eecs 489 COMPUTER NETWORKS

Lecture 21:
Physical and Link Layers

Internet Protocol Stack

application: supporting network applications
- HTTP, SMTP, FTP, etc.

transport: endhost-endhost data transfer
- TCP, UDP

network: routing of datagrams from source to destination
- IP, routing protocols

link: data transfer between neighboring network elements
- Ethernet, WiFi

physical: bits "on the wire"

Physical Layer: Signals

We only look at a very brief overview of the physical layer in this course

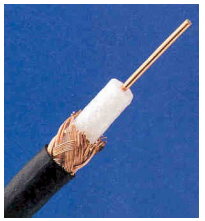
- to learn more, take EECS 455: Signals and Systems, EECS 554: Digital Communication and Coding, and/or EECS 557: Communication Networks

Physical Layer: Signals

Signal degrades (attenuates) as it travels further from the source (caused by resistance on the wire, cosmic interference, etc.)

How different types of PHY tech alleviates signal attenuation:

- coax: shielding of core reduces interference
- twisted pair: twisting a pair of wires changes the electrical property of the pair, reducing interference
- glass fiber:
  - LED or laser as signal source
  - more fragile but no interference
  - can carry more data
  - hard to splice



Shielded twisted pair (STP)

Unshielded twisted pair (UTP)

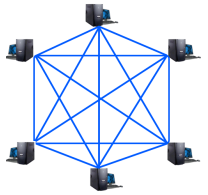
Transmission Distance Limitation

Transmission distance limited due to: signal loss caused by interference and sharing condition

Examples:

- serial line (RS-232): 15 m
- twisted pair:
  - Cat 5(e): 1-10Base-T (2 wires), 1GBase-T (4 wires): 100 m
  - Cat 6: ≤ 1GBase-T: 100 m, 10GBase-T: 37-55 m
  - Cat 6a: 10GBase-T: 100 m
- fiber: 2-100 km

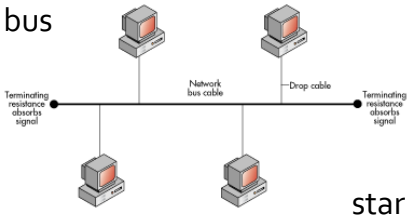
Wiring Scheme



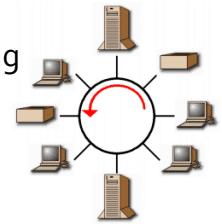
Point-to-point: $O(N^2)$ connections to connect N computers

Shared LAN:

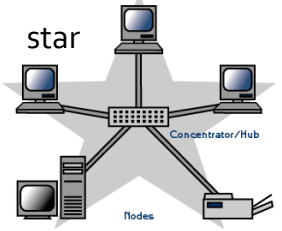
bus



ring



star



Wireless

Radio:

- satellite: order of Gbps, up-down latency of 250 ms (too long)
- cellular, WiFi, WiMax
- Bluetooth: 2.4 GHz short range radio, 721 Kbps – 2.1 Mbps, 1-100 m
- UWB: 3.1 GHz-10.6 GHz, 480-675 Mbps, 10 m, less interference due to use of short pulses
  - Wireless USB
  - Bluetooth 3.0
  - Wireless FireWire

Wireless

Microwave:

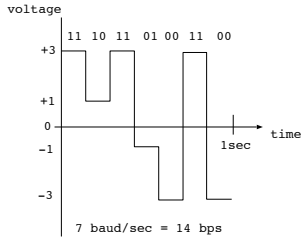
- high bandwidth: 1.5 Gbps
- can be aimed in a single direction
- requires "line-of-sight"
- most useful to connect buildings on campus

Infrared: shorter distance, no need for antenna

Bit vs. Baud Rate

Bit vs. Baud

- bits transmitted as electrical or optical signal
- bit rate: number of bits per second
- baud rate: signal/voltage level changes per second
- each level can represent multiple bits
  - for binary signaling, bit rate == baud rate
  - for M-ary (M levels) signaling, bit rate $\neq$ baud rate
  - example: 4-ary signaling carries 2 bits per level



Signals

Example: RS-232

- negative voltage (-15V) represents a 1
- positive voltage ($+15$V) represents a 0
- bit rate == baud rate
- 7 bits/character
- to allow asynchronous communication: 1 start bit, 1 stop bit

