Bluetooth

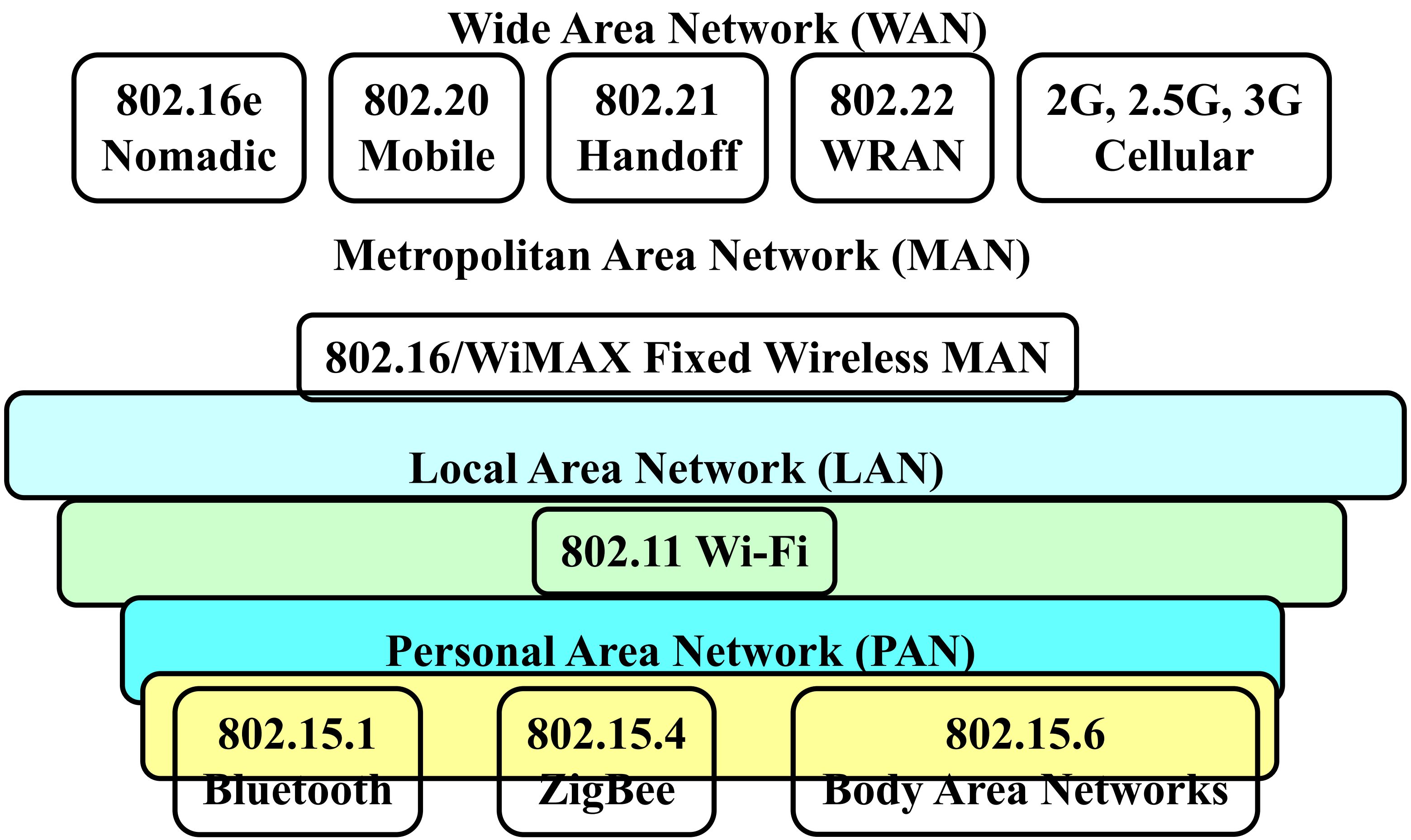
Bluetooth Classic
Bluetooth Low Energy (BLE) - Bluetooth 4
BLE Advanced – Bluetooth 5

Overview

1. Bluetooth History: Wireless Personal Area Networks (WPANs) and IEEE 802.15 projects, Bluetooth Special Interest Group (SIG), Bluetooth Versions
2. Bluetooth Markets and Applications
3. Bluetooth Classic: Network Topology, Channel Structure, Modulation and Data Rates, Frequency Hopping, Packet Format, Operating States, Power Saving, Protocol Stack, Application Profiles
4. Bluetooth Low Energy (BLE): Channel Structure, Frequency Hopping, PHY, MAC
5. Bluetooth 5: PHY, Advertising, and Frequency Hopping Extensions

Wireless Personal Area Networks (WPANs)

- 10m or less



WPAN: Design Challenges

- **Battery powered**: Maximize battery life. A few hours to a few years on a coin cell.
- **Dynamic topologies**: Short duration connections and then device is turned off or goes to sleep
- **No infrastructure:** No access point or base station
- **Avoid Interference** due to larger powered LAN devices
- **Simple and Extreme Interoperability**: Billions of devices. More variety than LAN or MAN
- **Low-cost**: A few dollars

IEEE 802.15 Projects

- **IEEE 802.15.1-2005**: **Bluetooth** 1.2
- **IEEE 802.15.4-2011**: Low Rate (250kbps) WPAN – **ZigBee**
- **IEEE 802.15.4f-2012**: PHY for Active **RFID**
- **IEEE 802.15.6-2012**: Body Area Networking. Medical and entertainment. Low power
- **IEEE 802.15.7-2011**: Visible Light Communications

Bluetooth

Bluetooth SIG → IEEE 802.15.1 → Bluetooth SIG

- Started with Ericsson's Bluetooth Project in 1994 for radio-communication between cell phones over short distances
- Named after Danish king Herald Blåtand (=Bluetooth) (AD 940-981) who was fond of blueberries
- Intel, IBM, Nokia, Toshiba, and Ericsson formed Bluetooth SIG in May 1998
- Version 1.0A of the specification came out in late 1999.
- IEEE 802.15.1 approved in early 2002 is based on Bluetooth Later versions handled by Bluetooth SIG directly
- Key Features:
  - Lower Power: 10 mA in standby, 50 mA while transmitting
  - Cheap: \$5 per device
  - Small: 9 mm^2 single chips

Example of a Bluetooth Chipset

MICROCHIP

RN4020

Bluetooth® Low Energy Module

Features

- Fully certified Bluetooth® version 4.1 module
- On-board Bluetooth Low Energy 4.1 stack
- ASCII command interface API over UART
- Device Firmware Upgrade (DFU) over UART or Over the Air (OTA)
- Microchip Low-energy Data Profile (MLDP) for serial data applications
- Remote commands over-the-air
- 64 KB internal flash
- Compact form factor: 11.5 x 19.5 x 2.5 mm
- Castellated SMT pads for easy and reliable PCB mounting
- Environmentally friendly, RoHS compliant
- Certifications: FCC, IC, CE, QDID, VCCI, KCC, and NCC

Operational

- Single operating voltage: 1.8V to 3.6V (3.3V typical)
- Temperature range: -30°C to 85°C
- Low-power consumption
- Simple, UART interface
- Integrated Crystal, I^2C Interface, Internal Voltage Regulator, Matching Circuitry, and PCB Antenna
- Multiple IOs for control and status
- GPIO, ADC
- Three Pulse Width Modulation (PWM) outputs

RF/Analog Features

- ISM Band 2.402 to 2.480 GHz operation
- Channels 0-39
- RX Sensitivity: -92.5 dBm at 0.1% BER
- TX Power: -19.0 dBm to +7.5 dBm
- RSSI Monitor

Applications

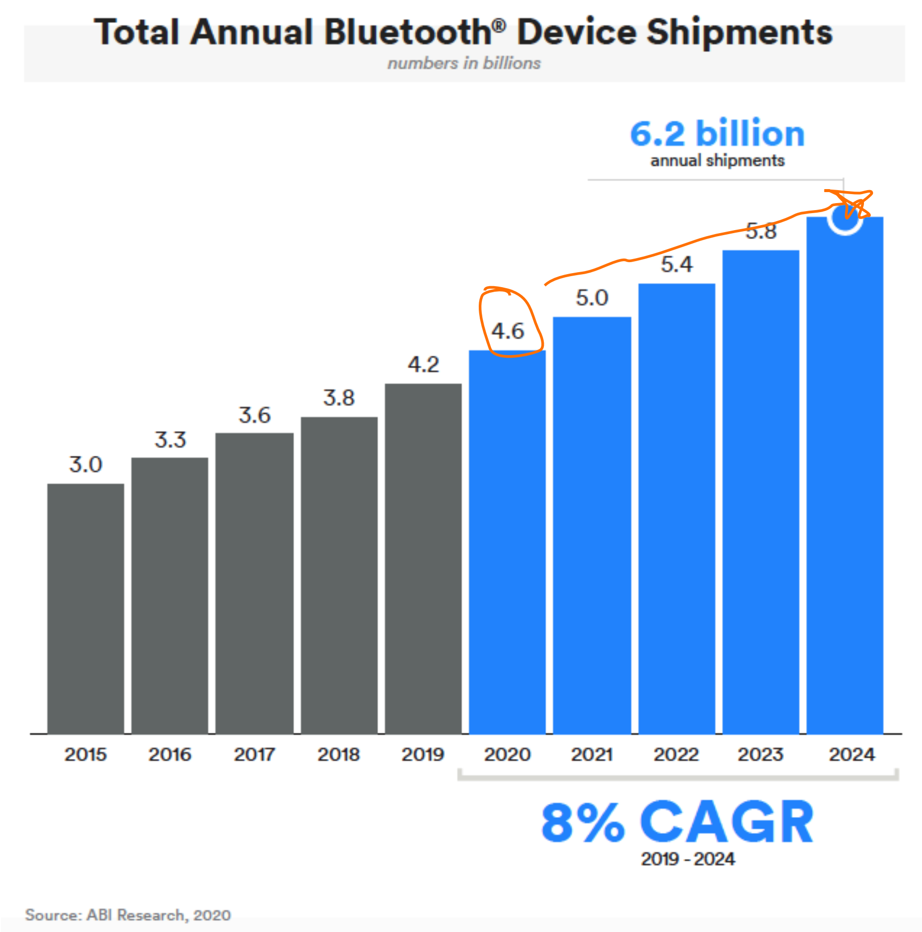
- Health/Medical Devices
  - Glucose meters
  - Heart rate
  - Scale
- Sports Activity and Fitness
  - Pedometer
  - Cycling computer
  - Heart rate
- Retail
  - Point of Sale (POS)
  - Asset tagging and tracking
  - Proximity advertising
- Beacon Applications
- Internet of Things (IoT) Sensor tag
- Remote Control
  - Embedded Device Control
  - AV consoles and game controllers
- Wearable Smart Devices and Accessories
- Industrial Control
  - Private (custom) services
  - Low bandwidth cable replacement

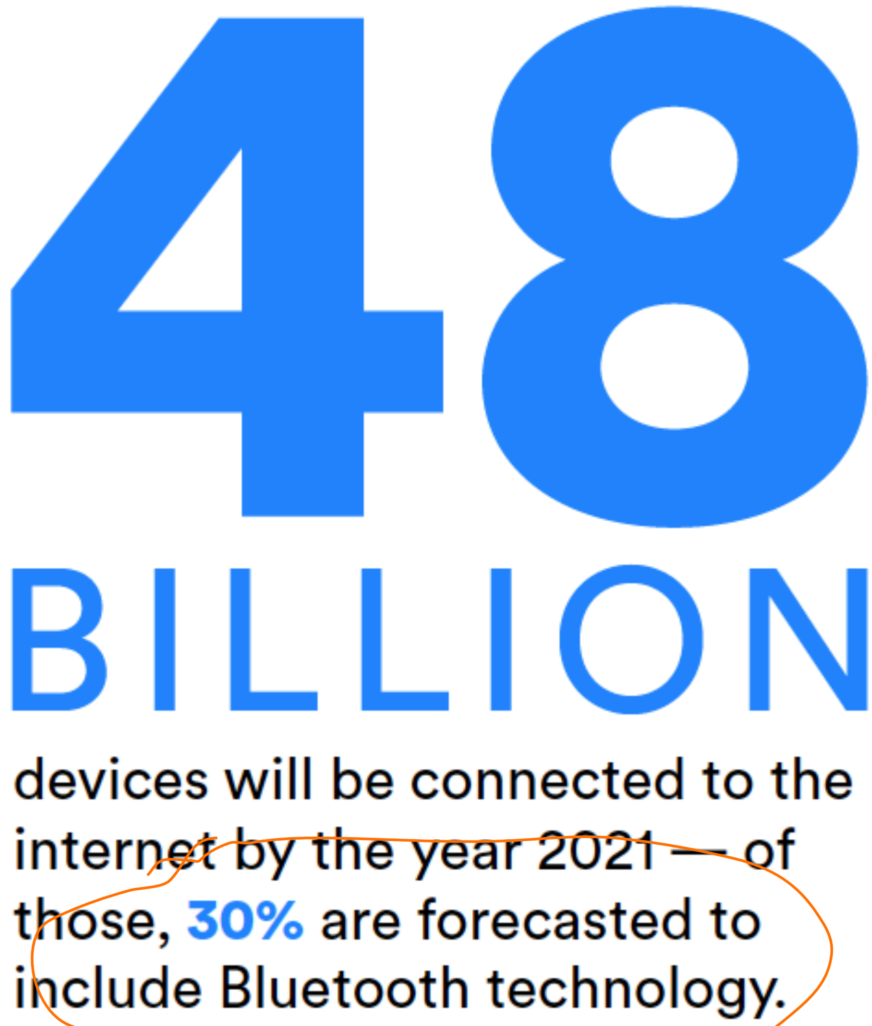
Bluetooth Versions

- **Bluetooth 1.1**: IEEE 802.15.1-2002
- **Bluetooth 1.2**: IEEE 802.15.1-2005. *Adaptive frequency hopping (avoid frequencies with interference)*.
- **Bluetooth 2.0** + Enhanced Data Rate (EDR) (Nov 2004): 3 Mbps using DPSK. For video applications. Reduced power due to reduced duty cycle
- **Bluetooth 4.0** (June 2010): Low energy. Smaller devices requiring longer battery life (several years). New incompatible PHY. Bluetooth Smart or BLE
- **Bluetooth 5.0** (December 2016): Make BLE go faster and further.

