Medium Access Control

CSC 343·643

Wake Forest University
Department of Computer Science

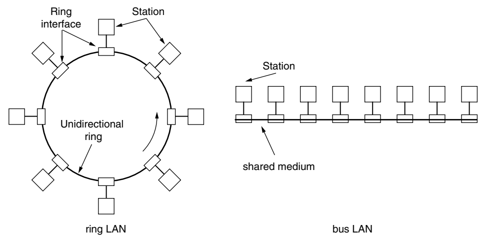
Fall 2013

Local Area Networks

- Previously we have primarily discussed point-to-point connections
  - Connecting two stations together
  - *Now consider connecting multiple stations together*
- Local Area Networks (LAN) have a *broadcast* characteristic
  - Each station connected to a medium shared by others
  - Transmission of one station is received by all others
  - Routing is **not** required

    What does this mean?
- Given multiple local stations, *how are they connected?*

LAN Topologies

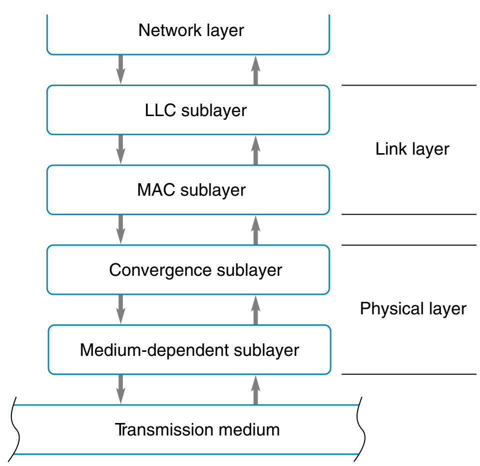


- Ring
  - Unidirectional, receive on one link and transmit on the other
    Is this routing?
- Bus
  - Stations *tap* into shared medium
    How are stations typically connected?
- Topology will impact the **Medium Access Control** (MAC)

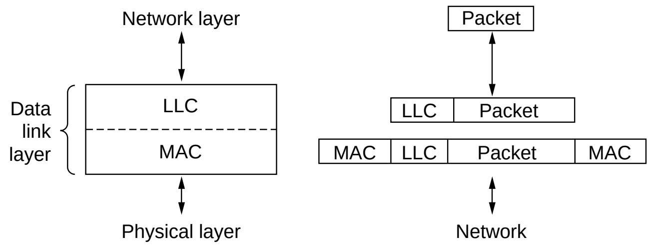
Medium Access Control

A LAN consists of multiple stations sharing a medium

- Method to control access → Medium Access Control (MAC)



- Technically MAC is a sublayer (bottom of the data-link layer)



- Logical Link Control (LLC) - At the top of the data-link layer
  - Provides error and flow control
- We will focus on the MAC sublayer
  - Question to resolve - *Who may send next?*
  - An allocation problem

  What about multiplexing?

Channel Allocation Problem

How do we allocate a shared channel among competing stations?

- Static channel allocation
  - TDM approach, each user has a specific time (slot)
  - Already determine static allocation is *inefficient*

    It is efficient for telephone networks, why not in a LAN?

- Dynamic channel allocation is better for bursty traffic

  Is there any reason for static allocation in computer networks?

MAC Types

Three general types: *contention*, *round-robin*, and *reservation*

- Contention - *No permission to send*
  - pure ALOHA, slotted ALOHA, CSMA, CSMA/CD
  - Bus topology (and wireless)
- Round robin - *Send when you have permission*
  - Token passing or polling
  - Ring or bus topology
- Reservation - *Request before sending*
  - Similar to TDM
  - Dynamic, must request *slot* before sending
  - Ring or bus topology

  Is control centralized or distributed in the above?

Pure ALOHA

Protocol developed at the University of Hawaii (1970's)

- Simple protocol: *Send when you have data to send*
- Collisions may occur
  - Two frames transmitted at (or during) the same time
  - Signal garbled, data is lost
  - *Even just a single bit overlap...*
- Since broadcast, sender can always tell if collision occurred
  - Listen during transmission[a]
  - If collision, wait a random time and try again

    Why random?

