Lecture 13: Interconnection Networks

- Topics: lots of background, recent innovations for power and performance

Interconnection Networks

- Recall: fully connected network, arrays/rings, meshes/tori, trees, butterflies, hypercubes

- Consider a k-ary d-cube: a d-dimension array with k elements in each dimension, there are links between elements that differ in one dimension by 1 (mod k)

- Number of nodes $N = k^d$

Number of switches	:	N
Switch degree	:	$2d + 1$
Number of links	:	Nd
Pins per node	:	$2wd$

(with no wraparound)

Avg. routing distance	:	$d(k-1)/2$
Diameter	:	$d(k-1)$
Bisection bandwidth	:	$2wk^{d-1}$
Switch complexity	:	$(2d + 1)^2$

Should we minimize or maximize dimension?

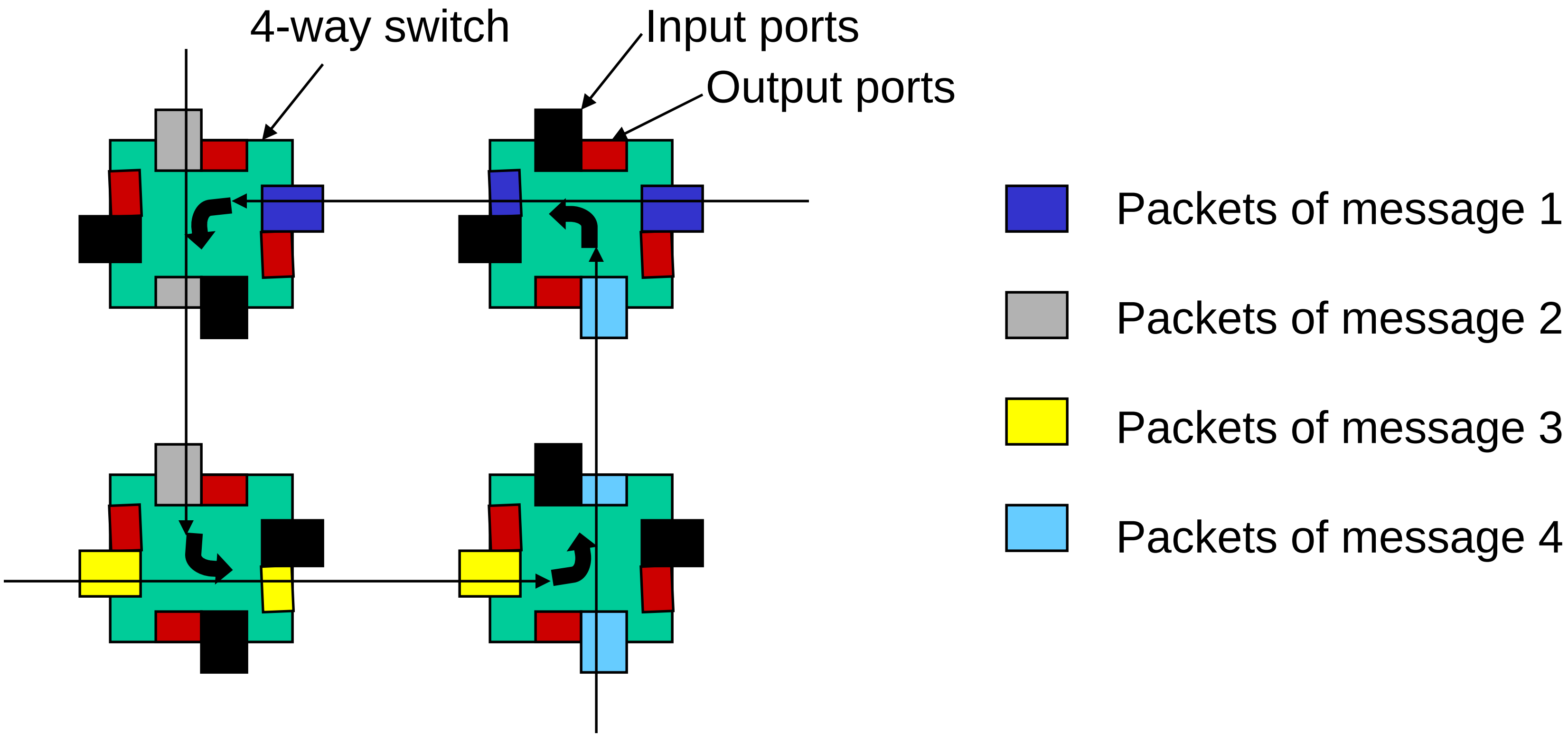
Routing

- Deterministic routing: given the source and destination, there exists a unique route

- Adaptive routing: a switch may alter the route in order to deal with unexpected events (faults, congestion) – more complexity in the router vs. potentially better performance

- Example of deterministic routing: dimension order routing: send packet along first dimension until destination co-ord (in that dimension) is reached, then next dimension, etc.

Deadlock Example

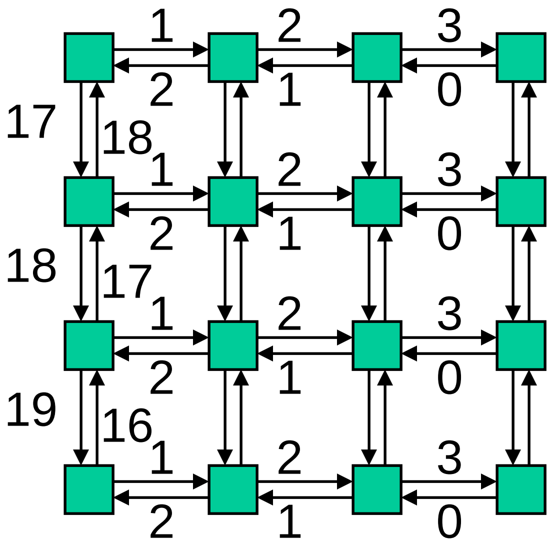


Each message is attempting to make a left turn – it must acquire an output port, while still holding on to a series of input and output ports

Deadlock-Free Proofs

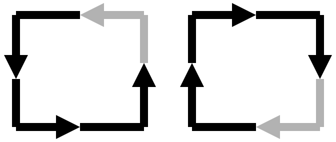
- Number edges and show that all routes will traverse edges in increasing (or decreasing) order – therefore, it will be impossible to have cyclic dependencies

- Example: k-ary 2-d array with dimension routing: first route along x-dimension, then along y

