changelog

12 Nov 2024: add solution slide for ethernet propogation delay exercise

multiaccess media

shared air for radio/light signals

shared wires for electrical/light signals

...

needs:

- way to tell what signals are for whom
- way to decide who 'talks' when/where

shared wires

- used to be how Ethernet worked
  - before Ethernet switches were ubiquitous
- how cable Internet works
  - shared line to many customers in area
  - need to share
- usually how fiber-to-the-home works
  - "passive optical network"
  - connect multiple fibers optically

multiple channels

can have multiple channels over single medium

typically: radio or electrical signal or light frequencies

sender/receiver separate channels electrically/optically

to start: will worry about coordinating one chnanel

wireless spectrum

most useful radio spectrum *licensed*

gov't gives exclusive rights (within some region) to specific organizations/poeple

- example: cellular, TV, satellite, air traffic control, etc.

a lot of computer networking uses *unlicensed* bands

- use without specific permission allowed
- still limits on power, procedures to avoid interference

selected unlicensed bands

approx. frequencies unlicensed in US (not everywhere)

902–928 MHz (802.15.4 (IoT focused))

2.4–2.5 GHz (802.15.4; 802.11b/g/n/ax/...; bluetooth; microwave ovens)

5.15 GHz–5.25 GHz (802.11a/n/ac/ax/...)

5.25 GHz–5.73 GHz (802.11a/n/ac/ax/...; also weather radar)

requires 'dynamic frequency selection'

5.73 GHz–5.85 GHz (802.11a/n/ac/ax/...)

5.93 GHz–7.12 GHz (802.11ax/...)

collisions

N nodes try to transmit one channel at same time

likely outcomes for some receiver:

receiver gets garbage

"collision"

receiver receives 1 of the N collisions

running example

based on Abramson, "The Aloha System—Another alternative for computer comunications" (1970)

suppose we have shared radio with nodes A1, A2, ..., An and B

A1, A2, ...An are all trying to transmit to B

takes 1 ms to send message

and want to collectively send k messages per second

randomly spaced (exponential distribution)

some probability

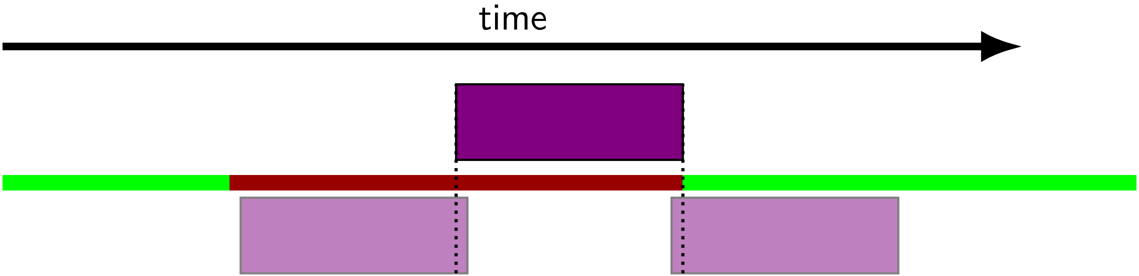
exponential distribution with mean λ

- our model for when packets sent
- "memoryless" distribution
- knowing when last packet sent tells you nothing about next
- (yes, not realistic)

probability events occur $< K$ time units apart

$1 - e^{-\lambda K}$

quiet time to avoid collisions



to avoid collision with 1 ms packet...

can't start packet less than 1 ms before

can't start packet less than 1 ms after

$\rightarrow$ need 2 ms without packet starting for no collision

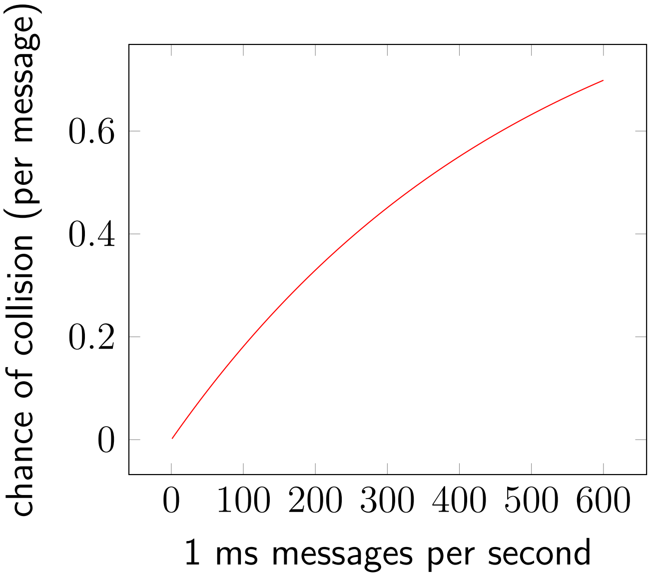
chance of collisions?(1)

to avoid collision when sending 1 ms packet

need no other packet to be sent in 2ms period around its start time

with k packets/sec, chance is approx $1 - e^{-\frac{2}{1000}k}$

chance of collision



retransmissions

what's going to happen when node can't send message

probably it will retransmit it...

which means real transmission rate will be some $R > k$

where k is rate messages are generated

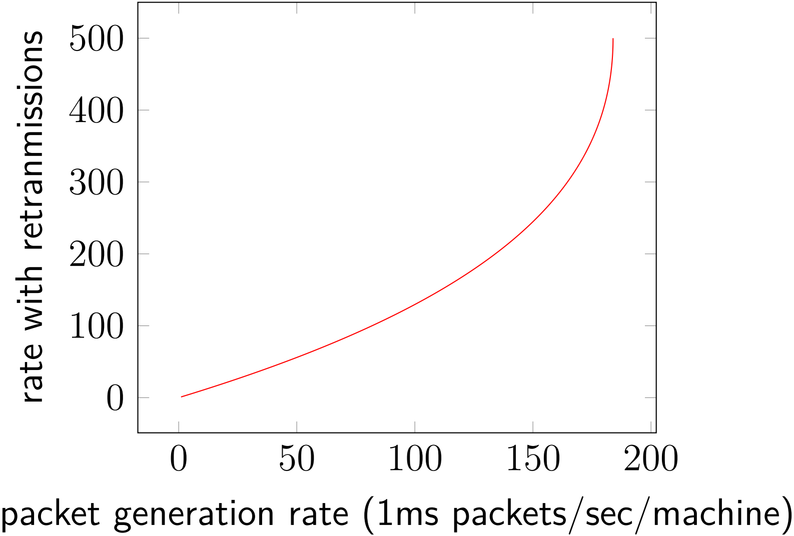
about $[1 - e^{-2R\frac{1}{1000}}]$ chance of each message generated

so $R = k + \left(1 - e^{-2R\frac{1}{1000}}\right) \cdot R$

$$R = k + R - Re^{-2R\frac{1}{1000}}$$

$$k = Re^{-2R\frac{1}{1000}}$$

retranmissions (plot)



thinking about result

sending 500 1ms packet or retransmission/second

using about half the capacity!

