Lecture 03
BLE Introduction

CS397/497 – Wireless Protocols for IoT

Branden Ghena – Spring 2022

Materials in collaboration
with Pat Pannuto (UCSD)

Administrivia

- Think about project ideas
  - Make posts on campuswire with ideas to iterate on
    - Also good for recruiting partners
  - I'm happy to schedule office hours to discuss!
- First lab is available
  - Goal: familiarize yourself with Wireshark
    - Install it, do some basic scanning, explore a little
  - Writeup: "prove to me that you did this lab"
    - Due next week Thursday by end-of-day
    - Submit on Gradescope
  - Next week: Wireshark + BLE

Today’s Goals

- Introduction to Bluetooth Low Energy
  - What are the goals of the protocol?
  - What do the lower layers look like?
  - What roles do devices take?

Bluetooth Low Energy Resources

- Good walkthrough of BLE:
  - https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf

- [5.2 specification] [4.2 specification] (link to PDF download)
  - Also: [Supplement v9]

- I used a mix of 5.2 and 4.2 for this
  - Will talk about BLE 5 differences as part of next lecture

Outline

- **BLE Background**
- BLE Layers
  - Physical Layer
  - Link Layer
- BLE roles
  - Advertising
  - Scanning

Basics of Bluetooth Low Energy (BLE)

- Direct device-to-device communication
  - Usually: Computer to Thing
  - Smartphone to device, Laptop to device, etc.

- Focus on making the "Thing" really low energy
  - Push energy-intensive requirements onto "Computer"

- Devices (Computer or Thing) are servers with accessible fields
  - Not the traditional send-explicit-packets interface you might be expecting
  - Lower layers are still exchanging packets to make it work

A note on outdated notation

- Master/Slave paradigm
  - Master is the “Computer” and is in charge of interaction
  - Slave is the “Device” and has little control over interaction parameters
  - Really common notation in EE side of the world.
    - Not intended to be harmful, but also literally inconsiderate.

- Field is changing for the better. It’s going to take some time.
  - **Central/Peripheral**
  - Device/Peripheral
  - Controller/Peripheral
  - Master/Minion
  - Primary/Secondary

BLE development

- Protocol development
  - Research product
  - Specification
  - Hardware support
  - Usefulness and iteration

- Bluetooth Low Energy
  - Research in early 2000s: Bluetooth Low End Extension and Wibree
  - Specification in 2009: Bluetooth version 4.0
  - Hardware support in 2011/12: iPhone 4s, nRF51 series
  - 4.1 and 4.2 (2014), 5.0 (2016, first in phones 2017, really 2019 though)

Bluetooth has a long history — the IoT is near-exclusively BLE (Bluetooth 4.0+) as opposed to Bluetooth Classic (<4.0)

Year	Bluetooth Standard	Data Rate	Modulation	Notes
1999	V1.0	1 Mb/s	GFSK	• The Bluetooth 1.0 Specification is released by the Bluetooth SIG
2003	V1.2	1 Mb/s	GFSK	• First FDA-approved Bluetooth medical system. Bluetooth product shipments grow to 1 million/week
2004	V2.0 + EDR	1 Mb/s 2 Mb/s 3 Mb/s	GFSK π /4–DQPSK 8-DPSK	• Introduction of Enhanced Data Rate (EDR) for faster data transfer. • Bluetooth product shipments surpasses to 3 million/week
2007	V2.1 + EDR	1 Mb/s 2 Mb/s 3 Mb/s	GFSK π /4–DQPSK 8-DPSK	• Introduction of secure simple pairing (SSP) and extended inquiry response (EIR) for Bluetooth devices
2009	V3.0+HS	1 Mb/s 2 Mb/s 3 Mb/s	GFSK π /4–DQPSK 8-DPSK	• Introduction of AMP (Alternative MAC/PHY) and the addition of 802.11 as a high-speed transport with data transfer speeds up to 24 Mbit/s.
2009	V3.0+HS	1 Mb/s 2 Mb/s 3 Mb/s	GFSK π /4–DQPSK 8-DPSK	• Introduction of AMP (Alternative MAC/PHY) and the addition of 802.11 as a high-speed transport with data transfer speeds up to 24 Mbit/s.
2010	V4.0 (Smart)	1 Mb/s 2 Mb/s 3 Mb/s	GFSK π /4–DQPSK 8-DPSK	• Introduction of Bluetooth Low Energy protocol and AES encryption
2013	V4.1	1 Mb/s 2 Mb/s 3 Mb/s	GFSK π /4–DQPSK 8-DPSK	• MWS (Mobile Wireless Standard) Coexistence • SIG membership surpasses 20,000 companies
2014	V4.2	1Mb/s 2Mb/s 3Mb/s	GFSK π /4–DQPSK 8-DPSK	• Smart sensor allows flexible internet connectivity • Increased privacy (Le Privacy 1.2 and LE Secure Connections) • LE Data Length Extension increases data throughput with packet capacity increase of 10x compared to previous versions.

