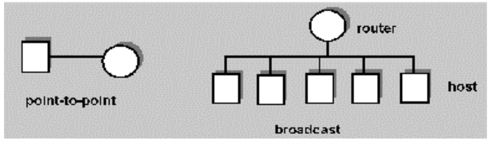
The Data Link Layer

- introduction
- point-to-point data link protocols
- the multiple access problem
- local area networks
- required reading:
  - Tannenbaum ch 3, 4
  - Kurose, Ross ch 5

Data Link Layer : Introduction

Services: reliably deliver a data link packet between two physically connected machines

- two link types: point-to-point, broadcast



Point-to-point links: one sender, one receiver

- framing: recognizing bits on the wire as packets
- reliable communications

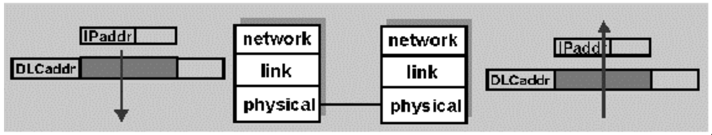
Data Link Layer : Introduction

broadcast links: many senders, potentially many receivers

- framing
- reliable communication
- accessing a shared medium
- addressing
- many senders many receivers

Data Link Layer: Introduction

- reliable communication: ARQ, checksum, timers, sequence numbers
- addressing
  - data link level addresses different from network layer addresses!
  - why do we need different data link address?



Data Link Layer Services

- Three possible services provided to network layer
  - Unacknowledged connectionless service
    - no error recovery, suitable for low error rate channels
  - Acknowledged connectionless service
    - suitable for unreliable channels
  - Acknowledged connection-oriented service
    - suitable for WAN subnets connected by point-to-point leases lines

ARQ-based Protocols

- Automatic repeat request (ARQ)
  - detect transmission errors and request retransmission
- Stop-and-wait ARQ
  - ensure each packet has been received correctly before sending next; uses acks/nacks
  - need to use sequence numbers
- Go back n ARQ
  - send packets numbered sequentially
  - receiver sends ack with the largest in-order packet received
  - n determines how many packets can be sent before waiting

Error detection and Correction

- Parity checks
  - one bit parity for every n bits
  - two dimensional parity
- Checksums
- Cyclic redundancy checks
  - add *r* bits to a *d* bit string such that *(d+r)* bits are divisible by a generator *G*
- Homework: read KR sec 5.2

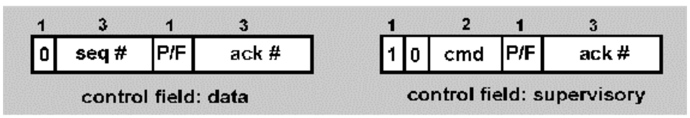
Point-to-point Data Link Control

- HDLC: high level data link protocol (it's old - data link was "high-level" way back when)
- HDLC frame format:

flag	address	control	data	checksum	flag
8 bits	8	8	arbitrary	16	8

- flag pattern (01111110) is used to mark beginning/end of frame
  - bit stuffing: if five consecutive 1's in data, sender adds a 0, receiver removes
- address of receiving node (for broadcast links)

HDLC: control field



- control field format for "data" frames:
  - 3-bit seq number
  - 3-bit ack number
  - 1 bit P/F to indicate sender-to-receiver to vice-versa
- control field format for "supervisory" frames:

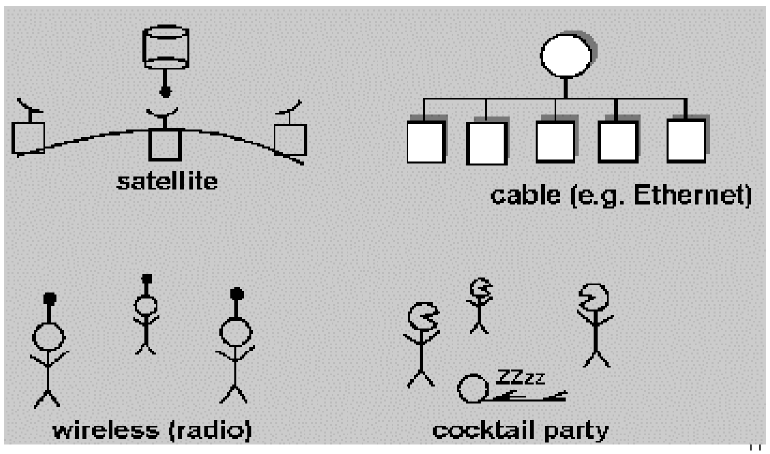
command	meaning
receive ready	ack
receive not ready	flow control:not ready
reject	NAK: resend go back N
selective reject	NAK: resend selective repeat

Broadcast links: Multiple Access Protocols

Single shared communication channel

- two or more simultaneous transmissions by nodes: **interference**
- only one node can send successfully at a time
- question: how to share this broadcast channel examples of multiple access environments:

Examples



Multiple Access Protocols

Distributed algorithm which 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

Some 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:

Human MAPs

- Give everyone a chance to speak
- Don’t speak until you are spoken to
- Don’t monopolize the conversation
- Raise your hand if you have a question
- Don’t interrupt when someone is speaking
- Don’t fall asleep when someone is talking

A taxonomy of multiple access protocols

Random access protocols: stations contend for channel, collisions (overlapping transmissions can occur):

- aloha
- slotted aloha
- carrier sense multiple access: Ethernet
- group random access

Taxonomy of MAPs (cont.)

