P561: Network Systems
Week 2: Local Area Networks

Tom Anderson
Ratul Mahajan

TA: Colin Dixon

Administrivia

Fishnet Assignment #1
- Due next week (week 3), start of class
- Electronic turnin
- No class trawler (do that for Fishnet #2)

Homework #1
- On web site
- Due two weeks (week 4), start of class

Next week: Internetworking, broadcast from MSR

Q&A from last time

How far can an optical link go without a repeater?
- About 20 km in practice
- 10 terabits/100 km in prototypes

Why do they call it MIMO beamforming?
- Can independently control the phase and amplitude of each antenna, which affects the receiver power.

Network Building Blocks

Links – carry information (bits)
- Wire, optics or wireless
- Point to point or broadcast

Switches/Routers -- move bits between links
- packet or circuit switching

Host – communication endpoint
- computer, PDA, toaster, …

Network -- delivers messages between hosts over a collection of links and switches

---

Internet Design Goals

Effective multiplexing of existing networks
- multiplexing = sharing
- using store & forward packet switching

Survivability in the face of failure
- Communication must continue despite loss of equipment

Heterogeneity
- *In networks and applications*

Distributed management

Network Sharing

Networks are shared among users
- This is an important benefit of building them

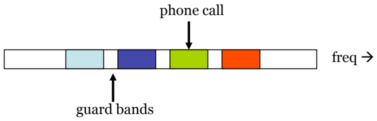
Problem: How to multiplex (share) a link among multiple users?

Well, we could statically partition the link:
- Frequency Division Multiplexing (FDM)
- (Synchronous) Time Division Multiplexing (TDM, STDM)

Frequency Division Multiplexing

Simultaneous transmission in different frequency bands

- Analog: Radio/TV, AMPS cell phones (800MHz)
- Also called Wavelength DM (WDM) for fiber

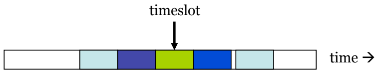


"Speaking at different pitches"

Time Division Multiplexing

Timeslice given frequency band between users

- Digital: used extensively inside the telephone network
- T1 (1.5Mbps) is 24 x 8 bits/125us; also E1 (2Mbps, 32 slots)



"Speaking at different times"

Advantage: lower delay

Disadvantage: synchronization

Statistical Multiplexing

Static partitioning schemes are not suited to data communications because peak rate >> average rate.

If we share on demand we can support more users

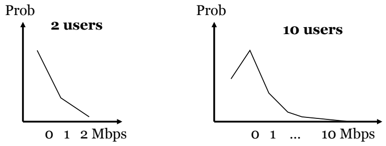
- Based on the statistics of their transmissions
- Occasionally we might be oversubscribed
- This is called <u>statistical multiplexing</u>

Statistical multiplexing is heavily used in data networks

Example

One user sends at 1 Mbps and is idle 90% of the time.

- 10 Mbps channel; 10 users if statically allocated



What are the likely loads if we share on demand?

Example continued

For 10 users, Prob(need 10 Mbps) = 10^{-10}

Not likely! So keep adding users …

For 35 users, Prob(>10 active users) = 0.17%, which is acceptably low

We can support three times as many users!

But: there is an important caveat here …

Workload Questions

How bursty is the data traffic to/from a single node?

- Self-similar at many time scales

How bursty is the data traffic to/from a campus?

How bursty is the data traffic in the core of the Internet?

- Elephants and mice

ALOHA

Packet radio network in Hawaii, 1970s

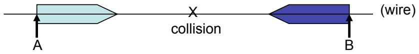
Wanted distributed allocation
- no special channels or single point of failure

Aloha protocol:
- Just send when you have data!
- There will be some collisions of course …
- Throw away garbled frames at receiver (using CRC); sender will time out and retransmit

Simple, decentralized and works well for low load
- What happens when load increases?