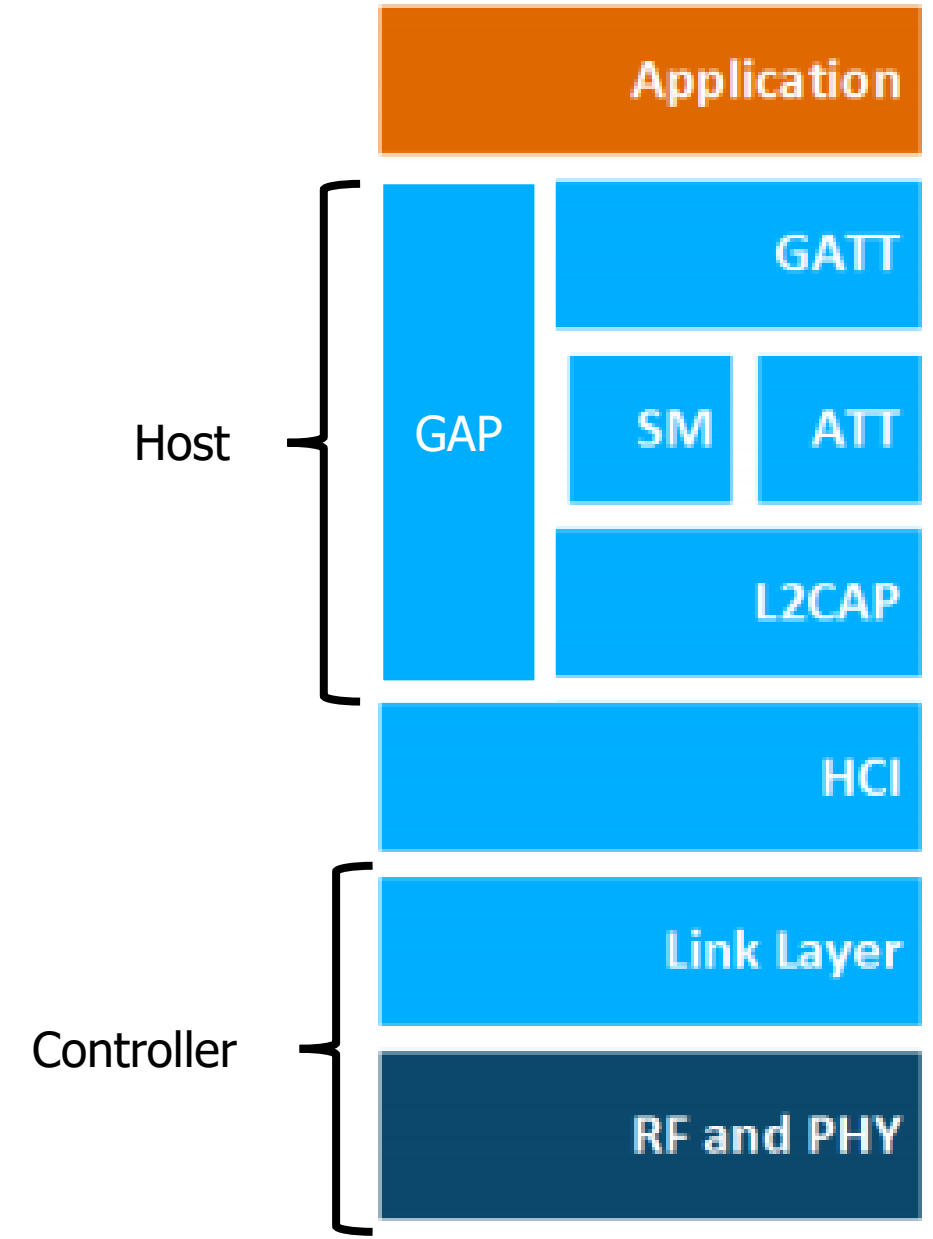
Bluetooth Specification

- Problem: a bit overwhelming…
  - 5.2 spec: **3256 pages**
  - We only care about Vol 6: Low Energy Controller
    - Part A: Physical Layer Specification
    - Part B: Link Layer Specification
    - CSS: Part A: Data Types Specification
    - So ~250 pages