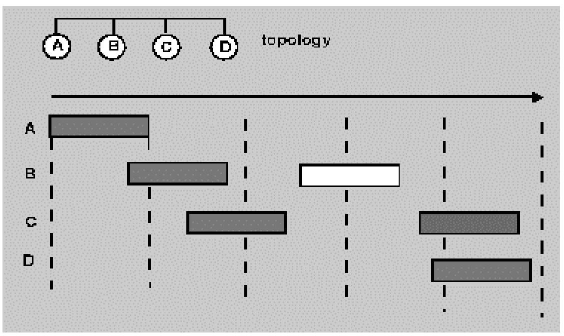
Controlled access protocols: stations reserve or are assigned channel, no collisions

- predetermined channel allocation: time division multiple access
- demand adaptive channel allocation
  - reservation protocols
  - token passing (token bus, token ring)

The Aloha Protocol

simple: if you have pkt to send, "just do it"

- if pkt suffers collision, will try resending later



Analyzing the Aloha Protocol

Goal: quantitative understanding of performance of Aloha protocol

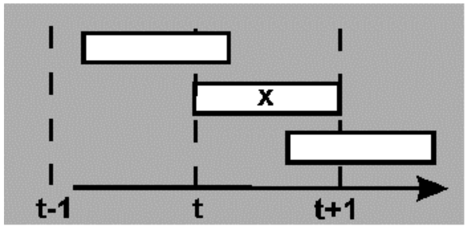
- fixed length pkts
- pkt transmission time is unit of time
- ***throughput*** S - number of pkts successfully (without collision) transmitted per unit time
  - in previous example, S = 0.2 pkt/unit time

- ***offered load G*** - number pkt transmissions attempted per unit time
  - ***note:*** $S<G$, but S depends on G
  - ***Poisson model:*** probability of k pkt transmission attempts in t time units

    Prob[k trans in t] = $((Gt)^k)(e^{-Gt})/k!$
  - infinite population model
- **capacity of multiple access protocol**: maximum value of S over all values of G

Analyzing Aloha (cont)

Focus on a given attempted packet transmission



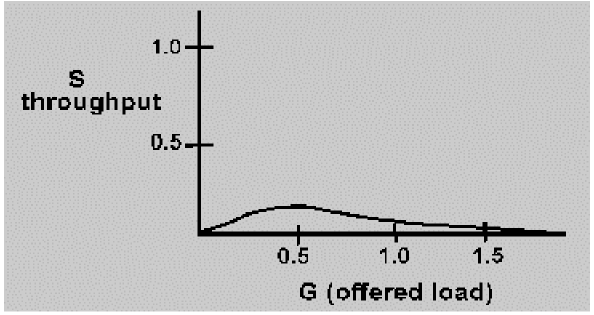
S = rate attempted pkt trans * Prob[successful trans]

= G*Prob[no other pkt's overlap with attempted trans]

= G*Prob[0 other attempted trans in 2 time units]

= Ge^{-2G}

Aloha throughput



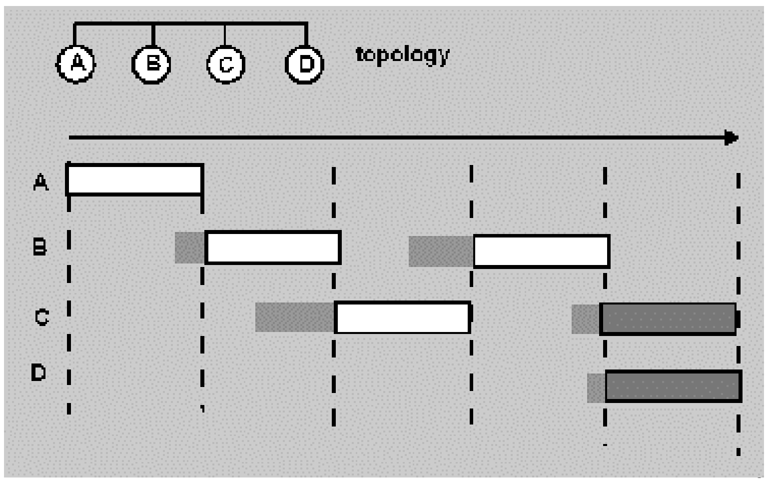
Note: maximum throughput is 18% of physical channel capacity

- you buy 1 Mb link, thoughput will never be more than 180Kb!