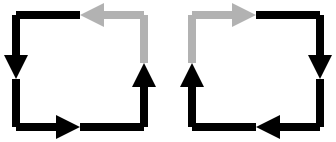


Breaking Deadlock II

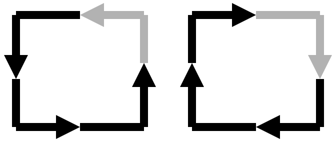
- Consider the eight possible turns in a 2-d array (note that turns lead to cycles)
- By preventing just two turns, cycles can be eliminated
- Dimension-order routing disallows four turns
- Helps avoid deadlock even in adaptive routing



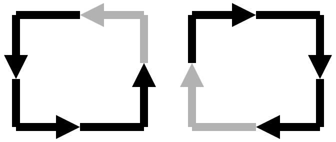
West-First



North-Last



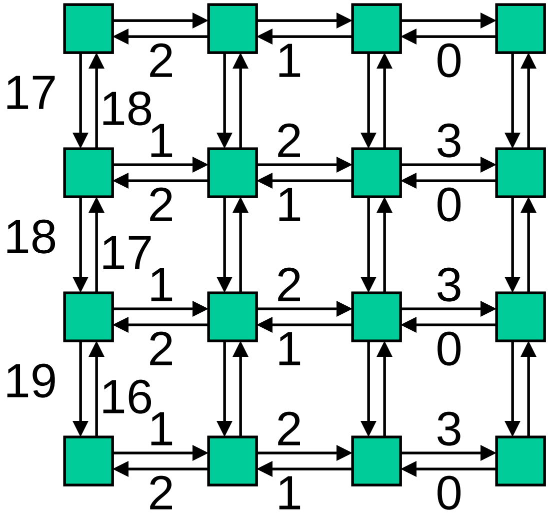
Negative-First

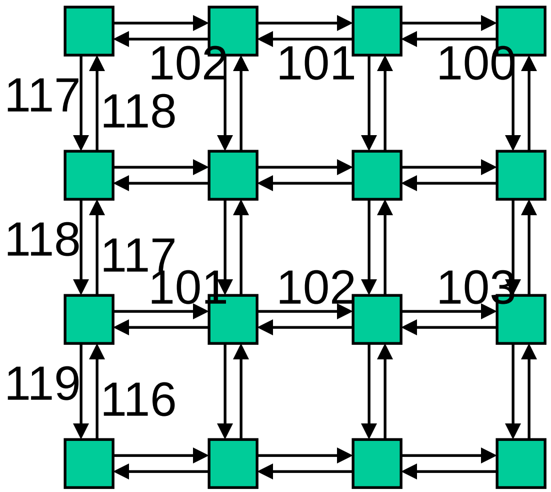


Can allow deadlocks

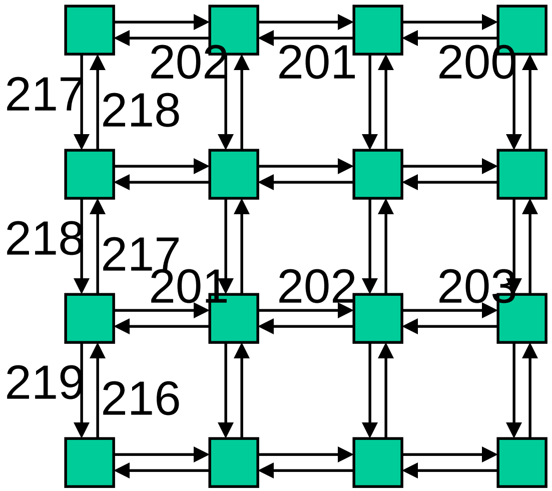
Deadlock Avoidance with VCs

- VCs provide another way to number the links such that a route always uses ascending link numbers





- Alternatively, use West-first routing on the 1st plane and cross over to the 2nd plane in case you need to go West again (the 2nd plane uses North-last, for example)



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

Flow Control

- The routing of a message requires allocation of various resources: the channel (or link), buffers, control state

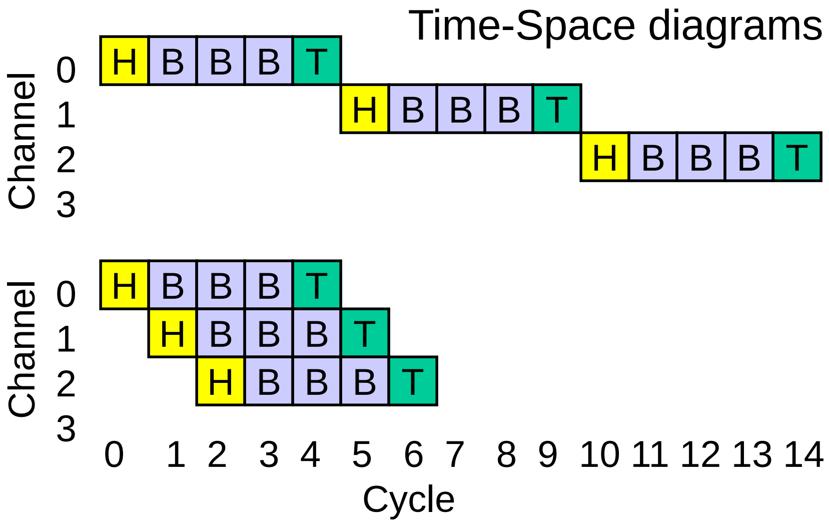
- Bufferless: flits are dropped if there is contention for a link, NACKs are sent back, and the original sender has to re-transmit the packet

- Circuit switching: a request is first sent to reserve the channels, the request may be held at an intermediate router until the channel is available (hence, not truly bufferless), ACKs are sent back, and subsequent packets/flits are routed with little effort (good for bulk transfers)

Buffered Flow Control

- A buffer between two channels decouples the resource allocation for each channel – buffer storage is not as precious a resource as the channel (perhaps, not so true for on-chip networks)

- Packet-buffer flow control: channels and buffers are allocated per packet
  - Store-and-forward
  - Cut-through



Time-Space diagrams

Flit-Buffer Flow Control (Wormhole)

- Wormhole Flow Control: just like cut-through, but with buffers allocated per flit (not channel)

- 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 and only compete for the buffer and physical channel

- Consumes much less buffer space than cut-through routing – does not improve channel utilization as another packet cannot cut in (only one VC per input port)

