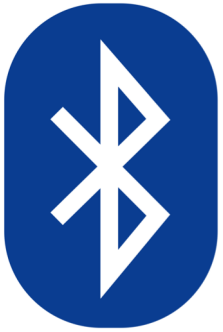
Advanced Computer Networking

CYBR 230 – Jeff Shafer – University of the Pacific



Bluetooth

Overview

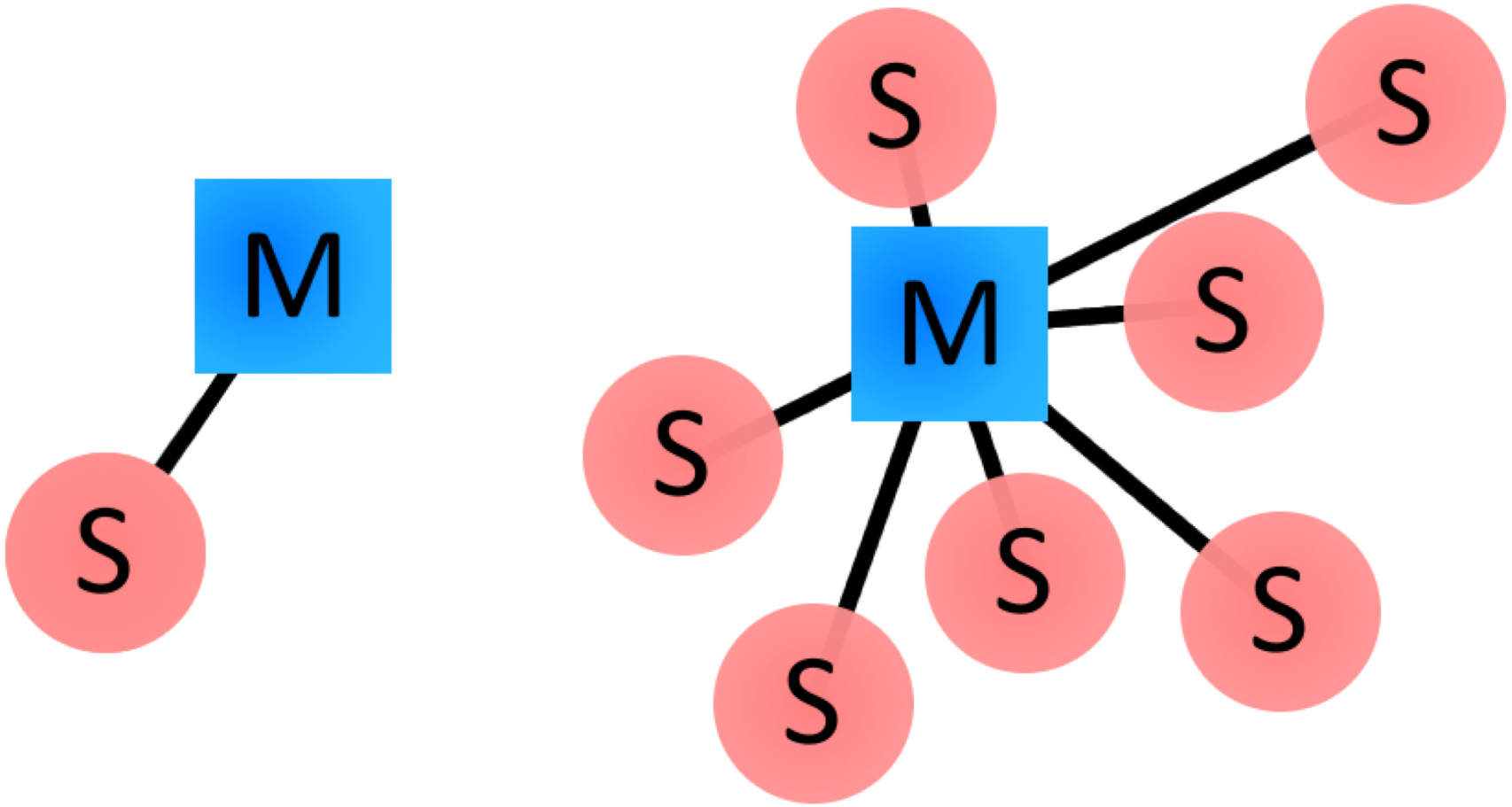
Design Goals

- Short range wireless communication
  - Voice, audio, data, …
- Low power
- Low cost
- Low skill (to configure/use)

Piconets

- Bluetooth network = “Piconet”
  - Ad-hoc computer network
- Master device: Coordinates communication
  - Can send data to any slave
  - Can request data from any slave
  - Defined as “device that establishes the Piconet”
- Slave device
  - Only allowed to communicate with master (can’t communicate with other slaves)
- Maximum of 8 devices
  - 1 master, 7 slaves

Piconets



Bluetooth Special Interest Group

- Develops Bluetooth standards
- Qualification and interoperability testing
- Licenses technologies and trademarks to manufacturers

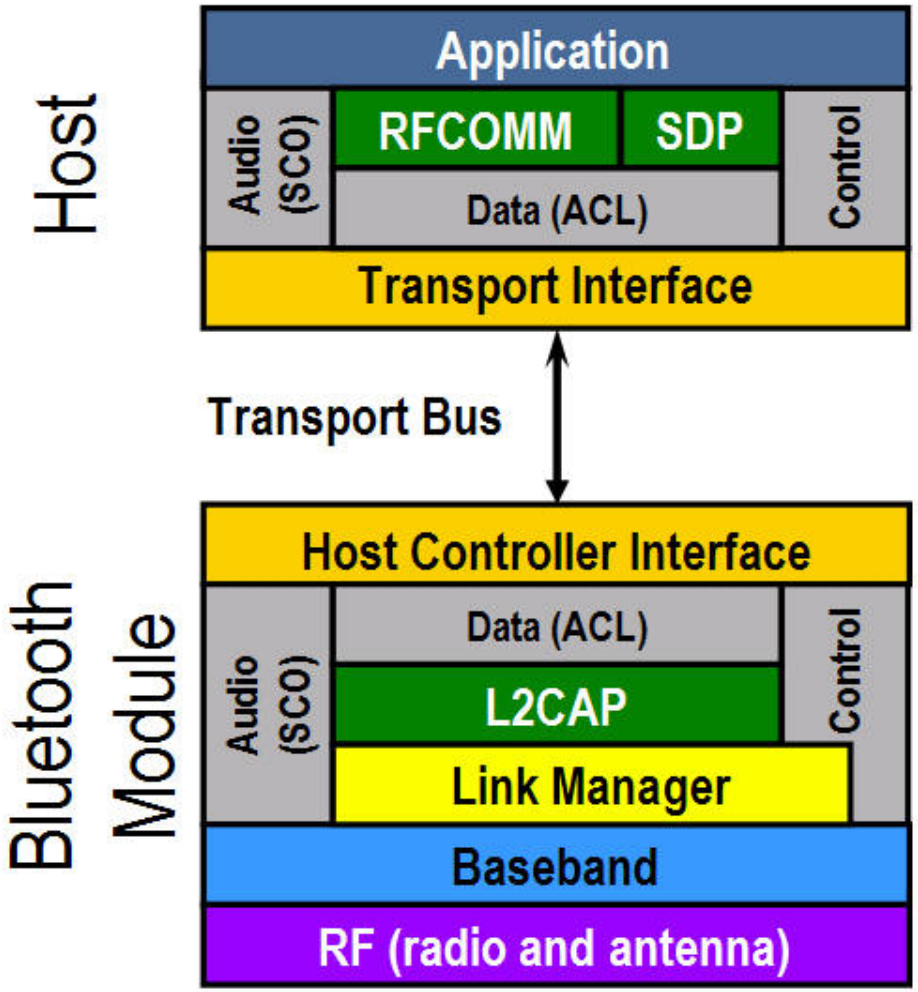
- "Promoter" Class Members
  - Apple
  - Ericsson
  - IBM
  - Intel
  - Lenovo
  - Nokia
  - Microsoft
  - Toshiba

Development History

Year	Version	New Features
1998	1.2	Extended Synchronous Connections (retransmits corrupted packets) *10 meter range, 1Mbps data rate*
2004	2.0	Enhanced Data Rate (EDR) - *3Mbps*
2007	2.1	Secure Simple Pairing
2009	3.0	Bluetooth High Speed (uses ad-hoc 802.11 WiFi - *24Mbps*)
2010-2014	4.0	Bluetooth Low Energy (BLE) protocol Significantly different protocol from Bluetooth “classic” and “high speed” *(50m range, 0.27Mbps data rate, significant power savings)*
2016	5.0	4x range, 2x speed, 8x message capacity

Protocol Stack

Protocol Stack



Host Application

API and Legacy Support Modules

Bluetooth HCI Driver

Physical I/F

Physical I/F

HCI Firmware

Logical Link Control & Data Adaptation

Physical Link Control

Data Processing & Transmission Mgmt.

