ECE 435 – Network Engineering
Lecture 22

Vince Weaver

https://web.eece.maine.edu/~vweaver

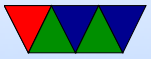
vincent.weaver@maine.edu

12:30pm Barrows 125

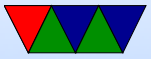
9 April 2024

Announcements

- Project status reports due next Friday
  - See pdf on website for full details
  - One sentence description of project topic
  - Briefly describe any progress made
  - Say whether you want to present the 18th, 23rd or 25th
- HW#10 will be posted

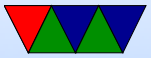


Wireless



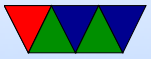
Why Wireless?

- Pros
  - Use anywhere
  - No wires
- Cons
  - Less reliability, noise
  - Less power availability
  - Less security



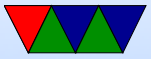
Wireless LAN

- 802.11. Started in 1990, no standard until 1997
- Operates in fixed ISM bands
  - Industrial/Scientific/Medical, no license needed
  - 900MHz, 2.4GHz, 5GHz
  - What issues come up with these bands?
    Microwave oven? Cordless phones, Bluetooth
  - Until 2002 ISM usage had to be spread spectrum
  - Up to 1W transmission power (50mW typical)



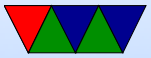
Wireless LAN Standards

- All of the various 802.11 have been sort of merged together, but people use the old letters out of habit



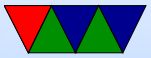
Wi-Fi

- What does Wi-Fi mean? Wireless Fidelity?
  Or Empty Marketing Term?
- Retroactively numbers given to older protocols
  - 802.11 = Wi-Fi 0
  - 802.11b = Wi-Fi 1
  - 802.11a = Wi-Fi 2
  - 802.11g = Wi-Fi 3
  - 802.11n = Wi-Fi 4
  - 802.11ac = Wi-Fi 5
  - 802.11ax = Wi-Fi 6/6E
  - 802.11be = Wi-Fi 7



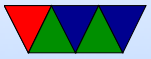
802.11 (1997) (Wi-Fi 0)

- Original, 1 or 2MBps, 2.4GHz, three implementations
  - infrared(?)
  - direct-sequence spread spectrum (DSSS)
    Takes a signal and spreads it along a wider frequency band but adding pseudo-random noise, then subtracting out at the other side.
  - frequency-hopping spread spectrum (FHSS)
    rapidly switch signal among a bunch of different frequencies in a pseudo-random fashion. Harder to



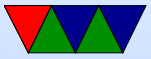
jam, causes less interference?

Initial seed, dwell time



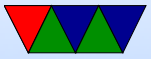
802.11b (1999) (Wi-Fi 1)

- 5.5Mbps and 11Mbps
- HR-DSSS (High Rate Direct Sequence Spread Spectrum)
- Walsh-Hadamard codes (error correction)
- actually came to market before 802.11a
- In the 2.4GHz frequency band, no licensing
- Various channels, 22MHz wide. Not all available in all countries. Some channels overlap.
- In the US have channels 1 through 11, but 1, 6, 11 are only non-overlapping ones



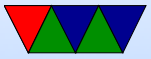
802.11a (1999) (Wi-Fi 2)

- 1.5 - 54Mbps
- Not compatible with B, 54Mbps in 5GHz band
- 5GHz less crowded, but signal doesn't go as far (7x less than b)
- OFDM (Orthogonal Frequency Division Multiplexing) Data is sent on multiple channels in parallel
- 52 channels: 48 data channels 4 pilot subcarriers



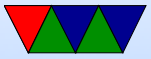
802.11g (2003) (Wi-Fi 3)

- 54Mbps, 2.4GHz
- Uses OFDM like 802.11a, but in the 2.4GHz band
- Backward compatible with b, which slows it down



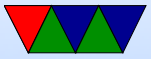
802.11n (2009) (Wi-Fi 4)

- 54Mbps - 600Mbps
- MIMO (multiple input/multiple output antennas)
- Can do spatial multiplexing, two antennas broadcast on same frequency by aiming signal