representing $\sim$ 186 1 ms non-retranmissions/second

$$\frac{1}{2e} = 0.186\ldots$$

using about 1/6th the capacity

results hold generally

seems pretty bad for shared channel efficiency!

carrier sense

channel can be ‘busy’ or not

radio/light:

- have some sort of signal detectable on frequency

“carrier sense”

way to detect whether channel busy

using carrier sense

simple idea: don't transmit if channel already busy

problem: then when do you transmit?

some options:

- never, lose the packet
- immediately when it stops being busy
- sometime after it stops being busy

'slots'

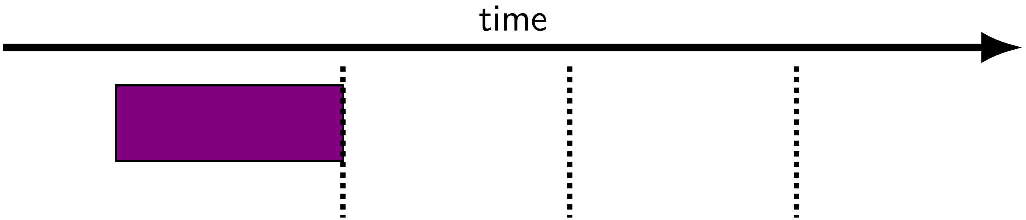
if packets take a fixed amount of time...

want to wait whole number of packet times

immediately after, one after, two after, three after

avoids requiring almost 2 packets worth of 'quiet' time

can synchronize by observing end of last transmission, or central clock



fixed-size frames, slots

'slotted' Aloha

challenge: synchronize everyone's timing

then: everyone chooses slots to (re)transmit in

but: not using carrier-sense before sending

if everyone sync'd, not useful with fixed-size packets

requires 1-packet-unit empty periods

36% utilization (twice for naive version)

802.11b typical slots

20μs slots

50μs interframe spacing (IFS)

includes time to receive ACK of packet

transmit first time after IFS (if idle)

transmit second time after IFS $+$ rand$(0,\ 2^5)$ slots

plus time spent by other seen transmissions

third time after rand$(0,\ 2^6)$ slots, etc.

802.11b packet lengths

variable bit rate: max = 11Mbit

approx 300–12000 bit frames
= 27 to 1090 microseconds

do carrier sense before transmitting always

different purpose for slots than slotted Aloha

collision detection

Wi-Fi: use ACKs to detect if transmission successful

problem: transmit whole packet, then know about collision

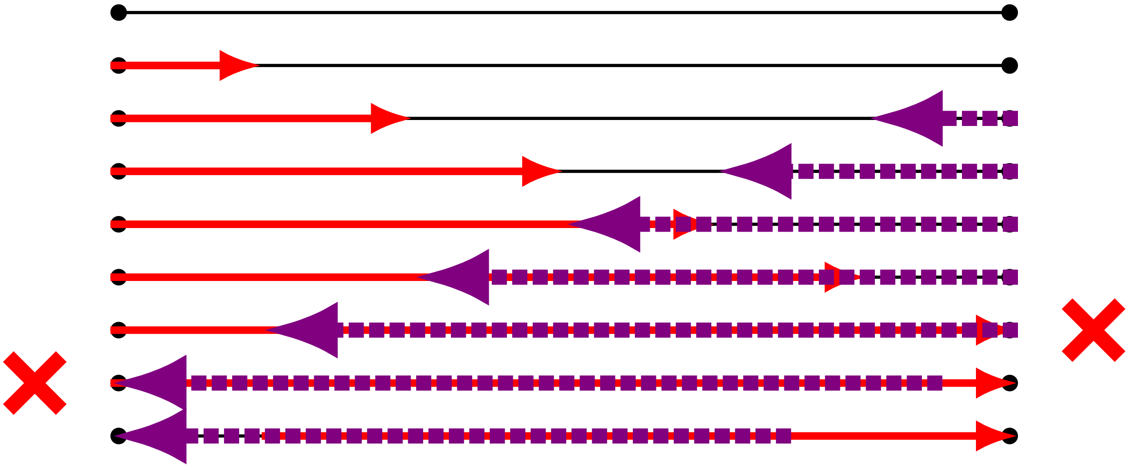
alternate idea: listen for collision, stop transmitting early

doesn't work on wireless (we'll talk about why later)

but does work on some wired shared media

part of reason for design of Ethernet header

collision detection takes time



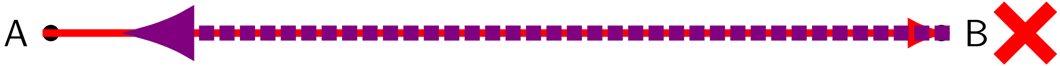
exercise: Ethernet cable length/delay

copper cable: about 2/3rds speed of light propogation

100Mbit ethernet has 64byte minimum frame size

exercise: maximum cable length with collision detection?

solution



worst case: A and B maximally far apart on network and collide

need each of them to detect collision and resend

if they're maximum distance, then anyone else will get interference

A transmits at time 0, takes X time units to reach B

B transmits at time $X - \epsilon$, then detects collision

need A to also detect collision before it finishes sending frame

maximum packet length is how much we can send in $2X$ time units

2X $=$ 64 times 8 $/$ 100Mbps $=$ 5.12 μ s; X $=$ 2.56 μs

propogation delay

transmission time for 64 bytes is 2.56 μs

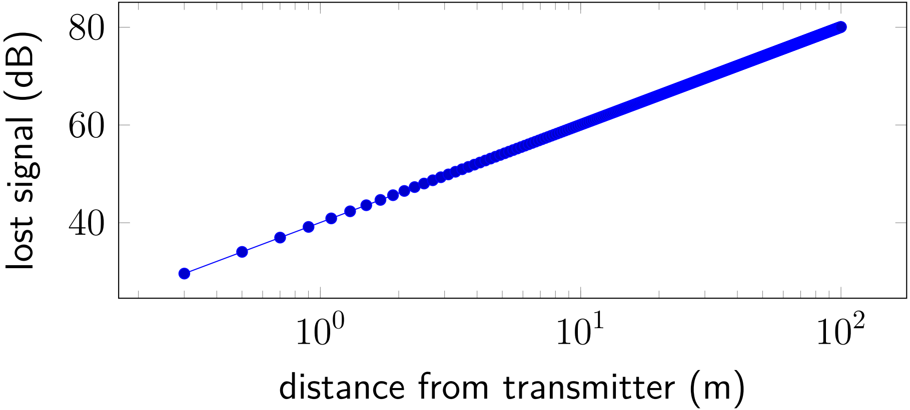
propogation delay: approx 0.2 meters/ns or 200 meters/μs