Transmission/Reception

Transport Protocol Group

- Radio Frequency (RF) Layer
  - Sends and receives modulated bit streams
- Baseband Layer
  - Packet framing, timing, and link flow control
- Link Manager
  - Manages connection states and power management
  - Enforcing Fairness among slaves
- Logical Link Control & Adaptation Protocol (L2CAP)
  - Multiplexing of higher level protocols
  - Segmentation & reassembly of large packets
  - Device discovery & QoS

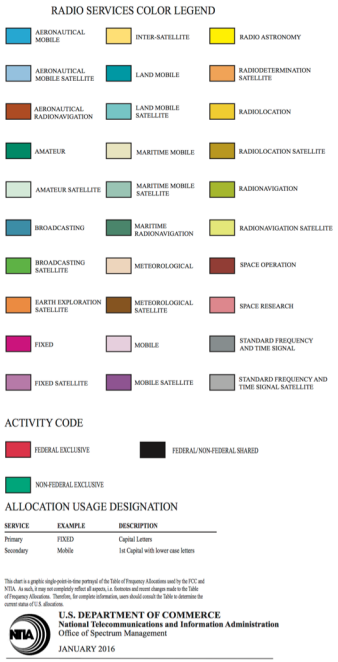
Middleware Protocol Group

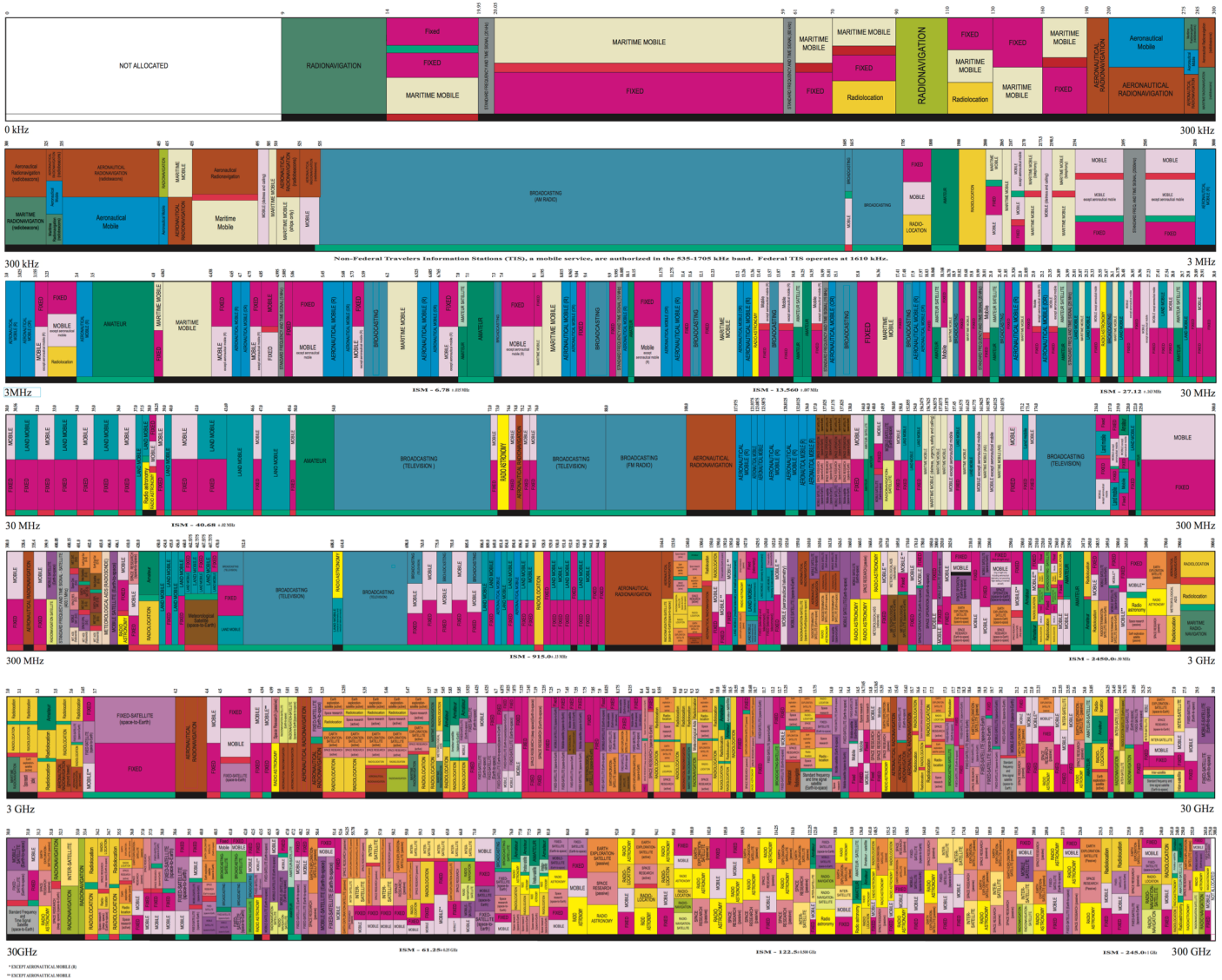
- Service Discovery Protocol (SDP)
  - Means for applications to discover device info, services and its characteristics
- TCP/IP
  - Packet data communication
- RFCOMM
  - Cable replacement protocol
  - Emulates serial port

Physical Layer (Radio)

UNITED STATES FREQUENCY ALLOCATIONS

THE RADIO SPECTRUM





https://www.ntia.doc.gov/page/2011/united-states-frequency-allocation-chart [Last Update: 2016]

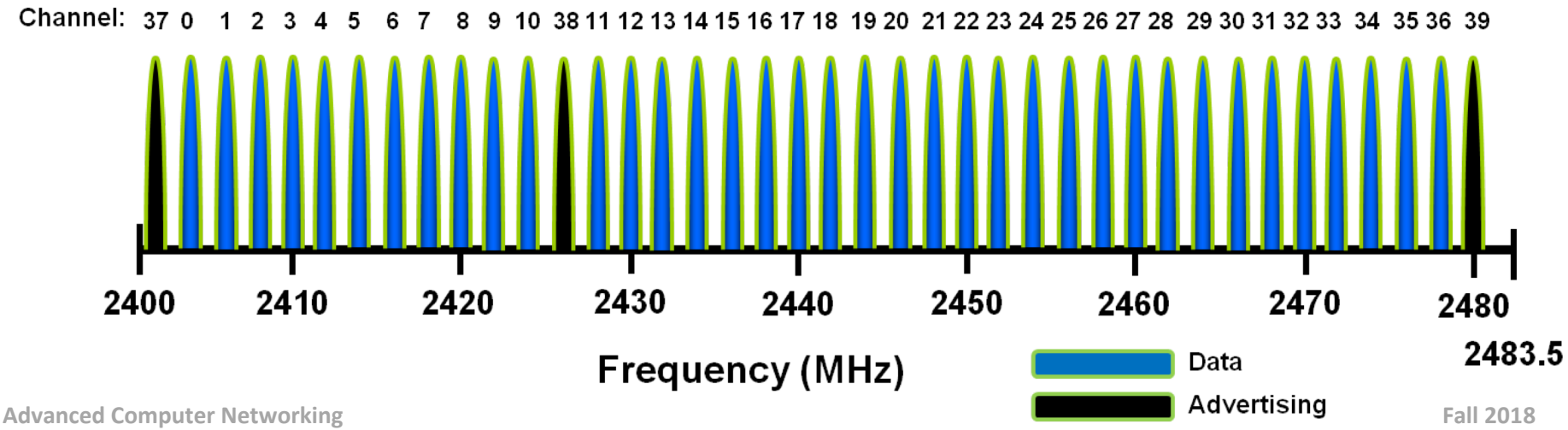
Physical Layer

- Frequency: 2.4000-2.4835GHz
  - **I**ndustrial, **S**cientific and **M**edical **(ISM)** radio band
- Other users in 2.4-2.5GhZ ISM band
  - ZigBee personal area networks (IEEE 802.15.4)
  - WiFi (IEEE 802.11)
  - Cordless phones
  - *All manner of unlicensed cruft*
  - Microwave ovens (*use this frequency for heating)*

Physical Layer

- Channels
  - Bluetooth "Classic" – 79 1MHz channels
  - Bluetooth Low Energy – 40 2MHz channels

