Module 17: "Interconnection Networks"

Lecture 37: "Introduction to Routers"

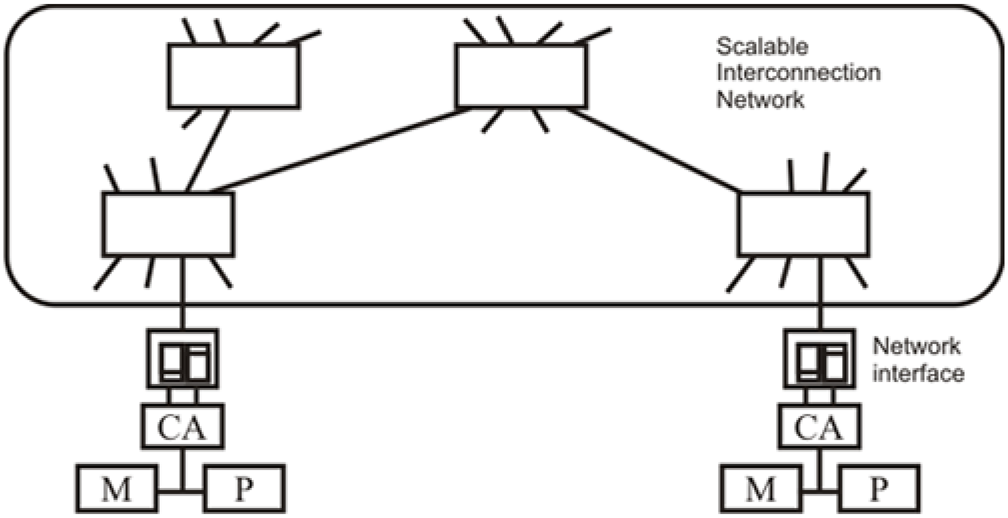
Interconnection Networks

- Fundamentals
- Latency and bandwidth
- Router architecture
- Coherence protocol and routing

[From Chapter 10 of Culler, Singh, Gupta]

