COMP3310/6331 – #7-8

Ethernet and WiFi

Dr Markus Buchhorn: *markus.buchhorn@anu.edu.au*

Acronym overload

- General "LANs": Ethernet, WiFi, Bluetooth, 4G?
  - Cheap, mass-produced, reasonably well-behaved, scalable – and simple!
- Carrier-grade: SONET/SDH, ATM, FrameRelay, GPON, …
  - Expensive, robust, service-level guarantees
- Data-centre: FibreChannel, Infiniband, FDDI, …
  - High-speed, low-latency, specific-purpose
- Wireless: RF, LiFi, whitespace, Zigbee, Z-wave, HaLow, 6LoWPAN, …
  - Regulated in some frequencies, free-for-all in others
  - Device-oriented: Low power (long battery life), long range, low datarates

Standards

- IEEE: community standards, active research, publications
  - Different to ISO, ITU, IEC – government-recognised standards bodies
  - Physical engineering,
    - Also naming, best practices, software/hardware architecture, …
- IEEE 802: standards and committee
  - *LAN/MAN networks carrying variable-sized frames* (not cells)

- 802.1: MAC details
- 802.3: Ethernet
- 802.11: Wireless LAN & Mesh
- 802.15: Wireless PAN
- 802.16: Broadband Wireless Access (WiMAX)

802.3 Ethernet

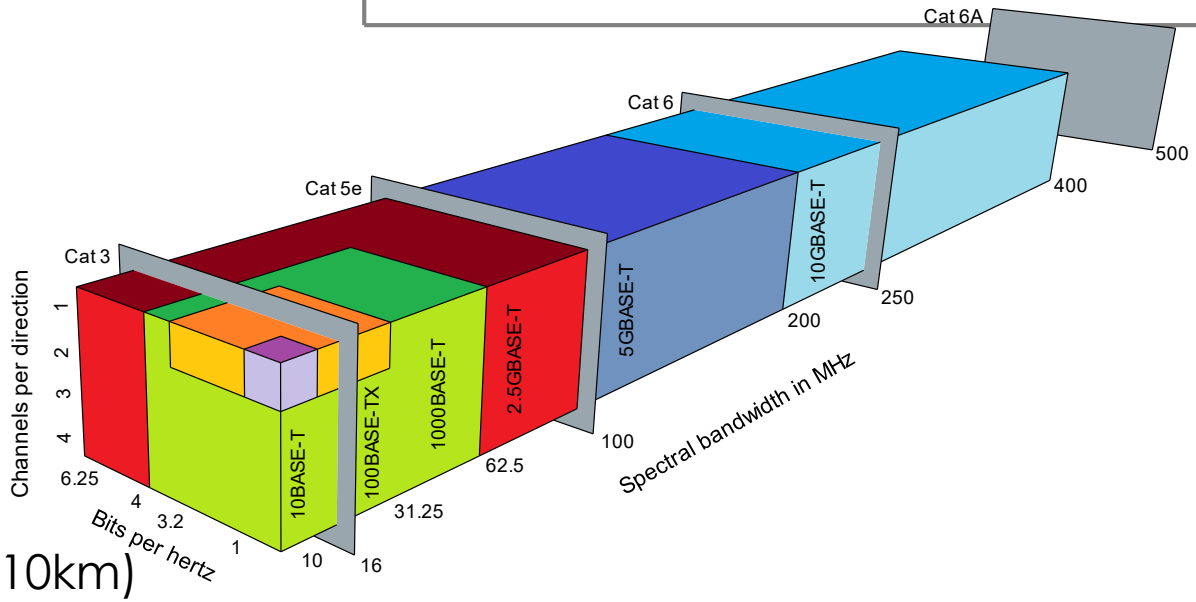
- 1983 – coax cables, vampire taps, 10Mb/s
- It's evolved a lot since then: faster, further, more robust, more functionality
- Lots of individual standards, sometimes superseding or merging earlier versions.
- Lots of backwards compatibility
- 802.3a - 802.3z (1985-1998)
- 802.3aa - az (1998-2010)
- 802.3ba - bz (2010-2016)
- 802.3ca - cs (2016-today)

Which Ethernet?

- **10Base2** vs **1000Base-LX** ??
- Naming:
  - Speeds: 10, 100, 1000 (Mb/s), 2.5G, 10G, 25G, 40G, 50G, 100G, 200G, 400G
  - Signal: BASE, PASS, BROAD
  - -? = media. T=Twisted Pair, S=short 850nm, L=long 1300nm, E=extralong 1550nm
    - C (or blank) =coax. Mostly. F=fibre. B=bidirectional (single fibre).
  - Last letter = encoding (8b/10b, …), or ignored
  - Last number = channel count (wavelengths, copper pairs).
    - Or reach (2,5,36 * 100m, or 10,20,30*1km). Or …

So…

- 1000BASE-LX:
  - 1Gb/s, Baseband
  - Fibre optic pairs,
  - 1310nm over MMF (500m) or SMF (10km)
  - 8b/10b NRZ encoding



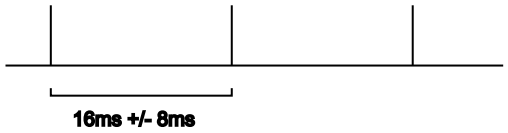
- 1000BASE-T
  - 1Gb/s, Baseband
  - Twisted Pair copper cables with 8P8C (RJ45) connectors, Cat5e or better
  - Uses all four pairs, in both-directions simultaneously
  - 4b/5b, PAM-5 encoding

Ethernet

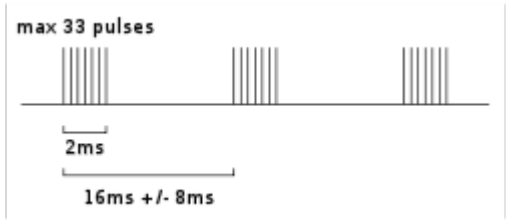
- Started over shared medium coax – CSMA-CD, Manchester
  - 10Base2, 10Base5
  - Moved to UTP, 10BaseT, using 1-4 pairs
  - A plethora of encodings, differential signalling, and ultimately fibre
  - Moved from shared media to fully-switched
- Half-duplex (bus) to full-duplex (SDM) to full-duplex (FDM)
- If not all 4 copper pairs in use, can run power, telephone over the others
  - And with FDM – power over data wires. 802.3af, at, bt, and bu (for cars)
- Very good plug-and-play
  - Well designed to cope with network changes
- Very good backwards compatibility
  - Link negotiation on connection

Auto-negotiation

- When plugging in an Ethernet device to a switch, need to agree:
  - Speed
  - Duplex
  - Cross-over (which wire does what)
- Need to detect a plug-in/disconnect.
- **Heartbeat** = Normal Link Pulses (NLP)



