Lecture 5.4: Multiple Access Protocols
Link Layer: Five Common Problems

• Basic problem: you can’t just send IP datagrams over the link!
• We first consider how to encode bits into the signal at the source and recover bits at the receiving node
• Once it is possible to transmit bits, we need to figure out how to package these bits into FRAME
• Assume each node is able to recognize the collections of bits making up a frame, the third problem is to determine if those bits are in error: Error Detection and Correction
• If frames arriving at destination contain errors, how to recover from such losses: ARQ
• Final problem related to multiple-access link: how mediate access to a shared link so that all nodes have a chance to transmit – Our focus will be on Ethernet
Two types of “links”:

- **point-to-point**
  - PPP for dial-up access
  - point-to-point link between Ethernet switch and host

- **broadcast** (shared wire or medium)
  - old-fashioned Ethernet
  - upstream HFC
  - 802.11 wireless LAN

- shared wire (e.g., cabled Ethernet)
- shared RF (e.g., 802.11 WiFi)
- shared RF (satellite)
Multiple Access Protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - collision if node receives two or more signals at the same time

`multiple access protocol`

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination
Ideal Protocol

Broadcast channel of rate $R$ bps

1. when one node wants to transmit, it can send at rate $R$.
2. when $M$ nodes want to transmit, each can send at average rate $R/M$.
3. fully decentralized:
   – no special node to coordinate transmissions
   – no synchronization of clocks, slots
4. simple
MAC Protocols

Three broad classes:

• **Channel Partitioning**
  – divide channel into smaller “pieces” (time slots, frequency, code)
  – allocate piece to node for exclusive use

• **Random Access**
  – channel not divided, allow collisions
  – “recover” from collisions

• **“Taking turns”**
  – nodes take turns, but nodes with more to send can take longer turns
TDMA: time division multiple access

- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle
Channel Partitioning: FDMA

FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle
Can we transmit at the same time and in the same frequency?

The answer is YES: Using both time and frequency, but assigning different codes

CDMA: Code division multiple access
Random Access Protocols

• When node has packet to send
  – transmit at full channel data rate $R$.
  – no \textit{a priori} coordination among nodes
• two or more transmitting nodes $\rightarrow$ “collision”,
• If there is collision, what should a node do?
• random access MAC protocol specifies:
  – how to detect collisions
  – how to recover from collisions (e.g., via delayed retransmissions)
• Examples of random access MAC protocols:
  – slotted ALOHA, ALOHA
  – CSMA, CSMA/CD, CSMA/CA
Assumptions:

• All frames same size, let say $L$ bits
• Time divided into equal size slots (time to transmit 1 frame). Given rate $R$, size of a slot?
• Nodes start to transmit only at beginning of slots
• Nodes are synchronized: Each node knows when the slots begin.
• if 2 or more nodes transmit in slot, all nodes detect collision even before the slot ends.
Slotted ALOHA: Operations

• When a node has a fresh frame, *wait until the beginning of next slot to transmit*

• When transmitting:
  • If no collision: No need re-transmission, can also prepare a new frame to transmit in the next slot
  • Collision
    • Assume it can detect before the end of slot
    • Prepare to retransmit with probability $p$ *in each subsequent slot*
Slotted ALOHA: Pros and Cons

- **Pros**:
  - Efficient use of channel bandwidth
  - Lower collision rate compared to ALOHA

- **Cons**:
  - Requires precise timing
  - Can be sensitive to network load

Diagram:
- Node 1:
  - Slot 1
  - Slot 2
  - Slot 4
- Node 2:
  - Slot 2
  - Slot 3
- Node 3:
  - Slot 3

Time slots: C, E, C, S, E, C, E, S, S.
Efficiency of Slotted ALOHA

Efficiency: long-run fraction of successful slots (many nodes, all with many frames to send)

- Suppose: N nodes with many frames to send, each transmits in slot with probability $p$
- Prob that given node has success in a slot?
- Prob that any node has a success?

