Week 9 Lecture 2

Link Layer, LANs

Recall from last lecture we talked about multiple access protocols. We looked at three different protocols: channel partitioning, random access, and taking turns. However, ethernet network setup is not setup using CSMA/CD, we use switches instead. But we do use CSMA/CA (collision avoidance) in 802.11 Wi-Fi.

# Switched LANs

MAC addresses and ARP

* 32-bit IP address: network layer addresses for interface. It is used for layer 3 (network layer) forwarding.
* 48-bit MAC address (for most LANs) burned in NICROM, also sometimes software settable. For example: 71-65-F7-2B-08-53. Each char represents a hexadecimal number, which is 4 bytes. 4\*12 = 48

*Note: MAC address doesn’t have strict meaning and it is completely portable.*

The hardware company gets MAC address from IEEE. This lecture we are looking at how ARP protocol works.

**Question: how to determine interface’s MAC address knowing its IP address?**

## ARP: address resolution protocol

Table

Description automatically generated

* MAC addresses are hard coded in read-only memory when adapter is built. IP addresses are configured or learned dynamically.
* MAC addresses have flat name space of 48 bits, IP addresses are hierarchical name spaces of 32 bits. The bits have special meanings.
* MAC addresses are portable, they can stay the same as the host moves, IP addresses are not portable, they depend on where the host is attached.

ARP table: Each IP node (host, router) on LAN has table of

* IP/MAC address mappings for some LAN nodes: < IP address; MAC address; TTL>
* TTL: time after which address mapping will be forgotten (typically 20 min)

**Addressing: routing in same LAN**

* All “F”s FF-FF-FF-FF-FF-FF broadcasted in the network, all nodes on LAN receive the ARP query

A want to send datagram to B, if B is on A’s ARP table, then A can know its MAC address from IP address. If B is not on A’s ARP table, then A will broadcast ARP query packet, containing B’s IP address, when B receives ARP packet, it replies to A with its (B's) MAC address. Then A will have B’s MAC address, and it stores B in its ARP table. B will also store A in its ARP table.

**Addressing: routing to another LAN**

send datagram from A to B via R: focus on addressing – at IP (datagram) and MAC layer (frame)

1. assume A knows B’s IP address (how? Through DNS query, A knows B’s IP address, then the question is how A knows that B is on a different network?)

* A knows B is not local through subnet mask to B’s IP address, discover B via DHCP. A takes B’s IP address and apply subnet mask, the resultant is the network part. If the network part is different from its own network part, then it knows it is not in the same network. Now A needs to take this datagram, put it into the ethernet frame and send it to the MAC address of the interface first hop router.

1. assume A knows IP address of first hop router, R (how?)

* default router (discovered via DHCP)

1. assume A knows R’s MAC address (how? Using ARP)

* A sends out the ARP query for the router, only the router will respond back, and A knows the MAC address of the particular router.

1. A now knows the Destination IP address, it creates link layer frame with R’s MAC address as destination, frame contains A-to-B IP datagram.

Diagram, schematic

Description automatically generated

1. Frame is sent from A to R, frame received by R, datagram removed, passed up to IP layer

Diagram, schematic

Description automatically generated

1. R forwards datagram with IP source A, destination B (forwarding table)
2. R creates link-layer frame with B's MAC address as destination, frame contains A-to-B IP datagram

The Source & destination MAC addresses will typically change from hop to hop, but the source & destination IP addresses will typically remain the same

## Ethernet

**Bus vs Star**

* bus: popular through mid-90s, all nodes in same collision domain (can collide with each other), CSMA/CD for media access control
* star: prevails today, active switch in centre, each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other), No sharing, no CSMA/CD

Ethernet: unreliable, connectionless

* connectionless: no handshaking between sending and receiving NICs
* unreliable: receiving NIC does not send acks or nacks to send NIC
  + data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost

## Switches

* **Definition:** A switch is a link layer device that stores and forwards ethernet frames. It examines incoming a frame’s MAC address, and selectively forwards it to one or more outgoing links.
* Switches are *transparent* in the sense that hosts are unaware of their presence. Switches also don’t need to be configured.
* Hosts have a dedicated, direct connection to a switch. Switches buffer packets and runs at full duplex.
* Switches are *self-learning* in the sense that they learn which hosts can be reached through which interfaces.

**Diagram

Description automatically generatedSwitch Forwarding Table:** how does switch know A’ reachable via interface 4, B’ reachable via interface 5?

