CS 439: Wireless Networking

MAC Layer – Bluetooth

© CS 439 Staff, University of Illinois

Bluetooth

- Harald Blaatand "Bluetooth" II
  - King of Denmark c. 958–986
- Runic stones in his capital city of Jelling
  - The stone's inscription ("runes") says:
    - Harald Christianized the Danes
    - Harald controlled the Danes
    - Harald believes that devices shall seamlessly communicate [wirelessly]

Where did the Bluetooth logo come from?

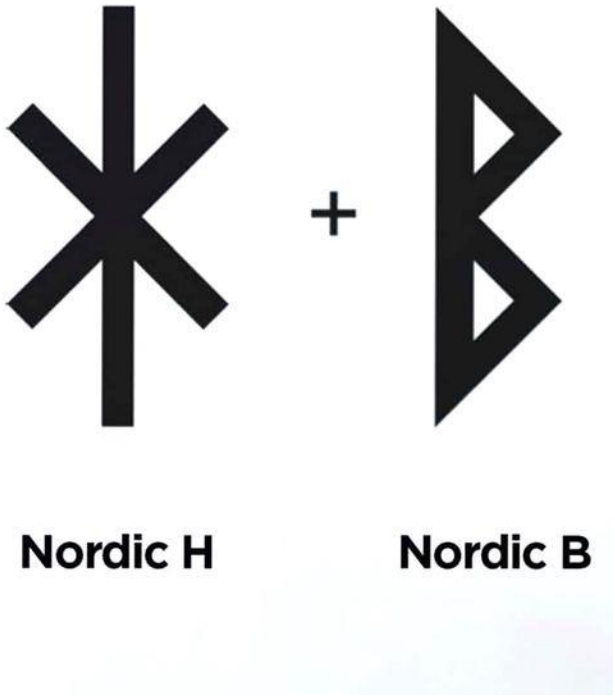
Where did the Bluetooth logo come from?

- Harald

Nordic H

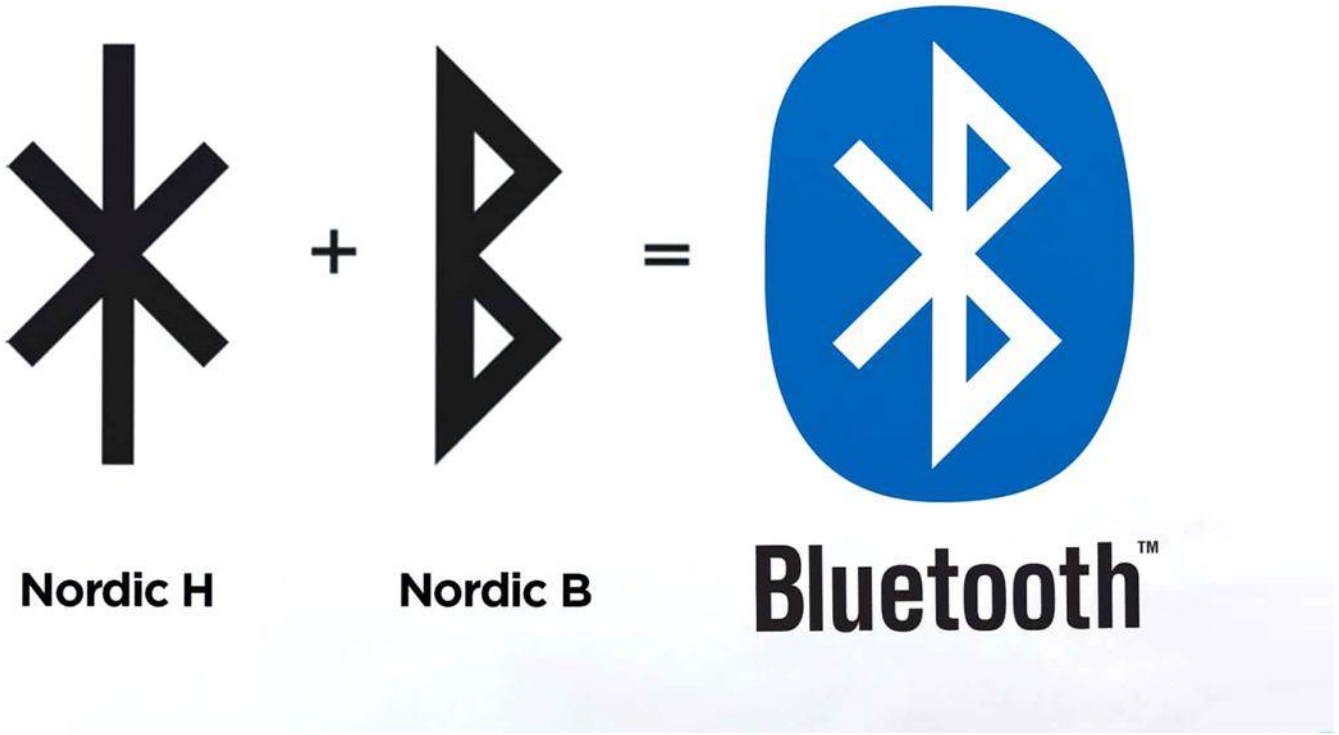
Where did the Bluetooth logo come from?

- Harald + Blaatand



Where did the Bluetooth logo come from?

- Harald + Blaatand = Bluetooth

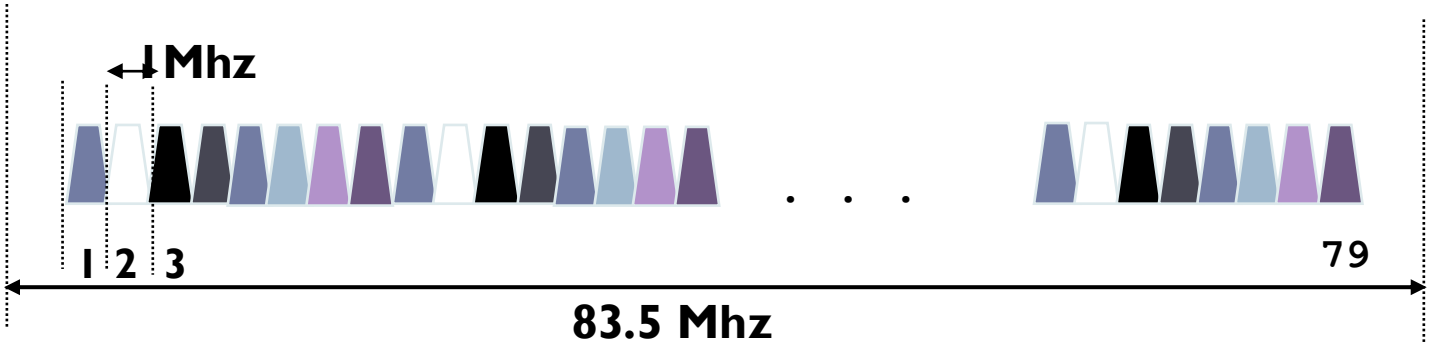


Bluetooth Progression

Initial
version

Classic Bluetooth

- Cable replacement
  - 2.4 GHz
  - FHSS over 79 channels (of 1MHz each), 1600hops/s
  - 1 Mbps basic rate (BR)
    - 2 and 3 Mbps with EDR (Bluetooth 2.0+)



Classic Bluetooth

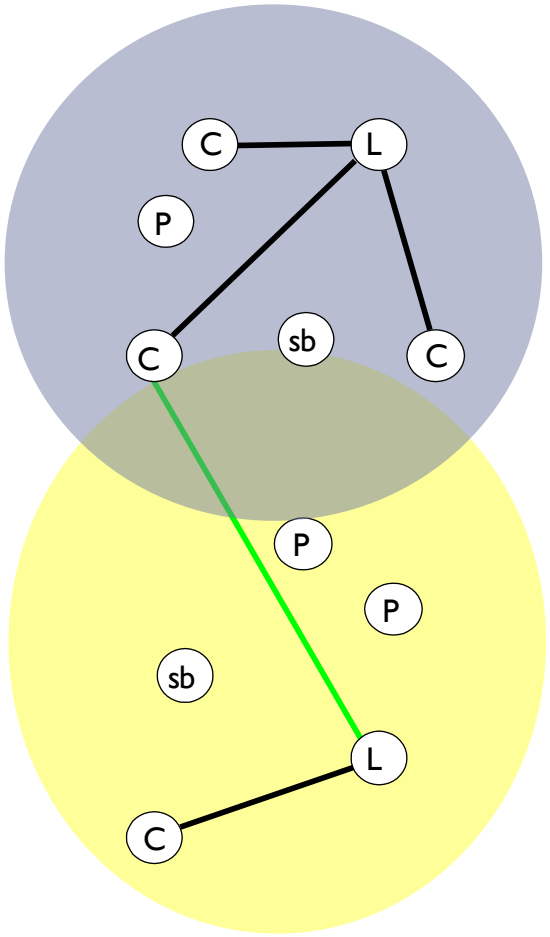
- Cable replacement
  - 2.4 GHz
  - FHSS over 79 channels (of 1MHz each), 1600hops/s
  - 1 Mbps basic rate (BR)
    - 2 and 3 Mbps with EDR (Bluetooth 2.0+)
  - Coexistence of multiple piconets
  - Range set by power class: ~1 m, ~10 m, or ~100 m
    - Max Tx power: 0 dBm (1 mW), 4 dBm (2.5 mW), or 20 dBm (100 mW)

Classic Bluetooth: Power Consumption

- Transmit power is set by device class
  - Class 3 — 0 dBm (1 mW) — ~1 m
  - Class 2 — 4 dBm (2.5 mW) — ~10 m
  - Class 1 — 20 dBm (100 mW) — ~100 m
    - Class 1 devices must support power control:
    - Peer asks for less power when the link is strong

Bluetooth Network Topology

- Radio designation
  - Connected radios can be leaders or clients
  - Radios are symmetric (same radio can be leader or client)
- Piconet
  - Leader can connect to 7 simultaneous or up to 255 parked clients per piconet
  - Each piconet has maximum capacity (1 Mbps)
  - Unique hopping pattern/ID
- Scatternet
  - High capacity system
  - Minimal impact with up to 10 piconets within range
  - Radios can share piconets!

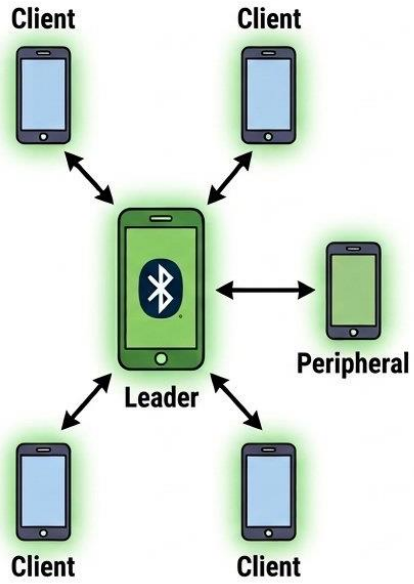


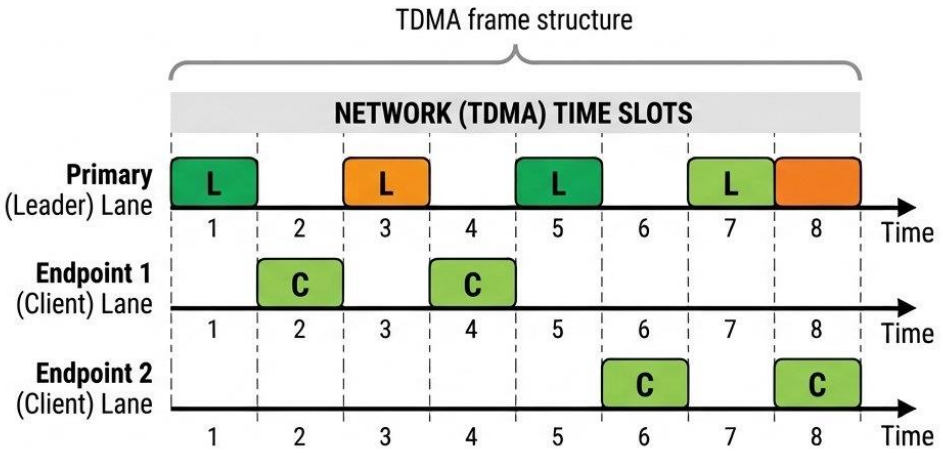
Bluetooth – Contention-free MAC

- Leader performs medium access control
  - Schedules traffic through polling
- Time slots alternate between leader's and clients' transmission
  - Leader-client
    - Leader includes client address
  - Client-leader
    - Only clients chosen by leader in previous leader-client slot allowed to transmit.
  - If leader has data to send to a client, client polled implicitly; otherwise, explicit poll.

Bluetooth – Contention-free MAC

- Leader performs medium access control
  - Schedules traffic through polling





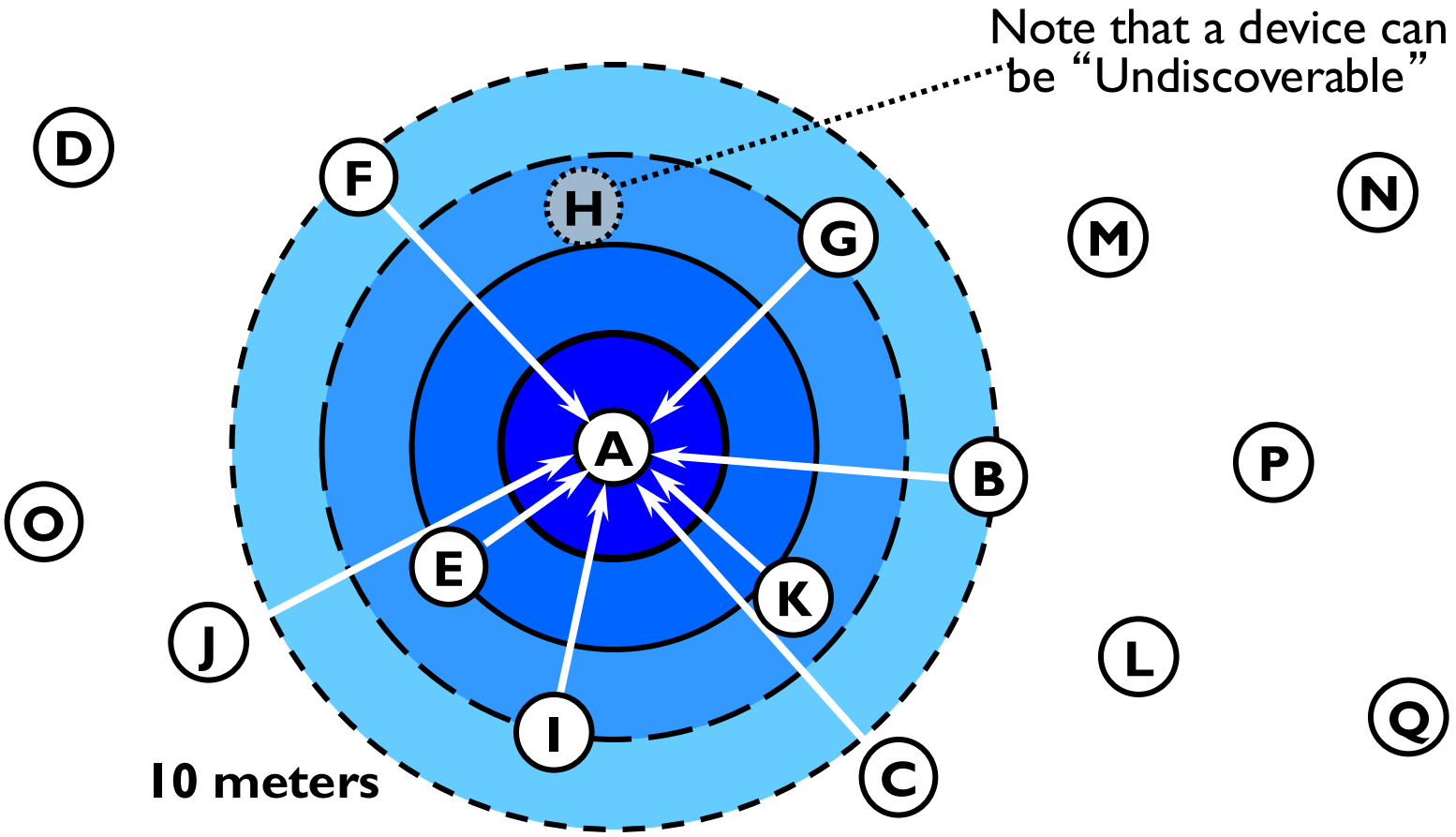
Classic Bluetooth: Power Consumption

- The link is always on while connected
  - The leader polls every 625 µs slot
    - Both radios keep listening even with no data
  - Good for streaming (audio, EDR 2-3 Mbps)
    - Costly for sensors and wearables
- Low-power connected modes (the link stays alive)
  - Sniff
    - Client listens only at agreed anchor points every T_sniff slots
  - Hold
    - Client skips the channel for a negotiated time, then resumes
  - Park
    - Client gives up its active address and wakes only for periodic beacons
- Longer sleep saves energy but adds latency