- **Capability** = Fast Link Pulses (FLP)
  - Encodes messages in 16bit words



- Allows both ends to negotiate and agree
  - *(if they're allowed to!)*

An Ethernet frame

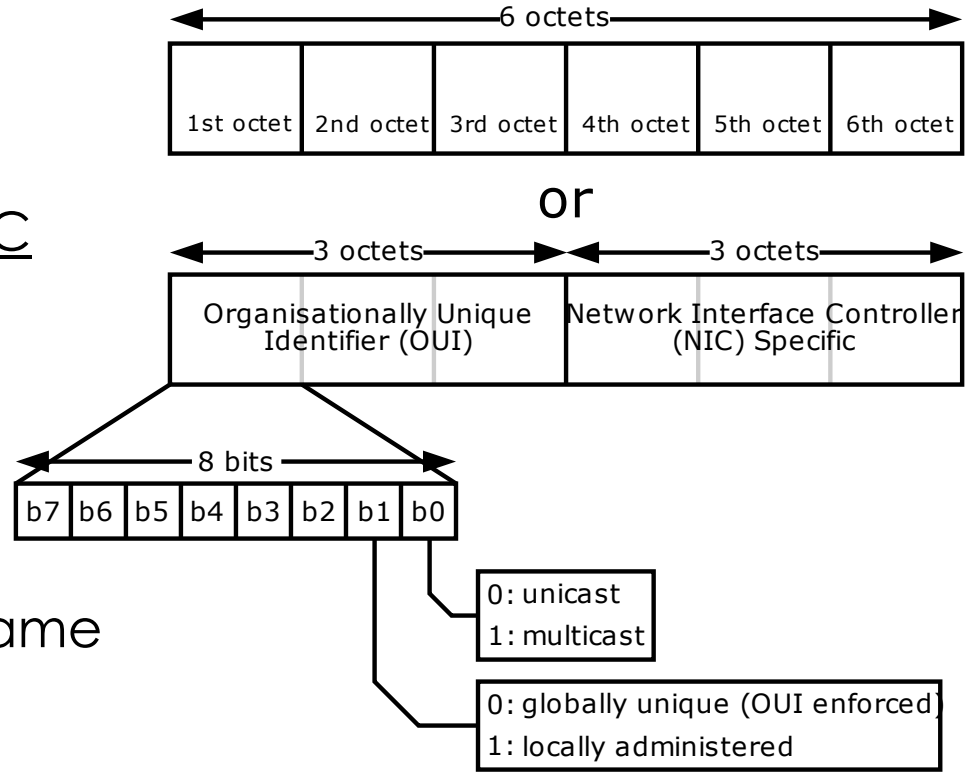
byte = 8bits

Preamble	Start of Frame	MAC dest	MAC src	802.1Q tag [opt]	Type / Length	Payload	Checksum
7 byte	1 byte	6 byte	6 byte	4 byte	2 byte	42-1500 byte	4 byte

- Every device that can listen will receive the frame
- If the frame destination is not yours, drop it, otherwise inform Operating System
  - *UNLESS you are in 'promiscuous mode', and listening to everything*

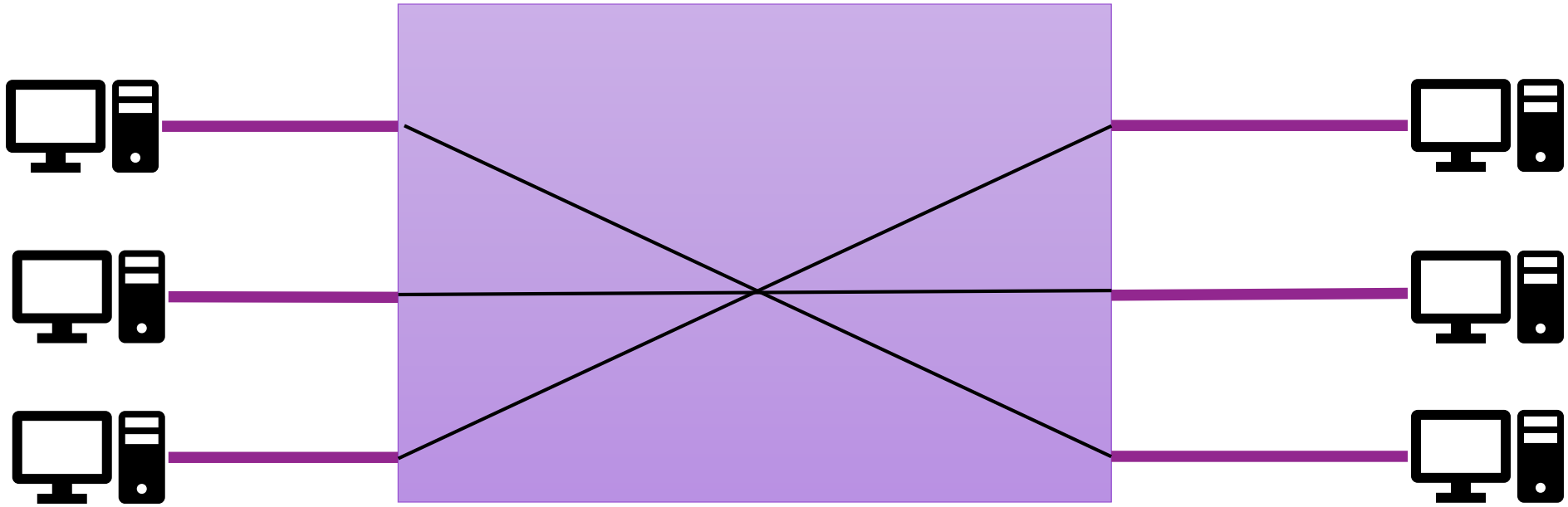
Addressing

- (802) MAC addresses
- Globally unique address (EUI-48)
- Various allocations of 48 bits to a NIC
- Written in hex: *38:10:d5:bc:be:99*
- ‘All ones’ ff:ff:ff:ff:ff:ff = broadcast frame
- *Some addresses are special messages*



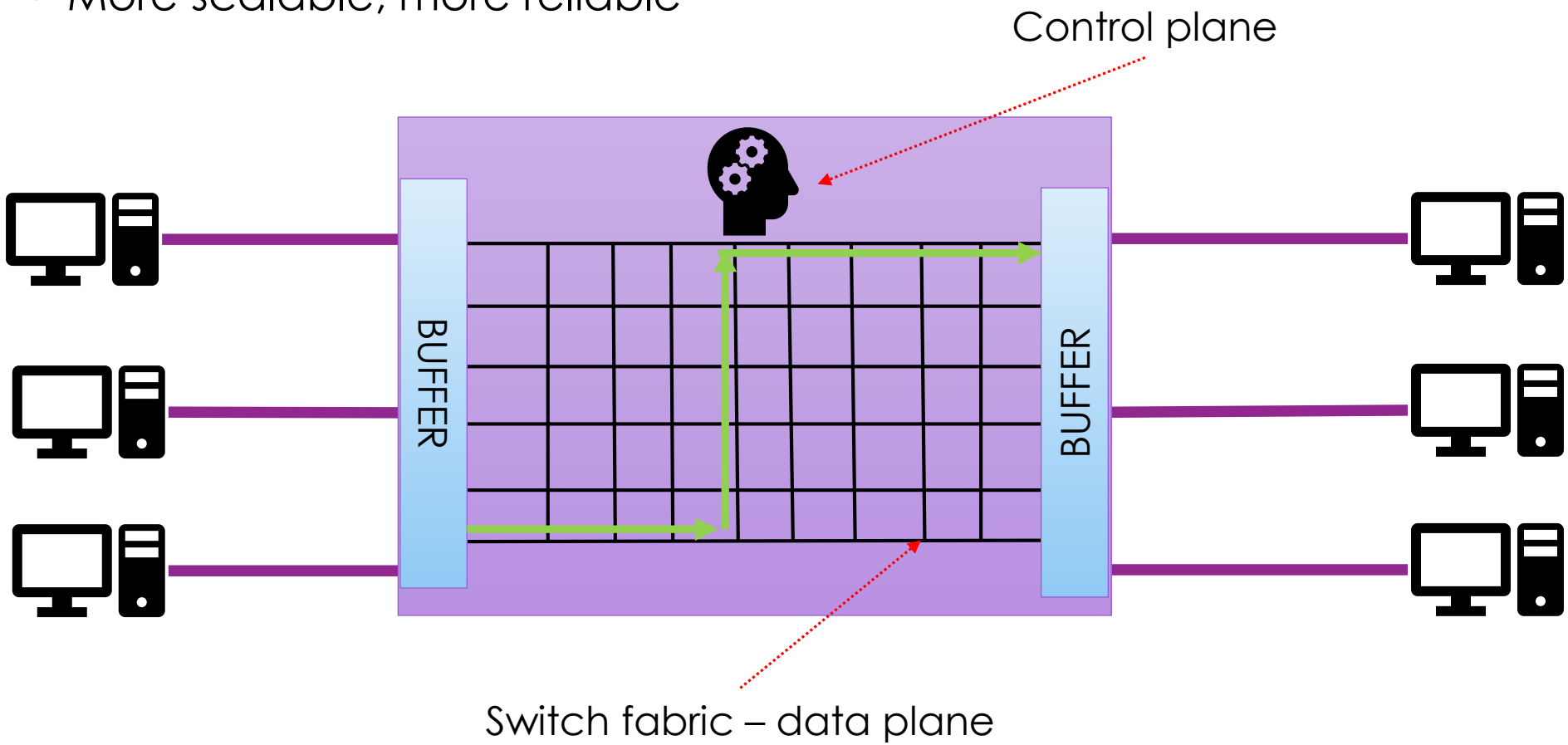
Ethernet hubs

- Shared media, CSMA and collisions – through a hub/repeater



Ethernet Switching

- More scalable, more reliable



What goes where?

