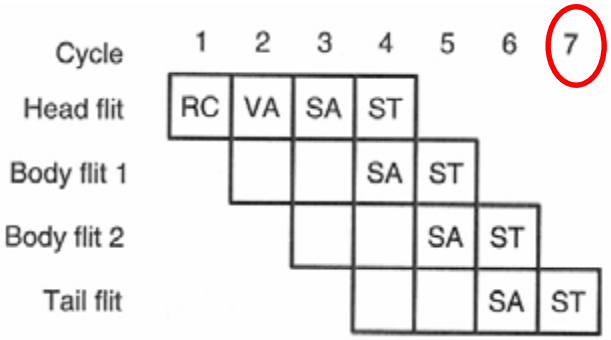


no pipeline stalls

- Cycle 7:
  - Tail traverses the switch
  - Output VC set to idle
  - Input VC set to idle (G = I), if buffer is empty
  - Input VC set to routing (G = R), if another head flit is in the buffer

The Router Pipeline

- Only the head flits enter the RC and VC stages



no pipeline stalls

- The body and tail flits are stored in the flit buffers until they can enter the SA stage

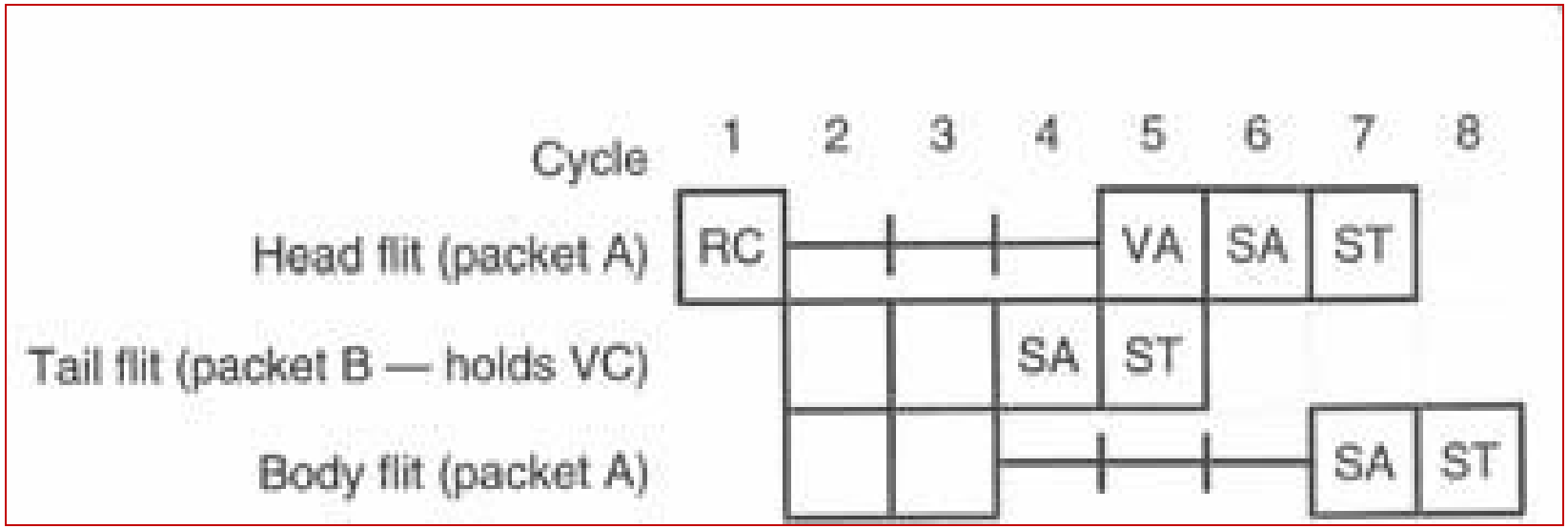
Pipeline Stalls

Pipeline stalls can be divided into:

- **Packet stalls**
  - can occur if the virtual channel cannot advance to its R, V, or A state
- **Flit stalls**
  - If a virtual channel is in active state and the flit cannot successfully complete switch allocation due to
    - Lack of flit, Lack of credit, Losing arbitration for the switch time slot

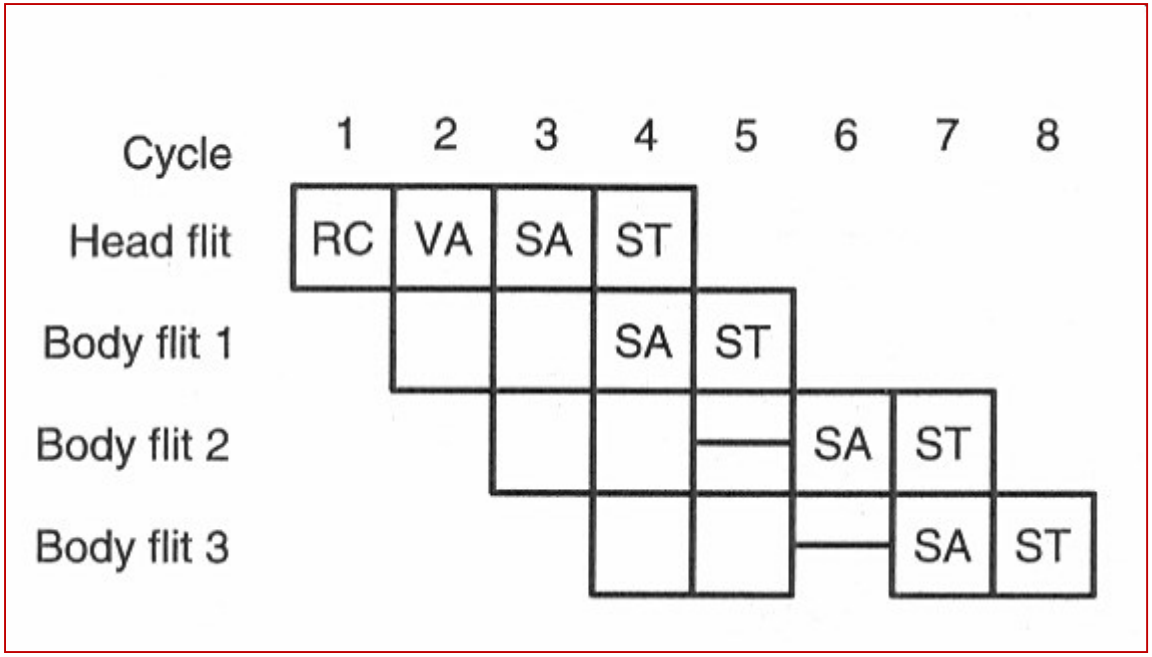
Example for Packet Stall

1. Virtual-channel allocation stall
   - Head flit of A can first enter the VA stage when the tail flit of packet B completes switch allocation and releases the virtual channel



Example for Flit Stalls

2. Switch allocation stall



Second body flit fails to allocate the requested connection in cycle 5

Example for Flit Stalls

3. Buffer empty stall

Cycle	1	2	3	4	5	6	7	8
Head flit	RC	VA	SA	ST				
Body flit 1				SA	ST			
Body flit 2						SA	ST	
Body flit 3							SA	ST

Body flit 2 is delayed three cycles. However, since it does not have to enter the RC and VA stage the output is only delayed one cycle!

