Interconnection Networks: Flow Control and Microarchitecture

Switching/Flow Control Overview

- Topology: determines connectivity of network
- Routing: determines paths through network
- Flow Control: determine allocation of resources to messages as they traverse network
  - Buffers and links
  - Significant impact on throughput and latency of network

Packets

- Messages: composed of one or more packets
  - If message size is <= maximum packet size only one packet created
- Packets: composed of one or more flits
- Flit: flow control digit
- Phit: physical digit
  - Subdivides flit into chunks = to link width
  - In on-chip networks, flit size == phit size.
    - Due to very wide on-chip channels

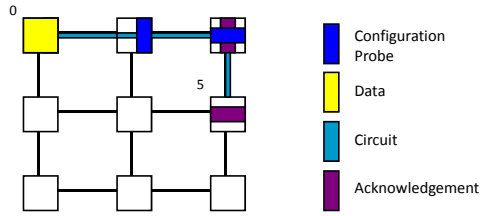
Switching

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

Circuit Switching

- All resources (from source to destination) are allocated to the message prior to transport
  - Probe sent into network to reserve resources
- Once probe sets up circuit
  - Message does not need to perform any routing or allocation at each network hop
  - Good for transferring large amounts of data
    - Can amortize circuit setup cost by sending data with very low per-hop overheads
- No other message can use those resources until transfer is complete
  - Throughput can suffer due setup and hold time for circuits

Circuit Switching Example

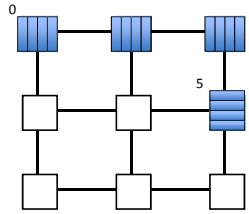


- Significant latency overhead prior to data transfer
- Other requests forced to wait for resources

Packet-based Flow Control

- Store and forward
- Links and buffers are allocated to entire packet
- Head flit waits at router until entire packet is buffered before being forwarded to the next hop
- Not suitable for on-chip
  - Requires buffering at each router to hold entire packet
  - Incurs high latencies (pays serialization latency at each hop)

Store and Forward Example

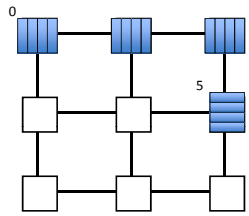


- High per-hop latency
- Larger buffering required

Virtual Cut Through

- Packet-based: similar to Store and Forward
- Links and Buffers allocated to entire packets
- Flits can proceed to next hop before tail flit has been received by current router
  - But only if next router has enough buffer space for entire packet
- Reduces the latency significantly compared to SAF
- But still requires large buffers
  - Unsuitable for on-chip

Virtual Cut Through Example

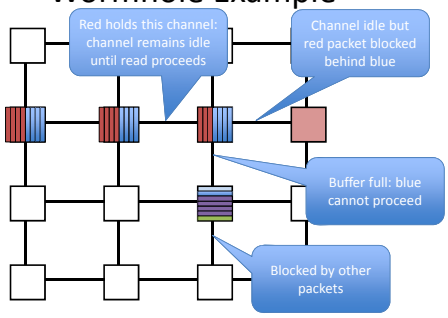


- Lower per-hop latency
- Larger buffering required

Flit Level Flow Control

- Wormhole flow control
- Flit can proceed to next router when there is buffer space available for that flit
  - Improved over SAF and VCT by allocating buffers on a flit-basis
- Pros
  - More efficient buffer utilization (good for on-chip)
  - Low latency
- Cons
  - Poor link utilization: if head flit becomes blocked, all links spanning length of packet are idle
    - Cannot be re-allocated to different packet
    - Suffers from head of line (HOL) blocking