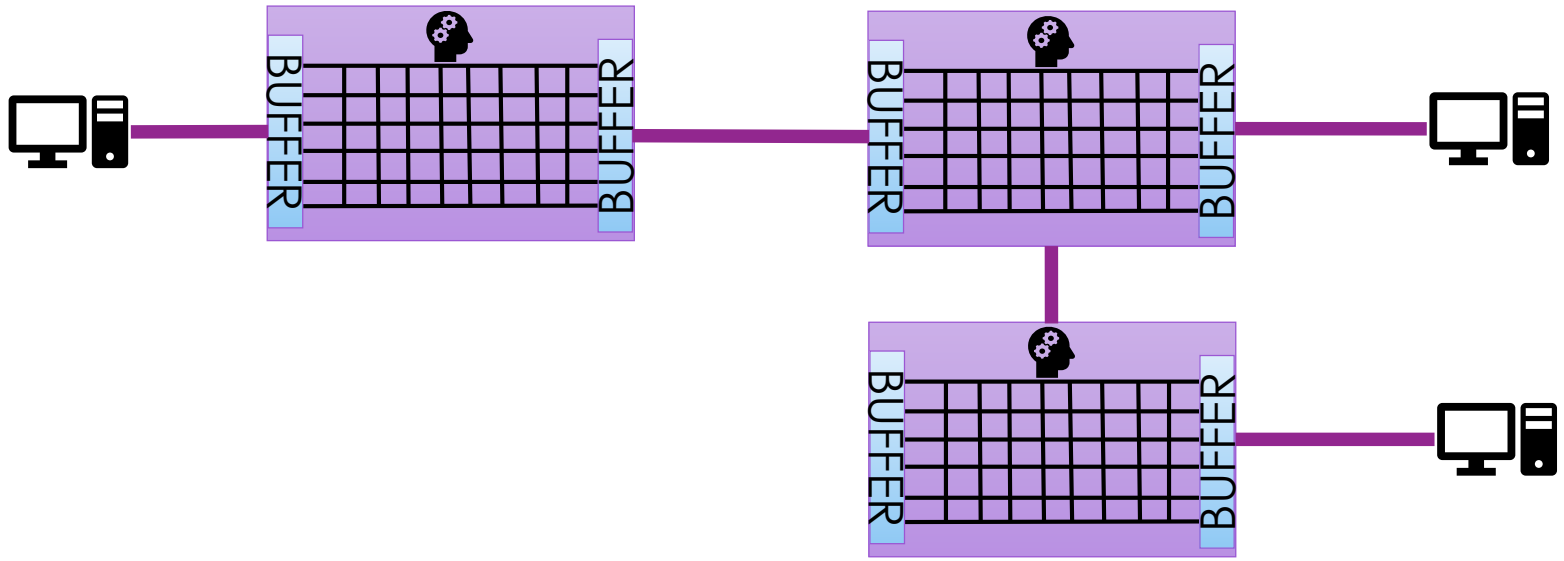
- Need to know which MAC address(es) is(are) on which port
  - Without being manually told

- Switches learn on the fly
  - Using source addresses, most trustworthy.

- **First** – listen to what's coming in, and record the source MAC address
- **Second** – if it's a new MAC destination:
  - Send it to all ports(*) *(unicast port flooding)*
  - Hope somebody replies, and then record their port.
- Broadcasting is now the switch's responsibility – not the cable.

Hierarchy of switches

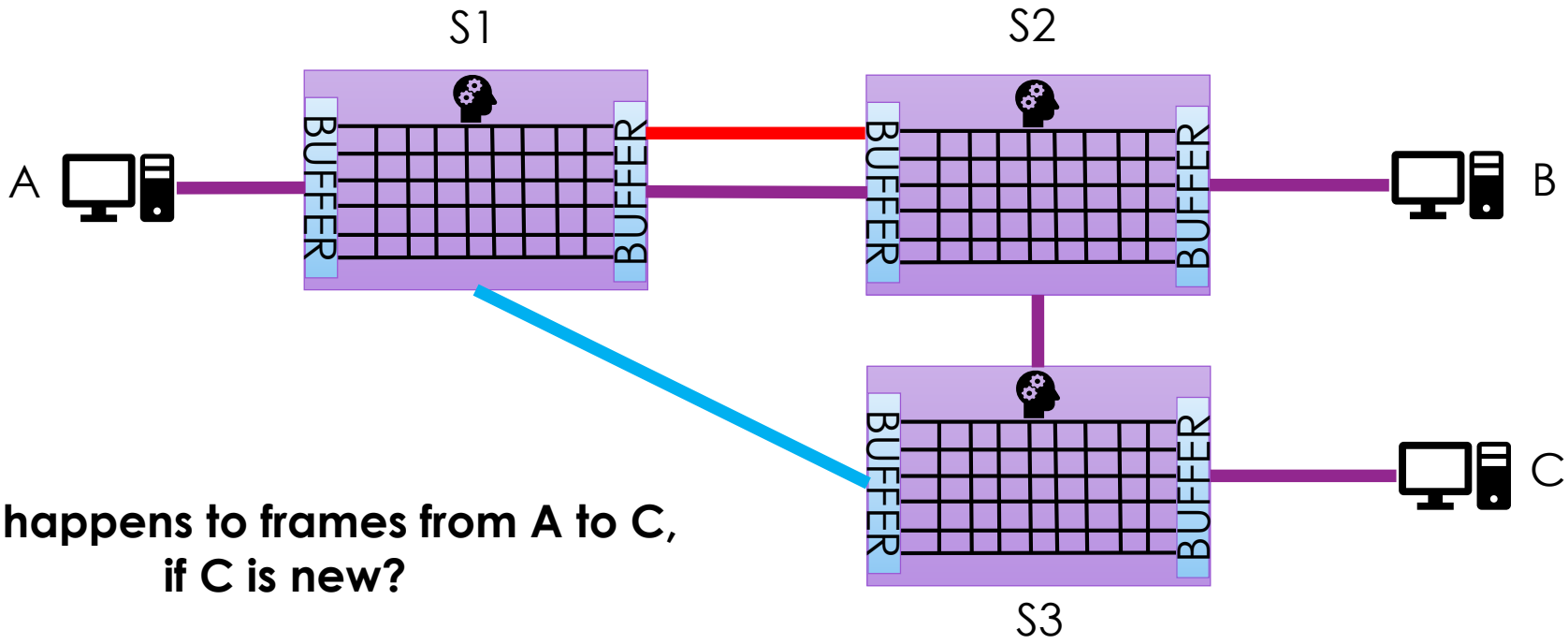
- This works well for any loop-free topology



And now a 'broadcast' domain can be defined as wide as you want

What about if there are loops?

