Lecture 13: Interconnection Networks

- Topics: flow control, router pipelines, case studies

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

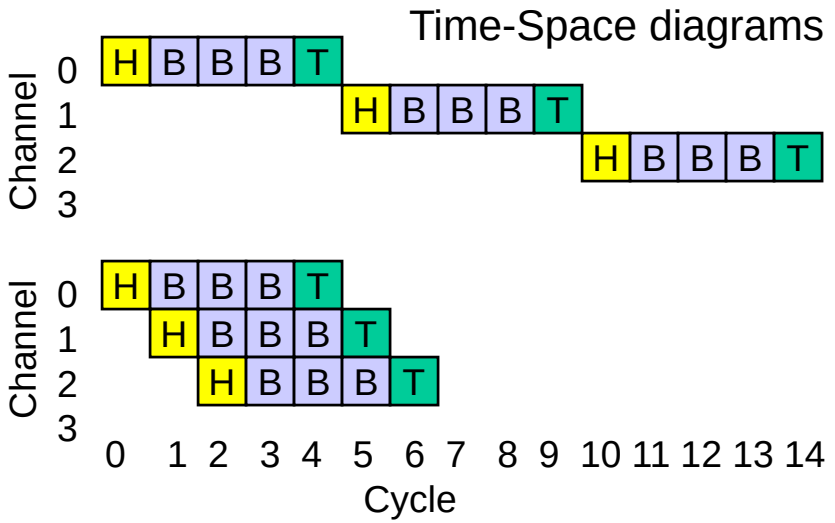
Flow Control

- 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



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)

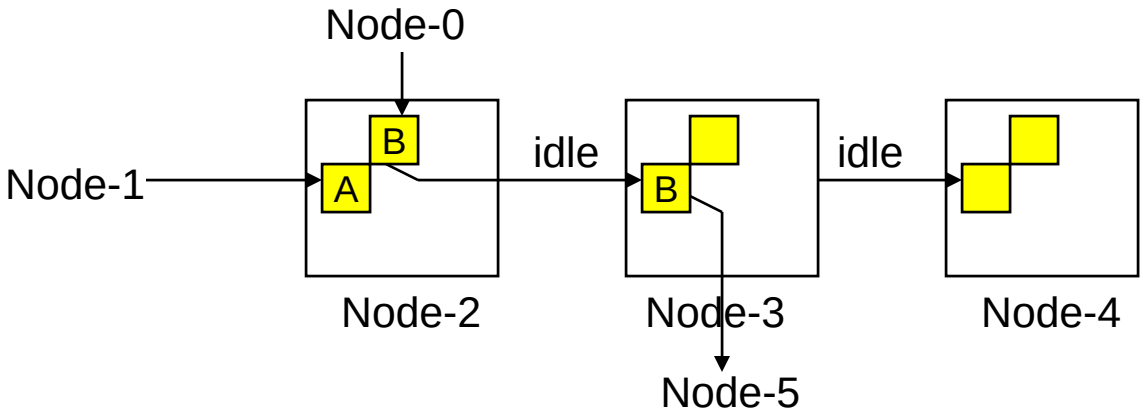
Virtual Channel Flow Control

- 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:

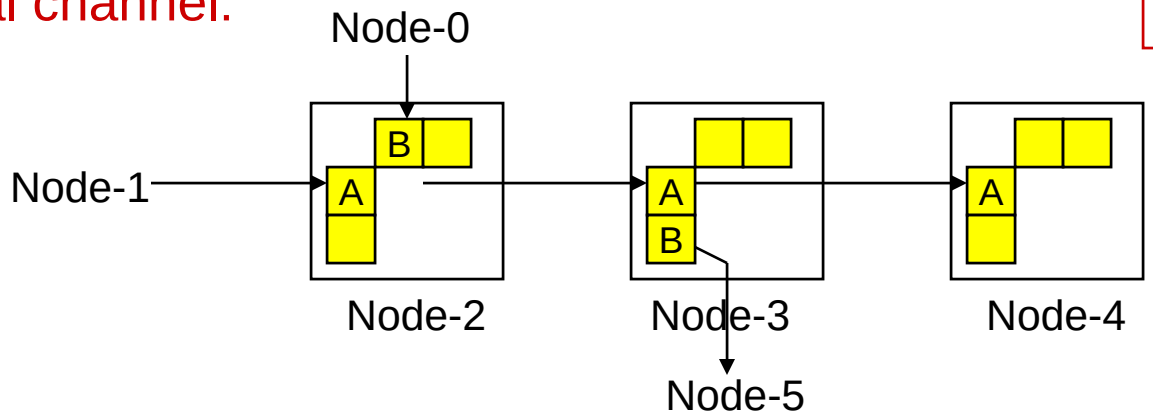
A is going from Node-1 to Node-4; B is going from Node-0 to Node-5



