CSCI-1680
Wireless

Rodrigo Fonseca

Based partly on lecture notes by Scott Shenker and John Jannotti
• TCP is due on Monday, Nov 26\textsuperscript{th}, 11:59pm
Wireless

• Today: wireless networking truly ubiquitous
  – 802.11, 3G, (4G), WiMAX, Bluetooth, RFID, …
  – Sensor networks, Internet of Things
  – Some new computers have no wired networking
  – 4B cellphone subscribers vs. 1B computers

• What’s behind the scenes?
Wireless is different

• Signals sent by the sender don’t always reach the receiver intact
  – Varies with space: attenuation, multipath
  – Varies with time: conditions change, interference, mobility

• Distributed: sender doesn’t know what happens at receiver

• Wireless medium is inherently shared
  – No easy way out with switches
Implications

• Different mechanisms needed
• Physical layer
  – Different knobs: antennas, transmission power, encodings
• Link Layer
  – Distributed medium access protocols
  – Topology awareness
• Network, Transport Layers
  – Routing, forwarding
• Most advances *do not* abstract away the physical and link layers
Physical Layer

• **Specifies physical medium**
  - Ethernet: Category 5 cable, 8 wires, twisted pair, R45 jack
  - WiFi wireless: 2.4GHz

• **Specifies the signal**
  - 100BASE-TX: NRZI + MLT-3 encoding
  - 802.11b: binary and quadrature phase shift keying (BPSK/QPSK)

• **Specifies the bits**
  - 100BASE-TX: 4B5B encoding
  - 802.11b @ 1-2Mbps: Barker code (1bit -> 11chips)
What can happen to signals?

• **Attenuation**
  - Signal power attenuates by $\sim r^2$ factor for omni-directional antennas in free-space
  - Exponent depends on type and placement of antennas
    - $< 2$ for directional antennas
    - $> 2$ if antennas are close to the ground
Interference

- **External sources**
  - E.g., 2.4GHz unlicensed ISM band
  - 802.11
  - 802.15.4 (ZigBee), 802.15.1 (Bluetooth)
  - 2.4GHz phones
  - Microwave ovens

- **Internal sources**
  - Nodes in the same network/protocol can (and do) interfere

- **Multipath**
  - Self-interference (destructive)
Multipath

- May cause attenuation, destructive interference
Signal (+ Interference) to Noise Ratio

• **Remember Shannon?**

• **Shannon-Hartley**

  \[ C = 2B \log_2(M) \text{ bits/sec} \quad (1) \]

• But noise ruins your party

  \[ C = B \log_2(1 + S/N) \text{ bits/sec} \quad (2) \]

  \[(1) \leq (2) \Rightarrow M \leq \sqrt{1 + S/N} \]

• Noise limits your ability to distinguish levels
  – For a fixed modulation, increases Bit Error Rate (BER)

• Could make signal stronger
  – Uses more energy
  – Increases interference to other nodes
Wireless Modulation/Encoding

- More complex than wired
- Modulation, Encoding, Frequency
  - Frequency: number of symbols per second
  - Modulation: number of chips per symbol
    - E.g., different phase, frequency, amplitude
  - Encoding: number of chips per bit (to counter errors)
- Example
  - 802.11b, 1Mspo: 11Mcps, DBPSK, Barker Code
    - 1 chip per symbol, 11 chips/bit
  - 802.11b, 2Mspo: 22Mcps, DQPSK, Barker Code
    - 2 chips per symbol, 11 chips/bit
Link Layer

• Medium Access Control
  – Should give 100% if one user
  – Should be efficient and fair if more users

• Ethernet uses CSMA/CD
  – Can we use CD here?

• No! Collision happens at the receiver

• Protocols try to *avoid* collision in the first place
Hidden Terminals

- A can hear B and C
- B and C can’t hear each other
- They both interfere at A
- B is a *hidden terminal* to C, and vice-versa
- Carrier sense at sender is useless
Exposed Terminals

- A transmits to B
- C hears the transmission, backs off, even though D would hear C
- C is an *exposed* terminal to A’s transmission
- Why is it still useful for C to do CS?
Key points

• No global view of collision
  – Different receivers hear different senders
  – Different senders reach different receivers

• Collisions happen at the receiver

• Goals of a MAC protocol
  – Detect if receiver can hear sender
  – Tell senders who might interfere with receiver to shut up
Simple MAC: CSMA/CA

• Maintain a waiting counter $c$
• For each time channel is free, $c$--
• Transmit when $c = 0$
• When a collision is inferred, retransmit with exponential backoff
  – Use lack of ACK from receiver to infer collision
  – Collisions are expensive: only full packet transmissions
• How would we get ACKs if we didn’t do carrier sense?
RTS/CTS

• **Idea:** transmitter can check availability of channel at receiver

• **Before every transmission**
  – Sender sends an RTS (Request-to-Send)
  – Contains length of data (in *time* units)
  – Receiver sends a CTS (Clear-to-Send)
  – Sender sends data
  – Receiver sends ACK after transmission

• **If you don’t hear a CTS, assume collision**

• **If you hear a CTS for someone else, shut up**
RTS/CTS
RTS/CTS
RTS/CTS
Benefits of RTS/CTS

• Solves hidden terminal problem
• Does it?
  – Control frames can still collide
  – E.g., can cause CTS to be lost
  – In practice: reduces hidden terminal problem on data packets
Drawbacks of RTS/CTS

