

Physical Layer

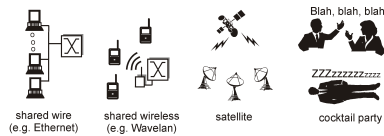
Srinidhi Varadarajan

P

Medium Access Links and Protocols

Three types of "links":

- point-to-point (single wire, e.g. PPP, SLIP)
- **broadcast** (shared wire or medium; e.g. Ethernet, Wavelan, etc.)



- switched (e.g., telephone systems, switched Ethernet, ATM etc)

Point-to-Point protocols

- Telephone networks
 - Switched hierarchy.
 - Local Loop is the last mile interface to customer premises equipment. (generally referred to in the networking world as the source of all evil)
 - Originally involved a physical connection between the sender and the receiver.
 - Nowadays, telephone networks use circuit switched medium access control
- Modems: Digital interface to the world of telephony

Modems: Signaling

- Modems:
 - Work over low bandwidth telephone lines (3000 Hz)
- Signaling schemes: (why not just use digital bit patterns?)
 - Possible choices:
 - Amplitude modulation (AM)
 - Frequency modulation (FM or FSK)
 - Phase modulation (PSK)

Modems Signaling

- Modern modems use a combination of PSK and AM
- Create charts called constellation patterns.
 - Multiple bits encoded per signal.
 - Trellis encoding is used to minimize the chance of error. Errors cause loss of several bits
- Echo cancellation/suppression
 - Needed for long-haul voice communication.
 - Prevents full duplex
 - **In-band signaling** at 2100 Hz is used to inhibit echo cancellation circuitry.
 - Newer solution uses end-point resources for echo suppression.

RS-232C, RS449: Point-to-Point Communication

- RS-232C and RS449 specify physical layer point-to-point serial communication
- 25 or 9 pin connectors, 15m cable length
 - $<-3V = 1, >+4V=0$,
 - BW: 20Kbps (originally, upgraded now to up to 115Kbps)
 - Main communication occurs using the RTS/CTS protocol.
- RS-449 is an upgraded RS-232C with 2 modes of communication
 - Unbalanced mode, physically is similar to RS-232C, with common ground signaling.
 - Balanced mode uses independent ground. Data rate 2Mbps with lengths up to 60m

Multiple Access protocols

- single shared communication channel
- two or more simultaneous transmissions by nodes: interference
 - only one node can send **successfully** at a time
- **multiple access protocol:**
 - distributed algorithm that determines how stations share channel, i.e., determine when station can transmit
 - communication about channel sharing must use channel itself!
 - what to look for in multiple access protocols:
 - synchronous or asynchronous
 - information needed about other stations
 - robustness (e.g., to channel errors)
 - performance

Multiple Access protocols

- claim: humans use multiple access protocols all the time
- class can "guess" multiple access protocols
 - multiaccess protocol 1:
 - multiaccess protocol 2:
 - multiaccess protocol 3:
 - multiaccess protocol 4:

MAC Protocols: a taxonomy

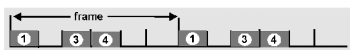
Three broad classes:

- **Channel Partitioning**
 - divide channel into smaller "pieces" (time slots, frequency)
 - allocate piece to node for exclusive use
 - **Random Access**
 - allow collisions
 - "recover" from collisions
 - **"Taking turns"**
 - tightly coordinate shared access to avoid collisions
- Goal: efficient, fair, simple, decentralized

Channel Partitioning MAC protocols: TDMA

TDMA: time division multiple access

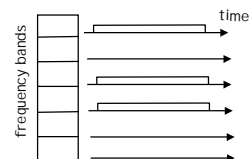
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle



Channel Partitioning MAC protocols: 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

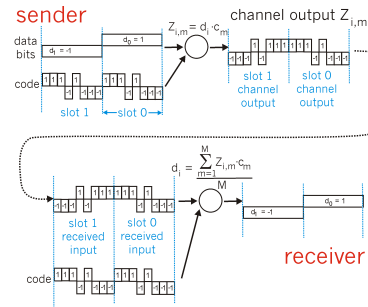


Channel Partitioning (CDMA)

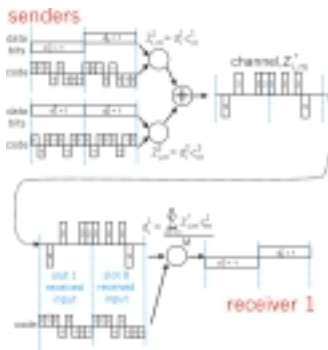
CDMA (Code Division Multiple Access)

- unique "code" assigned to each user; ie, code set partitioning
- used mostly in wireless broadcast channels (cellular, satellite, etc)
- all users share same frequency, but each user has own "chipping" sequence (ie, code) to encode data
- **encoded signal** = (original data) X (chipping sequence)
- **decoding**: inner-product of encoded signal and chipping sequence
- allows multiple users to "coexist" and transmit simultaneously with minimal interference (if codes are "orthogonal")

CDMA Encode/Decode



CDMA: two-sender interference

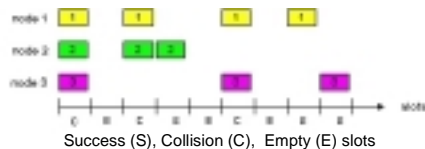


Random Access protocols

- When node has packet to send
 - transmit at full channel data rate R.
 - no *a priori* coordination among nodes
- two or more transmitting nodes -> "collision",
- **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 and CSMA/CD

Slotted Aloha

- time is divided into equal size slots (= pkt trans. time)
- node with new arriving pkt: transmit at beginning of next slot
- if collision: retransmit pkt in future slots with probability p , until successful.