[a] Actually wireless networks may not do this

Performance of Pure ALOHA

- Assume a fixed frame size (constant transmission time)
- Users generate new frames according to a Poisson distribution with mean n frames per frame-time
  - If $n > 1$ more frames generated than channel can handle
  - For reasonable performance, assume $0 < n < 1$
- Assume the probability of k transmission attempts during a frame-time is also Poisson, with mean g per frame-time

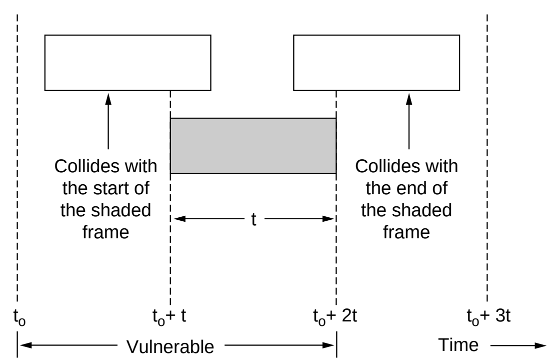
  We will consider the case where $g \geq n$, why?

- Under any load the throughput is

$$s = p_0 \cdot g$$

  where p_0 is the probability that a frame does not suffer a collision

- Assume you are sending the gray frame, let t be the time to send



- *When would another frame collide with the gray frame?*
  - The *vulnerable period* is between t_0 and $t_0 + 2t$

  Why is the collision at the beginning a possibility?

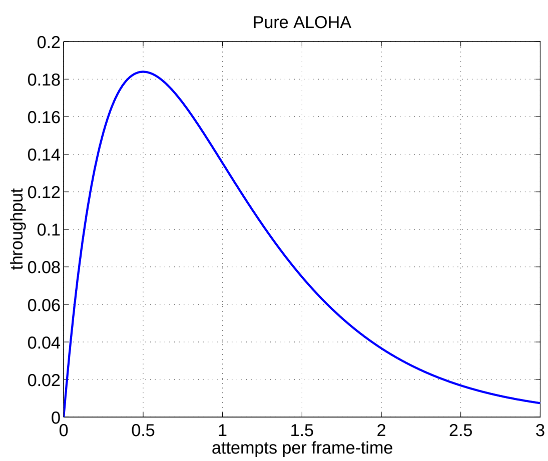
- The probability that k frames are transmitted during the *vulnerable period* is given from the Poisson distribution

$$p\{k \text{ transmissions in } 2g \text{ seconds}\} = \frac{(2g)^k \cdot e^{-2g}}{k!}, \quad k = 0, 1, 2, \ldots$$

- Remember, throughput is equal to the total arrival rate times the probability of a successful transmission

$$s = g \cdot p\{\text{no transmissions in } 2g \text{ seconds}\}$$

$$s = g \cdot \frac{(2g)^0 \cdot e^{-2g}}{0!} = g \cdot e^{-2g}$$



- Highest possible throughput is 18%

What is an example ALOHA network?

Slotted ALOHA

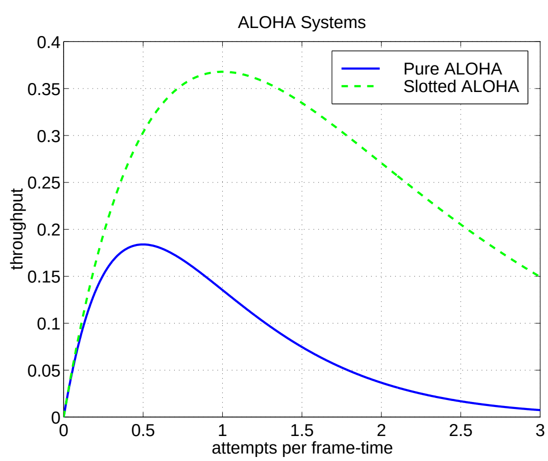
An *improvement* of ALOHA, where time is divided into slots

- Time divided into discrete slots (equal to one frame)
  - Requires users to agree on slot boundaries (*not trivial*)
  - Continuous protocol is now discrete
- Simple protocol: *Send when you have data, must wait until beginning of next slot*
- Vulnerable period is halved

  Why is the vulnerable period halved?

