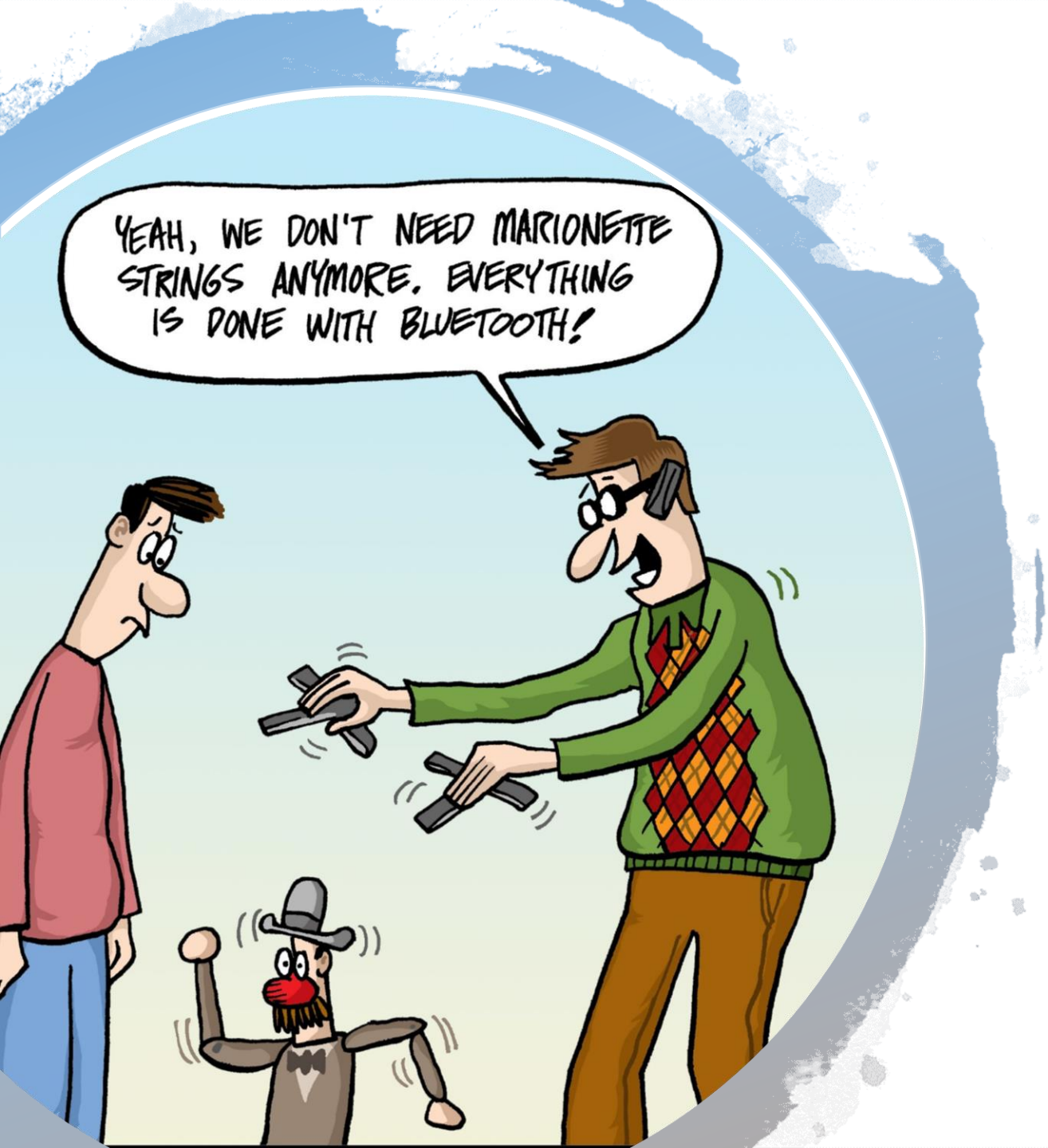
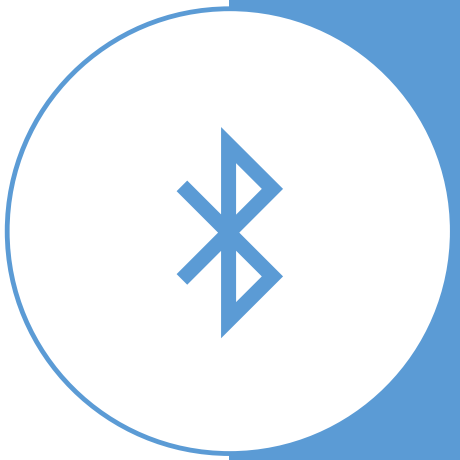


IoTSSC

Bluetooth

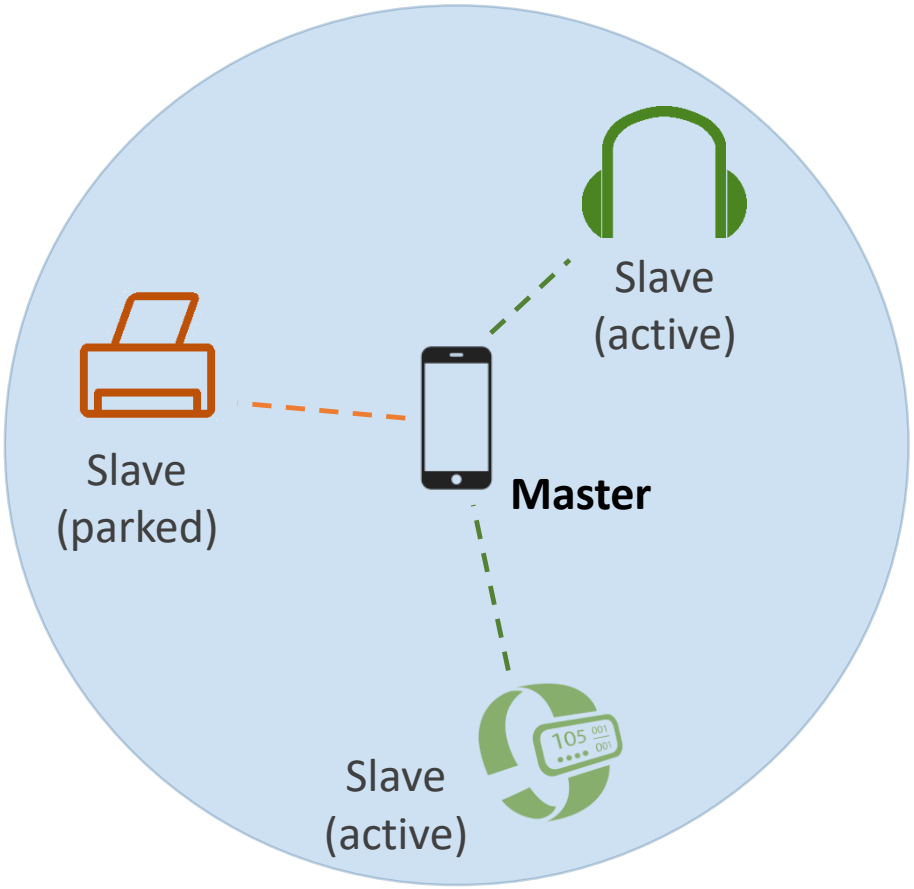
Bluetooth Classic (Basic Rate – BR/Enhanced Data Rate – EDR)

- In 1990s Ericsson wanted to connect other devices to mobile phone without cables
- Established a consortium which accumulated over the years 20k members (!) of different levels
- Standard recently advanced to version 5.2 (January 2020!), but many devices still talk Bluetooth 2.0
- Key features: enables a range of devices to connect to each other (pairing) and (securely?) transfer data between them.