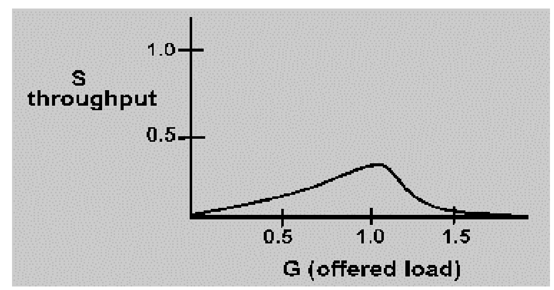
Slotted Aloha

- synchronous system: time divided into slots
- slot size equals fixed packet transmission time
- when pkt ready for transmission, wait until start of next slot
- packets overlap completely or not at all

Slotted Aloha performance



S = G*Prob[no other transmissions overlap]
= G*Prob[0 other attempted transmissions]
= G*Prob[0 other arrivals in previous slot]
= Ge^{-G}



Carrier Sensing Protocols

- Aloha is inefficient (and rude!): doesn't listen before talking!
- Carrier Sense Multiple Access: CSMA

non-persistent CSMA:

1. sense (listen to) channel
2. if {channel sensed busy}

then wait random time; go to 1

else transmit packet

Carrier Sensing Protocols (cont)

p-persistent CSMA:

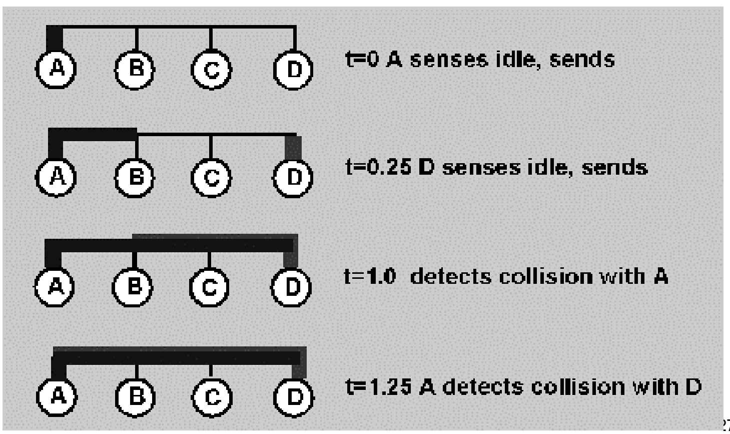
1. sense (listen to) channel
2. when {channel sensed idle}

transmit with probability *p*

else wait random time, go to 1

Carrier sensing protocols (cont)

- channel sensing will not avoid all collisions:



Carrier Sensing (cont.)

- performance will depend on channel length
  - large propagation delays: poor performance
  - length of CSMA networks must be limited

Can we do better?

CSMA/CD

- CSMA with collision detection(CD):
  - listen while talking!
  - stop transmitting when another pkt has collided with your pkt
- wait random time before attempting to resend
- worst case time to detect a collision?
- performance depends (as in CSMA) on channel length

Case Study: Ethernet

- CSMA/CD, 1-persistent
- IEEE 802.3 standard
- channel: coaxial cable (typically)
- *T*: minimum randomization interval

Collision resolution: binary backoff: pkt arrives (from upper layer) for transmission.

1. ***Set L=1, mark pkt as "ready"***
2. ***after successful transmission, all hosts with "ready" pkt can send***
3. ***if {collision}***

    L=L*2, up to 1024

    wait random amt of time over next L*T time units

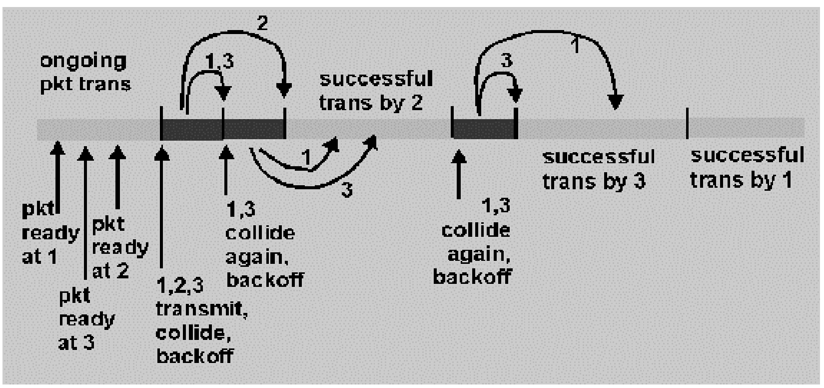
    after waiting, pkt is again "ready"

    go to 2

Note: backoff interval dynamically adjusts to load

- different hosts will have different values of *L*
- light load: small values of *L* (typically)
- heavy load: larger *L*

