

Lecture 26: Interconnection Networks

- Topics: flow control, router microarchitecture

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)

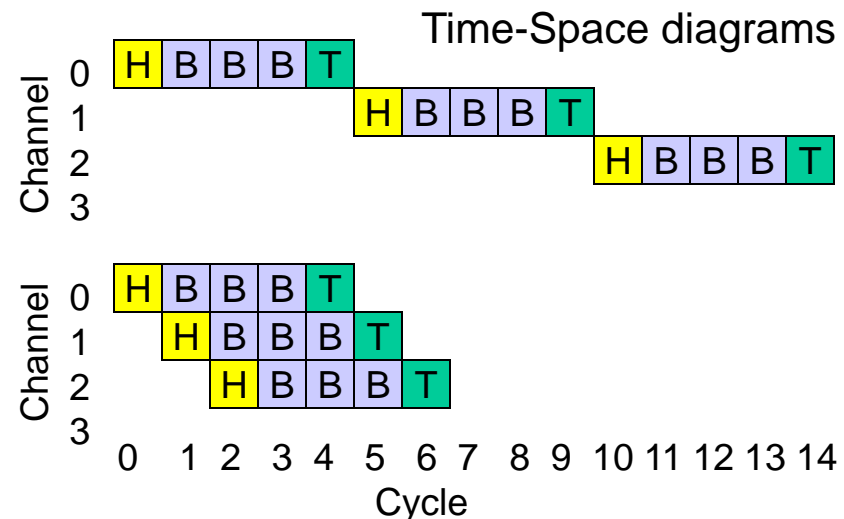
Buffered Flow Control

- A buffer between two channels decouples the resource allocation for each channel

- Packet-buffer flow control: channels and buffers are allocated per packet

- Store-and-forward

- Cut-through



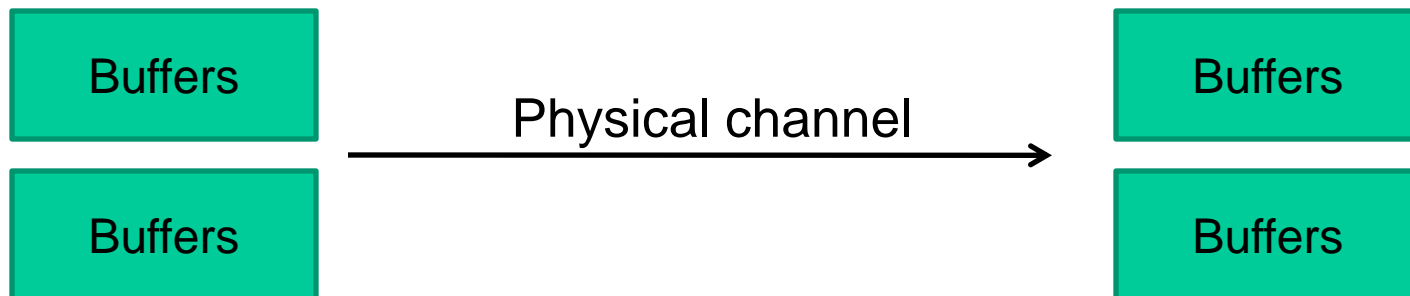
- Wormhole routing: same as cut-through, but buffers in each router are allocated on a per-flit basis, not per-packet

Virtual Channels



Flits do not carry headers. Once a packet starts going over a channel, another packet cannot cut in (else, the receiving buffer confuse the flits of the two packets). If the packet is stalled, other packets can't use the channel.

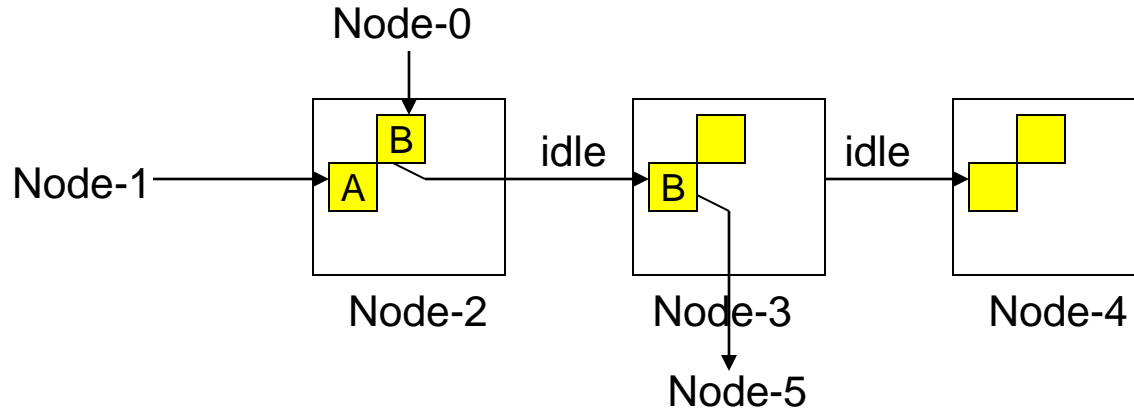
With virtual channels, the flit can be received into one of N buffers. This allows N packets to be in transit over a given physical channel. The packet must carry an ID to indicate its virtual channel.