Signal Digitization

The maximum rate at which you can transmit data is limited by how fast (in Hertz) the sender's hardware can change voltage level and how sensitive the receiver's hardware is to voltage level changes

Nyquist Sampling Theorem (1924):

> For a signal band limited in frequency at B Hz, we need to sample at $2B$ Hz to reconstruct the original signal from the samples

Signal Digitization

Conversely, and more generally, for an M-level signal, the maximum data rate (R) is determined by:

$$R = 2B \log_2 M \text{ bps},$$

where B is the line bandwidth (in Hz)

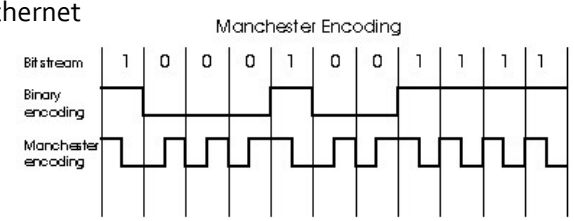
Example: RS-232, $M = 2$, phone line: $B = 3$ kHz

So, signal travelling over phone line using RS-232 signaling has a maximum data rate of $R = 2B = 6$ Kbps (modems don't use RS-232 signaling!)

Manchester Encoding

Problem with long strings of 0s or 1s
- no transition from low-to-high, or high-to-low
- receiver keeps average of signal it has received and uses the average to distinguish between high and low
- long flat strings make receiver sensitive to small changes

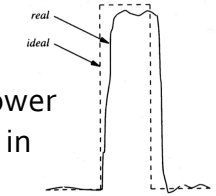
With Manchester encoding, each bit contains a transition
- allows sender and receiver to synchronize clocks with each other
  - no need for a centralized, global clock!
- used in 10BaseT Ethernet



Signal to Noise Ratio (dB)

Nyquist Sampling Theorem assumes noiseless channel

In reality, channels are noisy



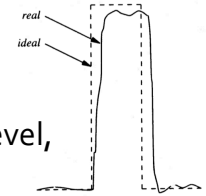
S/N: ratio of signal power (watts) to noise power (watts) usually given as signal-to-noise ratio in quantity of $10 \log_{10} S/N$, called dB (decibles)

Examples:
- $S/N = 10$, signal-to-noise ratio is 10 dB
- $S/N = 100$, signal-to-noise ratio is 20 dB

Signal to Noise Ratio (dB)

Shannon Capacity (1948): the maximum data rate (C) of a noisy channel with bandwidth B Hz and a given signal-to-noise ratio is:

$$C = B \log_2 (1+S/N) \text{ bps},$$



which gives M (levels of signal, or bits per level, required) on the order of $\sqrt{(1+S/N)}$

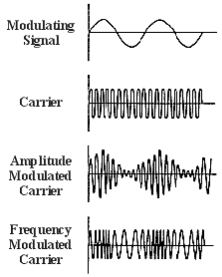
Phone lines have $B = 3$ kHz, $S/N = 30$ db, so $C = 29.9$ kbps $(3 \cdot \log_2(1+1000))$

Carrier Wave

Observation: a continuous, oscillating signal propagates further (with less signal loss) than other signals

Hence to send data long distances, we use a continuous sine wave as a carrier wave

Data is "carried" by modifying the carrier wave, a process called modulation



Two types of modulation:
1. Amplitude Modulation (AM): not as robust
2. Frequency Modulation (FM): more robust

Modem: modulator-demodulator

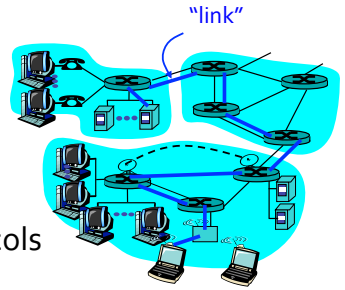
Transmission Bandwidths

http://en.wikipedia.org/wiki/List_of_device_bandwidths

Data Link Layer

The data-link layer has the responsibility of transferring packets from one node to an adjacent node over a link