Ethernet: example



More on ethernet

- 10 Mb/sec, 100 Mb/sec standards
- packet format:

preamble	start frame	destination address	source address	length	data	pad	checksum

- *preamble:* 7 bytes to allow sender/receiver clock synch
- *start-of-frame:* 1 byte, denotes start of from (like HDLC)
- *destination address:*
  - 48 bit address "physical address"
  - **different from IP address!!!!**
  - each Ethernet board in world has own unique address hard-wired (IEEE and vendor assigned)
  - dest. address all 1's for broadcast pkt: will be received by all hosts attached to LAN

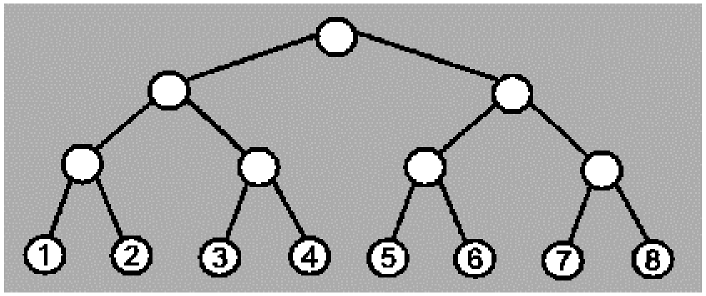
More on ethernet

preamble	start frame	destination address	source address	length	data	pad	checksum

- *source address:* 48-bit physical address
- *length:* 2 bytes, max packet length is 1500 bytes
  - recall IP fragmentation
- *data:* contains packet (e.g., IP packet) handed down from upper layer
- *padding:* used to insure data plus padding > 46 bytes
- *checksum*

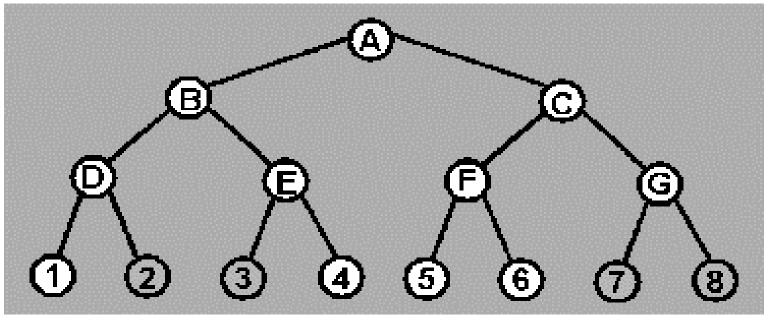
Group Random Access Protocols

- rather than random backoff to separate colliding stations, structured "search for exactly one station
- enable group of stations
- if collisions occur, divide group until only one ready station is enabled
- tree traversal: think of stations at leaves on logical binary tree:



1. all stations rooted at rootnode enabled
2. if {no stations send)
        return
    else if (one station sends)
        return
    else /* collision */
        resolve(leftchild(rootnode))
        resolve(rightchild(rootnode))

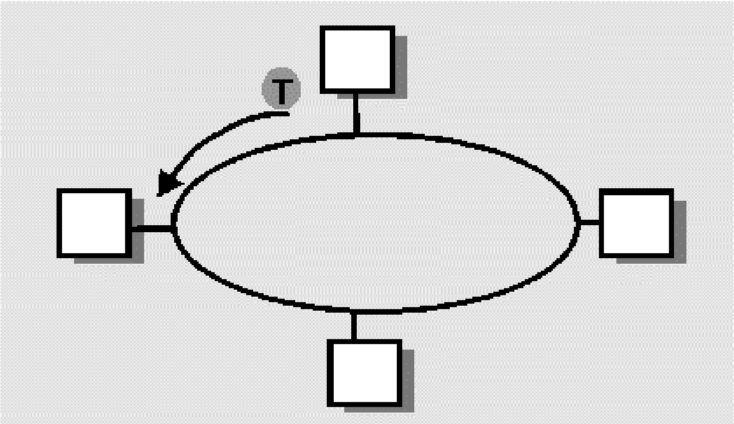
Group Random Access: example



suppose stations 2,3,7,8 ready with pkt
A enabled, collisions
B enabled collisions
D enabled, SUCCESS by 2
E enabled SUCCESS by 2
C enabled, collisions
F enabled, idle
G enabled, collisions (could have avoided!)
7 enabled, SUCCESS
8 enabled, SUCCESS

Token Passing Protocols

- token circulates among stations
- media:
  - token ring connection: IEEE802.5, FDDI
  - token bus, IEEE802.4
- to transmit
  - station must seize token
  - transmit packet while holding token
  - release (send out) token


T