ECE 435 – Network Engineering Lecture 17

Vince Weaver

http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

2 November 2017

Announcements

- Windstorm/Power Outage

- Don't forget project topics due
  Make sure to e-mail, even if you've discussed in person.

HW#7

- Go over HW#7

Example Link Protocols

- Obsolete/Fading
  - Token Ring
  - HIPPI
  - FDDI – fiber distributed data interface
  - Fibre Channel
  - ATM
  - ISDN
  - X.25
- Current

- Ethernet (802.3)
- PPP (? fading)
- WLAN (802.11)
- Bluetooth (802.15)
- FiberChannel (?)
- DSL
- LTE/WiMAX (802.16)

Ethernet History

- Proposed by Bob Metcalfe in 1973 (went on to found 3Com)
- Metcalfe, Boggs, Thacker, and Lampson listed on patent
- Inspired by ALOHAnet, a wireless network in Hawaii, allow users on various islands to connect to server on Oahu
- Various competing local networks, Ethernet won in the end

Token Ring

- Guaranteed Deterministic Delivery (vs Ethernet: best effort)
- Dates to 1970s
- Standardized by IBM, 1984, IEEE 802.5 (note, not RFCs anymore)
- 4Mbps, eventually shielded twisted pair, eventually 16Mbps, 100Mbps and 1Gbps
- 3-byte frame passed around gives permission to transmit
- More complex, no crossover cable (direct connect two

machines),

- Supports multiple identical MAC addresses
- Deterministic time to get to transmit
- Frames can have different access priorities
- Empty token passed around. If data to transmit, put in. Then passes around until it gets to receiver, removed, and back to passing empty token. When gets back to originator it knows it has been received.
- Token Bus, GM, IEEE802.4 (withdrawn) like ring, but virtual ring. Needed to know neighbors to pass token. Guaranteed worst case transmit time.

Why did Ethernet win?

- Other competitors: FDDI, ATM, DQDB.
- Why did it win? Simpler and thus cheaper.
- Why simpler? No priority mechanism, no QoS, no central control
- Could use cheaper twisted pair cable
- Token ring cards generally a lot more expensive than Ethernet

Ethernet Timeline

- Standardized in 1981
- Timeline:
  - 1972 – experimental 3Mbps
  - 1981 – DIX (DEC/Intel/Xerox) ver 1 (10Mbps)
  - 1982 – DIX ver 2
  - 1983 – IEEE 802.3/10BASE5 "Thick Ethernet", up to 500m, vampire taps, often yellow or orange (standard suggests yellow). Looks like garden hose. AUI connector, drill into cable, at 2.5m intervals (to

avoid reflections), terminated on each end. One bad connection could ruin for all

- 1985 – 10BASE2 – "thin net", thinner connections, BNC connectors, T connectors (185m, rounded up to 200m), 50 ohm terminator, grounding loops, one and only one must be tied to ground. How to detect network problem? Send pulse, look for echo
- 1990 – 10BASE-T – twisted pair (Cat3) , needed hub
- 1993 – 10BASE-F – fiber
- 1995 – 100BASE-T 100BASE-TX 4B5B MLT-3 cat5 two twisted pairs

- 1997 – Full-duplex
- 1998/1999 – 1000BASE-TX PAM-5, four twisted pairs, can transfer in both directions on one pair using DSP/echo cancelation
- 2006 – 10GBASE-T
- 2010 – 40G and 100G
- 2017 – 400GB

- Naming: Speed/BROAD,BASE,PASS/PHY
  - Almost all is baseband (narrow frequency, vs broadband).
  - PHY originally was distance could travel (in 100m)

but now medium type.

Ethernet Collisions

- In order to work properly, twice round-trip time needs to be less than time needed to transmit minimal (64-byte) frame, otherwise not possible to notice collision in time and frame loss
- This limits network size to collision domain
- Bits wasted is not bad, collision often caught in the preamble

Ethernet MAC

- CSMA/CD "Carrier sense multiple access with collision detection"
- First senses cable (how?)
- If busy, waits
- Sends. If collision, jams the cable aborts transmission, waits random back off time before retrying.
- Exponential backoff. Randomly choose time from 0 to

$2^k - 1$ where k is number of tries (capping at 10). Time slot is 512 bits for 10/100, 4096 for 1Gbs

- on newer full-duplex links no need for carrier sense and collision detection not needed

Manchester Encoding

- Does not use 0V for 0 and 5V for 1.
  Why? Idle is 0, so how can you tell how many zeros at beginning of signal?
- Could use +1V/-1V, but still would need way to sync signal on long runs of 0 or 1
- Manchester encoding
  - 1 is high to low transition.
  - 0 is low to high transition.
  - Always a transition in the middle of an interval.