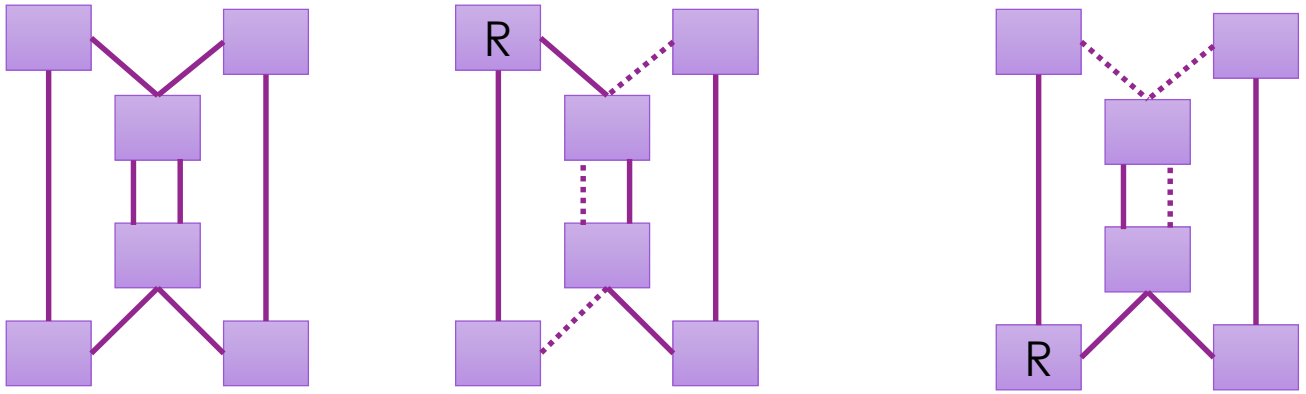
- Redundant links. Parallel links. Short cuts. Made a mistake. Evil intent.



What happens to frames from A to C, if C is new?

Spanning Tree

- Broadcast storms, MAC table updates, duplicate frames – BAD!
- Develop an overlay view that **spans** the network with a loop-free **tree**
  - Effectively: disable some links ("block ports")
  - Switches need to work this out themselves, and adapt, in real-time
  - And then forward frames accordingly



Spanning Tree (Protocol) rules – 802.1d, w, …

- *Before doing anything else - or on any change – or a timer - block all but STP traffic*
- Elect a root node (lowest address wins), and at the same time
- Grow the shortest tree, using distance (hop count) and value (speed) from root
  - Tie-breaker: lowest address
  - Record the ports that are on the tree towards the root

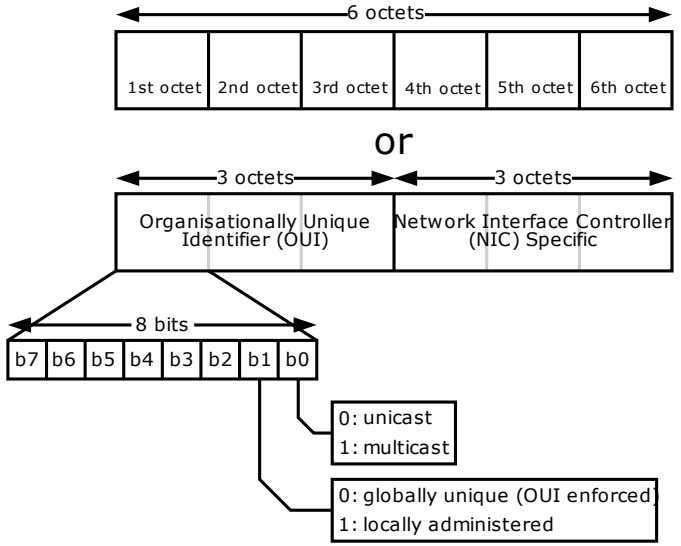
- Initially everyone thinks they are the root – and tells their neighbours so.
  - Some switches get disappointed quickly
  - They tell their neighbours
  - Everyone updates

https://www.youtube.com/watch?v=japdEY1UKe4

- Once converged: turn off ports (paths) that aren't on the tree
  - But remember they are there, if/when something changes

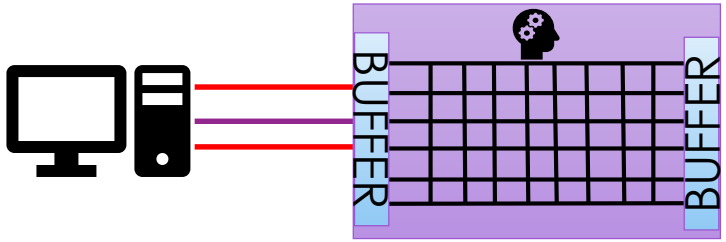
Casting

- Broad-cast: Everyone gets it. MAC destination = all 1's (ff:ff:ff:ff:ff:ff)
- Uni-cast: Only the intended recipient (should) get it. MAC destination
- Multi-cast: Everyone who is interested gets it
  - Special bit-flag in MAC address
  - Devices can 'subscribe' their NIC



Special features

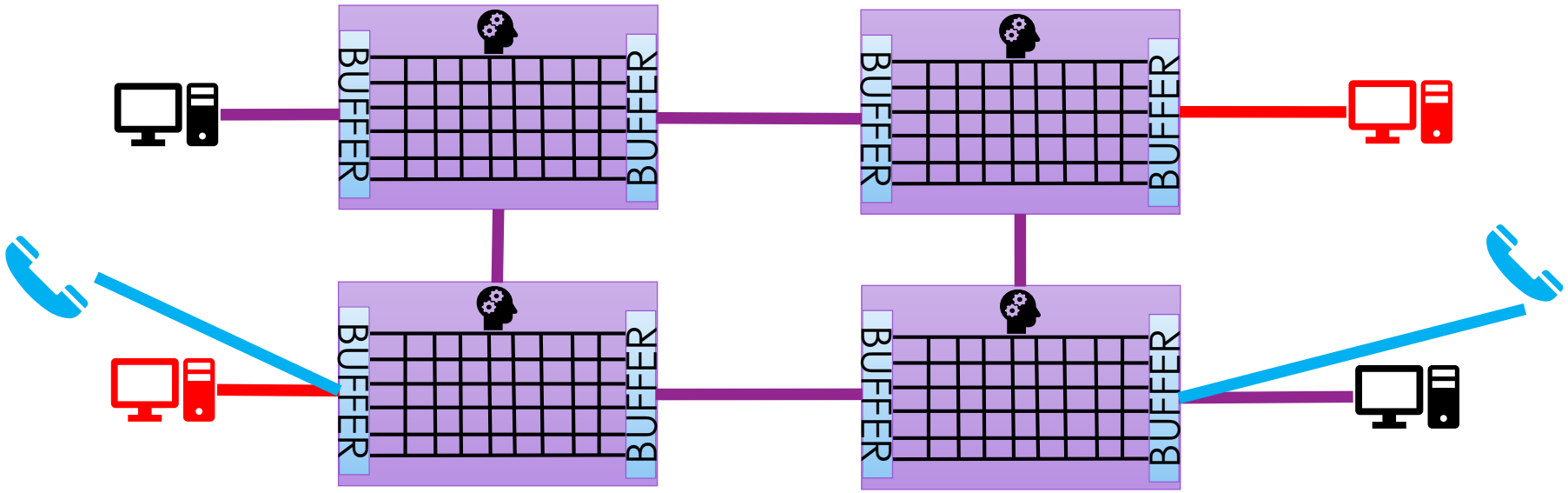
- Link aggregation/ "trunking"
  - When a single 1Gb/s or 10Gb/s won't do
  - Performance
  - Failover