Part II: NoC Building Blocks

Topology

Routing

Switching

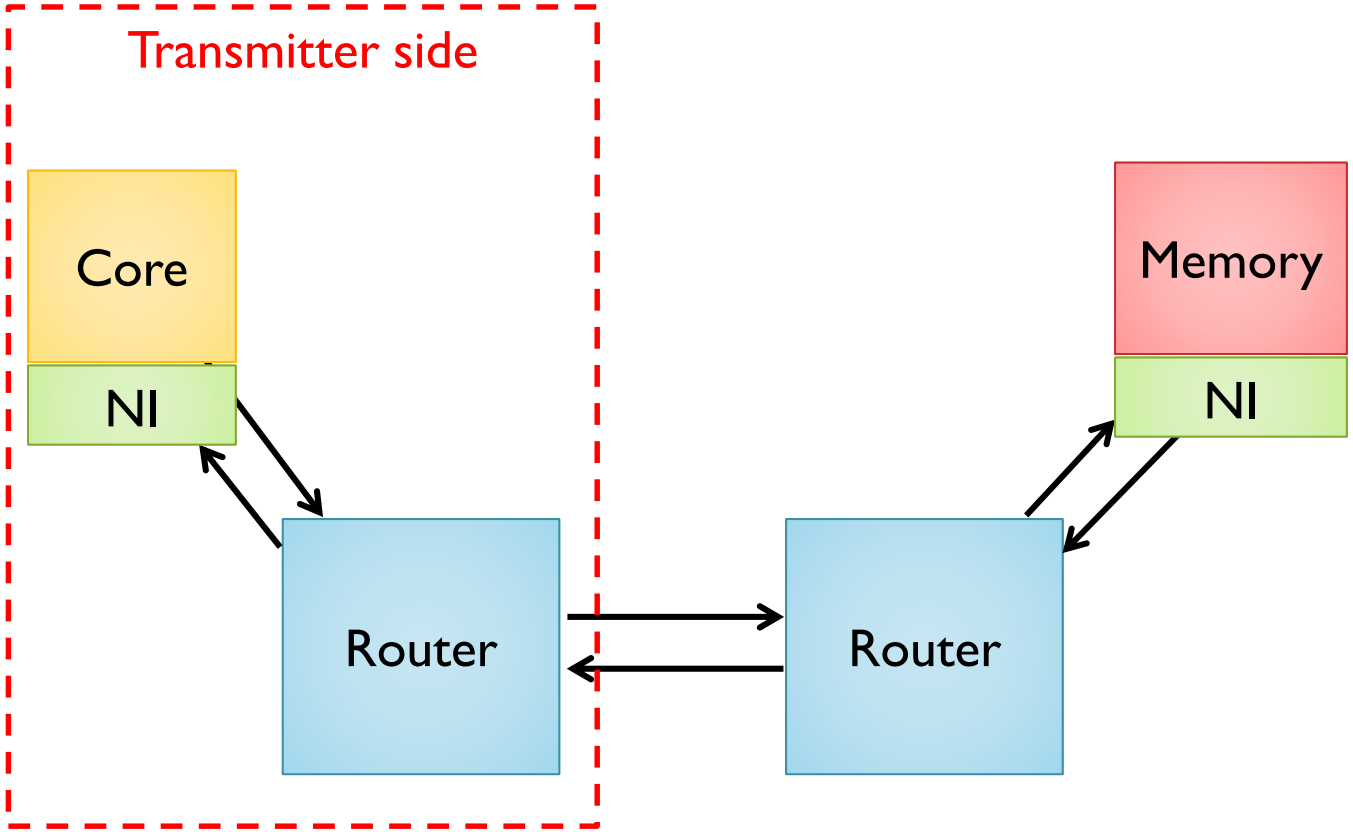
Virtual Channels

Flow Control

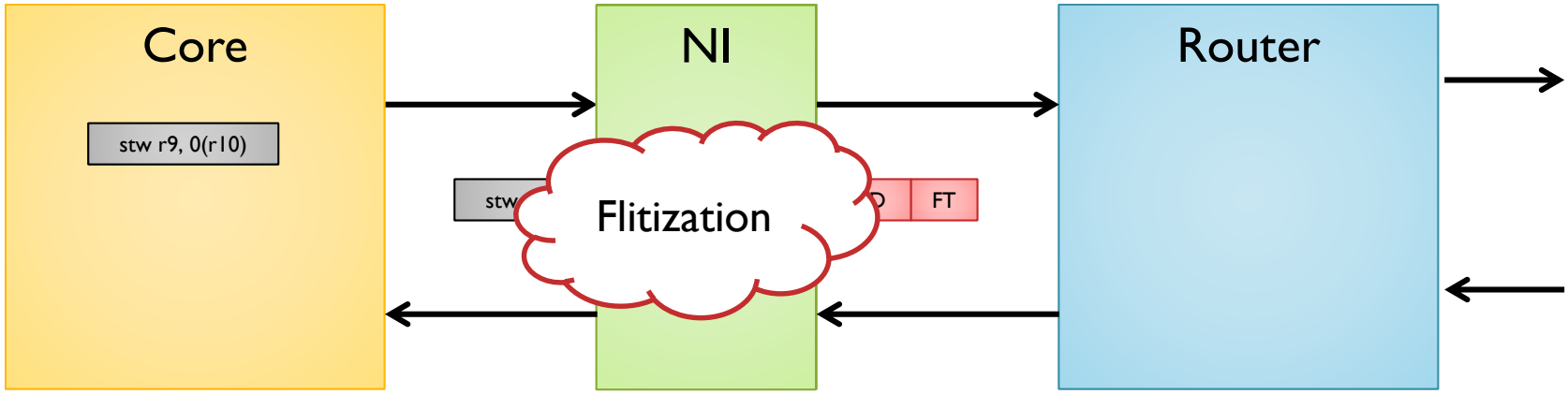