(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)

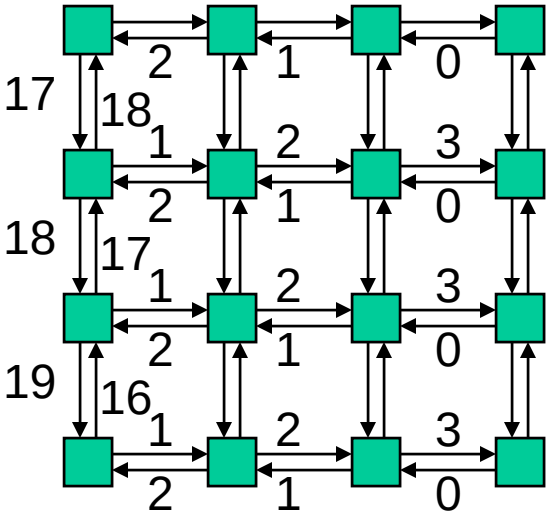
Buffer Management

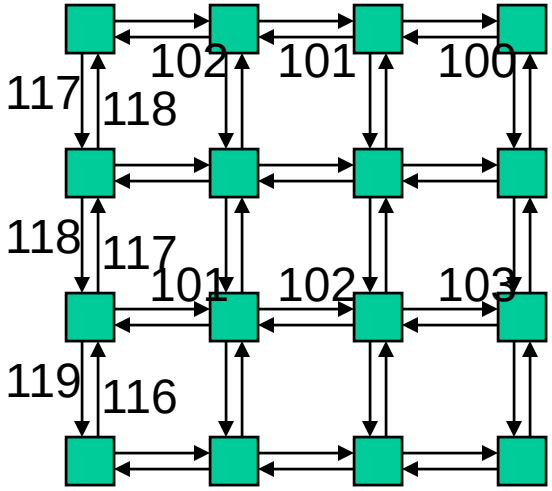
- Credit-based: keep track of the number of free buffers in the downstream node; the downstream node sends back signals to increment the count when a buffer is freed; need enough buffers to hide the round-trip latency

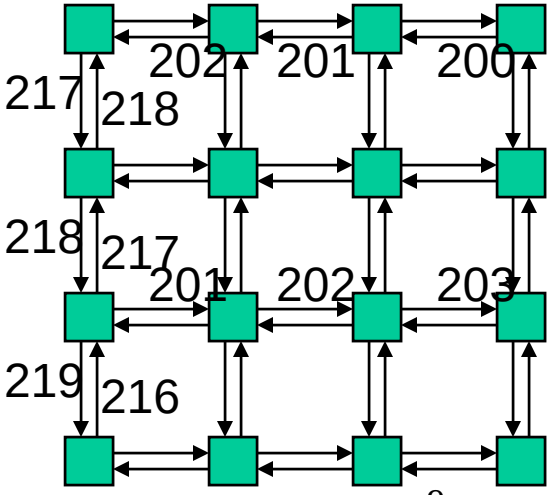
- On/Off: the upstream node sends back a signal when its buffers are close to being full – reduces upstream signaling and counters, but can waste buffer space

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)

Router Functions

- Crossbar, buffer, arbiter, VC state and allocation, buffer management, ALUs, control logic

- Typical on-chip network power breakdown:
  - 30% link
  - 30% buffers
  - 30% crossbar

Virtual Channel Router

- Buffers and channels are allocated per flit
- Each physical channel is associated with multiple virtual channels – the virtual channels are allocated per packet and the flits of various VCs can be interweaved on the physical channel
- For a head flit to proceed, the router has to first allocate a virtual channel on the *next* router
- For any flit to proceed (including the head), the router has to allocate the following resources: buffer space in the next router (credits indicate the available space), access to the physical channel