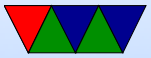
802.11ac (2009) (Wi-Fi 5)

- Most common currently (2022?)
- Wider channels, 80MHz-160MHz (vs 40MHz)
- 256 Quadrature Amplitude Modulation (QAM)
- MU-MIMO (multi-user MIMO)
- Theoretical max speed 7Gbps
- Up to 8 MIMO streams (spatial?)
- Downlink multi-user MIMO (4 clients), with 4 antennas
- Beam forming

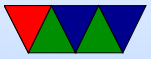


802.11ax (2019) (Wi-Fi 6)

- 2.4/5/6GHz
- Up to 11Gbps? (1Gbps more typical)
- high efficiency?
- Wi-Fi 6E
  - Extension that uses the “6 GHz” band 5.925 - 7.125GHz
  - 1.2GHz of spectrum (old one only 400MHz)
  - Indoors this is fine, stopped by walls
  - Outdoors might conflict with other users of the band,

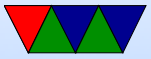


so has to do automatic frequency co-ordination where it checks database before using frequency



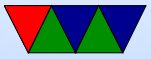
802.11be (20??) (Wi-Fi 7)

- To be released soon(?)
- 2.4/5/6GHz
- Up to 40Gbps?
- 4096-QAM



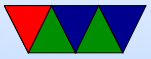
More obscure 802.11 variants

- 802.11ad – 7Gbps, 60GHz freq (frequency that high short distance, limited to inside room) "WiGig"
- 802.11af – white wi-fi, super wi-fi, operates in vacant UHF/VHF TV bands. Receiver uses GPS to find out where it is and what channels are free
- Many more



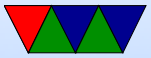
Terminology

- Station = device on wireless network
- Access Point (AP)



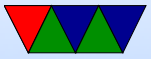
Wireless Network Topology

- Ad-hoc mode – peer to peer
- Distribution / Infrastructure mode – many to access point (AP) which has a wired connection
- In infrastructure mode all access goes through the AP



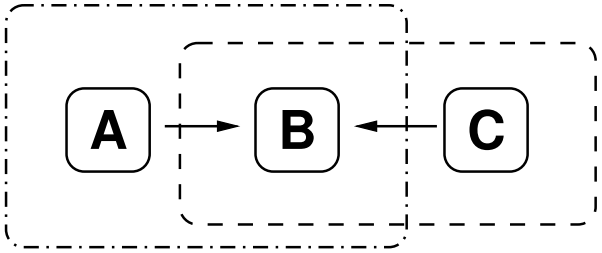
Service Sets

- A basic service set (BSS) is a group of nodes that all recognize each other
- An extended service set (ESS) is a group of overlapping BSSes with APs that are connected together
- An AP keeps the BSSes in line by periodically transmitting beacon frames

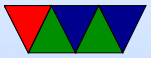


802.11 – Why not just Ethernet over the Air

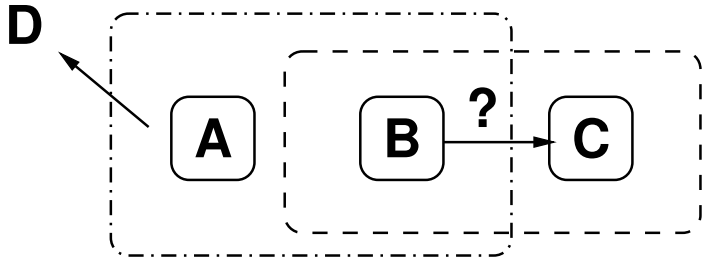
- Hidden station/terminal problem A in range of B, B in range of C, but A cannot see C. If A and C transmit at same time, they'll not get collision, only way of knowing is if not get ACK.



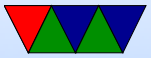
- Exposed station problem. A and C not overlap, but B



does not know this so it sees A transmitted to D and doesn't transmit to B even though it wouldn't cause collision.