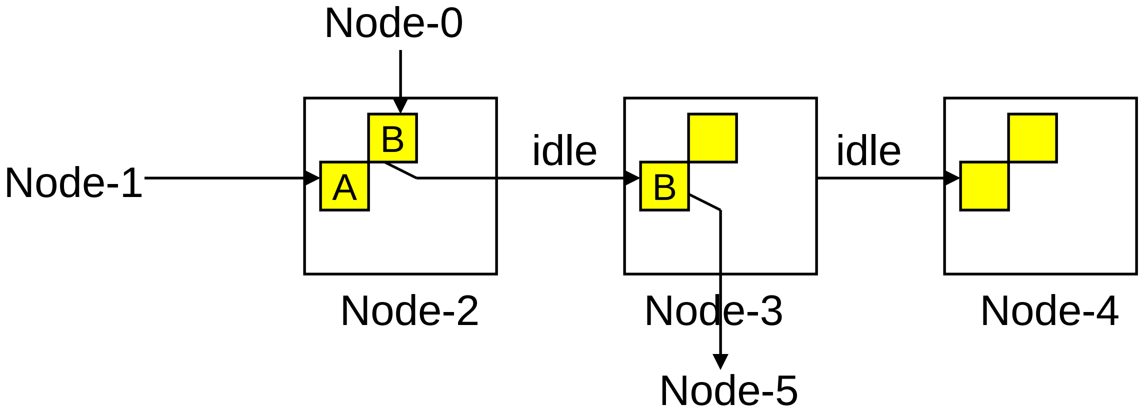
Virtual Channel Flow Control

- 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:

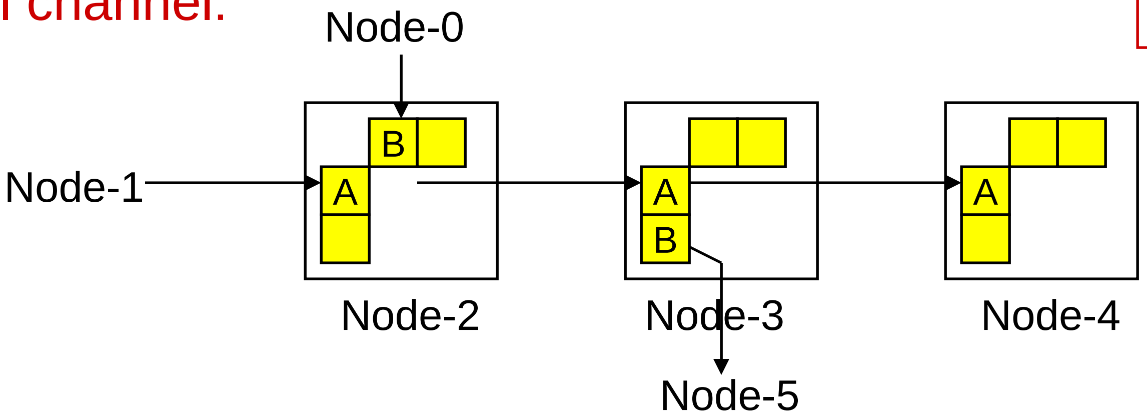
A is going from Node-1 to Node-4; B is going from Node-0 to Node-5



(blocked, no free VCs/buffers)

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:



(blocked, no free VCs/buffers)

Buffer Management

- Credit-based: keep track of the number of free buffers in the downstream node; the downstream node sends back signals to increment the count when a buffer is freed; need enough buffers to hide the round-trip latency

- On/Off: the upstream node sends back a signal when its buffers are close to being full – reduces upstream signaling and counters, but can waste buffer space

Router Pipeline

- Four typical stages:
  - RC routing computation: the head flit indicates the VC that it belongs to, the VC state is updated, the headers are examined and the next output channel is computed (note: this is done for all the head flits arriving on various input channels)
  - VA virtual-channel allocation: the head flits compete for the available virtual channels on their computed output channels
  - SA switch allocation: a flit competes for access to its output physical channel
  - ST switch traversal: the flit is transmitted on the output channel

A head flit goes through all four stages, the other flits do nothing in the first two stages (this is an in-order pipeline and flits can not jump ahead), a tail flit also de-allocates the VC

Speculative Pipelines

- Perform VA and SA in parallel
- Note that SA only requires knowledge of the output physical channel, not the VC
- If VA fails, the successfully allocated channel goes un-utilized

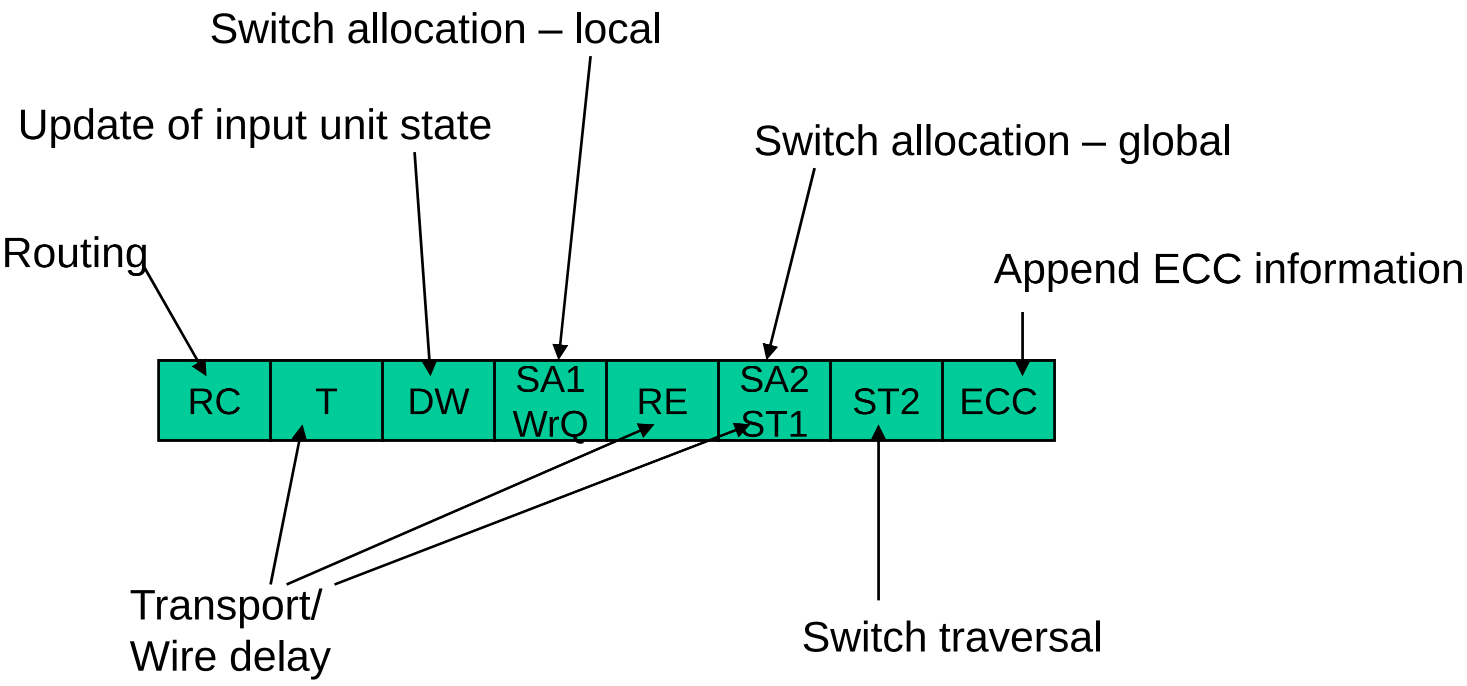
- Perform VA, SA, and ST in parallel (can cause collisions and re-tries)
- Typically, VA is the critical path – can possibly perform SA and ST sequentially