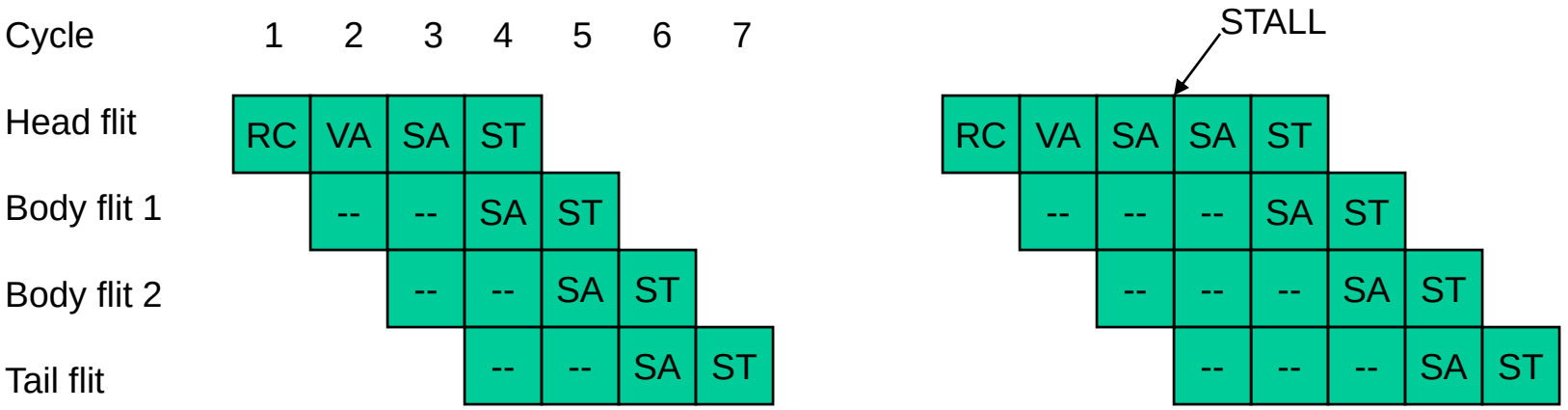
Router Pipeline

- 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

Router Pipeline

- Four typical stages:
  - RC routing computation: compute the output channel
  - VA virtual-channel allocation: allocate VC for the head flit
  - SA switch allocation: compete for output physical channel
  - ST switch traversal: transfer data on output physical channel

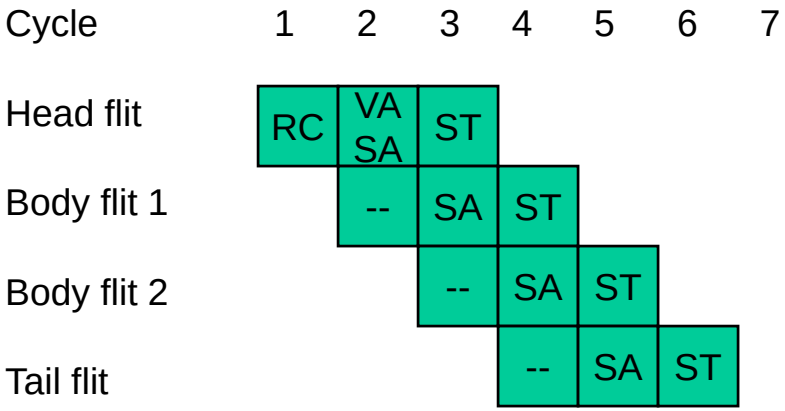


Stalls

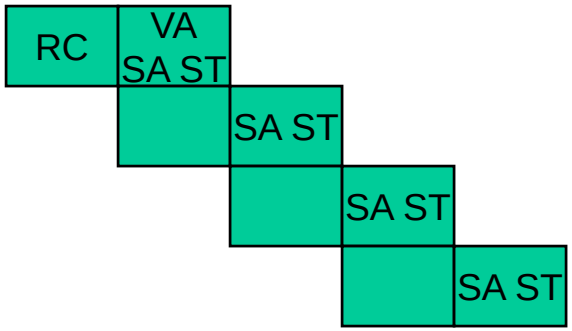
- Causes behind stalls:
  - RC fail: new head flit arrives, but the previous packet's tail flit is still competing for its output port
  - VA fail because no VCs available
  - SA fail because no credits (buffers) available
  - SA fail because no channel available

