ELEC / COMP 177 – Fall 2011

Computer Networking
→ Address Resolution Protocol

Some slides from Kurose and Ross, *Computer Networking*, 5th Edition

Schedule

- Thursday, Nov 3rd – **Homework #4 Due**
- Thursday, Nov 10th – **Project #2 Due**

Recap – IP Addresses

- IPv4 addresses are usually displayed in dotted decimal notation
  - Each byte represented by decimal value
  - Bytes are separated by a period
  - IP address `0x8002C2F2 = 128.2.194.242`
- IP addresses are hierarchical
  - Contain information on *where* a host is in the world
  - Address composed of network ID and host ID
  - www.pacific.edu: `138.9.110.12`

Recap – CIDR

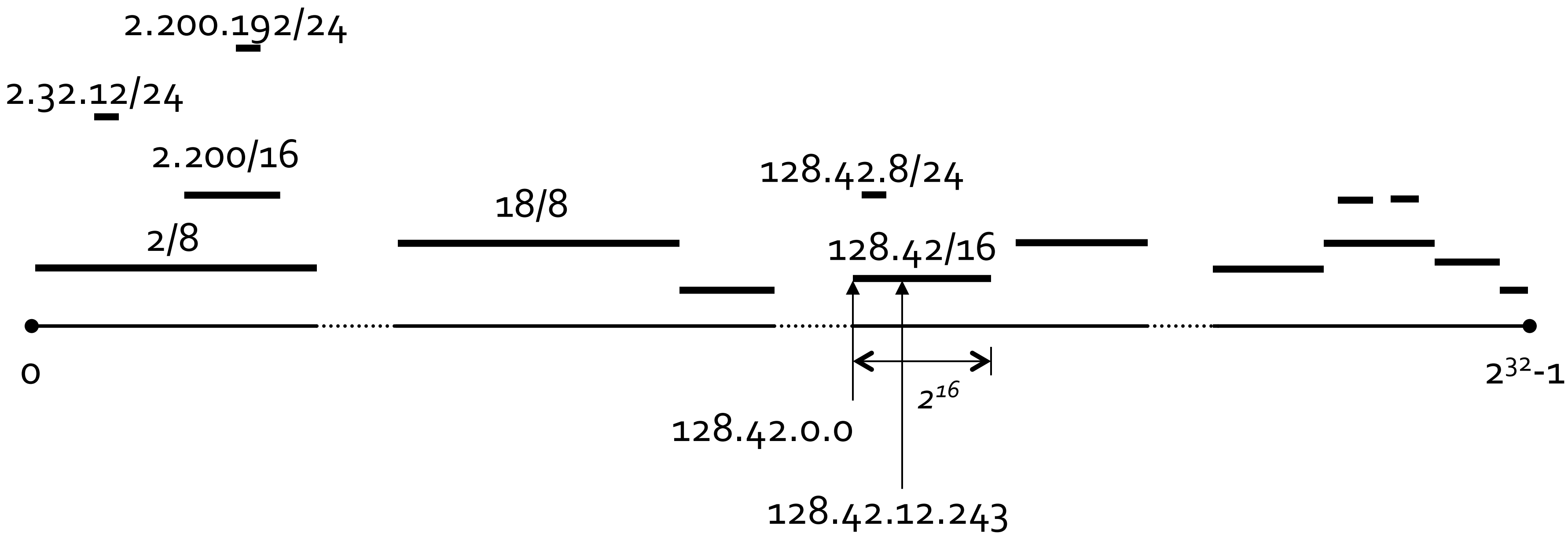
- Classless InterDomain Routing
  - Meant to provide more flexible routing
- Break up IP address space into prefixes
- Each prefix has its own routing entry
  - All traffic to Pacific (138.9/16) within the Internet should be routed the same way, regardless of how Pacific subnets its address space
  - 138.9/16 =

10001010 00001001 xxxxxxxx xxxxxxxx

Prefix (138.9) is **16 bits long**

16 bits remain to represent hosts in this prefix (aka subnet)