Example

- **Wormhole:**

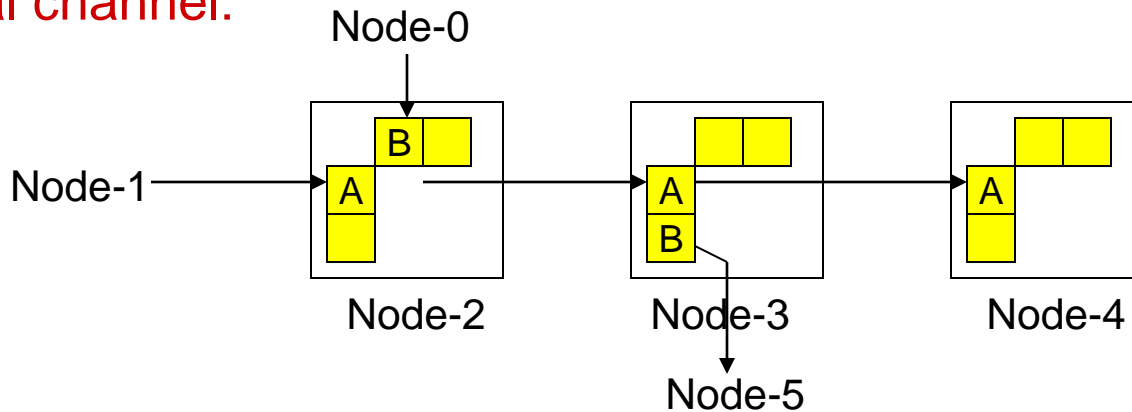
A is going from Node-1 to Node-4; B is going from Node-0 to Node-5



(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)

Virtual Channel Flow Control

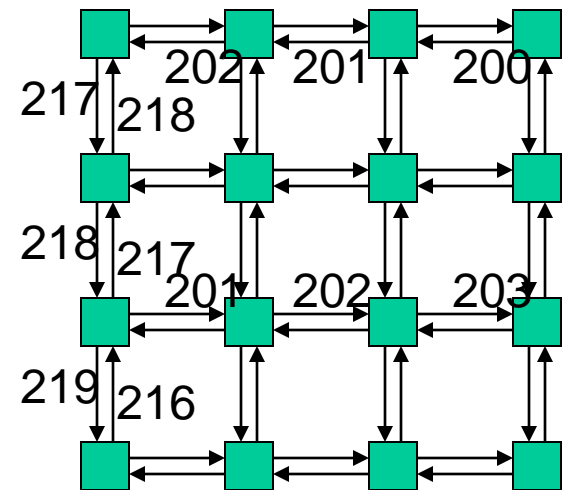
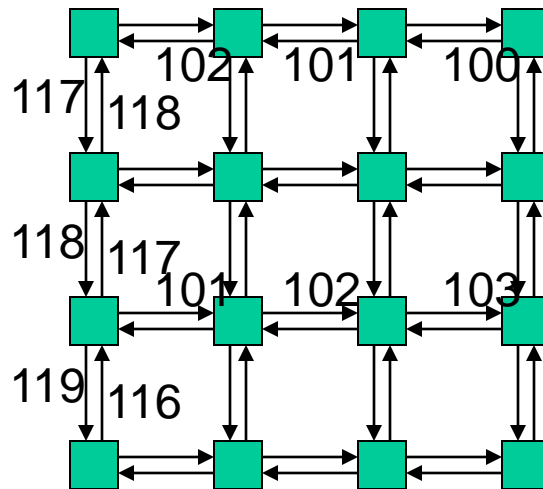
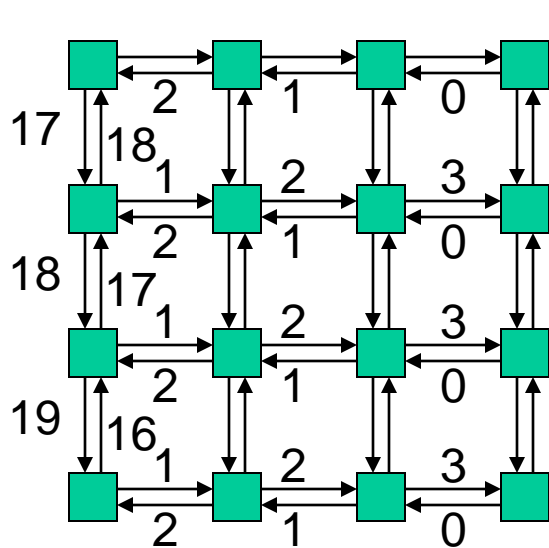
- Incoming flits are placed in buffers
- For this flit to jump to the next router, it must acquire three resources:
 - A free virtual channel on its intended hop
 - We know that a virtual channel is free when the tail flit goes through
 - Free buffer entries for that virtual channel
 - This is determined with credit or on/off management
 - A free cycle on the physical channel
 - Competition among the packets that share a physical channel