Bluetooth Low Energy (BLE) Channels

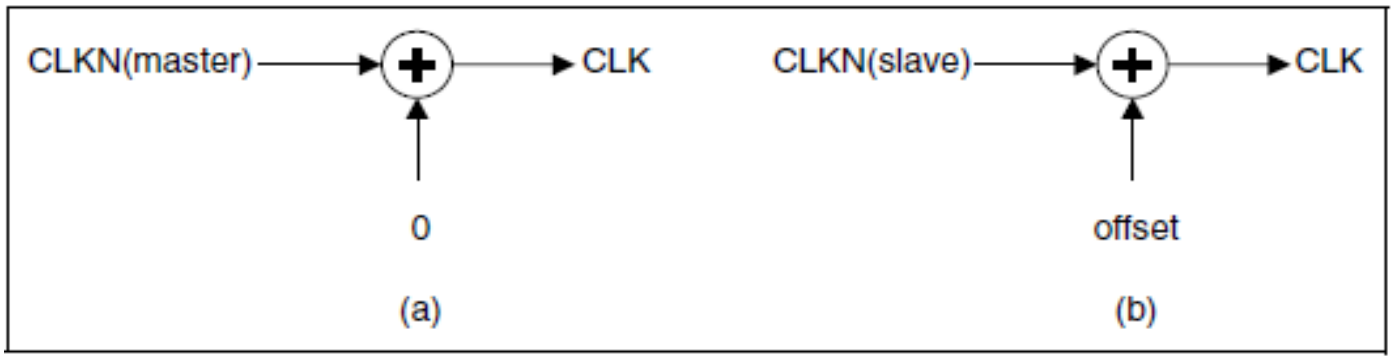


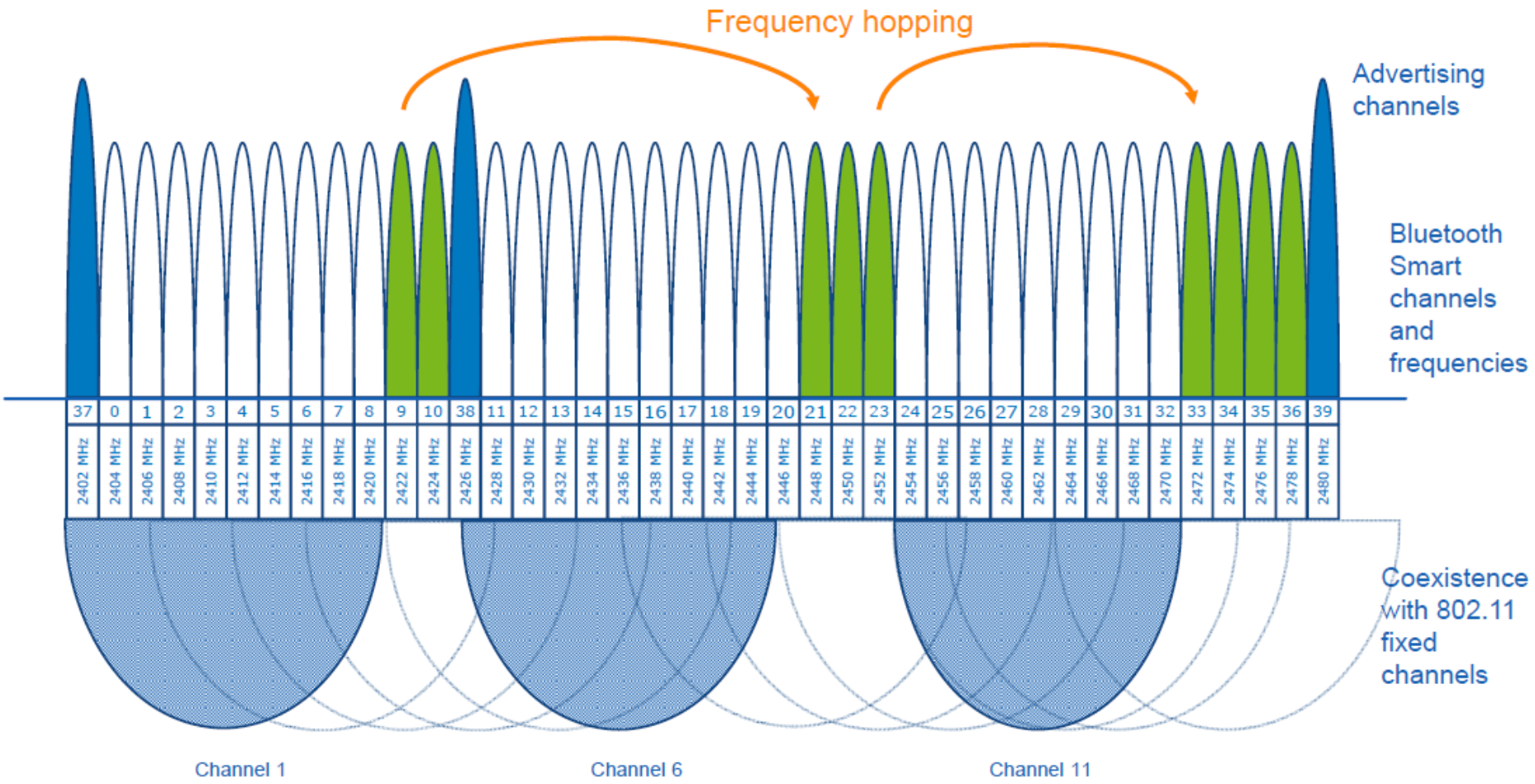
Physical Layer

- Frequency Hopping Spread Spectrum
  - Change frequency with every packet
  - Packets can be 1, 3, or 5 slots long (slot = 625us)
  - 1600 hops/second (1/1600 = 625us)
  - Pseudo-random frequency hop sequence seeded by master MAC address and controlled by master clock

Piconet Clock

- All timing and scheduling in Piconet is based on single clock (Master device)
- Clock is derived from the hardware clock plus an offset
  - Master device has an offset of 0
  - Slave devices obtain offset from network (master)
- All hardware clocks run at the same nominal rate, but mutual drift causes inaccuracies
  - Slaves offsets must be regularly updated

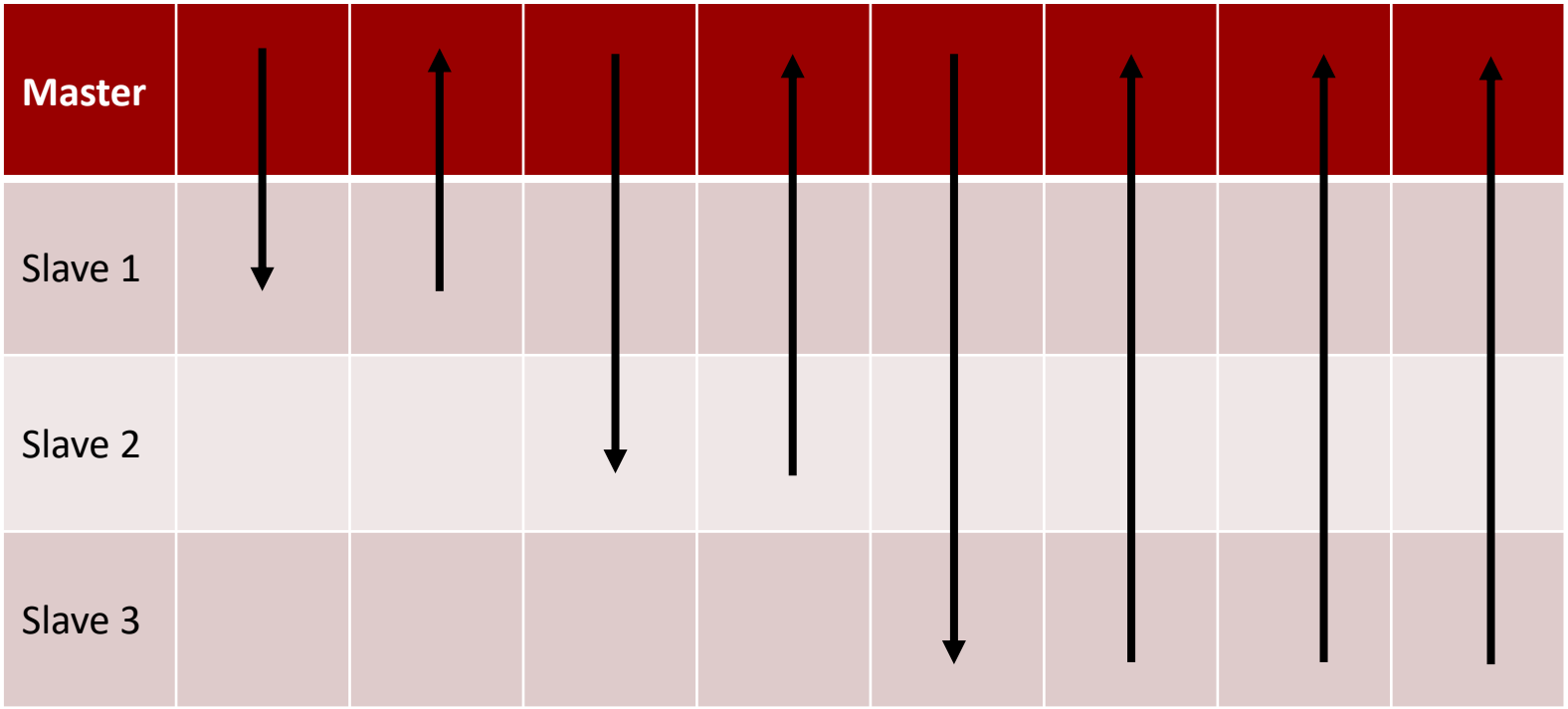



Frequency hopping
Advertising channels
Bluetooth Smart channels and frequencies
37 0 1 2 3 4 5 6 7 8 9 10 38 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 39
2402 MHz 2404 MHz 2406 MHz 2408 MHz 2410 MHz 2412 MHz 2414 MHz 2416 MHz 2418 MHz 2420 MHz 2422 MHz 2424 MHz 2426 MHz 2428 MHz 2430 MHz 2432 MHz 2434 MHz 2436 MHz 2438 MHz 2440 MHz 2442 MHz 2444 MHz 2446 MHz 2448 MHz 2450 MHz 2452 MHz 2454 MHz 2456 MHz 2458 MHz 2460 MHz 2462 MHz 2464 MHz 2466 MHz 2468 MHz 2470 MHz 2472 MHz 2474 MHz 2476 MHz 2478 MHz 2480 MHz
Coexistence with 802.11 fixed channels
Channel 1
Channel 6
Channel 11