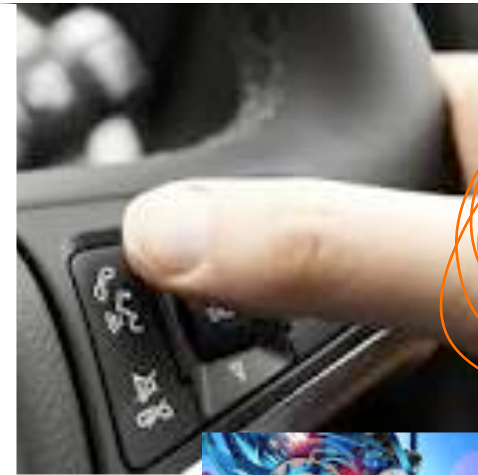
SLS

The Rise of Bluetooth

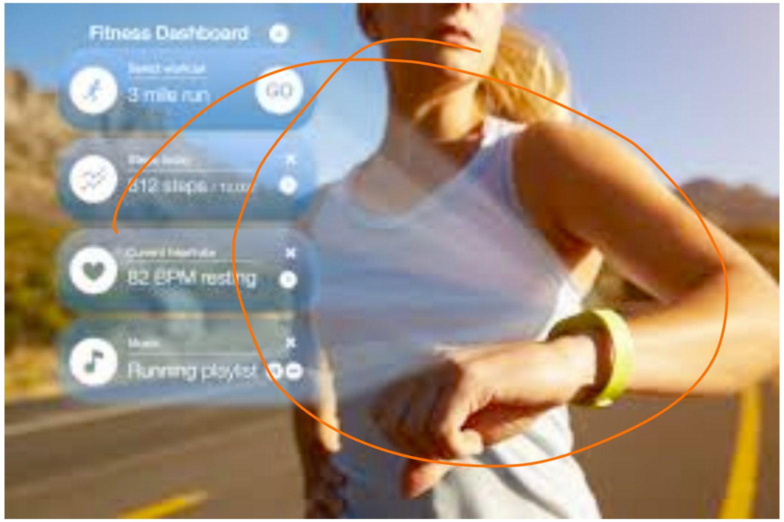




Source: Bluetooth SIG



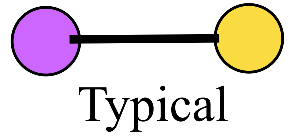
The Bluetooth Impact

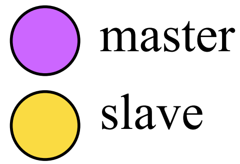


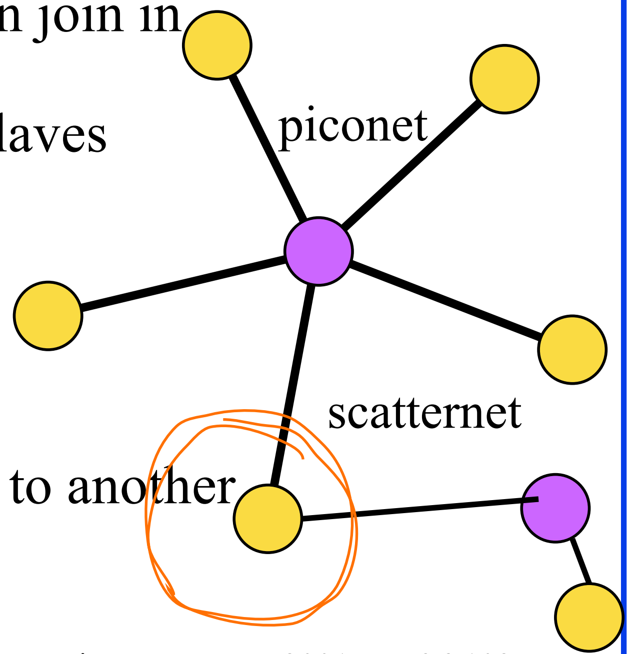
Bluetooth Classic

Bluetooth Network Topology: Piconet

- Piconet is formed by a master and many slaves (typically 1)
  - Up to 7 active slaves. Slaves can only transmit when requested by master
  - Up to 255 parked slaves
- Active slaves are polled by master for transmission
- Any device can become a master (initiator becomes master)
- Each station gets an 8-bit parked address
  $\Rightarrow$ 255 parked slaves/piconet
- A parked station can join in 2ms. Other stations can join in more time.
- Slaves can only transmit/receive to/from master. Slaves cannot talk to another slave in the piconet
- **Scatter net**: A device can participate in multiple Pico nets $\Rightarrow$ Timeshare and must synchronize to the master of the current piconet. *Active* in one piconet, *parked* in another.
- Routing protocol not defined (a node can only talk to another node if within Bluetooth range of 10m)

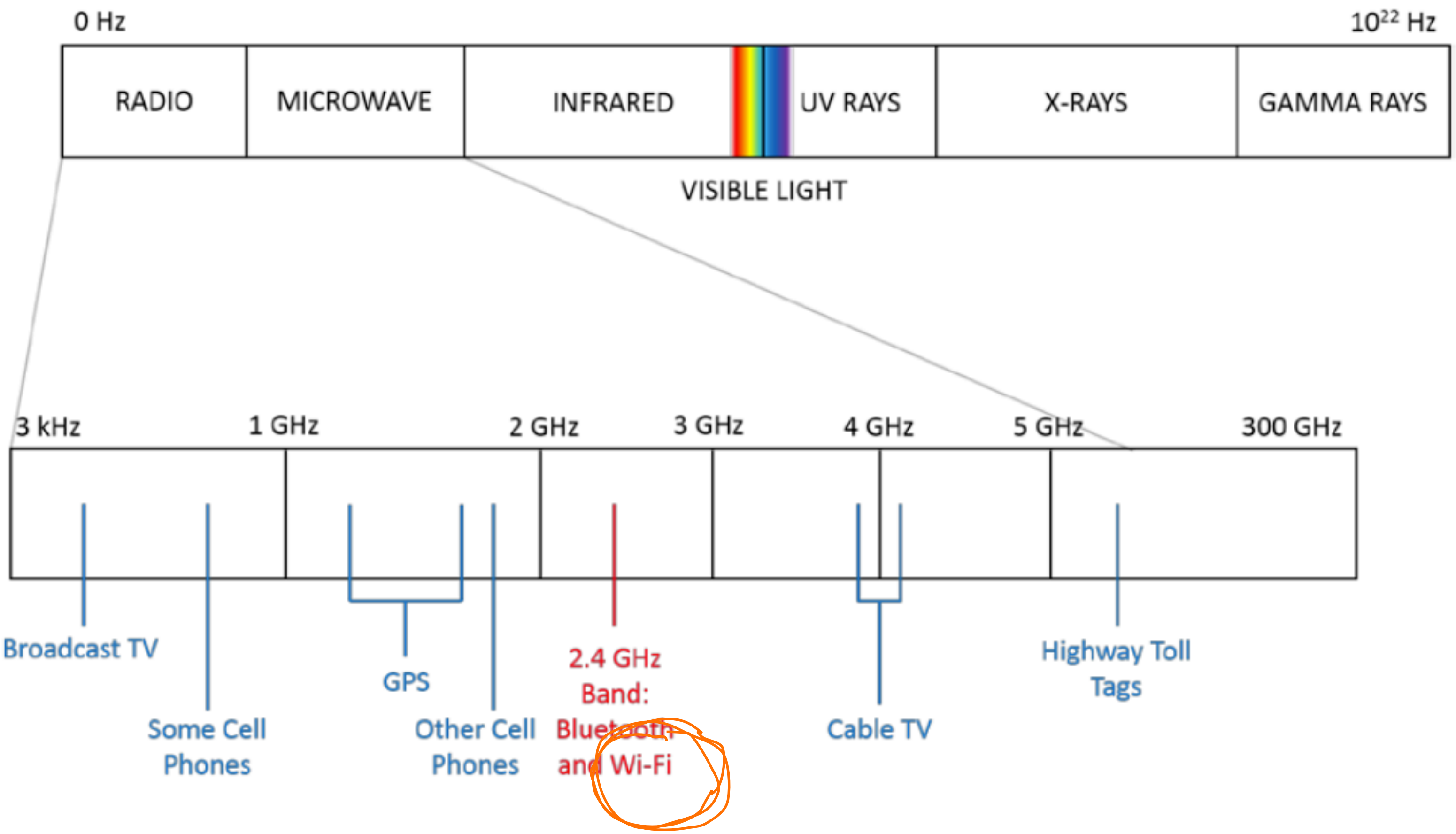




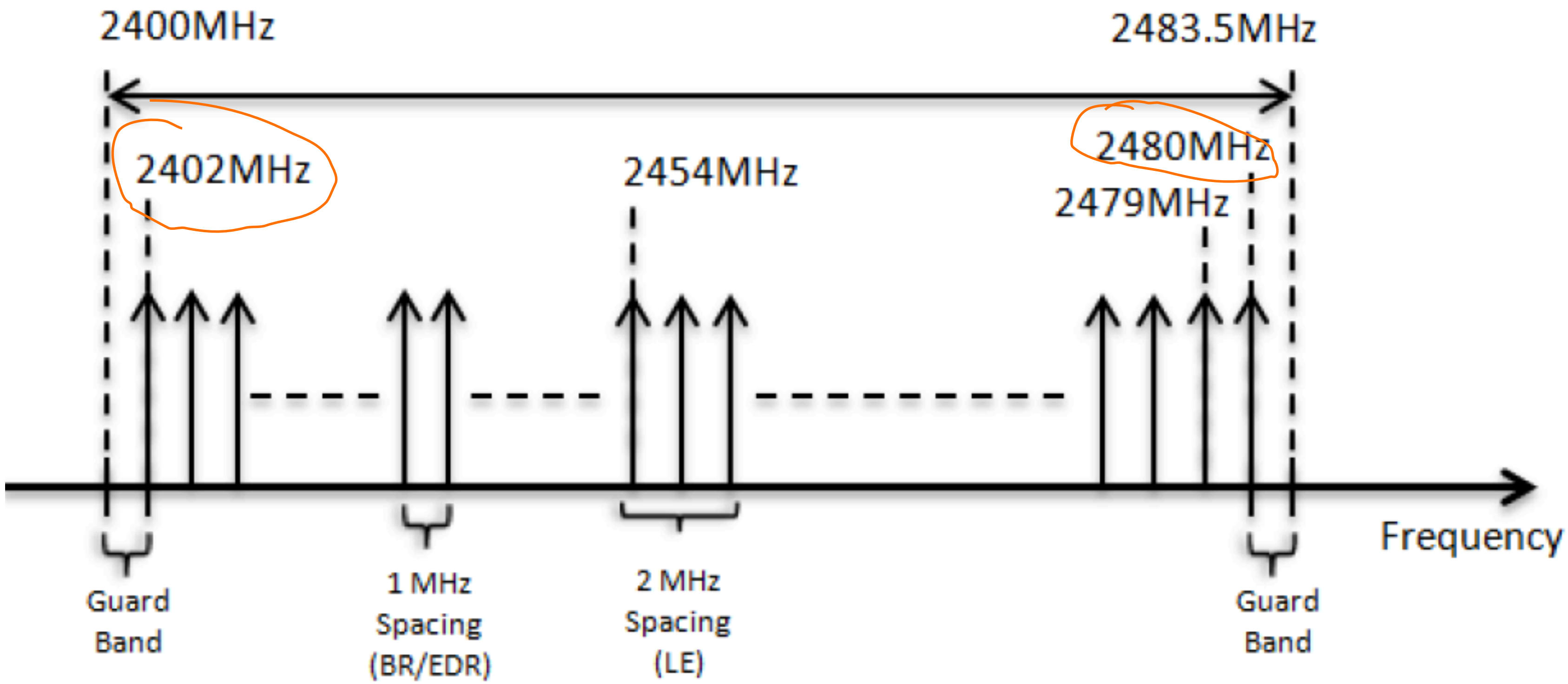


Ref: P. Bhagwat, "Bluetooth Technology for short range wireless Apps," IEEE Internet Computing, May-June 2001, pp. 96-103,

Bluetooth Operating Spectrum



Bluetooth Channels



f_c = (2402+k) MHz; k = 0, 1, …, 78

k: channel index (79 1-MHz wide channels)

Modulation and Data Rate

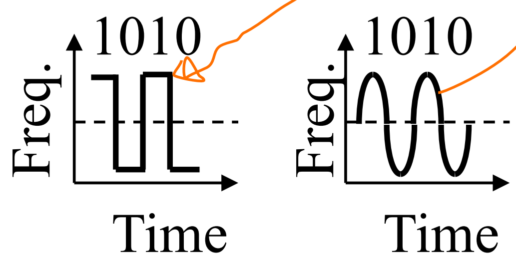
Wifi = QAM
amp + phase

- Basic rate (BR):
  - Binary Gaussian FSK (GFSK): 1 bit/symbol
  - Symbol duration = 1 μs: 1 Msps
  - Data rate: 1 Mbps
- Enhanced data rate (EDR):
  - Symbol duration is still 1 μs (1 Msps), but
  - μ/4-DQPSK; 2 bits/symbol; 2 Mbps
  - 8DPSK: 3 bits/symbol; 3 Mbps