Router Architecture

Network Interface

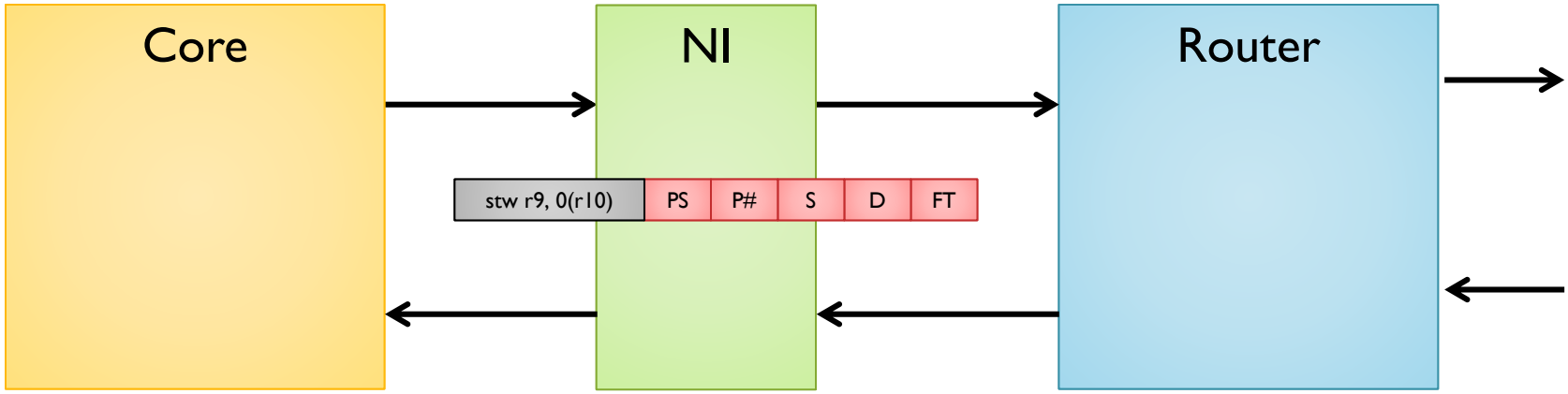
Network Interface



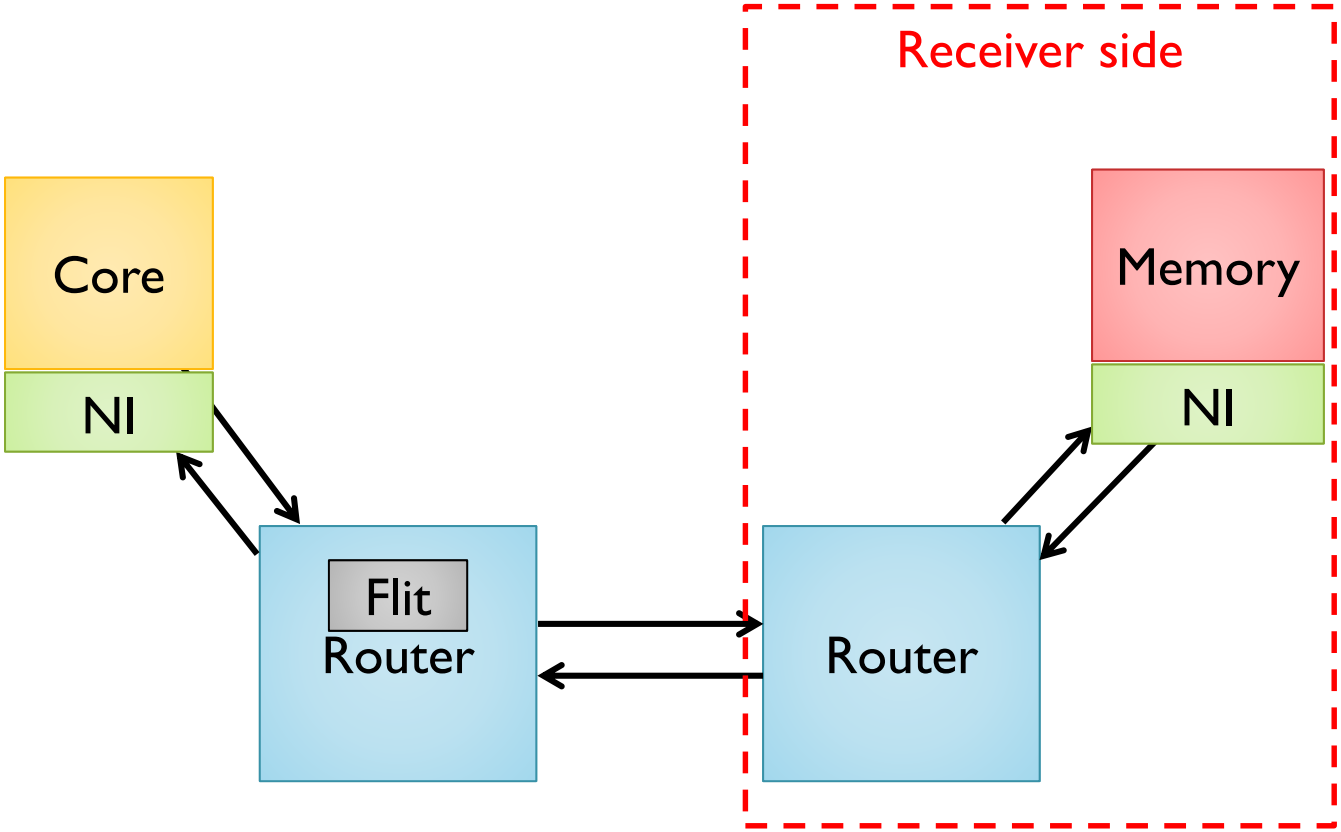
Network Interface