- As a result throughput is

$$s = g \cdot e^{-g}$$



- Highest possible throughput 36.8% occurs when $g = 1$
- Therefore, the best performance is when 37% slots are empty, 37% are successes, and 26% are collisions

Carrier Sense Multiple Access Protocols

- ALOHA systems were precursors to CSMA
  - All are contention-based access protocols

  Slotted ALOHA improved pure ALOHA by...

- CSMA increases performance by
  - *Listening to the carrier before sending* (as the name implies)
  - Does assume $t_{prop} << t_{trans}$
- Three different CSMA types
  - 1-persistent
  - non-persistent
  - p-persistent
- CSMA types differ on when they access an *idle* channel

CSMA Variations

1-persistent, listen to medium before sending

1. If idle transmit, otherwise go to step 2
2. If busy, continue to listen until channel idle, then transmit immediately

---

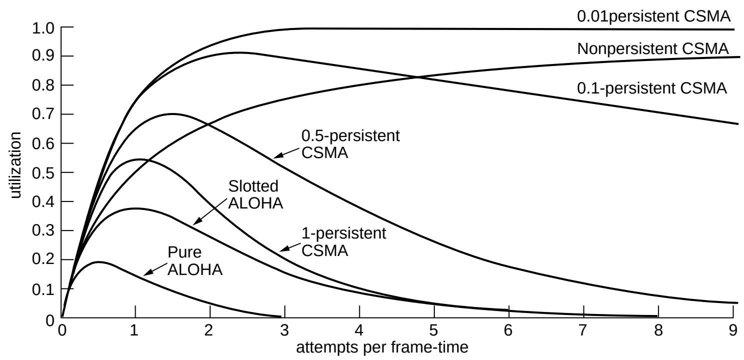
non-persistent, listen to medium before sending

1. If idle then transmit immediately, otherwise go to step 2
2. If busy then wait random amount of time, then go to step 1

---

p-persistent, listen to medium before sending

1. If idle transmit with probability p, otherwise go to step 2
2. If busy then wait until channel idle, then go to step 1



- Listening before sending increases utilization

  Persistent increases performance, why? The optimal value of p?

- One problem remains, if a collision occurs the medium remains unusable for the duration of the damaged frames

CSMA with Collision Detection

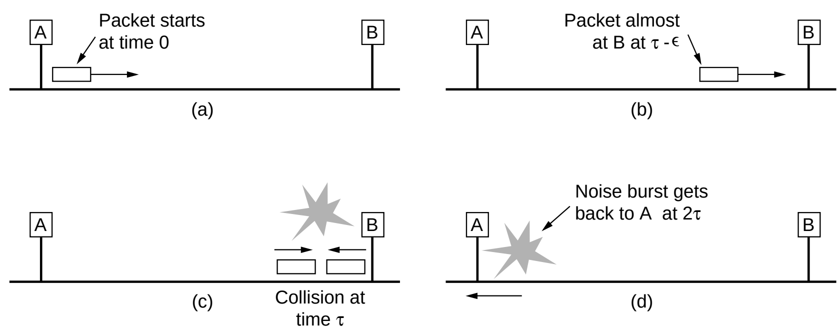
CSMA/CD improves performance by aborting transmission *immediately* once a collision occurs

- Transmission rules
  1. Medium idle transmit; otherwise go to step 2
  2. Medium busy, listen until idle then transmit immediately
  3. If collision detected, transmit brief jamming signal
  4. After jamming, wait random amount of time then go to step 1
- Wasted capacity is reduced to the time to detect a collision

  How long does it take to detect a collision?

  Just the propagation delay between the two farthest stations?