Binary FSK

sharp

smooth

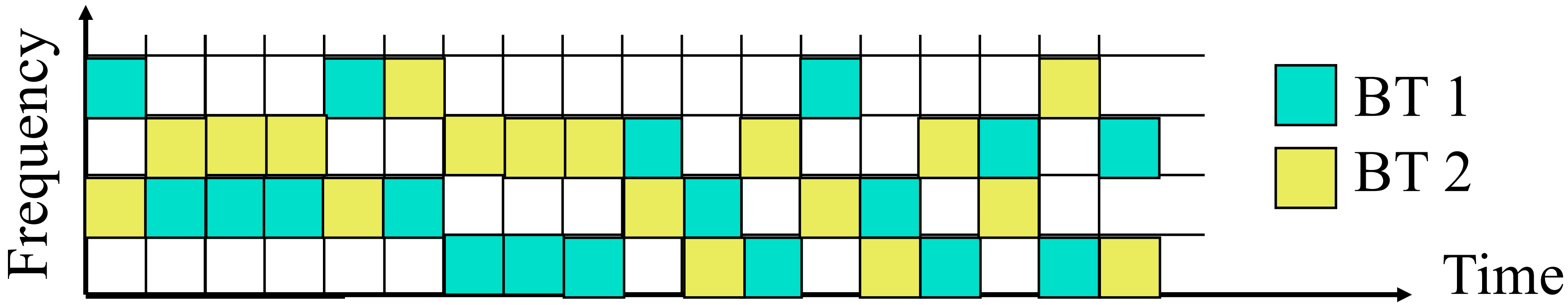


FSK GFSK

symbol rate x bits/symbol

Frequency Hopping (1)

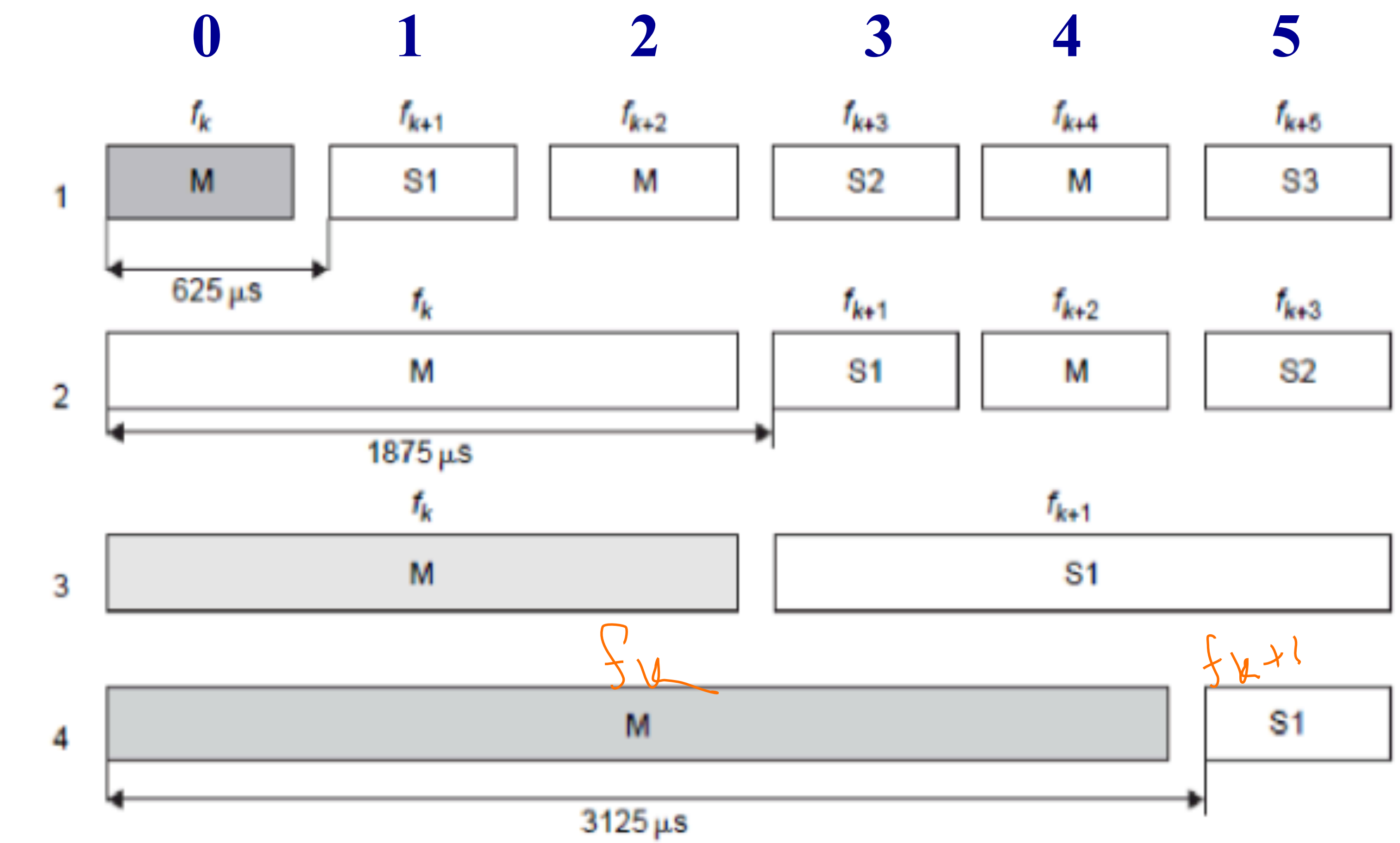
- Unlike WiFi, Bluetooth constantly switches channel within the same connection to avoid collisions with other nearby Bluetooth communications
- No two packets are transmitted on the same channel/frequency, but frequency is never switched in the middle of a packet transmission
- Such frequency switching is known as frequency hopping



Frequency Hopping (2)

- Bluetooth connections are slotted: packet transmission can start only at the beginning of a time slot
- 625 µs slots using a 312.5 µs (3200Hz) clock (1 slot = 2 clock ticks)
- Time-division duplex (TDD)
  ⇒ Downstream (master-to-slave) and upstream (slave-to-master) alternate
- Master starts in even numbered slots only.
- Slaves start in odd numbered slots only
- Slaves can transmit right after receiving a packet from master
- Packets = 1 slot, 3 slot, or 5 slots long
  - Enables master to start in even and slave in odd slots
- The frequency hop is skipped during a packet; frequency is hopped only at slot boundaries; at the beginning of the next slot after packet transmission/reception is complete; packet lengths may not align with slot boundaries

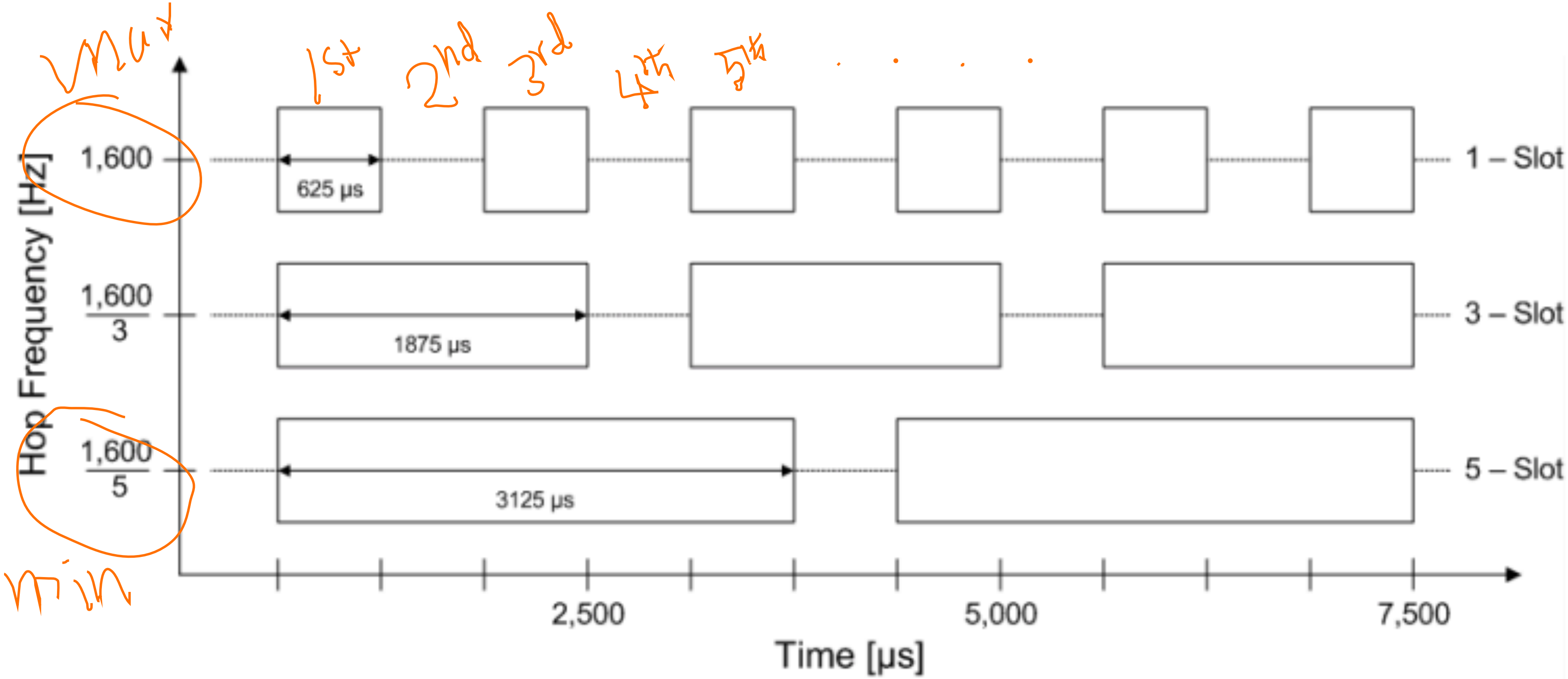
Frequency Hopping Illustrated



1. One-slot symmetrical; 2. Three-slot asymmetrical; 3. Three-slot symmetrical;
4. Five-slot asymmetrical

M=master, S = slave

Frequency Hopping Rate



1 frequency hop per packet: a packet can be 1,3, or 5 slot long (no hop in the middle of the packet); maximum FH rate = 1600Hz, minimum FH rate = 320Hz

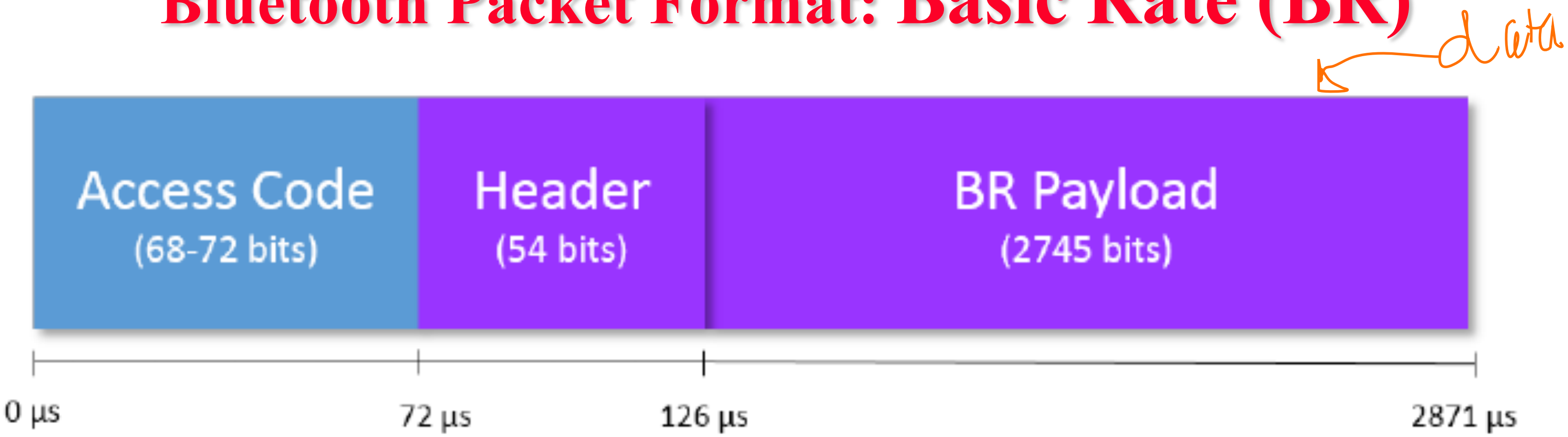
Example

- Consider a Bluetooth link where the *master* always transmits 3-slot packets. The transmission from the master is always followed up by a single-slot transmission from a *slave*. Assuming 625 µs slots, what is the effective frequency hopping rate (# of hopping per second)?

Answer: Given that frequency hopping cannot occur in the middle of a packet transmission, we only have 2 hops per 4 slots, or 1 hop per 2 slots.