Medium Access Control: Time Division Duplexing (TDD)

Slots: 625us

Hop Frequency Every Slot



Master transmits in even slots, Slave(s) transmit in odd slots

Medium Access Control: Time Division Duplexing (TDD)

- Master device uses polling approach to schedule transmissions and eliminate collisions within Piconet
  - **M** polls specific **S** *by address*
  - **S** responds with data
    (all other **S's** are silent)
  - **M** polls all **S's** in round-robin fashion

Power Classes (Bluetooth Classic)

- Transmit power (and range) varies by Bluetooth device

Class Number	Max Output Power (dBm)	Max Output Power (mW)	Typical Range
Class 1	20 dBm	100 mW	100 m
Class 2	4 dBm	2.5 mW	10 m
Class 3	0 dBm	1 mW	1 m
Class 4	-3 dBm	0.5 mW	0.5 m

- Bluetooth Low Energy: 10mW max, 10m range

Name	Bluetooth Classic	Bluetooth 4.0 Low Energy (BLE)	ZigBee	WiFi
IEEE Standard	802.15.1	802.15.1	802.15.4	802.11 (a, b, g, n)
Frequency (GHz)	2.4	2.4	0.868, 0.915, 2.4	2.4 and 5
Maximum raw bit rate (Mbps)	1-3	1	0.250	11 (b), 54 (g), 600 (n)
Typical data throughput (Mbps)	0.7-2.1	0.27	0.2	7 (b), 25 (g), 150 (n)
Maximum (Outdoor) Range (Meters)	10 (class 2), 100 (class 1)	50	10-100	100-250
Relative Power Consumption	Medium	Very low	Very low	High
Example Battery Life	Days	Months to years	Months to years	Hours
Network Size	7	Undefined	64,000+	255

https://learn.sparkfun.com/tutorials/bluetooth-basics

Baseband Layer

Baseband Layer

- In-order delivery of byte streams
- Coordinates frequency hop sequences for synchronization and transmission
  - Important that sender and receiver hop to the same frequency!
- Discovers neighboring devices
- Establishes links
  - Synchronous Connection Oriented (SCO)
  - Asynchronous Connection-Less (ACL)

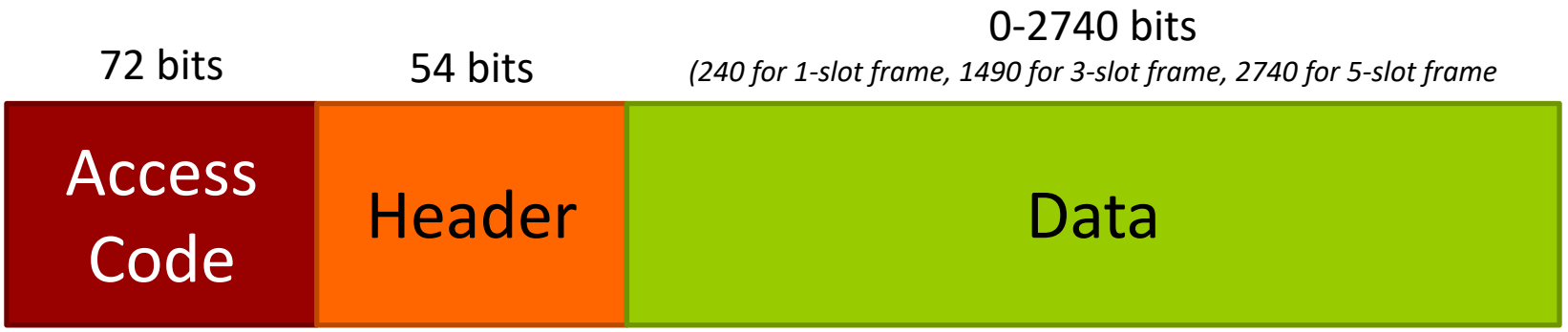
Link Types

- **S**ynchronous **C**onnection **O**riented **(SCO)**
  - Analogy: “circuit switching”
  - Point to point full duplex link between master and slave
  - Provides fixed bandwidth allocation (reserved by master)
  - Established by master, closed by master
  - Application: Voice communication
  - No retransmission if packets are lost

Link Types

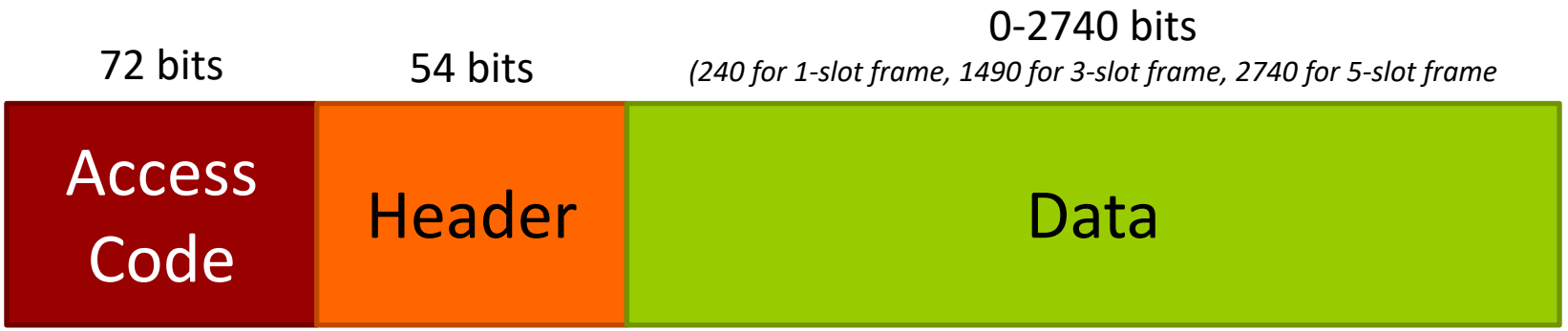
- **A**synchronous **C**onnection-**L**ess **(ACL)**
  - Analogy: “packet switching”
  - Provides momentary bandwidth between master and slave
    - Variable bandwidth based on usage
    - No reserved slots
  - Application: Data communication
    - Retransmissions if packets are lost
    - Used by advanced Bluetooth functions (including high quality audio)

Packet Format



- **Access Code** – Identifies a Piconet
  - Role: Synchronization, identification, signaling (inquiry, paging)
  - 4 bits – Preamble
  - 64 bits – Synchronization (derived from Master ID)
    - If bits don't match *your* Piconet, packet is ignored
  - 4 bits - Trailer

Packet Format



- **Header** – Addressing and Options

18 bits:
- 3 bits – Address of slave *(could be either sender or receiver)*
- 4 bits – Packet Type (12 data types, 4 control types)
- 1 bit – Flow Control
- 1 bit – ARQ (ARQ = **A**utomatic **R**epeat Re**q**uest. i.e. ACKN)
- 1 bit – Sequence Number (Identify retransmitted packets)
- 8 bits – HEC (Header Error Control, i.e. CRC)