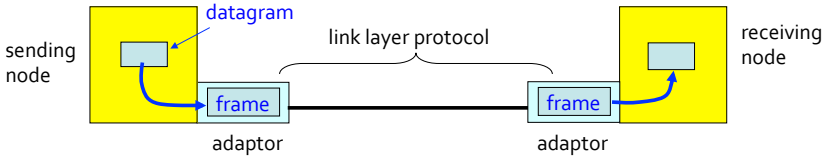
At the link layer, a packet is called a frame, and it encapsulates a network-layer datagram



A network datagram may be transferred by different link protocols over different links:

- e.g., Ethernet on the first link, frame relay on intermediate links, and 802.11 on the last link

Adaptors Communicating



Link layer implemented in "adaptor" (a.k.a. NIC)

- Ethernet card, USB card, 802.11 card

Sending side:

- encapsulates datagram in a frame
- adds error checking bits, flow control, etc.

Receiving side

- looks for errors, flow control, etc.
- extracts datagram, passes to receiving node

Adaptor is semi-autonomous link & physical layers

Ethernet

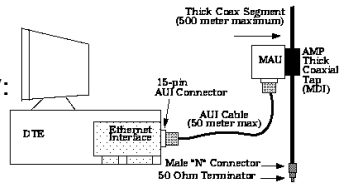
"Dominant" wired LAN technology:

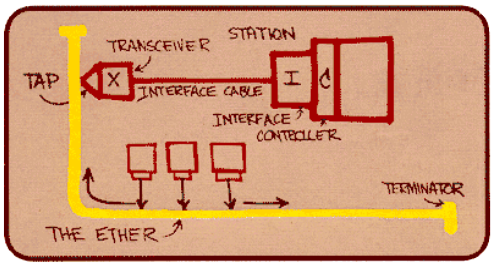
Cheap: $20 for 100Mbps!

First widely used LAN technology

Simpler, cheaper than token LANs and ATM

Kept up with speed race: 10 Mbps – 10 Gbps





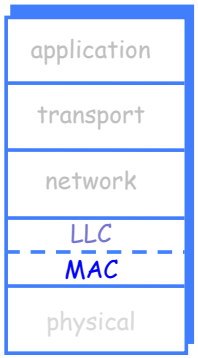
Metcalfe's Ethernet sketch

Data Link Layer

The data link layer can be further subdivided into:

1. Logical Link Control (LLC): error and flow control
2. Media Access Control (MAC): framing and media access

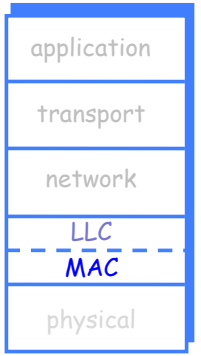
different link protocols may provide different services, e.g., Ethernet doesn't provide reliable delivery (error recovery)



Data Link Layer

MAC topics:

- framing and MAC address assignment
- LAN forwarding
- IP to MAC address resolution
  - IP to MAC: Address Resolution Protocol (ARP)
  - MAC to IP: Dynamic Host Configuration Protocol (DHCP)
- media access control



Framing

Why packetize/frame data?

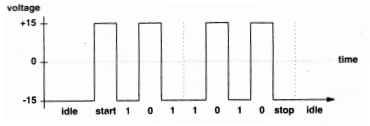
•

•

Framing allows sources with small amount of data (e.g., VoIP) to finish promptly

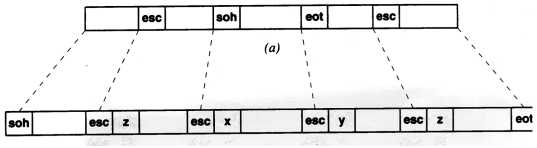
Framing

Framing is done by using a special bit pattern to denote start & end of frame (`soh` & `eot`)



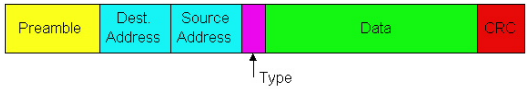
Bit stuffing: if `soh` & `eot` shows up in data, they must be protected/escaped