- **Overhead is too large for small packets**
  - 3 packets per packet: RTS/CTS/Data (4-22% for 802.11b)
- **RTS still goes through CSMA: can be lost**
- **CTS loss causes lengthy retries**
- **33% of IP packets are TCP ACKs**
- **In practice, WiFi doesn’t use RTS/CTS**
Other MAC Strategies

• **Time Division Multiplexing (TDMA)**
  – Central controller allocates a time slot for each sender
  – May be inefficient when not everyone sending

• **Frequency Division**
  – Multiplexing two networks on same space
  – Nodes with two radios (think graph coloring)
  – Different frequency for upload and download
ISM Band Channels

79 Channels

BlueTooth

Channel

802.15.4

802.11b

Channel 1

Channel 6

Channel 11

2400 MHz

2425 MHz

2450 MHz

2475 MHz

2480 MHz

22 MHz

25 MHz

3 MHz

5 MHz
Network Layer

• What about the network topology?
• Almost everything you use is *single hop*!
  – 802.11 in infrastructure mode
  – Bluetooth
  – Cellular networks
  – WiMax (Some 4G networks)

• Why?
  – Really hard to make multihop wireless efficient
WiFi Distribution System

• **802.11 typically works in *infrastructure mode***
  – Access points – fixed nodes on wired network

• **Distribution system connects Aps**
  – Typically connect to the same Ethernet, use learning bridge to route to nodes’ MAC addresses

• **Association**
  – Node negotiates with AP to get access
  – Security negotiated as well (WEP, WPA, etc)
  – Passive or active
Wireless Multi-Hop Networks

• Some networks are multihop, though!
  – Ad-hoc networks for emergency areas
  – Vehicular Networks
  – Sensor Networks
    • E.g., infrastructure monitoring
  – Multihop networking to share Internet access
    • E.g. Meraki
Many Challenges

• Routing
  – Link estimation

• Multihop throughput dropoff
The Routing Problem

- Find a route from S to D
- Topology can be very dynamic
Routing

• Routing in ad-hoc networks has had a lot of research
  – General problem: any-to-any routing
  – Simplified versions: any-to-one (base station), one-to-any (dissemination)

• DV too brittle: inconsistencies can cause loops

• DSDV
  – Destination Sequenced Distance Vector
DSDV

• Charles Perkins (1994)
• Avoid loops by using sequence numbers
  – Each destination increments own sequence number
    • Only use EVEN numbers
  – A node selects a new parent if
    • Newer sequence number or
    • Same sequence number and better route
  – If disconnected, a node increments destination sequence number to next ODD number!
  – No loops (only transient loops)
  – Slow: on some changes, need to wait for root
Many Others

- DSR, AODV: on-demand
- Geographic routing: use nodes’ physical location and do greedy routing
- Virtual coordinates: derive coordinates from topology, use greedy routing
- Tree-based routing with on-demand shortcuts
- ...
Routing Metrics

• How to choose between routes?
• Hopcount is a poor metric!
  – Paths with few hops may use long, marginal links
  – Must find a balance
• All links do *local retransmissions*
• Idea: use expected transmissions over a link as its cost!
  – $\text{ETX} = 1/(\text{PRR})$ (Packet Reception Rate)
  – Variation: ETT, takes data rate into account
Multihop Throughput

• Only every third node can transmit!
  – Assuming a node can talk to its immediate neighbors
  – (1) Nodes can’t send and receive at the same time
  – (2) Third hop transmission prevents second hop from receiving
  – (3) Worse if you are doing link-local ACKs

• In TCP, problem is worse as data and ACK packets contend for the channel!

• Not to mention multiple crossing flows!
Sometimes you can’t (or shouldn’t) hide that you are on wireless!

- Three examples of relaxing the layering abstraction
Examples of Breaking Abstractions

• TCP over wireless
  – Packet losses have a strong impact on TCP performance
  – Snoop TCP: hide retransmissions from TCP end-points
  – Distinguish congestion from wireless losses
4B Link Estimator

- Uses information from Physical, Routing, and Forwarding layers to help estimate link quality

\[
ETX = \frac{d_f \times d_r}{1}
\]

Measuring \(d_f\) and \(d_r\) with infrequent beacons can be highly inaccurate.

Lines show percentile errors.

Directly measuring ETX with the data path reduces path costs by 45%.

This requires a routing protocol that can adapt to such rapid edge cost changes.

\[
ETX_t = \alpha \cdot ETX_{t-1} + (1 - \alpha)E_t
\]

\(E_t\) = \(\left\{ \begin{array}{ll}
\text{acked} > 0 & \\
\text{acked} = 0 &
\end{array} \right.\)

4-Bit estimator every 5 packets.
Stanford’s Full Duplex Wireless

• Status quo: nodes can’t transmit and receive at the same time
  – Why? TX energy much stronger than RX energy

• Key insight:

• With other tricks, 92% of optimal bandwidth
Summary

• **Wireless presents many challenges**
  – Across all layers
  – Encoding/Modulation (we’re doing pretty well here)
  – Distributed multiple access problem
  – Multihop

• **Most current protocols sufficient, given over provisioning** *(good enough syndrome)*

• **Other challenges**
  – Smooth handoff between technologies (3G, Wifi, 4G…)
  – Low-cost, long range wireless for developing regions
  – Energy usage
Coming Up

• Next time: security
• Final project out today

• Have a good break!