Bluetooth architecture

- Typical range ~10m, which makes BT a wireless personal area network (WPAN) technology
- Basic network unit called **piconet**
- Master <-> slave architecture (up to 7 active slave devices)
- Up to 255 ‘parked’ nodes - low power state, only respond to activation from master



Bluetooth architecture

- Centralised communication paradigm (Time Division Duplex) – Master tells slaves when to talk.
- Master also controls a clock and keeps slaves synchronised.
- This means Slaves can stay pretty simple (hence cheap implementation).
- Direct slave-slave communication not possible.

Protocol stack

Current core specification over 3,000 pages.

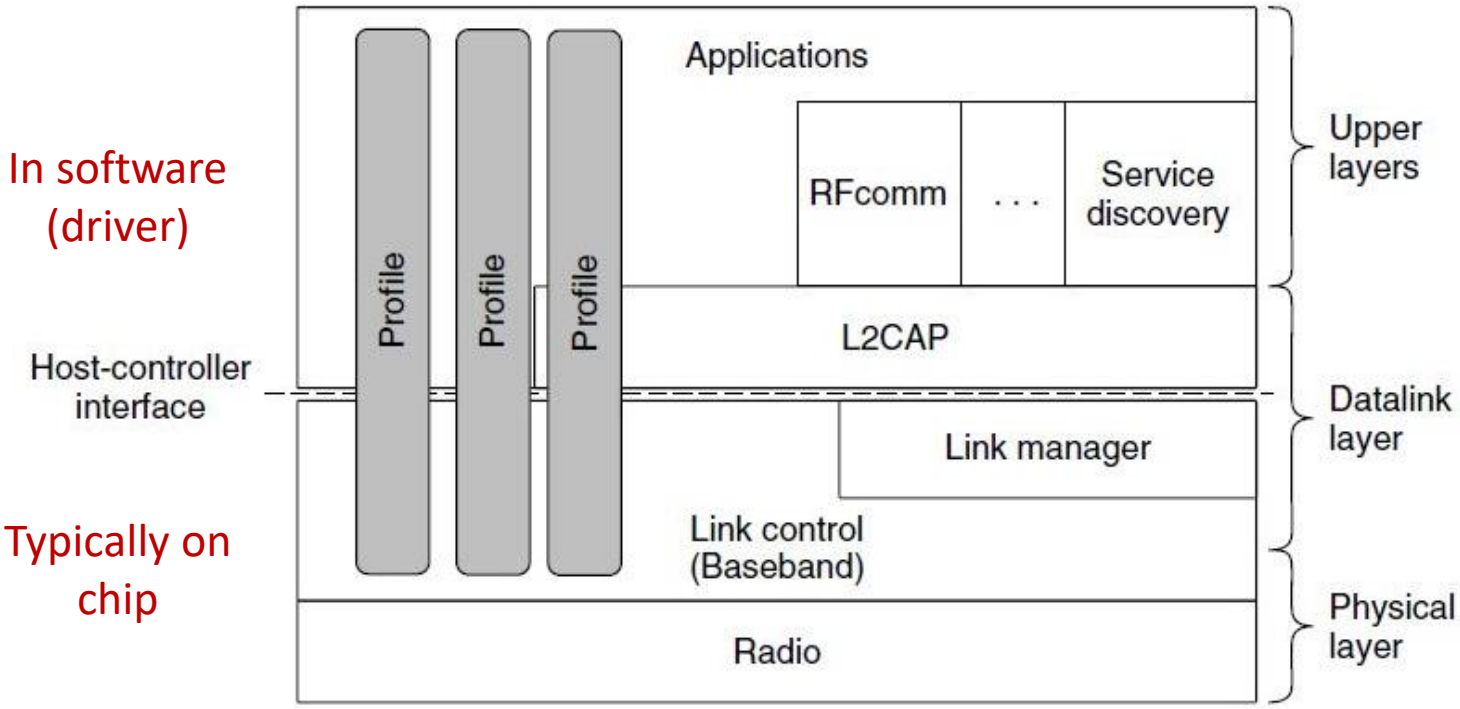
Not following the OSI or TCP/IP reference models.

Different protocol stacks for different applications (profiles) – 36 in total (not including Bluetooth Low Energy!).

Some layers present in all and there are many similarities.

Some profiles act as building blocks for others – for instance the Generic Access Profile (GAP) enables connection establishment between master/slave

Protocol stack

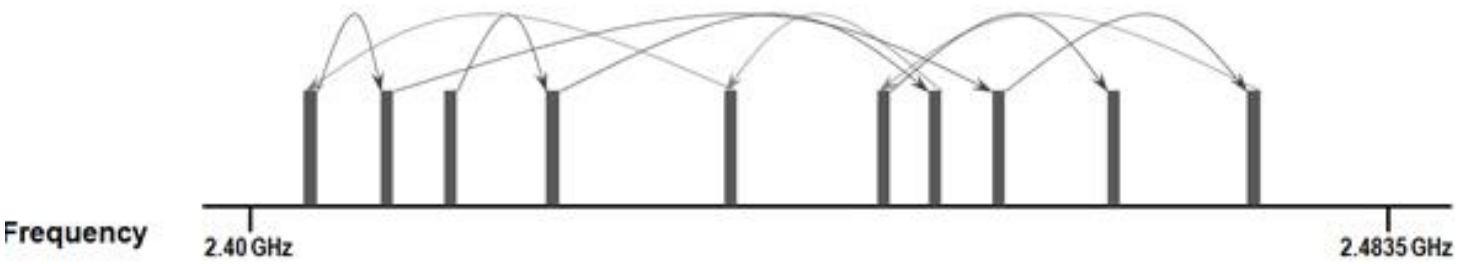


*A.S. Tanenbaum and D.J. Wetherall Computer Networks (5th ed), 2011.

- Physical radio layer quite distinctive (we will see why shortly)
- Link control = MAC+PHY (controlling timings, slot grouping)
- Link manager establishes logical channels (pairing, encryption)

Radio Layer

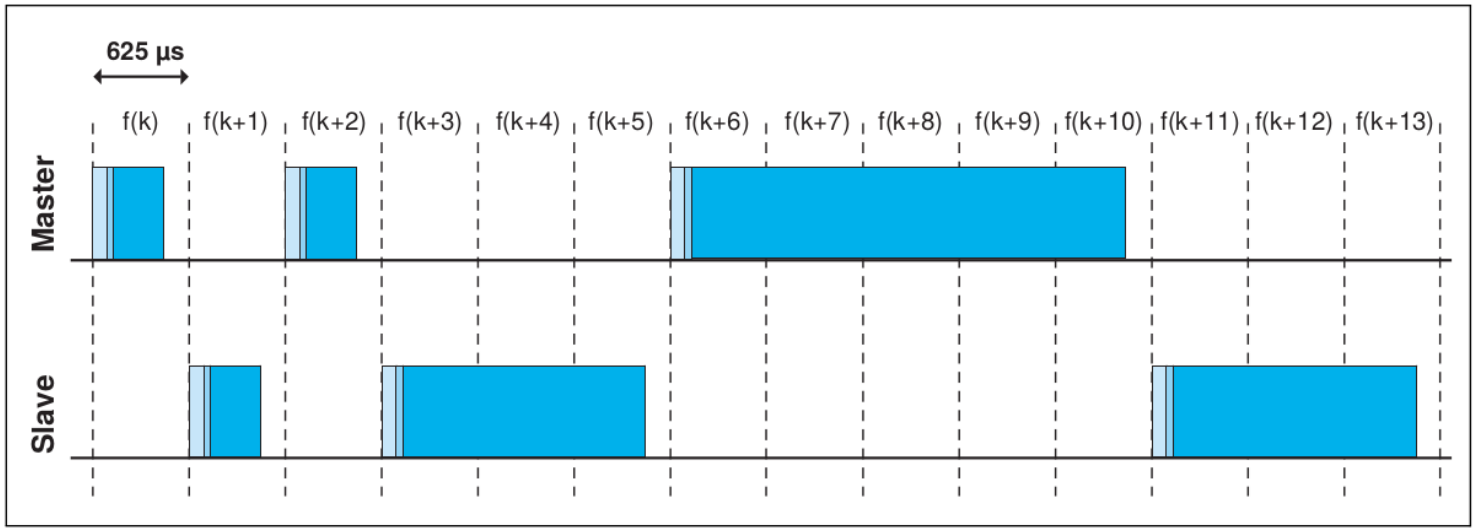
- Bluetooth operates in the 2.4GHz ISM band
- This is unlicensed but shared with other applications (Wi-Fi, baby monitors, microwave ovens, etc.)
- To ensure robustness to interference, signals are transmitted using a technique called Frequency Hopping Spread Spectrum (FHSS)