- Tip: be willing to just ignore things when skimming specs
  - 5.2 spec covers BLE and Bluetooth Classic and a bunch of upper layer stuff that we never have to care about

BLE Layers

- Host – Configuration and Server
  - GAP – Generic Access Profile
    - Configure advertising
  - GATT – Generic ATTribute profile
    - Configure connections
- HCI - Host Controller Interface (sigh)
- Controller - Communication
  - Link Layer – send packets
  - RF and PHY – send bits

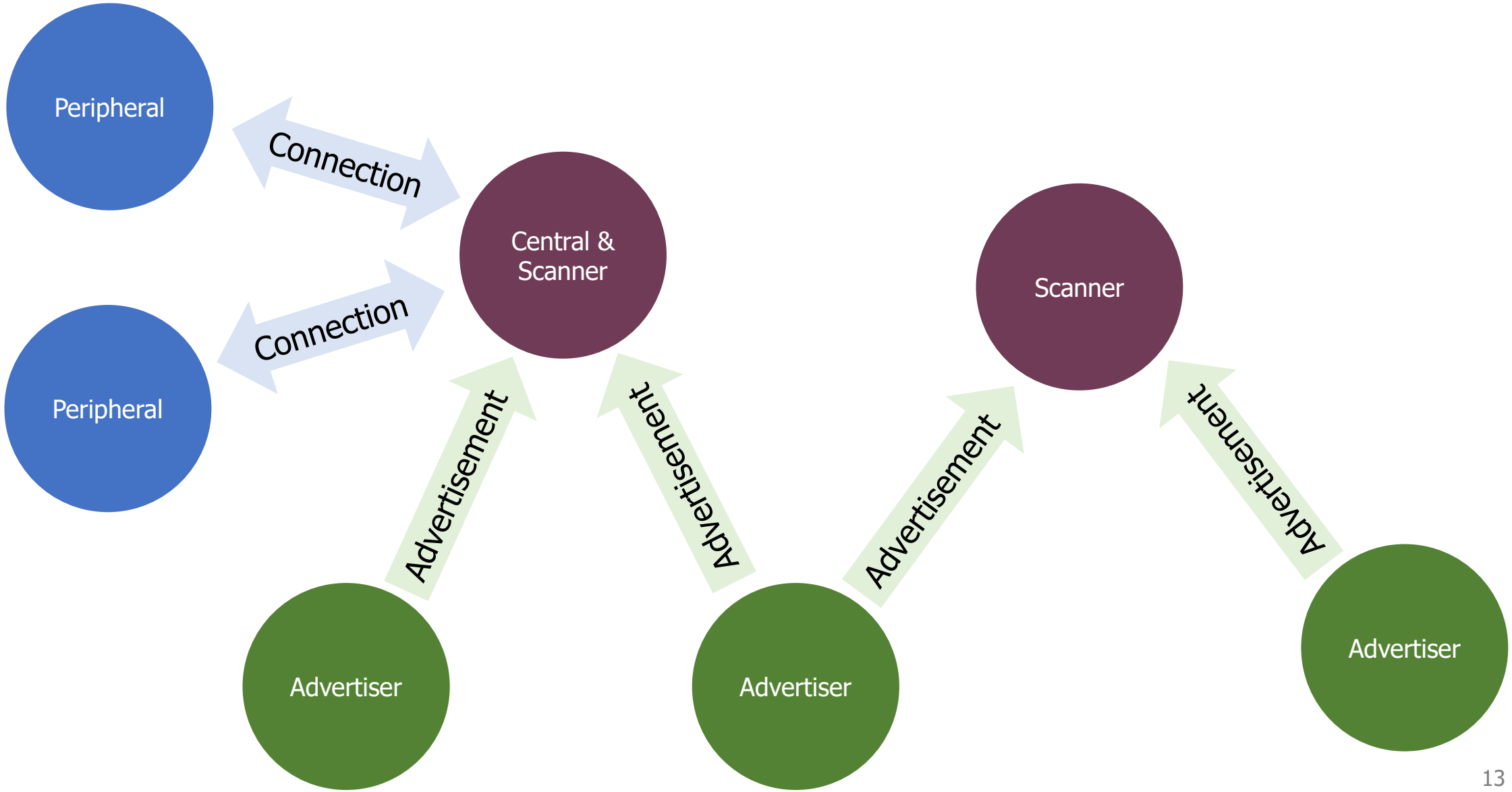


BLE mechanisms

- Advertising
  - Discovery
  - Advertisements – broadcast messages indicating device details
  - Ephemeral, uni-directional communication from Advertiser to Scanner(s)
  - ALOHA access control