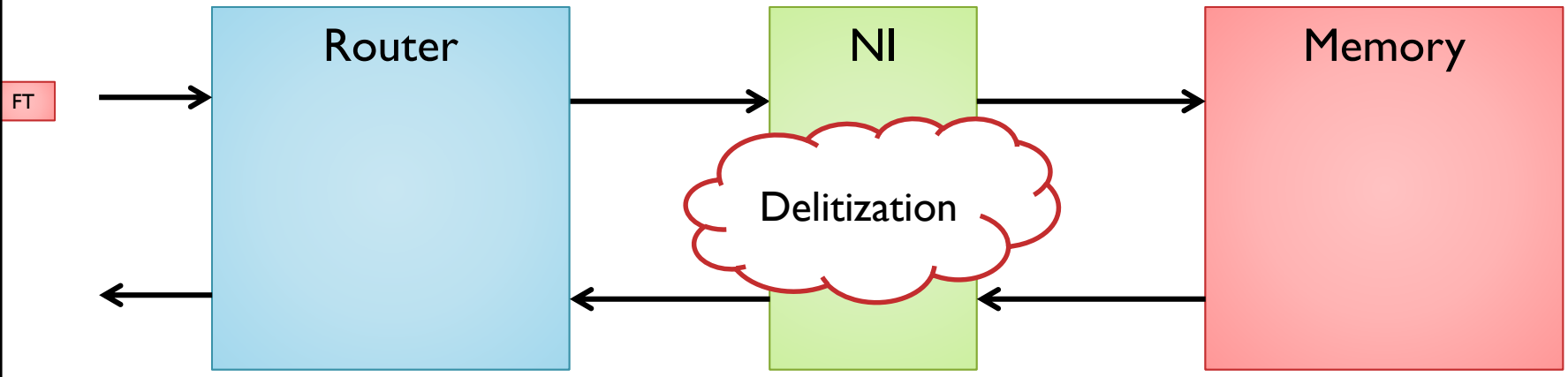
Network Interface



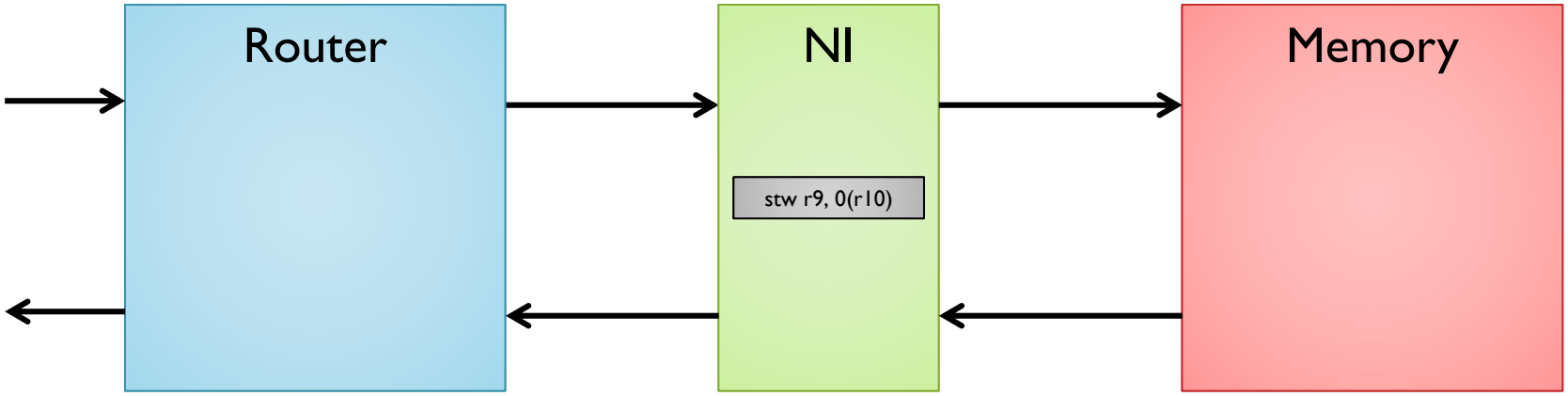
Network Interface



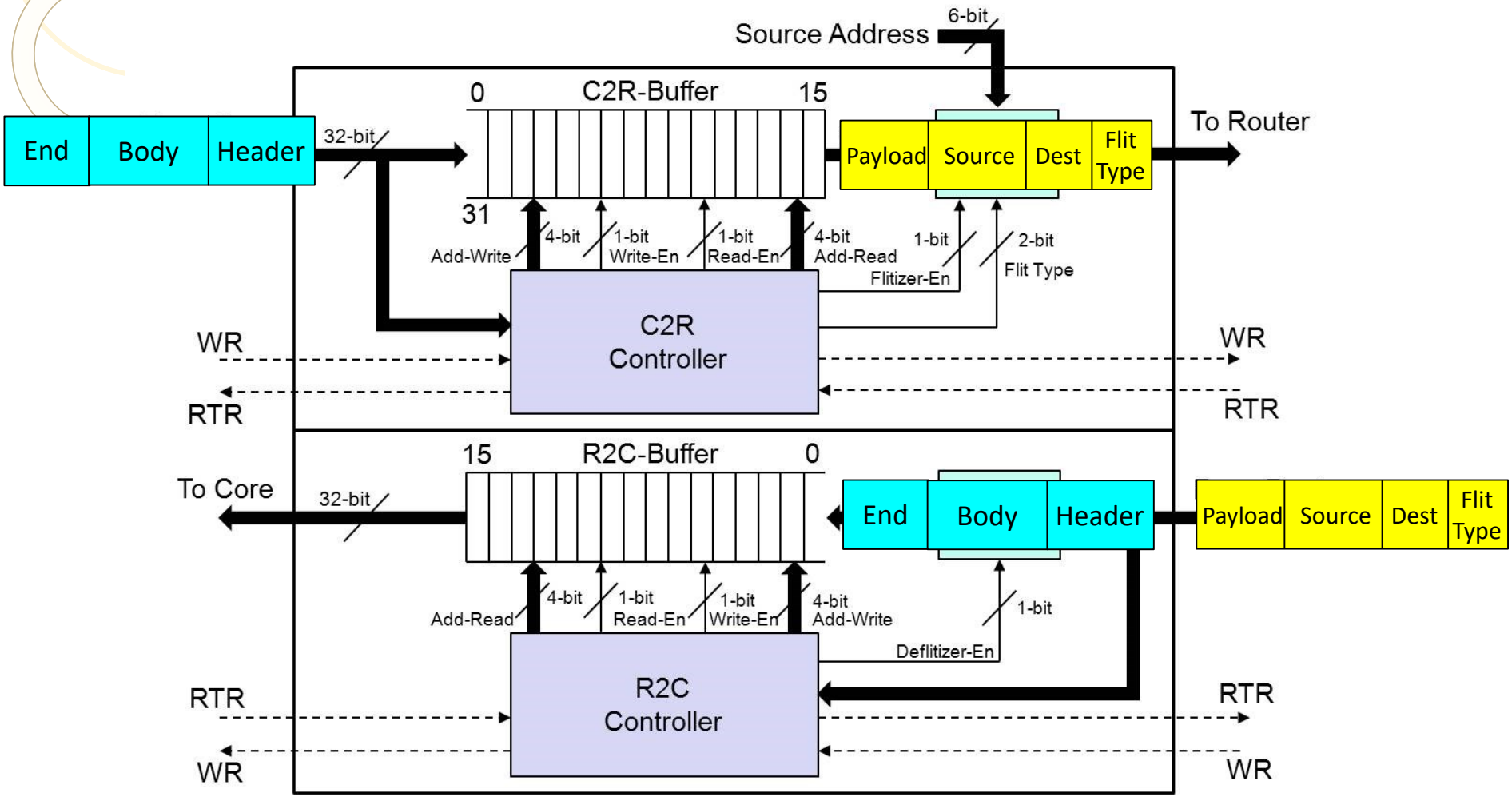
Network Interface



Network Interface



Network Interface



OASIS NoC NI

Summary

- NoC is a scalable platform for billion-transistor chips.
- Several driving forces behind it.
- Telecommunication devices, embedded and GP domains are attractive applications for NoC.
- Expected to change the way we structure and model VLSI systems.
- Many open research questions.