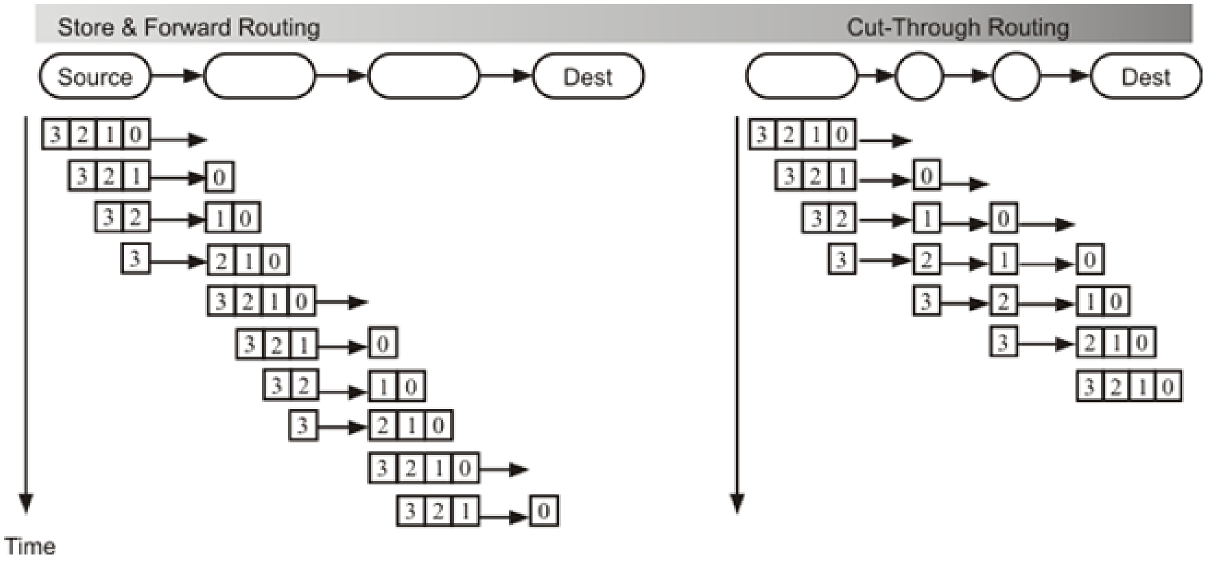
Previous Next

Fundamentals



- The switches or the routers directly talk to the NI
- The NI output and input queues normally map to the virtual channels of the connecting router
- Topology
  - The structure of the interconnect network
  - Direct network: each router is attached to a complete node (most popular)
  - Indirect network: Nodes are attached to few routers only; other routers cannot generate packets, but can only forward them in right direction
- Routing algorithms
  - Deterministic: fixed route between every pair of source and destination
  - Adaptive: based on congestion different routes may be selected dynamically
- Switching strategy
  - Circuit switching: the path from source to destination is first established and reserved before the message is transmitted (popular in phone world, but not in PCA)
  - Packet switching: A message is divided into several packets and each packet carries routing information in its header; leads to better utilization of network resources since individual packets need to be routed only (as opposed to the entire message together)
- Flow control
  - How to detect and avoid resource (buffer, channel, etc.) collision?
  - Minimum unit of information that can be transferred over a link at a time is called flit (flow control unit): may be as small as a phit (physical unit) or as large as a message
- Metrics to compare topology
  - Diameter: maximum shortest distance between any pair
  - Average distance: distance between two arbitrary nodes averaged over all pairs
  - Bisection bandwidth: aggregate bandwidth of minimum set of links which when removed leaves the network as two disjoint roughly equal collection of nodes
- Packet structure
  - Header: contains routing and control information, e.g., source, destination, size of data payload, message opcode, etc.; an intermediate router only needs to inspect the header to handle a newly arrived packet
  - Address: for CC-NUMA machines the cache line address
  - Payload: transmitted data; for CC-NUMA machines this is normally a cache line, or

uncached words, or empty;
    - Trailer: normally contains an error-checking code
- Life of a message in CC-NUMA
    - Starts when the coherence protocol engine (residing in the memory controller) of source node queues the message into one of the NI output queues
    - NI outbound scheduler picks messages from the head of one of the queues possibly according to round-robin scheme
    - The selected message is assembled by NI outbound hardware and is queued in the outgoing virtual channel of the router port connected to NI (from router's viewpoint this is an input port); any payload is copied into a message buffer of that port obeying the copy bandwidth
    - The scheduling algorithm of the router tries to match as many input ports to output ports; this forms the routing delay or hop time
    - The selected packets are pushed into the network obeying the node-to-network bandwidth
- Latency of a message
    - Overhead in NI (at source and dest.) + hop time + channel occupancy (time to push into network) + contention (queuing delay at various places)
    - Store-and-forward routing: each intermediate switch stores the message completely before forwarding it to the next switch; uncontended latency (ignore overhead) = h(n/b+d) where h is the number of hops, n is the size of the message, d is the hop time, b is the node-to-network BW
    - Cut-through routing: as soon as the complete header arrives, routing decision is taken and there is no need to wait for all packets to arrive; uncontended optimistic latency = n/b + hd (much like circuit switching)



Couple of things to notice:

- Time of flight (transmission delay through wires) is negligible
- In cut-through routing formula, the assumption is that routing delay is bigger than channel occupancy of a phit
- Contention control in cut-through routing
    - Virtual cut-through: buffer incoming packet if outgoing port is busy (in the worst case it behaves as store-and-forward)
    - Wormhole routing: allows buffering of few packets inside the router (the packets of a