Character In Data	Characters Sent
soh	esc x
eot	esc y
esc	esc z



Ethernet Frame Structure

Sending adaptor encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame



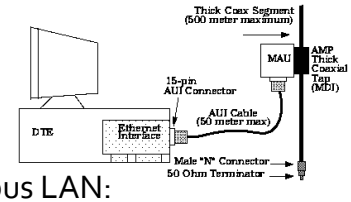
Preamble: 7 bytes of pattern `10101010` followed by one byte of pattern `10101011`, used to synchronize receiver-sender clock rates

Addresses: 6 bytes each

Type: indicates the higher layer protocol, e.g., IP, IPX, AppleTalk

CRC (cyclic redundancy check): checked at receiver, if error is detected, the frame is simply dropped

Frame Transmission



Frame transmission on a shared bus LAN:

- frames are tagged with destination MAC address
- frames sent to all hosts on the LAN
- the NIC on each host makes a copy of frame
- if the frame is addressed to the host, or a broadcast frame (e.g., ARP packet) the NIC sends the frame up to the CPU, otherwise discards frame
- a frame can also have a broadcast or multicast address
- NICs could be put in promiscuous mode (e.g., `tcpdump`, `ethereal`, network sniffer, network analyser)

Why Not Just Use IP Addresses?

LANs are designed for arbitrary network protocols

- not just for IP (e.g., IPX, Appletalk, X.25, ...)
  - though IP is now the main game in town
- different LANs may have different addressing schemes
  - though Ethernet is now the main game in town
  - Ethernet addresses: `00-15-C5-49-04-A9`
    - blocks: assigned to vendors by the IEEE
    - adapters: assigned by the vendor from its block

MAC address assignment

- static: Ethernet (48-bits): requires global address assignment
- configurable: requires DIP switch, EPROM
- dynamic (random number):
  - advantage: only need to be unique within a LAN
  - disadvantage: address changes between reboots

Address Resolution

IP routing on a LAN: assume hosts know their own network number and subnet mask:

- send directly to the destination if on the same LAN
- send to a default router otherwise

host must know the MAC address of either the destination or the default router

Address Resolution

Given a node's IP address, how does a host know its MAC address?

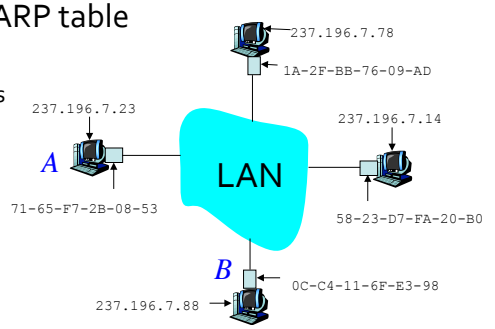
- MAC address can be inferred from the IP address (IPv6)
- from a statically configured table
- ask a server
- use the Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP)

How would host *A* discover host *B*'s MAC address, assuming it knows *B*'s IP address?

Each IP node (host, router) on the LAN maintains an ARP table

- ARP table: IP-to-MAC address mappings for some LAN nodes
  - < IP address; MAC address; ttl>
  - ttl (time to live): time after which, address mapping will be flushed (typically 20 min)
  - maintained in an LRU manner



ARP Protocol: Same LAN

A wants to send datagram to *B*, but *B*'s MAC address not in *A*'s ARP table

A broadcasts ARP query packet, containing *B*'s IP address
- destination is broadcast MAC address `FF-FF-FF-FF-FF-FF`
- all machines on LAN receive ARP query
- query packet also contains *A*'s own IP and MAC addresses

B replies to *A* with *B*'s IP and MAC addresses
- frame sent to *A*'s MAC address (unicast)
- *B* caches (saves) *A*'s IP-to-MAC address mapping in its own ARP table, or refreshes *A*'s entry if it already exists

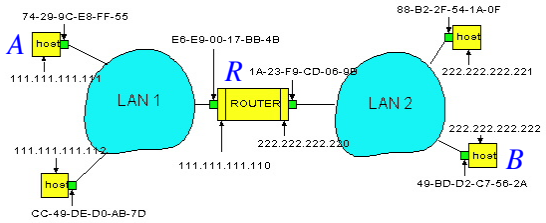
ARP Protocol: Same LAN

A caches *B*'s IP-to-MAC address pair in its ARP table until ttl expires, at which time it will be flushed
- soft state: information that times out (goes away) unless refreshed