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	

RC	VA SA ST			
		SA ST		
			SA ST	
				SA ST

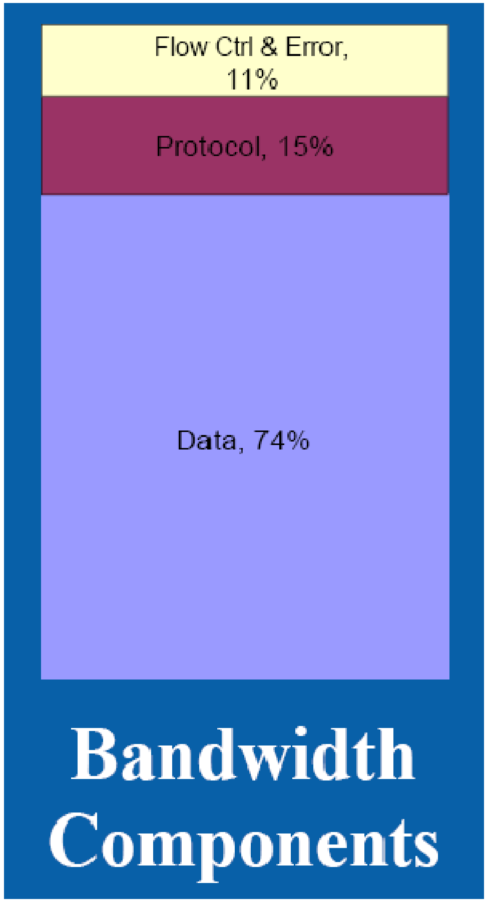
- Router pipeline latency is a greater bottleneck when there is little contention
- When there is little contention, speculation will likely work well!
- Single stage pipeline?