Bluetooth Device Discovery

- Device discovery
  - Send inquiry
    - A request for nearby devices (within 10 meters)
  - Discoverable devices
    - Send inquiry response
    - Include address
  - Can take up to 10.24 seconds
    - After which the inquiring device should know everyone within 10 meters of itself

Bluetooth Device Discovery - Inquiry



After inquiry procedure, A knows about others within range

Bluetooth Inquiry

- Sender
  - Inquiry sent on 16 different frequencies
    - 16 channel train
    - One full 16 channel train takes 10ms
- Receiver (device in standby mode)
  - Scans long enough sending inquiry on 16 frequencies
  - Scan must be frequent enough to guarantee wake up during a 16 channel train
    - Enters inquiry scan state about every 1.28 s (default), for 11.25 ms

Bluetooth Inquiry - Reliability

- Challenge
  - Noisy channels
  - Lost packets
    - Each 16-channel train is repeated 256 times (2.56 s) before switching to the other train
      - 4 passes = up to 10.24s
    - Designed to successfully communicate at least once with all devices within range

Bluetooth Progression

Initial
version

Bluetooth Progression

Initial version

Significantly Increased Speed

Data Rate ↑↓ (3Mbps)

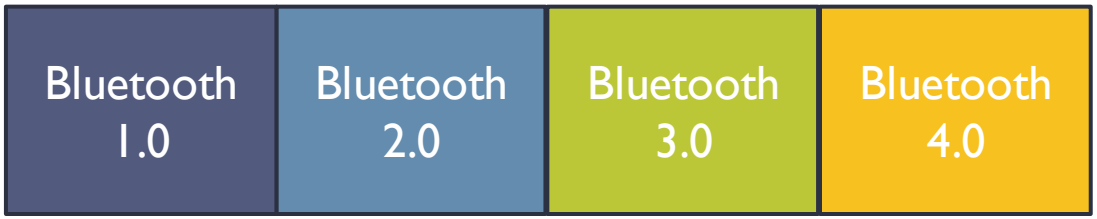
Improved power consumption

Expand to audio devices

Bluetooth Progression

Bluetooth + 802.11 (AMP)
Up to 24 Mbps
Efficiency/ Connection Stability/ Power Control …

Bluetooth Progression



Initial version	Significantly Increased Speed	High-Speed Bluetooth	Bluetooth Low Energy

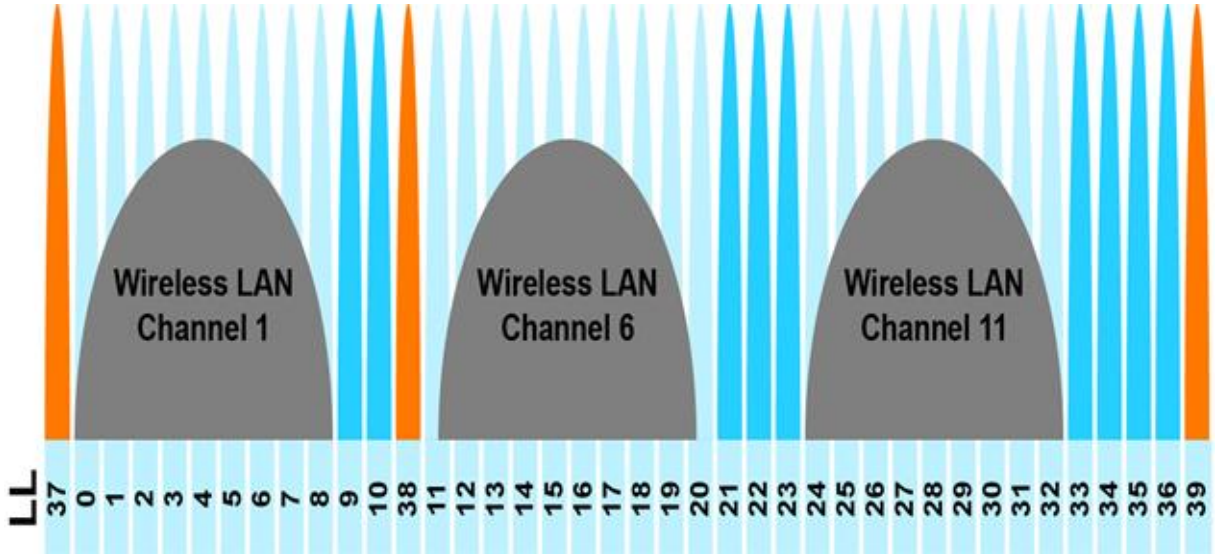
Market demands:
Low power (coin-cell operation)
Longer Range
Decent speed
Faster discovery

Bluetooth 4.0 Highlights

- BLE = Bluetooth Low Energy
  - Improved discovery
  - Key component: Beacons
    - Tags send out advertising beacons
    - Only 3 advertising channels
    - Devices scan all advertising channels for beacons
  - Challenge
    - Discovery happens in the 2.4Ghz band, competing with Wi-Fi

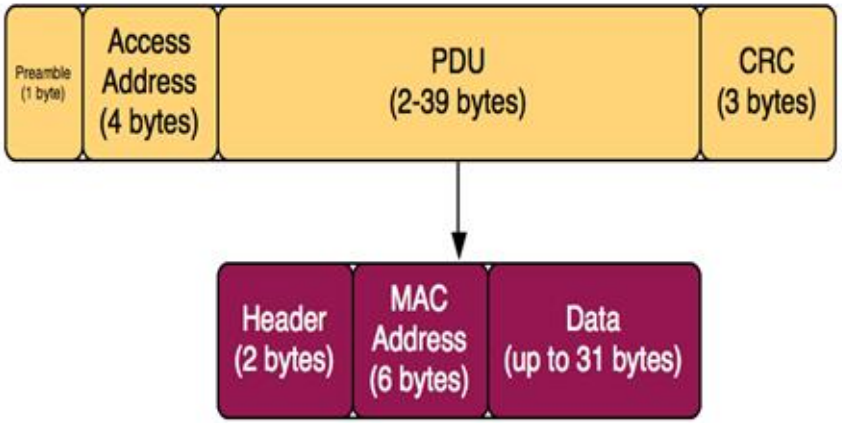
BLE: Channel Use and Coexistence with Wi-Fi

- Separate advertising and connected channels
  - Key: Three disjoint advertising channels (37, 38, 39)
  - Positioned between Wi-Fi channels (1, 6, 11)



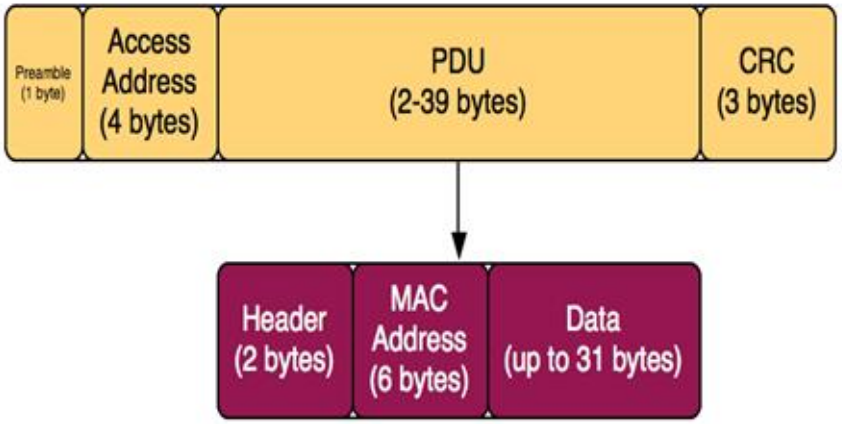
BLE: Advertising

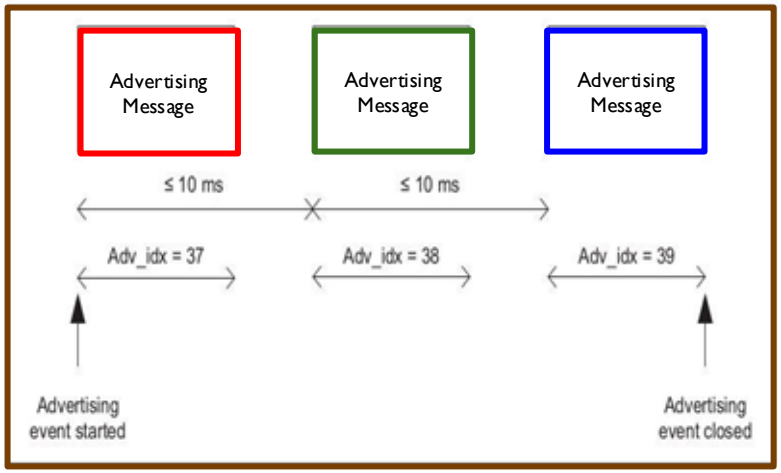
- Advertising Messages
  - Header + MAC Address + up to 31 Bytes of data
    - ~200 - 400 usec per packet
  - Two types: Non-scannable, Scannable



BLE: Advertising

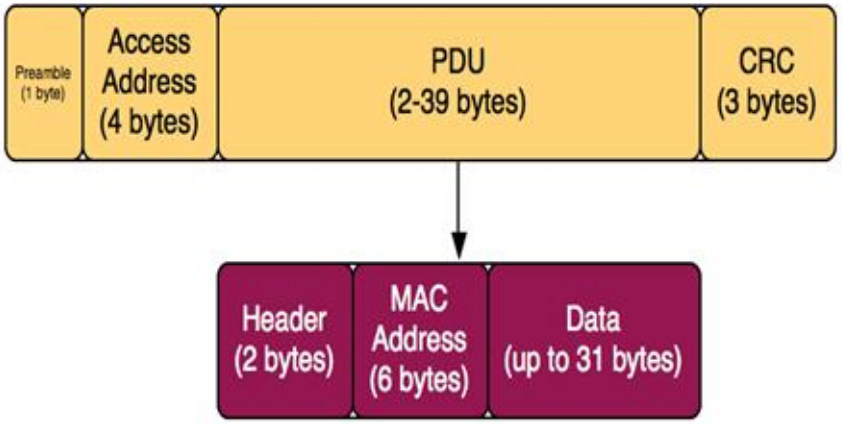
- Advertising Messages
  - Header + MAC Address + up to 31 Bytes of data
    - ~200 - 400 usec per packet
  - Two types: Non-scannable, Scannable
- Advertising Event
  - One advertising message sent out on each advertising channel (37, 38, 39)

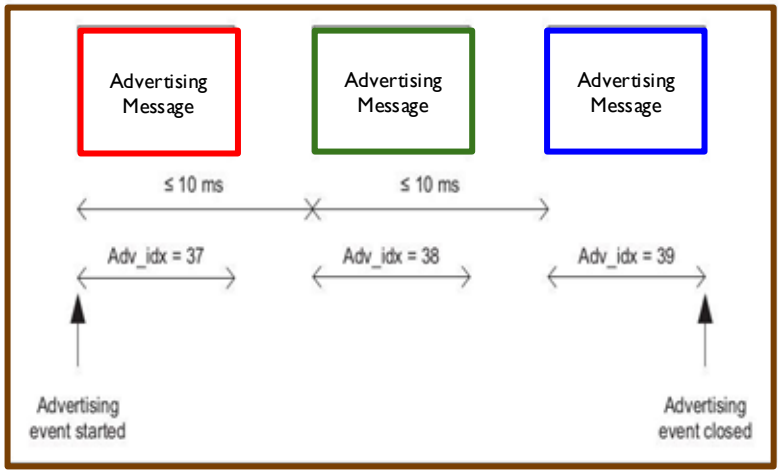




BLE: Advertising

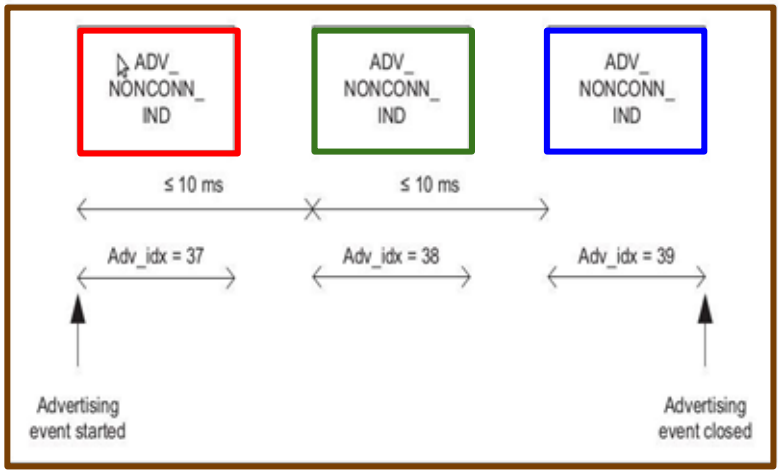
- Advertising Messages
  - Header + MAC Address + up to 31 Bytes of data
    - ~200 - 400 usec per packet
  - Two types: Non-scannable, Scannable
- Advertising Event
  - One advertising message sent out on each advertising channel (37, 38, 39)
- Advertising Interval
  - One advertising event per advertising interval
  - e.g., every 1 sec or 100 msec

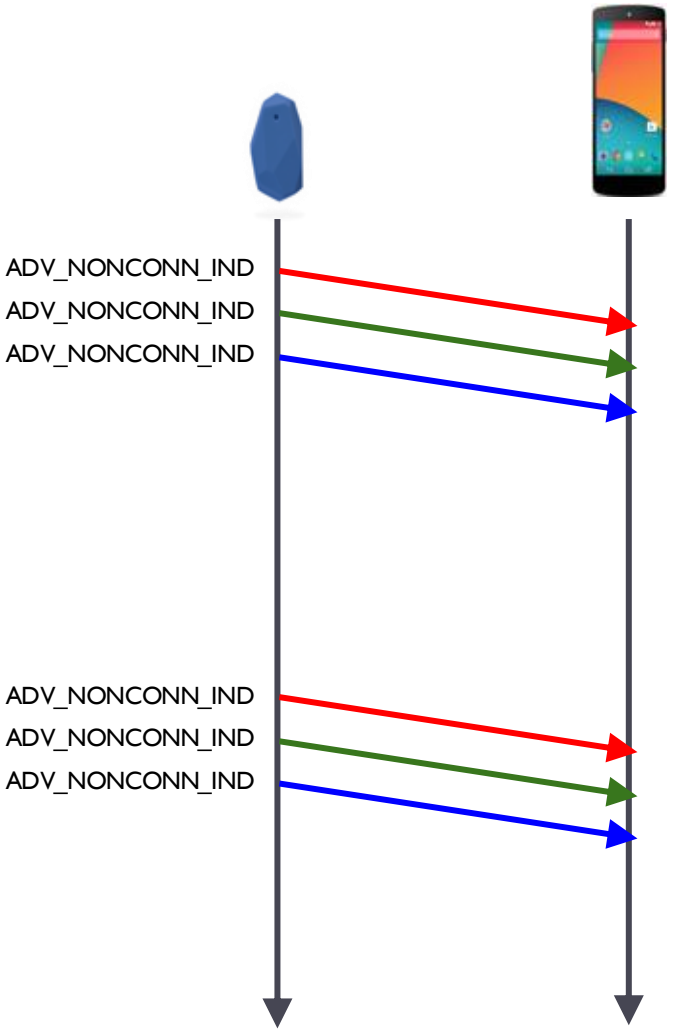




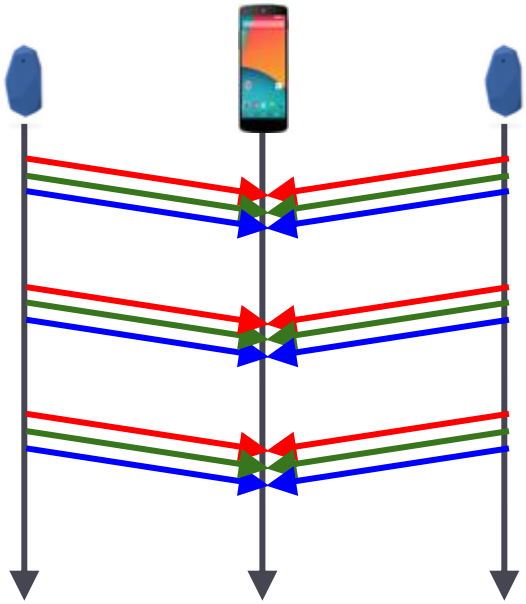
BLE: Tags Types - Non-Scannable

- Tags
  - Send ADV_NONCONN_IND messages
  - Typically sent back-to-back
- Scanners
  - Listen, but do not respond