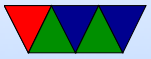


- To deal with this, Distributed Coordination Function (DCF) and point coordination function (PCF)



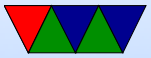
First some Timing Notes

- Network Allocation Vector (NAV), send along estimated time for how long things will take, other stations see this
- Interframe Spacing, 4 types
  - Short (SIFS)
  - PCF (PIFS)
  - DCF (DIFS)
  - EIFS



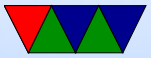
Also Note on Transmitter

- Half-duplex (full more expensive)
- Multiple frequency? Does each need own antenna?



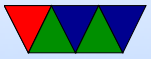
Notes on BSSID

- Each AP can handle more than one network group (i.e. guest, tempest, etc)
- BSSID (basic service set identifier)
- Each BSSID has 48-bit MAC. Randomly generated, with "local" bit set
- AP supposed to filter on this so only frames destined to correct BSS handled properly. Book stresses not everyone does this right.

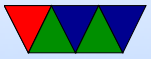


DCF – Distributed Coordination Function

- No central control
- Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)
- Different from Ethernet CSMA/CD (D=detection)
- Every time ready to transmit, looks to see if can transmit (listen to see if channel clear)
- If clear, waits DIFS (inter frame) waits random time (to avoid two waiters starting simultaneously, to try to pre-emptively avoid collisions), then transmits

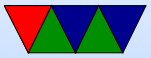


- If busy, waits until clear. Then it will wait a random backoff time before starting. Why? Multiple transmitters might have all been waiting and they would all instantly collide once clear.
- There is a short inter-frame interval (SIFS) which gives time for receiver to transmit an ACK packet.
- If source does not get an ACK, then it backs off and retries
- DCF not optimal, can take 60us to transmit ACK, whereas a 54MB connection could have send 3k of data in same time.



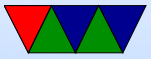
More on ACKs

- ACKs on unicast only, not sent on multicast/broadcast (so those are more unreliable)
- ACK, CTS, and fragments can send during SIFS



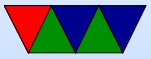
DCF: Fragmentation

- In some situations can fragment frames into smaller parts
- This is completely separate from IP fragmentation
- Why do it? – the longer the frame, the more likely it is to lose bits to interference. So split things up into smaller chunks likely to get through
- Fragment burst



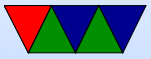
DCF: Error Handling

- Resend if error
- How detect error. No ACK?
- Short retry counter and long retry counter
- Backoff
  - Number of slots, based on how many retries
  - Each station randomly picks one of slots
  - If fails again, backs off and increases the slots

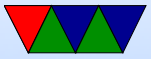


DCF: RTS/CTS Mode (used for "large" frames)

- Optional (rarely used) RTS/CTS mode
  - Before sending data, sends short RTS (request to send) packet
  - Receiver responds with short CTS (clear to send)
  - Data only sent if CTS sent properly
  - All stations can see both CTS *and* RTS, this and hopefully avoids collisions.
  - There's a duration field that hints how long it will take

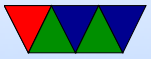


- ACK at end

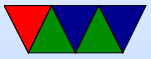


DCF: PCF – Point Coordination Function

- Also rarely used (mostly between infrastructure)
- PCF provides central control. A point coordinator in the AP periodically transmits a beacon to announce a contention-free period (CFP). Stations keep quiet.
- Sort of like time-division multiplexing
- Guaranteed a certain fraction of bandwidth
- For power saving, base station can tell receiver to go to sleep, and buffer packets for it until wakes up
- Can combine PCF and DCF in same cell.