Carrier Sense Multiple Access

We can do better by listening before we send (CSMA)
- good defense against collisions if “a” is small



“a”: number of packets that fit on the wire
- bandwidth * delay / packet size
- Small for LANs; large for satellite

What if the Channel is Busy?

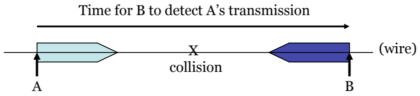
1-persistent CSMA
- Wait until idle then go for it
- Blocked senders can queue up and collide

p-persistent CSMA
- If idle send with prob p in each time slot until done
- Choose p so p * # senders < 1; how do you know p?

non-persistent CSMA
- Wait a random time and try again
- Better when loaded, but larger delay when unloaded

CSMA with Collision Detection

Even with CSMA there can still be collisions. Why?

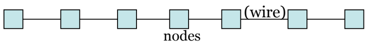


For wired media we can detect all collisions and abort (CSMA/CD):
- Requires a minimum frame size (“acquiring the medium”)
- B must continue sending (“jam”) until A detects collision

Classic Ethernet

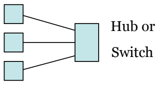
IEEE 802.3 standard wired LAN (1-persistent CSMA/CD)

Classic Ethernet: 10 Mbps over coaxial cable
- baseband signals, Manchester encoding, preamble, 32 bit CRC



Newer versions are much faster
- Fast (100 Mbps), 1 Gb, 10Gb

Modern equipment isn’t one long wire
- hubs and switches



Ethernet Frames (Classic)

Preamble (8)	Source (6)	Dest (6)	Len (2)	Payload (var)	Pad (var)	CRC (4)

Min frame 64 bytes, max 1500 bytes

CSMA/CD jam period is 48 bits

Max length 2.5km, max between stations 500m (repeaters)

Addresses unique per adaptor; globally assigned

Broadcast media:
- ARP, multicast, promiscuous mode monitoring

Binary Exponential Backoff

Build on 1-persistent CSMA/CD

On collision: jam and exponential backoff

- Jamming: send 48 bit sequence to ensure collision detection

Backoff:

- First collision: wait 0 or 1 frame times at random and retry
- Second time: wait 0, 1, 2, or 3 frame times
- Nth time (N<=10): wait 0, 1, …, 2^N-1 times
- Max wait 1023 frames, give up after 16 attempts
- Scheme balances average wait with load

Ethernet Capture

Randomized access scheme is not fair

Stations A and B always have data to send

- They will collide at some time
- Suppose A wins and sends, while B backs off
- Next time they collide and B's chances of winning are halved!

Ethernet Performance

Much better than Aloha or CSMA!

- Works very well in practice

Source of protocol inefficiency: collisions

- More efficient to send larger frames
  - Acquire the medium and send lots of data
- Less efficient as the network grows in terms of frames
  - recall "a" = delay / (frame size * transmission rate)
  - "a" grows as the path gets longer (satellite)
  - "a" grows as the bit rates increase (Fast, Gigabit Ethernet)

Why Did Ethernet Win?

Reliablity

- Token ring failure mode -- network unusable
- Ethernet failure mode -- node detached

Cost

- Passive tap cheaper to build than active forwarder
- Volume => lower cost => volume => lower cost …

Scalability

- Repeater: copy all packets across two segments
- Bridge: selectively repeat packets across two segs
- Switch: bridge k segments; Hub: repeater for k segs

Switched Ethernet

Build larger networks out of small building blocks

Redundancy for higher availability

Simple case: # of nodes < degree of switch

Scaling

What if # of nodes > degree of one switch?

- What does a data center network look like?

Fat Trees

Bisection bandwidth: the minimum bandwidth between any equal partitioning of the nodes

- Important if network communication is all to all

Internet PoPs

PoP = Point of Presence

- Use redundancy at each level to mask failures

Internet PoPs

PoP = Point of Presence

- Use redundancy at each level to mask failures

Inside a Switch

If switch degree is small enough, use a crossbar

- Need buffering at the inputs
- Performance degrades (badly) with head of line blocking

Inside a Switch

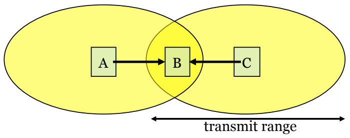
What if you want to build a wider switch?