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

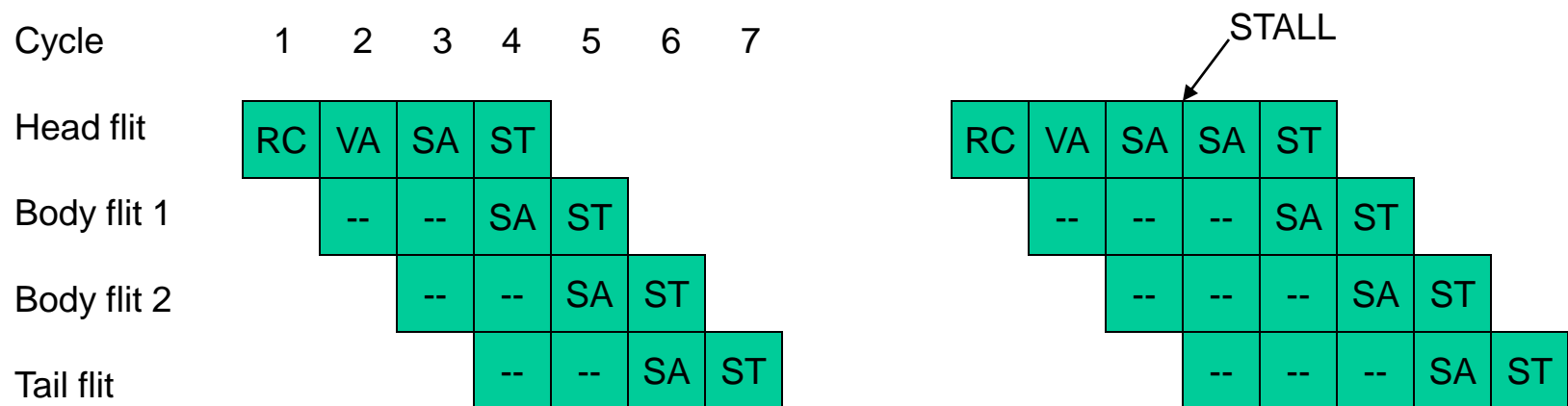
Router Pipeline

- 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

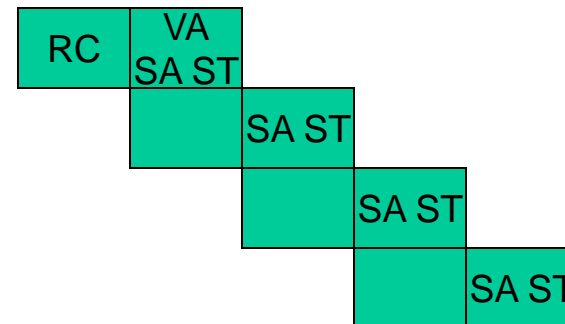
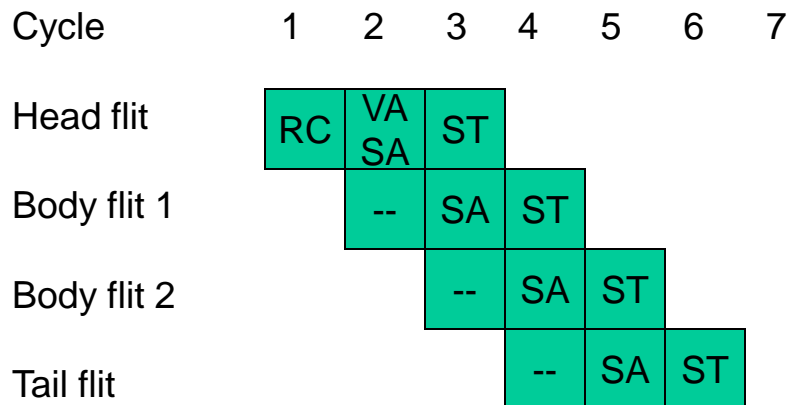
- Four typical stages:
 - RC routing computation: compute the output channel
 - VA virtual-channel allocation: allocate VC for the head flit
 - SA switch allocation: compete for output physical channel