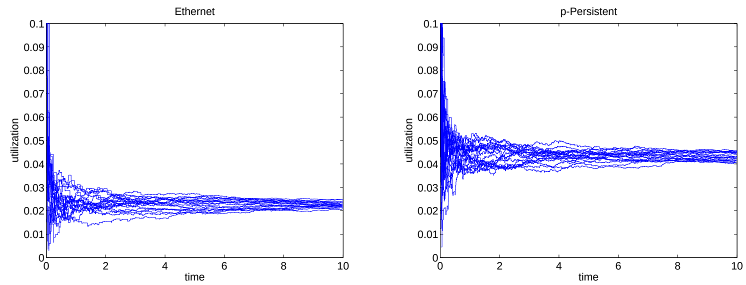
- Consider the worst case
  - Let the propagation time between the two farthest stations (A and B) be t_{prop}
  - At time t_0 station A transmits
  - At time $t_0 + t_{prop} - \epsilon$ station B transmits
  - Station B detect the collision immediately
  - Station A detects at time $t_0 + t_{prop} - \epsilon + t_{prop} \equiv t_0 + 2t_{prop}$



- Therefore a station cannot be certain it has *seized* the medium until $2t_{prop}$ has passed
- As a result CSMA/CD systems require frames to be long enough to allow collision detection prior to end of transmission
  - Collision detection can only occur during transmission

p-Persistent and CSMA/CD Performance

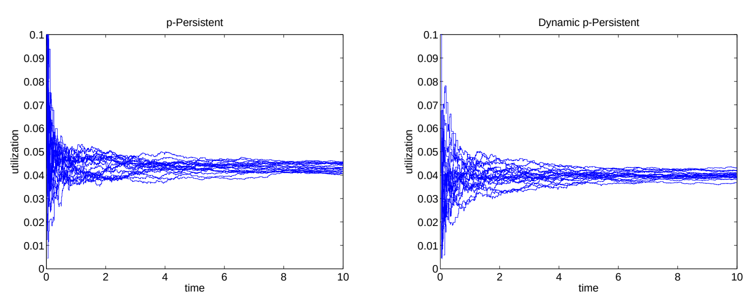
- Consider 20 stations using p-persistent or CSMA/CD
  - Assume the p value is set to the *optimal value*
- The average utilization per station for different MAC's



What is the difference between CSMA/CD and p-persistent?
Disadvantage of optimal p-persistent?

Dynamic p-Persistent

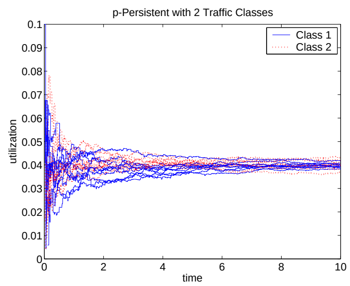
- Set the value of p based on the number of stations
  - A collision is a *good* indicator [Kester & Fulp 04]
- Let n_w be the number of collisions during time w, then $p = \frac{1}{2 \cdot n_w}$

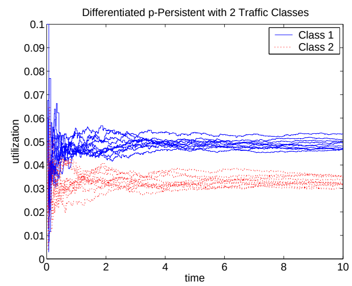


What are the advantages of dynamic p-persistent

Service Differentiation

- Consider 20 stations and 2 classes of traffic
  - Assume class 1 has higher priority than class 2
- Set p to increase the utilization of the higher class





What is the difficulty of setting p dynamically?

- This strategy can be used for any shared medium (wireless)

Token Passing

- Contention-based MACs have poor performance with *heavy* traffic
  - Due to the high number of collisions
- Token passing requires stations to take turns (*no collisions*)
- A small frame called a *token* is circulated when stations are idle
- A station wishing to transmit must wait for the token
  1. The station seizes the token and transmits its data frame
  2. Data frame will circulate the ring, the transmitting station will release the token if
     (a) Data transmission complete
     (b) Leading edge of data frame has returned (circulated)

Token passing and polling are similar since both require permission. What is the difference?