- Disadvantage, need twice as much bandwidth
- Differential Manchester
  - transition at start of interval means 0
  - lack of transition means 1
  - Still transition in the middle
  - More complex but better noise handling
- Ethernet uses Manchester, Token Ring uses differential Manchester
- Ethernet high 0.85V and low -0.85V

```
Manchester
  1       0    0      1       1
__   !   ___!   ___!___   !___
  |  !  |   !  |   !   |  !   |
  ---!---   !---   !   ---!   ---
     !      !      !      !

Differential Manchester
         0    0     1      1
__   !__   !__   !   __!__
  |  !  |  !  |  !  |  !  |
  ---!  ---!  ---!---  !   ---
```

Ethernet frame layout

- Packet format:
  Preamble = 7 bytes
  SFD (start of frame delimit) = 1 byte
  DA (destination address) = 6 bytes
  SA (source address) = 6 bytes
  T/L (type or length) = 2 bytes
  Data = 46-1500 bytes
  Frame Check Sequence = 4 bytes

- Preamble is fixed 1010...1010 in transmission order (LSB least significant bit first)
  On original Ethernet this was 10MHz 6.4us pulse used to synch clocks The PHY might do other things (100BASE-X adds 4B/5B stuff)
- SFD - indicates the start of the frame with value 10101011 in transmission order.
- DA = 48 bit destination MAC address
  first bit unicast or multicast (note, LSB first, so this is 0x01 as first octet, not 0x80)
  with first bit high, multicast, sent to many in a group,

all 1s is broadcast, send to everyone
1-3 OUI (organization unique identifier)
Globally Unique

- SA = 48 bit source MAC address
- T/L: Originally type field. 802.3 makes it length of *data* field (not length of frame). Later in 1997 802.3 approved as type too, so dual meaning. How tell difference? Since cannot be longer than 1500, any value bigger than 0x600 (1536) is type. 0x0800 = IPv4, 0x86dd = IPv6, 0x0806 = ARP

How tell length if type? Detect end, or checksum
How tell type if length? Will have 802.2 header

- Data – data from 46 to 1500 bytes
  Why limit 1500B? because RAM was expensive in 1978.
  If smaller than 46 bytes padded. Makes sure checksum works. Also if too short, could be done transmitting before a collision can be detected (light travel to furthest node and back)
- FCS – a 32-bit CRC code. Somwhat complicated, magic number at end 0xc704dd7b If incorrect FCS, silently drops. How can we do this? Up to upper protocol (say

TCP/IP) to figure out if need to resend. Makes things simple. No need to wait for ACKs.

- Frame size is variable. Often first two fields are excluded and said that Ethernet packets are between 64 and 1518 bytes long

ARP – address resolution protocol

- On local network, how do we find MAC address if we know IP?
  Hard-code in /etc/ethers?
  Request somehow?

- ARP (RFC826)
  - Device first checks ARP cache to see if already knows
  - Otherwise, broadcasts to ff:ff:ff:ff:ff:ff "who has this IP"

- ◦ Device reply with its IP and MAC (unicast)
- ◦ These are cached
- ◦ Timeout in case you reassign
- ◦ ARP announcement: can broadcast when your address changes so they can update (gratuitous ARP)
- ◦ Other optimizations(?)

- Used for many higher protocols, but not IPv6 which uses NDP (Neighbor Discovery Protocol)
- Security: ARP spoofing

RARP/BOOTP

- Some cases need to do RARP (Reverse ARP) 9RFC 903) have own MAC, find IP (netbooting is common reason)

- ARP packets nor forwarded, so extension called BOOTP that allowed network booting.

- BOOTP automated by DHCP.