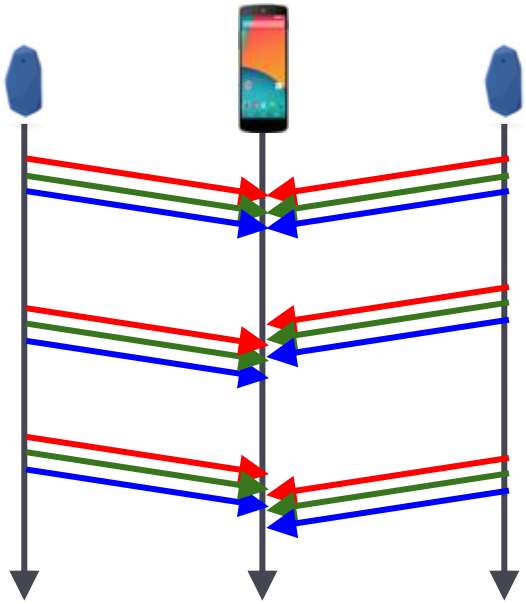


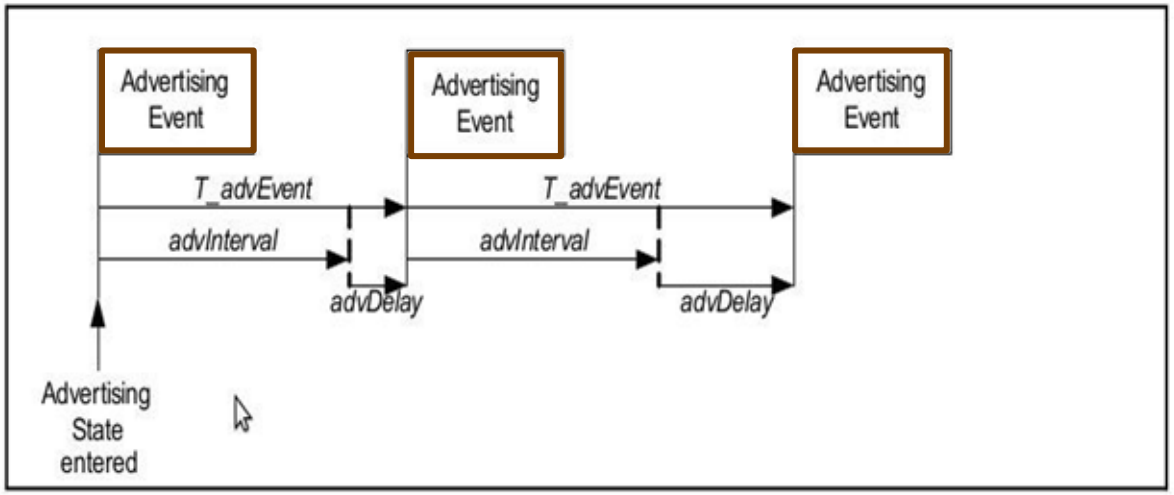
BLE: Advertising and Collisions



- If tags get synchronized, all advertising messages will collide

BLE: Advertising and Collisions

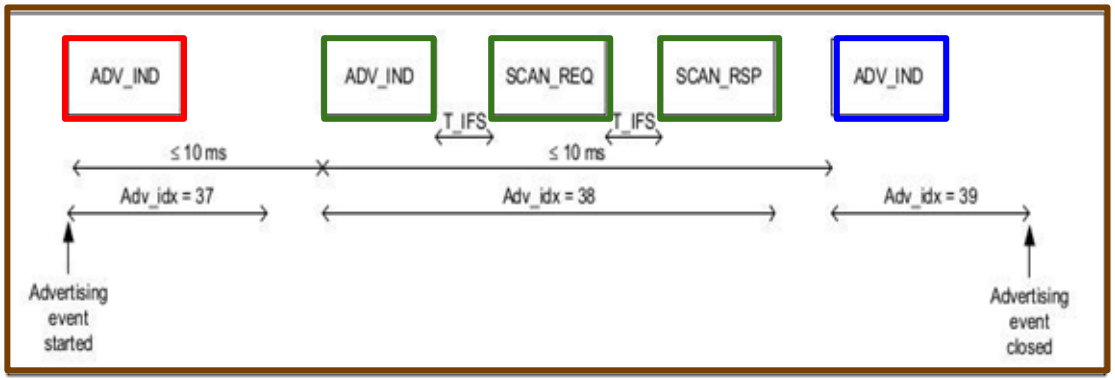


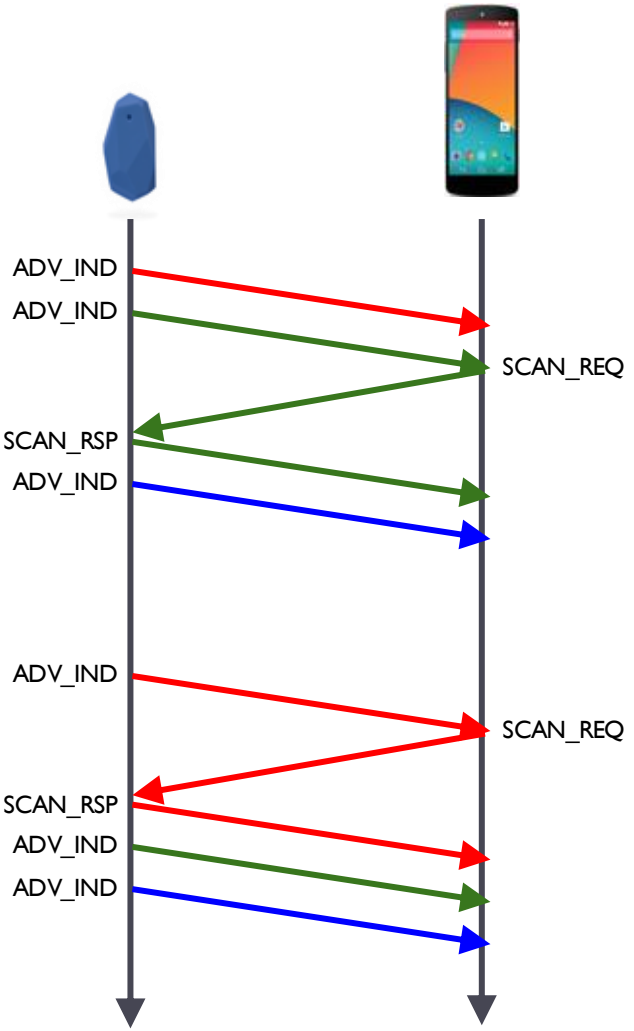


- Collision avoidance
  - Jitter advertising times
  - advDelay is added on to the end of each advertising event
  - advDelay = rand [0,10ms]

BLE: Tags Types - Scannable

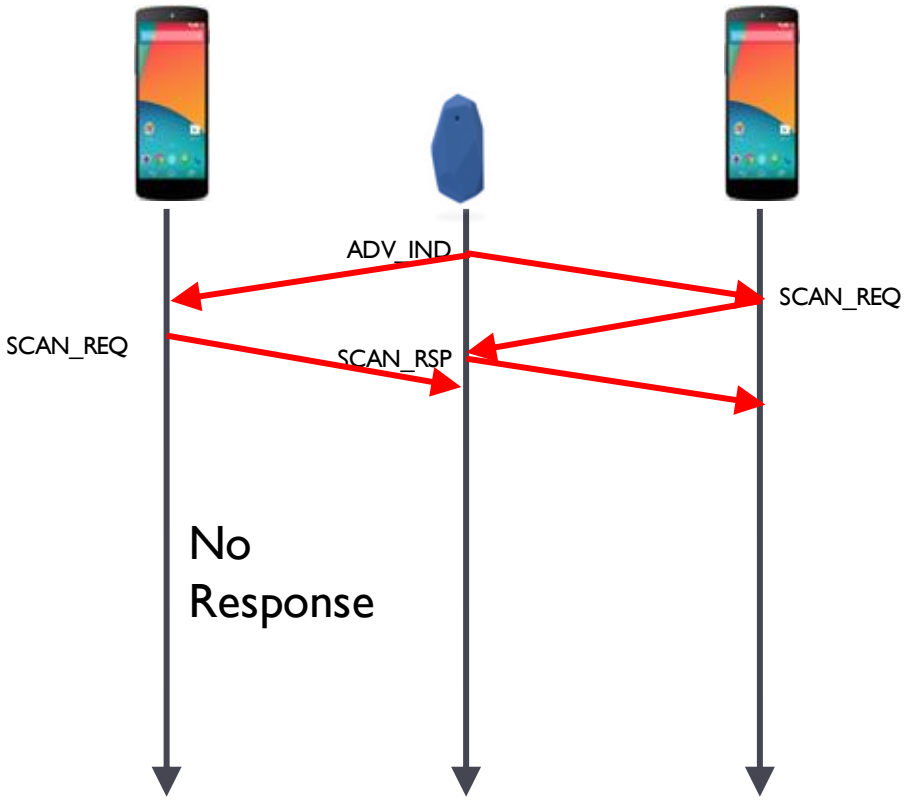
- Tags
  - Send ADV_IND messages
  - Tags wait ~150 usec for a request after beacon
- Scanners
  - Respond with SCAN_REQ message
- Tags
  - Respond with SCAN_RSP message
  - Up to 31 Bytes of extra data





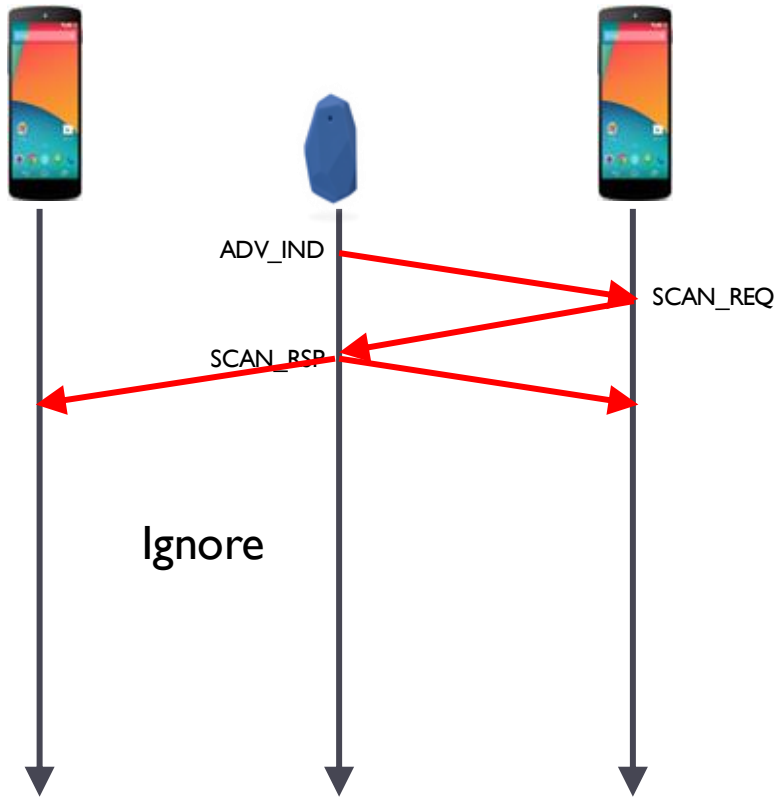
BLE: Scannable Tags

- One SCAN_RSP per channel per advertising event



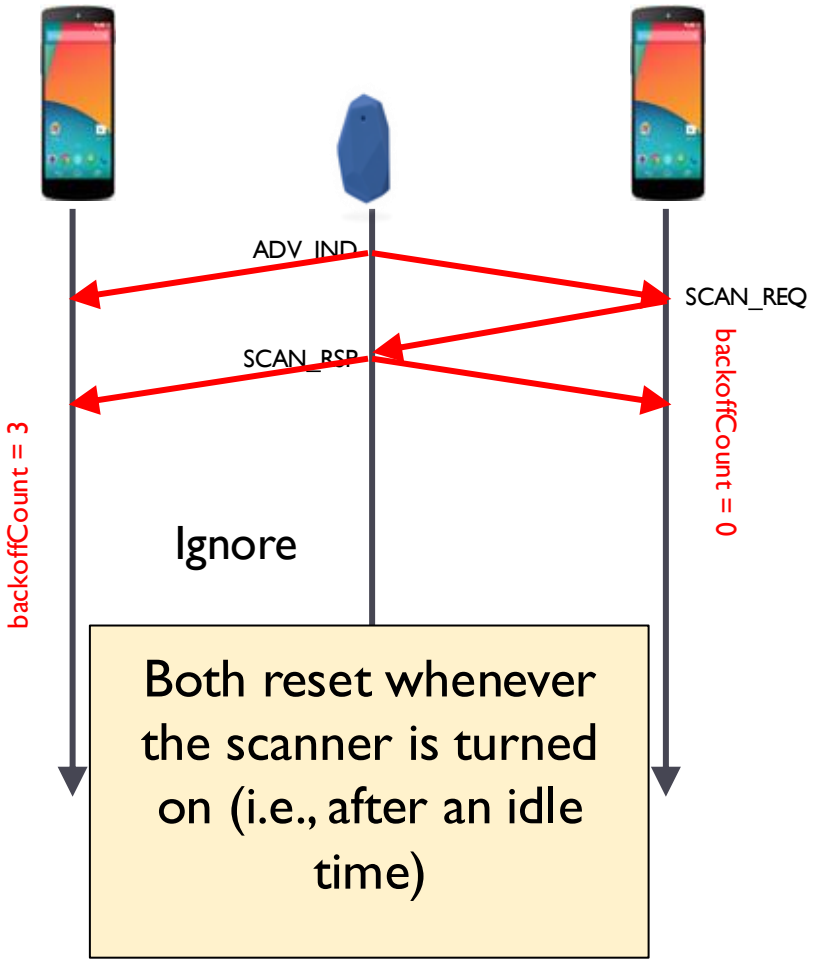
BLE: Scannable Tags

- ONLY accept SCAN_RSP if
  - SCAN_REQ was sent to that tag on that channel during that advertising event
- Some collision tolerance
  - Any requesting scanner can receive a SCAN_RSP as long as one SCAN_REQ is received and the tag responds
  - BUT, no SCAN_RSP if all SCAN_REQs collide



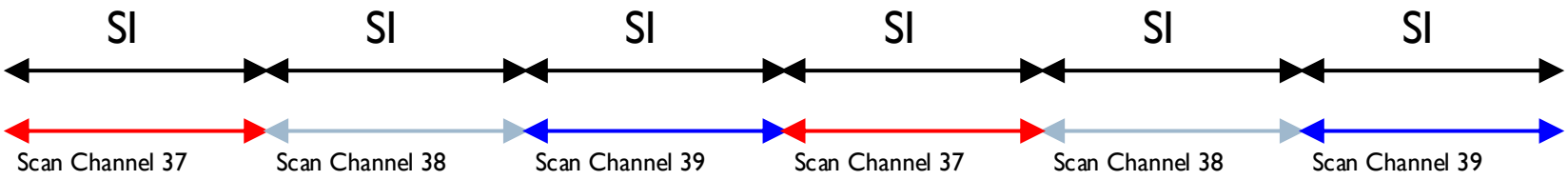
BLE: SCAN_REQ Collision Avoidance

- Scanner backoff
  - On starting scan
    - upperLimit = 1
    - backoffCount = 1
  - Only send SCAN_REQ if backoffCount == 0
    - Decrement backoffCount on receipt of ADV message
    - Failed SCAN_RSP
      - Double upperLimit (max 256)
    - Success:
      - Halve upperLimit (min 1)
  - After each SCAN_REQ
    - backoffCount = rand(1, upperLimit)