- 802.1AX (was 802.3ad) – up to 8 (identical) links
- *Link Aggregation Control Protocol (LACP)*
  - Send a 'do you do LACP?' every second
  - If yes, identify other common links and aggregate
- Various modes: round-robin, active-backup, random-alloc, …
  - Must not cause mis-ordered or partial frames, nor duplicates

Virtual LANs (VLANs)

- 802.1Q – add a 4-byte "tag" to the Ethernet frame



- *Now have 2+ 'broadcast domains' on the same network*
  - ***Separation** of traffic*
  - ***Prioritisation** of traffic (was 802.1p)*

Going big…

Preamble	Start of Frame	MAC dest	MAC src	802.1Q tag [opt]	Type / Length	**Payload**	Checksum
7 byte	1 byte	6 byte	6 byte	4 byte	2 byte	**42-1500 byte**	4 byte

- Standard maximum frame:
  - 1500 bytes data, 26-30(+12) bytes 'overhead' – at **best**
  - At 10Gb/s, one maximum frame every 1.5µs
  - Buffers overflow, congestion, drops

- What happens if we make the payload bigger?
  - **Jumbo Frames** – 9000 byte payload
  - Lower overhead, lower cpu load – but need a different checksum

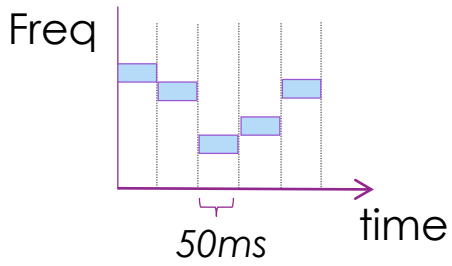
WiFi! (aka WLAN)

- WiFi is not wireless Ethernet
  - But has inherited a lot from it
  - Access points like 802.3 repeaters
- Much more challenging communications environment, clients
  - More work to be robust and perform well
  - Rate and power adaptation
- Based on CSMA/CA – with optional RTS/CTS (MACA)
  - Single access point for many clients…
- Along with OFDM, DSSS and MIMO

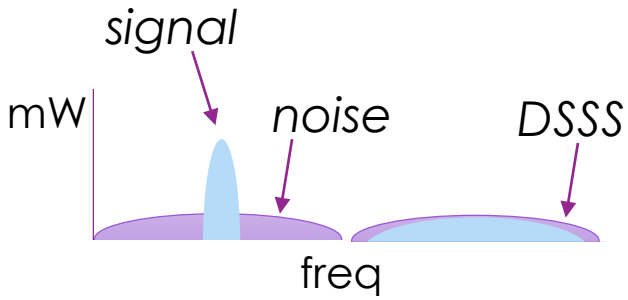
Acronym soup

- OFDM – Orthogonal Frequency Division Multiplexing

- MIMO – Multiple input, multiple output
  - Multiple antennas, beamforming (RX and TX)
  - Multiple paths, deconstructed interference, *voodoo magic*



- DSSS – Direct Sequence Spread Spectrum
  - Related to *Frequency Hopping Spread Spectrum*
  - Uses codes across a frequency band (CDMA)
  - Encoded 1b/10b - or even 1b/10,000b



Do not memorise, just enjoy…

Standards – IEEE 802.11

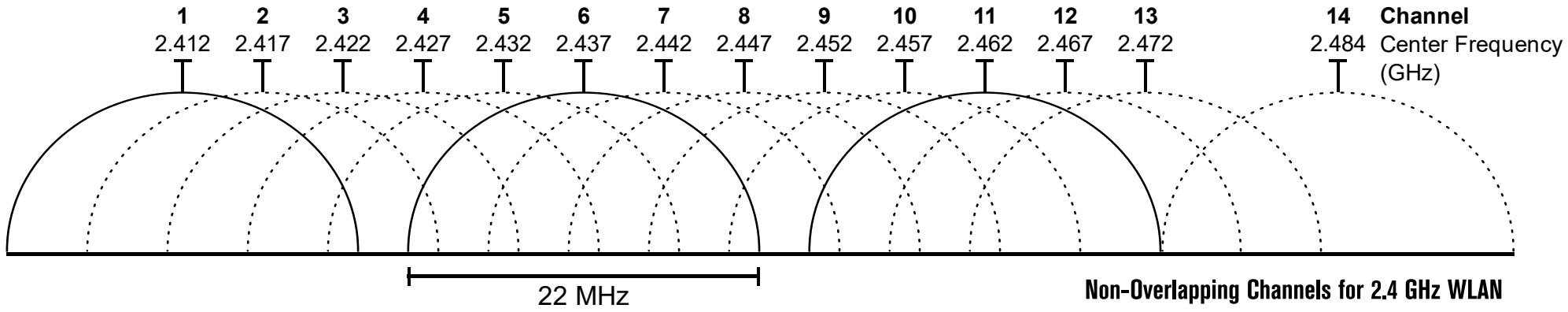
802.11	Hz	Bandwidth MHz	Datarate Mb/s	Range (m)	
a	5G	20	6-54	30-150(*)	
b	2.4G	22	1-11	30-150	
g	2.4G	20	6-54	30-150	
n	2.4G/5G	20-40	<600	70-250	
ac, ax	5G (2.4G)	20-160	<3500	30	
ad, ay	60G	2 or 160	<6600-10,000	3, 10	*WiGig*
af, ah	0.05-0.9	1-16	30-300	100-1000's	*TV, HaLow*

And another 30+ of "bonus features"…

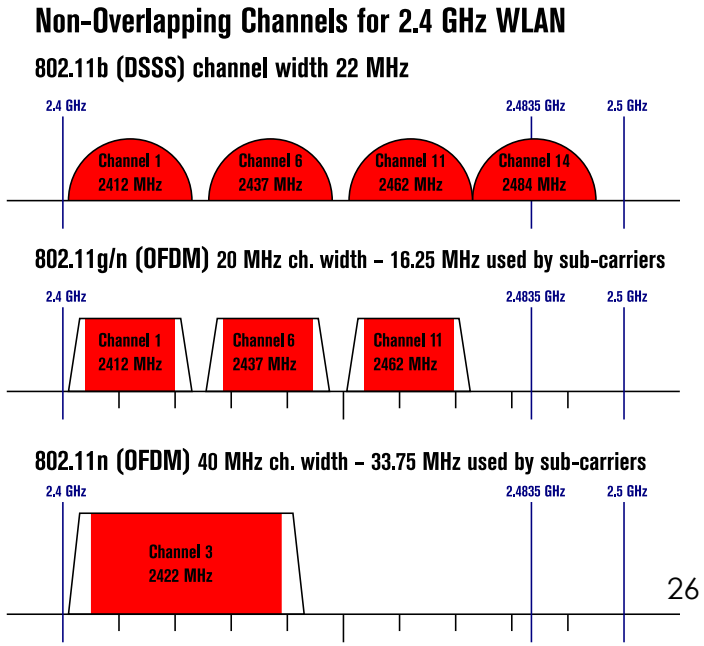
Those ranges

- Assume **area-coverage** is what you want
- Can get more directional and longer range with ‘cantennas’…