We start from an empty table, if A wants to know A’s MAC address, if will first check the switch, if it is on the table, then it can just send the frame. If the switch table empty, then it floods to connected interfaces. Then all other links receives the frame, A’ responds back. Destination A location knows it receives something back from A’, via interface 4. The response goes back via interface 1

Table

Description automatically generated

## Internal network switches

Diagram

Description automatically generated

When we have multiple switches, it works the same as in single-switch case! S1 floods the frame to all the links connected to S1, because S4 doesn’t know G, so S4 will flood the frame to S2 and S3. S2 and S3 don’t know anything so they just flood all interfaces. Then S3 will get response from G, So S4 knows S3 has G, then S1 knows from S4 about where G is, then A is able to send packets to G.

## Switches vs Routers

* both are store-and-forward
* both have forwarding tables

**Differences**:

* Different layers, switch only has two layers, router has 3 layers including the network layer
* routers: compute tables using routing algorithms, IP addresses
* switches: learn forwarding table using flooding, learning, MAC addresses

Diagram

Description automatically generated

Text

Description automatically generated

# Wireless Network

two important (but different) challenges

* wireless: communication over wireless link
* mobility: handling the mobile user who changes point of attachment to network

**Infrastructure mode**

* base station connects mobiles into wired network
* handoff: mobile changes base station providing connection into wired network

**Wireless Links**

**Wi-Fi**

There are multiple types of Wi-Fi protocols including 802.11b, 802.11a, 802.11g and 802.11n. They all have minor differences including their spectrum/range and bit rate.

important differences from wired link

* Decreased signal strength: radio signal attenuates as it propagates through matter (path loss)
* Interference from other sources: standardized wireless network frequencies (e.g., 2.4 GHz) shared by other devices (e.g., phone); devices (motors) interfere as well
* Multipath propagation (i.e. signals bouncing off objects): radio signal reflects off objects ground, arriving at destination at slightly different times

Diagram

Description automatically generatedThe *free space path loss* is calculated as

* *d* is the distance
* *λ* is the wavelength
* *f* is the frequency
* *c* is the speed of light

Chart, line chart

Description automatically generated**SNR: signal-to-noise ratio**

larger SNR – easier to extract signal from noise (a “good thing”). SNR versus BER tradeoffs

* given physical layer: increase power -> increase SNR->decrease BER
* given SNR: choose physical layer that meets BER requirement, giving highest throughput

SNR may change with mobility: dynamically adapt physical layer (modulation technique, rate)

Diagram

Description automatically generated

# A day in the life

1. The operating system on Bob’s laptop creates a DHCP request message (Section 4.3.3) and puts this message within a UDP segment (Section 3.3) with destination port 67 (DHCP server) and source port 68 (DHCP client). The UDP segment is then placed within an IP datagram (Section 4.3.1) with a broadcast IP destination address (255.255.255.255) and a source IP address of 0.0.0.0 since Bob’s laptop doesn’t yet have an IP address.

2. The IP datagram containing the DHCP request message is then placed within an Ethernet frame (Section 6.4.2). The Ethernet frame has a destination MAC address of FF: FF: FF: FF: FF: FF so that the frame will be broadcast to all devices connected to the switch (hopefully including a DHCP server); the frame’s source MAC address is that of Bob’s laptop, 00: 16: D3: 23: 68: 8A.

3. The broadcast Ethernet frame containing the DHCP request is the first frame sent by Bob’s laptop to the Ethernet switch. The switch broadcasts the incoming frame on all outgoing ports, including the port connected to the router.

4. The router receives the broadcast Ethernet frame containing the DHCP request on its interface with MAC address 00: 22: 6B: 45: 1F: 1B and the IP datagram is extracted from the Ethernet frame. The datagram’s broadcast IP destination address indicates that this IP datagram should be processed by upper layer protocols at this node, so the datagram’s payload (a UDP segment) is thus demultiplexed (Section 3.2) up to UDP, and the DHCP request message is extracted from the UDP segment. The DHCP server now has the DHCP request message.