- Each transmission takes place on a different channel, peers switch rapidly between them

Radio Layer

- 79 channels of 1MHz width, up to 1600 hops/sec
- Pseudo-random hopping sequence dictated by master
- Derived from the master clock and (part of the master device address), following a set of XOR and permutation operations – some confidentiality!
- Slot duration: 650us. A packet may occupy 1, 3, or 5 slots.



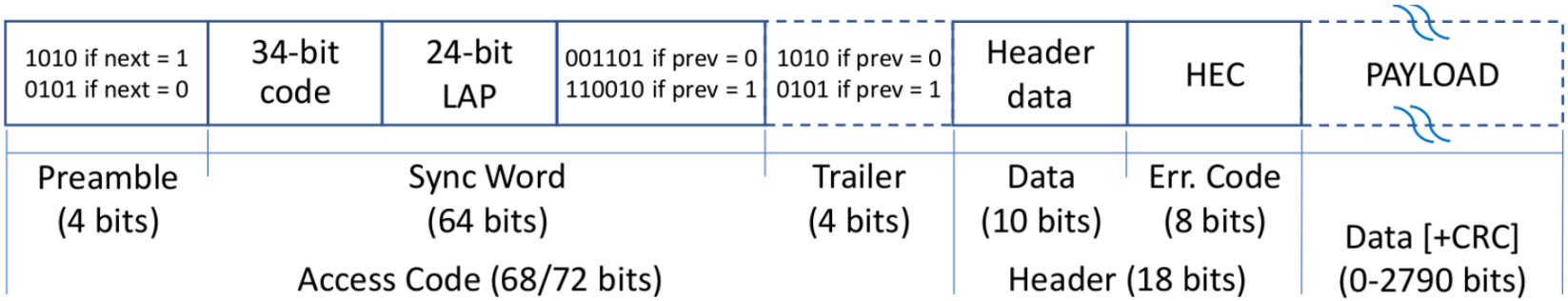
Radio Layer

- NB: Carrier frequency does not change during a single frame transmission
- Prior to transmission, information is modulated using Gaussian Frequency-Shift Keying
- This is similar to frequency modulation (where the frequency is changed with each symbol period), but a Gaussian filter is applied to data pulses, to make the transitions smoother and reduce side-band power (i.e. less interference to adjacent channels).
- Data rate is 1 Mb/s
- 2 and 3Mb/s also supported, but the modulation employed for these is differential quadrature phase-shift keying (symbols differentially encoded using phase shift)

Link layer

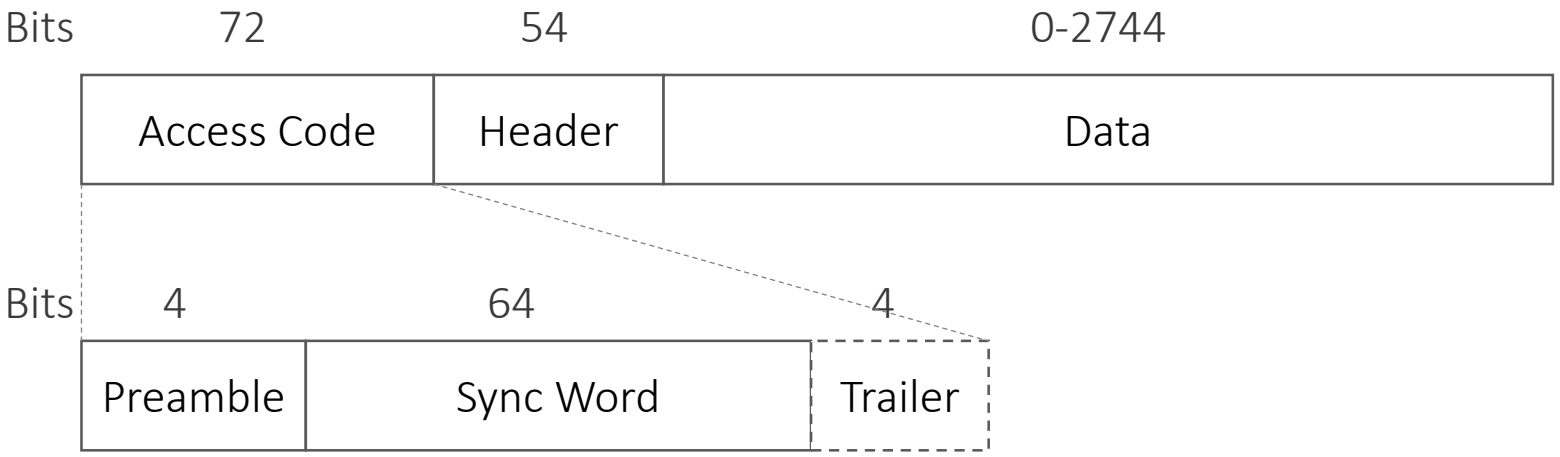
- Data preceded by a 72-bit Access Code and 54-bit Header always transmitted at the basic rate (1Mb/s)
- 16-bit CRC computed on payload
- Payload and Header scrambled with a ‘whitening’ word (linear feedback shift register initialised with portion of master clock) - the idea is to avoid long sequences of all zero/one bits

Bluetooth frame format



- Preamble (4 bits)
- Sync Word (64 bits)
- 18-bit header (transmitted 3 times, hence 54 bits)
- Payloads are optional (some frames used for discovery/control)
- Preamble together with the Sync Word (and Trailer) form the Access Code, not subject to any encoding (LAP appears in clear).

Access Codes

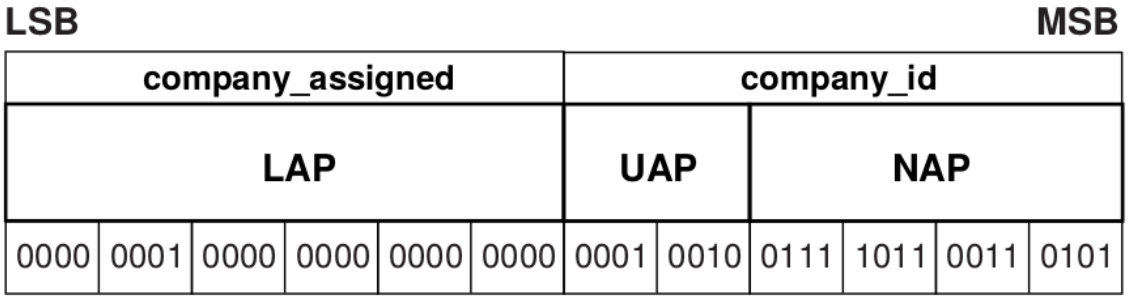


Access codes used for synchronisation and are of 4 types:

1. Channel Access Code (CAC) – used to identify piconet
2. Device Access Code (DAC) – used for signalling
3. Inquiry Access Code (IAC) of two types: general and dedicated

Sync words

- First you need to know how a BT device is identified
- 48-bit device BD_ADDR with lower, upper, and non-significant address parts