Reap – IP Prefixes



- IP address space can be viewed as a number line
  - Each segment represents an aggregated route
  - Segments can overlap
- Look for smallest segment that matches the destination address

Recap – Longest Prefix Match

- Allow more specific entries to supersede more general ones
  - 128.42.8/24
    - Route this traffic to Italy
  - 128.42/16
    - Route this traffic to Houston
    - Except for addresses that match a route with a longer prefix (i.e., 128.42.8/24)
- Allows significantly more route aggregation
- Simplifies things if companies move (physically or to another ISP) their block of IP addresses

Address Resolution Protocol

Routing

- IP routers determine next hop IP address using longest prefix match
  - Given a destination IP address, what is the IP address of the router to which the datagram should be forwarded next?
- Why do they need the next hop IP address?
  - Destination IP address remains unchanged in the datagram
  - Actually need the next hop *Ethernet* address!

IP vs. Ethernet Addresses

- Why store next hop IP address?
- Why not just store next hop Ethernet address?

- Ethernet address
  - Unique identifier for a piece of hardware
- IP address
  - Unique identifier for a system performing a function

Address Resolution Protocol

- Find link layer address given a network layer address
  - i.e., what is the Ethernet address for a given IP address?
- Every IP node (hosts and routers) has an ARP table
  - Mapping from IP to Ethernet addresses on their LAN
  - May be incomplete
  - Can include both static and dynamic entries

Static ARP Entries

- IP address `128.42.6.32` has Ethernet address `00:E0:81:5A:71:AB`
- Must be managed by the system administrator
  - What if that NIC fails and is replaced?
  - What if that system's IP address is changed?
- Actively managing IP $\longrightarrow$ Ethernet address mappings for all nodes in a LAN will get pretty tedious pretty fast

Dynamic ARP Entries

- Systems "discover" IP $\longrightarrow$ Ethernet address mappings, as needed
- Each entry has an IP address, an Ethernet address, and a timeout (typically around 20 minutes)
- ARP packets are broadcast on the LAN to discover mappings
  - ARP packets are encapsulated in Ethernet frames
  - **Why broadcast?**

Encapsulated ARP Request

Bytes:	6	6	2	28	18	4
	DA	SA	Type	ARP Packet	Pad	CRC

- Ethernet destination
  - If you knew the Ethernet address, you wouldn't need to send an ARP request!
  - Broadcast address: `ff:ff:ff:ff:ff:ff`
- Ethernet type
  - ARP: `0x0806`

ARP Request

- Hardware Type (2 bytes)
  - Ethernet `0x0001`
- Protocol Type (2 bytes)
  - IP: `0x0800`
- Hardware Address Length (1 byte)
  - Ethernet: 6
- Protocol Address Length (1 byte)
  - IPv4: 4
- Opcode (2 bytes)
  - Request: 1 (Response: 2)
- Source address (hardware and protocol)
- Destination address (hardware (unknown) and protocol)

Full ARP Request

<table>
<tr><td colspan="3">Destination MAC Address (ff:ff:ff:ff:ff:ff)</td></tr>
<tr><td>Destination MAC Address</td><td colspan="2">Source MAC Address</td></tr>
<tr><td colspan="3">Source MAC Address</td></tr>
<tr><td>Type (0x0806)</td><td colspan="2">HW Type (Ethernet: 0x0001)</td></tr>
<tr><td>Protocol Type (IP: 0x0800)</td><td>HW AddrLen (6)</td><td>Prot AddrLen (4)</td></tr>
<tr><td>Opcode (Request: 1)</td><td colspan="2">Source HW Address</td></tr>
<tr><td colspan="3">Source HW Address</td></tr>
<tr><td colspan="3">Source IP Address</td></tr>
<tr><td colspan="3">Destination HW Address</td></tr>
<tr><td>Destination HW Address</td><td colspan="2">Destination IP Address</td></tr>
<tr><td>Destination IP Address</td><td colspan="2">Padding</td></tr>
<tr><td colspan="3">Ethernet CRC</td></tr>
</table>