2.56 μs is about 512 meteres

free space wireless transmission

assuming 2.4GHz Wifi, no obstacles, omnidirectional transmission:

10 dB loss = $\frac{1}{10}$th power; 20 dB loss = $\frac{1}{100}$th power



signal to noise ratio

often measure power of signal versus power of 'noise'

simple model: noise 'noise floor' $+$ interfering transmission

theory (Shannon-Hartley): max bitrate $= B \log_2 \left(1 + \frac{S}{N}\right)$

B = bandwidth of channel

S, N = power per bandwidth of signal, noise (non-log scale)

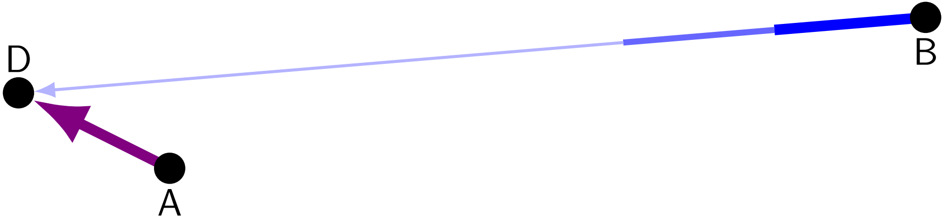
signal to noise ratio and bitrate

can recover from more intereference with lower bitrate

example: 802.11b supports 1, 2, 5.5, 11 Mbit/sec

most devices measure signal-to-noise ratio and set bitrate

receiving multiple nodes

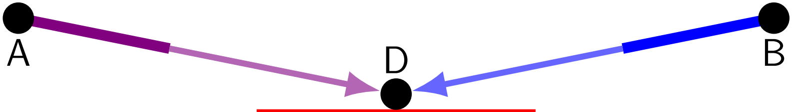


A's signal much much stronger at D

D may still receive A's signal...

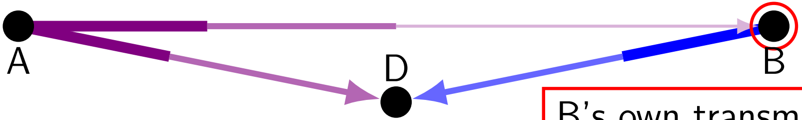
because B relatively too weak to interfere

collision (non-)detection



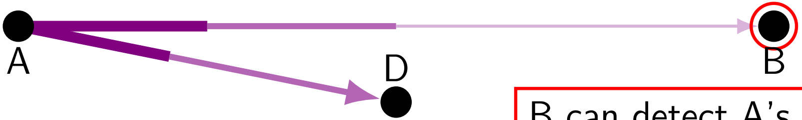
D receives junk

collision (non-)detection



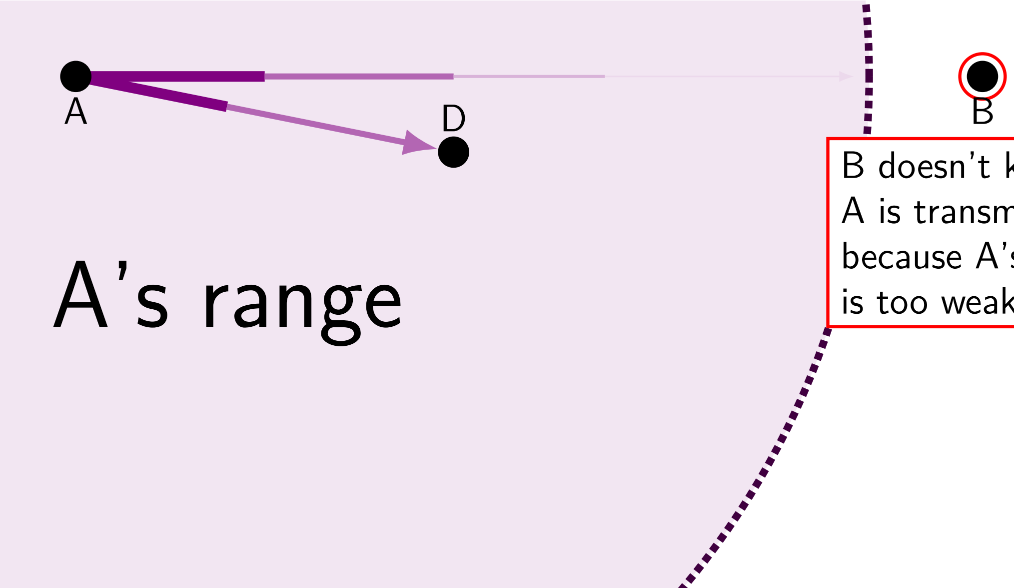
B's own transmission
is much stronger than A's
cannot detect collision itself

carrier-sense



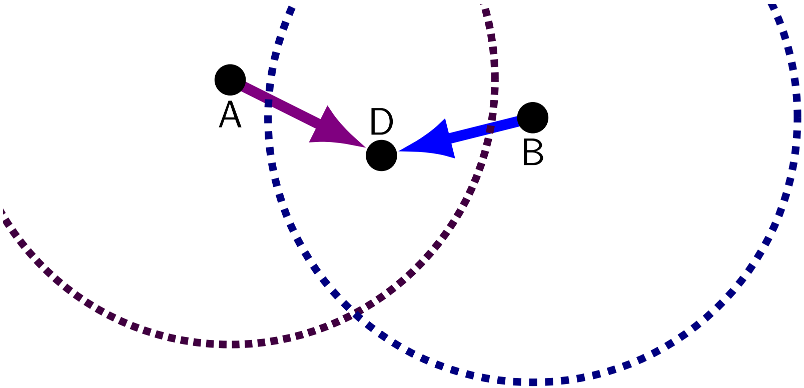
B can detect A's transmitting
avoid starting transmission now