- Max efficiency: find $p^*$ that maximizes $Np(1-p)^{N-1}$
- For many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:

  Max efficiency = $1/e = .37$

At best: channel used for useful transmissions 37% of time!
Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- when frame first arrives
  - transmit immediately
- collision probability increases:
  - frame sent at $t_0$ collides with other frames sent in $[t_0-1,t_0+1]$
Efficiency of Pure ALOHA

\[ P(\text{success by given node}) = P(\text{node transmits}) \cdot \]
\[ P(\text{no other node transmits in } [t_0-1,t_0]) \cdot \]
\[ P(\text{no other node transmits in } [t_0,t_0+1]) \]
\[ = p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1} \]
\[ = p \cdot (1-p)^{2(N-1)} \]

... choosing optimum \( p \) and then letting \( n \rightarrow \infty \) ...

\[ = \frac{1}{2e} = 0.18 \]

even worse than slotted Aloha!
**CSMA**: listen before transmit:
If channel sensed idle: transmit entire frame
• If channel sensed busy, defer transmission

• human analogy: don’t interrupt others!
CSMA – Carrier Sense Multiple Access

- Non-persistent CSMA Protocol:
  
  **Step 1:** If the channel is idle, transmit immediately
  
  **Step 2:** If the channel is busy, *does not continuously sense*, wait a random amount of time and repeat **Step 1**
  
  - Random backoff reduces probability of collisions
  - Waste idle time if the backoff time is too long

- 1-persistent CSMA Protocol:
  
  **Step 1:** If the channel is idle, transmit immediately (probability 1)
  
  **Step 2:** If the channel is busy, *continue to listen until channel becomes idle*, and then transmit immediately
  
  - There will always be a collision if two nodes want to retransmit
  - (usually you stop transmission attempts after few tries)
CSMA – Carrier Sense Multiple Access

\[ S = \text{the throughput} \text{ (number of successful transmissions per time unit)} \]
\[ G = \text{the offered load} \text{ (number of attempted transmissions per time unit)} \]
**CSMA – Carrier Sense Multiple Access**

- *p*-persistent CSMA Protocol:
  
  **Step 1:** If the channel is idle, transmit with probability *p* (or delay a period of time with probability (1-*p*)

  **Step 2:** If the medium is busy, continue to listen until medium becomes idle, then go to **Step 1**

  **Step 3:** If transmission is delayed by one time slot, continue with **Step 1**

  - A good tradeoff between non-persistent and 1-persistent CSMA
• Assume that $N$ nodes have a packet to send and the medium is busy

• Then, $N_p$ is the expected number of nodes that will attempt to transmit once the medium becomes idle

• If $N_p > 1$, then a collision is expected to occur

Therefore, network must make sure that $N_p < 1$ to avoid collision, where $N$ is the maximum number of nodes that can be active at a time
CSMA – Carrier Sense Multiple Access

$S = \text{the throughput}$ (number of \textit{successful} transmissions per time unit)
$G = \text{the offered load}$ (number of attempted transmissions per time unit)
CSMA – Carrier Sense Multiple Access

0.01-persistent CSMA
Nonpersistent CSMA
0.1-persistent CSMA
0.5-persistent CSMA
1-persistent CSMA
Slotted Aloha
Aloha
CSMA with Collision Detection

Carrier Sense- *listen before transmit*:
If channel sensed idle: transmit entire frame
- If channel sensed busy, defer transmission
- human analogy: don’t interrupt others!

Collision Detection:
- human analogy: some one else begins talking at the same time: Stop talking
- A node listens to the channel when transmitting. If detect another transmission: stop (will go back)
collisions can still occur:
propagation delay means two nodes may not hear each other’s transmission

collision:
entire packet transmission time wasted

note:
role of distance & propagation delay in determining collision probability
CSMA/CD: carrier sensing, deferral as in CSMA

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- After aborting, wait a random amount of time, sense the channel before transmitting.

**collision detection:**

- easy in wired LANs: measure signal strengths, compare transmitted, received signals
- difficult in wireless LANs?
CSMA/CD (Collision Detection)

t₀

collision detect/abort time

t₁

space

A B C D

time
• New frame: Sense channel, if idle, transmit. If busy, wait until idle
• When transmitting, if no collision, finish the frame
• If collision, abort transmission, e.g., stop transmitting frame
• After aborting, wait a random amount of time then sense the channel before transmitting
“Taking Turns” MAC Protocols

channel partitioning MAC protocols:
– share channel *efficiently* and *fairly* at high load
– inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

random access MAC protocols
– efficient at low load: single node can fully utilize channel
– high load: collision overhead

“taking turns” protocols
Try to look for best of both worlds!
Polling:

• master node “invites” slave nodes to transmit in turn
• typically used with “dumb” slave devices
• Concerns?
Token passing:
- control token passed from one node to next sequentially.
- token message
- Concerns?
Summary of MAC Protocols

• channel partitioning, by time, frequency or code
  – Time Division, Frequency Division

• random access (dynamic),
  – ALOHA, S-ALOHA, CSMA, CSMA/CD
  – carrier sensing: easy in some technologies (wire), hard in others (wireless)
  – CSMA/CD used in Ethernet
  – CSMA/CA used in 802.11

• taking turns
  – polling from central site, token passing
  – Bluetooth, FDDI, IBM Token Ring