Wireless Communication

Wireless is more complicated than wired …

1. Cannot detect collisions
   - Transmitter swamps co-located receiver
2. Different transmitters have different coverage areas
   - Asymmetries lead to hidden/exposed terminal problems

Hidden Terminals

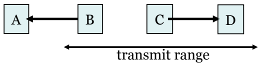


A and C can both send to B but can’t hear each other

- A is a hidden terminal for C and vice versa

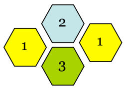
CSMA will be ineffective – want to sense at receiver

Exposed Terminals



B, C can hear each other but can safely send to A, D

Compare to spatial reuse in cell phones:



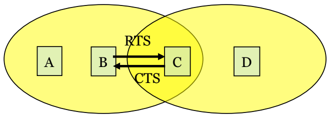
CSMA with Collision Avoidance

Since we can’t detect collisions, we avoid them
- CSMA/CA as opposed to CSMA/CD
- Not greedy like Ethernet

When medium busy, choose random backoff interval
- Wait for that many idle timeslots to pass before sending
- Remember p-persistence … a refinement

When a collision is inferred, retransmit with binary exponential backoff (like Ethernet)
- Use CRC and ACK from receiver to infer “no collision”
- Again, exponential backoff helps us adapt “p” as needed

RTS / CTS Protocols (MACA)



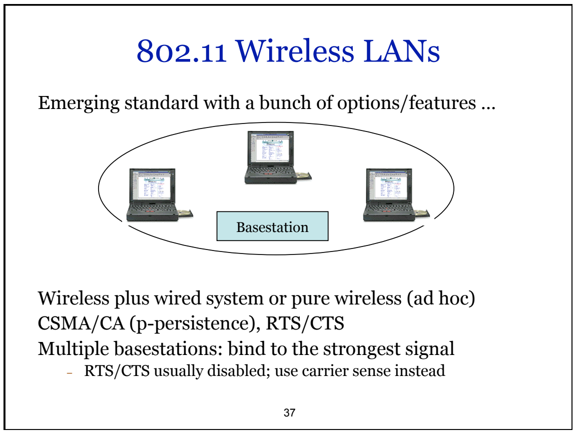
1. B asks C with Request To Send (RTS)
2. A hears RTS and defers to allow the CTS
3. C replies to B with Clear To Send (CTS)
4. D hears CTS and defers to allow the data
5. B sends to C

MACA Example

MACAW paper

Propose several additions to RTS/CTS:
- Link layer ACK
- Data size header
- Request to request to send
- Various backoff changes
  - Share backoff value
  - “MILD” backoff instead of binary
  - Per destination backoff

Goals were efficiency and fairness
- Did they succeed?



802.11 Wireless LANs

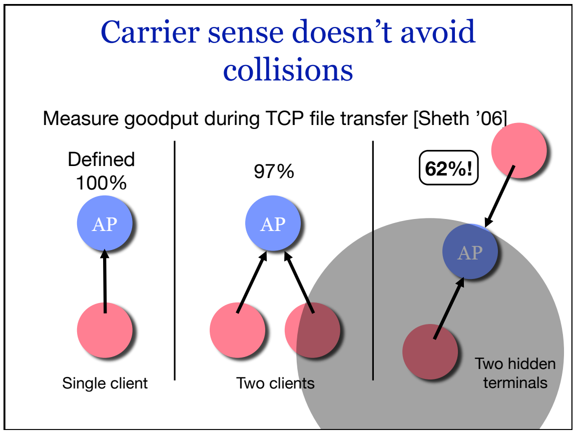
Emerging standard with a bunch of options/features …

Wireless plus wired system or pure wireless (ad hoc)