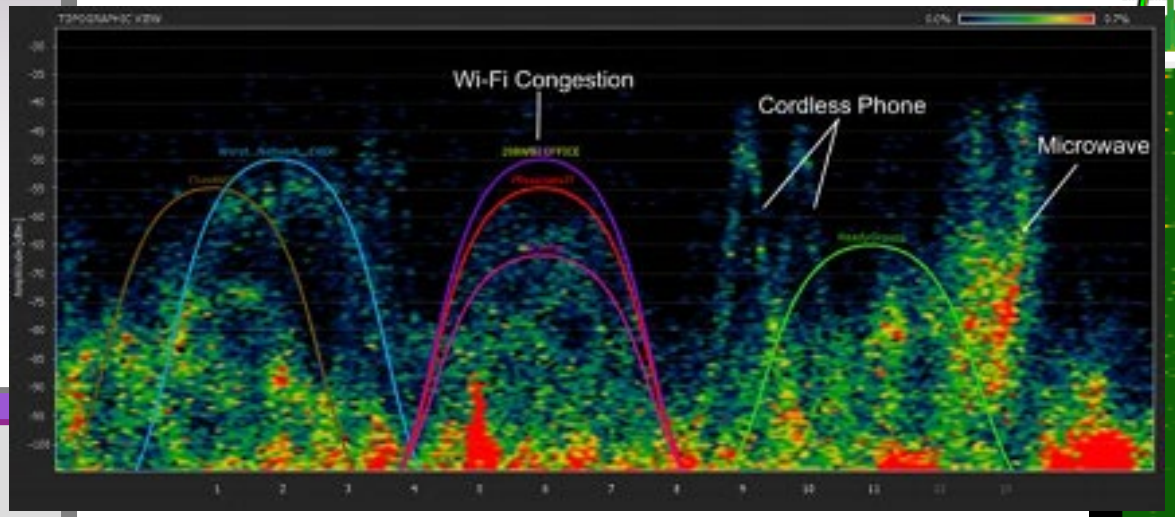
Channels @2.4GHz

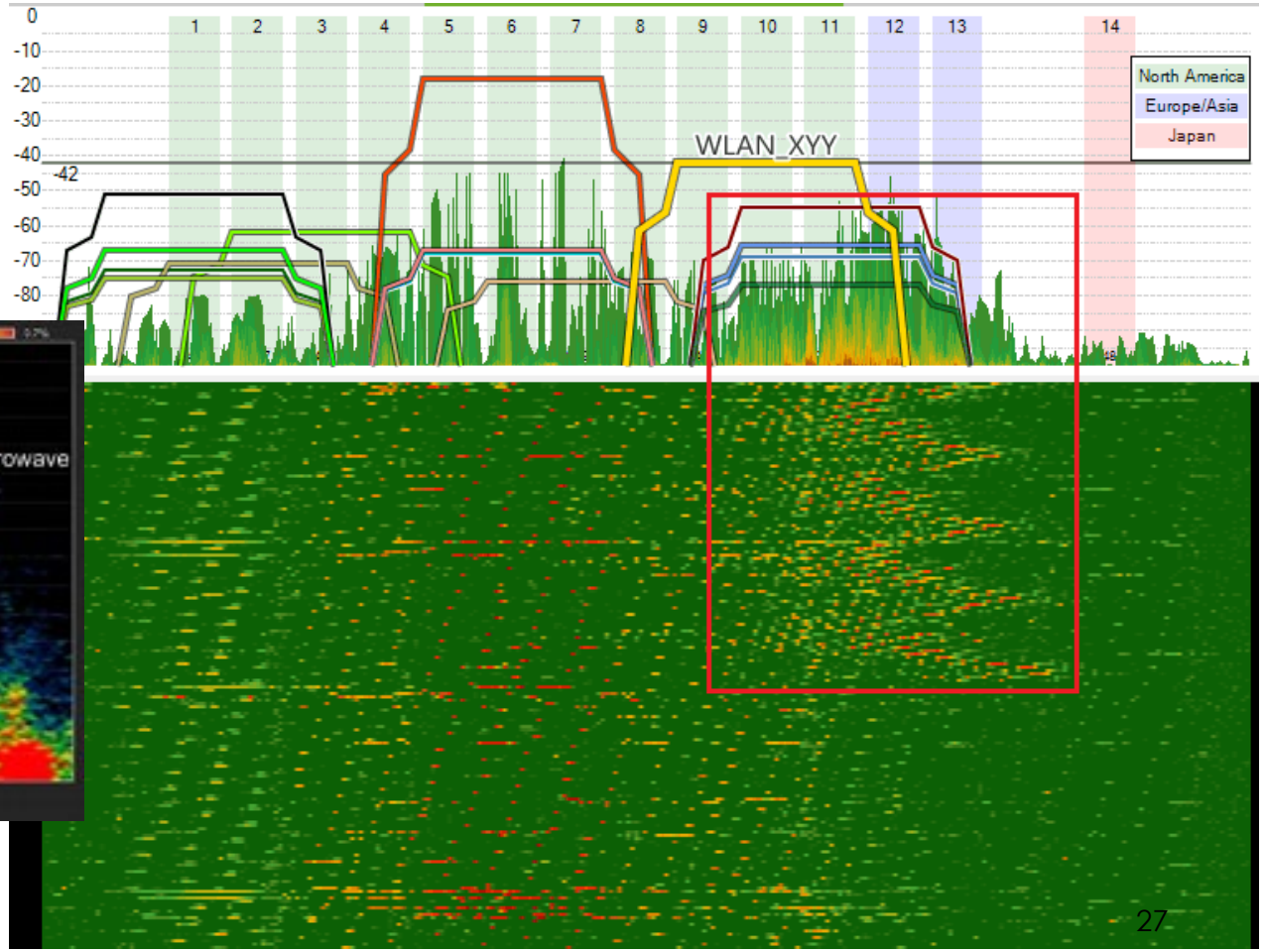


- TV channels – tune to central frequency
  - Not all channels in all countries
- (most) Channels overlap
- Channels taper at edges
- Different 802.11 interoperate – by stopping!