- Connections
  - Interaction
  - Bi-directional communication between Peripheral and Central
  - Maintained for some duration
  - TDMA access control

BLE network topology



Multiple roles at the same time

- Topology picture is a simplification of roles

- A single device can have multiple roles simultaneously
  - Scanning and Advertising simultaneously
  - Peripheral and Scanner and Advertiser simultaneously
  - Peripheral and Scanner and Central and Advertiser simultaneously
    - Getting a bit out of hand though

- Also possible:
  - One Peripheral can be connected to multiple Centrals
    - This is relatively new in BLE still, you'll find old docs saying you can't

Break + Check your understanding

- Which roles is each device likely to have?
  - Keyboard
  - Laptop
  - Smartphone

Break + Check your understanding

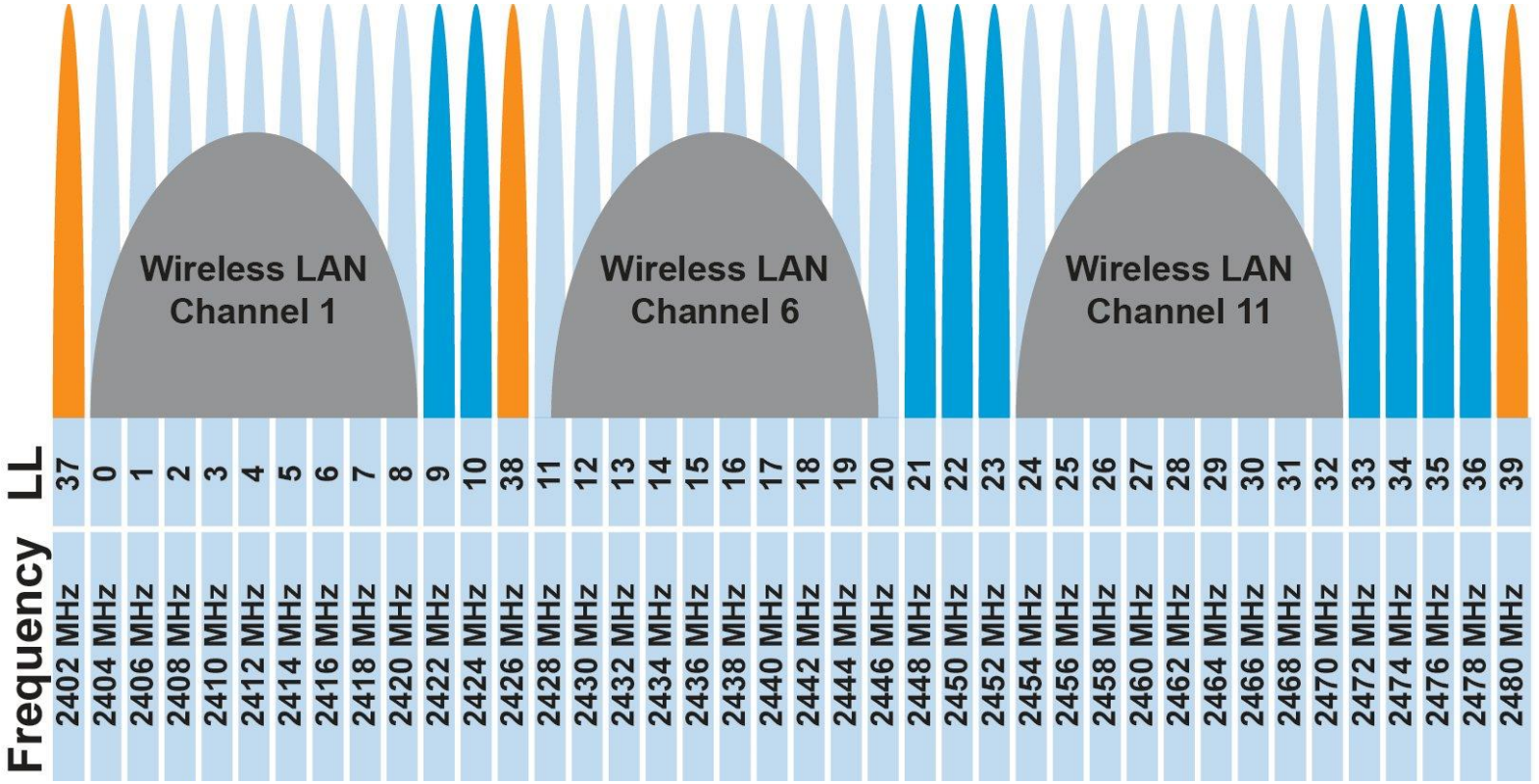
- Which roles is each device likely to have?
  - Keyboard
    - Advertiser and Peripheral
  - Laptop
    - Scanner and Central
  - Smartphone
    - Advertiser, Peripheral, Scanner, and Central

