CSE 461 - Module 7: Medium Access Control (MAC) Layer Part 1

Topic: How to share a “channel”

- Channels
  - Wires, RF spectrum, fiber
  - Full duplex / half duplex
- Share by:
  - time or frequency or code
- Static vs. dynamic sharing
  - Static: OTA TV (frequency), classrooms (time), ??? (code division)
  - Dynamic: seats in the HUB dining area, Go cars
    - Dynamic contention-free: gas station bathrooms that require a key
  - Static vs. Dynamic
    - Pro's of each
    - Con's of each
- Bursty traffic

Channel Properties

- Synchronized vs. unsynchronized transmissions
- Carrier sense, or not
- Collisions and collision detection

Code Division Multiplexing

- Section 2.5.5 of the text
- All stations are allowed to send at the same time, using the same range of frequencies
- Each station encodes data using a chip sequence
- Chip sequences are orthogonal to each other
  - Dot product of any two distinct sequences is 0
  - Dot product of a sequence with itself is m (length of the sequence)
  - Dot product of a sequence with the negations of itself is -m
- Example sequences:
  - A:(1, 1, 1, 1), B:(1, 1, -1, -1), C:(1, -1, -1, 1) , D:(1, -1, 1, -1)

- Transmit your sequence when you have a 1 to send, and transmit negation of your sequence when you have a 0 to send
- Example:
  - Data – A: 1, B:1, C:0, D:0
  - A:(1,1,1,1),   B:(1,1,-1,-1),   C:(-1, 1, 1, -1), D:(-1, 1, -1, 1)
- Decode
  - Your receive sum of the signals sent
  - Take dot product of that with chip sequence for station you want to listen to
  - If result is positive, that station sent a 1; if negative, it sent a 0
  - Example: A: 1, B: 1, C: 0, D: 0
    - (1,1,1,1) + (1,1,-1,-1) + (-1, 1, 1, -1) + (-1,1,-1,1) = (0, 4, 0, 0)
    - Listen for A: (0, 4, 0, 0) dot (1,1,1,1) = 0*1 + 4*1 + 0*1 + 0*1 = 4
      - So A sent 1
    - Listen for C: (0, 4, 0, 0) dot (1, -1, -1, 1) = -4
      - So C sent 0

Dynamic Multiple Access: Pure Aloha

- Original multiple access protocol
- Basic protocol: when you have data, send it
- No carrier sense, no collision detection
  - But does have lost frame detection and retransmission
- Collision resolution protocol
  - If you collide, pick a random delay in [0,T] and then send again
- Should T be big or small?
- Pure Aloha capacity
  - What is required for a transmission to succeed?
  - Assuming Poisson arrivals (basically, coin flips in each of infinitesimal time slots)
    - When the overall transmission rate is G, the probablity of no transmissions in a period of length t is e^{-Gt}
  - Now assuming transmissions are all of same duration (and calling that unit time)
    - A collision occurs if someone else starts a transmission either during our transmission or less than 1 time unit before we start
    - The probability of that is e^{-2G}
    - So rate of successful transmissions is Ge^{-2G}
      - From G=0 to G=1/2 this increases, and then decreases
      - Maximum is 1/2e (at G=1/2)

        - Note that the maximum occurs when the expected number of transmissions in a contention interval is 1
    - Is the achievable maximum goodput affected by the length of the transmission?

Slotted Aloha

- We can do better if we somehow synchronize transmissions
    - When you have data to send, wait to send until the next slot time arrives
        - Slots are the length of one transmission
- The collision window is now one transmission time, rather than two
- Probability of a collision is now e^{-G}, and maximum achievable throughput is 1/e (at G=1)