Slotted Aloha efficiency

Q: what is max fraction slots successful?

A: Suppose N stations have packets to send
 – each transmits in slot with probability p
 – prob. successful transmission S is:

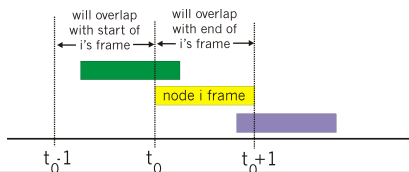
$$\text{by single node: } S = p(1-p)^{(N-1)}$$

$$\begin{aligned} \text{by any of } N \text{ nodes} \\ S &= \text{Prob (only one transmits)} \\ &= N p (1-p)^{(N-1)} \\ &\dots \text{ choosing optimum } p \text{ as } n \rightarrow \text{infty} \dots \\ &= 1/e = .37 \text{ as } N \rightarrow \text{infty} \end{aligned}$$

At best: channel use for useful transmissions 37% of time!

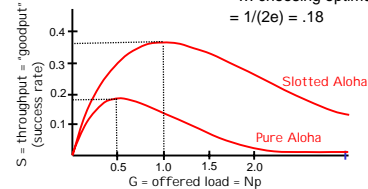
Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- pkt needs transmission:
 - send without awaiting for beginning of slot
- collision probability increases:
 - pkt sent at t_0 collide with other pkts sent in $[t_0-1, t_0+1]$



Pure Aloha (cont.)

$$\begin{aligned}
 P(\text{success by given node}) &= P(\text{node transmits}) \cdot \\
 &P(\text{no other node transmits in } [p_0-1, p_0]) \cdot \\
 &P(\text{no other node transmits in } [p_0, p_0+1]) \\
 &= p \cdot (1-p) \cdot (1-p) \\
 P(\text{success by any of } N \text{ nodes}) &= N p \cdot (1-p) \cdot (1-p) \\
 &\dots \text{ choosing optimum } p \text{ as } n \rightarrow \text{infy} \dots \\
 &= 1/(2e) = .18
 \end{aligned}$$



protocol constrains effective channel throughput!

CSMA: Carrier Sense Multiple Access

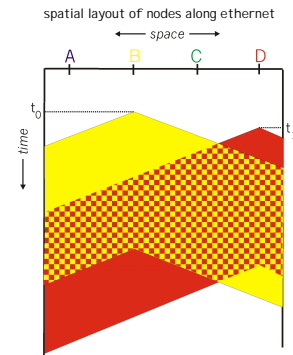
- CSMA:** listen before transmit:
- If channel sensed idle: transmit entire pkt
 - If channel sensed busy, defer transmission
 - **Persistent CSMA:** retry immediately with probability p when channel becomes idle (may cause instability)
 - **Non-persistent CSMA:** retry after random interval
 - human analogy: don't interrupt others!

CSMA collisions

collisions can occur:
propagation delay means two nodes may not yet hear each other's transmission

collision:
entire packet transmission time wasted

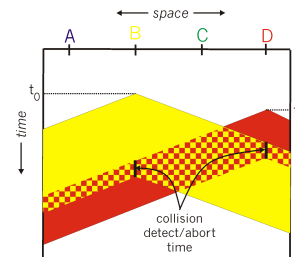
note:
role of distance and propagation delay in determining collision prob.



CSMA/CD (Collision Detection)

- CSMA/CD:** carrier sensing, deferral as in CSMA
- collisions *detected* within short time
 - colliding transmissions aborted, reducing channel wastage
 - persistent or non-persistent retransmission
 - collision detection:
 - easy in wired LANs: measure signal strengths, compare transmitted, received signals
 - difficult in wireless LANs: receiver shut off while transmitting
 - human analogy: the polite conversationalist

CSMA/CD collision detection



“Taking Turns” MAC protocols

channel partitioning MAC protocols:

- share channel efficiently 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

look for best of both worlds!

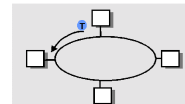
“Taking Turns” MAC protocols

Polling:

- master node “invites” slave nodes to transmit in turn
- Request to Send, Clear to Send msgs
- concerns:
 - polling overhead
 - latency
 - single point of failure (master)

Token passing:

- control **token** passed from one node to next sequentially.
- token message
- concerns:
 - token overhead
 - latency
 - single point of failure (token)



Reservation-based protocols

Distributed Polling:

- time divided into slots
- begins with N short **reservation slots**
 - reservation slot time equal to channel end-end propagation delay
 - station with message to send posts reservation
 - reservation seen by all stations
- after reservation slots, message transmissions ordered by known priority