Outline

- BLE Background
- **BLE Layers**
  - **Physical Layer**
  - Link Layer
- BLE roles
  - Advertising
  - Scanning

BLE frequency

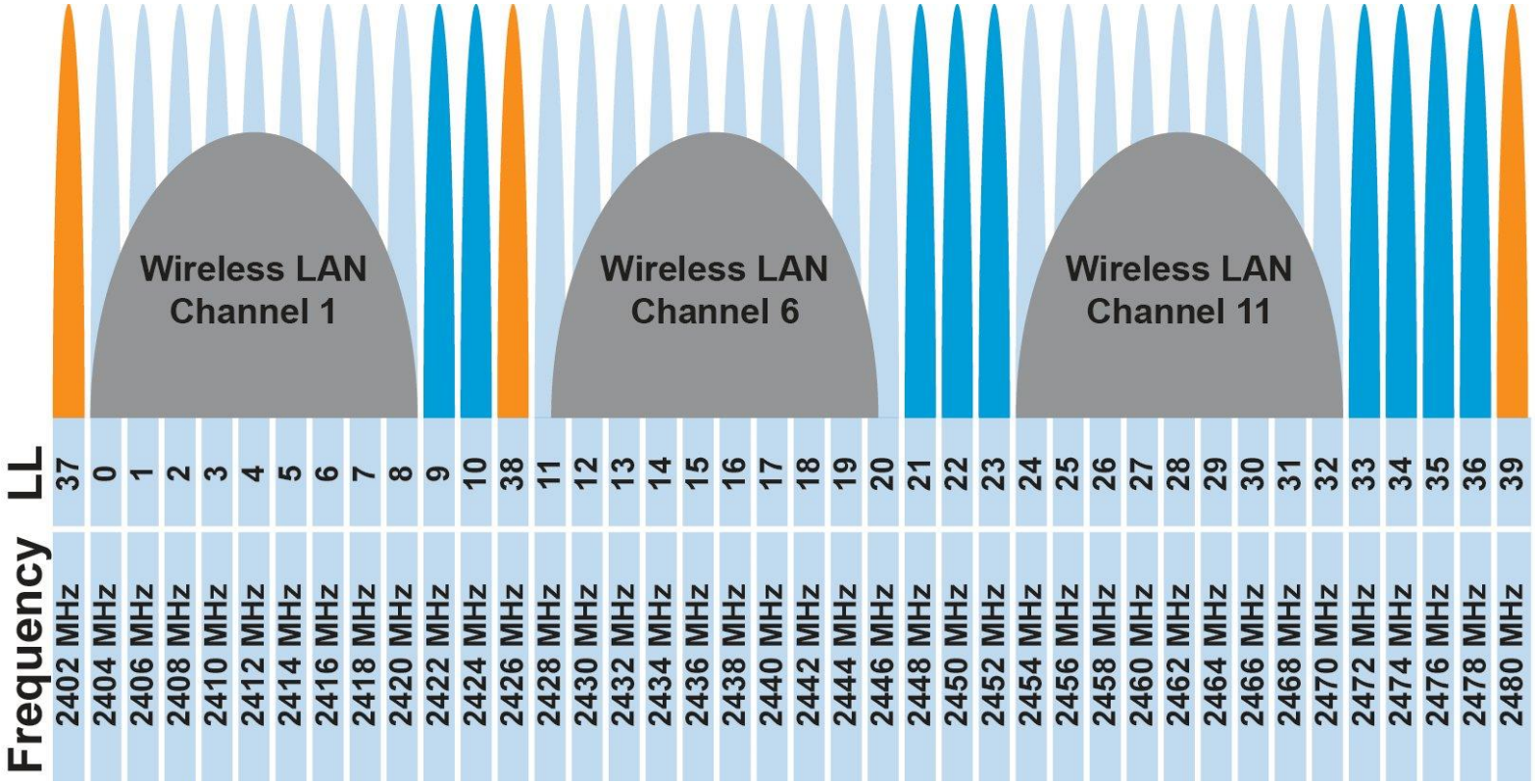
- 2.4 GHz carrier, Forty 2-MHz channels, 1 Mbps data rate
  - 37, 38, 39 for advertising
  - 0-36 for connection (FHSS)



Why doesn't BLE avoid WiFi altogether?