carrier-not-sensing

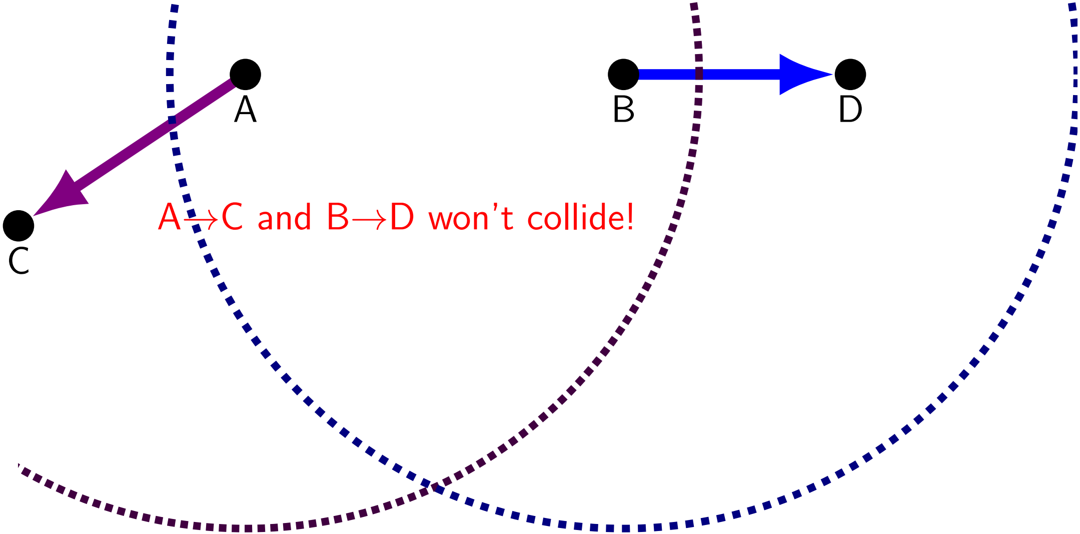


B doesn't know
A is transmitting
because A's signal
is too weak

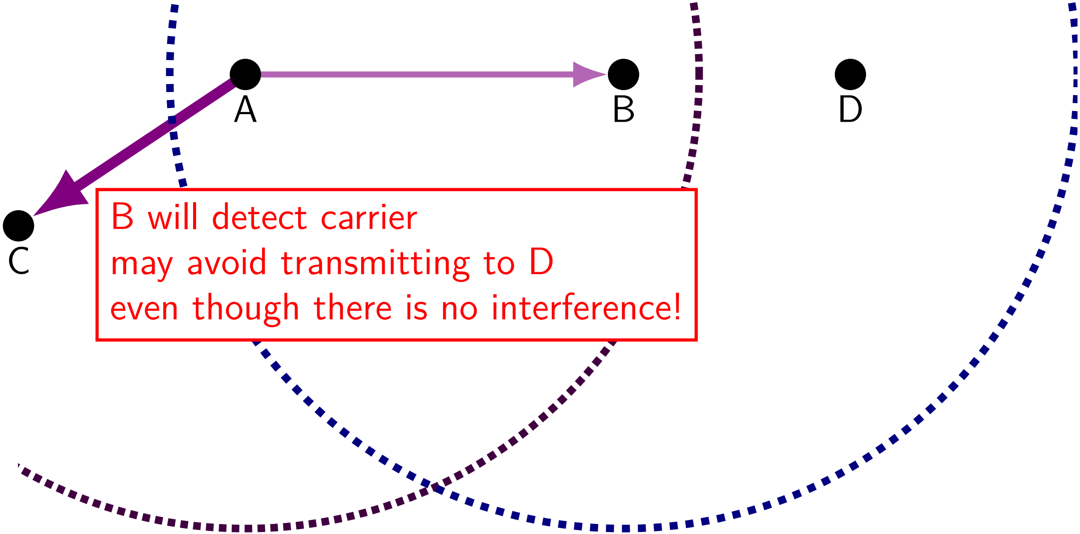
‘hidden node’ problem



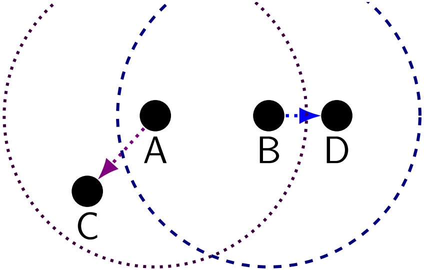
carrier false sensing



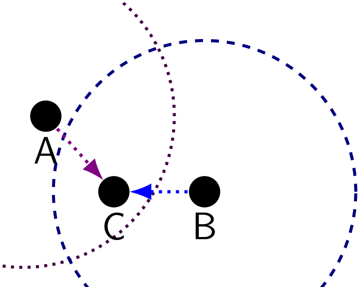
carrier false sensing



hidden/exposed node summary



exposed node problem: A and B detect each other's transmissions but don't interfere



hidden node problem: A and B interfere when sending to C but can't detect each other's transmissions

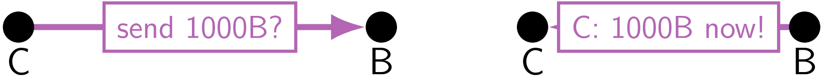
request-to-send/clear-to-send

can't detect likely interference at sender

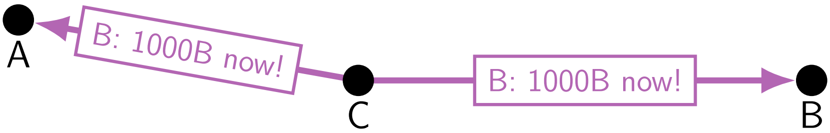
idea: have receiver tell when it's okay to transmit

sender→receiver: request-to-send

receiver→sender: clear-to-send



overhearing clear-to-send

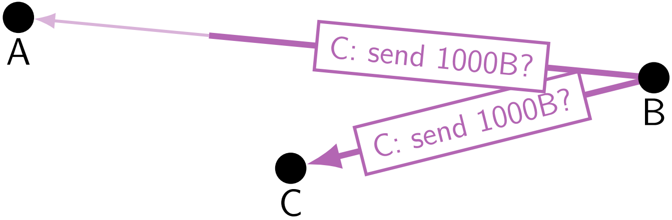


A can infer that B is transmitting to C

...and for how long!