A picture containing shape

Description automatically generated

Step 5: ACK + IP address + IP address of the DNS server, the IP address for the gateway router, and the subnet block

Step 4

Step 3

Step 1

Step 2

5. Let’s suppose that the DHCP server running within the router can allocate IP addresses in the CIDR (Section 4.3.3) block 68.85.2.0/24. In this example, all IP addresses used within the school are thus within Comcast’s address block. Let’s suppose the DHCP server allocates address 68.85.2.101 to Bob’s laptop. The DHCP server creates a DHCP ACK message (Section 4.3.3) containing this IP address, as well as the IP address of the DNS server (68.87.71.226), the IP address for the default gateway router (68.85.2.1), and the subnet block (68.85.2.0/24) (equivalently, the “network mask”). The DHCP message is put inside a UDP segment, which is put inside an IP datagram, which is put inside an Ethernet frame. The Ethernet frame has a source MAC address of the router’s interface to the home network (00: 22: 6B: 45: 1F: 1B) and a destination MAC address of Bob’s laptop (00: 16: D3: 23: 68: 8A).

6. The Ethernet frame containing the DHCP ACK is sent (unicast) by the router to the switch. Because the switch is self-learning (Section 6.4.3) and previously received an Ethernet frame (containing the DHCP request) from Bob’s laptop, the switch knows to forward a frame addressed to 00: 16: D3: 23: 68: 8A only to the output port leading to Bob’s laptop

7. Bob’s laptop receives the Ethernet frame containing the DHCP ACK, extracts the IP datagram from the Ethernet frame, extracts the UDP segment from the IP datagram, and extracts the DHCP ACK message from the UDP segment. Bob’s DHCP client then records its IP address and the IP address of its DNS server. It also installs the address of the default gateway into its IP forwarding table (Section 4.1). Bob’s laptop will send all datagrams with destination address outside of its subnet 68.85.2.0/24 to the default gateway. At this point, Bob’s laptop has initialized its networking components and is ready to begin processing the Web page fetch. (Note that only the last two DHCP steps of the four presented in Chapter 4 are actually necessary.)

Diagram

Description automatically generated

**Stage 2: sending data and receiving data from outside network**

* before sending HTTP request, need IP address of www.google.com: DNS.
* DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth.
* To send frame to DNS server, need MAC address of first hop router: ARP
* If we are using internal DNS server, then the ARP query will be sent directly to the DNS server

8. The operating system on Bob’s laptop thus creates a DNS query message (Section 2.5.3), putting the string “www.google.com” in the question section of the DNS message. This DNS message is then placed within a UDP segment with a destination port of 53 (DNS server). The UDP segment is then placed within an IP datagram with an IP destination address of 68.87.71.226 (the address of the DNS server returned in the DHCP ACK in step 5) and a source IP address of 68.85.2.101.

9. Bob’s laptop then places the datagram containing the DNS query message in an Ethernet frame. This frame will be sent (addressed, at the link layer) to the gateway router in Bob’s school’s network. However, even though Bob’s laptop knows the IP address of the school’s gateway router (68.85.2.1) via the DHCP ACK message in step 5 above, it doesn’t know the gateway router’s MAC address. In order to obtain the MAC address of the gateway router, Bob’s laptop will need to use the ARP protocol (Section 6.4.1).

10. Bob’s laptop creates an ARP query message with a target IP address of 68.85.2.1 (the default gateway), places the ARP message within an Ethernet frame with a broadcast destination address (FF: FF: FF: FF: FF: FF) and sends the Ethernet frame to the switch, which delivers the frame to all connected devices, including the gateway router.

11. The gateway router receives the frame containing the ARP query message on the interface to the school network and finds that the target IP address of 68.85.2.1 in the ARP message matches the IP address of its interface. The gateway router thus prepares an ARP reply, indicating that its MAC address of 00: 22: 6B: 45: 1F: 1B corresponds to IP address 68.85.2.1. It places the ARP reply message in an Ethernet frame, with a destination address of 00: 16: D3: 23: 68: 8A (Bob’s laptop) and sends the frame to the switch, which delivers the frame to Bob’s laptop.

12. Bob’s laptop receives the frame containing the ARP reply message and extracts the MAC address of the gateway router (00: 22: 6B: 45: 1F: 1B) from the ARP reply message.

13. Bob’s laptop can now (finally!) address the Ethernet frame containing the DNS query to the gateway router’s MAC address. Note that the IP datagram in this frame has an IP destination address of 68.87.71.226 (the DNS server), while the frame has a destination address of 00: 22: 6B: 45: 1F: 1B (the gateway router). Bob’s laptop sends this frame to the switch, which delivers the frame to the gateway router.