BLE frequency

- 2.4 GHz carrier, Forty 2-MHz channels, 1 Mbps data rate
  - 37, 38, 39 for advertising
  - 0-36 for connection (FHSS)



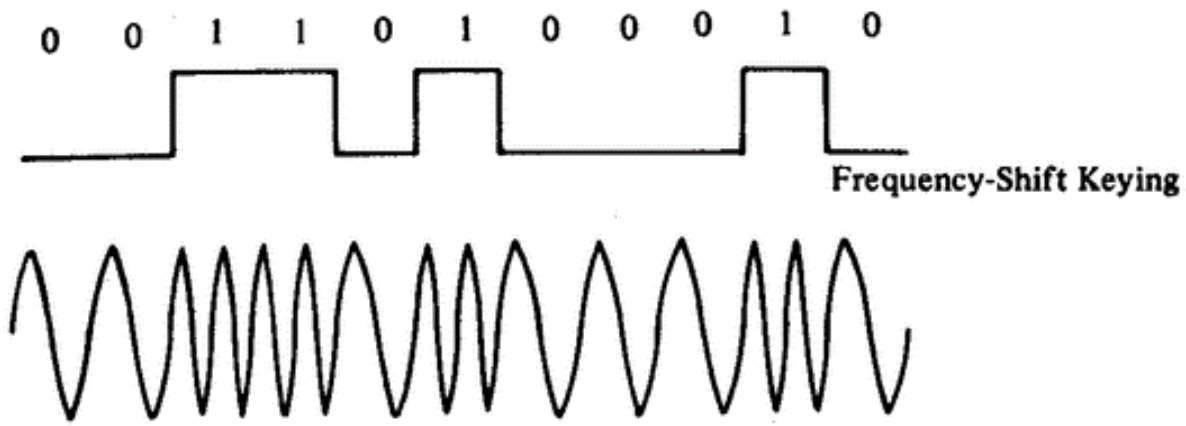
Why doesn't BLE avoid WiFi altogether?

Can't on 2.4 GHz

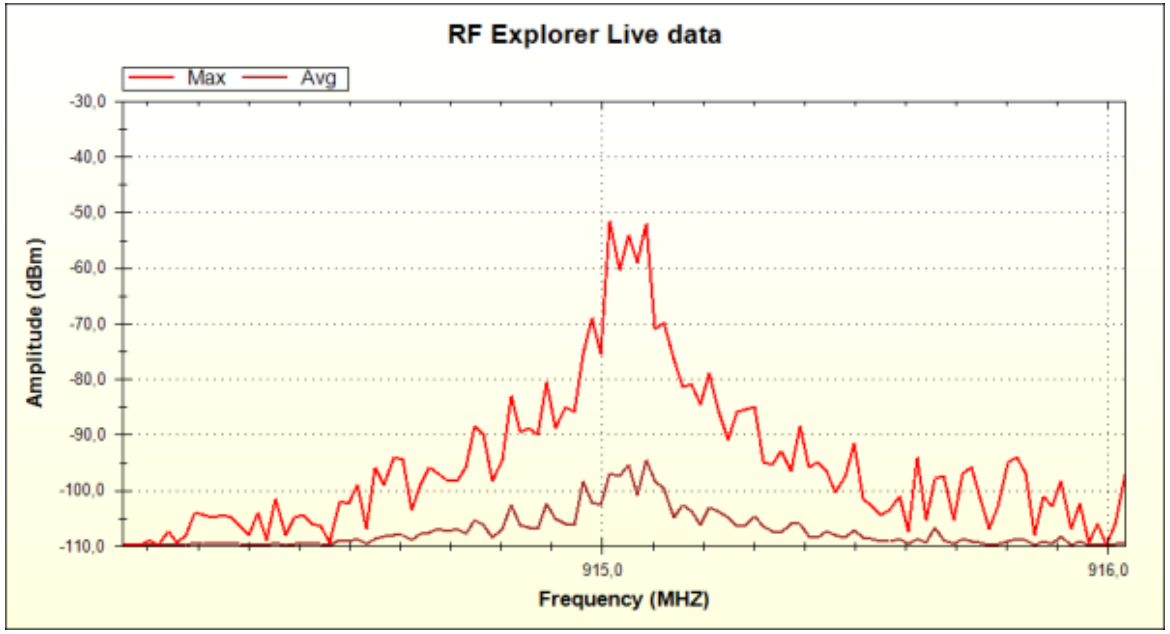
Wants 2.4 GHz for technology improvements