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

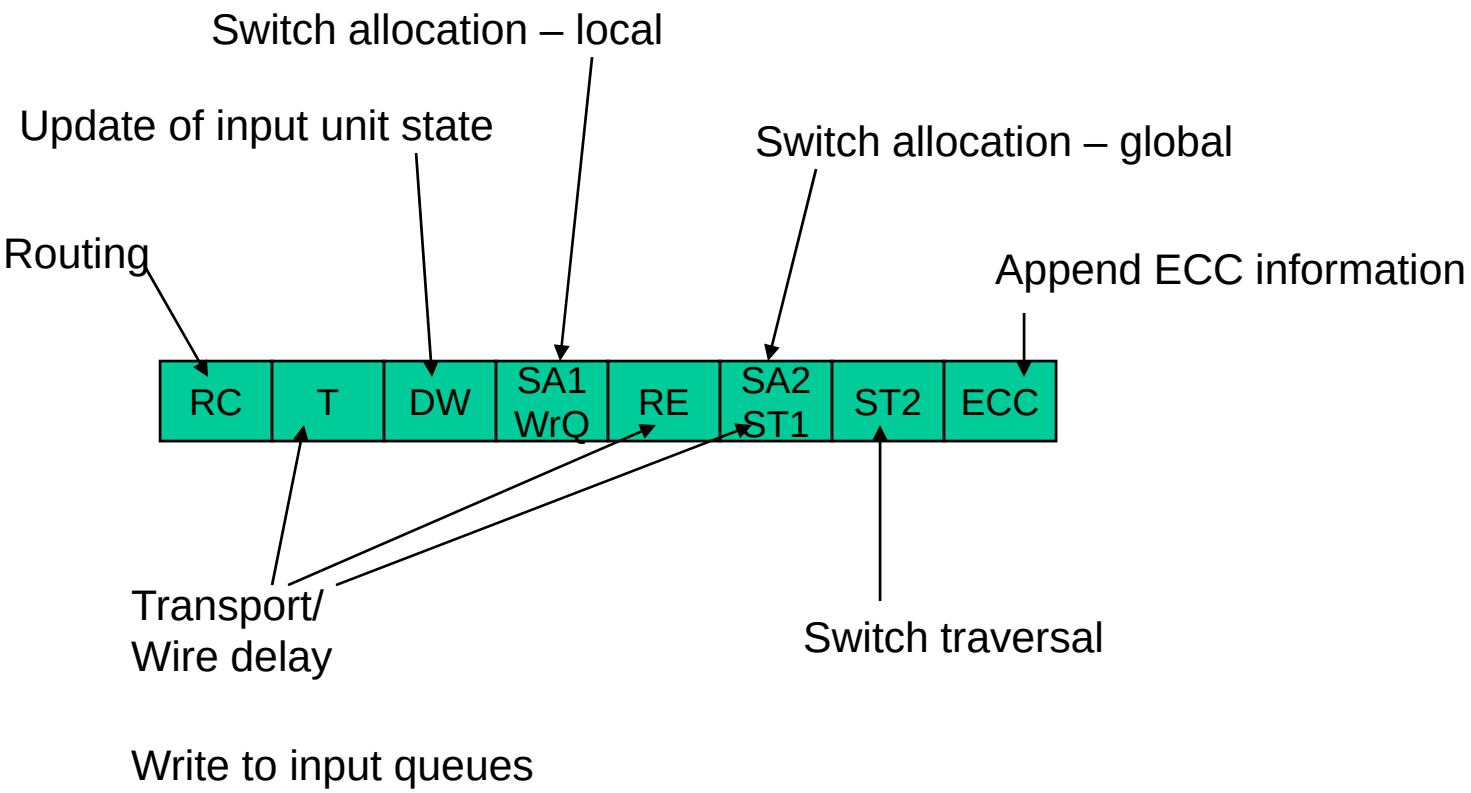


- Router pipeline latency is a greater bottleneck when there is little contention
- When there is little contention, speculation will likely work well!
- Single stage pipeline?