ARP is "plug-and-play":
- nodes create their ARP tables without human intervention
- try out `arp(8)` (may need root/administrator permission)

Forwarding to Another LAN

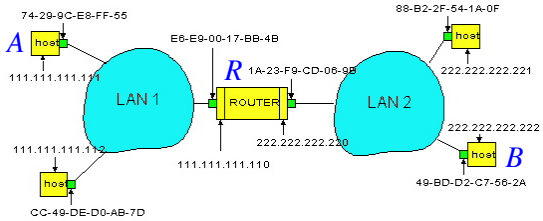
To send datagram from A to B via R, assuming A knows B's IP address (e.g., via DNS)



- router R has two ARP tables: one for each LAN
- A knows that its default router (R) has IP address `111.111.111.110`
- A looks up R's MAC address `E6-E9-00-17-BB-4B` from its ARP table, or if the mapping doesn't exist, it sends out an ARP request packet to resolve it

Forwarding to Another LAN

- A creates datagram with source IP A, destination IP B
- A creates link-layer frame with R's MAC address as destination, frame containing A-to-B IP datagram
- A's sends frame to R



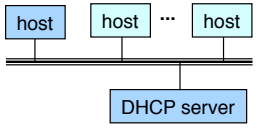
- R receives frame, extracts IP datagram from frame, sees that its destination is B
- R uses ARP to get B's MAC address, and creates a new frame containing A-to-B IP datagram with MAC destination address set to B's

Obtaining an IP Address

How does a host obtain its IP address?

1. static: hard-coded by system administrator in a file
   - Windows: Control Panel→Network→Configuration→TCP/IP→Properties
   - UNIX: /etc/rc.config
2. dynamic: ask a server:
   - Reverse ARP (RARP) (obsolete)
   - BOOT Protocol (BOOTP) (obsolete)
   - Dynamic Host Configuration Protocol (DHCP): dynamically request an address from a server when the host boots
     - "plug-and-play"

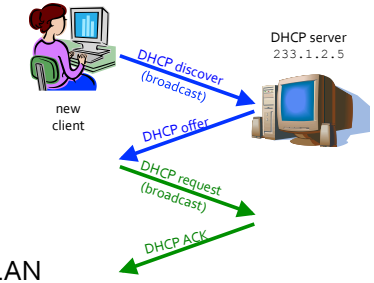
DHCP



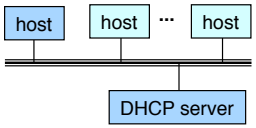
Dynamic Host Configuration Protocol

Client host:

- broadcasts a DHCP discover packet with its own MAC address
  - uses UDP/IP with IP broadcasting
    - limiting DHCP use within a LAN, but beyond a physical segment
- broadcast preceded by a random wait time, to prevent storming the LAN

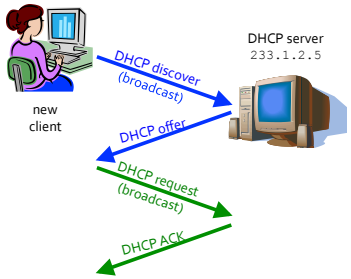


DHCP



Servers:

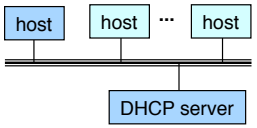
- maintain a pool of shared host identities
- if MAC address of a querying host is not in the database of permanent identities, assigns (leases) it a temporary identity from pool
- one or more DHCP servers respond with IP address offer



Client host:

- chooses one offer and requests it from the offering server
- if no reply, server may be down or busy, retry later

DHCP



To prevent too many replies:

- each host can be assigned a primary server
- on repeated query, non-primary servers wait a random time for response from other servers before replying

Dynamic IP addresses with DHCP:

- advantage: doesn't require manual configuration
- shortcoming: DHCP's interaction with DNS unspecified (dynamic DNS not yet/ever? widely deployed)

Other Information

Other information a newly booted machine may need:

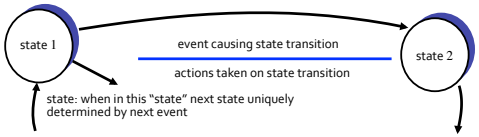
- subnet mask
- default router's address
- DNS server
- time server
- print server
- file server
- boot file (name and size, if thin/diskless client/netbook), etc.