- *Encode with 1/3 FEC (Forward Error Correction) for 54 bit total length*

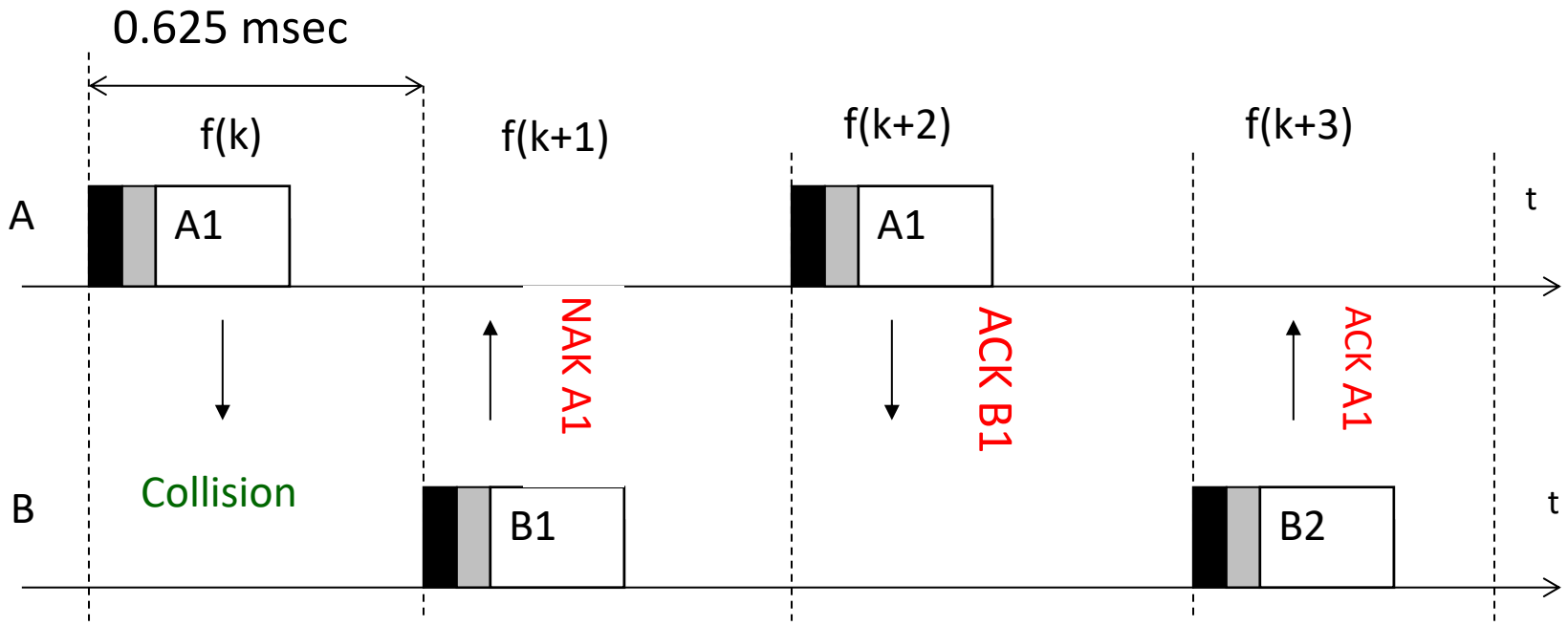
Addressing

- Active Member address **(AM_ADDR)**
  - 3 bits used in packet header
  - Address `001-111` representing slaves 1-7
  - Address `000` representing broadcast address
- Bluetooth Device address **(BD_ADDR)**
  - 48-bit unique addresses (MAC addresses)
    - Upper 24 bits – Organization Unique Identifier (OUI)
    - Lower 24 bits – Randomly assigned to device
  - Eg: `001122334455`

Error Correction

- ISM band is noisy!
- Multiple prevention & recovery mechanisms
  - 1/3 rate FEC for headers and voice
  - 2/3 rate FEC for DM packets (medium data rate)
  - Stop and wait ARQ (sender notified of receipt in next slot)
  - CRC to detect payload errors
- Frequency Hopping Spread Spectrum + Adaptive Frequency Hopping (AFH) helps to avoid continuously noisy bands

Automatic Repeat Request (ARQ)



Sender notified of packet receipt (yes or no) in timeslot immediately following transmission. (Simple stop and wait ARQ scheme)

Packet Types

Control

- **ID** – Signals piconet. Only contains access code
- **NULL** – Contains link control information. Only contains access code and header
- **POLL** – Similar to NULL, forces slaves to return response
- **FHS** – Frequency Hopping Synchronization. Updates real-time clock across piconet

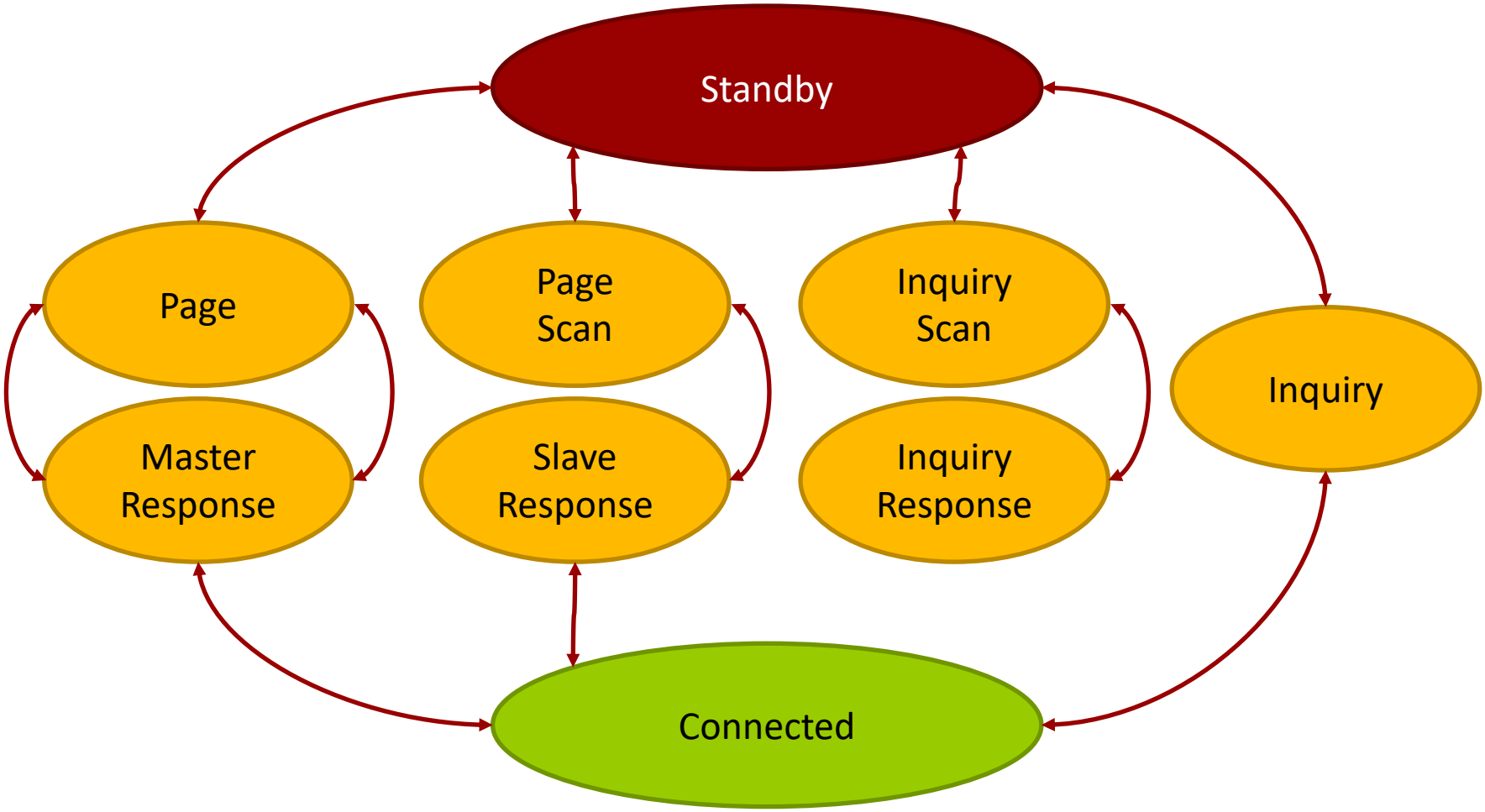
Data

- SCO – Synchronous Connection Oriented
  - **HV1**, **HV2**, **HV3** (High Quality Voice)
  - **DV** (Combined Data/Voice)
- ACL – Asynchronous Connectionless
  - **DM1**, **DM3**, **DM5** (Data, Medium Rate)
  - **DH1**, **DH3**, **DH5** (Data, High Rate)
- *1=1/3FEC, 2=2/3 FEC, 3=No FEC (tradeoff of time vs redundancy)*

Connection Establishment

- Chicken and Egg problem: How do master and slave find each other in a Frequency Hopping Spread Spectrum network?
  - How do they discover each other's existence?
  - Synchronize clock?
  - Share Bluetooth Device address?
  - Agree on pseudo-random Frequency Hop Sequence (FHS)

Connection Establishment State Machine



Master

Slave

User powers on headphones
Mode: Inquiry Scan
Headphone listens for inquiry packets

Phone periodically looks for new devices
Mode: Inquiry
Phone sends Inquiry packet

- *Addressed to GIAC (General/Unlimited Inquiry Access Code)*
- *Repeated on inquiry hop sequence of channels*

Headphone receives Inquiry
Mode: Inquiry Response
Headphone sends inquiry response with device address and FHS

Master

Slave

Phone now knows headset FHS (Frequency Hop Sequence)
Mode: Page
Phone sends Page packet addressed to headset