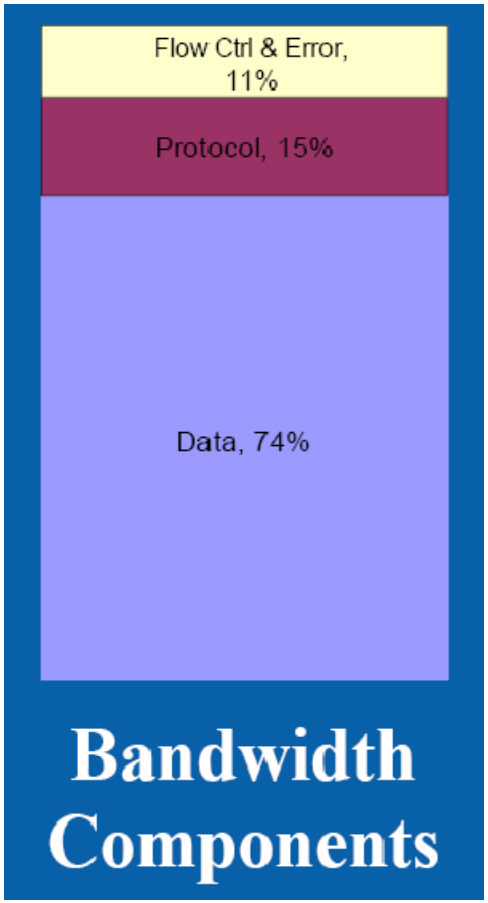
Case Study I: Alpha 21364 Router

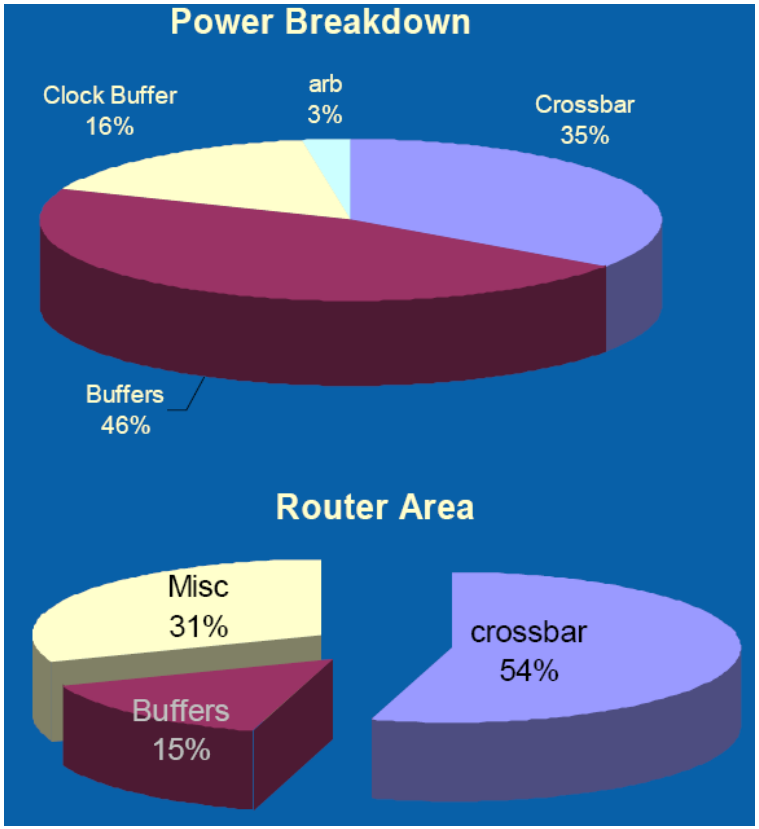
- Integrates a router on-chip to create a multiprocessor building block (up to 128 processors in a 2D torus)
- 4 external ports, deep 8-stage pipeline for high frequency, speculation, adaptive routing, cut-through flow control (resources per packet, the largest packet in the coherence protocol is only 76 B (19 flits), 316 packet buffers per router)
- Physical channels are allocated per packet – VCs enable deadlock avoidance
- Per-hop latency of 10.8 ns (13 processor cycles)