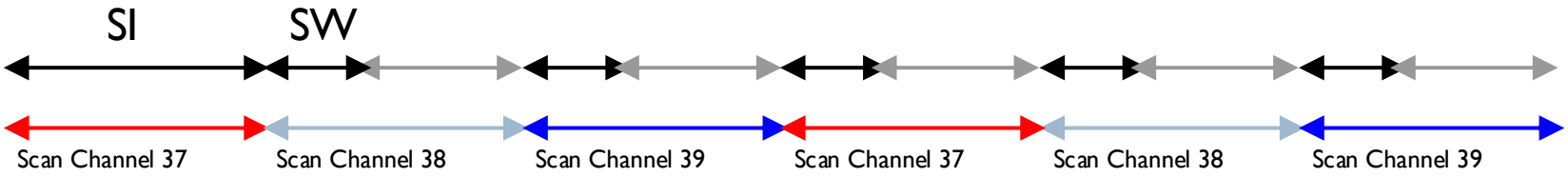
BLE Highlights: Low-level Scanning

- Scanners
  - Scan for beacons on sequential channels (37, 38, 39)
- Scan Interval (SI)
  - Time spent on a channel



BLE Highlights: Low-level Scanning

- Scanners
  - Scan for beacons on sequential channels (37, 38, 39)
- Scan Interval (SI)
  - Time spent on a channel
- Scan Window (SW)
  - Time spent scanning at beginning of Scan Interval
- Scan Time
  - If Scan Int == Scan Window ⇒ Always on



BLE Highlights: Application-level Scanning

- Scanners
- Application Scan Time
  - > Tag Advertising Interval
- Application Idle Time
- 100% scan time
  - Idle Time = 0

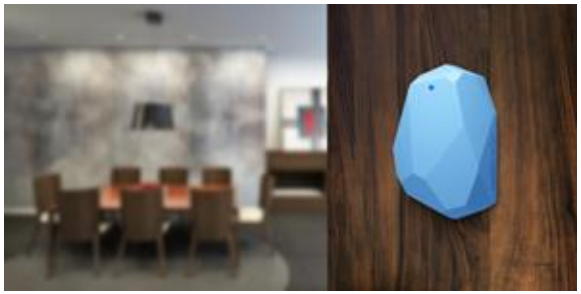
Application Scan Time

BLE Highlights: MAC Behavior

- No Carrier Sense
  - Tag does not listen for a clear channel before sending any message
- Minimal Contention Avoidance
  - Jitter length of advertising interval + rand [0, 10 ms]
  - Backoff for sending SCAN_REQ

BLE in the Real World

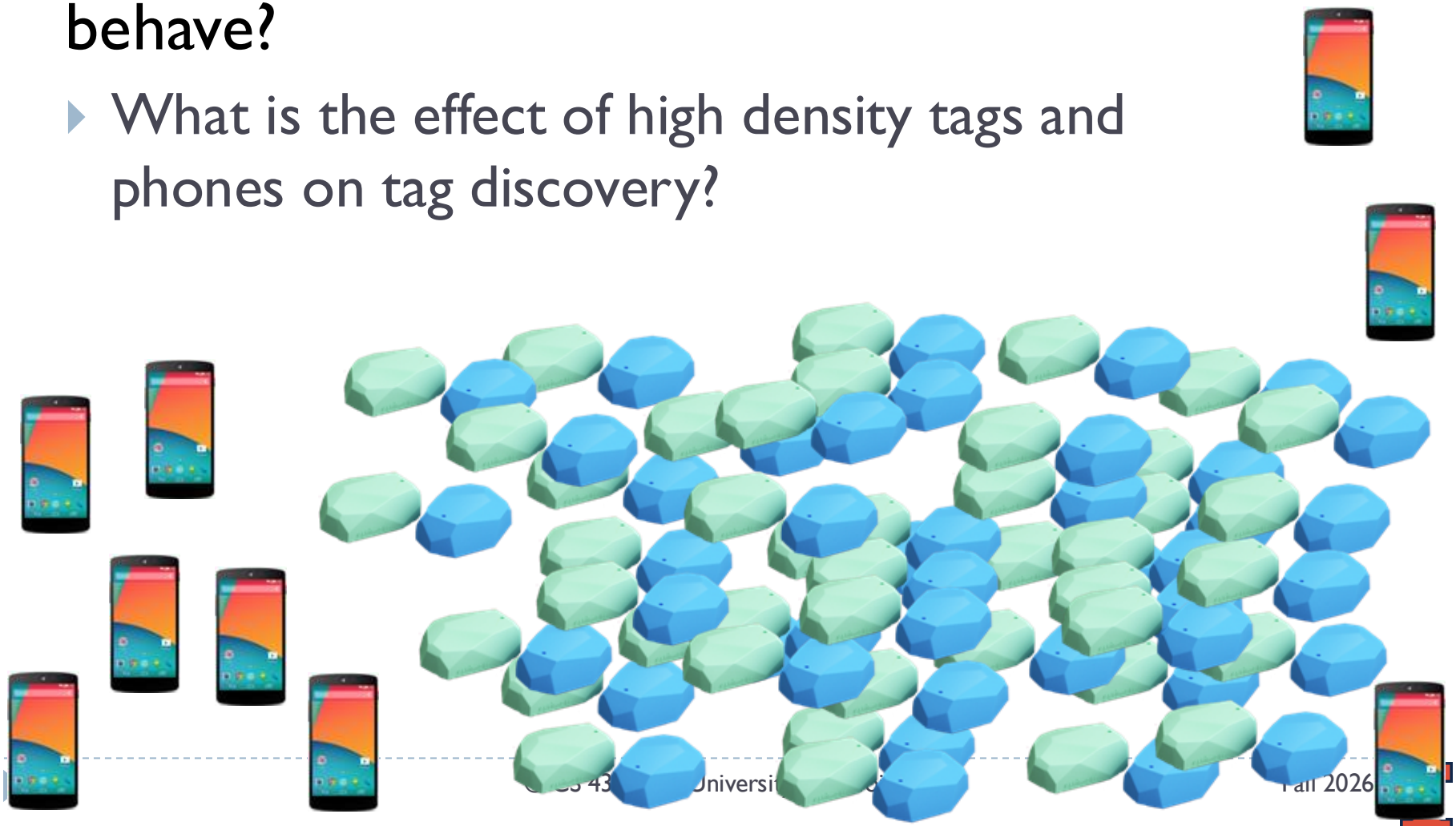
- BLE beacons (or tags)
  - Location-specific information
  - Deployed in public places
    - Stores, airports, museums
  - Accessed via devices with BLE



- Performance questions
  - How long does it take to detect a nearby tag?
  - Can we detect a tag within 5 sec with 95% success?

BLE in the Real World - Density

- As deployments increase, how will the tags behave?
  - What is the effect of high density tags and phones on tag discovery?

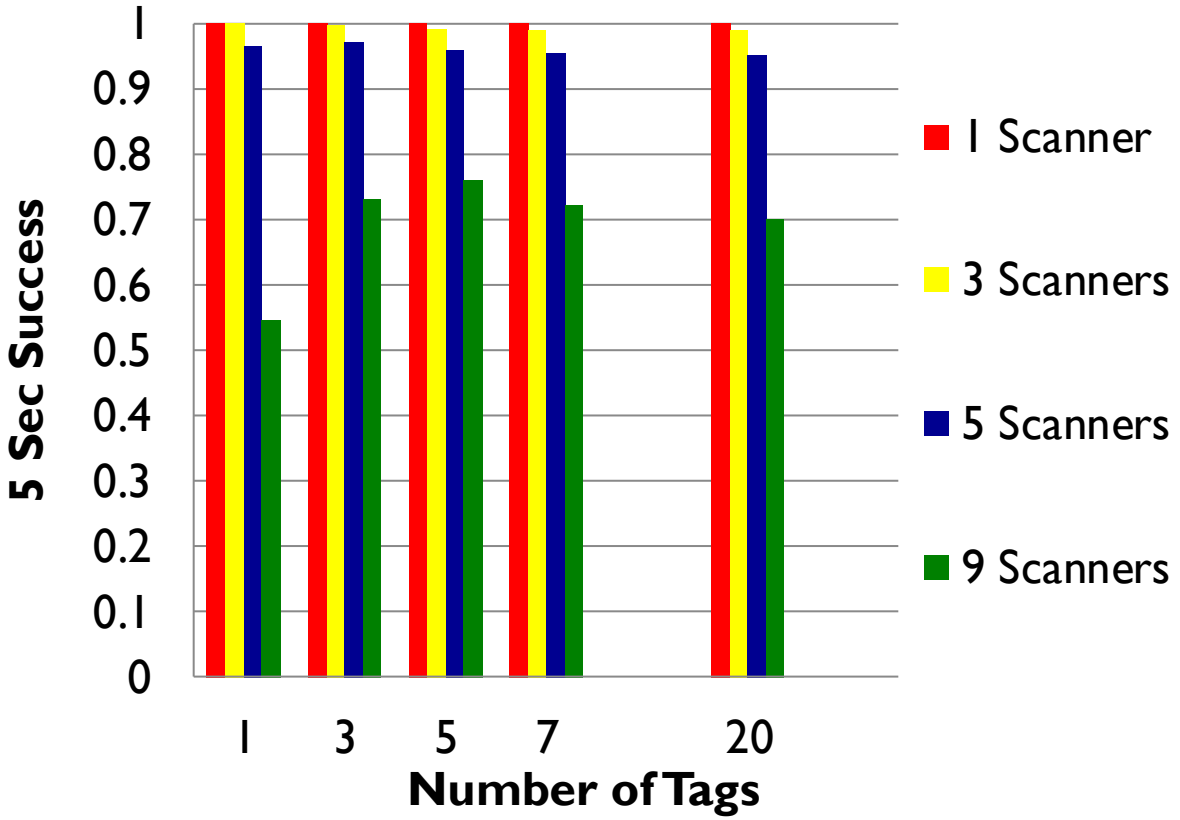


Evaluating Tag Behavior

- Metric
  - 5 Sec Success
    - Could the tag be found in 5 sec?
    - Checked every 1 sec over the whole run
- Environmental Impact
  - At what density of tags or phones does the system break down?

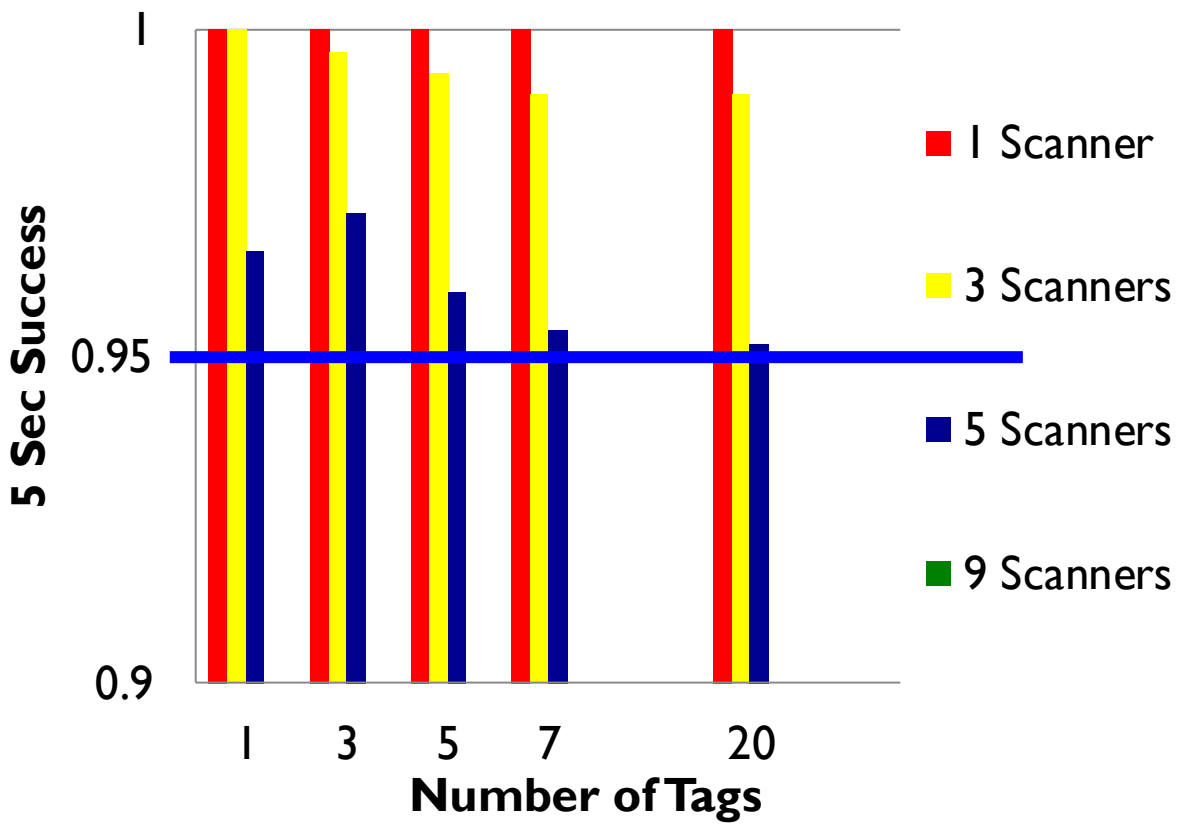
Evaluation: BLE Scan/Response

- 5 second success
  - Multiple chances to find the tag
  - Success decreases significantly as more phones are added
  - Number of phones is more important than number of tags



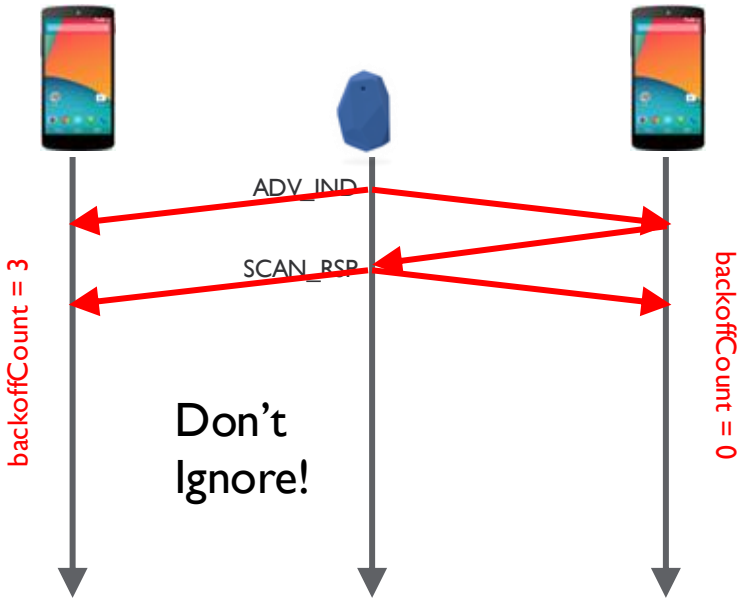
Evaluation: BLE Scan/Response

- 5 second success
  - Below target threshold for more than 5 phones



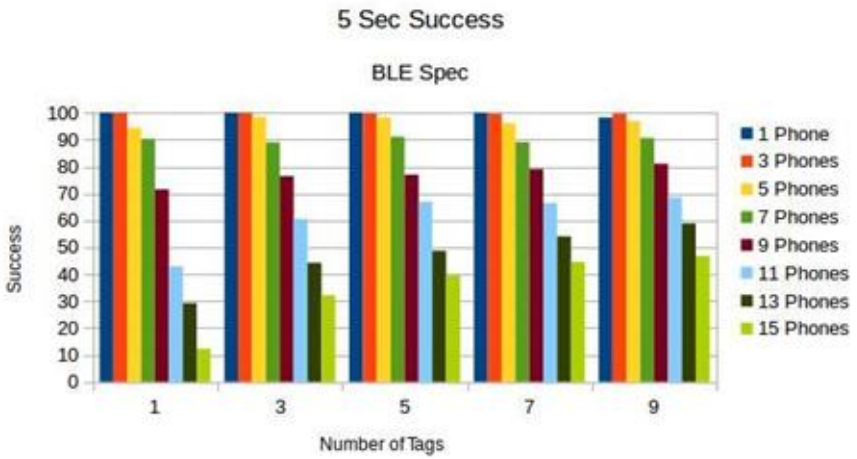
SCAN_REQ: Opportunistic Listening

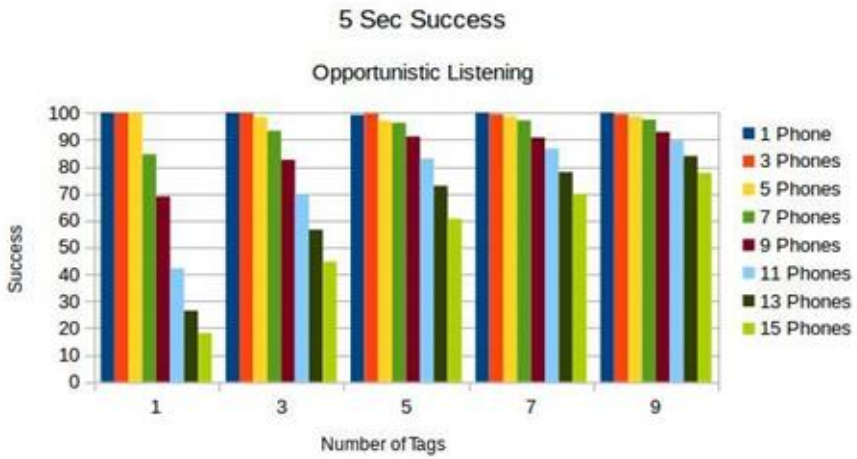
- Accept a SCAN_RSP on a channel if a SCAN_REQ would have been sent, but the backoff procedure indicated not to send it
  - Any requesting or backing off scanner can receive a SCAN_RSP as long as one SCAN_REQ is received and the tag responds
  - Still, No SCAN_RSP if all SCAN_REQs collide