 - ST switch traversal: transfer data on output physical channel



Speculative Pipelines

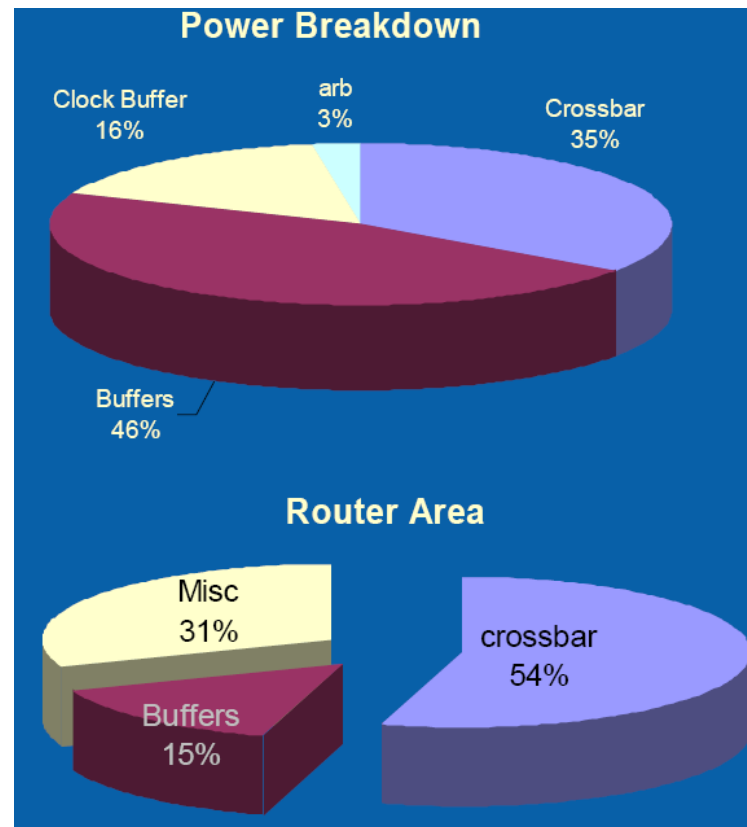
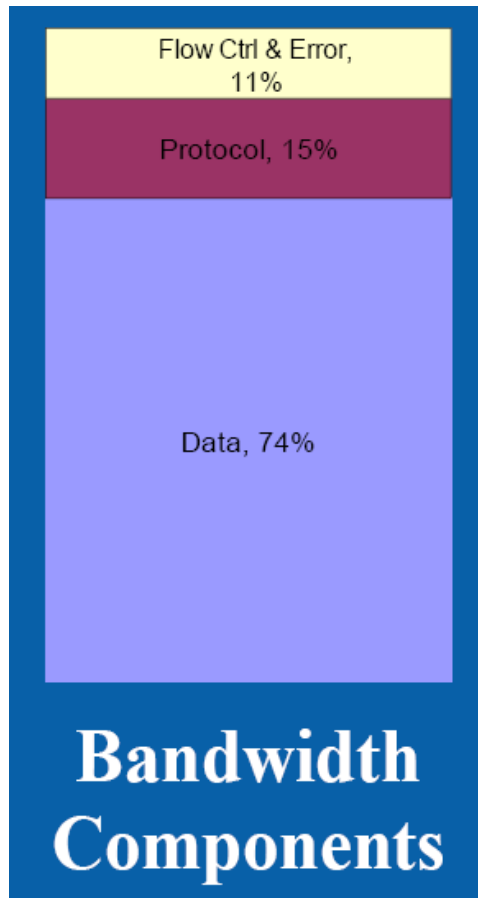
- Perform VA and SA in parallel
- Note that SA only requires knowledge of the output physical channel, not the VC
- If VA fails, the successfully allocated channel goes un-utilized

- 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?