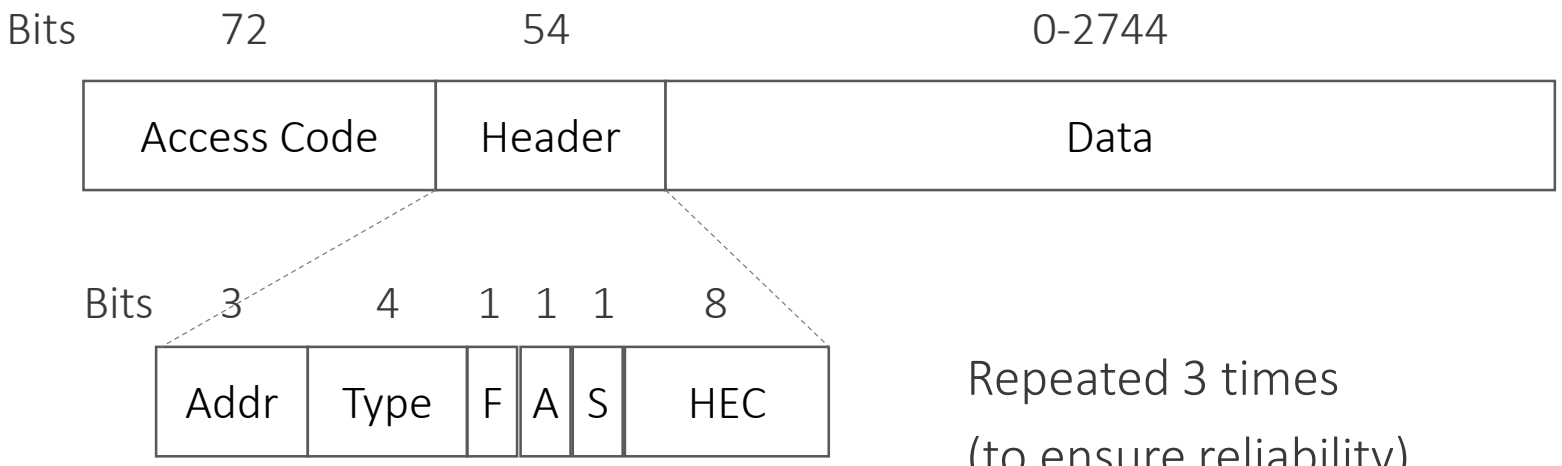
- LAP specific to the device, but 64 of these are reserved (1 for general, 63 for dedicated inquiries)

Sync words

- LAP: 0x9E8B33 used for general inquiries
  (i.e. discovering devices in range)

- Synch words build with
  - The LAP (most of the time of the master)
  - A Barker sequence appended to that (6 bits added)
  - (roughly speaking) XOR with a known 64-bit PN sequence

Bluetooth header



Repeated 3 times

(to ensure reliability)

- **Addr** identifies to which of the 8 actives devices the frame is sent
- **Type** identifies frame type, type of FEC used, and how many slots will be used to transmit the frame
- **F** (flow) – signal the slave's buffer is full
- **A** (acknowledgement) – piggybacked on a data frame
- **S** (sequence bit) – for detecting retransmissions

Bluetooth header

- **Header Error Check** - generated using a linear-feedback shift register (LFSR), whose internal 8-bit state is initialised with the master's UAP

- Header is then whitened using another LFSR whose 7-bit state is initialised with bits $c_6, \ldots, c_1$ of the master's clock (clk) and by setting the bit in position 6 to 1.

- The whitened header is then passed through a 1/3 FEC block.

Exercise

A slave wants to transmit 450 bytes of information using Bluetooth basic rate @ 1Mb/s.

How long will it take?

Exercise

A slave wants to transmit 450 bytes of information using Bluetooth basic rate @ 1Mb/s.

How long will it take?

Packet length: 72b (access code) + 54b (header) + 450*8b (payload) +16b (CRC) = 3,742b

At 1Mb/s this would require 3,742us.

Exercise

Slot size is 625us, Tx can occupy 1, 3, or 5 slots.

That is 625, 1875, or 3125us.

Packet cannot fit in 5 slots. How much info can you put into 5 slots then?

3,125b – 72b – 54b – 16b = 2,983b

BUT max payload is 2744.

Exercise

So you need another transmission for

450*8 - 2744 = 856b of data

With access code, header and CRC, this comes to 998b which is more than 1 slot but less than 3.

In one slot you can put 625 – 72 – 54 – 16 = 483b

Exercise

So we have

- First transmission 5 slots (2,744b) – 3,125us
- Master polls – 625us
- Second transmission 1 slot (483b) – 625us
- Master polls – 625us
- Third transmission 450*8-2744-483 = 373b

