Lecture 17: Router Design
Lecture 14 Overview

- End-to-end lookup and forwarding example

- Router internals
  - Buffering
  - Scheduling
Example: Sending to CNN

www.cnn.com
Basic Steps

1. Host A must learn the IP address of B via DNS
2. Host A uses gateway R to reach external hosts
3. Router R forwards IP packet to outgoing interface
4. Router R learns B’s MAC address and forwards frame
Host A Learns B’s IP Address

- Host A does a DNS query to learn B’s address
  - Suppose gethostbyname() returns 222.222.222.222
- Host A constructs an IP packet to send to B
  - Source 111.111.111.111, dest 222.222.222.222
Host A Learns B’s IP Address

- **IP packet**
  - From A: 111.111.111.111
  - To B: 222.222.222.222

- **Ethernet frame**
  - From A: 74-29-9C-E8-FF-55
  - To gateway: ????

---

[Diagram]

- LAN 1
  - Host A: 111.111.111.111
  - Host B: CC-49-DE-D0-AB-7D
  - LAN 1: 111.111.111.110

- ROUTER
  - E6-E9-00-17-BB-4B
  - 1A-23-F9-CD-06-9B

- LAN 2
  - Host R: 111.111.111.112
  - Host B: 222.222.222.222
  - LAN 2: 222.222.222.221

- Gateway
  - 88-B2-2F-54-1A-0F
  - 49-BD-D2-C7-58-2A
A Decides to Send Through $R$

- Host $A$ has a gateway router $R$
  - Used to reach dests outside of 111.111.111.0/24
  - Address 111.111.111.110 for $R$ learned via DHCP
- But, what is the MAC address of the gateway?
Host A learns the MAC address of R’s interface
  - ARP request: broadcast request for 111.111.111.110
  - ARP response: R responds with E6-E9-00-17-BB-4B

Host A encapsulates the packet and sends to R
A Sends Packet Through R

- IP packet
  - From A: 111.111.111.111
  - To B: 222.222.222.222

- Ethernet frame
  - From A: 74-29-9C-E8-FF-55
  - To R: E6-E9-00-17-BB-4B
Router $R$’s adapter receives the packet
- $R$ extracts the IP packet destined to 222.222.222.222

Router $R$ consults its forwarding table
- Packet matches 222.222.222.0/24 via other interface
R Wants to Forward Packet

- IP packet
  - From A: 111.111.111.111
  - To B: 222.222.222.222

- Ethernet frame
  - From R: 1A-23-F9-CD-06-9B
  - To B: ???
Router $R$ sends a packet to $B$.

- $R$ learns the MAC address of host $B$.
  - ARP request: broadcast request for 222.222.222.222
  - ARP response: $B$ responds with 49-BD-D2-C7-56-2A

- Router $R$ encapsulates the packet and sends it to $B$. 

---

**Diagram:**

- Host $A$ sends a packet to host $B$.
- Router $R$ receives the packet and forwards it to $B$. 
- $B$ responds with a MAC address.

---

**Notes:**

- MAC addresses and IP addresses are shown.
- The diagram illustrates the network topology with hosts and routers.
R Wants to Forward Packet

- **IP packet**
  - From A: 111.111.111.111
  - To B: 222.222.222.222

- **Ethernet frame**
  - From R: 1A-23-F9-CD-06-9B
  - To B: 49-BD-D2-C7-56-2A
What’s in a Router?

- **Physical components**
  - One or more **input interfaces** that receive packets
  - One or more **output interfaces** that transmit packets
  - A chassis (box + power) to hold it all

- **Functions**
  - **Forward** packets
  - **Drop** packets (congestion, security, QoS)
  - **Delay** packets (QoS)
  - **Transform** packets? (Encapsulation, Tunneling)
Router Functions

1. Receive incoming packet from link input interface
2. Lookup packet destination in forwarding table (destination, output port(s))
3. Validate checksum, decrement ttl, update checksum
4. Buffer packet in input queue
5. Send packet to output interface (interfaces?)
6. Buffer packet in output queue
7. Send packet to output interface link
Control Plane
- Complex
- Per-control action
- May be slow

Data plane
- Simple
- Per-packet
- Must be fast
Interconnect architecture

- Input & output connected via switch fabric

- Kinds of switch fabric
  - Shared Memory
  - Bus
  - Crossbar

- How to deal with transient contention?
  - Input queuing
  - Output queuing
First Generation Routers

Single CPU and shared memory;
All classification by main CPU
Second Generation Routers

- CPU
- Route Table
- Shared Bus(es)
- Direct DMA on cache hit
- Line Card Buffers
  - Forwarding Cache
  - MAC
- Cache of recent routes
Third Generation Routers

- Switch Fabric
  - Shared interconnect (frequently crossbar)
  - Centralized scheduler
  - Full forwarding table in line card
  - Fixed cells

- Line Card
  - Buffers
  - Forwarding Table
  - MAC

- CPU Card
  - CPU
  - Routing Table

- Line Card
  - Buffers
  - Forwarding Table
  - MAC
Output queuing

- Output interfaces buffer packets

**Pro**
- Simple algorithms
- Single congestion point

**Con**
- N inputs may send to the same output
- Requires *speedup* of N
  » Output ports must be N times faster than input ports
Input queuing

- Input interfaces buffer packets

- Pro
  - Single congestion point
  - Simple to design algorithms

- Con
  - Must implement flow control
  - Low utilization due to Head-of-Line (HoL) Blocking
Head-of-Line Blocking
IQ + Virtual Output Queuing

- Input interfaces buffer packets in per-output virtual queues

- **Pro**
  - Solves blocking problem

- **Con**
  - More resources per port
  - Complex arbiter at switch
  - Still limited by input/output contention (scheduler)
Virtual Output Queues
Switch scheduling

- Problem
  - Match inputs and outputs
  - Resolve contentions, no packet drops
  - Maximize throughput
  - Do it in constant time…

- If traffic is uniformly distributed its easy
  - Lots of algorithms (approximate matching)

- Seminal result (Dai et al, 2000)
  - Maximal size matching + \textit{speedup} of two guarantees
    100\% utilization for most traffic assumptions
Typical high-performance router

- IQ + VoQ + OQ
  - Speedup of 2
  - Central scheduler
  - Fixed-sized internal cells
- Pro
  - Can achieve utilization of 1
  - Can scale to > Tbps
- Con
  - Multiple congestion points
  - Complexity
For Next Time

- Read P&D 4.2