Opportunistic Listening: Simulation Comparison

- Significant increase in success rate as number of phones increases
- Cannot prevent SCAN_REQ collisions



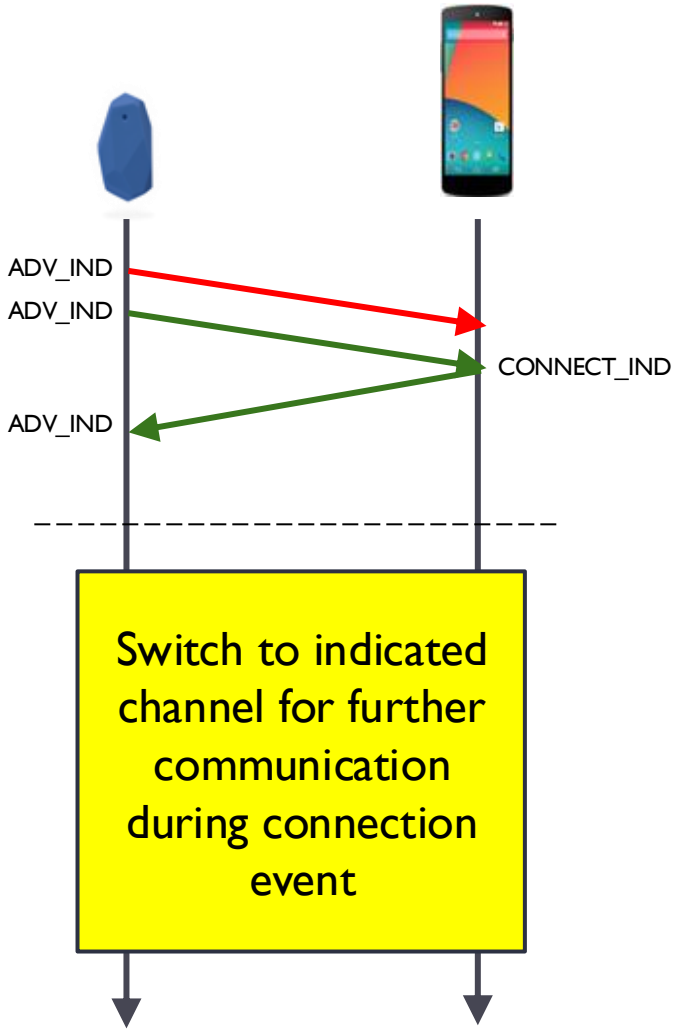


BLE Connected Mode: Highlights

- Connection Intervals
  - Devices communicate in short, synchronized bursts called Connection Events
  - Time between the start of two consecutive events
  - Radio is not transmitting 100% of the time
  - Central transmits first, and the Peripheral replies
- Peripheral Latency
  - Defines how many consecutive Connection Events the Peripheral is allowed to completely skip if it has no new data to report.
  - The moment the Peripheral has a data update (like a button press), it wakes up instantly at the next interval to transmit
- Connection Supervision Timeout
  - Safety timer to end connection
  - Expires if a device misses too many consecutive Connection Events without any communication
  - Both devices terminate the connection locally and return to their baseline scanning or advertising states

BLE Connected Mode: Setup

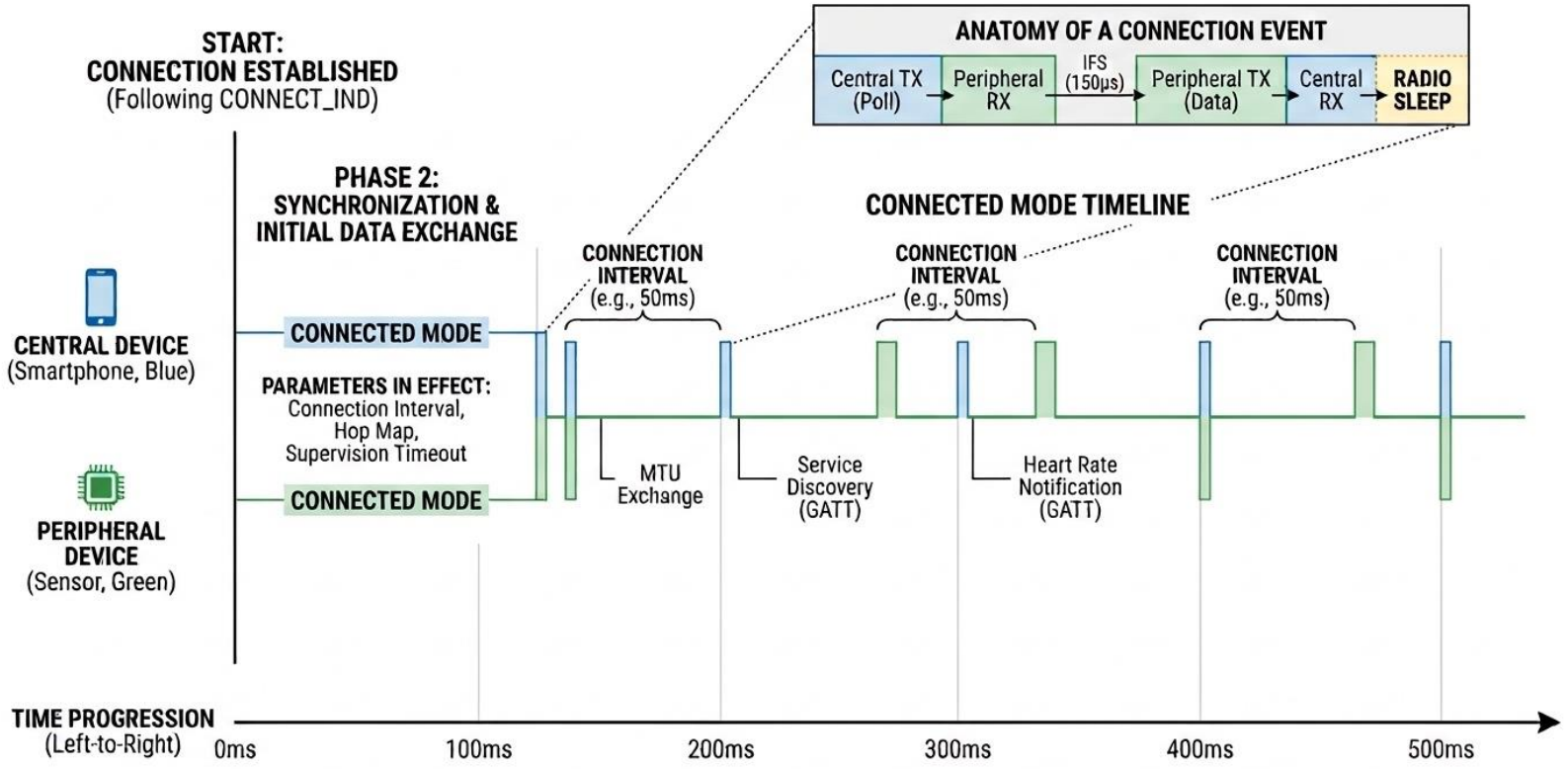
- Scanner (initiator) answers an ADV_IND with CONNECT_IND
- Carries access address, hop increment, channel map, interval, latency, timeout



BLE Connected Mode: Connection Events

- Connection event
  - Periodic wake-up
  - Anchor point
    - Central transmits first
    - Peripheral answers 150 µs later (T_IFS)
  - More packets may follow in the same event
    - Empty packets keep the link alive
  - Each event hops to the next data channel

BLE Connected Mode: Connection Timeline



BLE Connection Parameters

- Connection interval
  - 7.5 ms to 4 s, in steps of 1.25 ms
- Peripheral latency
  - May skip up to 499 events if it has no data
- Supervision timeout
  - 100 ms to 32 s without a valid packet ends the link
- Trade-off:
  - Short interval = low latency, high energy
  - Long interval + latency = low energy, high latency
  - Like a Wi-Fi listen interval, but wake times are exact
    - No beacon or TIM needed

BLE Connected Mode: Latency and Power

- Duty cycle = event time / connection interval
  - Empty-packet event at 1 Mbps
    - 80 µs + 150 µs + 80 µs = 310 µs of radio time
    - Interval 1 s: ~0.03% duty cycle
    - interval 30 ms: ~1%
    - interval 7.5 ms: ~4%
- Latency: data waits for the next event
  - Worst case ~ interval x (1 + peripheral latency)
  - Peripheral latency saves power but delays peripheral-to-central data

BLE Connected Mode: Latency and Power

- Tuning examples
  - Keyboard/mouse
    - Short interval (7.5-15 ms), low latency
  - Temperature sensor
    - Interval of seconds, latency > 0
- Updates and extensions
  - Central may change parameters with a connection update
    - 5.3 subrating switches quickly between low-latency and low-power timing
  - Longer events (data length up to 251 B) and 2M PHY shrink on-air time per byte

BLE: GATT and Pairing

- GATT/ATT: how apps use a connection
  - Peripheral is a GATT server
    - Services contain characteristics (values)
  - Central can read, write, or subscribe to notifications/indications
  - Default ATT MTU is 23 B (20 B payload)
    - Data Length Extension (4.2) allows 251 B link-layer payloads
- Pairing and security
  - LE Secure Connections (4.2)
    - ECDH P-256 key exchange
  - Association models
    - Just Works (no MITM protection), Passkey, Numeric Comparison, Out-of-Band
  - Resolvable private addresses
    - Rotating address stops long-term tracking
- Example
  - Matter devices advertise over BLE, then join Wi-Fi or Thread

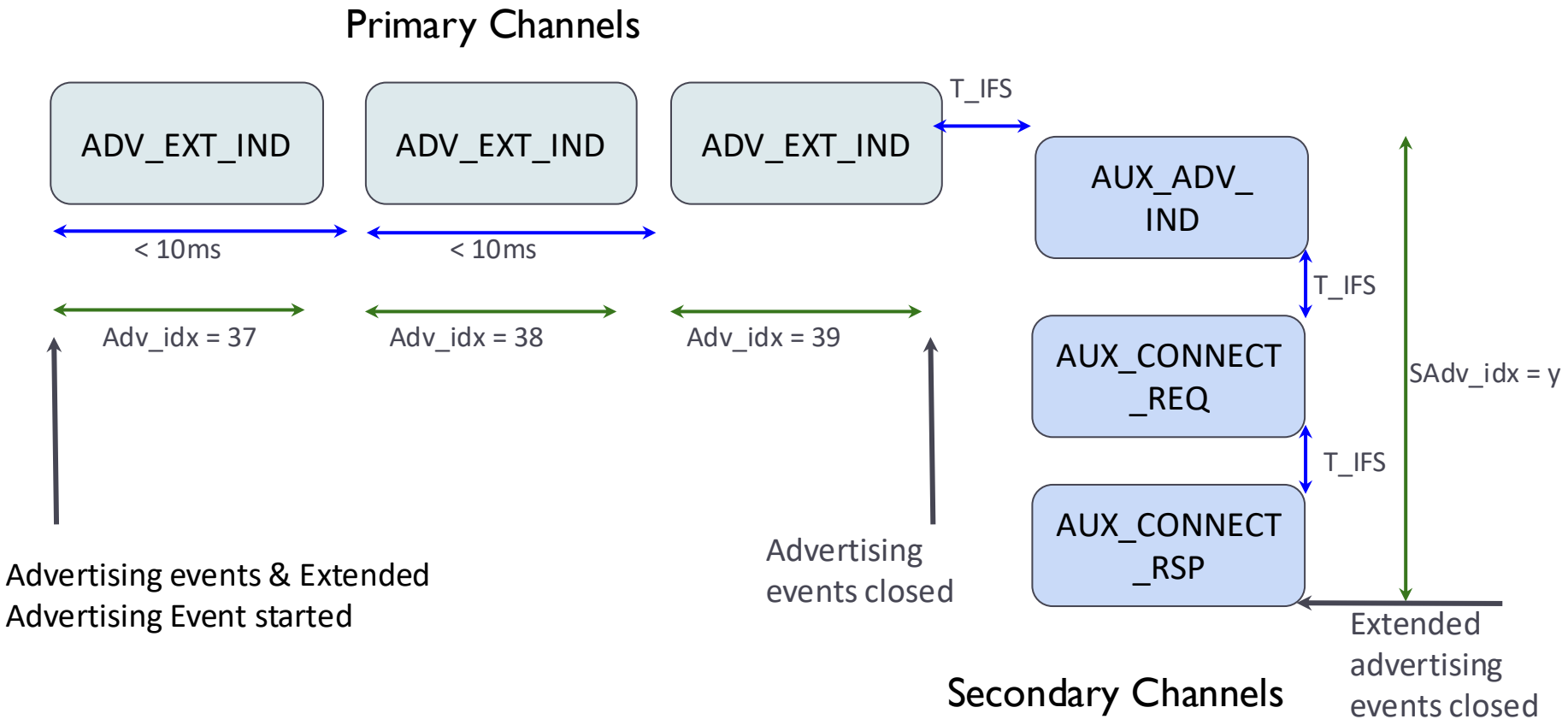
Bluetooth Progression

4.0 is too slow
Low range (especially indoors)
Power issues
Issues relating to multiple radios on the same device

Bluetooth 5.0: Extended Advertising

- Broadcast data congests three primary advertising channels
  - Send a small, lightweight packet ADV_EXT_IND (Extended Advertising Indicator).
  - No user payload data
  - Pointer (channel index and specific time offset) directing listeners to a secondary data channel.
- Secondary Data Channels (0 to 36)
  - Actual advertising data is offloaded to the remaining 37 secondary data channels
  - Typically used only for connected data traffic
  - Broadcaster transmits an auxiliary packet called AUX_ADV_IND containing the real payload data.

