CMSC417 Spring 2016 Lecture #18 4/13/2016

<u>Agenda</u>

⇒ p4 assigned, due 4/22 (in 9 days)

⇒ p3 grades up

⇒ Ethernet MAC

⇒ Error Detection

 ☐ checksums

⇒ missed why bother detecting collisions

⇒ missed why invert internet checksum

⇒ missed why we need a jamming signal

CMSC417 Spring 2016 Lecture #18 4/13/2016

Ethernet MAC

⇒ MAC == Media Access Control
- ☐ how do a bunch of devices share a wire?
- ☐ who talks when?
- ☐ for how long?
- ☐ what if people talk over each other?

...

⇒ Ethernet MAC is CSMA/CD
- ☐ Carrier Sense (CS)
- ☐ Mutiple Access (MA)
- ☐ With Collision Detetion (CD)

⇒ Origins in Aloha Net
- ☐ goal was to use wireless links to connect Hawaiian islands
- ☐ everyone had to share the same air so focused on sharing
- ☐ Ethernet shares the wire the way AlohaNet shared the air

⇒ Ethernet Frame
- ☐ 64-bit fixed preable — used to tell when a message starts and synchonize clocks
- ☐ 48-bit source and dest addresses
- ☐ 16-bit type
- ☐ 46-1500 <u>byte</u> body
- ☐ 32-bit CRC error checking code

CMSC417 Spring 2016 Lecture #18 4/13/2016

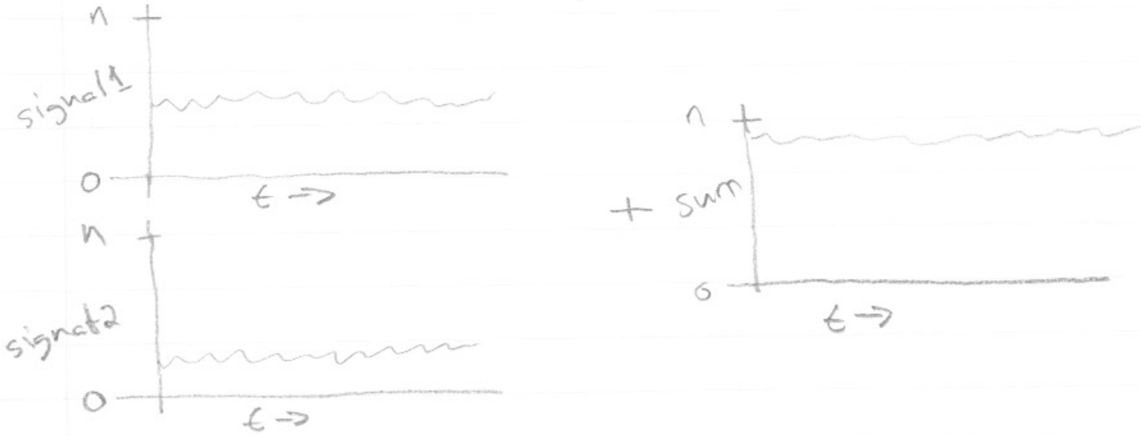
MAC algorithm (transmitter)

⇒ if you have something to send
- ☐ check if the wire is busy (CS)
- ☐ if it's free
  - send immediately
- ☐ if it's busy
  - poll continuosly until it's free and send immediately

⇒ multiple people can sense the wire is free and send at the same time (MA)
- ☐ if they do, its called a collision
- ☐ detect the collision (CD)
- ☐ wait a while
- ☐ sense the channel and try again

Collision Detection

⇒ how can you tell something went wrong?
- ☐ listen on the channel to see if you hear (only) what you sent
- ☐ check to see if there's too much signal strength on the wire

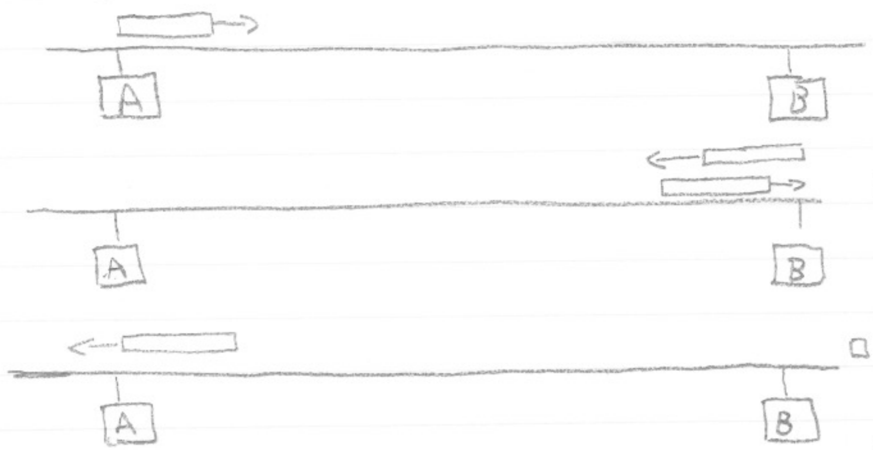


⇒ expect one transmitter to be ≤ X volts, if you see > X volts ⇒ collision

CMSC 417 Spring 2016 Lecture #18 4/13/2016

Collision Detetion (cont'd)

⇒ what if the collision occurs not at the sender?