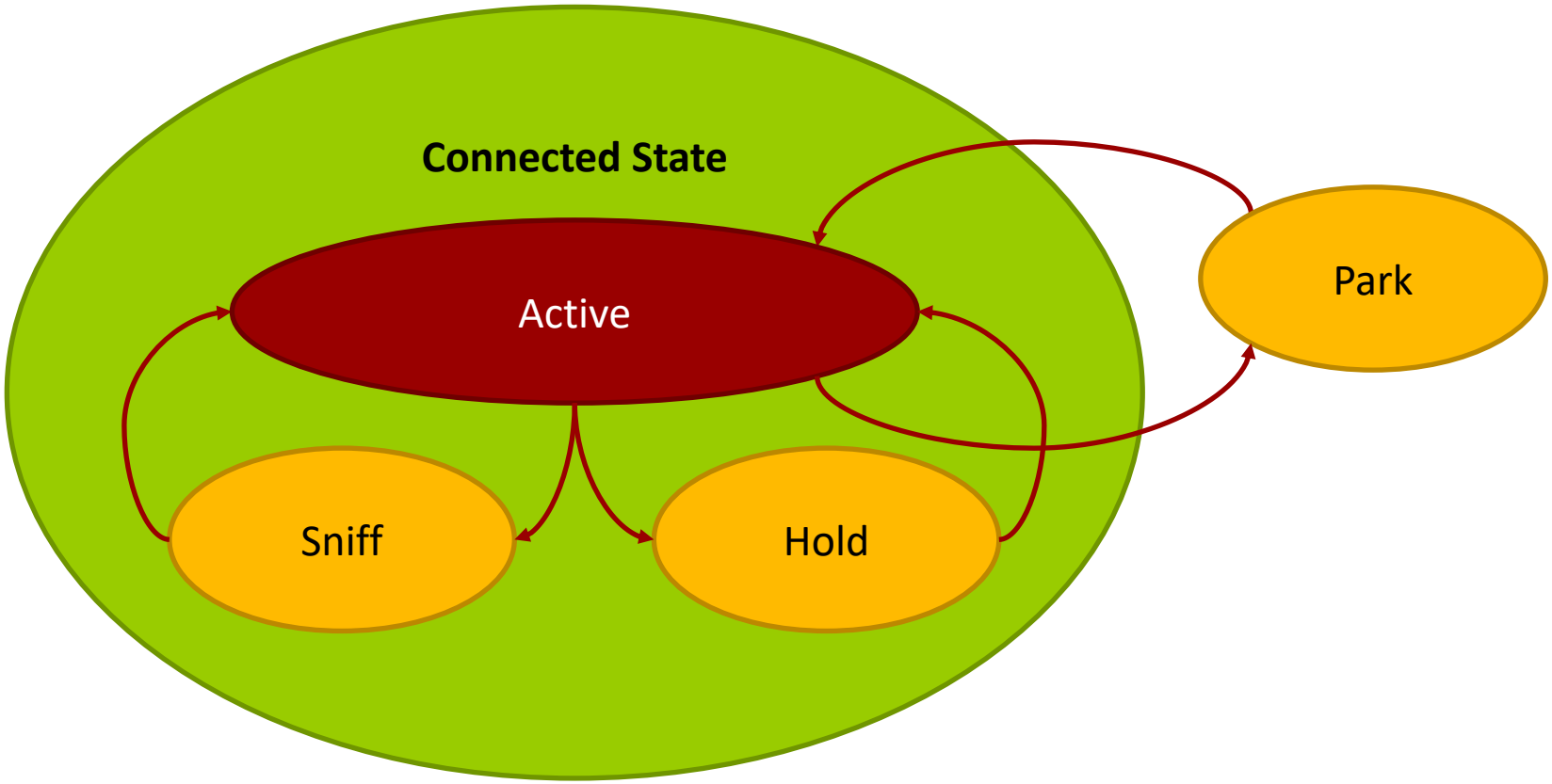
Mode: Page Scan
Headphone listens for page packet

Mode: Master Response
Phone sends FHS and clock to slave

Headphone receives Page
Mode: Page Response
Headphone sends Page response with its ID

Headphone receives Master Response
Mode: Connected
Headphone adopts master FHS and master clock

Connection Establishment State Machine



Modes in Connected State

- **Active** Mode
  - Slave listens (scans) to every master transmission
- **Sniff** Mode
  - Modest power / reduced bandwidth mode
  - Slave does not listen (scan) at every possible M-to-S slot, but at a reduced frequency coordinated with master
- **Hold** Mode
  - Slave does not listen for a time duration arranged with master (duty cycle < 1%)
  - No ACL (SCO only)
- **Park** Mode
  - Low power sleeping mode
  - Gives up Active Member address, maintains synchronization
  - Communication only via broadcast messages (infrequent)

Protocol Stack

- **LMP – Link Management Protocol**
  - Setup and control link between two devices
- **L2CAP – Logical Link Control and Adaptation Protocol**
  - Multiplex multiple logical connections between two devices
  - Segmentation, reassembly, retransmission, QoS
  - Only for **ACL** (Asynchronous Connection-Less) Links
- **SDP – Service Discovery Protocol**
  - Discover which profiles can be used between two devices

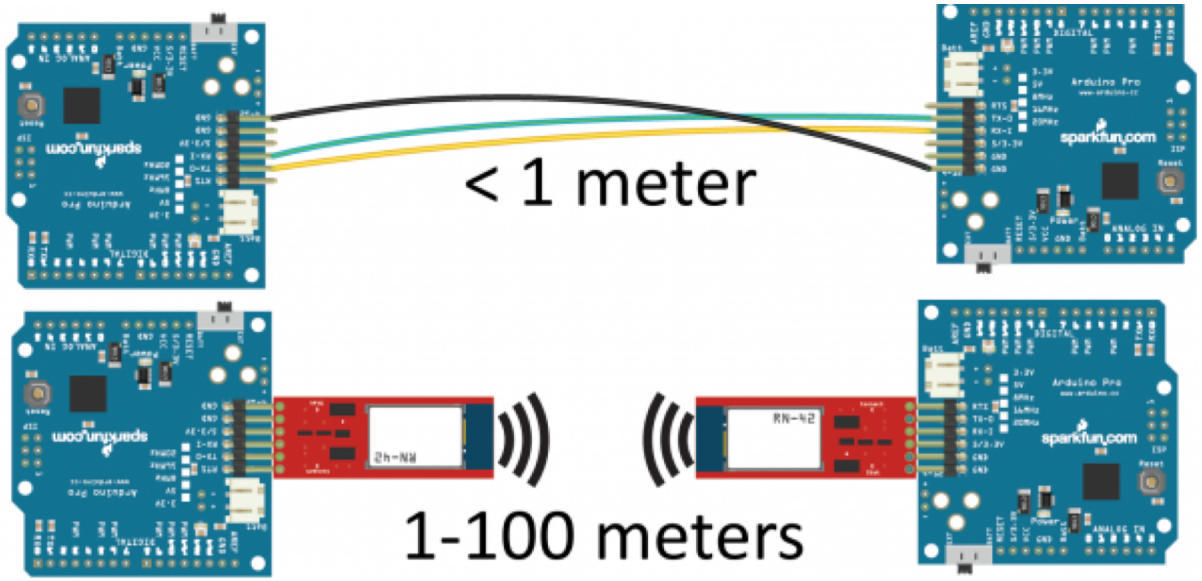
Profiles

Profiles

- Application Layer = “Profiles”
- Example use cases, attributes, services, data formats, … developed by the Bluetooth SIG to accomplish a given task
  - NOT an API
  - Intended to allow interoperability
- 36 profiles currently listed on Wikipedia (exhaustive list?)
  - https://en.wikipedia.org/wiki/List_of_Bluetooth_profiles

Serial Port Profile (SPP)

- Wireless replacement for RS-232 serial port



Human Interface Device (HID)

- Wireless replacement for user input devices
  - Mice, keyboards, joysticks, …
- Wrapper for **USB** Human Interface Device protocol

Hands-Free Profile (HFP) / Headset Profile (HSP)

- Hands-Free Profile (HFP)
  - Car audio (basic)
  - Synchronous Connection Oriented (SCO) link for audio stream (mono)
- Headset Profile (HSP)
  - Headsets (basic)
  - SCO link for audio (mono)
  - AT commands from GSM phones for basic control: ring, answer call, hang up, adjust volume

A2DP / AVRCP

- Examples: Headsets, soundbars, car audio, etc…
- Advanced Audio Distribution Profile (A2DP)
  - One-way audio stream (but higher quality, 2 channels)
  - Codecs: SBC, MPEG-1, MPEG-2, AAC, ATRAC
- A/V Remote Control Profile (AVRCP)
  - Typically implemented in parallel with A2DP
  - Remote control of audio stream (pause, fast forward, …)
  - Metadata (artist, title, …)

Profiles

- Many other profiles…
  - Basic Imaging Profile (BIP)
  - Basic Printing Profile (BPP)
  - Personal Area Networking Profile (PAN)
  - **Mesh Profile (MESH)**

Mesh Networking

Bluetooth History

- Bluetooth "Classic" - Bluetooth Basic Rate / Enhanced Data Rate (BR / EDR)
  - Wireless headsets, mice, keyboards, …
  - `1:1` communication pattern (Master<->Slave)
- Bluetooth Low Energy (BLE)
  - Activity trackers, wearables, beacons …
  - `1:many` communication pattern (Beacons broadcasting)
- What's next? `Many:many` communication pattern (i.e. **Mesh** networking)
  - Devices can interconnect and form larger networks
  - Devices can relay messages for other devices not in direct radio range
  - Applications in IoT devices
    - Building automation, commercial lighting, sensor networks
  - Introduced in Bluetooth 5 (2017)
  - Competes with other IoT wireless standards: Zigbee, Z-Wave, LoRa
    - ***One Bluetooth to Rule Them All!***

Bluetooth 5 Mesh Networking

- High-Level API
  - **Publish**: Devices send messages to specific mesh addresses (e.g. binary value corresponding to concept like “front yard lights”)
  - **Subscribe:** Devices receive messages that were sent to specific mesh addresses
- New device mode: **Relay**
  - Relays retransmit messages received from other devices, thus extending network range
  - Maximum of 127 hops

Bluetooth 5 Mesh Networking

- Routing algorithm: **Flooding**
  - All devices within range receive messages
  - All devices which are *relays* retransmit messages
- Advantages
  - Simplicity (both in relay implementation and lack of end-user configuration)
- Disadvantages
  - Duplication of messages / overhead

Bluetooth 5 Mesh Networking

- All packets include TTL
  - Limits number of hops / times packet can be retransmitted
- All devices transmit **heartbeat** messages periodically
  - Allow relays to learn about topology and distance from other devices
  - TTL is *adaptive* – Set based on size (max hops) of network
- All devices have **message cache**
  - Have I see this message before? If yes, no need to process it further (or forward it, etc...)