Wormhole Example

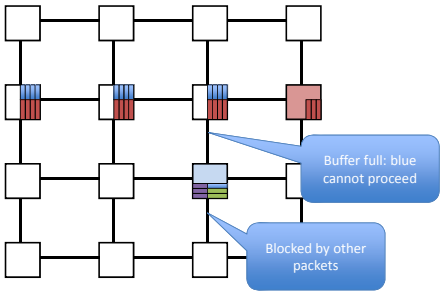


- 6 flit buffers/input port

Virtual Channel Flow Control

- 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

Deadlock

- Using flow control to guarantee deadlock freedom give more flexible routing
- Escape Virtual Channels
  - If routing algorithm is not deadlock free
  - VCs can break resource cycle
  - Place restriction on VC allocation or require one VC to be DOR
- Assign different message classes to different VCs to prevent protocol level deadlock
  - Prevent req-ack message cycles

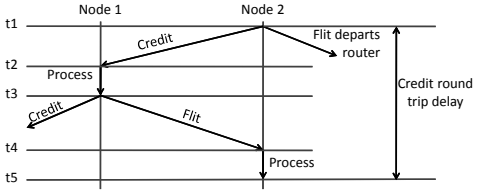
Buffer Backpressure

- Need mechanism to prevent buffer overflow
  - Avoid dropping packets
  - Upstream nodes need to know buffer availability at downstream routers
- Significant impact on throughput achieved by flow control
- Credits
- On-off

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

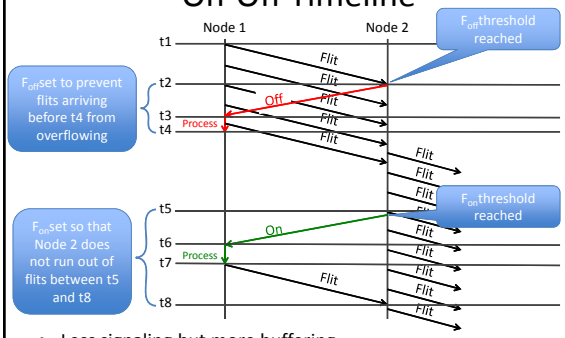


- Round-trip credit delay:
  - Time between when buffer empties and when next flit can be processed from that buffer entry
  - If only single entry buffer, would result in significant throughput degradation
  - Important to size buffers to tolerate credit turn-around

On-Off 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

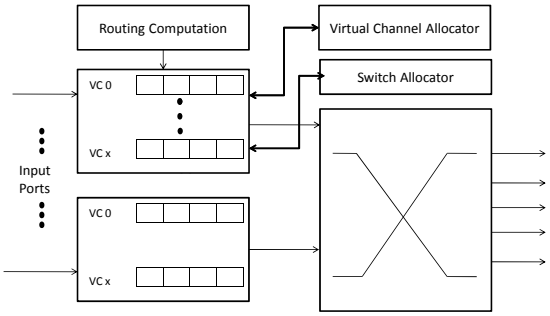
Flow Control Summary

- On-chip networks require techniques with lower buffering requirements
  - Wormhole or Virtual Channel flow control
- Dropping packets unacceptable in on-chip environment
  - Requires buffer backpressure mechanism
- Complexity of flow control impacts router microarchitecture (next)

Router Microarchitecture Overview

- Consist of buffers, switches, functional units, and control logic to implement routing algorithm and flow control
- Focus on microarchitecture of Virtual Channel router
- Router is pipelined to reduce cycle time

Virtual Channel Router

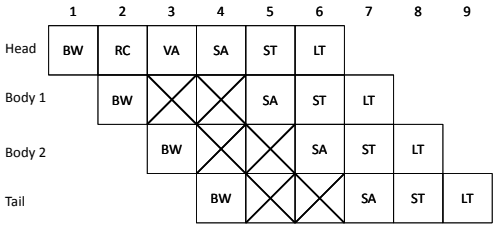


Baseline Router Pipeline

BW	RC	VA	SA	ST	LT

- Canonical 5-stage (+link) pipeline
  - BW: Buffer Write
  - RC: Routing computation
  - VA: Virtual Channel Allocation
  - SA: Switch Allocation
  - ST: Switch Traversal
  - LT: Link Traversal

Baseline Router Pipeline (2)