- A sends
- B sends just as A's frame arrives ⇒ collision
- B's message arrives at A
  A has no idea ther was a collision

⇒ collision can happen at sender or reciever

⇒ sender wants to know to resend
- why? why not let the higher layer deal with retransmission?
- assume that people sent a frame b/c they wanted it to get there
- you can fix the problem (much) faster here, avoid hurting TCP

Jamming signal

⇒ on detecting collision, switch from sending frame to sending a 32-bit jamming signal

⇒ lets the other end know there was a collision

⇒ how can you tell a jamming signal from a 32-bit message with the same value? no CRC

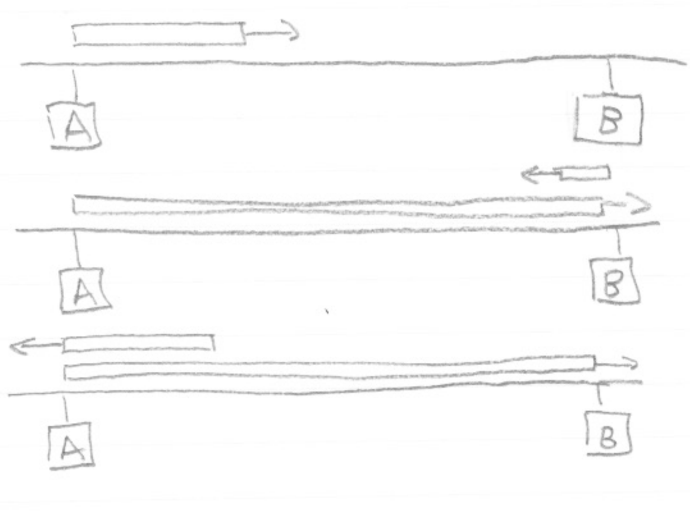
⇒ what if the jamming signal is corrupted?
- let higher levels retransmit

Collision Detection (cont'd)

⇒ In example, A discovers its collision with B on receiving the jamming signal
- ☐ when?
- ☐ is A still listening?

⇒ after an RTT, so is A still listening?
- ☐ two solns
  - A listens for a full RTT after sending
  - limit the RTT so that A must still be listening
- ☐ Ethernet uses the second
- ☐ min frame size is 512 bits (w/o pramble)
  - 48 + 48 + 16 + 46×8 + 32 = 512 bits

    dst src type body CRC
- ☐ it takes 51.2 μs to send 512 bits @ 10 Mbps
  - 10 Mbps was Ethernet's original rate
- ☐ limit RTT of Ethernets to 51.2 μs
  - A will still be sending (and listening) when the jamming signal comes back



with receiver's help, sender can always detect collisions

A is still sending and listening to get B's

CMSC417 Spring 2016 Lecture #18 4/13/2016

Exponential Backoff

⇒ how long shoud you wait to try again on a collision?

- ≥ 0 ms
- clearly different times for different transmitters
- randomly pick 0 or 51.2 ms?
  - ⇒ good for 2 people
  - ⇒ exponentially worse as # of transmitters grows
- make progress as long as min randomly generated wait time is unique
  - ⇒ i.e., sombody goes first
- to get this regualry, need v=n slots for n transmitters

soln

⇒ generate a random integer, n, from 0 to $2^c - 1$ when you've had c collisions

⇒ wait for n × 51.2 ms

⇒ dynamically spreads senders out until you're sending w/o conflicts

⇒ called exponential backoff b/c avg. wait after c collisions is

$$\frac{2^c - 1}{2} = 2^{c-1} - \frac{1}{2} = \Theta(2^c)$$

⇒ in practice systems have a max # of retries and a maximum value of c to bound delay

CMSC417 Spring 2016 Lecture #18 4/13/2016

Ethernet Max Cable Length revisited

- ⇒ Speed of light/electricity in fiber/copper is about $2/3 \times C = 2 \times 10^6$ m/s
- ⇒ $51.2\ \mu s \times 2 \times 10^6$ m/s $\sim= 10^4$ m
- ⇒ it needs to be round trip, so half that 5×10^3 m
- ⇒ 5 segments allowed (really 4 repeaters) 10^3 m
- ⇒ half it for safety ⇒ 500 m (cable length in original Ethernet)

Internet Checksum

- ⇒ sum all 16-bit words using ones complement arithmatic
- ⇒ use ones complement (negation) of that as the checksum. why?
- ⇒ ones complement is where $-x$ is represented by flipping the bits of x
  - e.g., 4-bits, 3 = 0011, -3 = 1100
    5 = 0101, -5 = 1010
- ⇒ when summing, you need to add shifted overflow bits to make thing work
  - -5 + -3 should be -8
  - 1100 // -3
    +1010 // -5
    10110
    + └→ 1
    0111 = -8 b/c 8 = 1000