The effective hopping rate = $1/(2 \times 625 \times 10^{-6}) = 800$ hops/s = 800Hz

Bluetooth Packet Format: Basic Rate (BR)

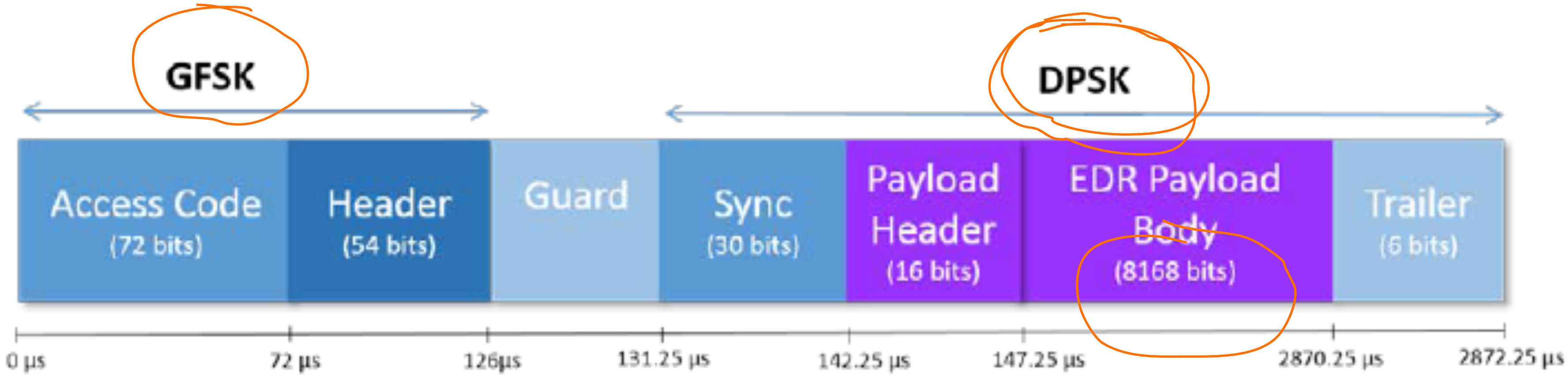


- Packets can be up to five slots long. 5 slots =625x5=3125 µs.
  - Maximum packet size = 72+54+2745 = 2871 µs (@1Mbps)
  - Some *residual* slot-time cannot be used (2871 < 3125)
- Access codes:
  - Channel access code identifies the piconet
  - Device access code for paging requests and response
  - Inquiry access code to discover units
- Header: member address (3b)+type code (4b)+flow control (1b)+ack/nack (1b)+sequence number (1b)+header error check (8b)=18b, which is encoded using 1/3 rate FEC resulting in 54b

Example

- How many slots are needed to transmit a Bluetooth Basic Rate packet if the payload is (a) 400 bits, (b) 512 bits, and (c) 2400 bits. Assume that the non-payload portions do not change.
- Answer:
  - Bluetooth transmissions are 1, 3, or 5 slots (2, 4, 6, etc. not allowed)
  - Non-payload bits (max) = 54+72 = 126 bits
  - Each slot can carry 625 bits at most
  - (a) 400b payload → 400+126 = 526b packet → 1 slot
  - (b) 512b payload → 512+126 = 638b packet → 2 slots would be sufficient, but will have to be padded for a 3-slot transmission (2-slot packets not allowed)
  - (c) 2400b payload → 2400+126 = 2526b packet → 5 slots

Bluetooth Packet Format: Enhanced Data Rate (EDR)



- Modulation changes within the packet; facilitated by a *guard interval lasting between 4.75 µs and 5.25 µs*
- GFSK for Access Code and Header
- µ/4-DQPSK (2Mbps) or 8DPSK (3Mbps) after guard interval
- EDR payload can accommodate more data than BR, but still fits within maximum 5-slot due to higher data rates

Bluetooth Address Format

company assigned						company id					
LAP						UAP		NAP			
0000	0001	0000	0000	0000	1100	0010	0111	1011	0011	0101	1001

LSB MSB

pseudo random generator

000666 = Roving Networks

- The Bluetooth device address is a unique 48-bit address sent in the *access code* field of the packet header.
- The first (most significant) 24 bits represent the OUI (Organization Unique Identifier) or the Company ID
- The main purpose of the Bluetooth address is for identification and authentication, but
- The address is also used to seed the frequency hopping pseudorandom generator, to synchronize master and slave clocks, and to pair devices.

Frequency Hopping with Pseudorandom Number Generator

- In Bluetooth Classic, FH is defined by a pseudorandom generating algorithm seeded with the following values
  - UAP and LAP of the master device address, and
  - Bits 1-26 of the 28-bit Bluetooth clock

28	27	26	. . . . .	3	2	1

- The pseudorandom pattern would repeat itself after 2^{27} hops
  - Would take 23.3 hours@1600Hz to repeat!
  - In practice the pseudorandom sequence is never repeated

Bluetooth is both Time and Frequency Synchronised

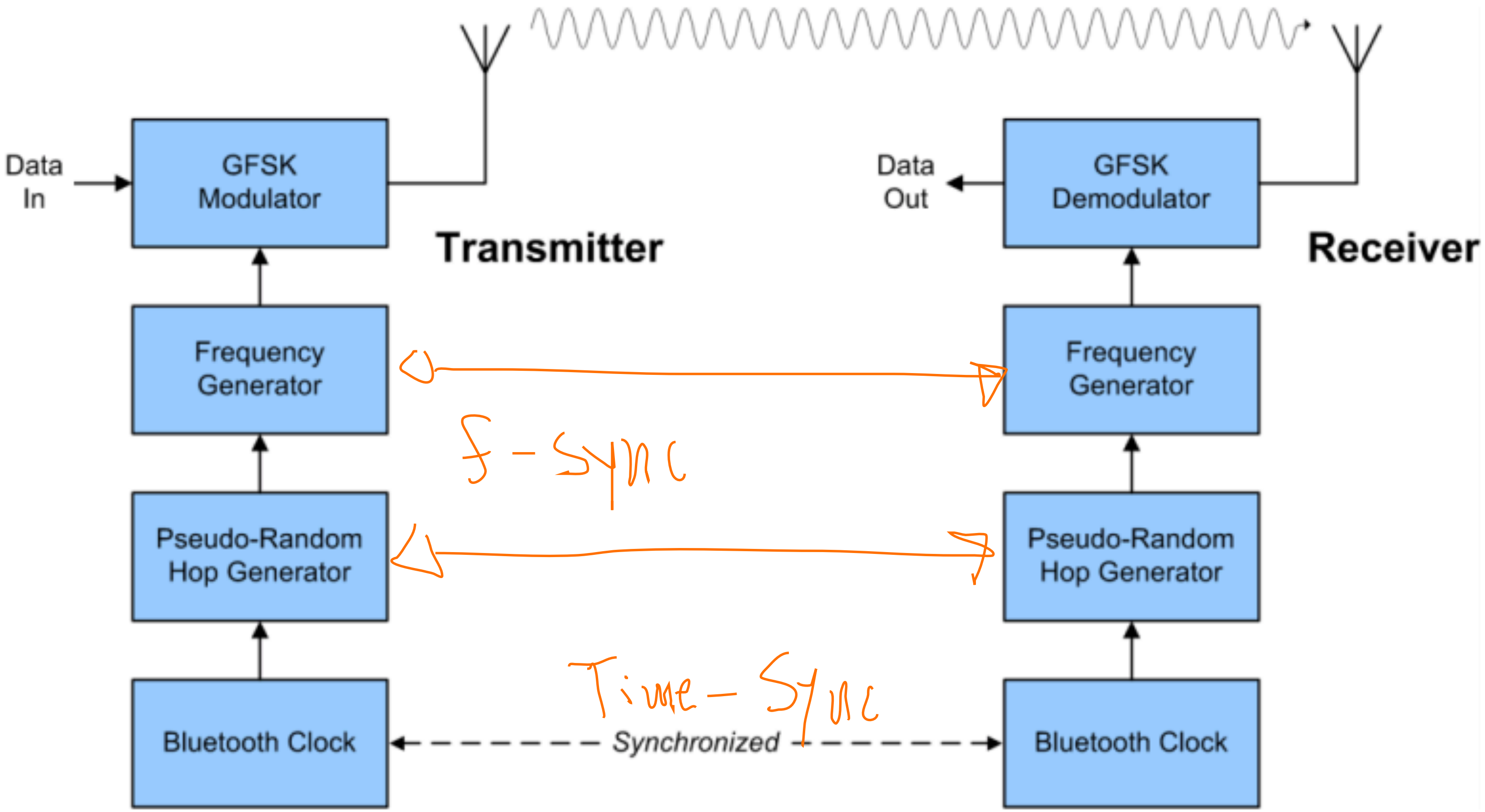
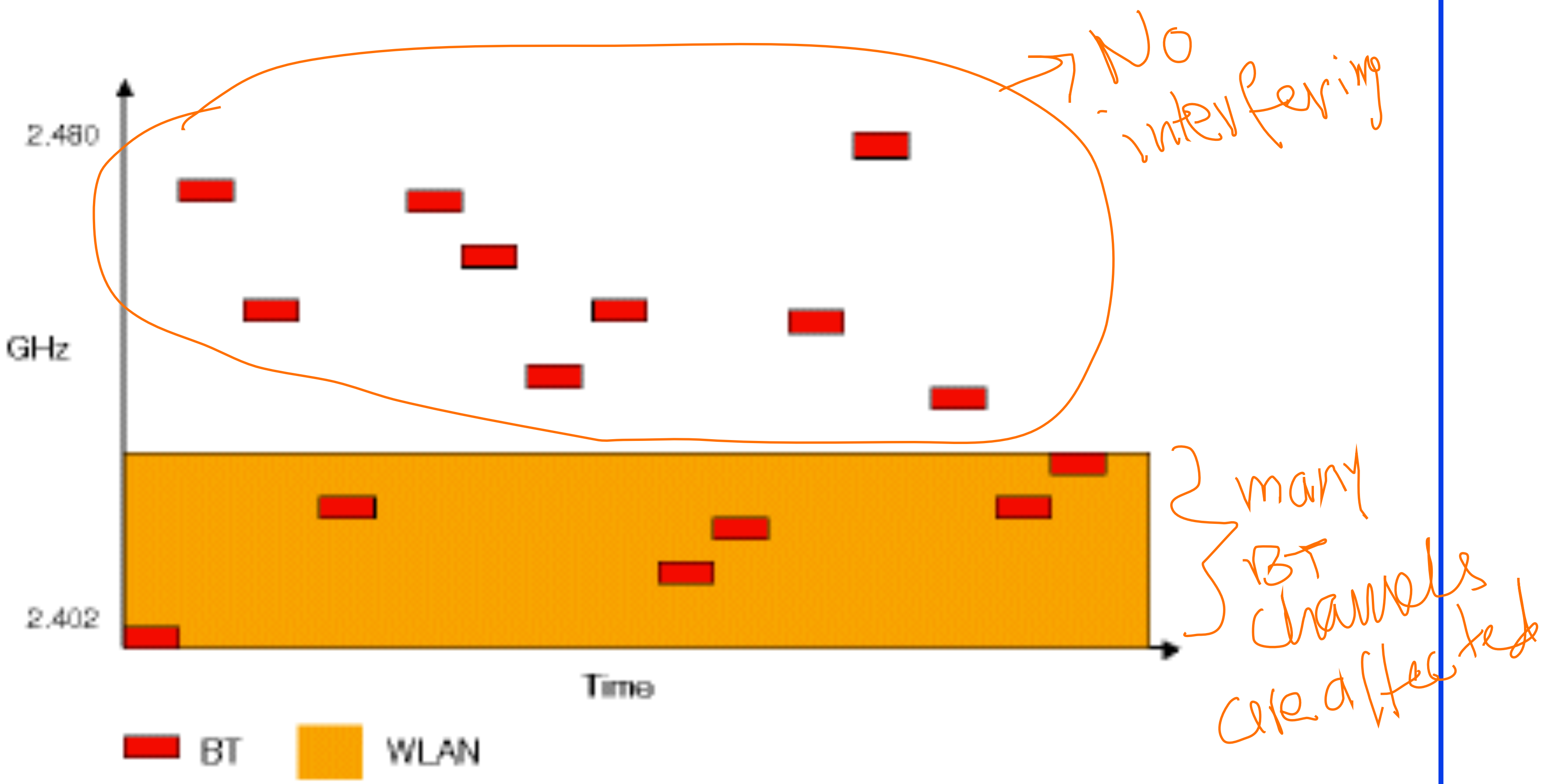


Illustration of Pseudorandom FH

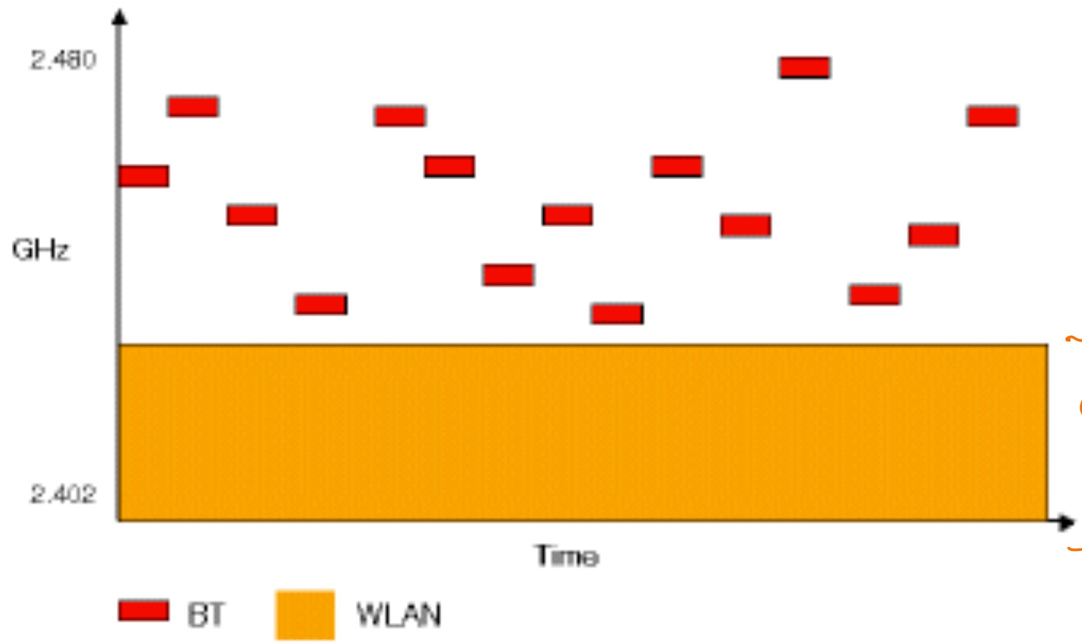
Time	Slot/Hop	Hopset															
10.000 ms	17 (TX)		▲ TX														
9.375 ms	16 (RX)															▼ RX	
8.750 ms	15 (TX)				▲ TX												
8.125 ms	14 (RX)						▼ RX										
7.500 ms	13 (TX)										▲ TX						
6.875 ms	12 (RX)															▼ RX	
6.250 ms	11 (TX)										▲ TX						
5.625 ms	10 (RX)									▼ RX							
5.000 ms	9 (TX)																▲ TX
4.375 ms	8 (RX)									▼ RX							
3.750 ms	7 (TX)										▲ TX						
3.125 ms	6 (RX)							▼ RX									
2.500 ms	5 (TX)						▲ TX										
1.875 ms	4 (RX)																▼ RX
1.250 ms	3 (TX)											▲ TX					
0.625 ms	2 (RX)				▼ RX												
0.000 ms	1 (TX)			▲ TX													
	Bluetooth Channel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	Frequency	2402 MHz	2403 MHz	2404 MHz	2405 MHz	2406 MHz	2407 MHz	2408 MHz	2409 MHz	2410 MHz	2411 MHz	2412 MHz	2413 MHz	2414 MHz	2415 MHz	2416 MHz	2417 MHz