A can avoid interfering, even if out of range of carrier-sense

overhearing ready-to-send



if A can hear B's request-to-send...

A should avoid sending until after B should get clear-to-send

exercise: request-to-send advantage

assume:

- 1μs request-to-send/clear-to-send messages
- 100μs messages sent
- chance of collision proportional to message lengths

exercise: effect of using RTS/CTS (versus only carrier-sense) for...

2 nodes transmitting to one with no hidden nodes?

2 nodes hidden from each other transmitting to one?

exercise: RTS/CTS suitablity

RTS/CTS is a better idea when...?

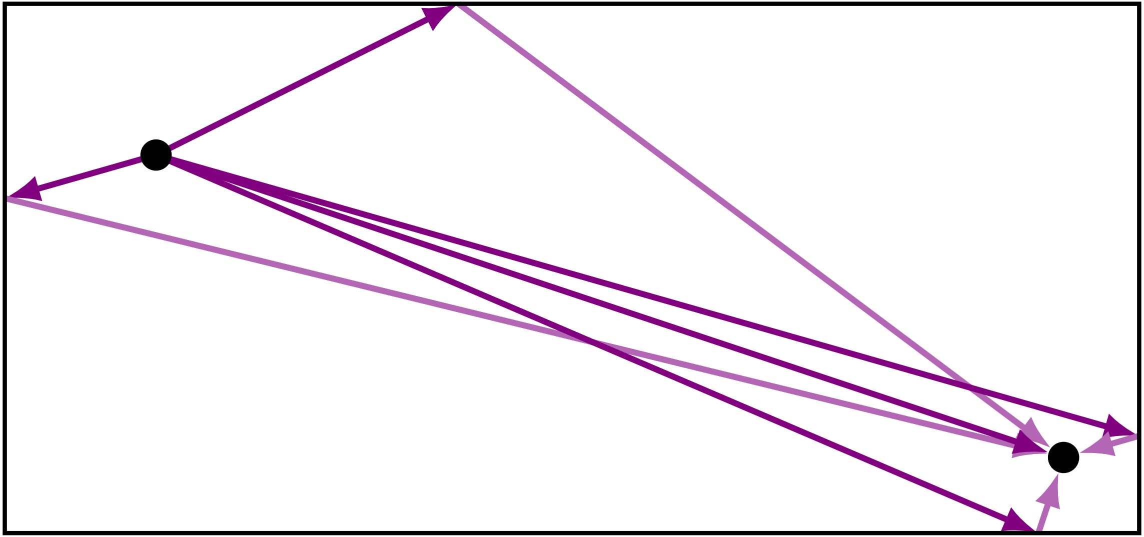
A. almost all messages are sent by one node instead of being split evenly

B. messages are shorter

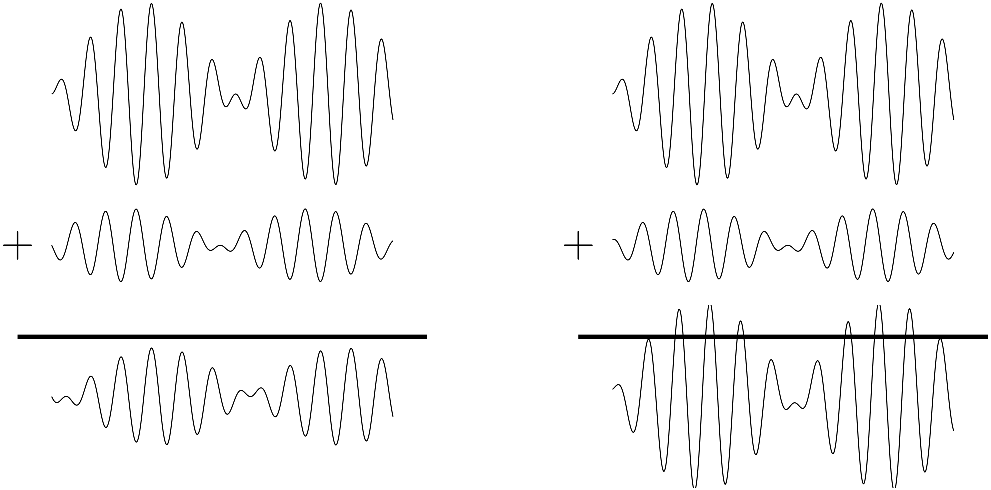
D. more nodes can hear others who are transmitting to different base stations

C. more nodes cannot hear others who are transmitting to the same base station

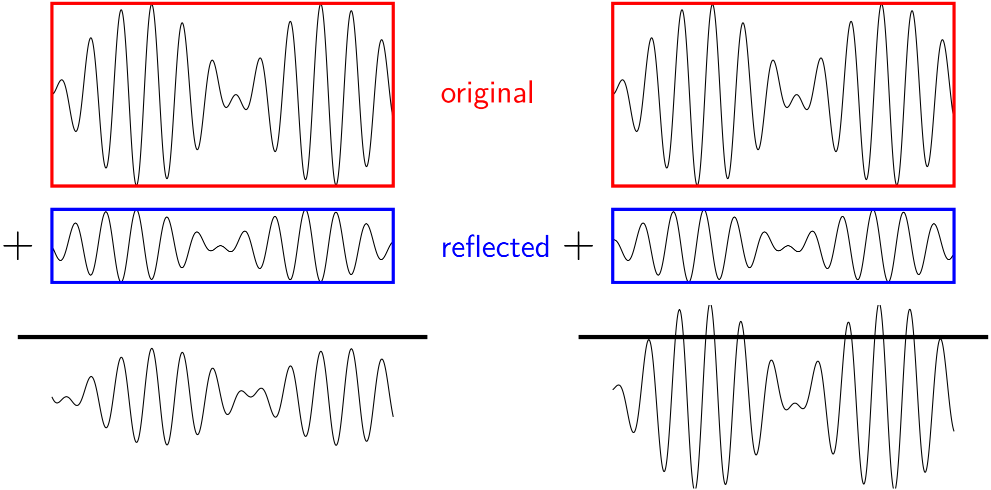
multipath interference



simple self-interference



simple self-interference



interference and wavelengths

radio wave frequency determines wavelength

- wifi: centimeters
- distance between ‘peaks’ in power
- half wavelength = distance between peak and trough

if distances make peaks align with troughs, lower power

if distances make peaks align with peaks, higher power

interference and symbols

separate from wavelength: how long are ‘symbols’

the things we turn into bits

for long paths/high symbol rate, might receive delayed signal overlaid on non-delayed

multipath interference

quite complicated to predict in three dimensions, variable reflectivity, ...