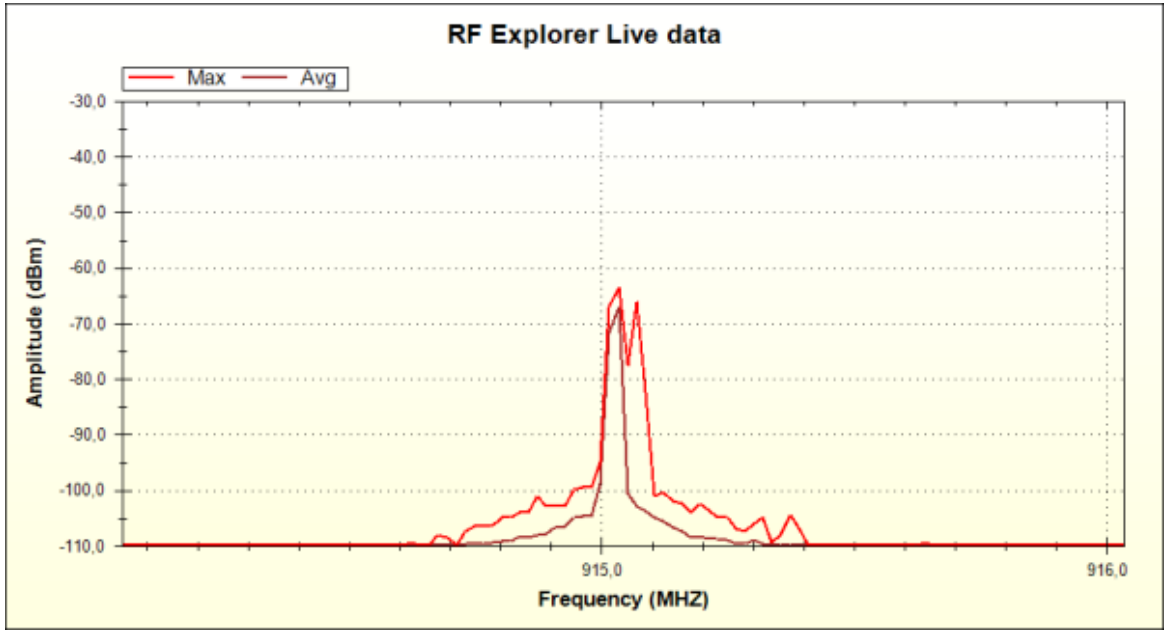
BLE modulation

- Gaussian Frequency-Shift Keying (GFSK)
  - Improvement on base Frequency-shift Keying
  - Smoother transitions between bits -> reduces nearby interference