CSMA/CA (p-persistence), RTS/CTS

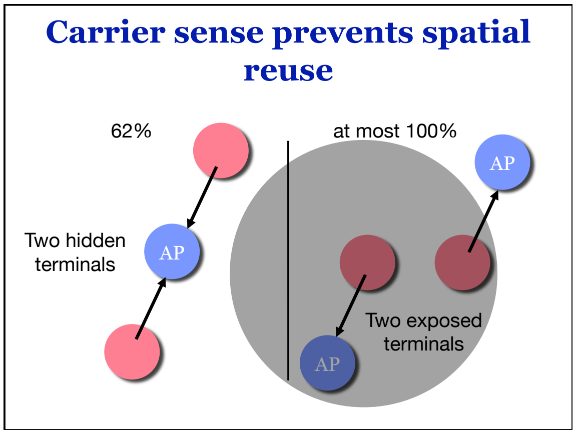
Multiple basestations: bind to the strongest signal

- RTS/CTS usually disabled; use carrier sense instead

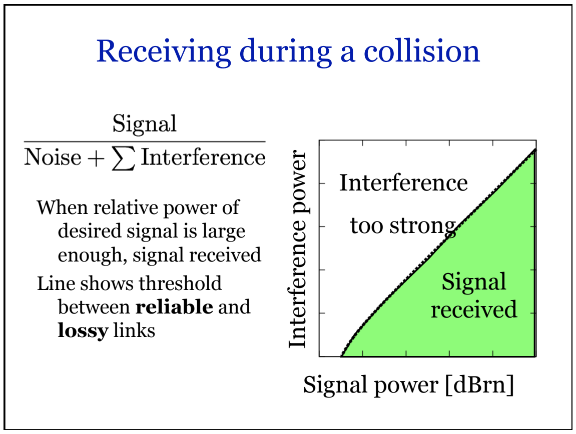


Carrier sense doesn't avoid collisions

Measure goodput during TCP file transfer [Sheth '06]



Carrier sense prevents spatial reuse



Receiving during a collision

$$\frac{\text{Signal}}{\text{Noise} + \sum \text{Interference}}$$

When relative power of desired signal is large enough, signal received

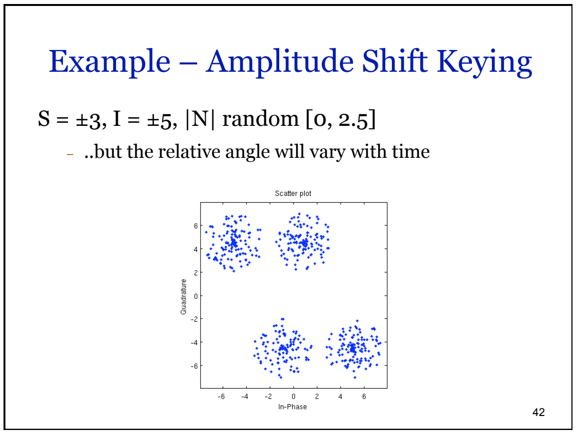
Line shows threshold between **reliable** and **lossy** links

The SINR model is overly simplistic

SINR treats interference and noise as **equally** detrimental

Noise is **random**, but interference has **structure** intended to communicate data

Key idea: exploit structure of interference to overcome its effects



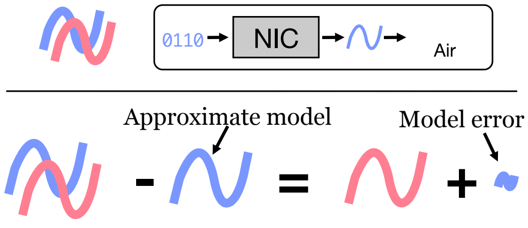
Example – Amplitude Shift Keying

S = ±3, I = ±5, |N| random [0, 2.5]