varies on frequency

- switching to close different channel may change a lot

can help and hurt signal reception

especially bad when 'inter-symbol interference'

- can be mitigated some by choice of message encoding

contributes to irregular 'dead zones', etc.

deliberate multipath / MIMO

can have:

multiple sending antennas (inputs) (MI)

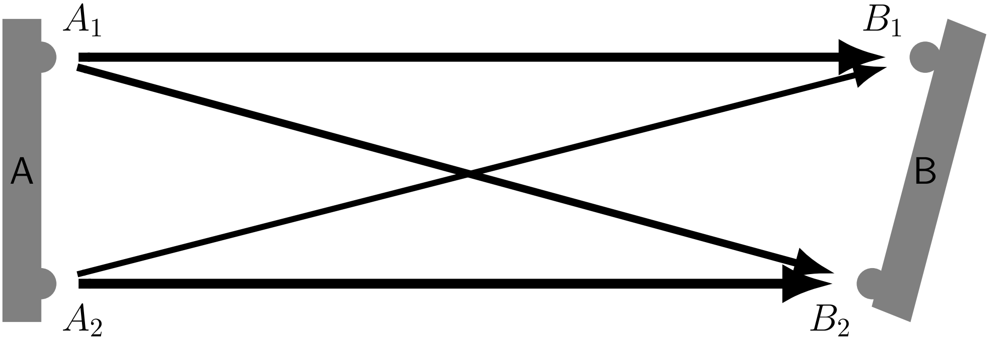
multiple receiving antennas (outputs) (MO)

some simple ideas:

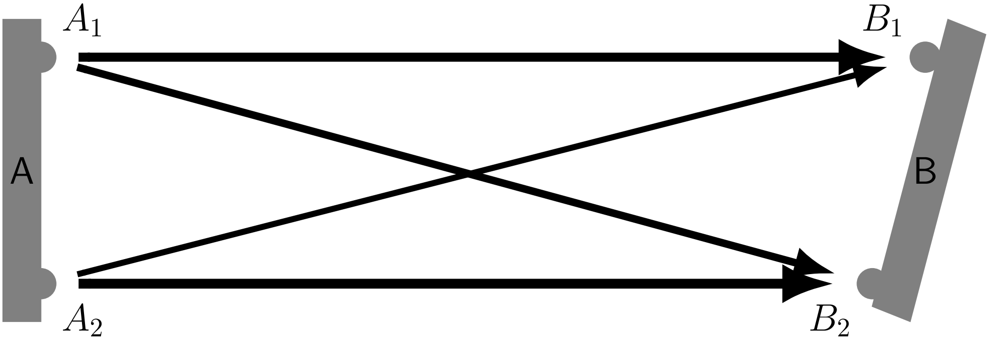
receive signal twice, hope one antenna not in dead zone

send signal twice, have receiver tell you which is better

multiple spatial streams



multiple spatial streams



signals might mix based on relative strengths, say

$$
\begin{aligned}
B_1 &\sim 0.9A_1 + 0.7A_2 \\
B_2 &\sim A_1 + 1.2A_2
\end{aligned}
$$

simplified multipath model

$$\begin{aligned} B_1 &\sim & 0.9A_1 + && 0.7A_2 \\ B_2 &\sim & A_1 + && 1.2A_2 \end{aligned}$$

if we can estimate coefficients...

B can solve for A_1, A_2

A can send with pattern to help estimate coefficients:

example: send full power from A_1 ($(B_1, B_2) = (0.9, 1)$), then from A_2 ($(B_1, B_2) = (0.7, 1.2)$)

correlation problem

when two 'spatial streams' correlated, harder to solve for them

example: suppose $(B_1, B_2) = (A_1 + A_2, A_1 + 1.1A_2)$

$A_2 = 10B_1 - 10B_2$

need 10x precision in B_1, B_2 to get 1x precision in A_2

assignment

upcoming wireless assignment

one receiver, many senders

free space model (no multipath)

can only change senders:
- get channel sense/ACK info
- choose channel
- choose timing of transmissions

wifi modes

“ad-hoc”: everyone sends/receives from whoever

“infrastructure”: designated *access points*

mostly use infrastructure networks

infrastructure mode

networks identified by SSID (Service Set IDentifier)

human readable name

single SSID can have many access points (example: eduroam)

clients *associated* with one access point at a time

clients only *only* send to their current access point

...even if sending to other nodes on network

exercise: why through AP?

case where going through AP helps performance?

case where going through AP hurts performance?

beacons (approx. once per 100ms)

```
▸ Frame 1074: 433 bytes on wire (3464 bits), 433 bytes captured (3464 bits) on interface mon0, id 0
▸ Radiotap Header v0, Length 54
▸ 802.11 radio information
▾ IEEE 802.11 Beacon frame, Flags: ........
    Type/Subtype: Beacon frame (0x0008)
  ▸ Frame Control Field: 0x8000
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: ff:ff:ff:ff:ff:ff
    Destination address: ff:ff:ff:ff:ff:ff
    Transmitter address: d8:07:b6:d9:[redacted]
    Source address: d8:07:b6:d9:[redacted]
    BSS Id: d8:07:b6:d9:[redacted]
    .... .... .... 0000 = Fragment number: 0
    1010 0000 0000 .... = Sequence number: 2560
    [WLAN Flags: ........]
▾ IEEE 802.11 Wireless Management
  ▸ Fixed parameters (12 bytes)
  ▾ Tagged parameters (343 bytes)
    ▸ Tag: SSID parameter set: "[redacted]"
    ▸ Tag: Supported Rates 6(B), 9, 12(B), 18, 24(B), 36, 48, 54, [Mbit/sec]
    ▸ Tag: DS Parameter set: Current Channel: 157
    ▸ Tag: Traffic Indication Map (TIM): DTIM 0 of 1 bitmap
    ▸ Tag: Country Information: Country Code US, Environment All
    ▸ Tag: RSN Information
    ▸ Tag: Vendor Specific: Microsoft Corp.: WPA Information Element
    ▸ Tag: RM Enabled Capabilities (5 octets)
    ▸ Tag: Supported Operating Classes
    ▸ Tag: HT Capabilities (802.11n D1.10)
    ▸ Tag: HT Information (802.11n D1.10)
    ▸ Tag: Extended Capabilities (6 octets)
    ▸ Tag: Interworking
    ▸ Tag: Advertisement Protocol
    ▸ Tag: VHT Capabilities
    ▸ Tag: VHT Operation
    ▸ Tag: Tx Power Envelope
    ▸ Ext Tag: HE Capabilities
    ▸ Ext Tag: HE Operation
    ▸ Ext Tag: MU EDCA Parameter Set
    ▸ Ext Tag: Spatial Reuse Parameter Set
    ▸ Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element
    ▸ Tag: Vendor Specific: Microsoft Corp.: WPS
    ▸ Tag: Vendor Specific: Wi-Fi Alliance: Multi Band Operation - Optimized Connectivity Experience
    ▸ Tag: Vendor Specific: Metalink LTD.
```