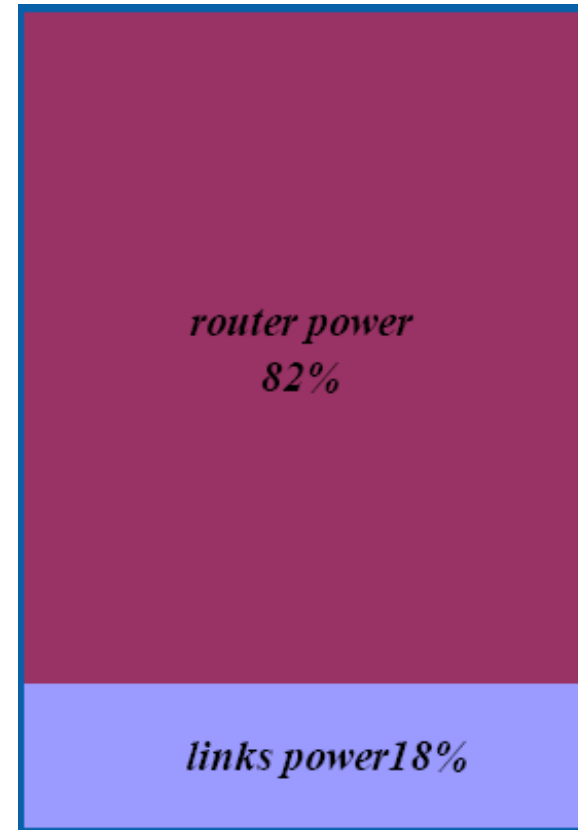
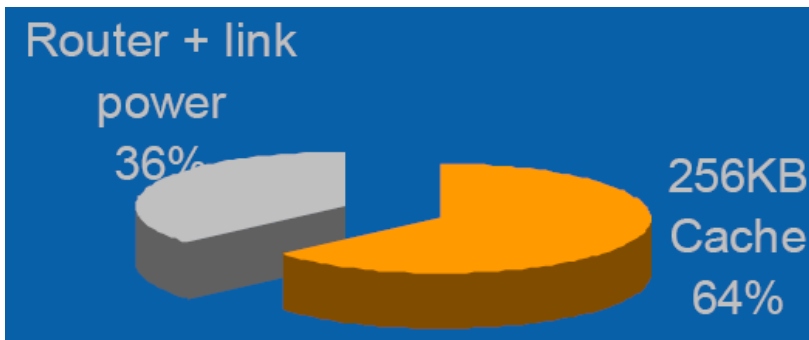
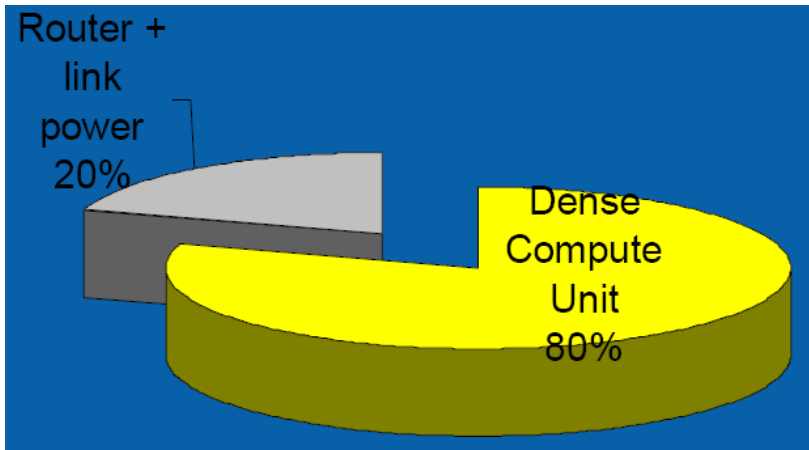
Recent Intel Router



- 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

Title

- Bullet