- ..but the relative angle will vary with time

How interference cancellation works

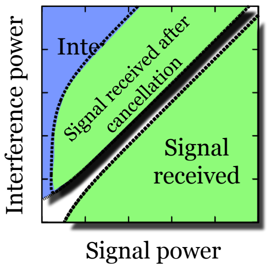
$$\text{Received} = \text{Noise} + \sum \text{Distorted Signals}$$



Implementing interference cancellation

Successive interference cancellation (SIC)

Strong interferer decoded, modeled, and then canceled



Interference cancellation for IEEE 802.15.4

Physical layer for 2.4 GHz ZigBee stack

- Low power, low rate wireless networking using O-QPSK with 8x direct sequence spread spectrum
- Similar to slowest rates of WiFi and good for SIC
- 2M chips/s and 2.5 MHz spectral mask

- **Real PHY** that fits well with USRP limitations

How to model an interfering signal?

Key step in interference cancellation is approximating and subtracting interference

Any **error** in the model increases the noise floor and makes post-cancellation **performance worse**

Model specific environment features - simple but limited

Channel filter computation is complex and misses non-linearities

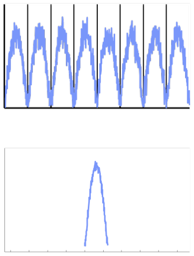
Data-dependent models by averaging

Symbols blended in time by filters and channel; received at time *i* depends on ...,*i*-*1*, *i*, *i*+*1*,...

Build an RF template for each bit pattern by averaging received waveforms

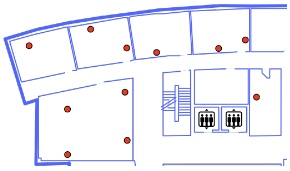
Uncorrelated noise, interference will average out

We use 3 consecutive symbols (6 bits; 64 models)



Experimental setup

Deployed an 11-node wireless testbed in UW CSE



Non-LOS, co-channel WiFi APs, varying environment

Connectivity between nodes ranges from perfect communication to completely hidden

Experimental methodology

Implemented three ZigBee receivers

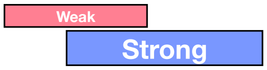
- Two conventional single-packet receivers
- Successive interference cancellation

Generate random two-packet collisions for all pairs of senders while logging digitized, raw RF at other 9 nodes

Replay logs to each receiver to allow direct comparison

Baseline receiver implementation

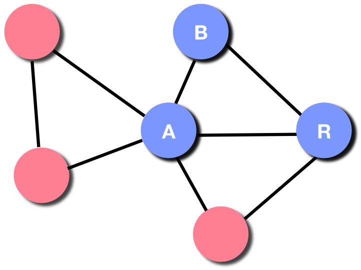
By locking onto one transmission, a receiver can miss a second, stronger packet



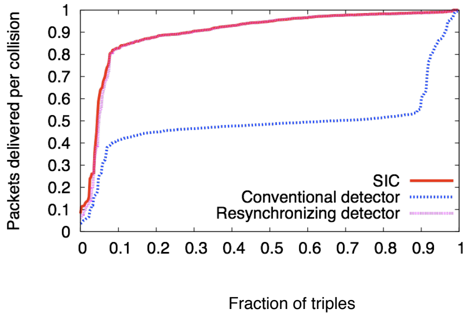
Observed commercial hardware of both types

We compare successive cancellation against both

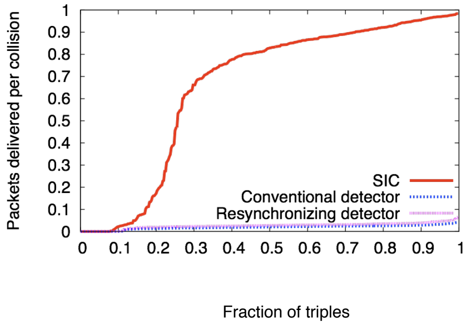
Experiment analysis



Stronger transmitter



For the weaker transmitter



Delivery vs. Interference

SINR [dB] ≅ Signal power [dB] - Interference power [dB]