Today’s Lecture

Today’s Lecture

Router consists of
- Set of input interfaces where packets arrive
- Set of output interfaces from which packets depart
- Some form of interconnect connecting inputs to outputs

Router implements
- (1) Forward packet to corresponding output interface
- (2) Manage bandwidth and buffer space resources

Generic Architecture

- Input and output interfaces are connected through an interconnect
- Interconnect can be implemented by
  - Shared memory
    - Low capacity routers (e.g., PC-based routers)
  - Shared bus
    - Medium capacity routers
    - Point-to-point (switched) bus
    - High capacity routers

Shared Memory (1st Generation)

Typically < 0.5Gbps aggregate capacity
Limited by rate of shared memory

(*) Slide by Nick McKeown

Shared Bus (2nd Generation)

Typically < 5Gb/s aggregate capacity; Limited by shared bus

(*) Slide by Nick McKeown
**Point-to-Point Switch (3rd Generation)**

Typically ~ 100Gbps aggregate capacity

(*Slide by Nick McKeown*)

---

**Output Queued Routers**

- Only output interfaces store packets
  - **Advantages**
    - Easy to design algorithms: only one congestion point
  - **Disadvantages**
    - Requires an output speedup (Ro/C) of N, where N is the number of interfaces not feasible

---

**What a Router Looks Like**

**Cisco GSR 12416**

- Capacity: 160Gb/s
- Power: 4.2kW

**Juniper M160**

- Capacity: 80Gb/s
- Power: 2.6kW

*Slide by Nick McKeown*

---

**Input Queued Routers**

- Only input interfaces store packets
  - **Advantages**
    - Easy to build
    - Store packets at inputs if contention at outputs
    - Relatively easy to design algorithms
    - Only one congestion point, but not output...
    - Need to implement backpressure...
  - **Disadvantages**
    - Hard to achieve utilization $\approx 1$ (due to output contention, head-of-line blocking)
    - However, theoretical and simulation results show that for realistic traffic an input/output speedup (RI/C) of 2 is enough to achieve utilizations close to 1

---

**Interconnect**

- Point-to-point switch allows simultaneous transfer of packet between any two disjoint pairs of input-output interfaces
- **Goal:** come-up with a schedule that
  - Provides Quality of Service
  - Maximizes router throughput
- **Challenges:**
  - Address head-of-line blocking at inputs
  - Resolve input/output speedups contention
  - Avoid packet dropping at output if possible
- **Note:** packets are fragmented in fix sized cells at inputs and reassembled at outputs

---

**Head-of-line Blocking**

- Cell at head of an input queue cannot be transferred, thus blocking the following cells

*Slide by Nick McKeown*
Disadvantages

- Harder to design algorithms
- Easy to build

Two congestion points

Maintain at each input N virtual queues, i.e., one per output port

Input Interface

- Packet forwarding: decide to which output interface to forward each packet based on the information in packet header
  - Examine packet header
  - Lookup in forwarding table
  - Update packet header

Solution to Avoid Head-of-line Blocking

- Identify the output interface to forward an incoming packet based on packet’s destination address
- Routing tables summarize information by maintaining a mapping between IP address prefixes and output interfaces
  - How are routing tables computed?
  - Route lookup → find the longest prefix in the table that matches the packet destination address

Combined Input-Output Queued (CIOQ) Routers

- Both input and output interfaces store packets
- Advantages
  - Easy to build
  - Utilization 1 can be achieved with limited input/output speedup (≈ 2)
- Disadvantages
  - Harder to design algorithms
  - Two congestion points
  - Need to design flow control

Input Interface

- Packet with destination address 12.82.100.101 is sent to interface 2, as 12.82.100.xxx is the longest prefix matching packet’s destination address
Patricia Tries

- Use binary tree paths to encode prefixes

- Advantage: simple to implement
- Disadvantage: one lookup may take $O(m)$, where $m$ is number of bits (32 in the case of IPv4)

Another Forwarding Technique: Source Routing

- Each packet specifies the sequence of routers, or alternatively the sequence of output ports, from source to destination

Source Routing (cont’d)

- Gives the source control of the path
- Not scalable
  - Packet overhead proportional to the number of routers
  - Typically, require variable header length which is harder to implement
- Hard for source to have complete information
- Loose source routing $\rightarrow$ sender specifies only a subset of routers along the path

Output Functions

- Buffer management: decide when and which packet to drop
- Scheduler: decide when and which packet to transmit

Example: FIFO router

- Most of today’s routers
- Drop-tail buffer management: when buffer is full drop the incoming packet
- First-In-First-Out (FIFO) Scheduling: schedule packets in the same order they arrive

Source Routing (cont’d)

- Packet classification: map each packet to a predefined flow/connection (for datagram forwarding)
  - Use to implement more sophisticated services (e.g., QoS)
- Flow: a subset of packets between any two endpoints in the network

Output Functions (cont’d)
Packet Classification

- Classify an IP packet based on a number of fields in the packet header, e.g.,
  - source/destination IP address (32 bits)
  - source/destination port number (16 bits)
  - Type of service (TOS) byte (8 bits)
  - Type of protocol (8 bits)
- In general fields are specified by range

Example of Classification Rules

- Access-control in firewalls
  - Deny all e-mail traffic from ISP-X to Y
- Policy-based routing
  - Route IP telephony traffic from X to Y via ATM
- Differentiate quality of service
  - Ensure that no more than 50 Mbps are injected from ISP-X

Scheduler

- One queue per flow
- Scheduler decides when and from which queue to send a packet
  - Each queue is FIFO
- Goals of a scheduler:
  - Quality of service
  - Protection (stop a flow from hogging the entire output link)
  - Fast!

Example: Priority Scheduler

- Priority scheduler: packets in the highest priority queue are always served before the packets in lower priority queues

Example: Round Robin Scheduler

- Round robin: packets are served in a round-robin fashion

Discussion

- Priority scheduler vs. Round-robin scheduler
  - What are advantages and disadvantages of each scheduler?