Sending a Packet from a Host

- Host setup
  - IP address
  - Subnet – what IP addresses are on the same LAN
  - Gateway – where to send traffic outside the LAN

Sending a Packet from a Host

- Example:
  - My IP = 183.19.205.67
  - My Netmask = 255.255.254.0
  - DestIP = 183.19.204.83 – **Same subnet or not?**
- What is my subnet?
  - MyIP= 10110011 00010011 11001101 01000011
  - Netm= 11111111 11111111 11111110 00000000
- Is the destination on the same subnet?
  - DestIP = 10110011 00010011 11001100 01010011
  - Netm = 11111111 11111111 11111110 00000000
  - **Yes! The subnet addresses match**

Sending a Packet from a Host

- Destination **on LAN**
  - Create ARP request for *destination IP*
  - Broadcast to everyone on the LAN
  - Destination should reply with its MAC address
- Destination **not on LAN**
  - Create ARP request for *gateway router IP*
  - Broadcast to everyone on the LAN
  - Gateway should reply with its MAC address

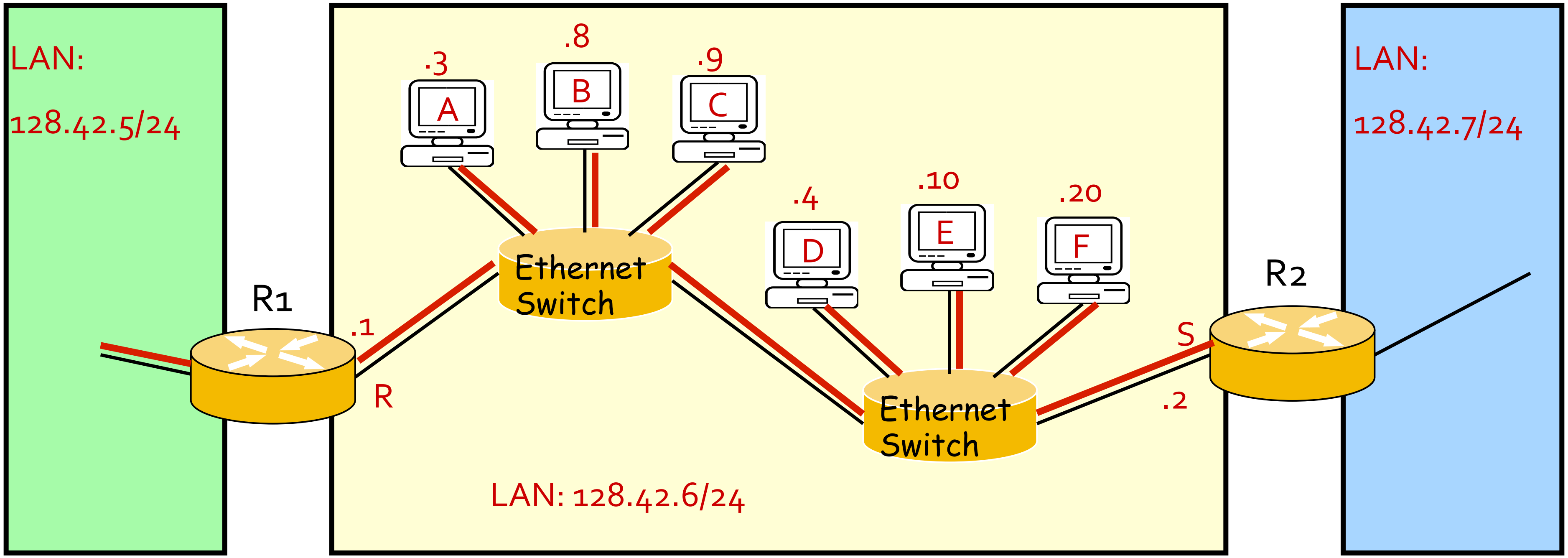
Learning MAC addresses

- Hosts learn IP → Ethernet address mappings
  - ARP responses are stored in ARP tables
  - ARP requests are stored in ARP tables (whether the host is the target or not!)
- ARP entries time out
  - Allow machines to change IP and/or MAC addresses transparently
  - Eliminate stale entries (machines turn off, move, crash, etc.)