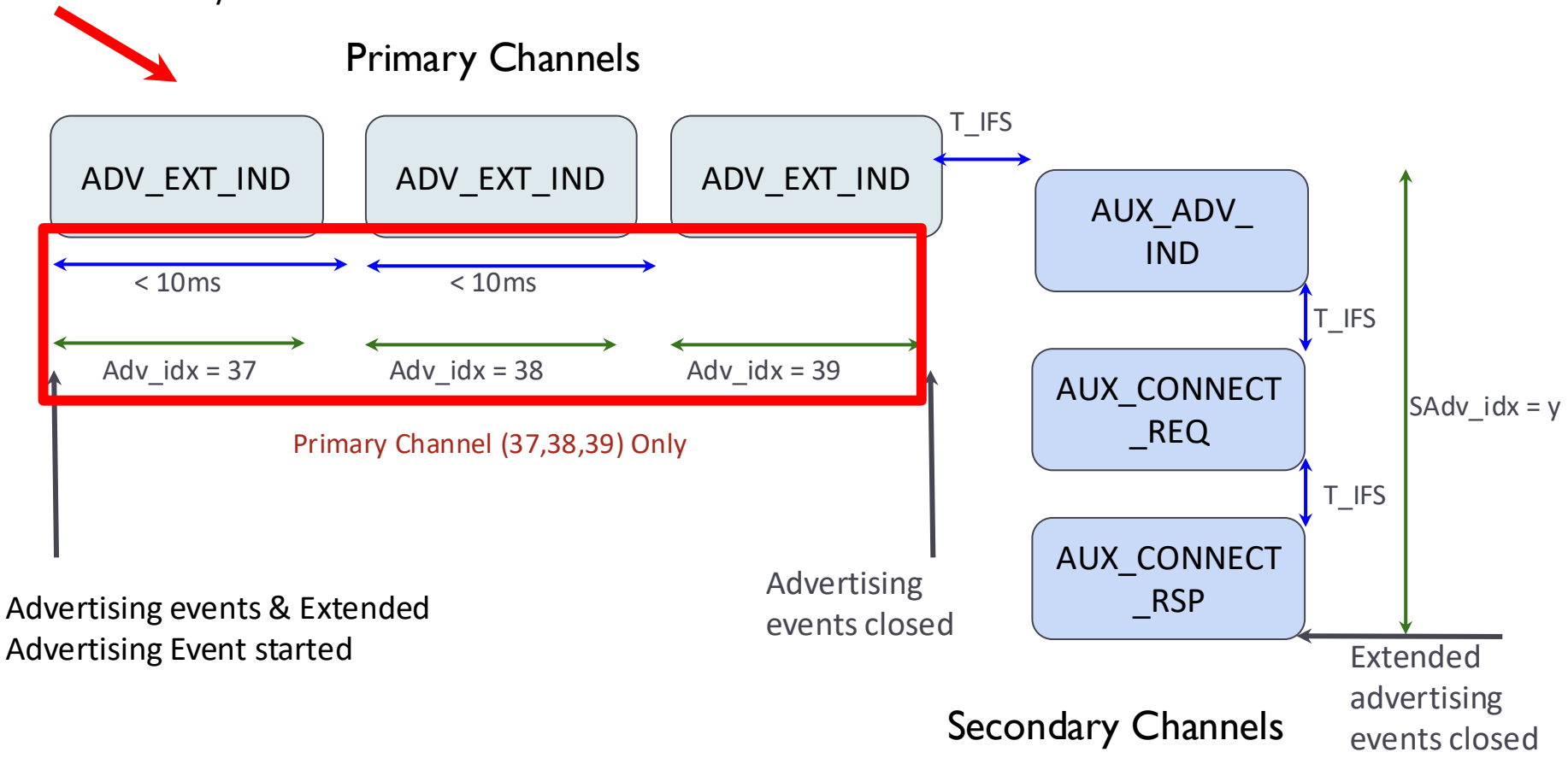
Bluetooth 5.0: Extended Advertising



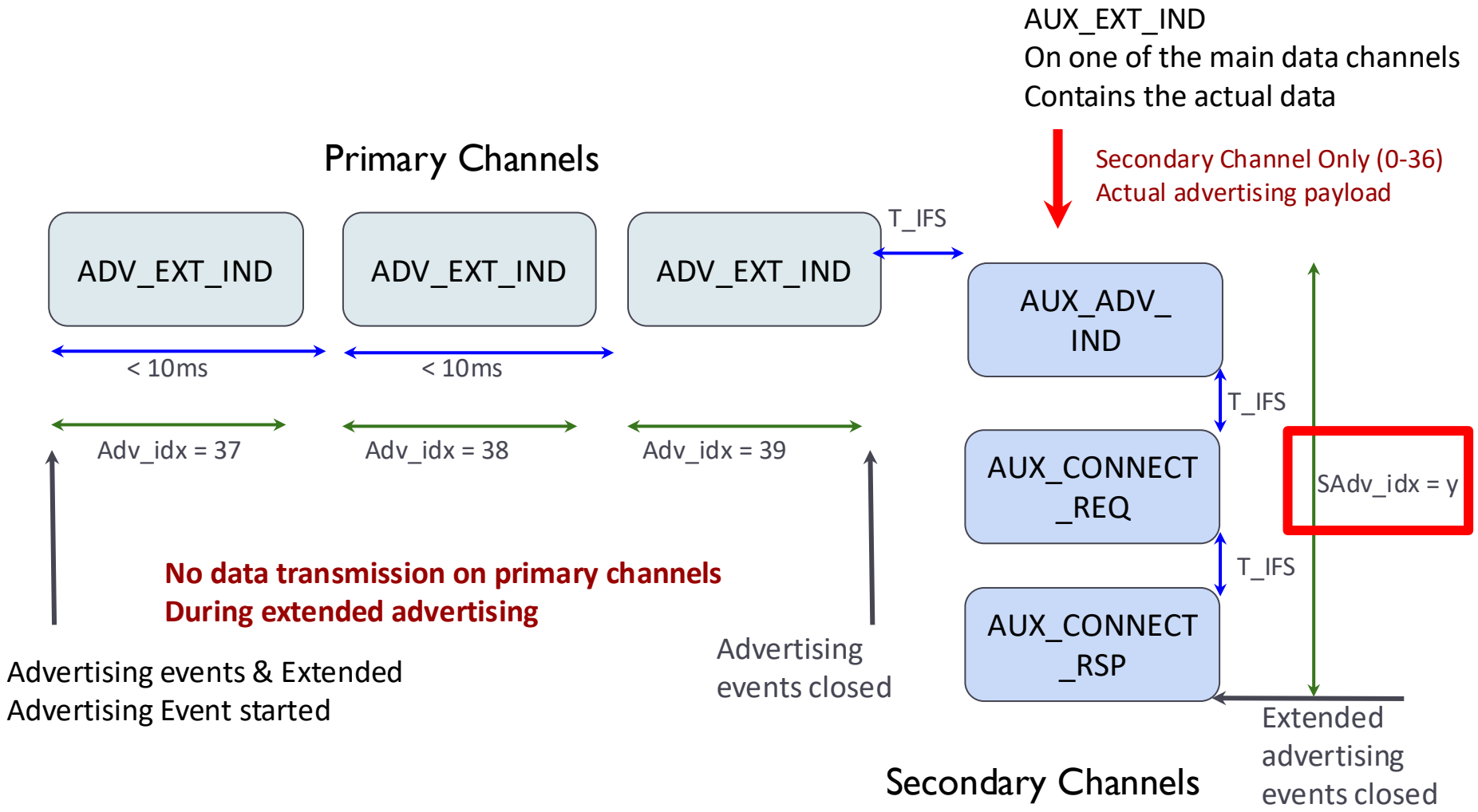
Bluetooth 5.0: Extended Advertising

ADV_EXT_IND
Contains no data, just pointer to secondary channel

1. Channel indices
2. Timing info
3. PHY layer setting



Bluetooth 5.0: Extended Advertising



Bluetooth 5.0: Payload Increase

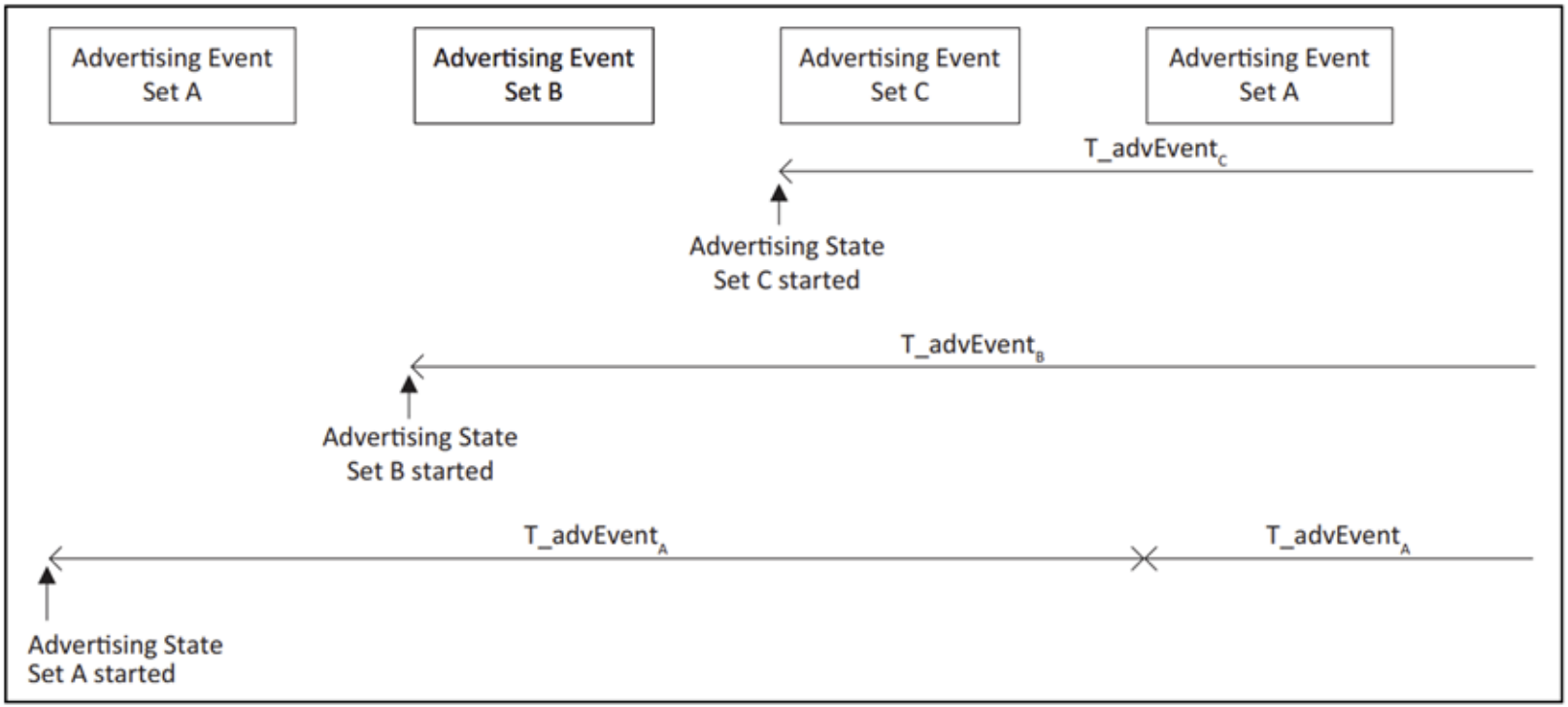
- Bluetooth 4.0
  - Tiny payload in advertising messages - 31 bytes
- Bluetooth 5.0
  - Use Extended Advertising
  - Not on primary channel
  - Enables full 255-byte payload
  - Avoids high power connected communication

Bluetooth 5.0: Multiple Advertising Sets

- Bluetooth 4.0
  - One advertising message is repeatedly sent
- Bluetooth 5.0
  - Advertising Handles
    - A single device can behave like multiple virtual beacons
  - Enables multiple distinct advertising packets, each with its own independent settings
    - Supports simultaneous use of devices with varying energy and delay needs

Bluetooth 5.0: Multiple Advertising Sets

- Multiple, independent advertising sets simultaneously
- Enhances flexibility and efficiency of advertising



Bluetooth 5.0: Coded PHY

- Bluetooth 4.0
  - Limited range
- Bluetooth 5.0
  - Extend range ~4x (S=8 coding) at the same Tx power
  - No increase in transmission power.
  - Add Forward Error Correction (FEC)
    - Works with lower signal strength
    - Reduces throughput

Bluetooth 5.0: Coded PHY

Parameter	LE 1M	LE Coded S2	LE Coded S8	LE 2M
Symbol Rate	1 Msps	1 Msps	1 Msps	2 Msps
Data Rate	1 Mbps	500 kbps	125 kbps	2 Mbps
Error Correction	None	FEC	FEC	None
Range Multiplier	1	~ 2	~ 4	~ 0.8

- Symbols per sec
  - S2: 2 symbols = 1 bit
  - S8: 8 symbols = 1 bit

Bluetooth 5.0: Coded PHY

Parameter	LE 1M	LE Coded S2	LE Coded S8	LE 2M
Symbol Rate	1 Msps	1 Msps	1 Msps	2 Msps
Data Rate	1 Mbps	500 kbps	125 kbps	2 Mbps
Error Correction	None	FEC	FEC	None
Range Multiplier	1	~ 2	~ 4	~ 0.8

Baseline for BLE 4.0

Bluetooth 5.0: Coded PHY

Parameter	LE 1M	LE Coded S2	LE Coded S8	LE 2M
Symbol Rate	1 Msps	1 Msps	1 Msps	2 Msps
Data Rate	1 Mbps	500 kbps	125 kbps	2 Mbps
Error Correction	None	FEC	FEC	None
Range Multiplier	1	~ 2	~ 4	~ 0.8

2Mbps max for BLE 5.0
(High data rate, reduced range)

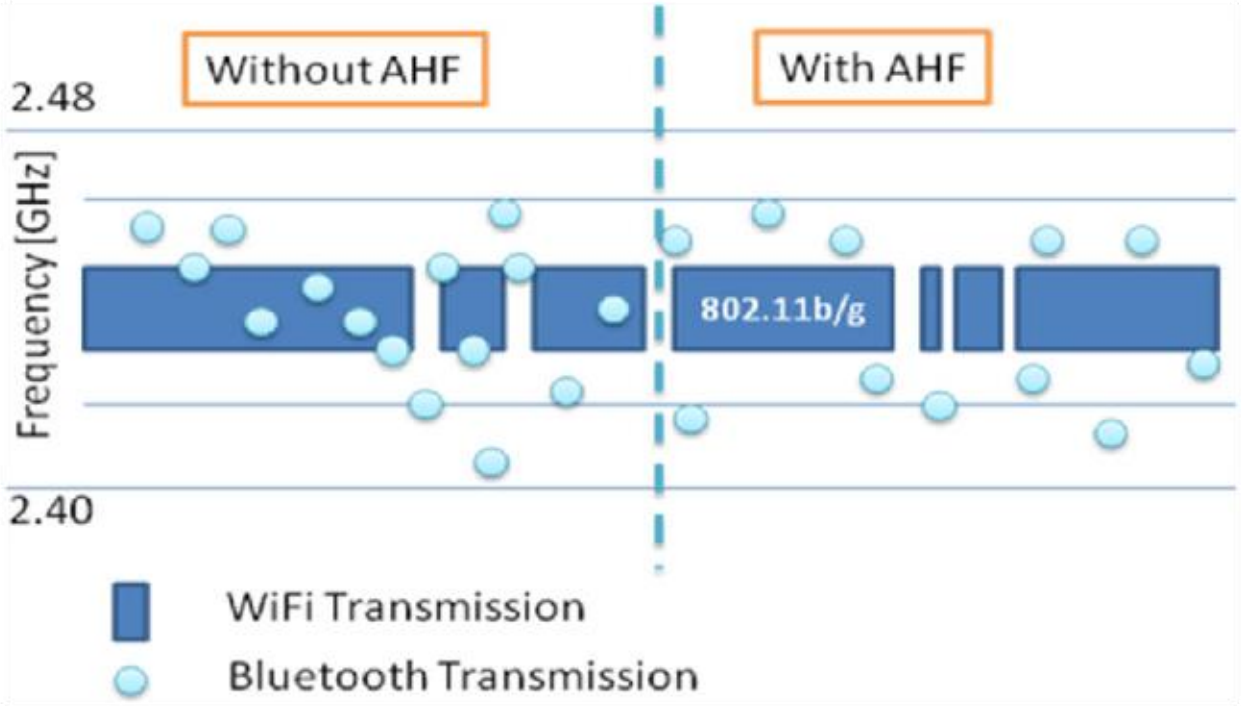
Bluetooth 5.0: Coded PHY

Parameter	LE 1M	LE Coded S2	LE Coded S8	LE 2M
Symbol Rate	1 Msps	1 Msps	1 Msps	2 Msps
Data Rate	1 Mbps	500 kbps	125 kbps	2 Mbps
Error Correction	None	FEC	FEC	None
Range Multiplier	1	~ 2	~ 4	~ 0.8