Queries and replies for all of these may be batched together for efficiency

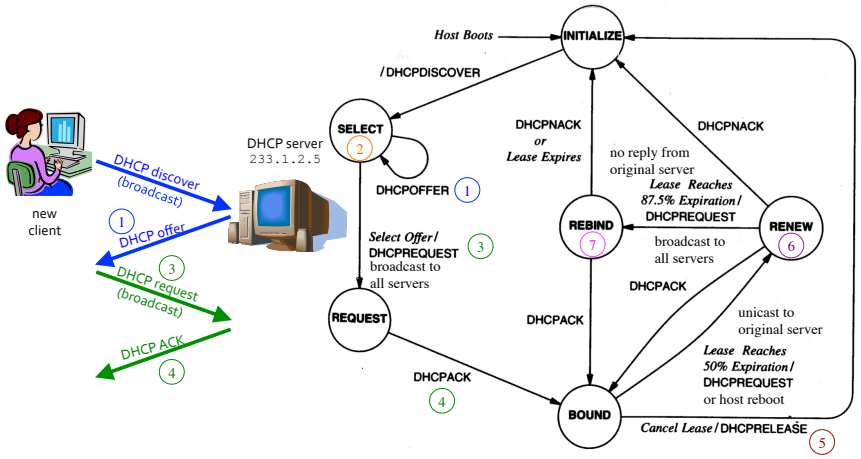
Finite State Machine

Finite state machine (FSM) is a useful tool for designing and documenting protocol:

- consists of a number of states
- is a graph showing the transition from one state to one or more states
- labels on the edges show:
  - what event causes each transition, e.g., receiving a certain type of packet
  - and what actions or side effects each transition may cause, if any



DHCP Simplified Finite State Machine



DHCP

① `DHCPOFFER` message from the server
- configuration parameters (proposed IP address, mask, gateway router, DNS server, …)
- lease time (the time the information remains valid)

② `SELECT`: multiple servers may respond
- multiple servers on the same broadcast media
- each may respond with an offer
- the client can decide which offer to accept

③ `DHCPREQUEST`: accepting one of the offers
- client broadcasts a `DHCPREQUEST` echoing the parameters
- other servers see the acceptance and update their lease database

[after Rexford]

DHCP Leases

④ `DHCPACK`: server confirmation
- the DHCP server responds with a `DHCPACK` to confirm

⑤ `DHCPRELEASE`: why is a lease time necessary?
- client can release the IP address (`DHCPRELEASE`)
  - `ipconfig /release` at the CLI
  - clean shutdown of the computer
- or, the host might not release the address
  - the host crashes
  - buggy client software
- and you don't want the address to be allocated forever
- performance trade-offs
  - short lease time: returns inactive addresses quickly
  - long lease time: avoids overhead of frequent renewals

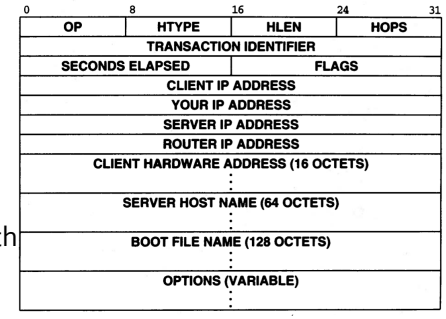
[after Rexford]

DHCP Leases

⑥ `RENEW`: lease reaches 50% expiration or upon reboot
- renew lease with original server
- allows client to cache IP address across boot
  - upon reboot, client tries to renew lease of cached address

⑦ `REBIND`: lease reaches $7/8^{th}$ expiration
- original server doesn't respond
- broadcast to all servers
- if no server can renew, lease a new address
- how to keep lease timers consistent across multiple servers is not part of the DHCP standard [RFC 2131]

DHCP Packet Format



Opcode:
- `DHCPREQUEST`
- `DHCPACK`

Hardware Type:
- Ethernet (1), FireWire (24), etc.

`hlen`: hardware address length

Transaction ID:
- a random number chosen by client to associate messages with responses

Seconds elapsed:
- seconds since client began an address acquisition or renewal process