probes/probe responses

beacons — listen for a few seconds, find out about nearby networks

probably need to scan multiple channels

can also send explicit probes to learn about networks

receive responses, with essentially same kind of information

association

client → AP: association response (SSID=...)

AP → client: association response (SSID=...)

\+ things related to WiFi Security (encryption, passwords, etc.)

client → AP: deassociation (sometimes)

moving between APs

multiple APs can broadcast have same SSID

generally: nodes should listen for beacons, measure signal-to-noise ratio

eventually decide to change association

wifi data frames

```
▸ Frame 2389: 233 bytes on wire (1864 bits), 233 bytes captured (1864 bits) or
▸ Radiotap Header v0, Length 68
▸ 802.11 radio information
▾ IEEE 802.11 QoS Data, Flags: .p....F.
     Type/Subtype: QoS Data (0x0028)
   ▸ Frame Control Field: 0x8842
     .111 1110 1100 1111 = Duration: 32463 microseconds
     Receiver address: a0:b3:39:43:e8:fb
     Transmitter address: d8:07:b6:d9:ae:4e
     Destination address: a0:b3:39:43:e8:fb
     Source address: d8:07:b6:d9:ae:50
     BSS Id: d8:07:b6:d9:ae:4e
     STA address: a0:b3:39:43:e8:fb
     .... .... .... 0000 = Fragment number: 0
     0001 1110 1111 .... = Sequence number: 495
     [WLAN Flags: .p....F.]
   ▸ Qos Control: 0x0000
   ▸ CCMP parameters
▸ Data (131 bytes)
```

wifi data

A ⟶ AP1 ⟶ AP2 ⟶ B

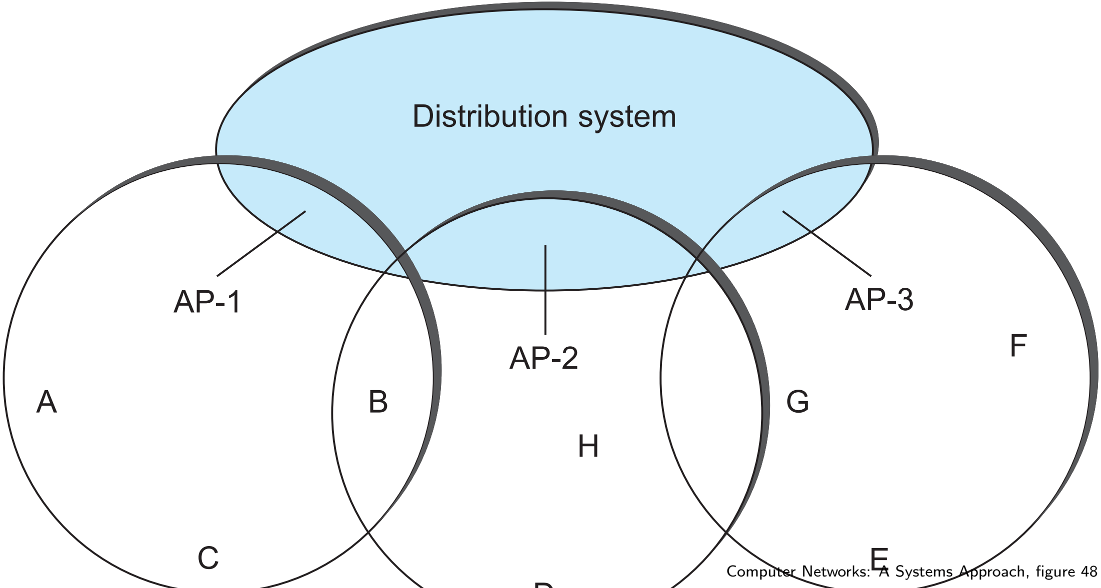
receiver/transmitter address — this hop

source/destination address — final soruce/destination

can differ if...

- destination is a broadcast/multicast address
- wireless equivalent of ‘switching’

multiple APs, one network



Computer Networks: A Systems Approach, figure 48

APs as switches

multiple APs can act as *one network*

example: nodes connected to different APs can send packets to each other by MAC address

distribution system needs to share neighbor-table-like information

switching APs

nodes can switch APs...

check for new APs when signal strength too low

periodically check for beacons + signal strength

switching APs = same as joining network, but...

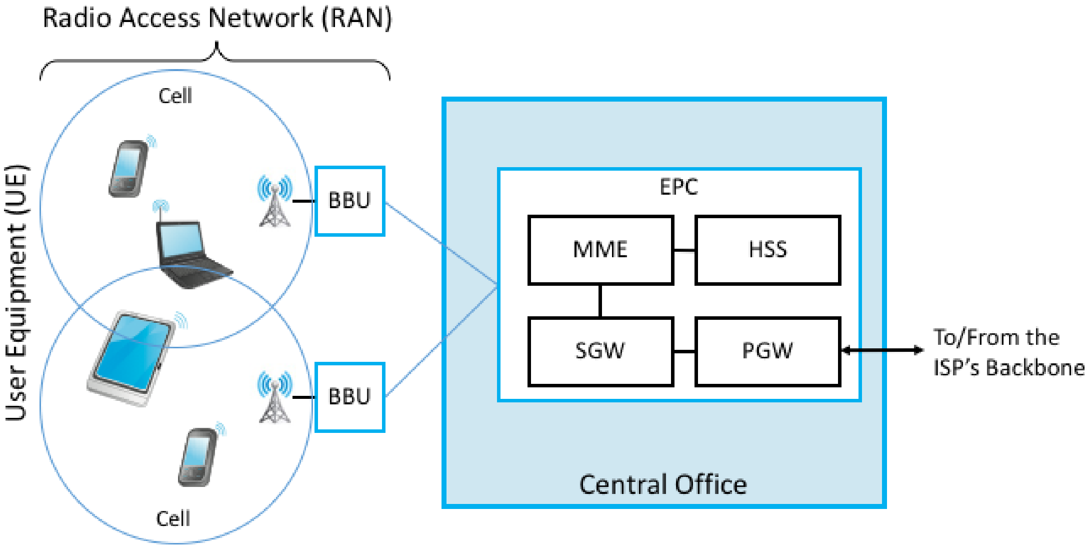
- can keep IP address, etc.
- maybe optimizations to avoid redoing wifi security, etc.