Alpha 21364 Pipeline



Recent Intel Router

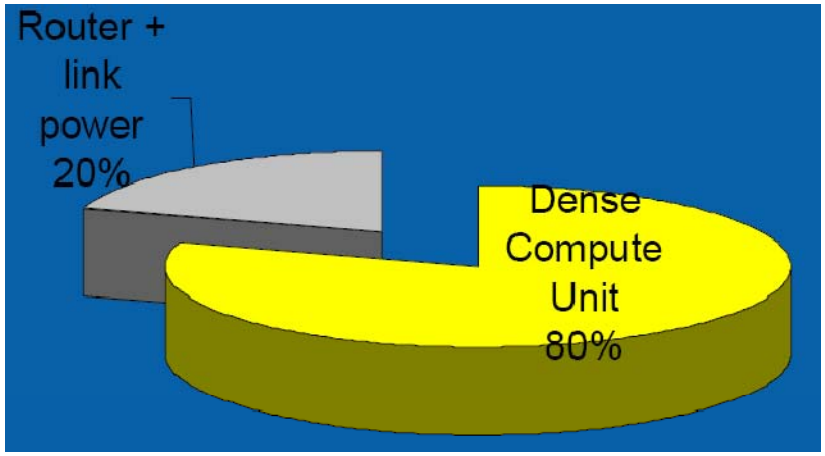


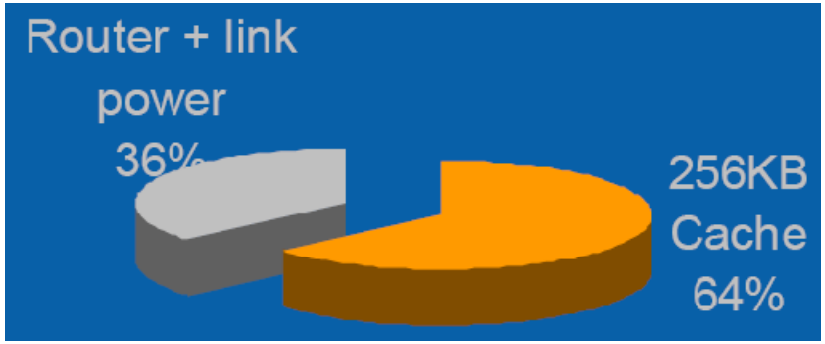


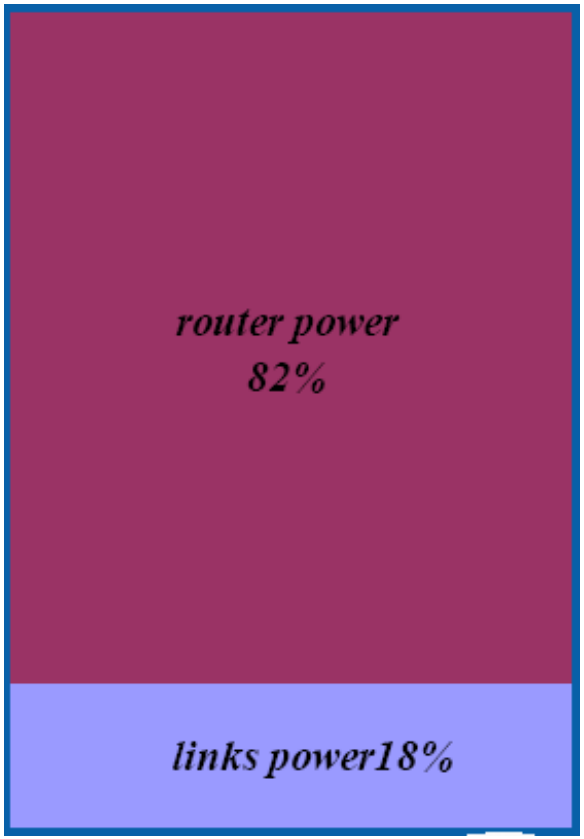
- Used for a 6x6 mesh
- 16 B, > 3 GHz
- Wormhole with VC flow control