  Add access code, header, CRC -> 72+54+373+16 = 515b -> 515us

Total: 3,125+625*3+515 = 4,265us

Effective rate: 450*8/4,625 = 844kb/s

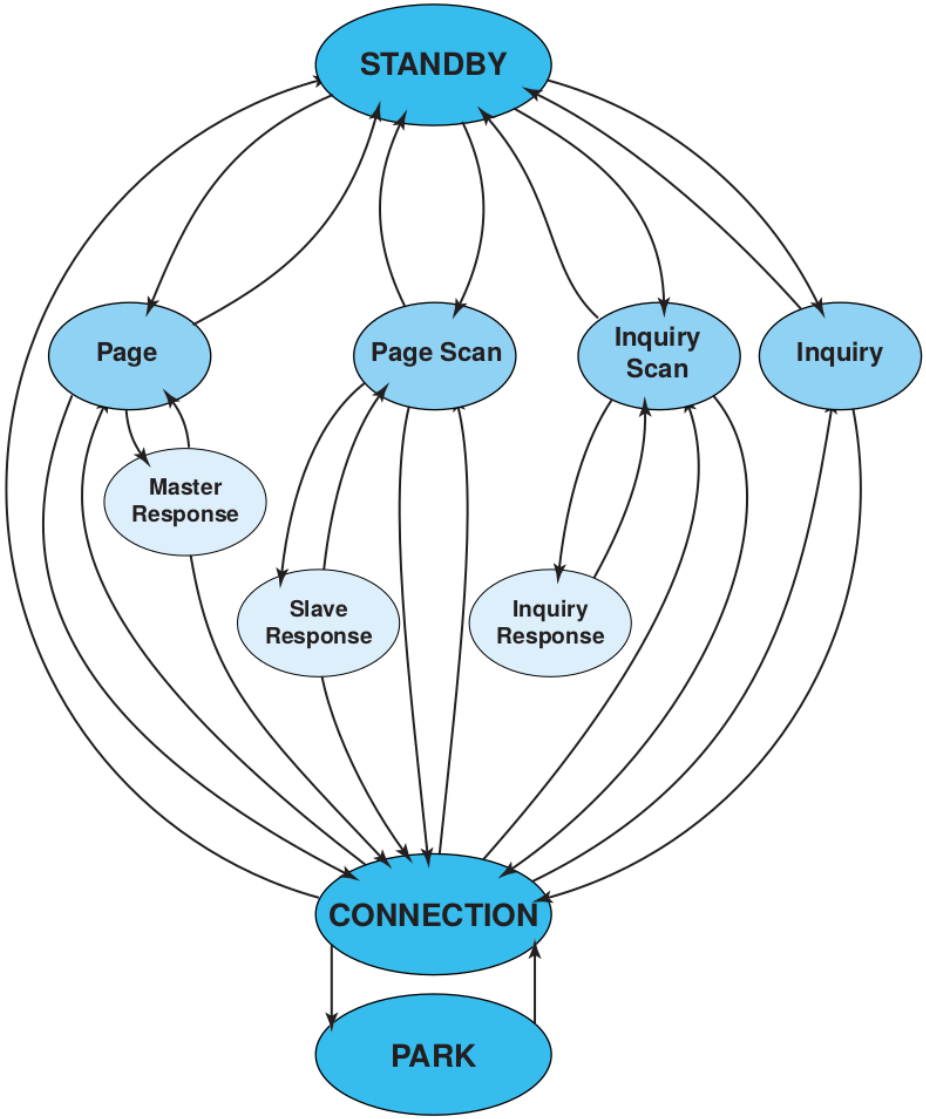
Error correction

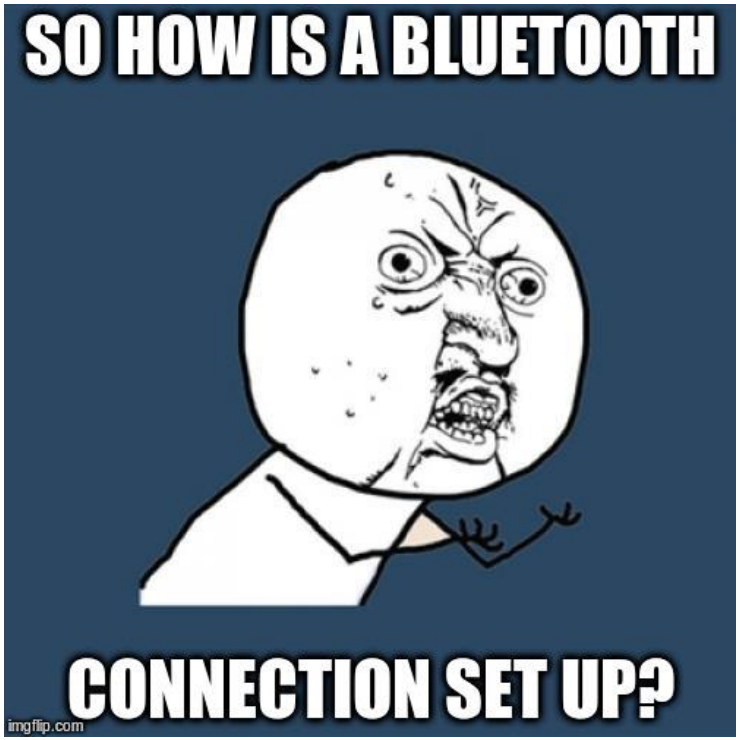
- Forward error correction (FEC) can be applied on the header and payload to increase information redundancy and robustness to errors
- FEC with rates 1/3 and 2/3 supported, that is each information bit is repeated three times and respectively packet is encoded with a polynomial that on average produces one redundant bit for every 2 bits of information.

Logical Link Control Adaptation (L2CAP)

- Performs framing (if needed), ensures reliability (if needed)
- Not all applications will used L2CAP (e.g. audio applications that send a continuous flow of samples)
- Also performs segmentation and reassembly, CRC checks, and retransmission when required,
- Default MTU 672 bytes (minimum 48 bytes mandatory)
- L2CAP determines to which protocol to pass packets

Link controller operation





Establishing a connection - inquiry

- First the master needs to discover the potential slave(s), if indeed discoverable
- A device wishing to discover other devices enters the ‘inquiry’ substate.
- Send inquiry message over 32 wake-up carriers, equally distributed over 79MHz range, hopping following a pseudo-random sequence.
- A device allowing to be discovered enters ‘inquiry scan’ substate -> listens for 11.25ms according to own hopping sequence, every 1.28s.

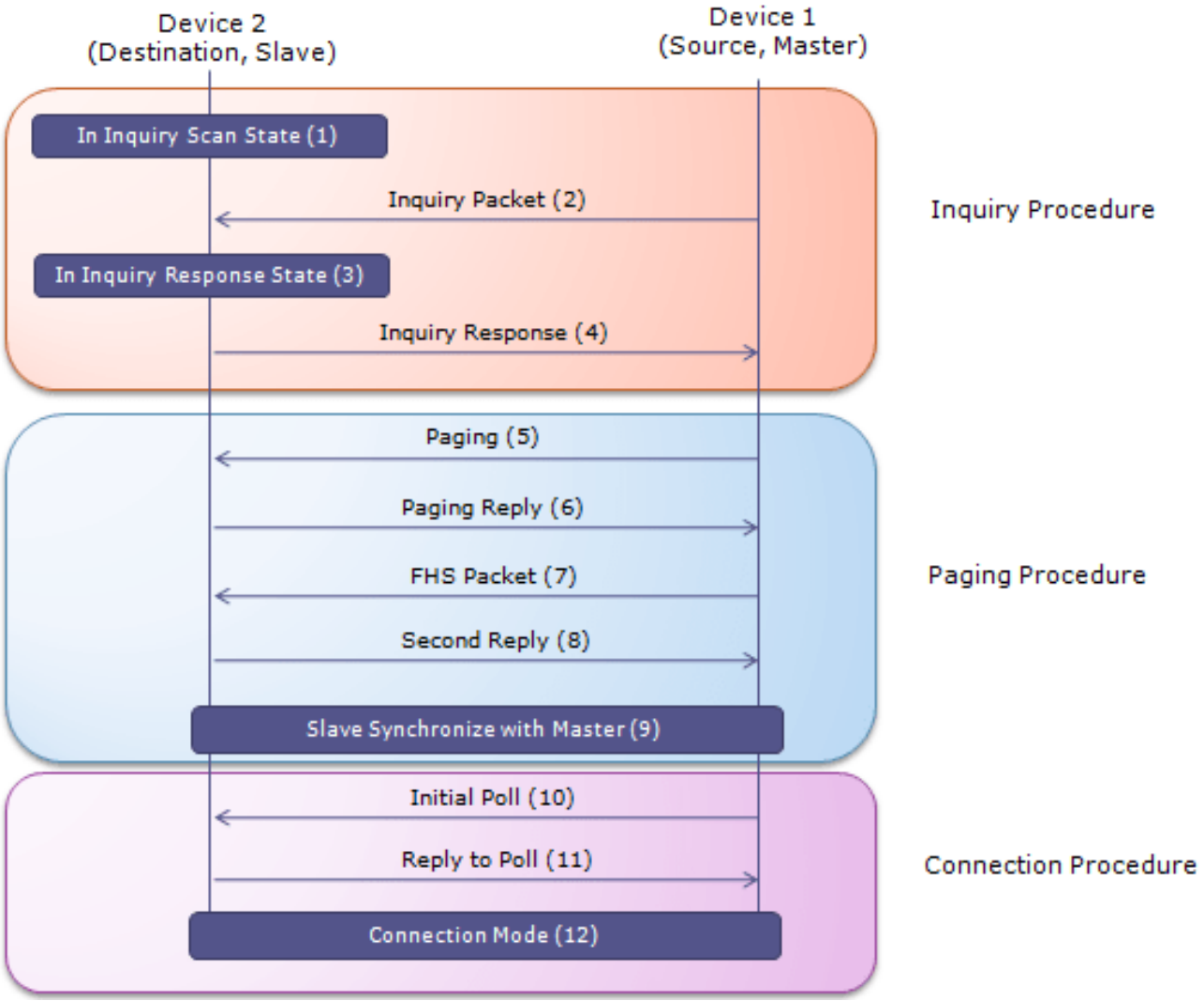
Establishing a connection - inquiry

- When receiving first inquiry packet, device remains on same channel, initiates back-off (to minimise chances of collision with other devices, when responding)
  - waits for a random number of time slots uniformly distributed in [0, 1024)
  - returns to inquiry scan mode
- Upon receiving a second Inquiry, device responds immediately with a FHS (Frequency Hopping Synchronisation) packet containing its address and clock offset, and enters ‘page scan’ substate.
- A master wishing to connect a new device enters ‘page’ substate when receiving FHS.