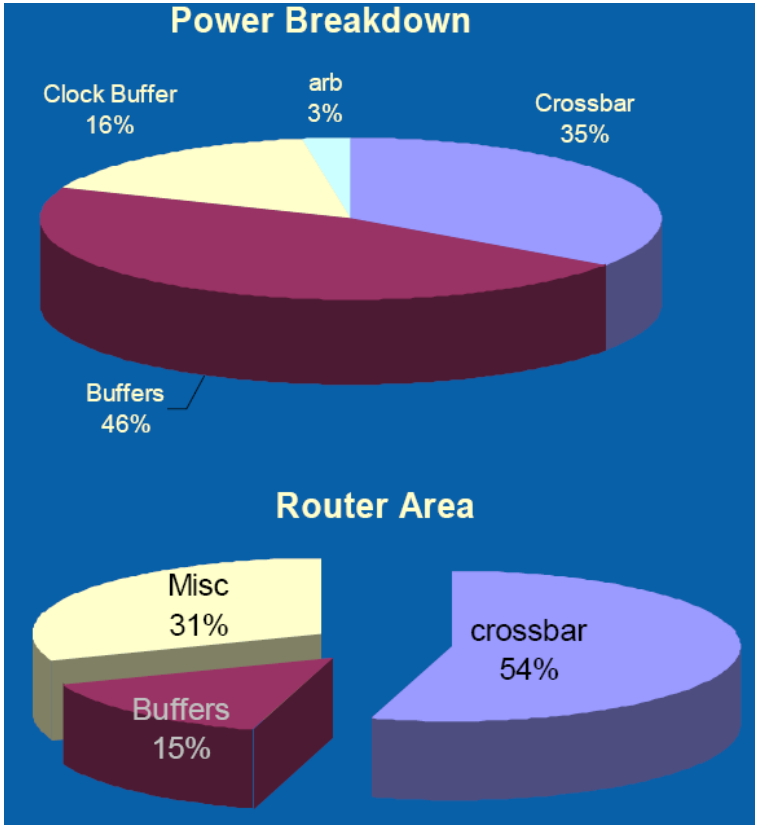
Alpha 21364 Pipeline



Write to input queues

Recent Intel Router

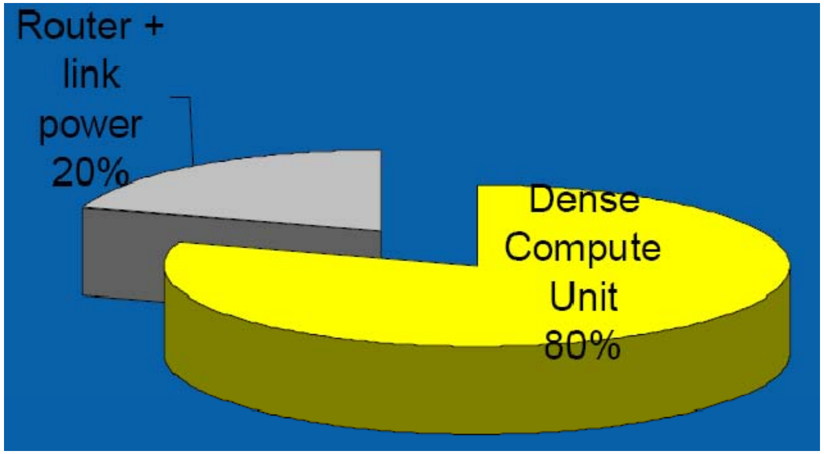


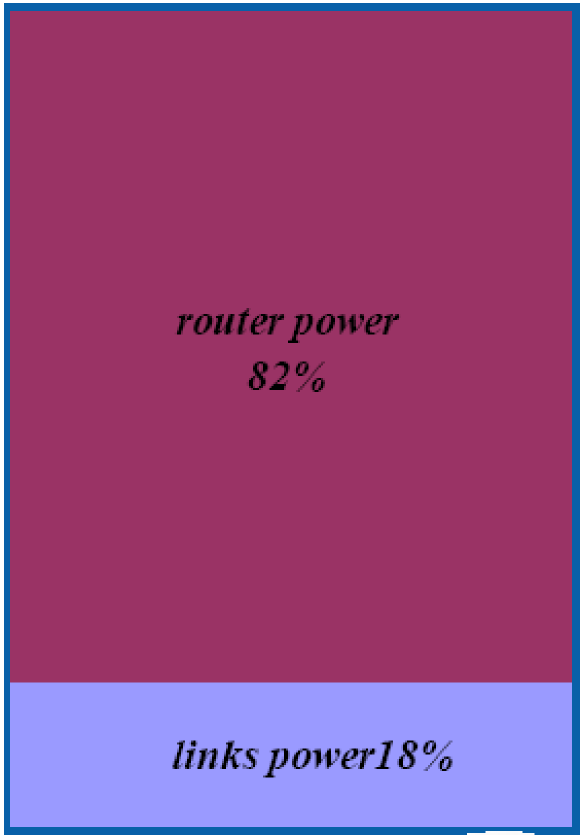


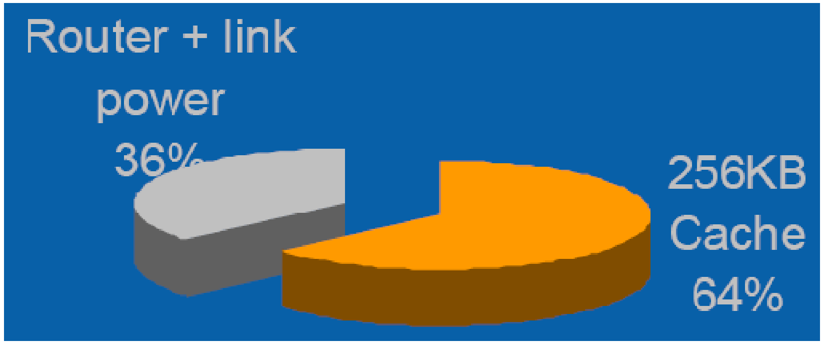
- Used for a 6x6 mesh
- 16 B, > 3 GHz
- Wormhole with VC flow control

Source: Partha Kundu, “On-Die Interconnects for Next-Generation CMPs”, talk at On-Chip Interconnection Networks Workshop, Dec 2006

Recent Intel Router

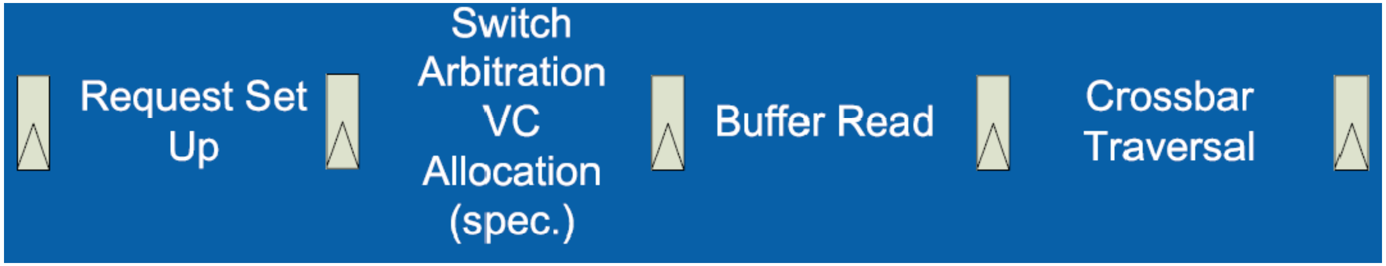






Source: Partha Kundu, “On-Die Interconnects for Next-Generation CMPs”, talk at On-Chip Interconnection Networks Workshop, Dec 2006

Recent Intel Router

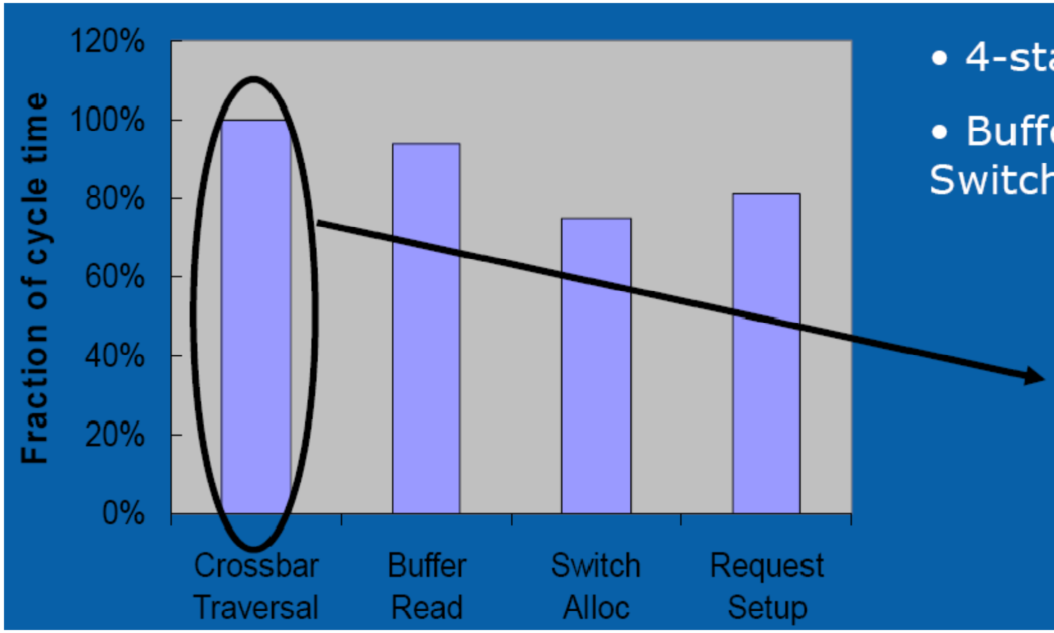


Request Set Up

Switch Arbitration VC Allocation (spec.)