Collision with WiFi: fixed (non-adaptive) hopping



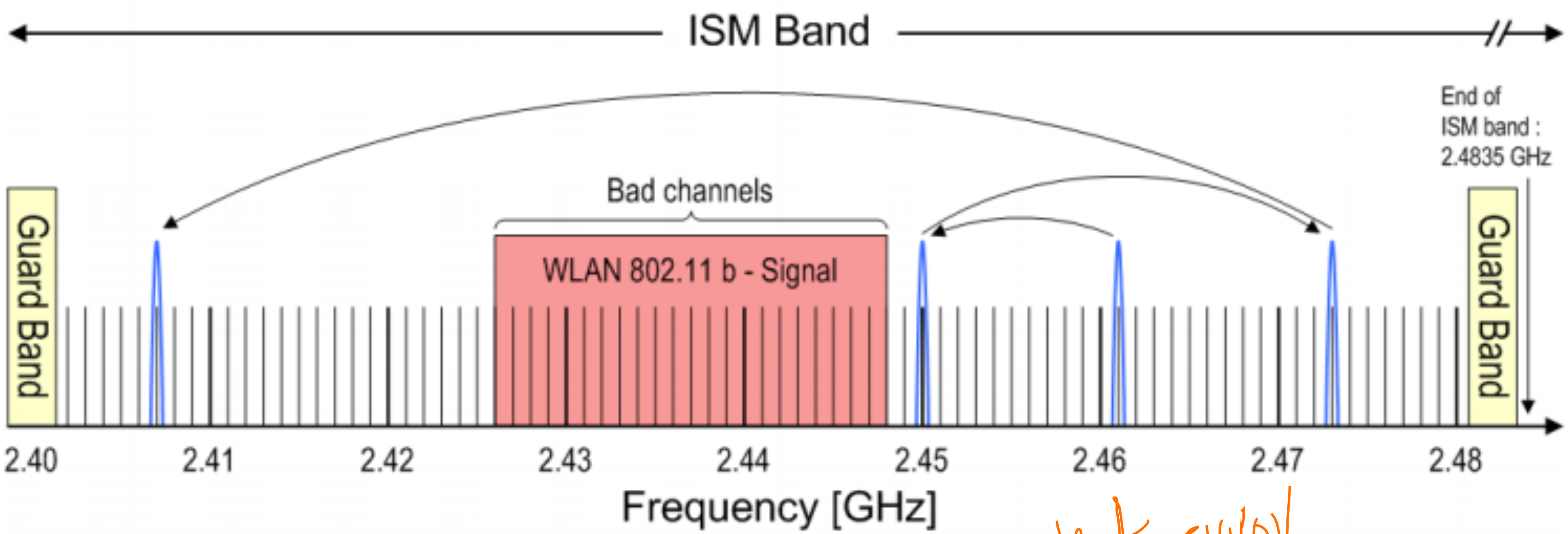
Collision Avoidance via Adaptive FH (AFH)

- Mark interfering channels as *bad channels*
- Avoid bad channels; hop between *good channels* only
- Minimum available (good) channels to hop = 20 (max. 79-20=59 channels can be marked as bad)
- AFH available only during Connected state (i.e., when two devices are exchanging data)



do not hop inside this area

AFH Illustration: hopping only between good channels



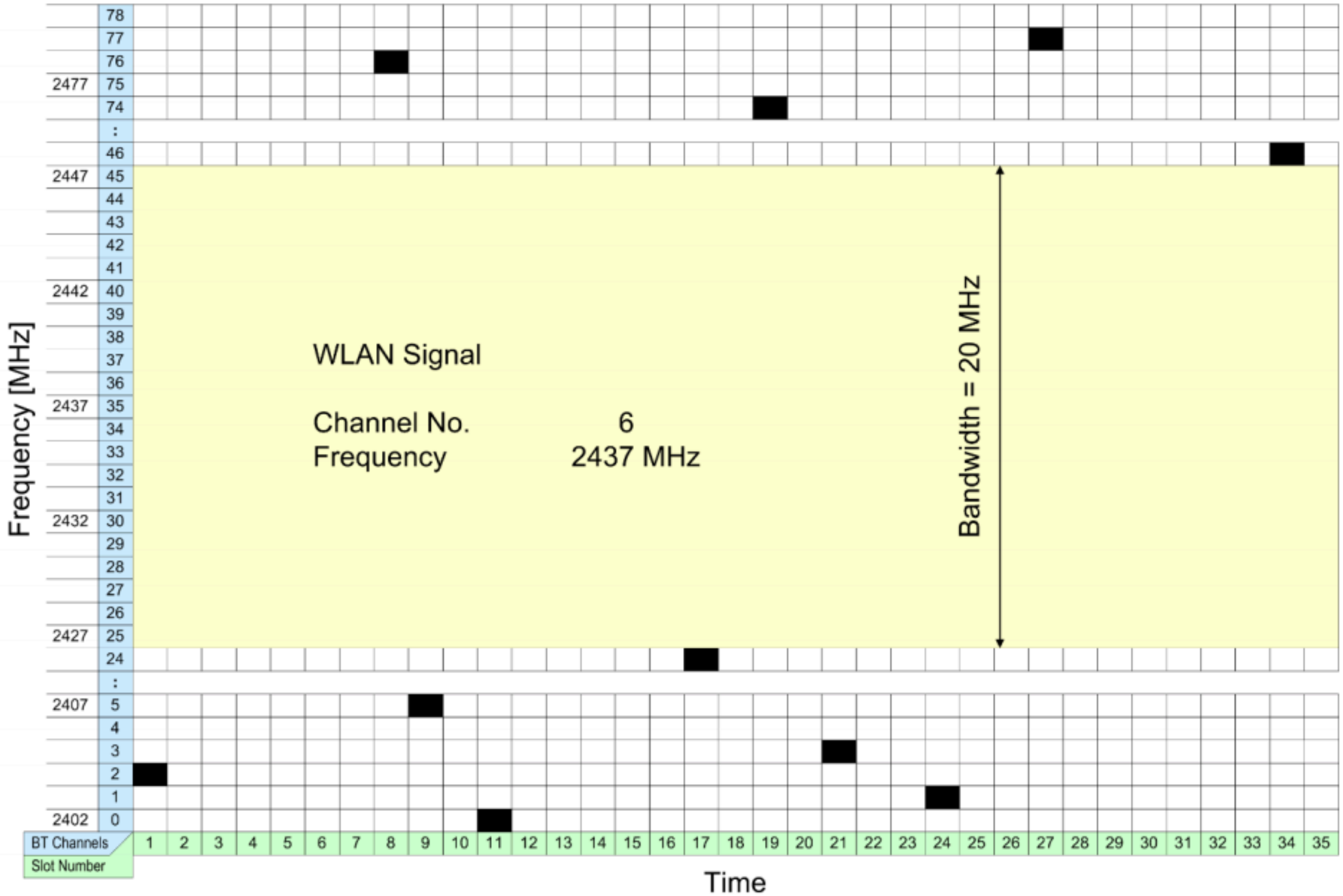
Packet error

Channel assessment: RSSI/SNR, PER (left to chipset vendor; not specifid in standard)

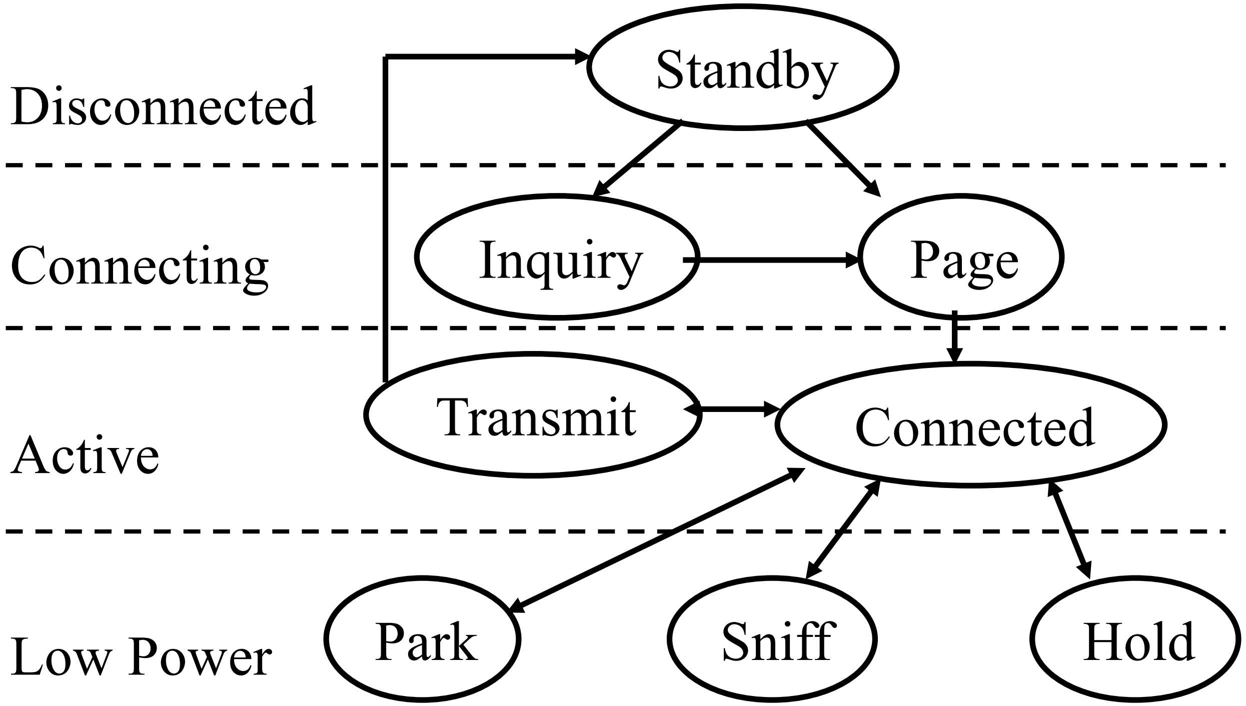
- Black: used (by another piconet)
- White: available (good to use)
- Yellow: Bad