- Routing computation performed once per packet
- Virtual channel allocated once per packet
- body and tail flits inherit this info from head flit

Router Pipeline Optimizations

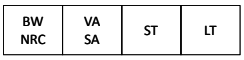
- Baseline (no load) delay

$$= (5cycles + link\ delay) \times hops + t_{serialization}$$

- Ideally, only pay link delay
- Techniques to reduce pipeline stages
  - Lookahead routing: At current router perform routing computation for next router
    - Overlap with BW

Router Pipeline Optimizations (2)

- Speculation
  - Assume that Virtual Channel Allocation stage will be successful
    - Valid under low to moderate loads
  - Entire VA and SA in parallel



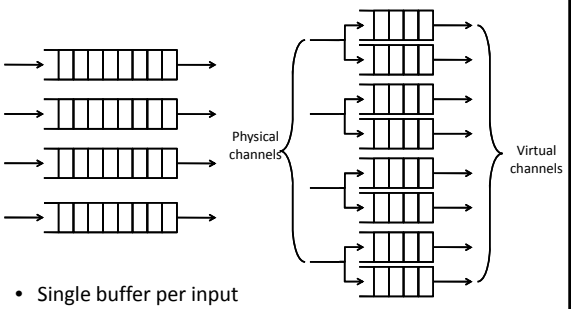
  - If VA unsuccessful (no virtual channel returned)
    - Must repeat VA/SA in next cycle
  - Prioritize non-speculative requests

Router Pipeline Optimizations (3)

- Bypassing: when no flits in input buffer
  - Speculatively enter ST
  - On port conflict, speculation aborted

  - In the first stage, a free VC is allocated, next routing is performed and the crossbar is setup

Buffer Organization



- Single buffer per input
- Multiple fixed length queues per physical channel

Arbiters and Allocators

- *Allocator* matches N requests to M resources
- *Arbiter* matches N requests to 1 resource
- Resources are VCs (for virtual channel routers) and crossbar switch ports.
- Virtual-channel allocator (VA)
  - Resolves contention for output virtual channels
  - Grants them to input virtual channels
- Switch allocator (SA) that grants crossbar switch ports to input virtual channels
- Allocator/arbiter that delivers high matching probability translates to higher network throughput.
  - Must also be fast and able to be pipelined

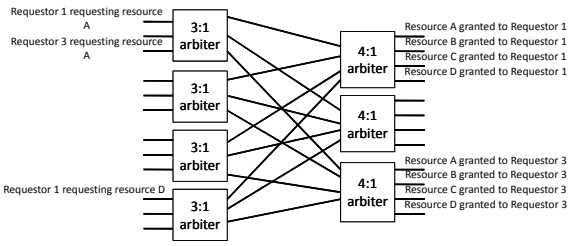
Round Robin Arbiter

- Last request serviced given lowest priority
- Generate the next priority vector from current grant vector
- Exhibits fairness

Matrix Arbiter

- Least recently served priority scheme
- Triangular array of state bits w_{ij} for i<j
  - Bit w_{ij} indicates request i takes priority over j
  - Each time request k granted, clears all bits in row k and sets all bits in column k
- Good for small number of inputs
- Fast, inexpensive and provides strong fairness

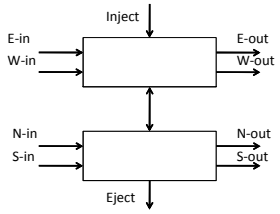
Separable Allocator



- A 3:4 allocator
- First stage: decides which of 3 requestors wins specific resource
- Second stage: ensures requestor is granted just 1 of 4 resources

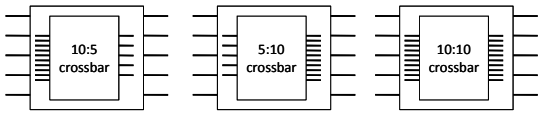
Crossbar Dimension Slicing

- Crossbar area and power grow with $O((pw)^2)$



- Replace 1 5x5 crossbar with 2 3x3 crossbars

Crossbar speedup



- Increase internal switch bandwidth
- Simplifies allocation or gives better performance with a simple allocator
- Output speedup requires output buffers
  - Multiplex onto physical link