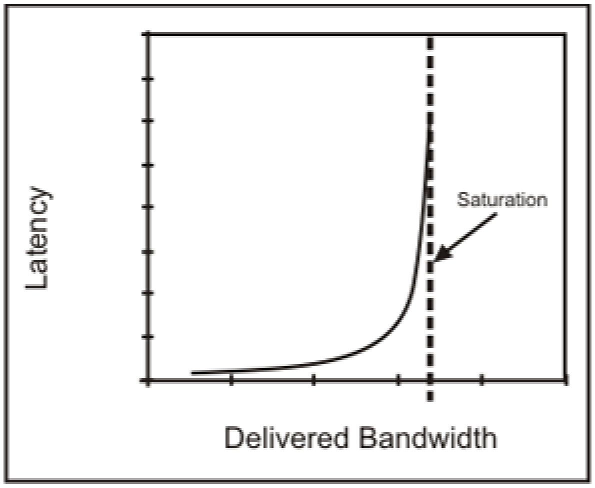
message stay blocked at several routers along the route like a worm)

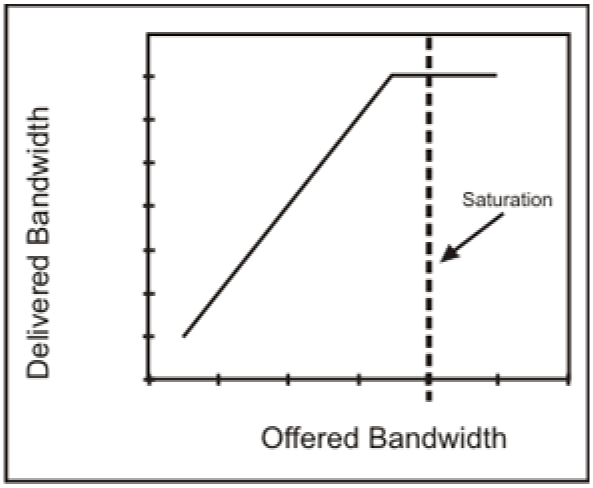
- General contention-control
    - What happens to incoming packets if router buffers are full?
    - General solution in data communication or in WAN is to drop packets and retry based on time-out (TCP/IP, ATM, etc.)
    - In parallel computers packets are normally not dropped; a link-level flow control blocks the packets in the last router’s output port: may cause tree saturation