Buffer Read

Crossbar Traversal



- 4-stage pipeline
- Buffer Read not in parallel with Switch Arbitration

Crossbar traversal sets the cycle time

Source: Partha Kundu, “On-Die Interconnects for Next-Generation CMPs”, talk at On-Chip Interconnection Networks Workshop, Dec 2006

Data Points

- On-chip network's power contribution
  in RAW (tiled) processor: 36%
  in network of compute-bound elements (Intel): 20%
  in network of storage elements (Intel): 36%
  bus-based coherence (Kumar et al. '05): ~12%

- Contributors:
  RAW: links 39%; buffers 31%; crossbar 30%
  TRIPS: links 31%; buffers 35%; crossbar 33%
  Intel: links 18%; buffers 38%; crossbar 29%; clock 13%

Network Power

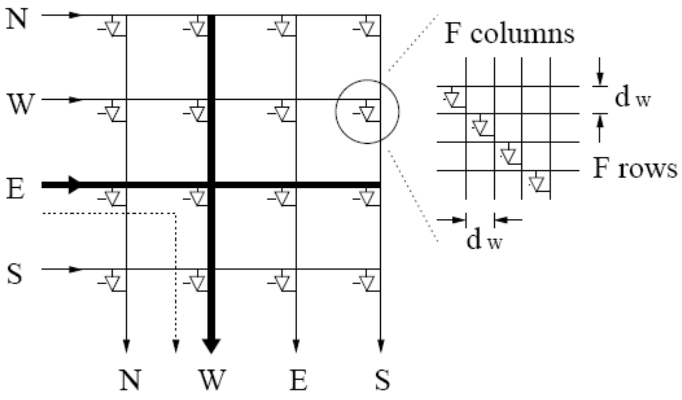
- Power-Driven Design of Router Microarchitectures in On-Chip Networks, MICRO’03, Princeton

- Energy for a flit = $E_R \,.\, H + E_{wire} \,.\, D$
$$= (E_{buf} + E_{xbar} + E_{arb}) \,.\, H + E_{wire} \,.\, D$$

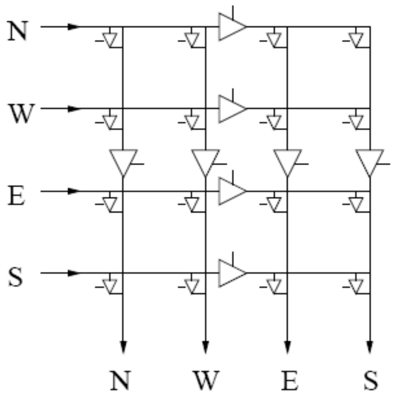
E_R = router energy	H = number of hops
E_{wire} = wire transmission energy	D = physical Manhattan distance
E_{buf} = router buffer energy	E_{xbar} = router crossbar energy
E_{arb} = router arbiter energy	

- This paper assumes that $E_{wire} \,.\, D$ is ideal network energy (assuming no change to the application and how it is mapped to physical nodes)

Segmented Crossbar



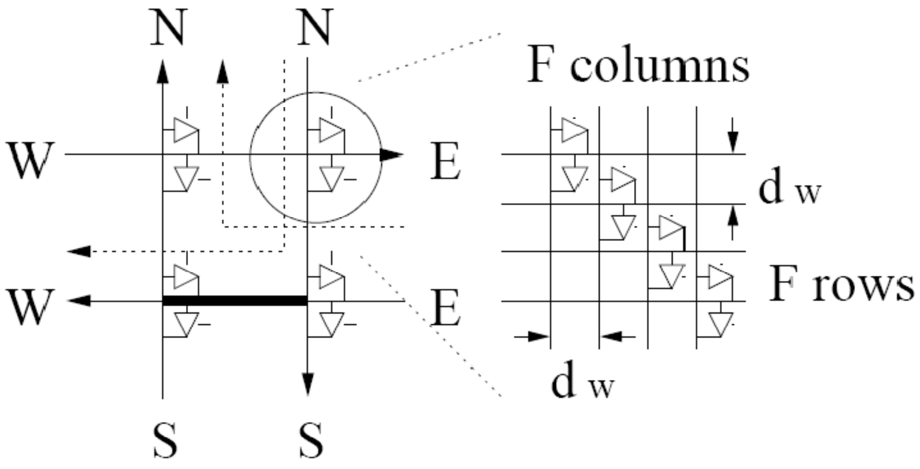
(a) A 4×4 matrix crossbar.



(b) A 4×4 segmented crossbar with 2 segments per line.

- By segmenting the row and column lines, parts of these lines need not switch → less switching capacitance (especially if your output and input ports are close to the bottom-left in the figure above)
- Need a few additional control signals to activate the tri-state buffers
- Overall crossbar power savings: ~15-30%

Cut-Through Crossbar

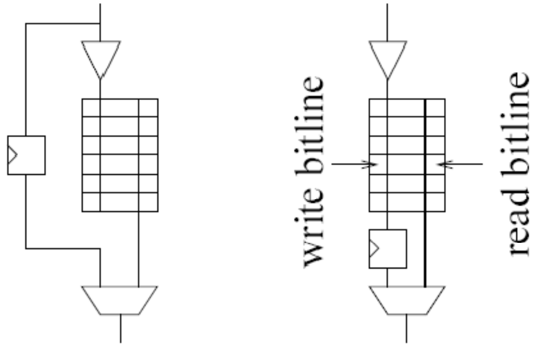


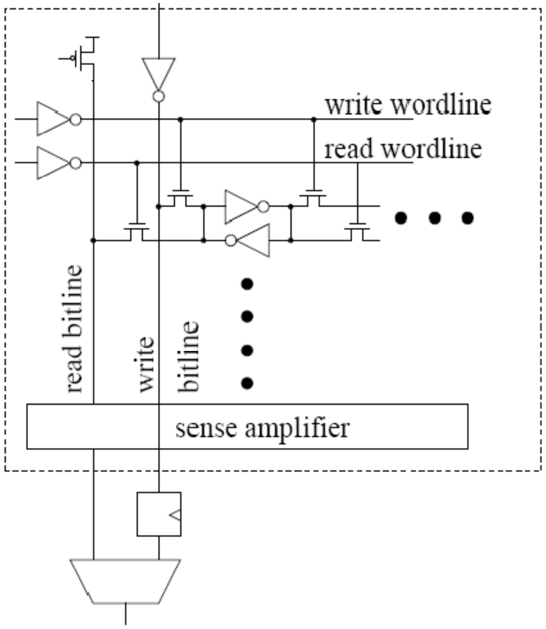
(a) A 4×4 cut-through crossbar.