CSMA/CD and Token Passing Performance

- Want to compare *utilization* contention and token passing

$$u = \frac{\text{throughput}}{\text{data rate}}$$

- Define the following variables

  r data rate of channel

  d maximum distance between any two stations

  v velocity of signal propagation

  l frame length (fixed)

- Throughput is number of bits transmitted per unit time

$$\text{throughput} = \frac{l}{\frac{d}{v} + \frac{l}{r}}$$

- Using the definition of a

$$a = \frac{t_{prop}}{t_{frame}} = \frac{d/v}{l/r} = \frac{r \cdot d}{l \cdot v}$$

  rewrite the utilization equation (using the throughput equation) as

$$u = \frac{1}{1 + a}$$

- Therefore utilization will depend on a (propagation delay and frame transmission time)

Token Passing

- Assume a LAN with n stations, with maximum normalized propagation delay of a
- Furthermore, assume each station is always prepared to send
- Time on the ring will alternate between data transmission and token passing
  - Refer to a single instance of a data frame followed by a token as a *cycle*

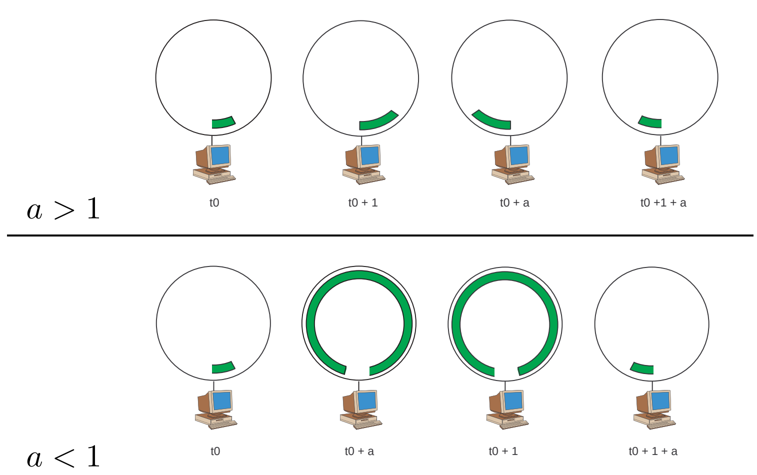
  Define the following variables

c	average time for cycle
t_{data}	average time to transmit data frame
t_{token}	average time to transmit token

- The average cycle rate is $\frac{1}{c} = \frac{1}{t_{data}+t_{token}}$, therefore

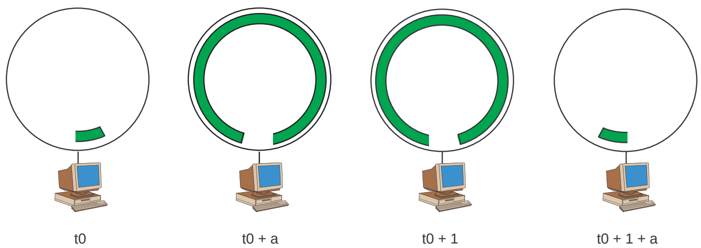
$$u = \frac{t_{data}}{t_{data} + t_{token}}$$

- Must consider two cases



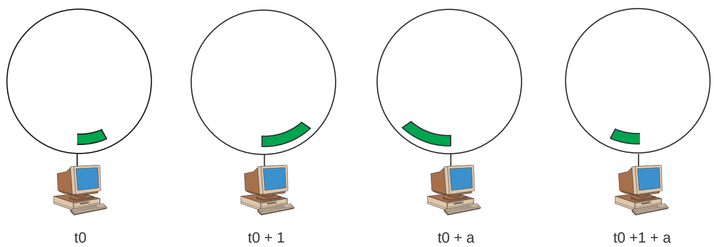
- For $a < 1$
  - Station starts at time t_0
  - Receives leading edge of frame at time $t_0 + a$
  - Completes transmission at time $t_0 + 1$
  - Station releases token, takes $\frac{a}{n}$ time to reach next station, therefore cycle time is $1 + \frac{a}{n}$, utilization is