cellular networks

for wifi networks, feasible to track device locations centrally mostly

need/have something more complex for cellular networks

a confusing picture



cellular mobility model

cellular networks: base stations don't know how to route to each end-host in whole cell network

cell phone (UE) → base station (BBU) → service gateway (SGW) → PDN gateway (PGW)

central 'mobility management entity' (MME) sets up 'tunnels' for steps above

- coordinates with home subscriber server (HSS)
- base stations don't track full routing information

PDN gateway stays stable so IP address can stay the same

- PDN gateway = gateway to actual Internet

Roofnet (Cambridge, MA, 2005)

Bicket, Aguayo, Biswas, Morris, “Architecture and Evaluation of an Unplanned 802.11b Mesh Network (MobiCom’05)

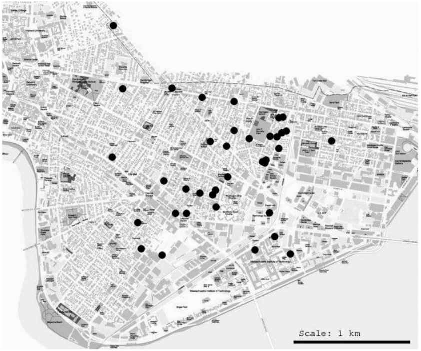


Figure 1: A map of Roofnet. Each dot represents a wireless node. The map covers the south-east portion of Cambridge, Massachusetts. The Charles River curves along the lower boundary of the map, MIT is at the lower right, and Harvard is at the upper left.

roofnet routing

up to five hops for packet to get from one node to another

link-state routing protocol between nodes

- no explicit configuration needed

sending nodes computed list of hops + included in packet

- idea called “source routing”
- prevents transient routing loops
- important because conditions (e.g. weather) changes connectivity

everyone using same channel!

link-state/distance vector for wireless

no explicit list of links/networks

instead: periodically broadcast and see who responds

keep track of signal strength/reliabliy/etc. for routing metrics

modern mesh networks

common for distribution network between APs to be (partly) wireless

sysadmin view:
- plug some APs into internet connection
- put other APs in appropriate place
- APs figure out how to make it work

typically using self-organizing mesh network ideas
- likely similar routing protocols to what we discussed

usually properietary networking protocols
- vendor lock-in problem
- (though there is now a recent Wifi standard)

wifi polling mode

wifi has rarely used "polling" mode

access point tells everyone when to transmit/not transmit

periodically 'poll' stations for more traffic

can be mixed with periods allowing normal CSMA/CA

carrier sense multiple access with collision avoidance

exercise: pro/con

deliberate scheduling

so far:

- devices decide when to try to send or reply to requests
- one channel supporting one transmitter/receiver

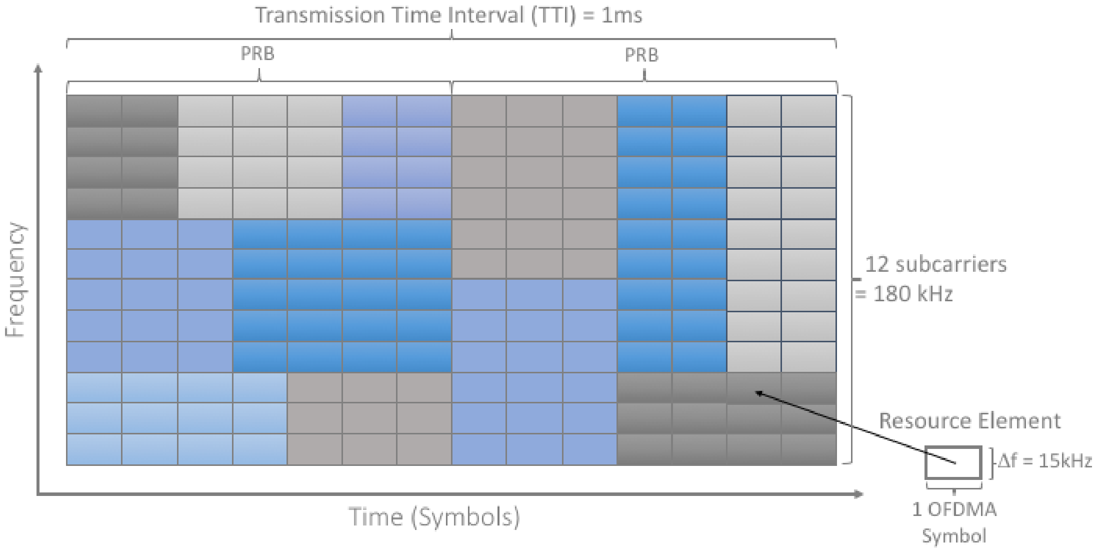
alternate idea (cellular networks): divide

several available channels

- typically different frequencies
- signal encoding chosen to avoid interference between close frequencies

slots within each time interval

Computer Networks: A Systems Approach, Figure 54



central scheduler idea

'base stations' track active user devices

users have bandwidth reservations

base stations send out schedule of slots every $\sim$ 0.1–1ms

protocol for handoff between base stations

devices send back quality feedback to aid scheduling

dealing with propagation delay

schedule of slots will also have time delays

goal: compensate for propogation delay

base station needs to track estimate of propogation delay

dealing with new nodes

what about nodes without a reservation?

mobile base stations advertise a 'random access channel'

used for reserving extra resources primarily

use more wifi-like contention here

exercise

central scheduling v carrier-sense

exercise: which handles better...

- utilizing the most of the available bandwidth
- independently controlled access points/base stations
- communications between two nearby nodes

backup slides