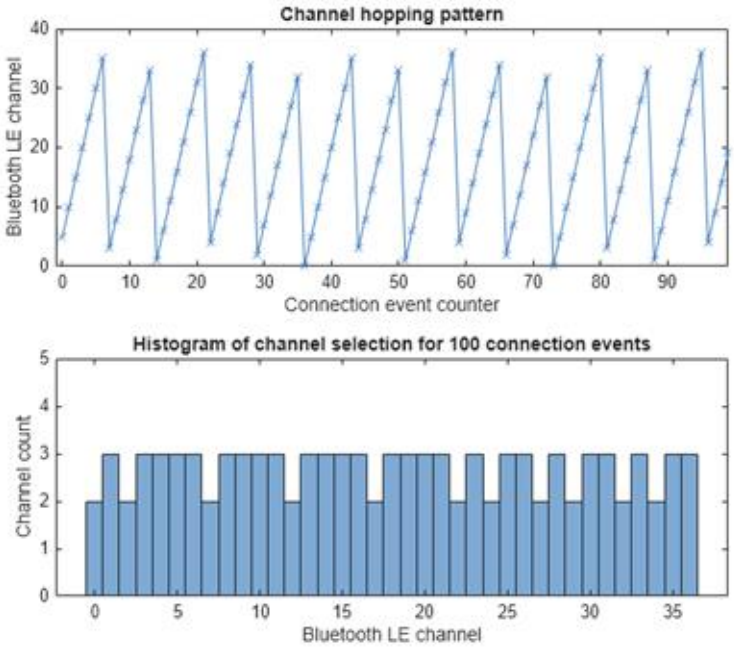
Quadruple distance for coded PHY (Long range application, reduced data rate)

Bluetooth 5.0: Channel Selection (CSA #1 vs. #2)

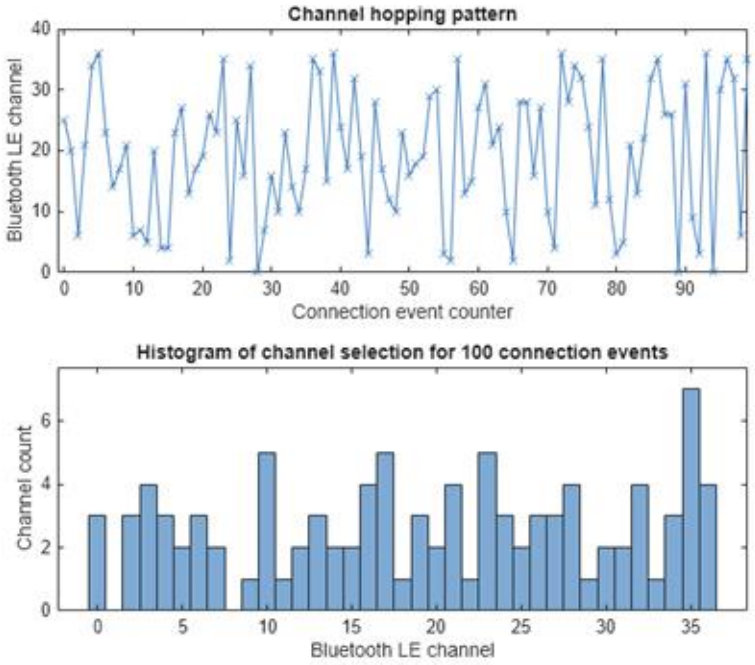
- Adaptive Frequency Hopping
  - Channels (0-36) set to used or unused, algorithmically determine sequence
  - Channel Selection Algorithm (CSA) #1: 12 distinct sets
  - CSA #2 allows for many distinct sequences, reduces collisions



Bluetooth 5.0: Channel Selection (CSA #1 vs. #2)



CSA 1 (no randomization)



CSA 2 (pseudo randomization)

Bluetooth 5.0: Adaptive Frequency Hopping

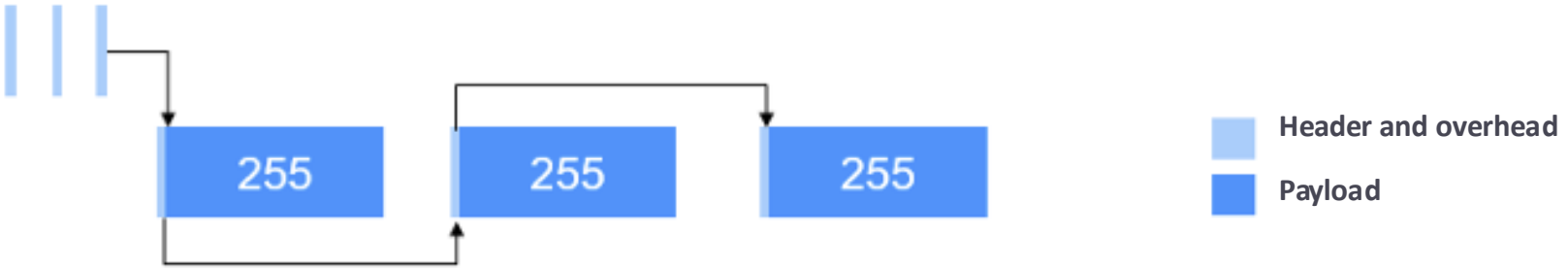
- How AFH works
  - Channel Map
    - Packet-error/RSSI measurements, plus what the host knows about its own Wi-Fi channel
    - Needs time to measure and update the map
    - A strong nearby Wi-Fi transmitter can block more than its nominal channel
  - Classic
    - Channels marked used/unused
    - Unused channels are replaced in the hop sequence
    - At least 20 channels must stay in use
  - BLE
    - Central keeps a data-channel map (0-36, at least 2 used)
    - Map updated during a connection

Bluetooth 5.0: Adaptive Frequency Hopping

- What it fixes
  - Connected data avoids busy Wi-Fi channels
  - Wi-Fi also sees less Bluetooth interference
- What it cannot fix
  - BLE advertising channels 37/38/39 are fixed and outside the map
  - Classic inquiry/page does not use AFH
  - Three 20 MHz networks (1/6/11) cover most of Classic's 79 channels, possibly fewer than the required 20 clean ones

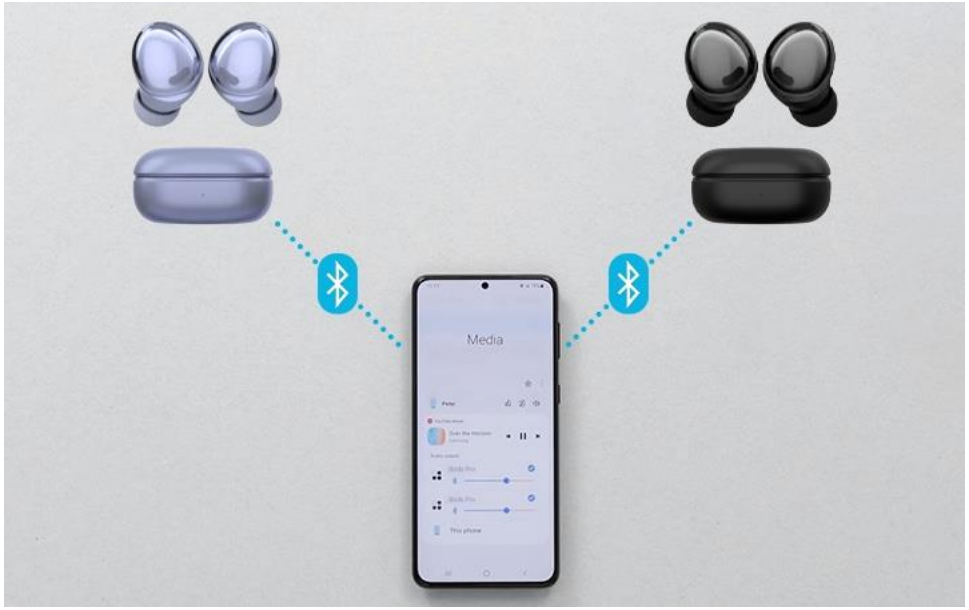
Bluetooth 5.0: Packet Chaining

- Controller can chain packets
  - AuxPtr header fields (references to Auxiliary Packets containing payload)
  - AuxPtr includes the channel number 0-36
  - Up to 1,650 bytes
- Improves efficiency, data transfer rate, power, etc.



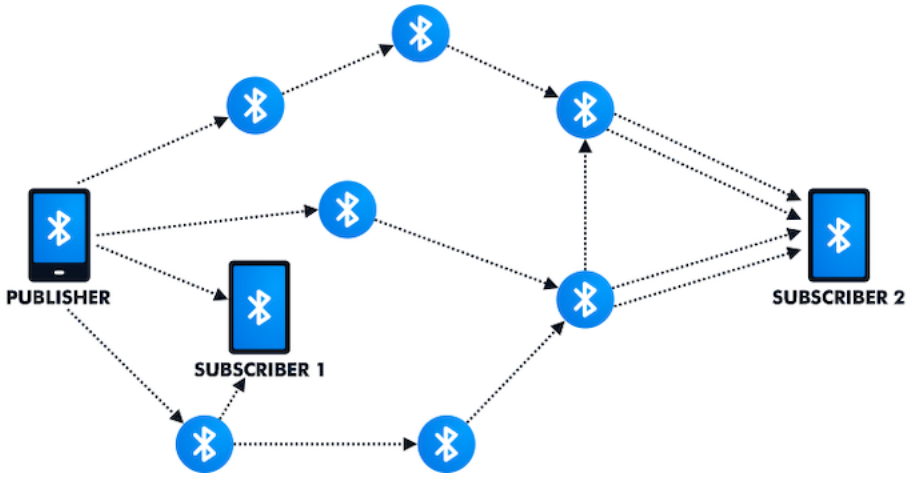
Dual Audio (Vendor Feature)

- Vendor feature (not in the 5.0 spec): one phone streams audio to two audio devices simultaneously



Bluetooth Mesh

- Mesh Networking
  - Devices can relay packets

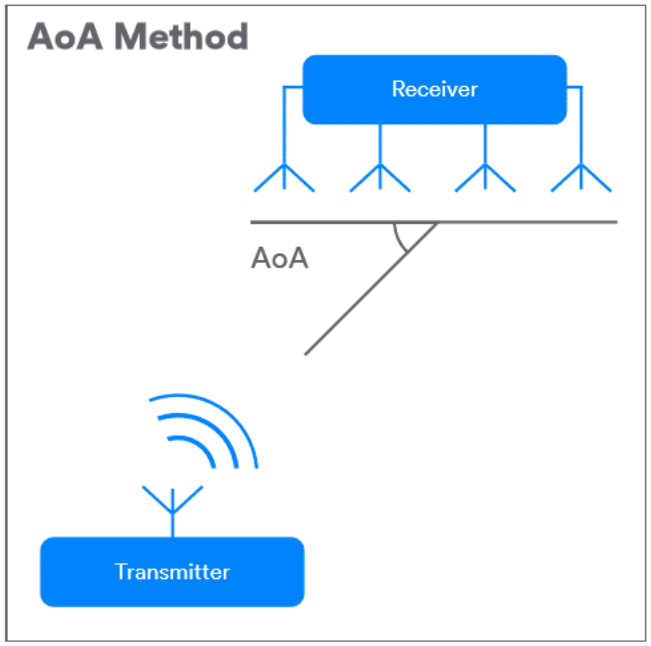


Bluetooth Mesh

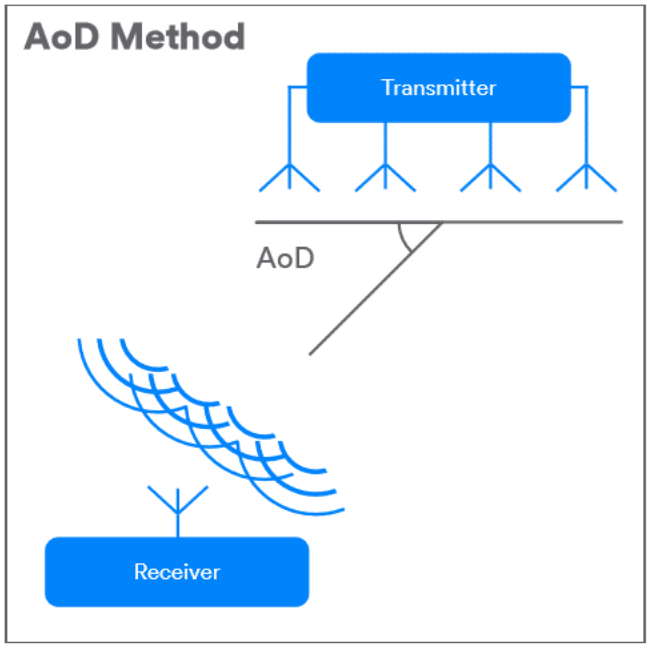
- Many-to-many topology
  - Devices relay messages to reach nodes outside direct radio range
- Uses managed flooding, not point-to-point routing
  - Every node that hears a message relays it once
  - Hop-count (TTL) field prevents infinite loops
- Use case
  - Large-scale deployments where a single-hub star topology can't reach or scale
  - Hundreds or thousands of nodes (smart building lighting, sensors)
- Trade-off
  - Robust
    - No single point of failure, self-healing if a node drops out
  - Cost
    - Overhead grows with network size, since messages are flooded rather than routed

Bluetooth 5.1: Direction Finding

- Estimating a device's direction, not just its presence
  - Angle of Arrival (AoA) and Angle of Departure (AoD)
- Enables indoor positioning (~22 cm outdoors in independent tests)

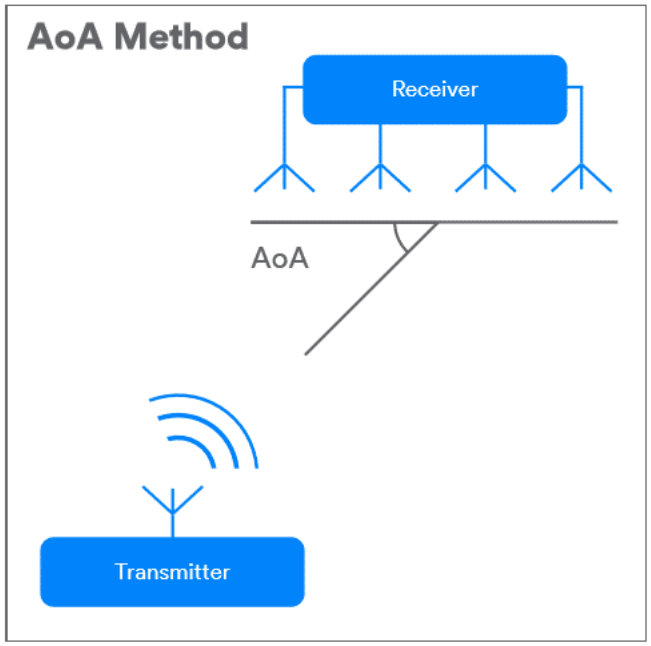


Bluetooth direction finding using angle of arrival (AoA)



Bluetooth direction finding using angle of departure (AoD)

Bluetooth 5.1: Direction Finding

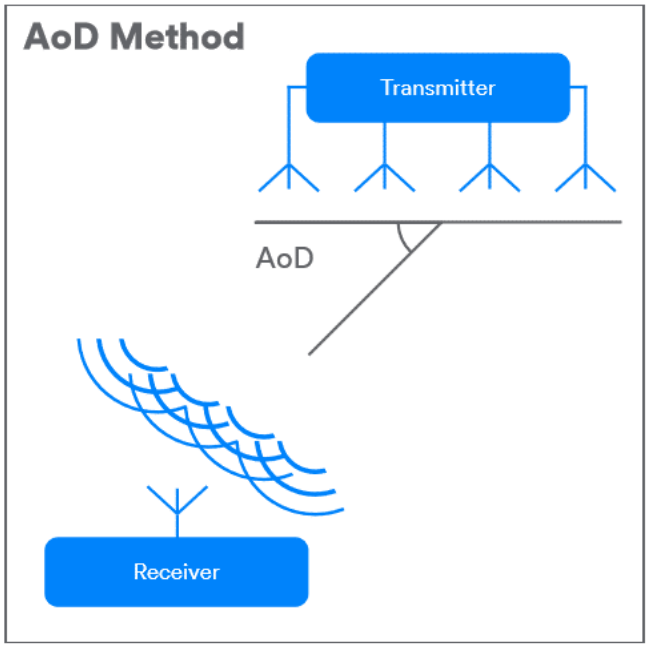


Bluetooth direction finding using angle of arrival (AoA)

- Transmitter uses one antenna
- Receiver uses an antenna array
- Receiver measures phase differences across antennas to compute angle