Channel Map

Master updates the map dynamically and sends it to slaves



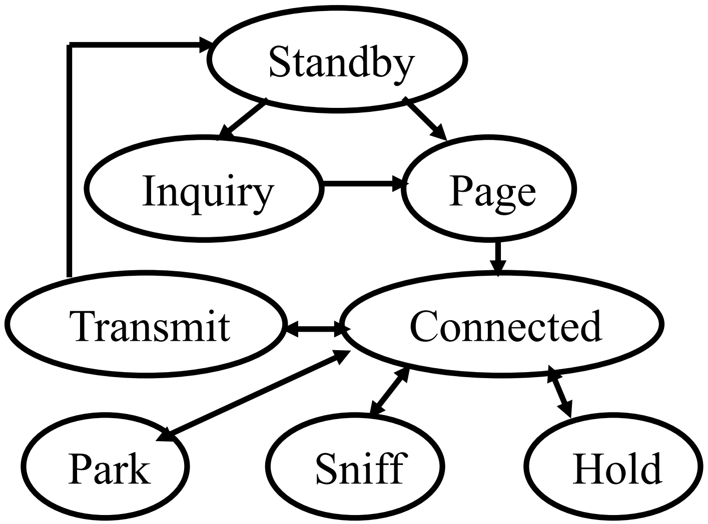
Bluetooth Operational States



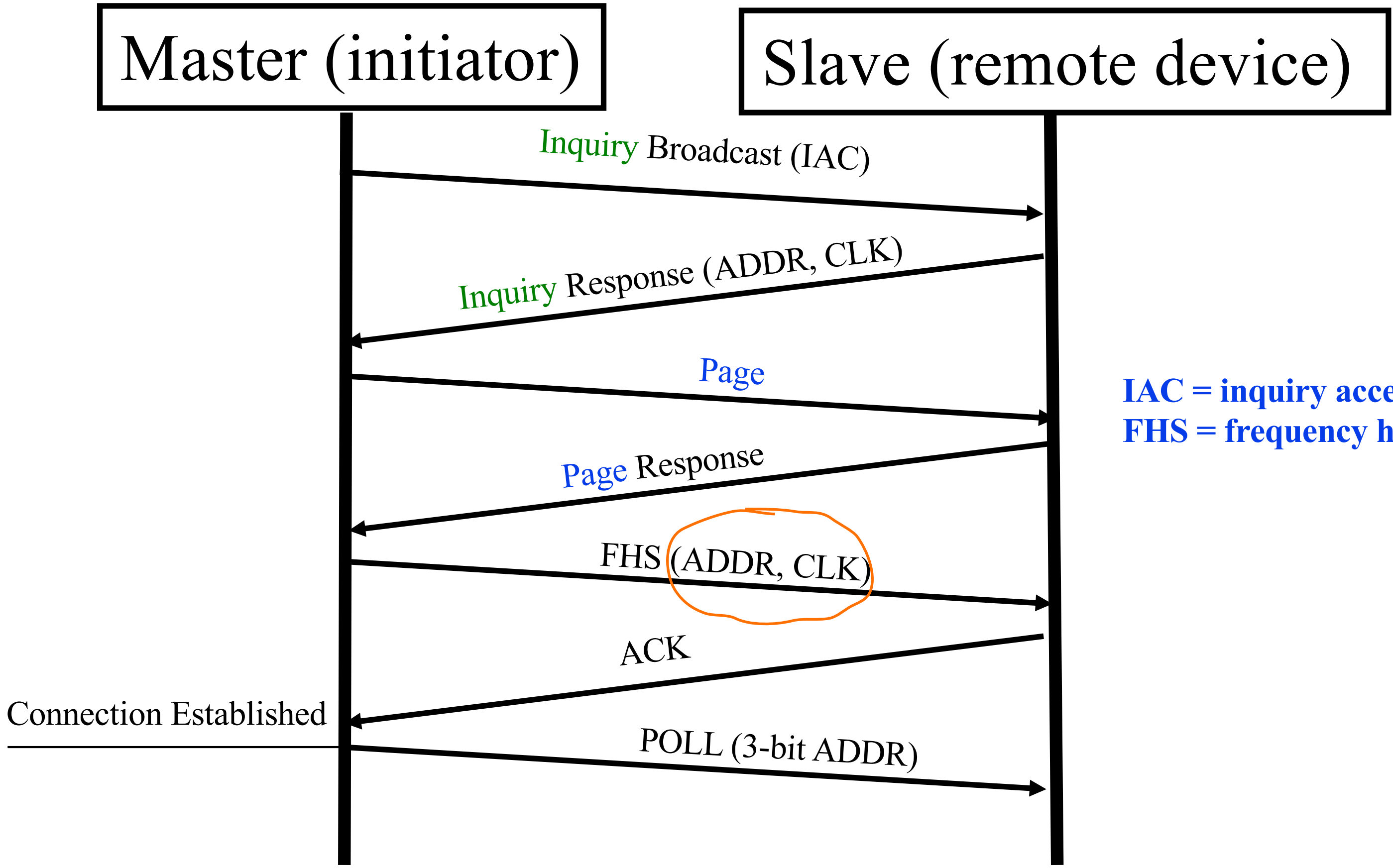
- 8 distinct states grouped under 4 high-level states
- **Standby**: Initial state
- **Inquiry**: Master broadcasts an inquiry packet. Slaves scan for inquiries and respond with their address and clock after a random delay (CSMA/CA)

Bluetooth Operational States (Cont)

- **Page**: Master in page state invites a slave device to join the piconet. Slave enters page response state and sends page response to the master.
- Master informs slave about its *clock* and *address* so that slave can participate in piconet.
- **Connected**: A short 3-bit logical address (*member address* within *control header* field) is assigned for the slave
- **Transmit**: station is transmitting or receiving a packet



Bluetooth Connection Establishment Procedure
Inquiry and Paging Flow Diagram



IAC = inquiry access code
FHS = frequency hopping synchronization

Bluetooth Connection Establishment Procedure
Inquiry and Paging Frequency Hopping

- Inquiry/page hopping sequence
  - Hop over 32 subset of 79 channels/frequencies (to speedup)
  - 32 is divided into two 16-channel *trains*
  - For inquiry, each train is repeated 256 times before *switching* to the other train; must have 3 train switches (1st → 2nd → 1st → 2nd): each train effectively repeated 256 x 2 times
  - Master sends two inquiry/page packets using 2 different frequencies per slot (hops in the middle of the slot; hops frequency in 312.5µs!), and listens for responses (both frequencies) in the following slots (to speed up) → eventually 2 frequencies covered in 2 slots
- Connection establish time
  - 16 x 625 µs = 10 ms for completing a train once
  - Inquiry time (maximum) = 256 x 4 x 10 ms = 10.24 s
  - There is an additional paging time

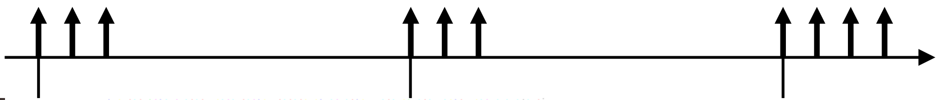
3-bit addr

Power Saving Modes in Bluetooth

Three inactive (power-saving) states:

1. **Hold**: Go inactive for a single short period and become active after that
2. **Sniff**: Low-power mode. Slave listens periodically after fixed sniff intervals.
3. **Park**: Very Low-power mode. Gives up its 3-bit active member address and gets an 8-bit parked member address. Wake up periodically and listen to beacons. Master broadcasts a train of beacons periodically

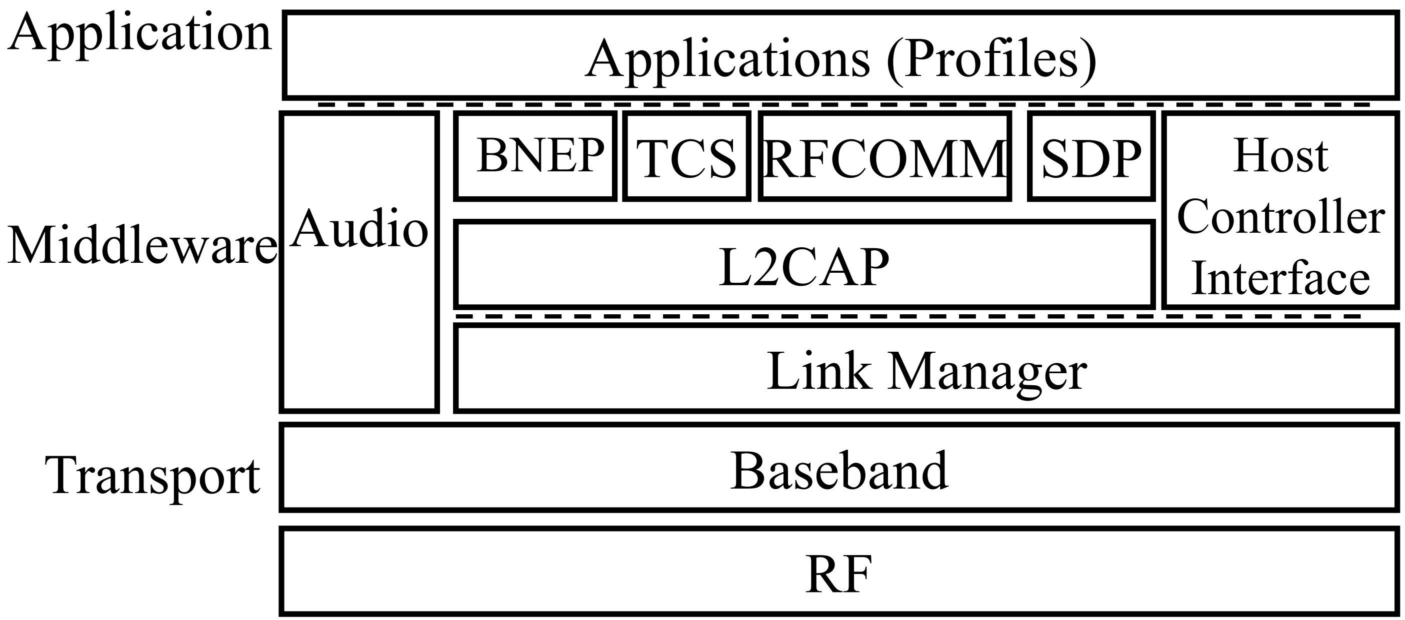
≤ 2ms delay



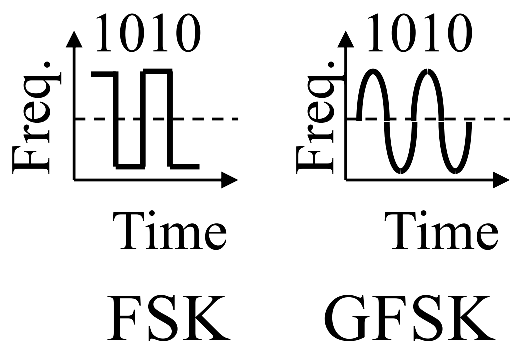
Sniff

Park

Bluetooth Protocol Stack



- **RF**: Gaussian Frequency Shift Keying (GFSK) modulation
- **Baseband**: Frequency hop selection, connection, MAC



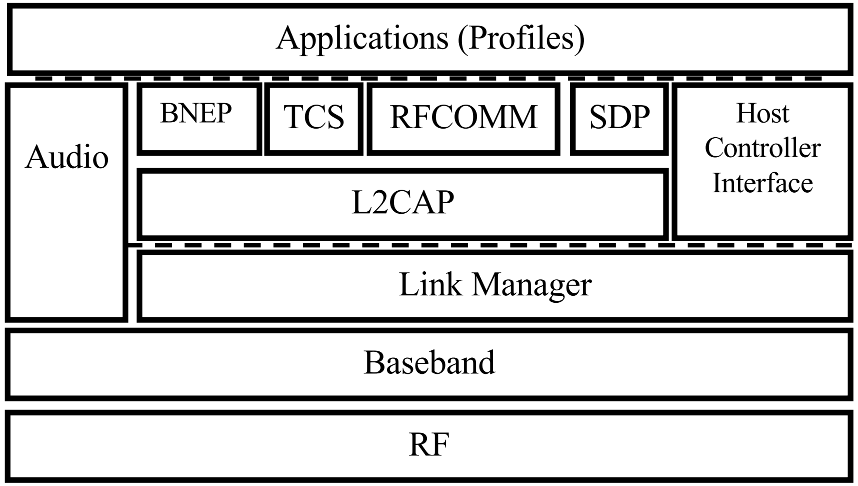
Baseband Layer

- Each device has a 48-bit IEEE MAC address
- 3 parts:
  - Lower address part (LAP) – 24 bits
  - Upper address part (UAP) – 8 bits
  - Non-significant address part (NAP) - 16 bits
- UAP+NAP = Organizationally Unique Identifier (OUI) from IEEE
- LAP is used in identifying the piconet and other operations
- Clock runs at 3200 cycles/sec or 312.5 µs (twice the hop rate)

Upper Address Part	Non-sig. Address Part	Lower Address Part
8b	16b	24b

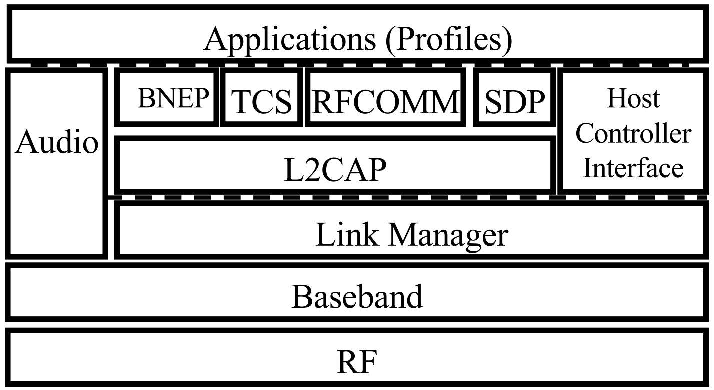
Bluetooth Protocol Stack (Cont)

- **Link Manager:** Negotiate parameters, Set up connections
- **Logical Link Control and Adaptation Protocol (L2CAP)**:
  - Protocol multiplexing
  - Segmentation and reassembly
  - Controls peak bandwidth, latency, and delay variation
- Host **Controller Interface**: Chip independent interface to Bluetooth chip. Allows same software to run on all chips.
- **RFCOMM Layer**: Presents a virtual serial port
  - Sets up a connection to another RFCOMM
- **Service Discovery Protocol (SDP):** Devices can discover the services offered and their parameters
  - E.g., Bluetooth keyboard,
  - Bluetooth mouse
  - Bluetooth headset
  - …



Bluetooth Protocol Stack (Cont)

- **Bluetooth Network Encapsulation Protocol (BNEP):** To transport Ethernet/IP packets over Bluetooth
- **IrDA Interoperability protocols**: Allow existing IrDA applications to work w/o changes. IrDA object Exchange (IrOBEX) and Infrared Mobile Communication (IrMC) for synchronization
- **Audio** is carried over 64 kbps over SCO links over baseband
- **Telephony control specification binary (TCS-BIN)**: Call control including group management (multiple extensions, call forwarding, and group calls)
  - Telephony has both audio and control
  - Bluetooth telephone very popular in cars
- **Application Profiles**: Set of algorithms, options, and parameters
  - To support specific applications