References

- William James Dally (Author), Brian Patrick Towles (Author), Principles and Practices of Interconnection Networks (The Morgan Kaufmann Series in Computer Architecture and Design) 1st Edition, ISBN-10, Morgan Kaufmann, January 1, 2004
- Akram Ben Ahmed, Shohei Miura, A. Ben Abdallah, Run-Time Monitoring Mechanism for Efficient Design of Network-on-Chip Architectures, *to appear in the 6th International Workshop on Engineering Parallel and Multicore Systems (ePaMuS2013'), July 2013.*
- Akram Ben Ahmed, A. Ben Abdallah, **Low-overhead Routing Algorithm for 3D Network-on-Chip**, *IEEE Proc. of the The Third International Conference on Networking and Computing (ICNC'12), pp. 23-32, 2012.*
- Akram Ben Ahmed, A. Ben Abdallah, **LA-XYZ: Low Latency, High Throughput Look-Ahead Routing Algorithm for 3D Network-on-Chip (3D-NoC) Architecture**, *IEEE Proceedings of the 6th International Symposium on Embedded Multicore SoCs (MCSoC-12), pp. 167-174, 2012.*
- Akram Ben Ahmed, A. Ben Abdallah, **ONoC-SPL Customized Network-on-Chip (NoC) Architecture and Prototyping for Data-intensive Computation Applications**, *IEEE Proceedings of The 4th International Conference on Awareness Science and Technology, pp. 257-262, 2012.*
- A. Ben Ahmed, A. Ben Abdallah, **Efficient Look-Ahead Routing Algorithm for 3D Network-on-Chip (3D-NoC)**,IEEE Proceedings of the 6th International Symposium on Embedded Multicore SoCs (MCSoC-12,) pp. 167-174,2012.
- R. Okada, **Architecture and Design of Core Network Interface for Distributed Routing in OASIS NoC**, Technical Report, ASL- Parallel Architecture Group, School of Computer Science and Engineering, The University of Aizu, March 2012.
- A. Ben Ahmed, A. Ben Abdallah, K. Kuroda,**Architecture and Design of Efficient 3D Network-on-Chip (3D NoC) for Custom Multicore SoC**, IEEE Proc. of the 5th International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA-2010), pp.67-73, Nov. 2010. (**best paper award**) (**slides**)
- K. Mori, A. Esch, A. Ben Abdallah, K. Kuroda, **Advanced Design Issues for OASIS Network-on-Chip Architecture**, IEEE Proc. of the 5th International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA-2010),pp.74-79, Nov. 2010. **slides**
- T. Uesaka, **OASIS NoC Topology Optimization with Short-Path Link**, Technical Report, Systems Architecture Group,March 2011
- K. Mori, A. Ben Abdallah, OASIS NoC Architecture Design in Verilog HDL, Technical Report,TR-062010-OASIS, Adaptive Systems Laboratory, the University of Aizu, June 2010. **slides**
- Shohei Miura, Abderazek Ben Abdallah, Kenichi Kuroda, PNoC: Design and Preliminary Evaluation of a Parameterizable NoC for MCSoC Generation and Design Space Exploration, The 19th Intelligent System Symposium (FAN 2009), pp.314-317, Sep.2009.
- Kenichi Mori, Abderazek Ben Abdallah, Kenichi Kuroda, **Design and Evaluation of a Complexity Effective Network-on-Chip Architecture on FPGA**, The 19th Intelligent System Symposium (FAN 2009), pp.318-321, Sep. 2009.
- A. Ben Abdallah, T. Yoshinaga and M. Sowa, "**Mathematical Model for Multiobjective Synthesis of NoC Architectures**", IEEE Proc. of the 36th International Conference on Parallel Processing, Sept. 4-8, 2007.
- A. Ben Abdallah, Masahiro Sowa, "**Basic Network-on-Chip Interconnection for Future Gigascale MCSoCs Applications: Communication and Computation Orthogonalization**", JASSST2006, Dec. 4-9th, 2006.

1. Book: **Multicore Systems-on-Chip: Practical Hardware/Software Design, 2nd Edition**, Author: A. Ben Abdallah, Publisher: Springer, (2013) , ISBN-13: 978-9491216916. [Amazon]