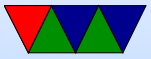


- Problem can happen if two different APs in range, in this case PCF won't be able to help collision problem if it's the other AP causing it



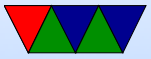
Does my router use PCF or DCF

- It appears that most use DCF, PCF is somewhat uncommon
- There is 802.11e which enhances this to Enhanced distributed channel access (EDCA)
- Introduces HCF (Hybrid Coordination Function)
- Still most are using DCF



Wireless Frames

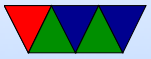
Different types have different layout



Wireless Data Frames

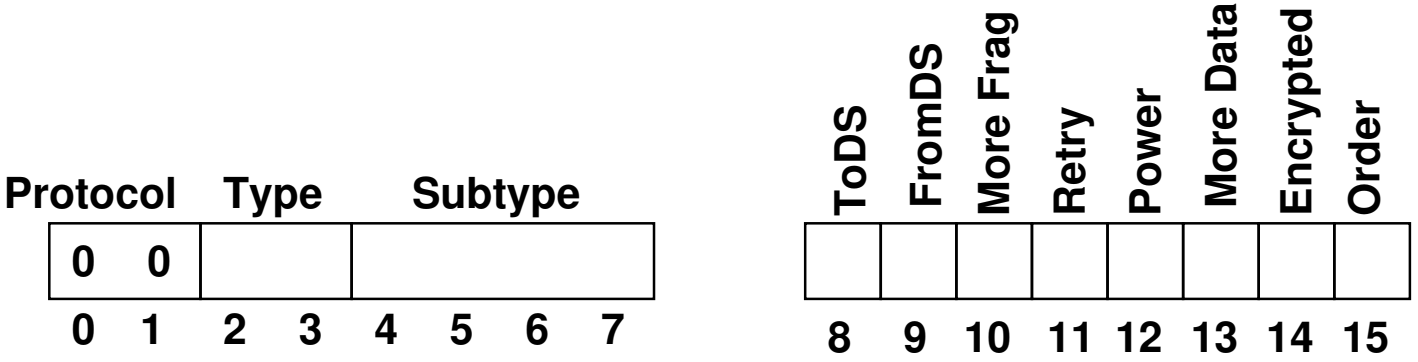
FC	Duration	Addr 1	Addr 2	Addr 3	SEQ	Addr 4	Body	FCS
2	2	6	6	6	2	6	0-2312	4

Note: the IEEE spec lists the fields LSB first. This sort of makes sense as they get transmitted in that order, but we usually write valus MSB first so this makes it a huge pain to decode.

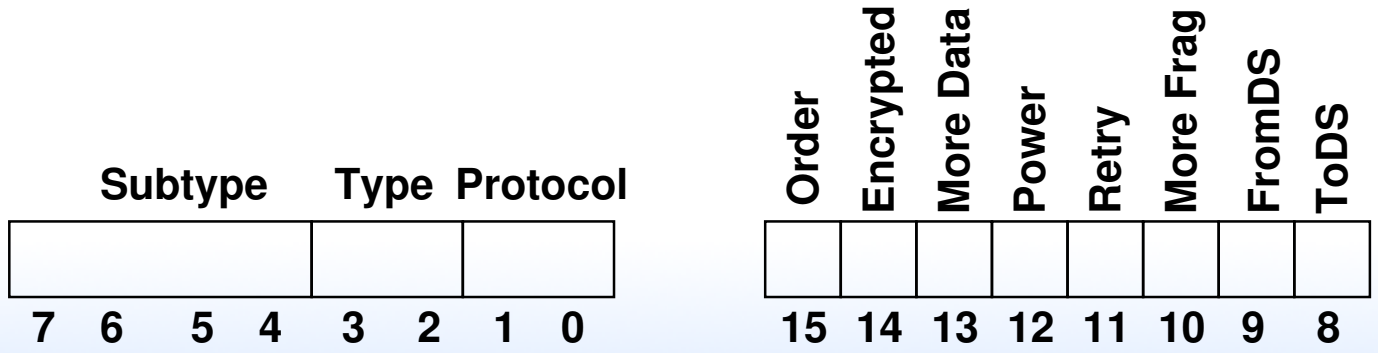


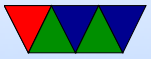
Wireless Frames – Frame Control (FC)

From the specification:



How we'd see the data in a hexdump:

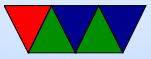




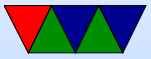
- Protocol Version (2 bits) [only 00 in practice]
- Type (2 bits): note, bit order is 3/2 so flipped depending how you decode
  00=management, 01=control, 10=data, 11=reserved
- Subtype (4 bits): bits 7,6,5,4
  see next slide for expansion
- ToDS/FromDS(1,1) (going to or from the access point)

	ToDS=0	ToDS=1
FromDS=0	mgmt	from station
FromDS=1	to station	bridge

- MF – more fragments to follow

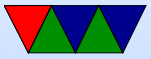


- Retry
- Power Management (into or out of sleep)
- More (more data coming, more than 1 frame being sent)
- W or Protection: WEP (or other encryption enabled)
- O frames must be in-order



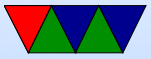
Wireless Frames – Management Frames (00)

- 0000 – Association request
- 0001 – Association response
- 0010 – Reassociation request
- 0011 – Reassociation response
- 0100 – Probe request
- 0101 – Probe response
- 1000 – Beacon
- 1001 – Announcement traddic indication message (ATIM)
- 1010 – Disassociation
- 1011 – Authentication
- 1100 – Deauthentication
- 1101 – Action (spectrum management)



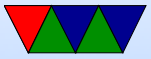
Wireless Frames – Control Frames (01)

- 0000 - 0111 (resereved)
- 1000 – Block Acknowledge Request
- 1001 – Block Acknowledgement
- 1010 – Power Save Poll
- 1011 – RTS
- 1100 – CTS
- 1101 – ACK
- 1110 – Contention-free end
- 1111 – CF-END+CF-Ack



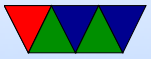
Wireless Frames – Data Frames (10)

- 0000 – Data
- 0001 – Data+CF-Ack
- 0010 – Data+CF-Poll
- 0011 – Data+CF-Ack+CF-Poll
- 0100 – Null data (no data)
- 0101 – CF-ACK (no data)
- 0110 – CF-Poll (no data)
- 0111 – CF-Ack+CF-Poll (no data)
- 1000 - 1111 - same as above but with QoS



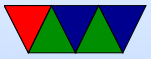
Wireless Frames – Duration

- Duration/ID (2 bytes) – how long will occupy channel
- Network Allocation Vector (NAV)
- TODO: this varies based on various things, including fragmentation
- is microseconds?
- Special meaning, top bits 10 = contention free period, max 32k. top bits 11 = poll, for sleeping devices



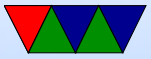
Wireless Frames – Addresses

- Note that destination/receiver are not necessarily same
  Destination: where the bytes will be used
  Receiver: device that is going to decode the radio waves
- Same with transmitter/sender
  Sender is the device who put together the bytes in the packet
  Transmitter is device that sent it out over the radio waves
- Special case when broadcast/multicast, BSSID also



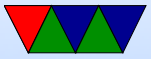
checked

- See table (source IEEE 802-11 2012 Table 8-19)
- How addresses defined depends on tods/fromds fields
- Addr1 (6 bytes) Receiver
  Usually destination, not always
- Addr2 (6 bytes) Transmitter
- Addr3 (6 bytes) Base Station Source? filtering?
- Address4 (6 bytes) Base Station Dest (for wireless bridges, uncommon to use)
- More on addresses
  - basic service set identifier (BSSID)



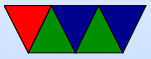
MAC address of the access point (randomly generated?)

- source address (SA)
- destination address (DA)
- transmitting address (TA) who sent it,
- receiving STA address (RA) destination, this might not be addr1 on CTS/ACK frames.



