

ECE 435 – Network Engineering

Lecture 19

Vince Weaver

`http://web.eece.maine.edu/~vweaver`

`vincent.weaver@maine.edu`

9 November 2017

Announcements

- Will get back to you about projects, PAPI release interfering
- Class website was down (server compromised) but hopefully it has been transitioned to a better place
- Trials and tribulations of being a sysadmin
- In-class lab on Tuesday – more details in coming e-mail



Things from Last Time

- Some Bridging notes
- What about power usage of the Ethernet interfaces?



Bridging

- How do you connect together multiple groups of machines into one big LAN?
- An interconnection at the link layer is called a MAC bridge, or bridge. Also a Layer-2 switch
- IEEE 802.1D
- Transparent bridge, as users are not aware of them
- Bridge acts in promiscuous mode (receives every frame



on the LAN) so it can find ones that need to forward on across the bridge

- How does bridge learn the MAC addresses? self-learning. It watches for frames coming in and their source address. Puts in table. How does it learn where destination is? It broadcasts to all. Once the destination also sends a frame (so its source is known) then the switch updates its table and no longer broadcasts.
- How do you handle machines that are moved? Aging mechanism. If not heard from for a while, expire the



table

- Multicast or Broadcast, can follow GMRP or GARP to limit how far it is broadcast



Bridge vs Switch

- Before 1991 a switch was a bridge (in the standard)
- In 1991 Kalpana made a “switch” and differentiated it by cut-through instead of store and forward
- Store and forward – whole frame received before resent
larger latency, no problem with broadcast, can check FCS
- cut-through – can start transmitting before receiving completely (destination MAC at beginning). Slightly



better latency, broadcast not possible, too late to check FCS

- These day most are store and forward
- Differences
 - repeater – purely electronic, resends voltages (original Ethernet allowed four)
 - hubs – frames coming in one port sent to all others creates a collision domain
 - bridge – connects two or more LAs. Each line own collision domain

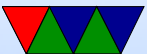


can maybe bridge different types of networks
(Ethernet/token, wired/wireless)

- switch –
- router – actually strips off headers and looks at packets



Wireless



Why Wireless?

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



Wireless LAN

- What does WiFi mean? Nothing really.
- 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



Wireless LAN Standards

- All of the various 802.11 have been sort of merged together, but people use the old letters out of habit
- Original 802.11 (1997) 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) 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) 1.5 - 54Mbps
 - Not compatible with B, 54Mbps in 5GHz band
 - 5GHz less crowded, but signal doesn't go as far
 - 48 data channels 4 syc
 - OFDM (Orthogonal Frequency Division Multiplexing)
Data is sent on multiple channels in parallel
- 802.11g (2003) 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) 54Mbps - 600Mbps
 - MIMO (multiple input/multiple output antennas)
 - Can do spatial multiplexing, two antennas broadcast on same frequency by aiming signal
- 802.11ad – 7Gbps, 60GHz freq
- 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



Wireless Frames

- Diagram (TODO)
- Frame format
 - Frame Control (2 bytes)
 - Protocol Version (2 bits) [only 0 in practice]
 - Type (2 bits): data, control, management
 - Subtype (4 bits) [rts,cts]
 - ToDS/FromDS(1,1) (going to or from the cell)
 - MF (1) more fragments to follow
 - Retry(1)



Power Management(1) (into or out of sleep)

More(1) [more coming]

W(1) WEP [encryption]

O(1) frames must be in-order

- Duration/ID (2 bytes) – how long will occupy channel
- Addr1 (6 bytes) Receiver
- Addr2 (6 bytes) Transmitter
- Addr3 (6 bytes) Base Station Source?
- Sequence control (2 bytes) Frame (12) and fragment (4)
- Address4 (6 bytes) Base Station Dest?



- Frame body (0-2312)
- FCS Checksum (4 bytes)
- Addr3/Addr4 optional, usually if bridge.
- Management frames: similar to data frame, restricted to one cell
- Control frames: Short, only two addresses, no data, no sequence, often just RTS/CTS/ACK



Wireless Network Topology

- Ad-hoc mode – peer to peer
- 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)



DCF – Distributed Coordination Function

- Basic DCF
 - No central control
 - Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)
 - Different from Ethernet CSMA/CD.
 - Every time ready to transmit, looks to see if can transmit (listen to see if channel clear)
 - If clear, waits DIFS (inter frame) time and transmits
 - If busy, waits until clear and then also a random backoff



time before starting

- 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
- Optional 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 too to hint how long it will take
- ACK at end
- Fragmentation – the longer the packet, the more likely it is to lose bits to interference. So split things up into smaller chunks likely to get through



PCF – Point Coordination Function

- 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.



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

- 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
- WPA – Wi-Fi Protected Access (WPA) and Wi-Fi Protected Access II (WPA2) 64 or 128 bit encryption key



- WPA2

4-way handshake (recent issues with that on many Linux machines)

Ironically had issues for too closely following IEEE standard



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



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?