Evaluating Interconnection Networks

- Network latency
  - Zero-load latency: average distance * latency per unit distance
- Accepted traffic
  - Measure the max amount of traffic accepted by the network before it reaches saturation
- Cost
  - Power, area, packaging

Interconnection Network Evaluation

- Trace based
  - Synthetic trace-based
    - Injection process
      - Periodic, Bernoulli, Bursty
  - Workload traces
- Full system simulation

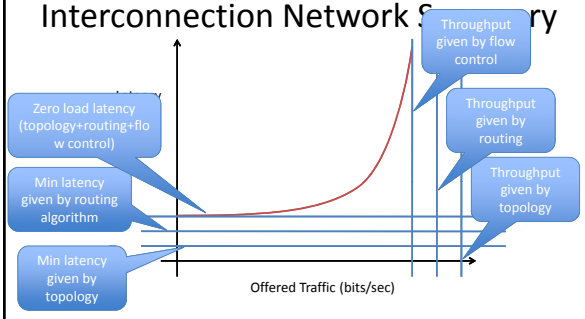
Traffic Patterns

- Uniform Random
  - Each source equally likely to send to each destination
  - Does not do a good job of identifying load imbalances in design
- Permutation (several variations)
  - Each source sends to one destination
- Hot-spot traffic
  - All send to 1 (or small number) of destinations

Microarchitecture Summary

- Ties together topological, routing and flow control design decisions
- Pipelined for fast cycle times
- Area and power constraints important in NoC design space

Interconnection Network S ry



- Latency vs. Offered Traffic

Suggested Reading

- Flow control
  - William J. Dally. Virtual-channel flow control. In Proceedings of the International Symposium on Computer Architecture, 1990.
  - P. Kermani and L. Kleinrock. Virtual cut-through: a new computer communication switching technique. Computer Networks, 3(4):267–286.
  - Jose Duato. A new theory of deadlock-free adaptive routing in wormhole networks. IEEE Transactions on Parallel and Distributed Systems, 4:1320–1331, 1993.
  - Amit Kumar, Li-ShiuanPeh, ParthaKundu, and Niraj K. Jha. Express virtual channels: Toward the ideal interconnection fabric. In Proceedings of 34th Annual International Symposium on Computer Architecture, San Diego, CA, June 2007.
  - Amit Kumar, Li-ShiuanPeh, and Niraj K Jha. Token flow control. In Proceedings of the 41st International Symposium on Microarchitecture, Lake Como, Italy, November 2008.
  - Li-ShiuanPeh and William J. Dally. Flit reservation flow control. In Proceedings of the 6th International Symposium on High Performance Computer Architecture, February 2000.
- Router Microarchitecture
  - Robert Mullins, Andrew West, and Simon Moore. Low-latency virtual-channel routers for on-chip networks. In Proceedings of the International Symposium on Computer Architecture, 2004.
  - Pablo Abad, Valentin Puente, Pablo Prieto, and Jose Angel Gregorio. Rotary router: An efficient architecture for cmp interconnection networks. In Proceedings of the International Symposium on Computer Architecture, pages 116–125, June 2007.
  - Shubhendu S. Mukherjee, PetterBannon, Steven Lang, Aaron Spink, and David Webb. The Alpha 21364 network architecture. IEEE Micro, 22(1):26–35, 2002.
  - Jongman Kim, ChrysostomosNicopoulos, Dongkook Park, Vijaykrishnan Narayanan, Mazin S. Yousif, and Chita R. Das. A gracefully degrading and energy- efficient modular router architecture for on-chip networks. In Proceedings of the International Symposium on Computer Architecture, pages 4–15, June 2006.
  - M. Galles. Scalable pipelined interconnect for distributed endpoint routing: The SGI SPIDER chip. In Proceedings of Hot Interconnects Symposium IV, 1996.