Previous Next

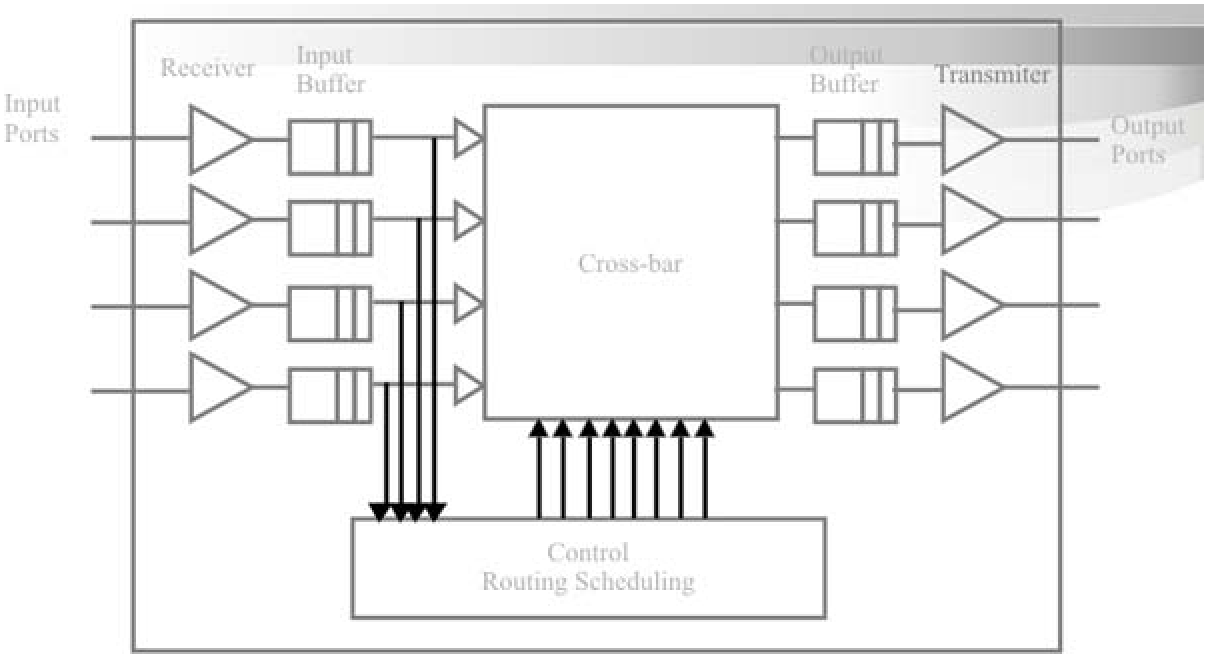
Latency and bandwidth

- Latency gets affected by delivered bandwidth and the delivered bandwidth may be lower than the actual bandwidth under contention, i.e. when bandwidth demand (called offered bandwidth) is much higher than affordable link bandwidth





Router architecture



- Number of input ports is normally equal to the number of output ports which is the degree of the router
- In a direct network one input and one output port would connect to the host node's NI outbound and NI inbound control respectively
- A single VLSI chip
  - Pin count is essentially number of ports (input and output) multiplied by channel width
  - High speed serial links offer lowest pin count, but the clock and control must be encoded within the serial bit stream
  - Parallel links require high pin count and one extra channel is devoted to transmit the clock; also

    must be careful about the cross-channel clock skew
  - Flow control is realized with a separate set of wires comprising ready and acknowledge signals
- Channel buffers
  - With no buffering a blocked packet at an input port may block all subsequent packets at that port creating a tree saturation very fast
  - Input buffering: provide FIFO buffers at each input port; each input port independently requests its output port; one severe problem is head-of-line blocking: two packets from two different ports may ask the same output port and only the selected one can proceed, but the port that is not selected may block subsequent packets destined for a different output port (buffers are FIFO)
  - Output buffering: for each input port partition the buffer storage equally among output ports (normally after crossbar) or provide FIFO buffers per output port; solves head-of-line blocking
  - Shared pool: provide buffer storage (typically SRAM) shared among all input ports; must provide high read/write bandwidth
  - Virtual channels: each input port is equipped with independent virtual channels and an incoming packet is deposited into one of them; normally, a packet is copied from input virtual channel to the same output virtual channel of the requested output port and the virtual channels of an output port are multiplexed onto one physical link
- Output scheduling
  - Simply speaking, each output port can carry out an independent arbitration across all input ports and select one (assumption?)
  - Selection algorithm can be round-robin, oldest first, static priority, etc.
  - For adaptive routing each input port may request multiple output ports and hence output ports cannot arbitrate independently
  - In such a situation, formulate the problem as an online bipartite matching; start with a random selection of requests at each output; assign unselected outputs via improvement iterations

Previous Next

Coherence protocol and routing

- Have already discussed the necessity of at least two queues in each direction in NI; how do they talk to the router?
    - Let’s call the queues as request and reply (in each direction): gets specified by the source coherence engine
    - These queues form request and reply virtual networks in the system
    - Each output queue of NI may map to several input virtual lanes in the router (at least one)
    - Each port of the router has equal number of virtual lanes, e.g. the request virtual lanes form the request virtual network
- The coherence protocol normally does a static assignment of message types to virtual networks
    - A message originating from request lane will be carried along the route in the request network and will arrive at the destination in the input request queue of NI
    - Within each virtual network there may be several virtual channels per port of the router to avoid routing deadlock cycles, head-of-line blocking, and to aid adaptive routing
    - Three-lane protocols normally have a third virtual network to carry requests generated by requests e.g., interventions and invalidations
    - Stanford FLASH runs four-lane coherence protocols and uses all the four virtual lanes of SGI Spider router

Previous Next