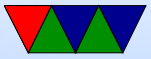
Wireless Frames – Sequence/Fragment

- Sequence control (2 bytes)
- Fragment (4)
- Frame (sequence) (12 bits)
- TODO: book has more on this



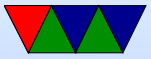
Wireless Frames – Body

- Frame body (0-2312)
- Can actually be 0 (no need to pad for collisions) control frames can be size 0
- Actually can be a bit more
- Why that size? Idea is for about 2k of data, then with the additional frame/packet/encryption overhead
- In practice rarely would see much bigger than 1500 bytes because things would have to be fragmented once they hit wired ethernet



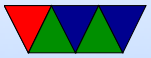
Wireless Frames – Encapsulation

- How can you tell what is in a frame (IPv4, IPv6, ARP, etc?)
- Ethernet has a type field, but we don't
- For wifi we have to encapsulate it
- Two ways to do this
  - RFC 1042 (sometimes called IETF)
  - IEEE 802.1H (tunnel)
  - Due to Microsoft precedent, Appletalk/IPX use 802.1H, IPv4/IPv6/etc use IETF



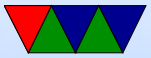
Wireless Frames – RFC1042 Encapsulation

- Start with wifi frame
- Add SNAP field (AA AA) (?)
- Add type
- Then include data
- Double check this
- Relatively straightforward to go ethernet to wifi and wifi to ethernet



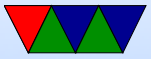
Wireless to Wired

- Validate, discard if not for BSSID
- Decrypt
- Reassemble frame
- Setup ethernet header
- Recalculate FCS
- Transmit



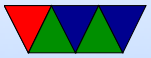
Wired to Wireless

- Setup wifi header
- Encapsulate
- Queue to transmit
- Encrypt
- Recalculate FCS
- Transmit



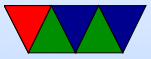
Wireless Frames – CRC

- FCS CRC (4 bytes)
- Same as Ethernet, but has to be recalculated if move from Ethernet to Wifi due to different header
- ACK if correct
- If incorrect, no NACK, just drop it, so wait for timeout



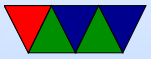
Multi-rate Handling

- Common speeds handled by all devices
- Speeds used by two currently talking (station and AP usually)
- Rate Fallback (slow down if too many errors)

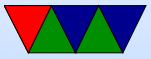


Frames

- Class 1 – send any time
- Class 2 – only if authenticated
- Class 3 – only if associated

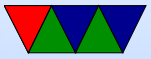


Control Frames



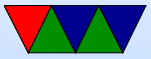
Control Frames – RTS

- Subtype 1011
- Duration
- Addr1: station
- Addr2: transmitter



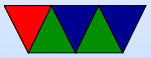
Control Frames – CTS

- Subtype 1100
- Answer RTS
- Used in 802.11g to avoid interference with 802.11b

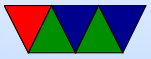


Control Frames – ACK

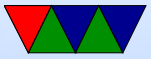
- Subtype 1101
- Addr1: receiver addr



Control Frames – PS-Poll

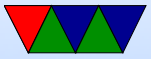


Management Frames – Authentication



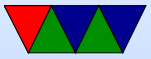
Management Frames – Capabilities / Beacon

- Address of AP
- Listen Interval
- Association ID/Timestamp
- Reach code
- Status code
- SSID – plain text, 32 bytes (usually ASCII), easier than MAC for keeping BSSID separate
- Supported rates, mandatory and optional (originally



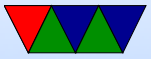
multiple of 500k/s)

- Association ID
- Freq-Hop
- DS (channel)
- Traffic Indication Map
- Country – 3 bytes (country abbreviation plus I/O for indoor/outdoor)



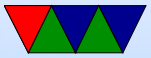
Management Frames – Beacon Interval

- announce existence of 802.11 at regular interval
- time around 1ms, but called kilo-microseconds (kus). kilo is 1024, ms=1000



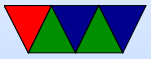
Management Frames – Beacon

- Probe request
- Probe response
- Disassociation
- Deauthentication
- Association Request
- Reassociation Request



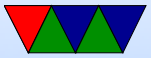
Reliability

- Wireless can be noisy and unreliable
- What do you do if there's packet loss?
  - Send slower
  - Send shorter frames
  - Fragment frames