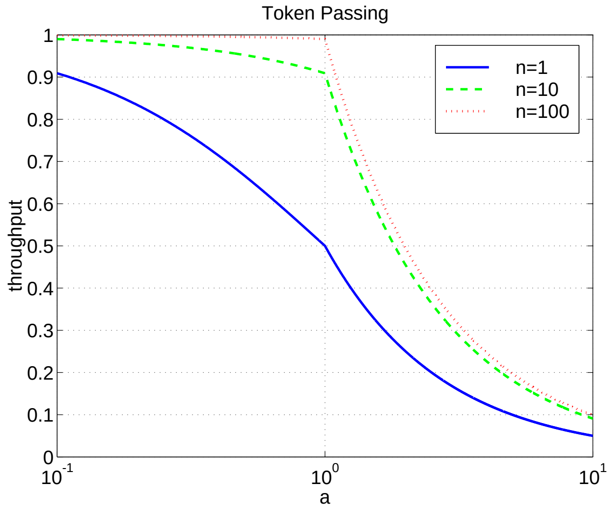
$$u = \frac{1}{1 + \frac{a}{n}}$$



- For $a > 1$
  - Station starts at time t_0
  - Completes transmission at time $t_0 + 1$
  - Receives leading edge of frame at time $t_0 + a$
  - Station releases token, takes $\frac{a}{n}$ time to reach next station, therefore cycle time is $a + \frac{a}{n}$ and utilization is

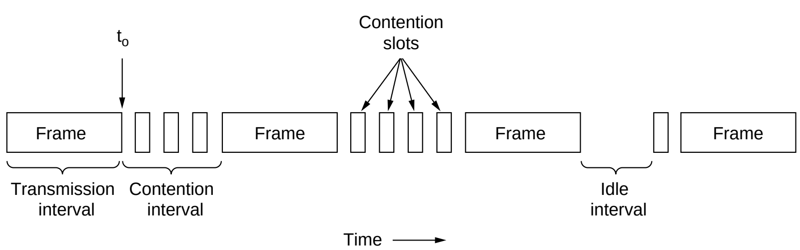
$$u = \frac{1}{a + \frac{a}{n}}$$





CSMA/CD Performance

- Assume time is divided into slots, where a slot is twice the propagation delay (max time to detect a collision)
- Assume n stations, where each station transmits with probability p for every slot



- There are two types of intervals to consider
  - Transmission, duration is $\frac{1}{2a}$ slots
  - Contention

- To compute contention interval, must determine the probability q that one station seizes the medium

$$q = \binom{n}{1} p(1-p)^{n-1} = n \cdot p(1-p)^{n-1}$$

where p is the probability a station transmits during a slot
Maximum occurs when $p = \frac{1}{n}$

$$q = \left(1 - \frac{1}{n}\right)^{n-1}$$

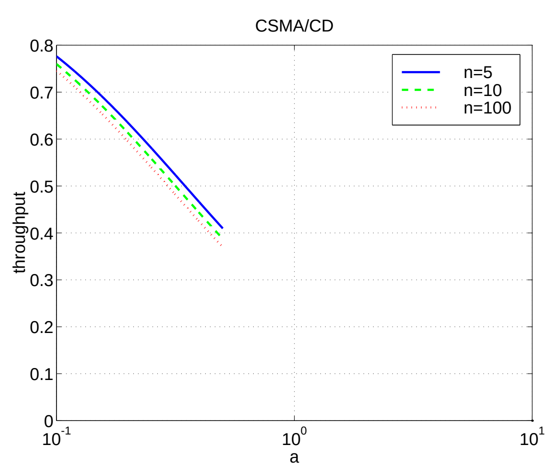
- The mean length of the contention interval w is

$$E[w] = \sum_{i=1}^{\infty} i \cdot p\{i \text{ slots in a row with collision followed by one success}\}$$

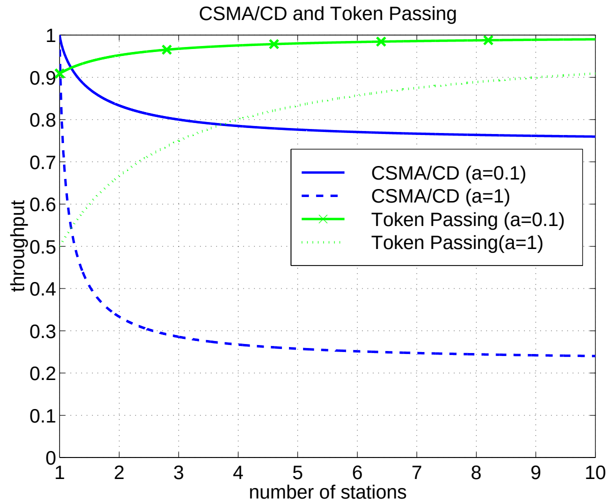
$$= \sum_{i=1}^{\infty} i \cdot q(1-q)^i = \frac{1-q}{q}$$