Bluetooth 5 Mesh Networking

- New concept for low-power devices as part of wireless mesh
- New device mode: **Low power nodes**
  - Battery powered, limited energy, must sleep for majority of time so battery can last for months/years
  - Example: Temperature sensor
    - Reads temperature and transmits periodically
    - Sleeps otherwise to conserve energy
  - But what if the temperature sensor also needs to *listen* for infrequent data from network? (New security keys, for example, or admins changing the reporting interval…)
- New device mode: **Friend**
  - Device that is not power constrained
  - Will store messages addressed to low power nodes and deliver them upon request by lower power node (which can remain sleeping)

Bluetooth 5 Mesh Networking

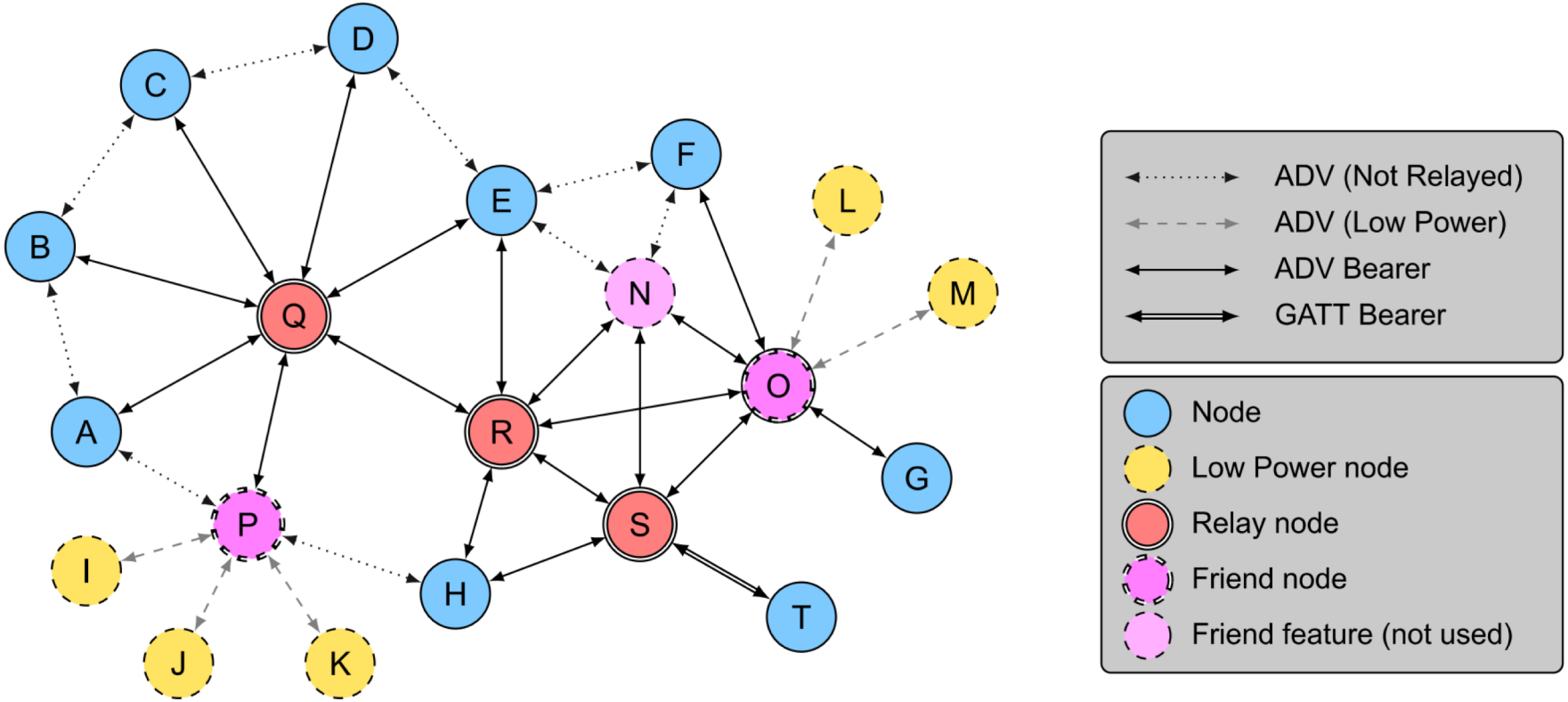
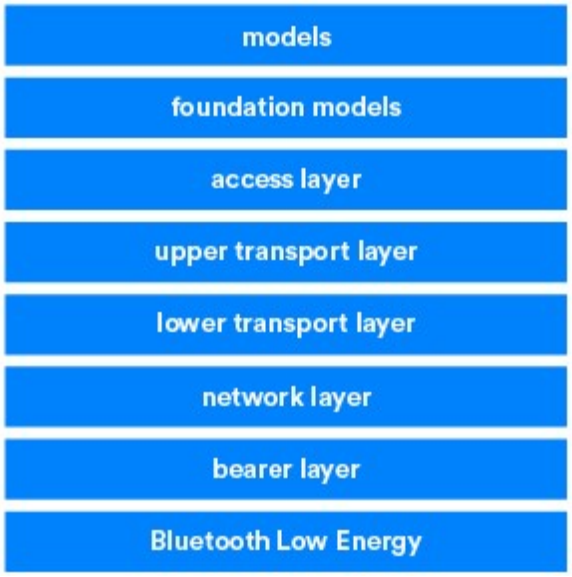


Figure 2.8: Example Topology of a mesh network

https://www.bluetooth.com/specifications/mesh-specifications

Bluetooth 5 Mesh Networking



- Mesh network adds a new protocol stack which sits **on top of** Bluetooth Low Energy stack
- *Complexity….*

https://blog.bluetooth.com/an-intro-to-bluetooth-mesh-part2

Bluetooth 5 Mesh Security

- “Bluetooth® mesh networking was designed with security as its number one priority and from the ground up.”
  - https://blog.bluetooth.com/bluetooth-mesh-security-overview
- So, problem solved?

Bluetooth 5 Mesh Security

- Encryption: All mesh messages are encrypted and authenticated
- Separation of Concerns:
  - Network security
  - Application security
  - Device security
- Area isolation: Mesh networks can be divided into subnets (cryptographically separate)
- Key refresh: Security keys can be rotated
- Message obfuscation: Privacy for senders
- Replay attack protection
- Trashcan attack protection: Nodes can be removed securely from network (keys blacklisted, new keys provisioned and deployed)
- Secure device provisioning: Add nodes to mesh network in controlled process

https://blog.bluetooth.com/bluetooth-mesh-security-overview

Bluetooth 5 Mesh Security

- Three levels of security keys enable separation of concerns
- **Device Key**
  - Used for device provisioning and configuration
- **Network Key**
  - Allow node to join a subnet
  - Allow node to decrypt and authenticate up to the network layer
- **AppKeys**
  - Allow node to decrypt and authenticate application-layer data
- Design flexibility: Node can relay messages for the network (subnet) without the ability to decrypt/read/modify the application data being passed on the subnet

Bluetooth 5 Mesh Security

- Privacy key (based on **NetKey**) encrypts network headers such as source address
  - Prevents passive analysis (from non-subnet members without NetKey) from tracking nodes and users
- Sequence Number (SEQ) field allows replay attacks to be identified and ignored

Security

Security

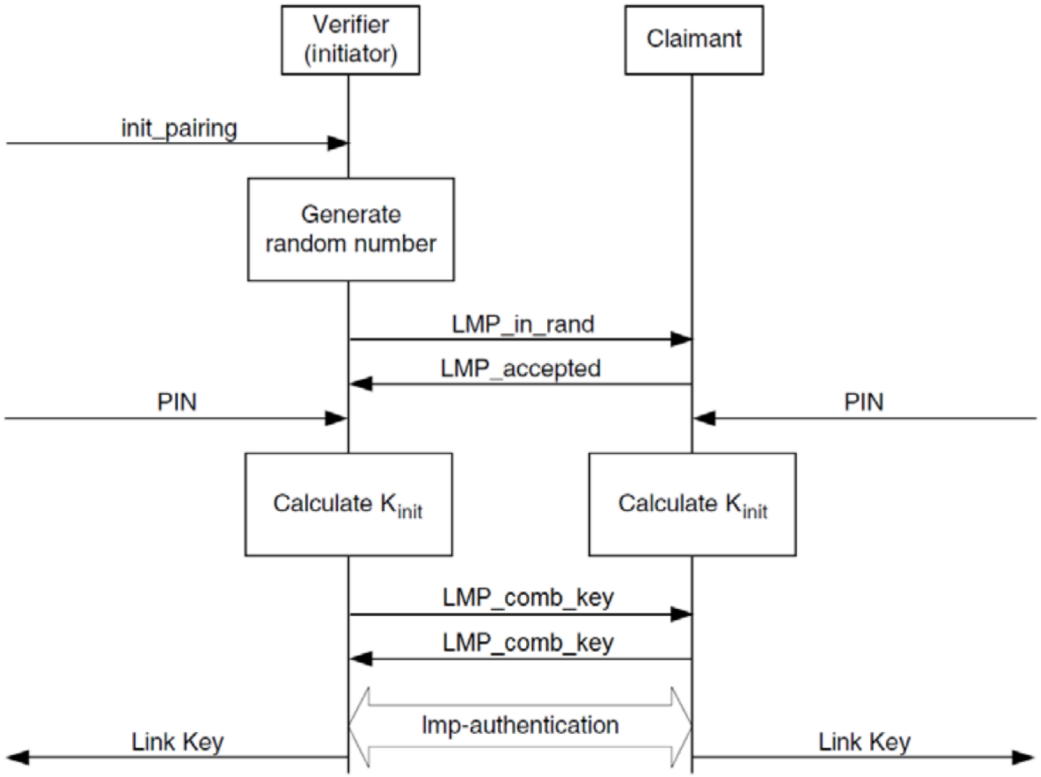
- Early Bluetooth (1.x) was a security wasteland…
  - Was possible to obtain link key just by passively sniffing the pairing process
- Method: Link Manager Protocol (LMP) Pairing (w/ PIN-code)

