Lecture 11 (Router Architecture and IP Datagram)

1 Basics

1. Routers have n incoming and outgoing ports where $n \geq 2$
2. It has a switching fabric which handles the transfer from input to output port
3. There exists a routing processor which decides the control of the data
4. Efficient router processes data at "wire speed"

2 Input Port Functions

1. Line termination (physical layer)
2. Link layer protocol then receives the data
3. Decentralised switching:
   - performs lookup, forwarding, queuing
   - lookup is done by "looking-up" header field values in the forwarding table (match + action)
     i. destination-based forwarding
     ii. generalised forwarding (based on any set of header values)
   - queuing is done if input rate is faster than forwarding speed
   - aim is to prevent queuing as much as possible

2.1 Destination Based Forwarding

1. Forwarding table has a map of range of destination addresses with link interface
2. Overlapping mapping may exist
3. To resolve overlap, the longer prefix mapping is used
4. Matching is done using Ternary Content Addressable Memories (TCAMs), it returns the address in 1 clock cycle

3 Switching Fabrics

1. *Switching Rate* is rate at which packets can be tranferred from input to output
2. Ideal switching rate is n times the line rate
3. It is of three types:

i. Memory
   - single transfer at a time, links change via memory
   - first generation routers used this type
ii. Bus
   - common bus on which the packet travels on when moving from input to output
iii. Interconnection network
   - $n \times n$ connection for direct movement
   - parallel transfer by breaking and joining datagram also exists in more advanced versions using -
     a. smaller switches
     b. parallel planes

4 Input Port Queuing

4.1 Head of the Line (HOL) Blocking

Head of the queue at input blocks the flow of the packets after it whose output ports have no traffic

5 Output Port Queuing

1. To prevent HOL, queuing is done at output
2. Drop policy has to be implemented
3. Priority scheduling can be done (but net-neutrality a big issue)

5.1 Buffering in the Queue

1. Average buffering is usually $RTT \times C$ (C is link capacity)
2. The recent recommendation is $\frac{RTT \cdot C}{\sqrt{n}}$
3. Too much buffering can lead to delays
4. Packets that can be potentially dropped can be marked so that the sender can identify the same and choose a different route in future (ECN flag in TCP connection)

5.2 Packet Scheduling

1. FCFS policy
2. Priority-based
3. Round robin (input link-wise)
4. Weighted fair queuing (round robin with weights)

6 Net Neutrality

1. No blocking

2. No throttling
3. No paid prioritization

7 IP Datagram

1. Version - 3 bits
2. Header length - 5 bits
3. Type of service - 8 bits
4. Length (of data) - 16 bits
5. Identifier - 16 bits
6. Flags + fragment offset - 16 bits
7. Time to live - 8 bits
8. Upper layer (TCP/UDP) - 8 bits
9. Header checksum - 16 bits
10. Source IP - 32 bits
11. Destination IP - 32 bits
12. Options

Total length of IP header is 20 bytes + options