Establishing a connection - paging

- The master follows a hopping sequence derived from the slave's address and sends 'page' messages to potential slave it wishes to connect to
- This contains a 24-bit Device Address Code (DAC) derived from the slave's physical address
- Slave responds with 'page response' in the next slot, which contains a slave ID
- Master sends FHS, which the slave will use to compute its hopping sequence; this is derived from part of master's address and clock
- Slave confirms the receipt with second 'page response'
- Master assigns a 3-bit Active Member Address (AMA) and may send an initial 'poll'
- The connection is considered established and both move into 'CONNECTION' state.

Summarising (without details)

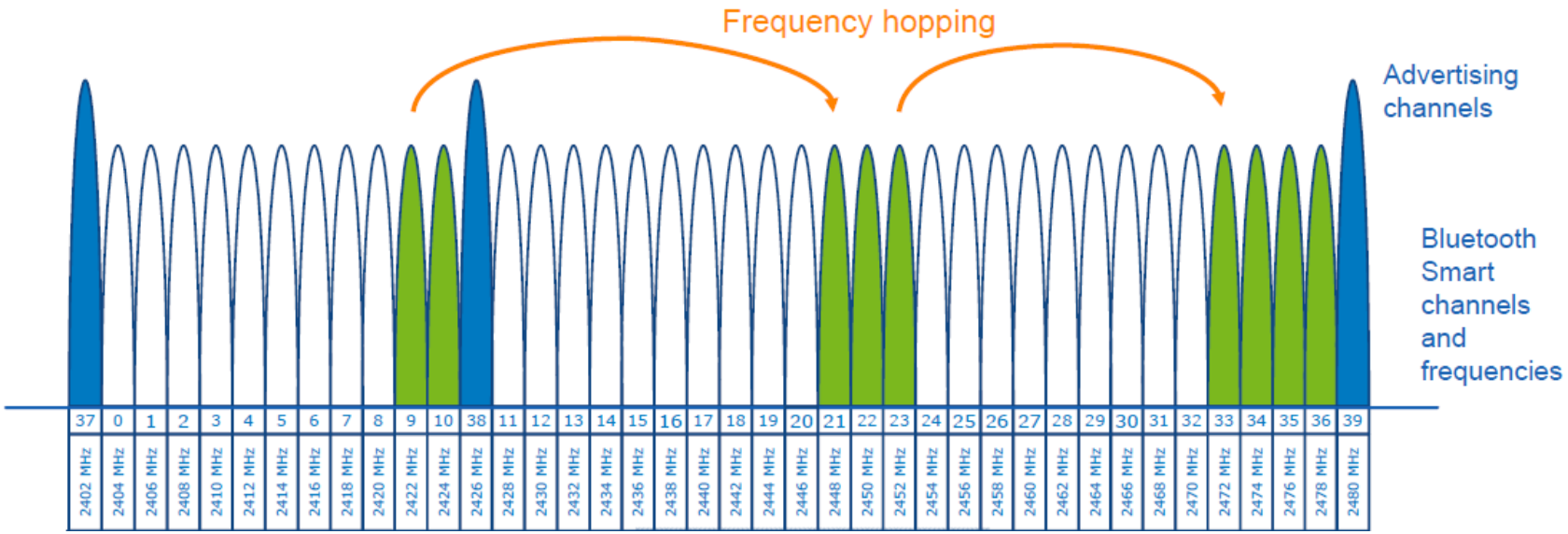


Bluetooth Low Energy (BLE)

Also known as Bluetooth Smart

Bluetooth Low Energy (BLE)

- Many similarities, but also a couple of differences
  - Master == Central; Slave == Peripheral
  - 40 channels, 2MHz wide
  - 3 of these used to continuously advertise presence – ch 37, 38, 39 (these are not adjacent in terms of frequency)



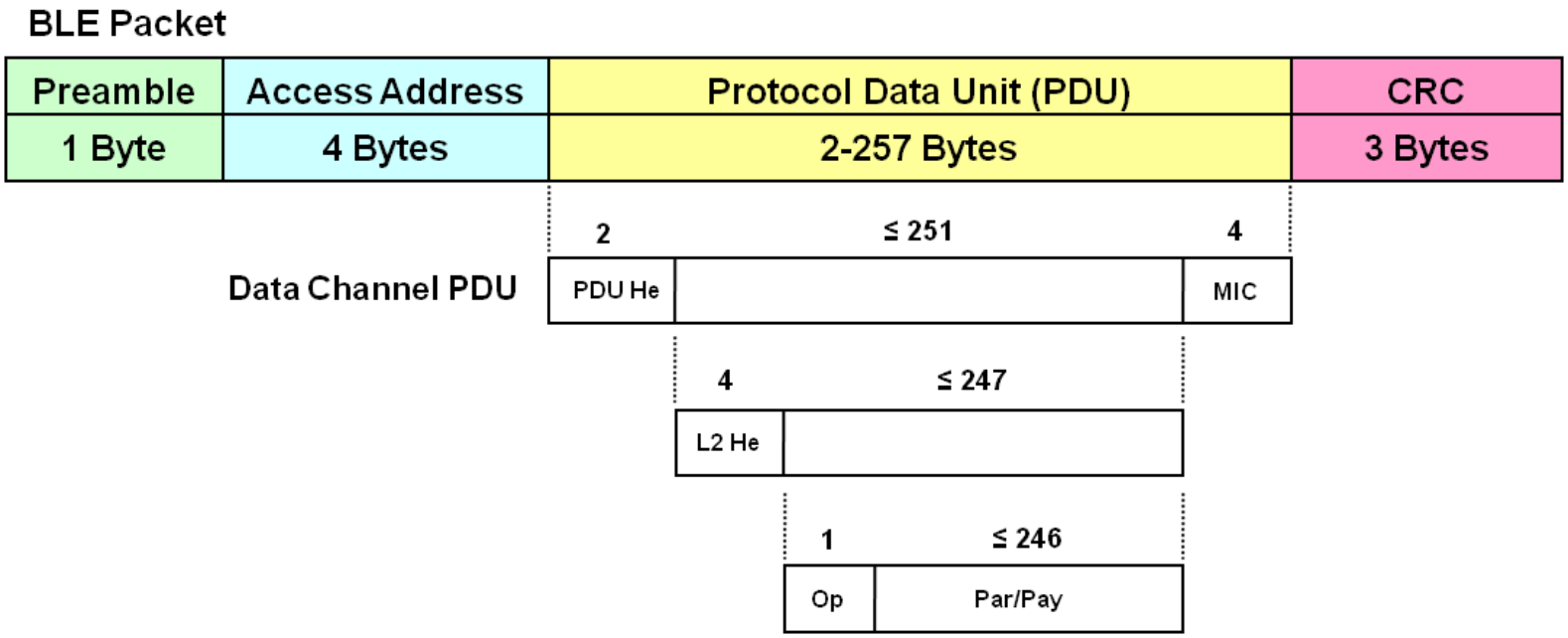
BLE connection set-up

- Device discovery much faster (periodic advertisement interval between 20ms-10.24s + a random delay between 0-10ms to avoid collisions)
- Four types of advertisement packets:
  - ADV_IND – used by peripheral to requests connection to any central device
  - ADV_DIRECT_IND - connection request directed at a specific central device.
  - ADV_NONCONN_IND - Non connectable devices, advertising information to any listening device (beacons)
  - ADV_SCAN_IND - Similar to ADV_NONCONN_IND, with optional additional information via scan responses.