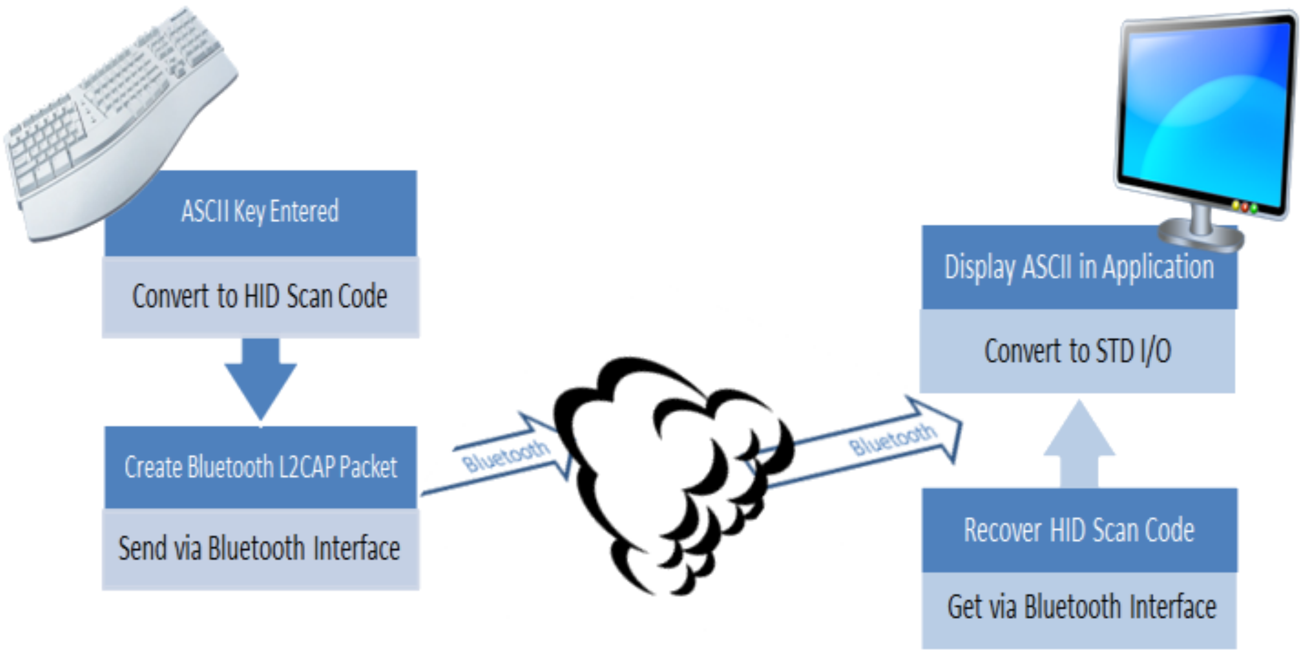
Application Profile Examples

- Headset Profile
- Global Navigation Satellite System Profile
- Hands-Free Profile
- Phone Book Access Profile
- SIM Access Profile
- Synchronization Profile
- Video Distribution Profile
- Blood Pressure Profile
- Cycling Power Profile
- Find Me Profile
- Heart Rate Profile
- Basic Printing Profile
- Dial-Up Networking Profile
- File Transfer Profile

With IoT, the list is expected to grow rapidly over the coming years

Ref: Bluetooth SIGn, "Adopted Bluetooth Profiles, Services, Protocols and Transports," https://www.bluetooth.org/en-us/specification/adopted-specifications

Connecting a wireless keyboard with HID Bluetooth profile

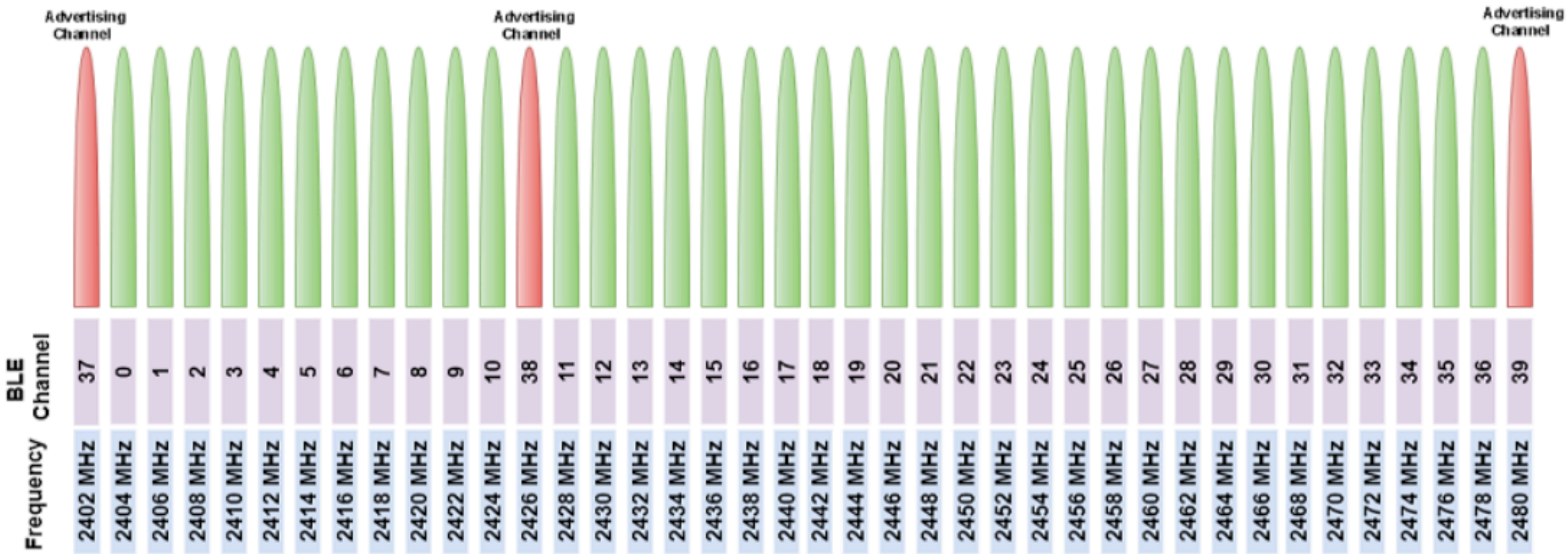


Bluetooth Low Energy (BLE) a.k.a Bluetooth 4

Bluetooth LE or BLE

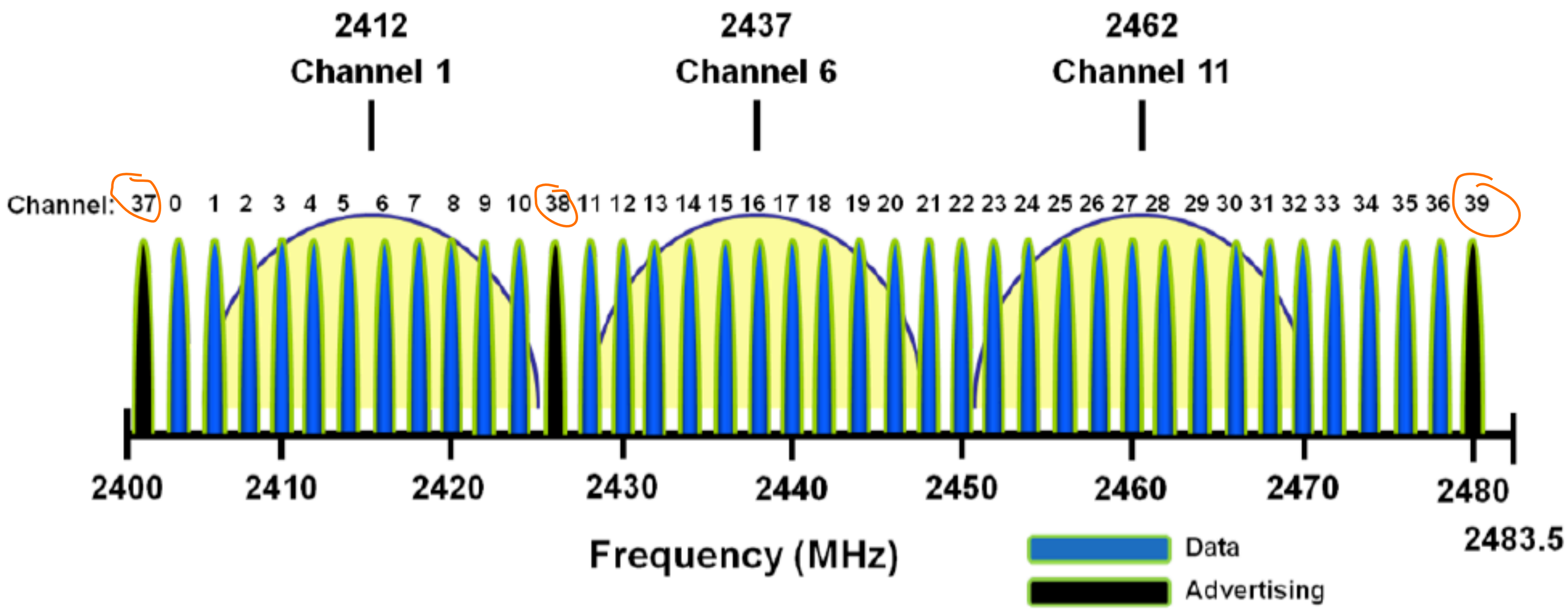
- **Low Energy**: 1% to 50% of Bluetooth classic
- **For short broadcast**: Your body temperature, Heart rate, Wearables, **sensors**, automotive, industrial.
  Not for voice/video, file transfers, …
- **Small messages**: 1Mbps data rate but throughput not critical.
- **Battery life**: In years from coin cells
- **Simple**: Star topology. No scatter nets, mesh, …
- **Lower cost** than Bluetooth classic
- New protocol design based on Nokia's **WiBree** technology
  Shares the same 2.4GHz radio as Bluetooth
  $\Rightarrow$ Dual mode chips
- Most smartphones (iPhone, Android, …) have dual-mode chips

BLE Channels



- 40 2MHz-wide channels: 3 (37,38,39) for advertising and 37 (0-36) for data
- Advertising channels specially selected to avoid interference with popular default WiFi channels (1,6,11)

BLE Advertising Channels Avoiding Popular WiFi Channels



BLE Modulation and Data rate

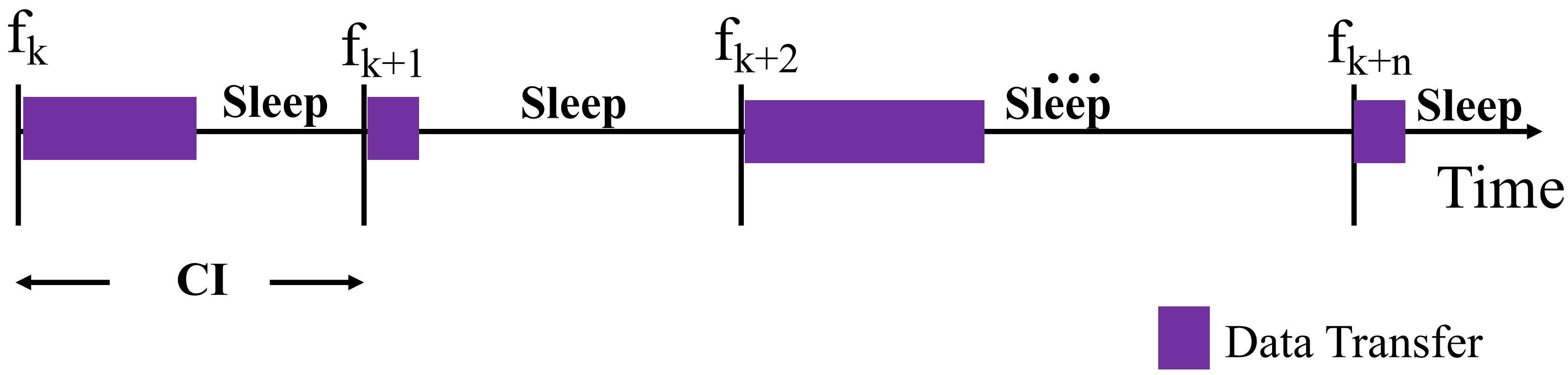
- Binary GFSK over 2MHz channel: More significant frequency separations for '0' and '1' allows longer range with low power
  - Note that with Bluetooth Classic, channel bandwidth is only 1MHz, so frequency separations are smaller
- 1 million symbols per second → 1 Mbps data rate

Benefit of Advertising Channels

- BLE simplifies discovery and broadcasting by using only three advertising channels (instead of 32 channels for inquiry/paging in BT Classic)
- A BLE device can broadcast advertising beacons on these 3 channels giving information about the device, so other devices can connect, but can also broadcast some sensor data
- Data channels are used to exchange data bidirectionally between two devices

Connection Events and Connection Intervals

- In BLE connections, devices wake up periodically after every connection interval (CI) time; transmit some data (connection event) and then go back to sleep until the next connection event
- Send a short blank packet if no data to send during a connection event
- More than one packet can be sent during a connection event
- Connection interval time can vary from 7.5ms to 4s and is negotiated during connection set up
- Hop frequency (switch to different data channel) at each event



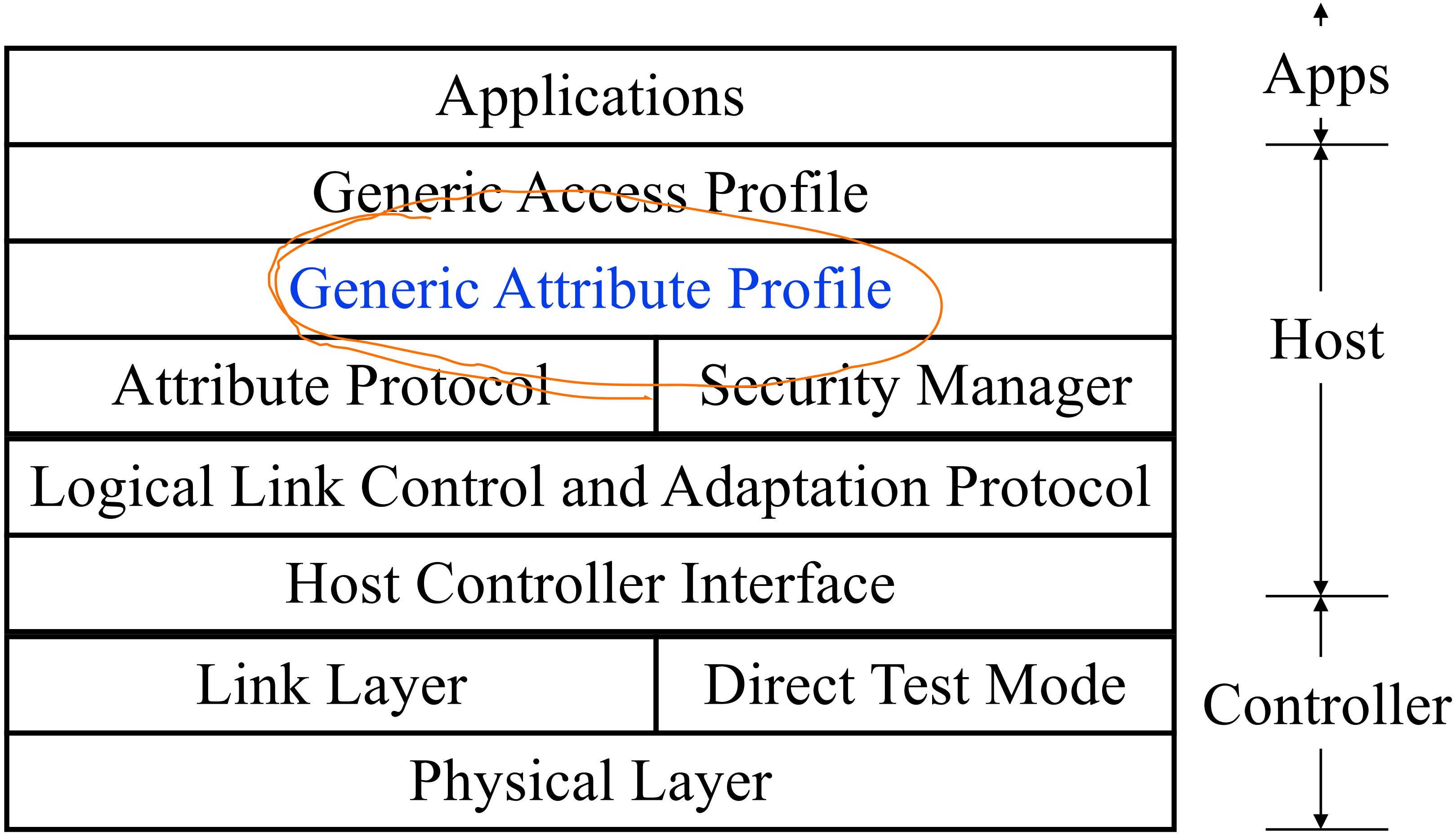
BLE Frequency Hopping Algorithm
a.k.a Algorithm #1