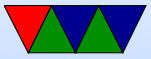


802.11e QoS

- Leaves idle time before sending next frame
- Different sizes for different traffic
- DIFS – DCF inter-frame spacing
  - SIFS – short (control frames)
  - AIFS1 – Arbitrary (high priority)
  - DIFS – regular DIFS
  - AIFS4 – low priority
  - EIFS – extended, for errors
- TXOP – transmission opportunity

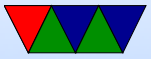


- Usually fixed number of frames so faster devices held back
- Instead, provide equal airtime rather than equal frames



Power Saving

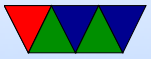
- Important for mobile devices
- AP beacon frames from AP every 100ms
- Can indicate you want power saving, then go to sleep, on next beacon wake up and notice from beacon if data available to read (AP will buffer)
- APSD – auto power-save, AP will buffer frames until device sends something, send buffered data knowing it's awake



Wireless Services

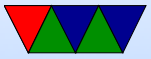
Must provide 9 services

- intracell for dealing with things outside of a cell
  - Association – allow stations to connect to base stations. When arriving announce its identity and capability
  - Disassociation – either side may break the association, should do it before shutting down
  - Reassociation – can change preferred base station, useful for handover (but best-effort)



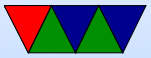
- Distribution – determines best way to route frames
- Integration – in case frame needs to be sent through a non-802.11 network

- Intercell
  - Authentication – check password
  - Deauthentication – to leave network
  - Privacy – encryption
  - Data delivery – modeled on Ethernet, no guarantees frames will get in



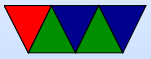
Encryption

- Important as anyone can eavesdrop, even from a distance
- https://arstechnica.com/gadgets/2019/03/802-eleventy-who-goes-there-wpa3-wi-fi-security-and-what-



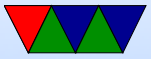
WEP (obsolete)

- WEP – Wired Equivalent Privacy
  - Used RC4 and CRC32
  - Deprecated 2004
  - Meant to be 64 bit, originally 40 but due to export limitations
  - Later 128-bit. Can enter in hex or ASCII chars
  - Can be cracked fairly quickly these days (10 mins on a laptop)



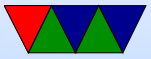
WPA (obsolete)

- WPA – Wi-Fi Protected Access (WPA)
  - 802.11i – Temporal Key Integrity Protocol (TKIP)
  - 64 or 128 bit encryption key
  - TKIP replace CRC, harder to crack, RC4
  - WPA-personal Pre-shared key, AKA the password. 128 bits derived from 256 bits. If ASCII, PBKDF2 applied and then SSID used as salt (to prevent rainbow tables)
  - WPA-enterprise is more complicated key setup



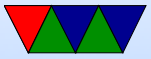
WPA2

- WPA2 Wi-Fi Protected Access II (WPA2)
- IEEE 802.11i-2004
  - 4-way handshake (recent issues with that on many Linux machines), Ironically had issues for too closely following IEEE standard
  - AES?
- Was broke in 2017, have to do with key exchange?



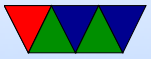
WPA3

- WPA3 – 2018
  - 802.11-2016
  - Replaces PSK (pre-shared key exchange) with SAE
  - Simultaneous Authentication of Equals instead of pre-shared key
- Supposed to help with weak passwords, and also configuring devices without displays



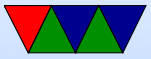
Wireless encryption security notes

- Rainbow tables
- WPA/WPA2 can try to crack weak passwords offline
- WPA3 need active connection to network to do it
- WPA/WPA2 lack of forward secrecy, once password broke can decrypt all future and past traffic



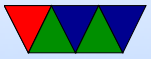
Authenticated

- Three states: not authenticated, authenticated but not associated, authenticated and associated
- device sends probe requests. Advertise data rates and what version of 802.11 supported. BSSID of ff:ff:ff:ff:ff:ff so all access points that hear it will respond
- if an access point (AP) supports a common data rate, it will respond with SSID, data rate, encryption mode, etc
- device chooses an access point and authenticates.