- CSMA/CD utilization is determined using the length of the transmission interval and the cycle

$$u = \frac{\frac{1}{2a}}{\frac{1}{2a} + \frac{1-q}{q}} = \frac{1}{1 + 2a\frac{1-q}{q}}$$



CSMA/CD and Token Passing Comparison



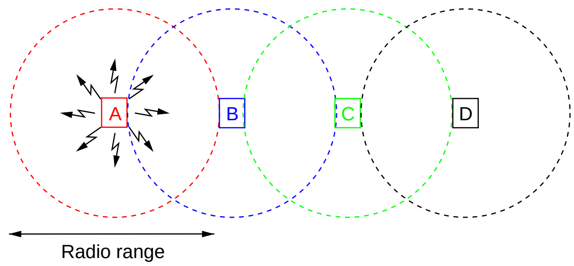
If token passing performs better than CSMA/CD in most cases, why is CSMA/CD more popular?

Wireless MAC

- A Wireless LAN is considered a type of broadcast network
  - If a station transmits, everyone with range can hear
  - A set of stations use the same *channel*
  - As a result, it is very similar to a bus topology
  - Stations share a medium $\rightarrow$ similar MAC problems right...
- You cannot use a CSMA type of MAC
  - CSMA considers interference only at the sender
  - With wireless, conditions at sender $\neq$ conditions at receiver

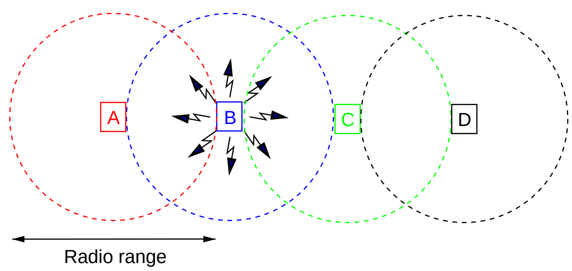
Wireless Station Problems

Assume in the following two diagrams, a station's range is one hop



station A transmitting

- **Hidden station problem** using CSMA
  - Assume station A is transmitting to B
  - If C senses the medium, it will not hear A
  - If C then transmits it will interfere with B
  - Frame transmitted by A is lost, but A would never sense it...



station B transmitting

- **Exposed station problem** using CSMA
  - Assume station B is transmitting to A
  - If C senses the medium and concludes it cannot send to D
  - However, it could send to D without interfering with A

    How is this possible?

- Therefore, before transmitting a station needs to know the activity around the receiver

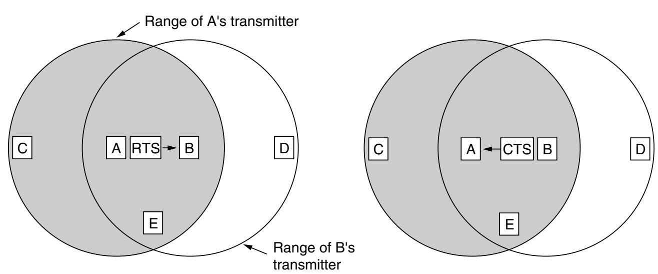
MACA

Multiple Access with Collision Avoidance (MACA) is a wireless MAC

- Assume station A wishes to send to station B
  - A starts with a Request To Send (RTS) message
  - B replies with a Clear To Send (CTS) message
  - If A receives a CTS from B, then it transmits the data
- Neighboring stations react as follows
  - Stations hearing the RTS wait long enough for the CTS
  - Stations hearing the CTS wait long enough for the data

What is long enough?

Does this solve the hidden and exposed station problems?



A sending an RTS to B   B responding with CTS to A

- Consider station C, within range of A but not B
  - C hears the RTS from A but not the CTS from B
  - C can transmit while the data is transmitted
- Consider station D, within range of B but not A
  - D does not hear the RTS from A, but hears the CTS from B
  - D waits to transmit after the data is sent

Can collisions still occur?