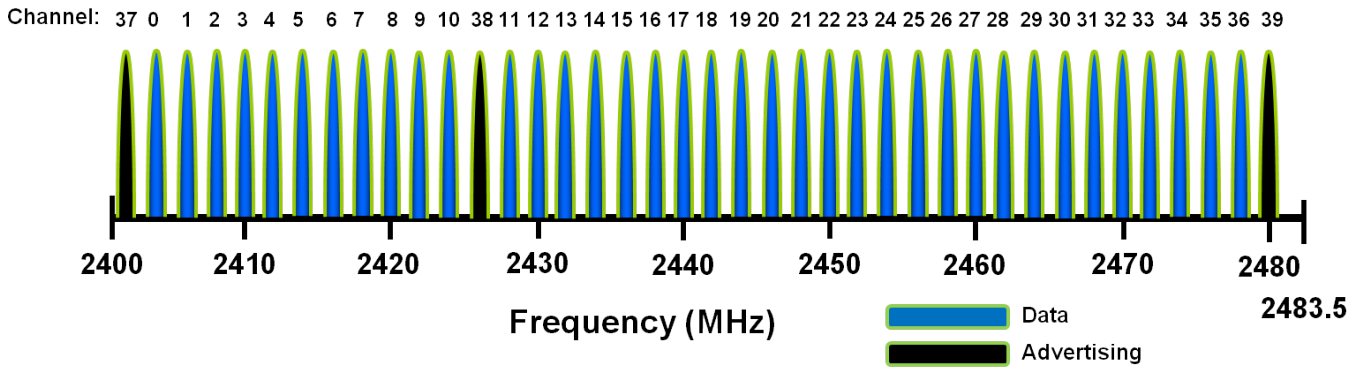


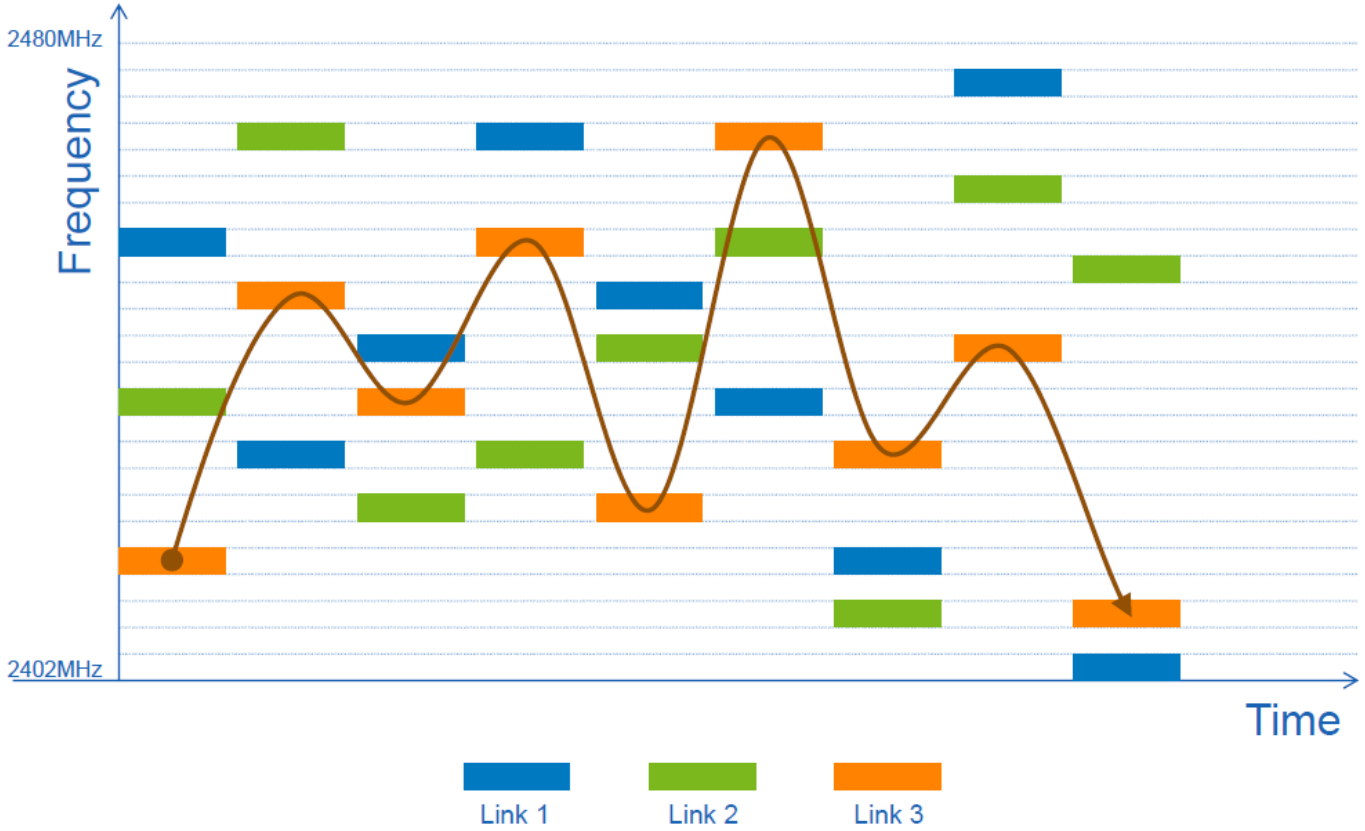
Microwaves, cordless phones, …





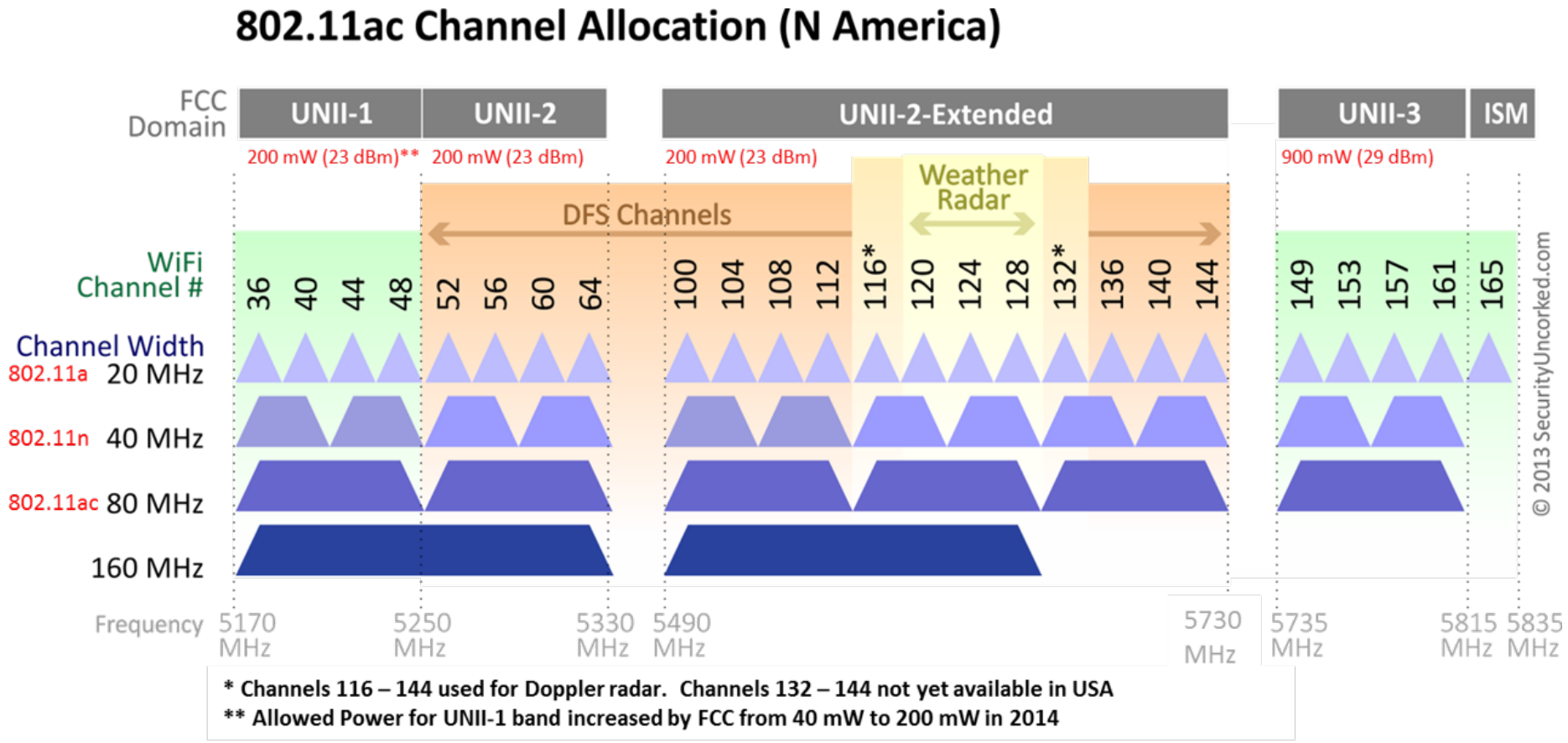
Bluetooth...