BLE connection set-up

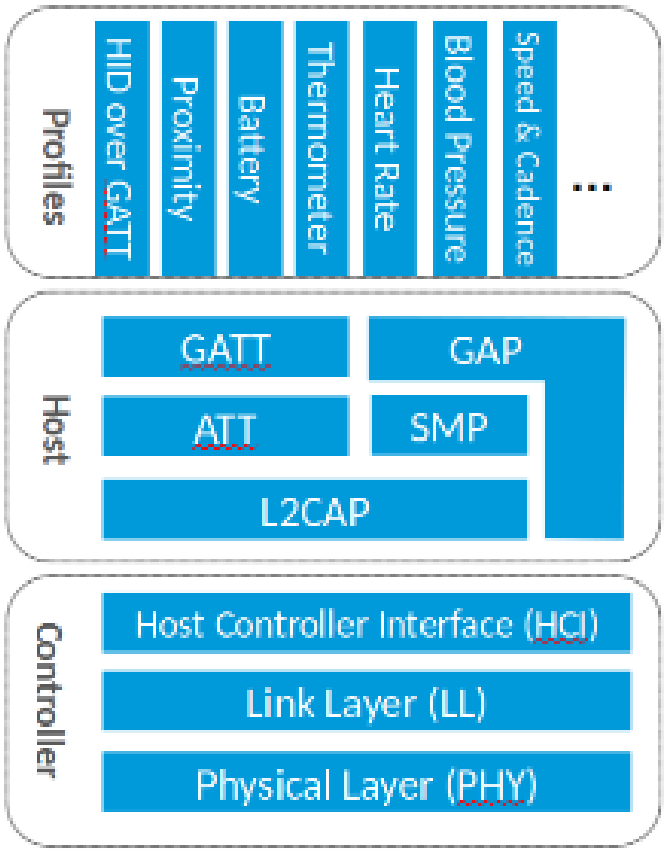
- If a connection request is accepted, a hop increment is agreed and the two peers hop accordingly, by adding increment to current channel index, modulo 37
- A channel map can also be agreed, i.e. some channels can be avoided.
- The central also assigns a (private) address to a connecting peripheral, which is generated randomly
- This access address will uniquely identify the physical channel between two devices
- Hopping interval can also be renegotiated after connection set-up (energy saving).

BLE data frame format



Note CRC is 24 bits in BLE

BLE stack



- Some new profiles
- Attribute Protocol (ATT) mandatory for all data transfers
- Generic Access Profile (GAP) controls advertisement and connections, device roles (central/ peripheral)
- Generic Attribute Profile (GATT) – set of procedures for discovering and accessing attributes (service specific)

Profiles

- Extend across protocol stack, selecting the relevant features
- Used to describe an application and collection of services offered (blood pressure monitoring, device information service, etc).
- Each profile makes use of a particular set of GATT services – user-defined profile definition allowed
- Services are collections of characteristics that define the behaviour of part of a device.
- Characteristics are attribute types that have a name, uniform type identifier and assigned number

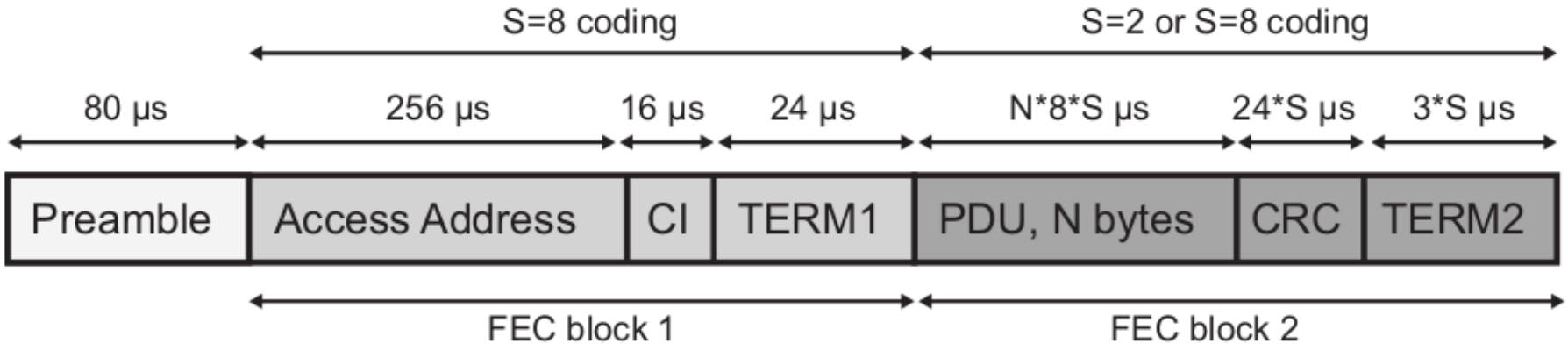
Bluetooth 5

Four physical layer modes (all GFSK based)

- LE Uncoded 1M - 1 Msym/s, 1 Mb/s
- LE Uncoded 2M - 2 Msym/s, 2 Mb/s
- LE Coded S=2, 1 Msym/s, 500 kb/s
- LE Coded S=8, 1 Msym/s, 125 kb/s

Coded packet can increase range (through message redundancy – forward error correction)

LE coded packets



- Preamble not coded
- Access address, coding indicator (CI) and first terminator (3 bits) coded with S=8
- CI indicate whether the second part of the packet is S=8 or S=2 coded.

LE coding

- FEC encoder with rate ½. Two polynomials used:

$$G_0(x) = 1 + x + x^2 + x^3$$

$$G_1(x) = 1 + x^2 + x^3$$

- Bit coming from G_0 transmitted first, then bit coming from G_1
- Pattern mapper then used to convert each bit from the encoder into P symbols (depending on scheme)

Bir from FEC encoder	Output sequence (S=2)	Output sequence (S=8)
0	0	0011
1	1	1100

Securing connections

- If the communication is intended to be secure, connection set-up is followed by secure simple pairing.
- Four association modes:
  - Numeric Comparison (same numbers shown on both devices)
  - Just Works (all zero sequence, suitable for devices with no display or input)
  - Out of band (e.g. Near Field Communication)
  - PassKey entry (6-digit pin shown on one device, keyed in on the second – e.g. keyboard)