- *Fixed* hopping instead of *pseudorandom*
- $f_{k+1} = (f_k + h) \bmod 37$
  - Where h (hop increment) is a fixed value negotiated during connection setup
  - Note: Data channels range from 0-36

0-36

- Example hopping sequence for h=10: 0→10→20→30→3→13
- Adaptive FH: If the hopping lands on a *bad* channel, the channel is remapped to a *good* channel using a channel *remapping* algorithm

Bluetooth Smart Protocol Stack



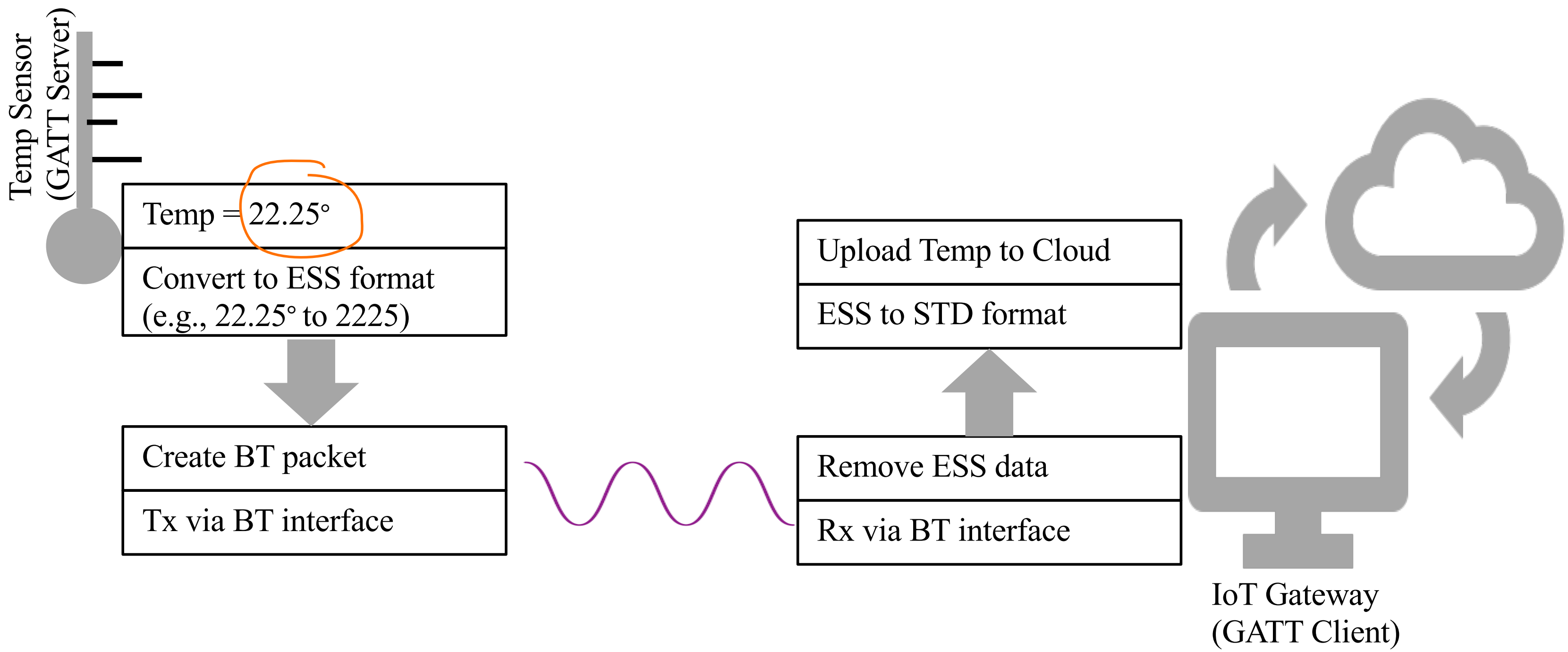
Generic Attribute (GATT) Profile

- Defines data formats and interfaces with the Attribute Protocol
  - Define attributes instead of applications (a major difference from Bluetooth Classic); temperature, pressure, heart rates are examples of attributes
  - New applications can be supported by using appropriate attributes
- Type-Length-Value (TLV) encoding is used
- Each attribute has a 16-bit Universally Unique ID (UUID) standardized by Bluetooth SIG
  - 2^{16}=65 thousand unique attributes can be defined!
- 128-bit UUID if assigned by a manufacturer
  - Manufacturers can define their own attributes and still interoperate
- Allows any client to find a server, read/write data
  Allows servers to talk to generic gateways
- Allows security up to AES-128
- Each to encode in XML
- Makes profile (application) development easier

Example of BLE GATT services and characteristics

Service	Characteristic	Format
Environmental Sensing Service (ESS) UUID = 0x1756	Temperature (2A1C)	16-bit little endian value representing the measured temperature. Unit: 0.01 deg C
	Humidity (2A6F)	16-bit little endian value representing the measured relative humidity. Unit: 0.01%
	Pressure (2A6D)	32-bit little endian value representing the measured pressure. Unit: 0.1 Pa (0.001 hPa)
Heart Rate Service (HRS) UUID = 0x180D	Hear Rate Measurement (2A37)	1-2B integer representing BPM

Temperature sensing using BLE: environmental sensing service profile



Bluetooth Gateway Devices

- A gateway device helps connect a Bluetooth device to the Internet. Smart phone, Tablets, PC, …
- A generic app can forward the data to the URL sent by the device

Bluetooth Smart Applications

- Proximity: In car, In room 303, In the mall
- Locator: Keys, watches, Animals
- Health devices: Heart rate monitor, physical activities monitors, thermometer
- Sensors: Temperature, Battery Status, tire pressure
- Remote control: Open/close locks, turn on lights

Beacons

- Advertizing based on proximity
- Peripherals (your phone) broadcasts its presence if Bluetooth is turned on
- Primary aim of these broadcasts is to allow device discovery
- Advertising packets consist of a header and max 27B of payload with multiple TLV-encoded data items
  - May include signal strength → Distance
- iPhones can send/received iBeacons
- Can be used for customized advertising, indoor location, geofencing
- PayPal uses this to identify you.
  You can pay using a PIN and your phone.

Bluetooth 5

"Go Faster. Go Further"

Bluetooth 5: Motivation

- BLE (Bluetooth 4) was great in terms of reducing energy consumption and extending battery life
- BLE, however, could not support high data rate applications, such as audio and file transfer (e.g., quick firmware updates), and the range was still limited for some new IoT applications
- Bluetooth 5 extends BLE to realise a faster (2x) and longer range (4x) Bluetooth without compromising the battery life; advertising is also improved
- Bluetooth 5 is seen as a significant new milestone in the evolution of Bluetooth; expected to support many new markets in home and industrial automation, health and fitness tracking, and so on.

Bluetooth 5: Major Improvements

- Two new PHYs: one for 2x higher speed and the other for 4x longer range than BLE 4
- New Advertising
- Improved frequency hopping

Benefits and use cases for 2x speed

- Quick firmware updates for millions of home and industrial automation devices
- Sports and fitness wearable multi-dimensional and buffered data uploads to edge/cloud
- Medical device data uploads, e.g., ECG, EEG, …
- Higher spectral efficiency for the congested 2.4GHz space

occupying the channel longer

2x faster

PHY: 2M

- Two mega symbols per sec: symbol duration = 500ns
  - Symbol duration reduced by half from BLE 4
- Binary GFSK, but with higher frequency deviation to combat inter-symbol interference arising from shorter symbols:
  - Frequency deviation (from central frequency) to denote ‘1’ or ‘0’ in FSK > 370kHz (180kHz in BLE 4)

PHY: Coded

- 1 Mega symbols per sec: the same as in BLE 4
- However, to increase the range, data is coded with FEC; two coding rates
  - ½: cuts data rate by half → 500Kbps; 2x range increase against BLE 4
  - ¼: → 250Kbps; 4x range increase
- BLE 4 and BT Classic do not employ any FEC (*not coded*)

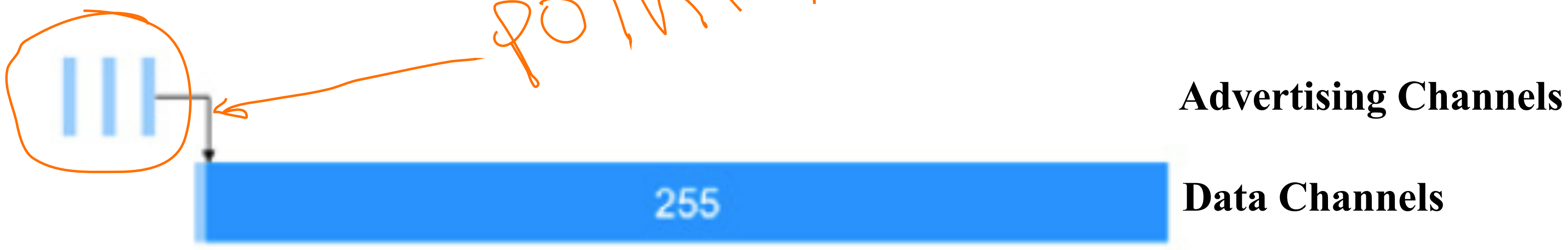
Advertising Extensions

- Motivation: Bluetooth beacons is a major advertising use case
- BLE 4 typically allow just ID or URL to be advertised in the beacon due to limited advertising packet size (31 bytes payload) and heavy load on advertising channels
  - BLE 4 uses channels 37,38,39 for advertising; all beacon have to be transmitted on all three channels
- Bluetooth 5 allows advertising packets up to 255B payload
  - Devices and products can advertise many more things and status, such as a fridge can advertise its contents, temperature, expiry dates of sensitive items, etc.

Advertising Extension: Channel Offload

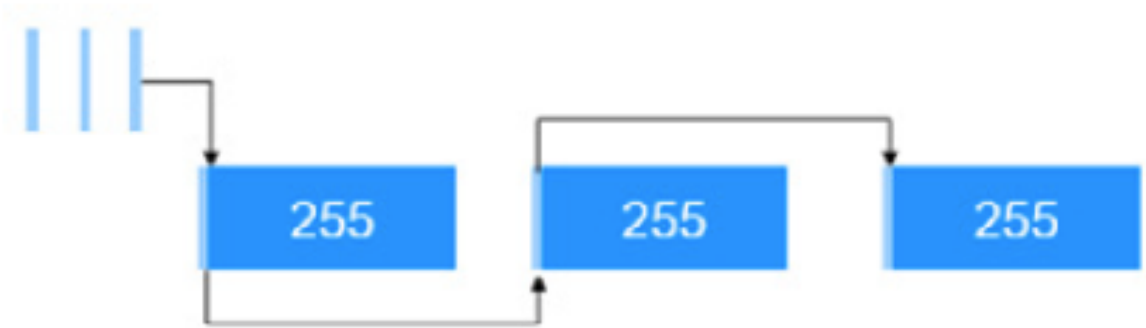
- Only header is transmitted over advertising channels and the actual payload is offloaded to a data channel
- Note: BLE 4 reserves data channels only for data transfers during Connection Events when connections are established; Channel offload allows use of data channels in connectionless manner

pointer to data channels



Advertising Extension: Packet Chaining

- Chain multiple 255B packets together to carry a very large advertising message



Frequency Hopping Extension

- BLE 4 supports only a simple hopping algorithm
  - Algorithm #1: fixed hopping increment only
- Fixed hopping increment limits the number of possible sequences to choose from
- Bluetooth 5 supports pseudorandom hopping like the BT Classic
  - Algorithm #2: large number of sequences possible

BT Classic — pseudorandom

BT LE — Al #1

BT 5 — Al #1 + pseudorandom

Summary

1. Bluetooth Classic uses frequency hopping over 79 1-MHz channels with 1, 3, and 5-slot packets.
2. Bluetooth 4 is designed for short broadcasts by sensors. 40 2-MHz channels are used with 3 channels reserved for advertising and 37 used for data transfers.
3. BT Classic uses flat application profiles to support different types of communication services, which requires different application profiles to be defined for different types of sensing and communications.
4. BLE has a hierarchical service structure to group many sensing measurements into a given service type, which scales for large variety of devices and services expected in the IoT era.
5. Bluetooth 5 extends BLE to support higher data rate and longer-range. It also has an improved advertising structure that allows advertisement of more comprehensive information and contents.