- Attempts to optimize the common case: in dimension-order routing, flits make up to one turn and usually travel straight
- 2/3rd the number of tristate buffers and 1/2 the number of data wires
- “Straight” traffic does not go thru tristate buffers
- Some combinations of turns are not allowed: such as E → N and N → W (note that such a combination cannot happen with dimension-order routing)
- Crossbar energy savings of 39-52%

Write-Through Input Buffer

- Input flits must be buffered in case there is a conflict in a later pipeline stage
- If the queue is empty, the input flit can move straight to the next stage: helps avoid the buffer read
- To reduce the datapaths, the write bitlines can serve as the bypass path
- Power savings are a function of rd/wr energy ratios and probability of finding an empty queue



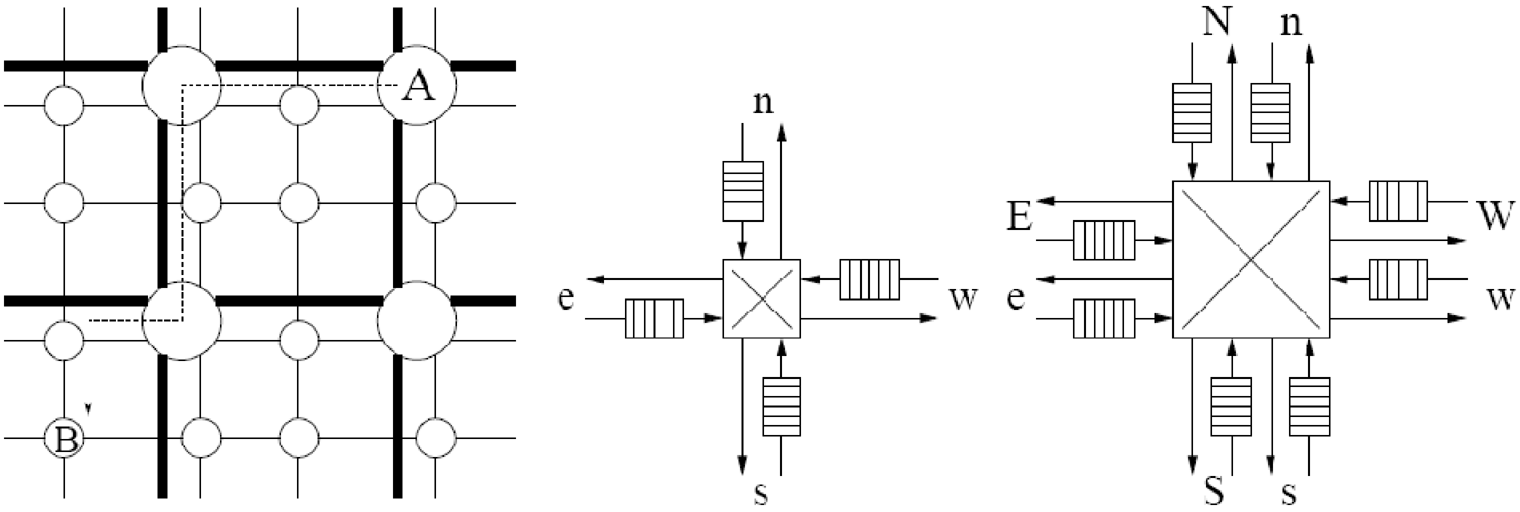


Express Channels

- Express channels connect non-adjacent nodes – flits traveling a long distance can use express channels for most of the way and navigate on local channels near the source/destination (like taking the freeway)

- Helps reduce the number of hops

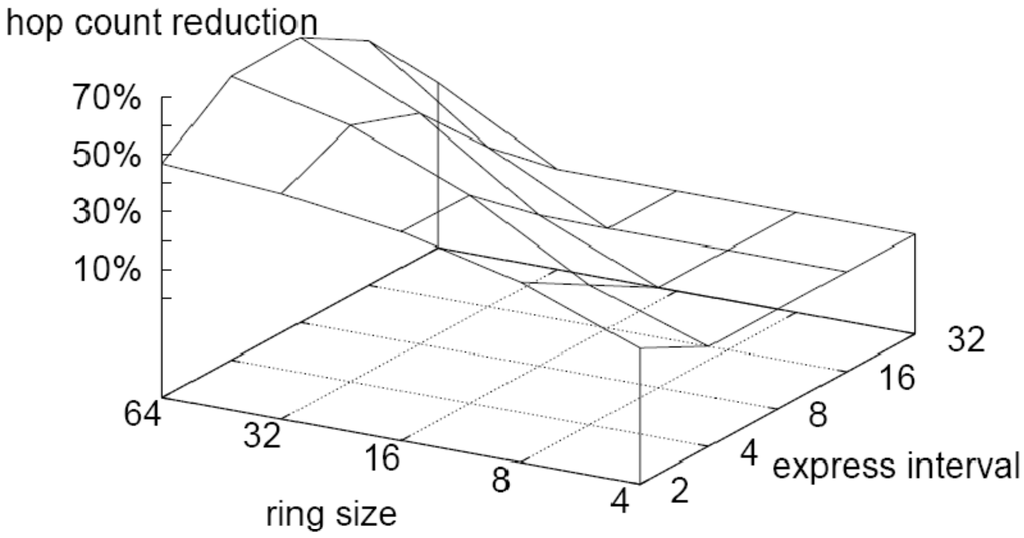
- The router in each express node is much bigger now



Express Channels

- Routing: in a ring, there are 5 possible routes and the best is chosen; in a torus, there are 17 possible routes

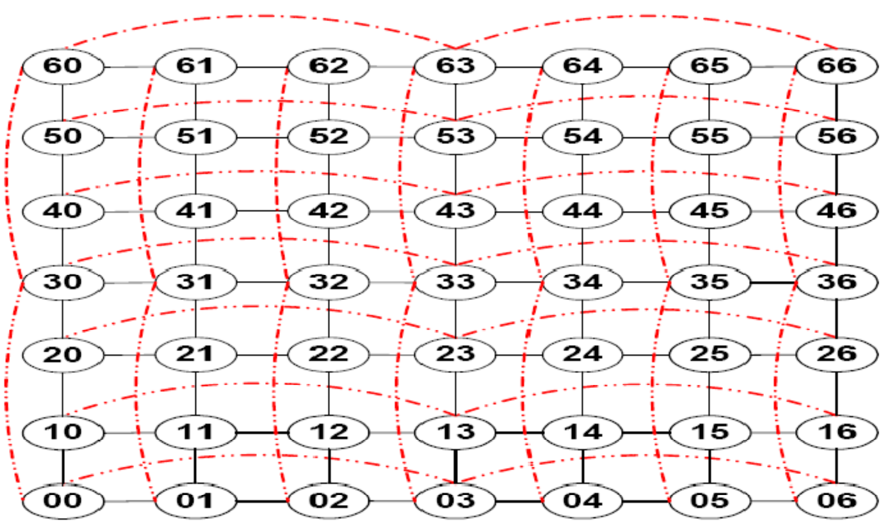
- A large express interval results in fewer savings because fewer messages exercise the express channels