**Stage 3 using DNS**

The gateway router receives the frame and extracts the IP datagram containing the DNS query. The router looks up the destination address of this datagram (68.87.71.226) and determines from its forwarding table that the datagram should be sent to the leftmost router in the Comcast network in Figure 6.32. The IP datagram is placed inside a link-layer frame appropriate for the link connecting the school’s router to the leftmost Comcast router and the frame is sent over this link.

15. The leftmost router in the Comcast network receives the frame, extracts the IP datagram, examines the datagram’s destination address (68.87.71.226) and determines the outgoing interface on which to forward the datagram toward the DNS server from its forwarding table, which has been filled in by Comcast’s intra-domain protocol (such as RIP, OSPF or IS-IS, Section 5.3) as well as the Internet’s inter-domain protocol, BGP (Section 5.4). IP datagram forwarded from first hop router in campus network into comcast network, routed (tables created by RIP, OSPF, IS-IS and/or BGP routing protocols) to DNS server

16. Eventually the IP datagram containing the DNS query arrives at the DNS server. The DNS server extracts the DNS query message, looks up the name www.google.com in its DNS database (Section 2.5), and finds the DNS resource record that contains the IP address (64.233.169.105) for www.google.com. (Assuming that it is currently cached in the DNS server). Recall that this cached data originated in the authoritative DNS server (Section 2.5.2) for google.com. The DNS server forms a DNS reply message containing this hostname-to-IP address mapping and places the DNS reply message in a UDP segment, and the segment within an IP datagram addressed to Bob’s laptop (68.85.2.101). This datagram will be forwarded back through the Comcast network to the school’s router and from there, via the Ethernet switch to Bob’s laptop. DNS server replies to client with IP address of www.google.com

17. Bob’s laptop extracts the IP address of the server www.google.com from the DNS message. Finally, after a lot of work, Bob’s laptop is now ready to contact the www.google.com server!

**Stage 4 HTTP and TCP**

18. Now that Bob’s laptop has the IP address of www.google.com, it can create the TCP socket (Section 2.7) that will be used to send the HTTP GET message (Section 2.2.3) to www.google.com. When Bob creates the TCP socket, the TCP in Bob’s laptop must first perform a three-way handshake (Section 3.5.6) with the TCP in www.google.com. Bob’s laptop thus first creates a TCP SYN segment with destination port 80 (for HTTP), places the TCP segment inside an IP datagram with a destination IP address of 64.233.169.105 (www.google.com), places the datagram inside a frame with a destination MAC address of 00: 22: 6B: 45: 1F: 1B (the gateway router) and sends the frame to the switch.

19. The routers in the school network, Comcast’s network, and Google’s network forward the datagram containing the TCP SYN toward www.google.com, using the forwarding table in each router, as in steps 14–16 above. Recall that the router forwarding table entries governing forwarding of packets over the inter-domain link between the Comcast and Google networks are determined by the BGP protocol (Chapter 5).

20. Eventually, the datagram containing the TCP SYN arrives at www.google.com. The TCP SYN message is extracted from the datagram and demultiplexed to the welcome socket associated with port 80. A connection socket (Section 2.7) is created for the TCP connection between the Google HTTP server and Bob’s laptop. A TCP SYNACK (Section 3.5.6) segment is generated, placed inside a datagram addressed to Bob’s laptop, and finally placed inside a link-layer frame appropriate for the link connecting www.google.com to its first-hop router.

21. The datagram containing the TCP SYNACK segment is forwarded through the Google, Comcast, and school networks, eventually arriving at the Ethernet card in Bob’s laptop. The datagram is demultiplexed within the operating system to the TCP socket created in step 18, which enters the connected state.

22. With the socket on Bob’s laptop now (finally!) ready to send bytes to www.google.com, Bob’s browser creates the HTTP GET message (Section 2.2.3) containing the URL to be fetched. The HTTP GET message is then written into the socket, with the GET message becoming the payload of a TCP segment. The TCP segment is placed in a datagram and sent and delivered to www.google.com as in steps 18–20 above.

23. The HTTP server at www.google.com reads the HTTP GET message from the TCP socket, creates an HTTP response message (Section 2.2), places the requested Web page content in the body of the HTTP response message, and sends the message into the TCP socket.

24. The datagram containing the HTTP reply message is forwarded through the Google, Comcast, and school networks, and arrives at Bob’s laptop. Bob’s Web browser program reads the HTTP response from the socket, extracts the html for the Web page from the body of the HTTP response, and finally (finally!) displays the Web page!