FHSS, 0.6ms timeslice

802.11 WiFi Channels @5GHz



802.11 Frames

X Mb/s *Y Mb/s*

Preamble	Start Frame	PLCP	**Header**	**Payload**	Checksum
7/16 byte	1 byte	6 byte	30 byte	**0-2312 byte**	4 byte

Frame Control	**Duration/ Connection**	**Addr 1**	**Addr 2**	**Addr 3**	**Seq Ctrl**	**Addr 4**
2 byte	2 byte	6 byte	6 byte	6 byte	2 byte	6 byte

PLCP = Physical Link Convergence Protocol – (rate, checksum, length)

802.11 Frame-types

Frame Control	Duration/ Connection	Addr 1	Addr 2	Addr 3	Seq Ctrl	Addr 4
2 byte	2 byte	6 byte	6 byte	6 byte	2 byte	6 byte

- All those addresses… src, dest, AP and 'other'
- Frame Control:
  - Control Frames
    - Control the communication with the Access Point
  - Management Frames
    - Manage the relationship with the Access Point
  - Data Frames
    - Send data…

Reliability

- LANs should be simple
  - LANs should not do overly-smart things
- But: LANs should perform efficiently, effectively

- Who takes care of errors?
  - Defined as 'failure to get through correctly' – for multiple reasons

- Three approaches:
  - Detect errors and drop frames *(something else will take care of it – 802.3)*
  - Detect errors and fix frames at receiver *(forward error correction)*
  - Detect errors and sender sends again *(Automated Repeat reQuest – ARQ – 802.11)*

ARQ by ACK

- For every frame I send, receiver should ACKnowledge receipt
  - As long as it arrives correct!
  - If they don't ACK, within a timeout, send it again
- What happens if the ACK is lost?
  - Send again, but flag it's a resent frame
- What happens if timeout is too short?
  - Send again, but flag it's a resent frame
- Stop-and-wait ARQ
  - Helps with high delays
  - Single-bit sequence number (alternate 0,1)
  - ACK includes that sequence number
- Robust, but **throttles** performance as (bandwidth*delay) goes up

802.11 Control Frames

- Request To Send (RTS)
- Clear To Send (CTS)
- Acknowledge (ACK)

- *Request for RTS (RRTS)*
- *Data Sending (DS)*

Client by association

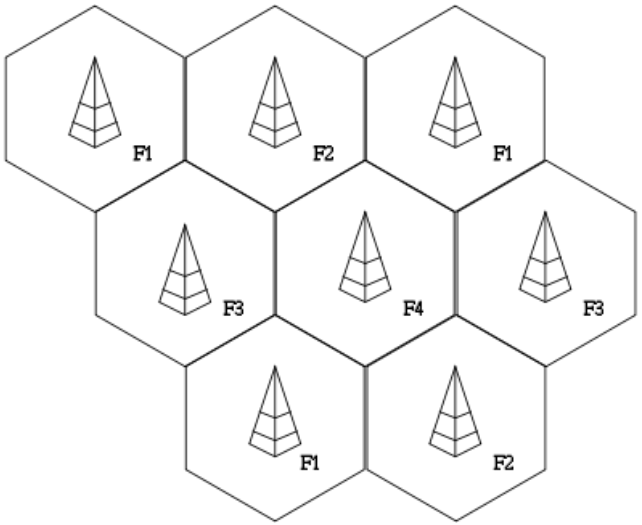
- Need to know what's available:
  1. SSID (service set identifier) – a wireless (V)LAN
  2. Access Points that accept connections for that SSID
- So either:
  - Listen for AP's offering services, or
  - Call out for AP's offering services
- Identify who you are (authentication)
- Associate with an AP (resource allocation)
- And then keep it running, while everything changes…

802.11 Management Frames

- **Beacon**
- **Probe Request** and **Probe Response**

- **Authentication** (open or shared-key)
  - **Deauthentication**

- **Association Request** and **Association Response**
  - **Disassociation**
  - **Reassociation Request** and **Reassocation Response**



What's out there

- **Beacon**: (broadcast) *I'm an AP and can offer these SSIDs at these rates in these frequencies with these standards and …*

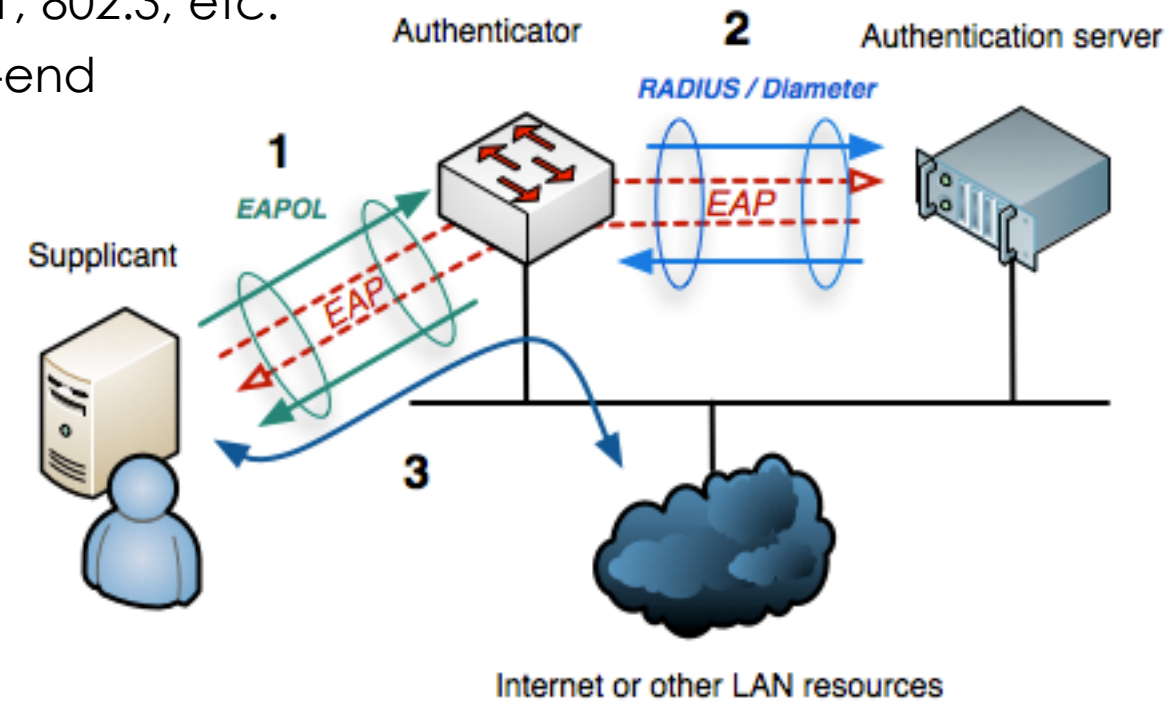
- **Probe request**: (broadcast) *I'm a client, what can you offer?*
  - **Probe Response** (targeted): *I can offer these SSIDs …*
  - Can also **Probe request** '*do you offer SSID X?*'

Authentication

- Open
  - Laptop: "Hi, I'm Markus' laptop"
  - AP: "Welcome, Markus' laptop"
- Shared key
  - Laptop: "Hi, I'm Markus' laptop"
  - AP: "Sure you are. Encrypt the following with our shared key"
  - Laptop: "Here you are…"
  - AP: "Ok, welcome Markus' laptop"
- Not that it actually cares about your identity…
  - Can also have username/pw, MAC filters, Wireless Protected Setup (WPS), …
- Deauthentication
  - Either direction

Pass-through authentication

- **802.1X**
  - Uses *Extensible Authentication Protocol* (EAP)
  - Can be used on 802.11, 802.3, etc.
  - Typically RADIUS back-end
    - Keep PW's off AP's



Encryption – everything is sniffable

- Wired Equivalent Privacy WEP
  - Don't go there. Single key, easily calculated from traffic sniffing.

- WiFi Protected Access WPA
  - With Pre-shared-key (PSK)="personal" or 802.1X="enterprise",
  - Temporal Key Integrity Protocol (TKIP) – per-frame-key
  - Better integrity checks than simple CRC
  - Heaps better. Still broken, largely through WPS ("easy-to-join" feature)

- WPA2
  - Lots of additional measures. Much stronger encryption and other protections. KRACKed (2017)
- WPA3
  - Warm off the press (Jan 2018)
- Question around what in a frame is encrypted/protected, and when – some info leaks.
  - My neighbour is talking to an interesting site?

If nothing else: WiFi and others in 2.4G

- Get on to 5G!
- Collision with Bluetooth and microwave ovens and DECT phones and …
  - Note 802.15.2: WG on "bluetooth and wifi co-existence" – is 'hibernating'.