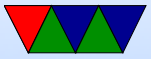


Originally this would have been WEP, but deprecated so often happens in open and usually succeeds. Device sends a 802.11 open authentication frame, seq 0x01

- AP responds saying open with seq 0x02
- if AP receives frames other than auth or probe from device, responds with a deauth to make it start over
- A device can be authenticated to multiple APs but only associated with one
- device determines who to associate with and requests

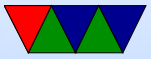


- AP responds and creates association ID
- once associated then WPA/WPA2 has to happen still before data can flow

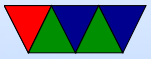


Encryption Issues

- News from last year, security issues with WiFI:
- KRACK attack (key reinstallation attack)
  - 4-way WPA2 handshake
  - You can resend 3rd way of handshake with the key, and other side will accept it (in case it was lost) and re-start encryption from beginning
  - This leads to same key being used to encrypt multiple frames
  - That makes reversing the key trivial

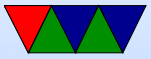


- Frag attack
  - `https://lwn.net/Articles/856044/`
  - (fragmentation bit is outside of the encrypted/protected, so by messing with that you can get rogue chunks of encrypted data inserted into frames)

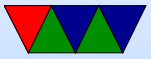


Security Issues

- Packet sniffing
- Easier to tap into a network undetected. Long range antennas
- Malicious association – go into an area with own access point that machines will connect to
- MAC spoofing, set your MAC to an existing machines
- Denial of Service – flood the router so it can't respond



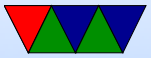
- Deauthentication attack– continually spoof an "I'm leaving" packet from all MAC addresses on network
- Hide you SSID? How effective is that?
- Encryption breaking, see long list of issues on WPA Wikipedia page



Transmission Power

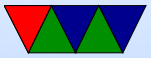
- 802.11b signal typically around 32mW
- Often use dBmW (often shorted dBm) where 0dBm=1mW
- 1dBm = 0.001258925W
- Convert -68 dBm to Watts
  - $P = 1W * 10^{P_{dBm}/10}/1000$
  - -68 dBm = 160pW

- Convert 1W to dBm



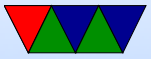
- $P_{dBm} = 10 * log_{10}(1000 * P_W/1W)$
- 1W = 30dBm

- Juno space probe (13 Oct 2016)
  - 8.4GHz, received -135.75dBm (2.7e-20kW) 18kb/s (math is right. why report kW not W though?)



Channels

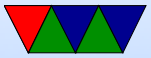
- 802.11b, DSSS 2.4GHz, 2412MHz as first channel, 14 channels 5MHz apart 1-14.
- 802.11g same as 802.11b when talking to b, but a modes when talking to other g
- 802.11a 5GHz band, channels 1-199 starting at 5005MHz 5MHz apart
- CMSA/CA – uses RTS/CTS. 802.11g needs to do this if 802.11b present, slowing things down 20-50%



Linux Interface

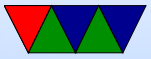
- iwconfig
- iw dev
- iwlist scanning

```
wlan0     IEEE 802.11abg  ESSID:"Whatever"
          Mode:Managed  Frequency:2.452 GHz
                        Access Point: 00:1C:10:11:B4:C6
          Bit Rate=54 Mb/s   Tx-Power=200 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:XXXXX
          Power Management:off
          Link Quality=42/70  Signal level=-68 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
```



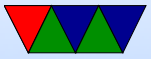
```
Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

What does some of this mean? RTS threshold: can old do CTS/RTS if file is too big
same with Fragment threshold



Capturing 802.11 packets

- Is it possible to gather raw 802.11 packets, like you can with ethernet/tcpdump?
- Tricky. Often wireless networks restrict raw access to the transmitter/receiver for regulatory issues (don't want random code on computer able to blast out radio signals that could interfere with shared network)
- There is a special "monitor" mode you can put some wifi cards into
- None of my machines support it by default



- For pi3/pi4 there are custom firmware replacements (nexmon) that in theory enable it plus other hacking

```
sudo iw dev
sudo iw wlan0 set monitor none
then use wireshark
```