More

- 802.11b signal typically around 32mW
Often use dBmW where 0dBm=1mW
 $1\text{dBm} = 0.001258925\text{W}$
 $P = 1\text{W} * 10^{P_{dBm}/10}/1000$ 200??? -68 dBm = 160pW
 $P_{dBm} = 10 * \log_{10}(1000 * P_W/1\text{W})$
- 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
- CMA/CA – uses RTS/CTS. 802.11g needs to do this
if 802.11b present, slowing things down 20-50%



Linux Interface

wlan0

IEEE 802.11abg ESSID:"Whatever"

Mode:Managed Frequency:2.452 GHz Access Point: 00:1C:10:11

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



Bluetooth!

- No time to cover this Fall 2017 due to storm cancellation



Bluetooth Applications

- Headsets
- Wireless controllers (Wii, PS3)



Bluetooth

- 1994 Ericsson. With IBM, Intel, Nokia and Toshiba formed a SIG.
- Named after Harald Blaatand (Bluetooth II (940-981) a Viking king who “united” (conquered) Denmark and Norway. Unite various standards.
- Get rid of cables, specifically serial cables
- Interferes with 802.11
- IEEE came in and decided to take standard and make it 802.15.1 but no longer maintains it



Bluetooth Architecture

- Basic unit: piconet, master node and up to seven *active* slave nodes within 10m
- Many can exist in an area, and can be connected by a bridge. Connected piconets are called a scatternet
- There can also be up to 255 “parked” nodes in a picnoet
- When parked, can only respond to activation on beacon
- Hold and siff?



- Slaves designed to be cheap, so dumb. Master is smart and runs them. slave/slave communication not possible
- Master broadcasts clock 312.5us. Master transmits in even, slave in odd.



Bluetooth Applications – Profiles

Bluetooth V1.1 has 13 different application protocols.

- Required
 - generic access – link management
 - service discovery – discovering services
- - Serial port
 - Object exchange
- Networking
 - LAN access
 - Dial-up



- Fax
- Telephony
 - Cordless phone
 - Intercom
 - Headset
- File exchange
 - Object push
 - File transfer
 - Synchronization



Bluetooth Layering

- Radio layer – 2.4GHz, 10 meters. 79 channels of 1MHz. Frequency shift keying, 1 Mbps but consumed by overhead Frequency hopping spread spectrum, 1600 hops/sec dwell of 625 usec. All nodes in piconet hop at once, with master controlling this
Interferes with 802.11. Bluetooth hops faster so causes more trouble.
power output class: 100mW class 1, 2.5mW class 2, 1mW class 3.



- Baseband layer – Asynchronous Connection-less link (ACL) packet-switch data at irregular info, no guarantees. one per slice
Synchronous Connection Oriented (SCO) – for real time data. Three per slave. Error correction. Each can send 64kpbs PCM audio
- L2CAP layer – accept packets of 64kB and break into frames. Handles multiplexing.



Bluetooth Frames

- Several different formats
- 72 bits access (identify master, as can be in range of multiple)
- 54 bit header (addr(3) frame type(4), flow [buffer full](1), Ack (1) seq(1) checksum. This is repeated 3 times. Majority wins (redundancy, cheap small protocol)
- Data 0-2744 bits. SCO always 240 bits.



Bluetooth 1.1 (2002)

- First stable version



Bluetooth 1.2

- Adaptive frequency hopping, skip busy frequencies
- Up to 721kbps
- eSCO allow retransmitting corrupted packets, at expense of audio latency
- HCI host controller interface, three wire



Bluetooth 2.0 (2004)

- BR/EDR 2 and 3Mbps



Bluetooth 3.0 (2009)

- 25Mbps
- Alternative MAC, bluetooth set up connection but 802.11 used to transmit data?



Bluetooth 4.0 (Bluetooth Low Energy) (2010)

- 25Mbps/200 feet
- Entirely new stack, designed for low power rapid setup links
- Not backwards compatible, but same frequency range
- New profiles



Bluetooth 5.0 (2017)

- Internet of things
- 50Mbps/800 feet



Setting up Connections

- In discoverable mode, will transmit name, class, services, etc on demand
- Has unique 48 bit number but that's rarely seen
- Bonding/Pairing – to avoid people stealing info from your device, require some sort of user interaction to connect for the first time. Before 2.1 it was a 16-byte pin code



Security

- Prior to 2.1 security can be turned off, and only good for 23.5 hours



Linux Bluetooth

- Competing implementations
- Install bluez
- bluetoothctl

```
[NEW] Controller B8:27:EB:05:9D:BB pi3 [default]  
[bluetooth]# exit  
[DEL] Controller B8:27:EB:05:9D:BB pi3 [default]
```



```
root@pi3:/home/vince# bluetoothctl
[NEW] Controller B8:27:EB:05:9D:BB pi3 [default]
[bluetooth]# scan on
Discovery started
[CHG] Controller B8:27:EB:05:9D:BB Discovering:
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# scan on
Failed to start discovery: org.bluez.Error.InPr
[bluetooth]# scan on
Failed to start discovery: org.bluez.Error.InPr
```



[NEW] Device 64:8A:44:9D:DC:FD 64-8A-44-9D-DC-F
[NEW] Device D3:E8:9D:CA:71:63 D3-E8-9D-CA-71-6
[CHG] Device D3:E8:9D:CA:71:63 RSSI: -89