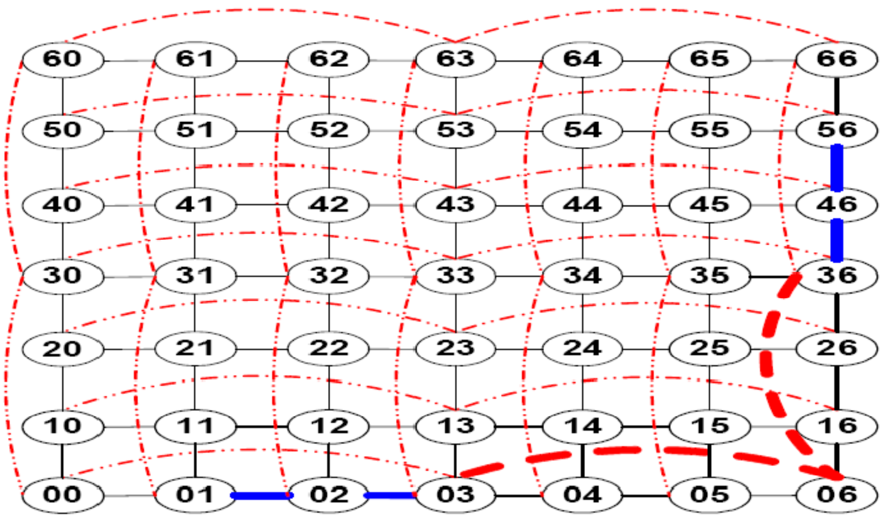


Express Virtual Channels

- To a large extent, maintain the same physical structure as a conventional network (changes to be explained shortly)

- Some virtual channels are treated differently: they go through a different router pipeline and can effectively avoid most router overheads





(b) VCs acquired from nodes 01 to 56

Router Pipelines

- If Normal VC (NVC):
  - at every router, must compete for the next VC and for the switch
  - will get buffered in case there is a conflict for VA/SA

- If EVC (at intermediate bypass router):
  - need not compete for VC (an EVC is a VC reserved across multiple routers)
  - similarly, the EVC is also guaranteed the switch (only 1 EVC can compete for an output physical channel)
  - since VA/SA are guaranteed to succeed, no need for buffering
  - simple router pipeline: incoming flit directly moves to ST stage

- If EVC (at EVC source/sink router):
  - must compete for VC/SA as in a conventional pipeline
  - before moving on, must confirm free buffer at next EVC router

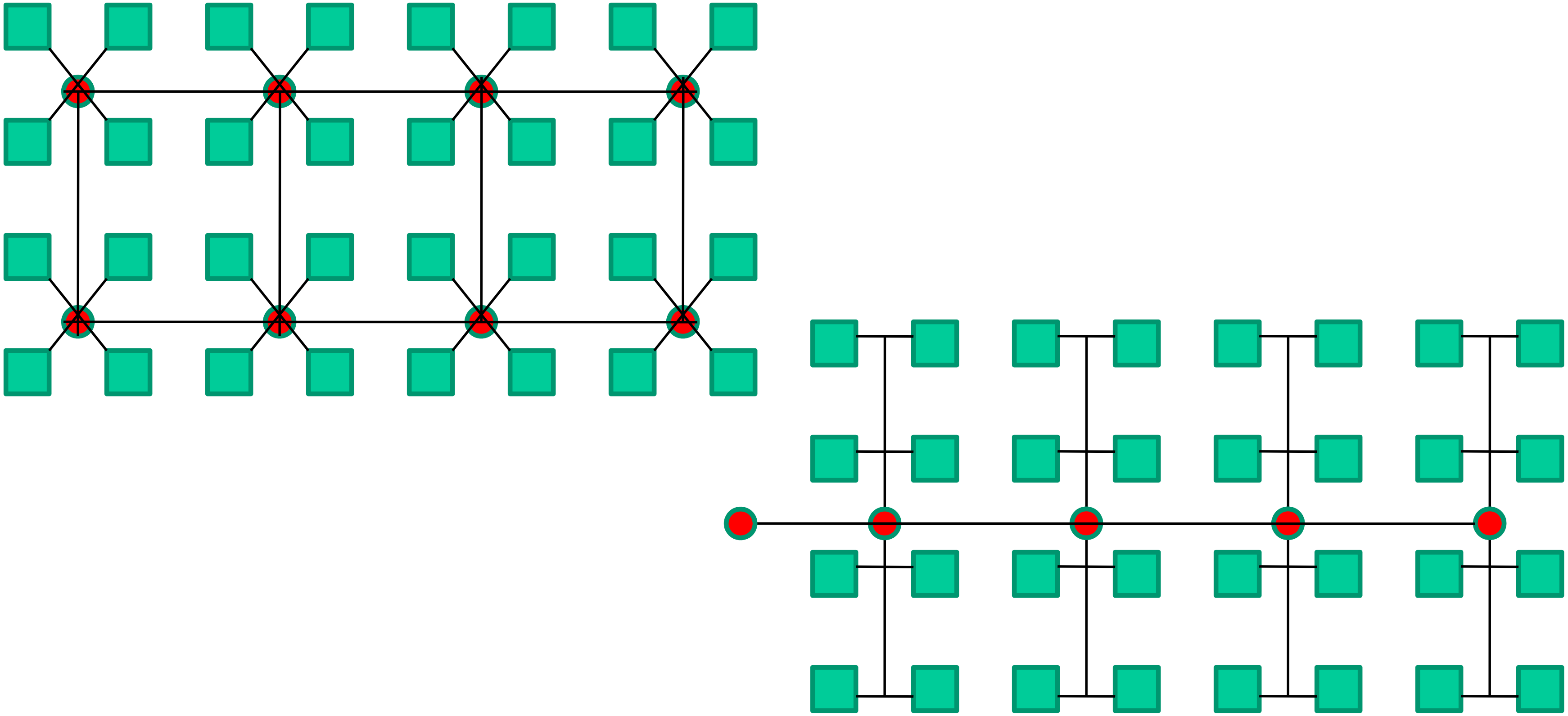
Bypass Router Pipelines

- Non aggressive pipeline in a bypass node: an express flit simply goes through the crossbar and then on the link; the prior SA stage must know that an express flit is arriving so that the switch control signals can be appropriately set up; this requires the flit to be preceded by a single-bit control signal (similar to cct-switching, but much cheaper)

- Aggressive pipeline: the express flit avoids the switch and heads straight to the output channel (dedicated hardware)… will still need a mechanism to control ST for other flits

Other Innovations

- Concentrated mesh, hierarchical/hybrid networks, rings



Title

- Bullet