Bluetooth 5.1: Direction Finding

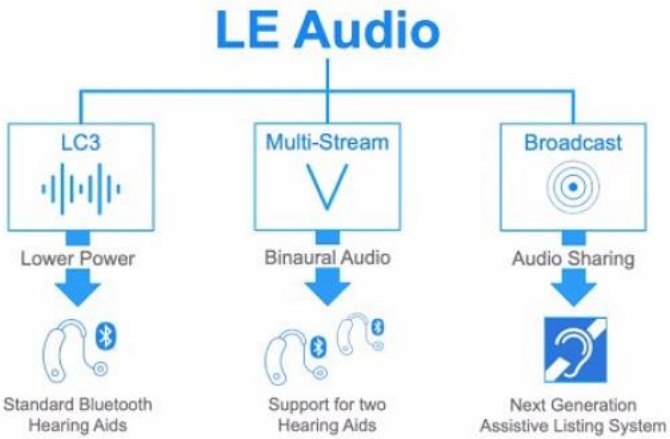
- Transmitter uses an antenna array (switches between antennas)
- Receiver has one antenna
- Reciever measures phase over time as the transmitter switches



Bluetooth direction finding using angle of departure (AoD)

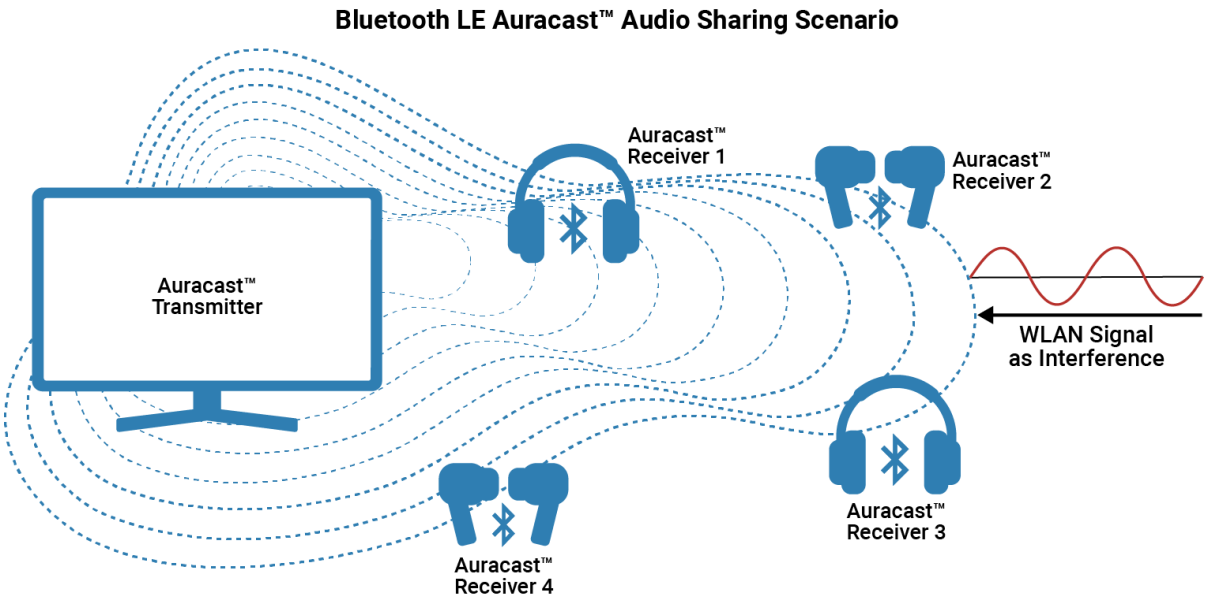
Bluetooth 5.2: LE Audio

- LE Audio
  - Moves audio streaming onto BLE (was previously using power-hungry Classic Bluetooth)
  - LC3 codec
    - Better audio quality than the old SBC codec at lower bitrate, or same quality at much lower power
  - Multi-Stream Audio
    - Synchronized streams to left and right earbuds, each received directly from the phone (many Classic earbuds relay: one earbud receives, then forwards)



Bluetooth 5.2: Auracast

- Auracast (broadcast audio)
  - One source broadcasts to any number of receivers in range; no connection or pairing needed
    - Source audio broadcasts to every headset in the room
    - Receivers "tune in," like a radio station



LE Audio: Isochronous Channels

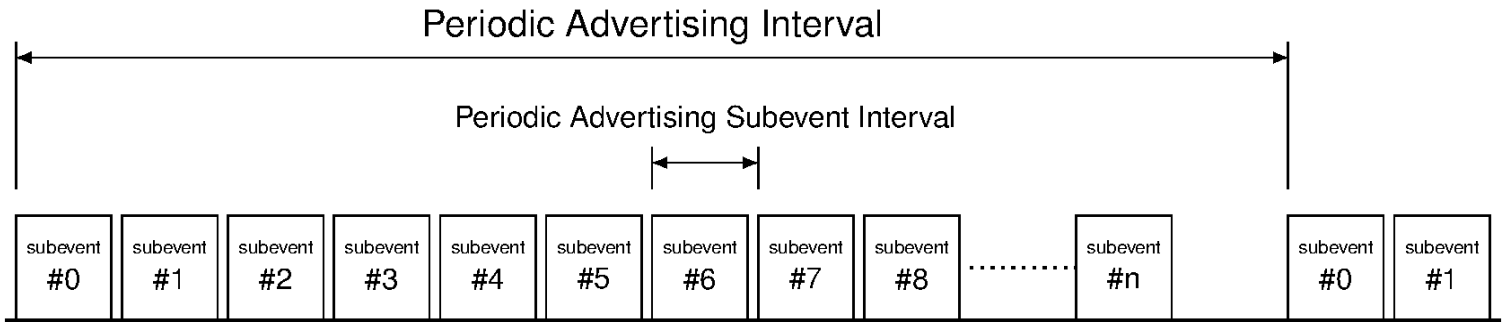
- Problem
  - Audio needs steady timing, not just reliability
  - Regular channels retransmit data until success
  - Delay is unbounded
  - Late audio is useless
- Isochronous (ISO) channels
  - Fixed interval, bounded delay
  - Data goes in time-bound SDUs each ISO interval
  - Lost data only retransmitted inside flush window
  - Dropped if not delivered by flush timeout
- Two flavors
  - CIS (Connected Isochronous Stream)
    - Acknowledged, one per earbud, grouped in a CIG
  - BIS (Broadcast Isochronous Stream)
    - No acknowledgment, grouped in a BIG; basis of Auracast
- Presentation delay
  - All receivers play a sample at the same instant, so left and right stay in sync
- Also motivated by hearing aids
  - low power, direct streaming, shared broadcasts in public venues

Bluetooth 5.4: PAwR

- Problem
  - SCAN_REQ/SCAN_RSP collide as density grows
- PAwR (Periodic Advertising with Responses)
  - Periodic advertising
  - Advertiser sends at a fixed interval
  - Scanners wake only at those times

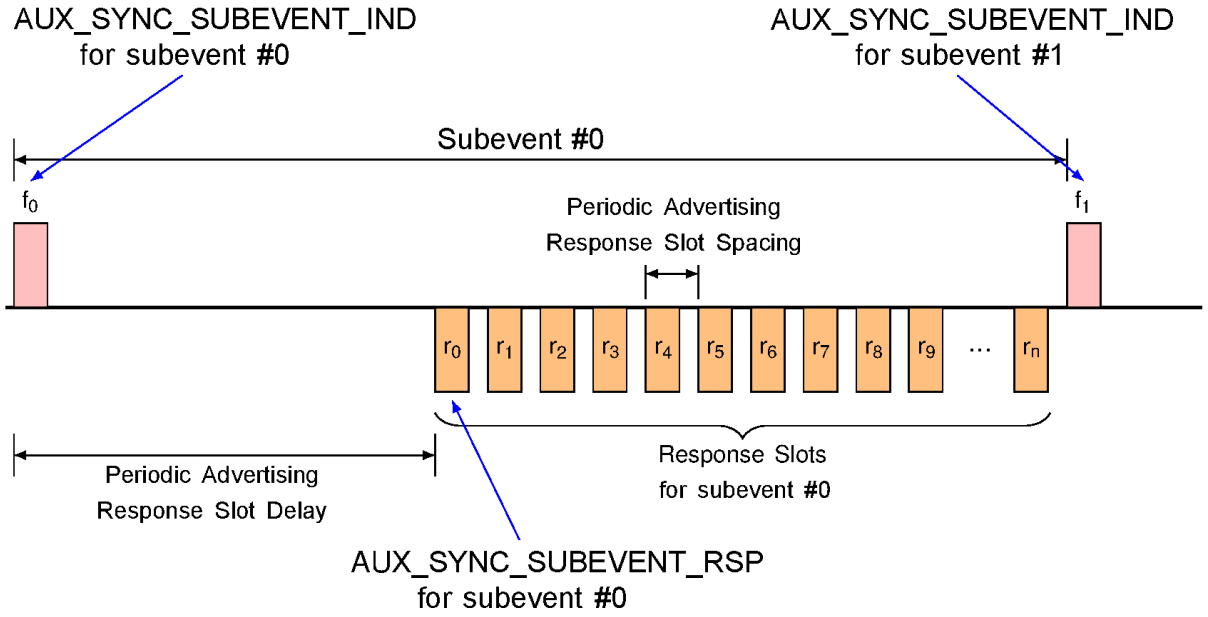
Bluetooth 5.4: PAwR

- Scheduled response channel (i.e., TDMA)
  - Each event is split into subevents (up to 128)
  - Clients are assigned to a group
  - Advertiser transmits to that group
    - Devices reply in their own slot (up to 255 slots)
    - Contention free!



Bluetooth 5.4: PAwR

- Scheduled response channel (i.e., TDMA)
  - Advertiser transmits to the group
    - Devices reply in their own slot (up to 255 slots)
    - Contention free!

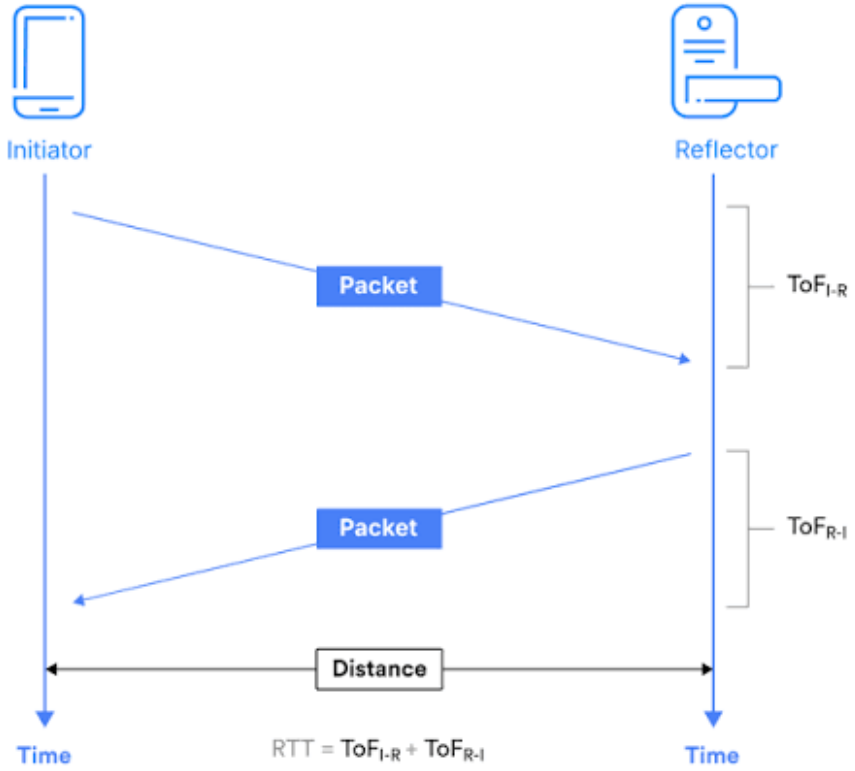


Bluetooth 5.4: PAwR

- TDMA has no random backoff
  - A tag wakes only for its subevent and its own response slot, then sleeps
- Connectionless two-way link
  - Advertiser can reach thousands of tags without individual connections
  - Advertiser sends commands and collect replies
- Privacy and security
  - Encrypted advertising data keeps payloads private
- Use cases
  - Electronic shelf labels (price updates, acknowledgments), sensors, asset tags

Bluetooth 6.0: Channel Sounding (2024)

- Secure, precise distance measurement (ranging) between two devices

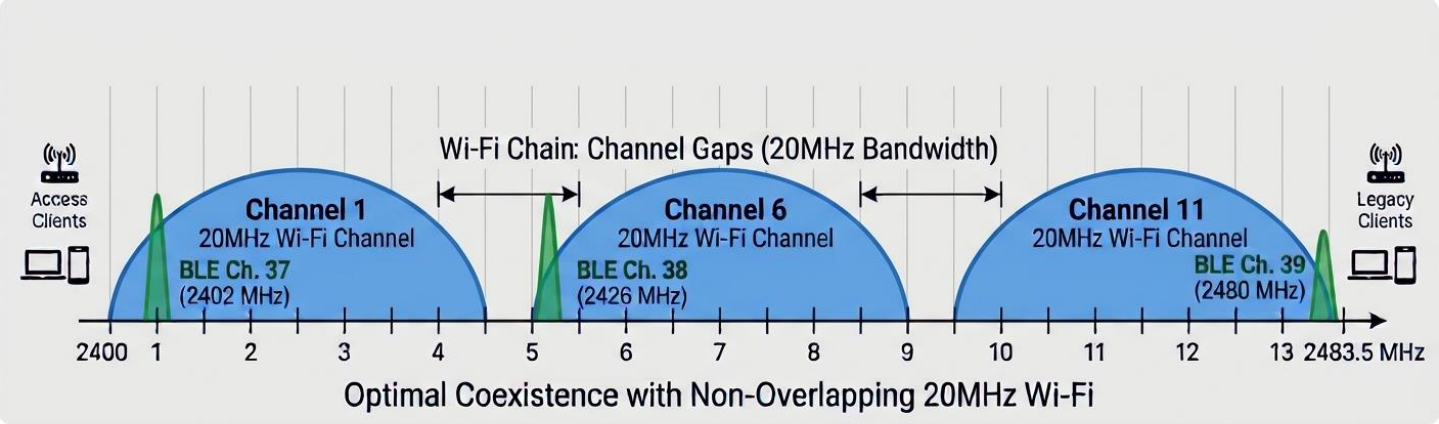


Bluetooth 6.0: Channel Sounding (2024)

- Secure, precise distance measurement (ranging) between two devices
- Combines two methods
  - Phase-Based Ranging (PBR)
    - Measures phase across multiple frequencies
  - Round-Trip Time (RTT)
    - Measures time-of-flight directly
- Design target
  - ±tens of cm;
  - ~±20 cm in early implementations

Wi-Fi Channel Bonding vs. BLE Advertising

- 2.4 GHz Channel Bonding
  - Allowed but uncommon



Wi-Fi and BLE Coexistence

- Wi-Fi hurts BLE
  - BLE advertising has no carrier sense
    - Advertisements overlapping Wi-Fi frames are usually lost
    - Channel 39 is often cleanest
  - Bonding makes AFH harder
    - 40 MHz blocks ~20 of 37 BLE data channels
    - May not leave minimum of 20 channels free for hopping
- BLE hurts Wi-Fi much less
  - BLE is 1-2 MHz wide with low duty cycle
    - Hits a few OFDM subcarriers, mostly corrected by FEC
  - Classic streaming hops over 79 channels
    - ~Half of hops land in a 40 MHz channel
    - AFH must avoid it

Wi-Fi and BLE Coexistence

- Same device (combo chip)
  - Shared antennas cannot transmit Bluetooth and Wi-Fi at the same time
    - Frequency separation is not enough on a shared chip or antenna
    - Bluetooth transmitting can desensitize the Wi-Fi receiver, and vice versa
  - Radios arbitrate in time
  - Active Bluetooth may push 2.4 GHz Wi-Fi down to 20 MHz or throttles it

Wi-Fi and BLE Best Practice

- Move Wi-Fi away from 2.4 GHz
  - 5 and 6 GHz Wi-Fi do not overlap Bluetooth
  - Bluetooth stays at 2.4 GHz
- Wi-Fi 2.4GHz channel planning
  - Use 20 MHz channels 1/6/11
  - Avoids BLE advertising channels
- Bluetooth side
  - AFH and BLE's short, low-duty-cycle bursts