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

Recent Intel Router

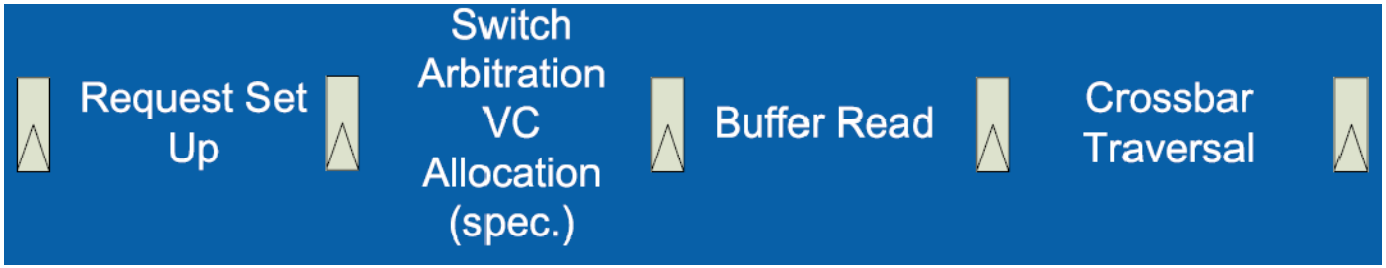


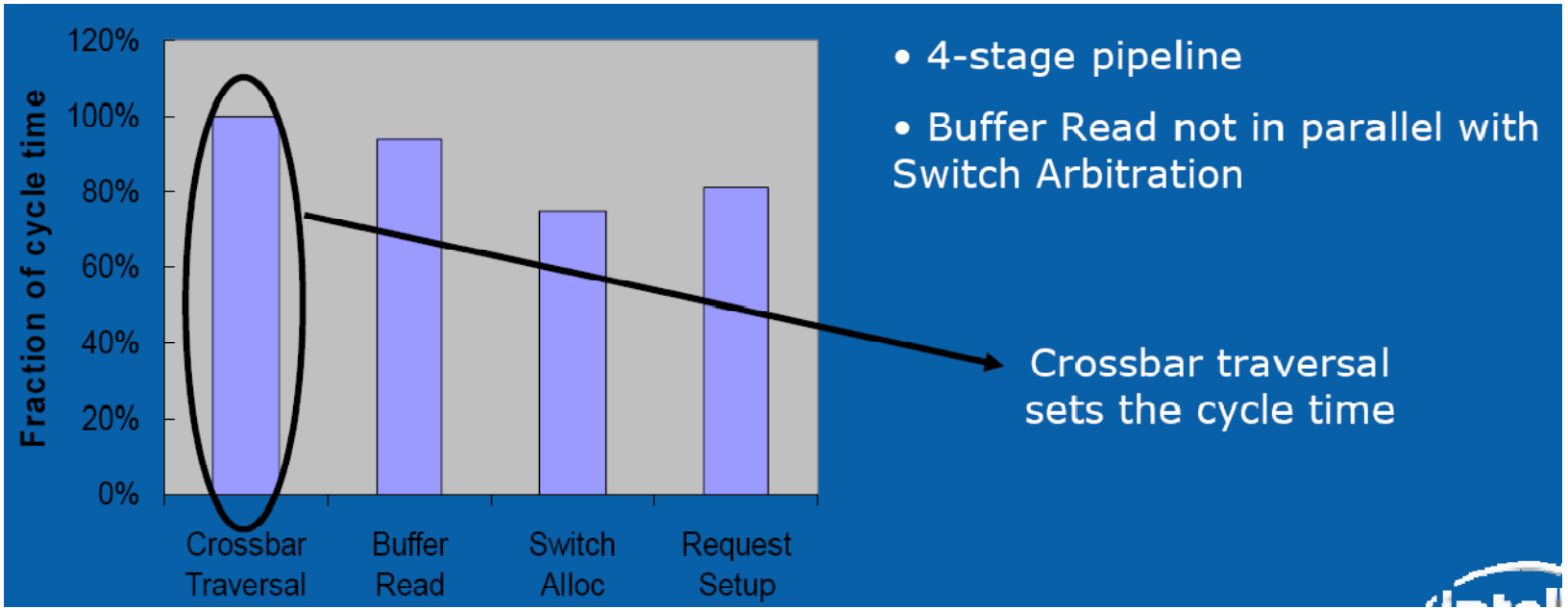




Source: Partha Kundu, “On-Die Interconnects for Next-Generation CMPs”, talk at On-Chip Interconnection Networks Workshop, Dec 2006

Recent Intel Router





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

Title

- Bullet