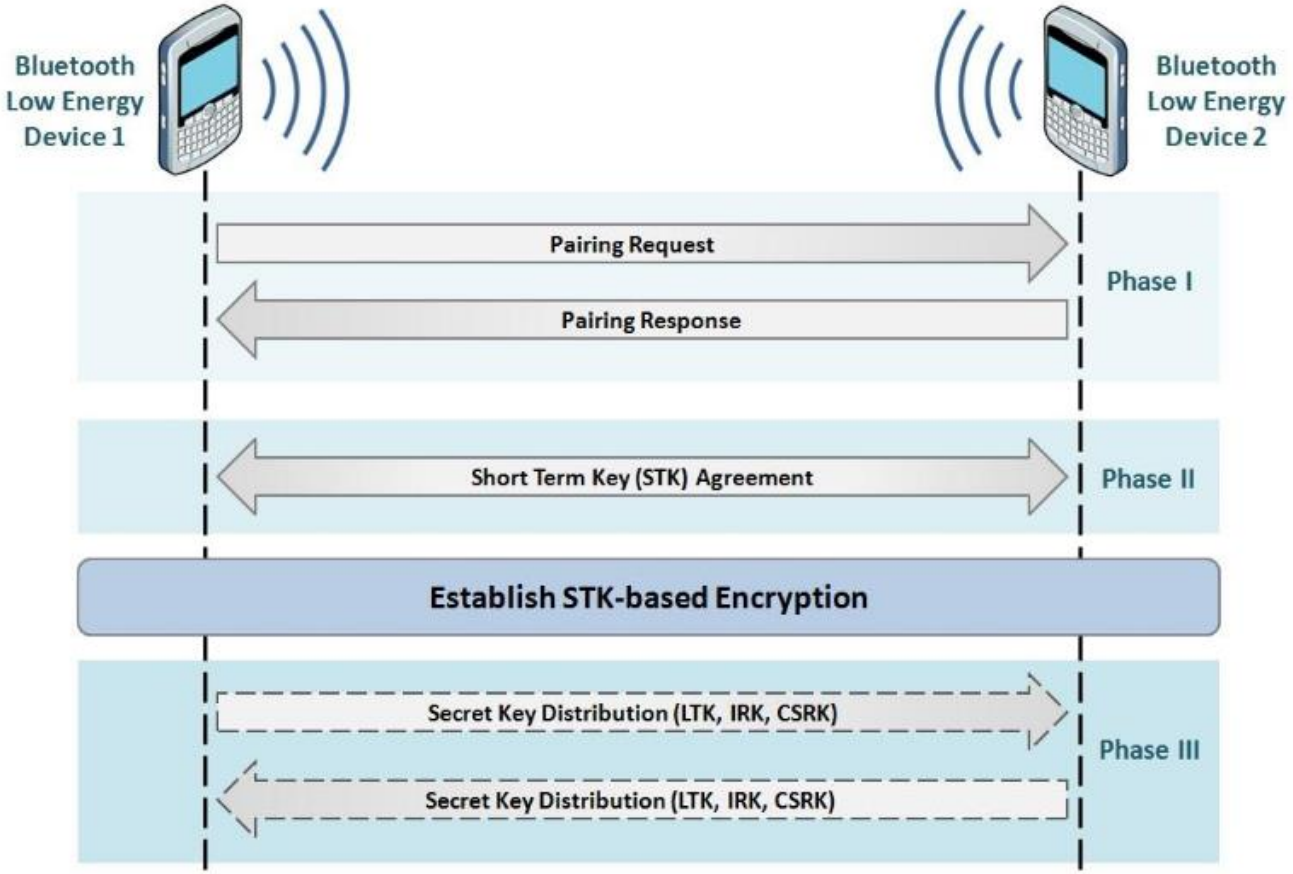
Securing connections

- Irrespective of method, this is used to generate a temporary key (TK), from which a short term key (STK) is then derived
- Knowing the numbers used during pairing does not help an attacker decrypt the data exchanged later (this was not the case with Bluetooth 2.0).
- Intercepting the initial communication may allow an attacker to brute force the TK, following which the STK may be computed as the other data is sent in clear and methodology known.

Securing connections

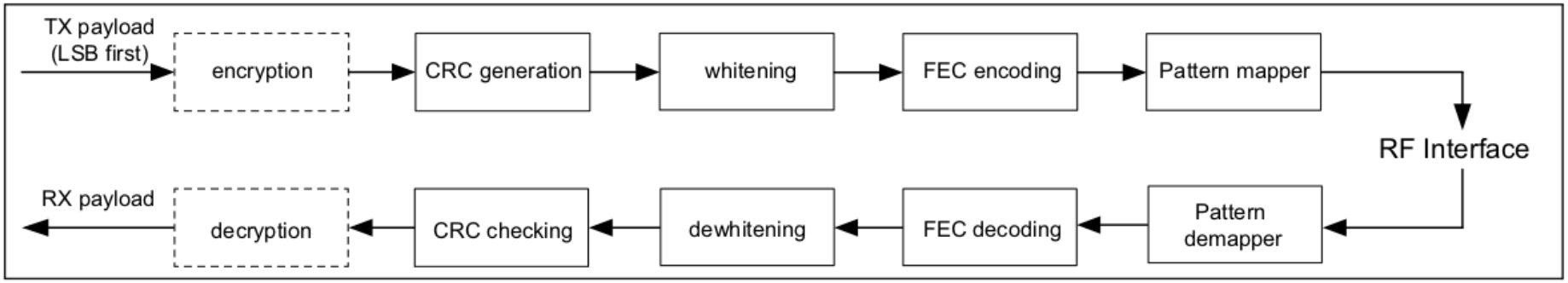
- A key distribution protocol is then used on the ‘temporarily secured’ channel to transmit a Long-Term Key (LTK), along with
- An Identity Resolving Key (IRK) to support private device addresses, and
- Connection Signature Resolving Key (CSRK), to support data signing.

Securing connections



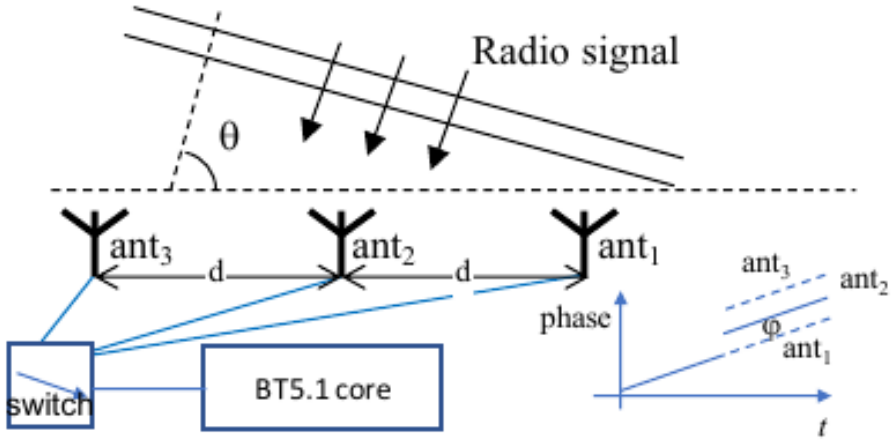
Source: NIST

Bitstream processing



Bluetooth 5.1 Positioning

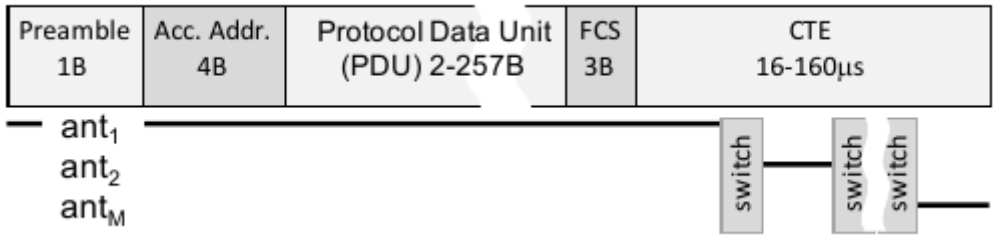
- Core idea: direction finding via phase difference between signal arrival at different antennas



- Phase signal difference: $\varphi = 2\pi(d/\lambda)\cos\theta$, where d distance between antennas, λ is the wavelength

Bluetooth 5.1 Positioning

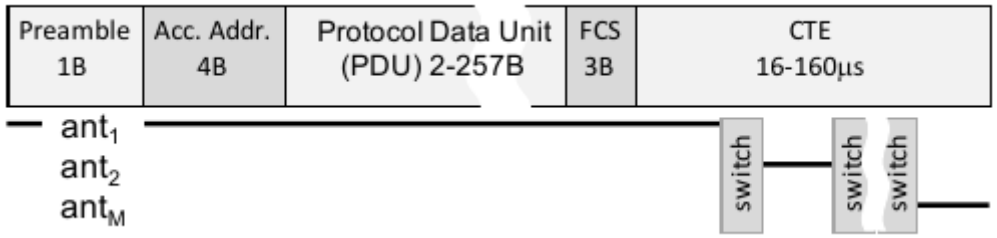
- Single receiver chain used to preserve energy efficiency -> switching between antennas
- Constant Tone Extension (CTE) follows CRC to indicate this



- Any problem with this approach?

Bluetooth 5.1 Positioning

- Single receiver chain used to preserve energy efficiency -> switching between antennas
- Constant Tone Extension (CTE) follows CRC to indicate this



- Any problem with this approach?
  - No protection of CTE -> subject to interference + can be used for malicious purposes

Further reading: http://homepages.inf.ed.ac.uk/ppatras/pub/wintech19.pdf