Security

- Goal: Establish *shared secret* between two Bluetooth devices: **Link Key**
- Link Key is used to authenticate two devices to each other
  - Can be re-used when devices re-pair in the future
- **Encryption Key** is derived from the Link Key and used to exchange data between two devices
  - Encryption key changes for each connection

Security – LMP Pairing

- Shared inputs to algorithm
  - BDADDR of the two devices
  - 16-byte random number created by the initiator
  - PIN code entered by the user on both devices (except for devices with “fixed PIN”)
    - Not transmitted on air – “secret”



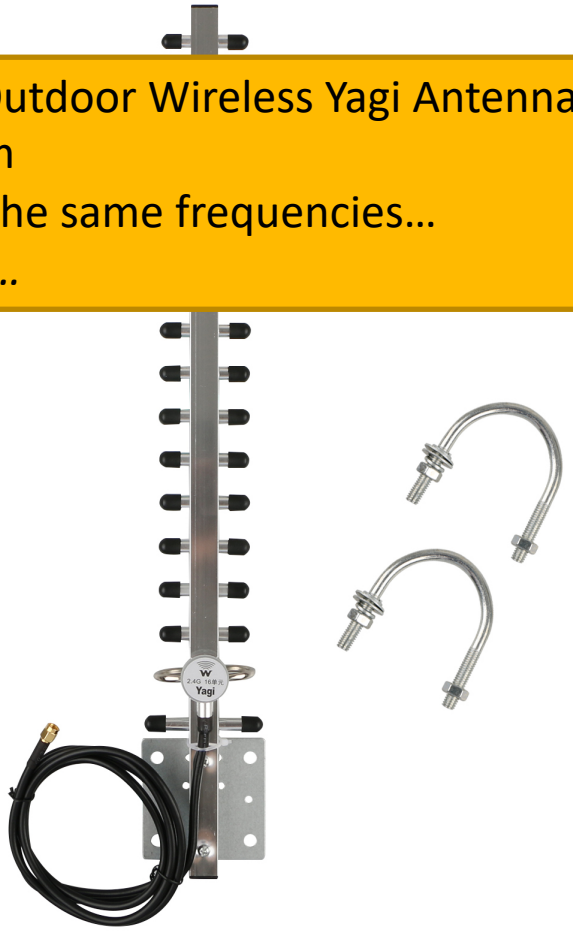
https://www.ellisys.com/technology/een_bt07.pdf

Security – LMP Pairing

- Weakness – PIN code is the only part of the Link Key calculation that is not transmitted over the air, but it is only 4 digits
  - 10,000 keys for attacker to brute force if they have passively captured the rest of the LMP Pairing process
- *Bluetooth is low power and short range, do we really need to worry about security?*

Security

2.4GHz 25dBi Directional Outdoor Wireless Yagi Antenna - $12.97 from Walmart.com
Intended for WiFi, but it's the same frequencies…
Combine with an amplifier…



Ubertooth One wireless development platform
https://shop.hak5.org/products/ubertooth-one

Security

- Bluetooth 2.1 made a serious effort at fixing security problems at a *protocol* level with **Secure Simple Pairing**
  - *This design has secure in the name….*
  - Still possible to have implementation errors (specification incorrectly implemented or specification unclear)

Security – Secure Simple Pairing

- Increased security: PIN code is not part of Link Key calculation
  - *May be present for device authentication only*
- Elliptic-Curve Diffie-Hellman (ECDH) Key Exchange used to establish shared secret (192-bit number) that is not transmitted over the air
  - Each device has public key (shared) and private key (not shared)

Security

- Bluetooth 4.1 made further security enhancements under the name **Secure Connections**

Security Mechanism	Legacy	Secure Simple Pairing	Secure Connections
Encryption	E0	E0	AES-CCM
Authentication	SAFER+	SAFER+	HMAC-SHA256
Key Generation	SAFER+	P-192 ECDH HMAC-SHA-256	P-256 ECDH HMAC-SHA-256

Table 1.1: Security algorithms

BlueBorne Vulnerabilities (2017)

BlueBorne Vulnerabilities

- Reported by Armis (IoT security Firm), Sept 12 2017
- Not 1, not 2, …. but 8 zero-day vulnerabilities across a variety of Bluetooth platforms
  - Android, iOS, Windows, Linux, Amazon Echo, …
- Think “airborne” (as in, contagious…)

https://armis.com/blueborne/

http://go.armis.com/blueborne-technical-paper

BlueBorne Attack Overview

1. Locate active Bluetooth devices (even if not set to discoverable mode)
2. Obtain MAC address of Bluetooth device
3. Probe device, use OS fingerprinting to tailor attack
4. Exploit Exploit Exploit
5. Choice:
    1. Man-in-the-Middle attack and control communication?
    2. Full control over device?

BlueBorne on Android

- **CVE-2017-0785** – Information Leak Vulnerability
- Weakness in Service Discovery Protocol (SDP)
  - Used to identify services offered by neighboring devices

- Attacker sends set of requests to service
- Service discloses memory bits in reply
- Potential for attacker to recover encryption keys, eavesdrop on communication, etc…

BlueBorne on Android

- **CVE-2017-0781** – Remote Code Execution Vuln #1
- Weakness in Bluetooth Network Encapsulation Protocol (BNEP)
  - Used for Internet connection sharing (i.e. tethering)

- Attacker can accurately trigger memory corruption, which can be leveraged to produce arbitrary code (which is then executed)
- **No user interaction, authentication, or pairing required (due to lack of authorization validation)**

BlueBorne on Android

- **CVE-2017-0782** – Remote Code Execution Vuln #2
- Weakness in Bluetooth Network Encapsulation Protocol (BNEP), Personal Area Networking (PAN) profile
  - Used for Internet connection sharing (i.e. tethering)
  - Specifically, establishing an IP network
- Attacker can accurately trigger memory corruption, which can be leveraged to produce arbitrary code (which is then executed)
- **No user interaction, authentication, or pairing required (due to lack of authorization validation)**

BlueBorne on Android

- **CVE-2017-0783** – Bluetooth Pineapple – MITM Attack
- Weakness in Personal Area Networking (PAN) profile
  - Used in tethering to establish an IP network
- Attacker can create malicious network interface, reconfigure IP routing, and force network traffic through malicious interface
- **No user interaction, authentication, or pairing required (due to lack of authorization validation)**

Android Take Over Demo

armis

https://www.youtube.com/watch?v=Az-l90RCns8

BlueBorne on Windows

- **CVE-2017-8628** – Bluetooth Pineapple – MITM Attack
- *Same as Android, same flawed interpretation of specification*
- Weakness in Personal Area Networking (PAN) profile
  - Used in tethering to establish an IP network
- Attacker can create malicious network interface, reconfigure IP routing, and force network traffic through malicious interface
- **No user interaction, authentication, or pairing required (due to lack of authorization validation)**

BlueBorne™

Windows MiTM Demo

armis

https://www.youtube.com/watch?v=QrHbZPO9Rnc

BlueBorne on Linux

- **CVE-2017-1000250** – Information Leak Vulnerability
- *Similar to Android, same flawed interpretation of specification*
- Weakness in Service Discovery Protocol (SDP)
  - Used to identify services offered by neighboring devices
- Attacker sends set of requests to service
- Service discloses memory bits in reply
- Potential for attacker to recover encryption keys, eavesdrop on communication, etc…

BlueBorne on Linux

- **CVE-2017-1000251** – Stack Overflow in BlueZ
- Stack overflow in Bluetooth stack in Linux kernel
- Weakness in Logical Link Control and Adaption Protocol (L2CAP)
- Attacker can cause memory corruption, and leverage that to gain full control of device

BlueBorne on iOS

- **CVE-2017-14315** – Remote Code Execution
- Weakness in Low Energy Audio Protocol (LEAP)
  - Audio commands are not properly validated
- Attacker can cause memory corruption, and leverage that to gain full control of device