Forwarding a Packet in a Router

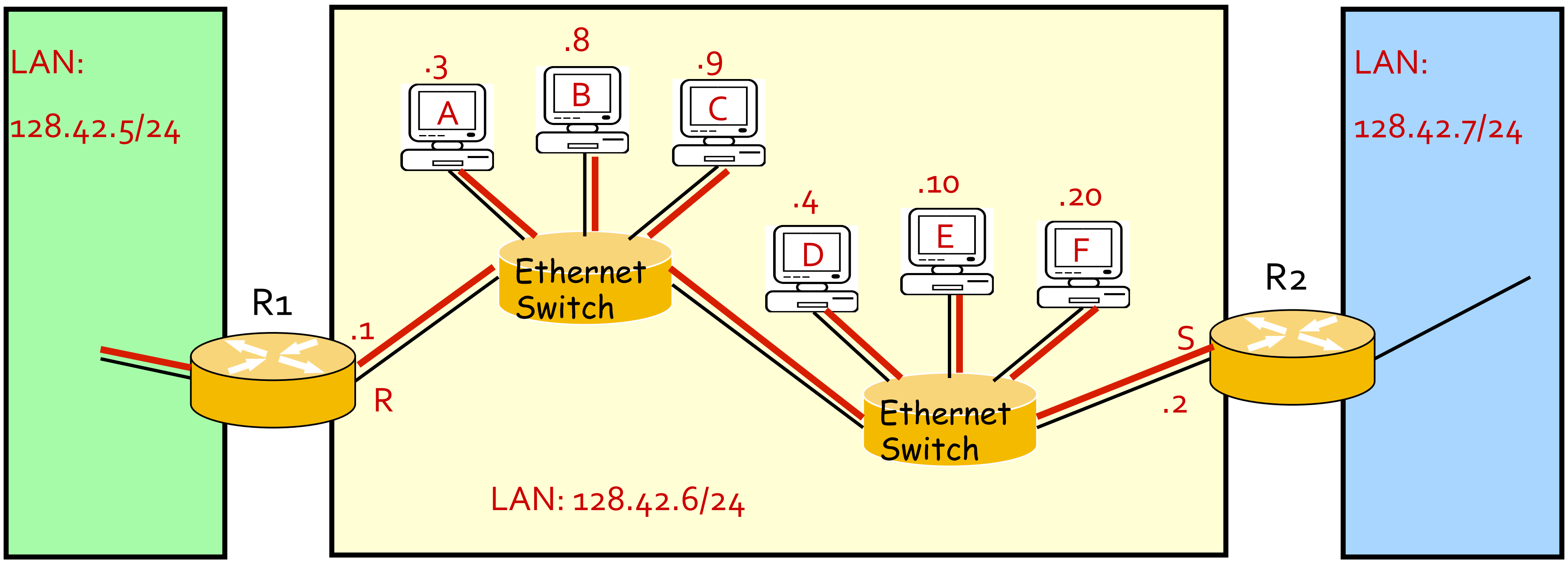
- Lookup destination IP address in forwarding table
  - Yields a next hop port and IP address
  - What if it doesn't?
- Lookup next hop IP address in ARP table
  - Yields a next hop MAC address
  - What if it doesn't?
- Forward modified packet out the next hop port with the next hop MAC address

Forwarding Packets



[illegible] arrives at R1
Result: [illegible]: 128.42.6.9, port: 128.42.6.1

Forwarding Packets



... arrives at R1

Result: next hop: 128.42.6.2, port: 128.42.6.1