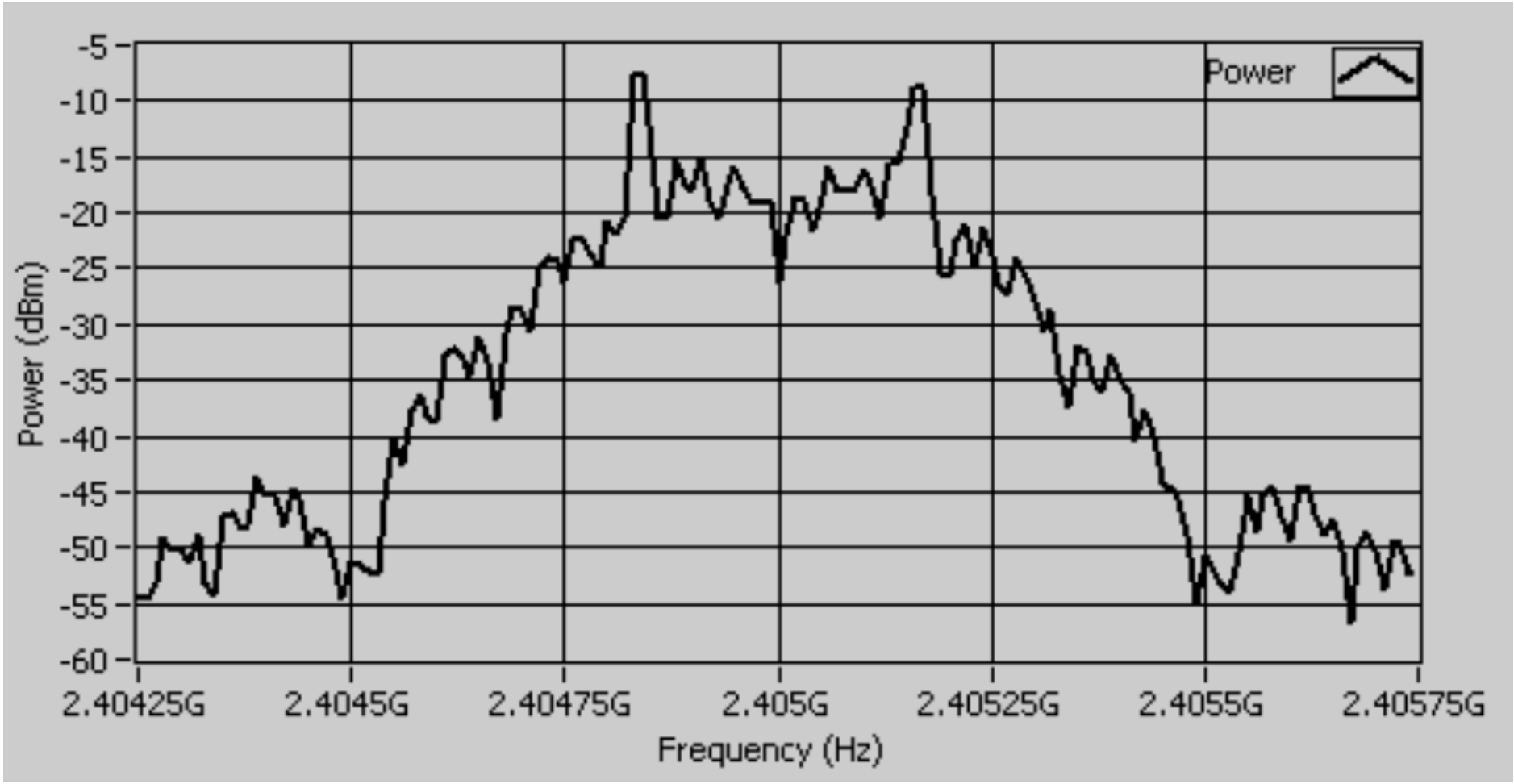


Gaussian FSK lessens spectral leakage at the expense of some loss in intersymbol discriminability





An example from `good case` BLE hardware



BLE signal strength

The requirements for a Bluetooth low energy radio are as follows:

Feature	Value
Minimum TX power	0.01 mW (-20 dBm)
Maximum TX power	100 mW (20 dBm)
Minimum RX sensitivity	-70 dBm (BER 0.1%)

The typical range for Bluetooth low energy radios is as follows:

TX power	RX sensitivity	Antenna gain	Range
0 dBm	-92 dBm	-5 dB	160 meters
10 dBm	-92 dBm	-5 dB	295 meters

The range to a smart phone is typically 0-50 meters due to limited RF performance of the phones.

- Remember nRF52840 capabilities
  - Transmit: up to 8 dBm
  - Receive sensitivity: -95 dBm

Outline

- BLE Background
- **BLE Layers**
  - Physical Layer
  - **Link Layer**
- BLE roles
  - Advertising
  - Scanning

Packet structure

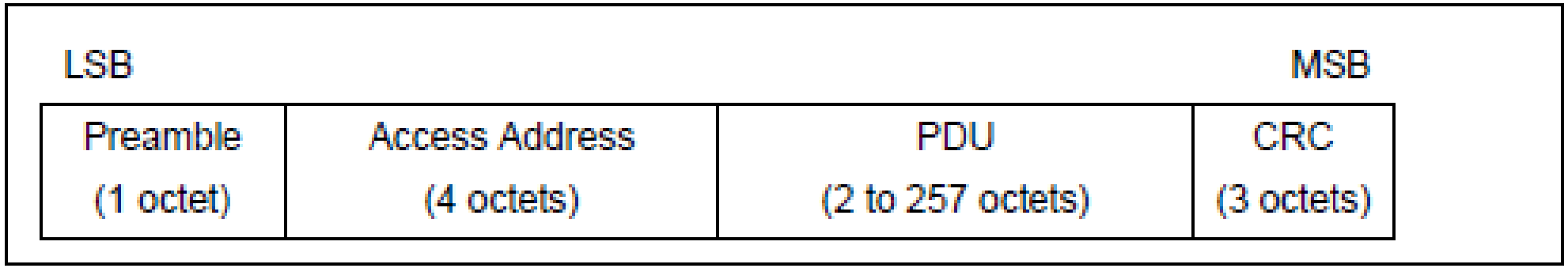


Figure 2.1: Link Layer packet format

- Same packet structure for both advertisements and connections
  - Fields are filled in little endian (the opposite of network byte order 😠)
- Access address unique for each connection (randomly chosen)
  - In Advertising always set to 0x8E89BED6

Device addresses

- Public and private address forms
- Public
  - 48 bits: 24-bits of company ID, 24-bits of company assigned number
  - Literally the same MAC address scheme as Ethernet and WiFi
- Private
  - Top two MSbs specify type
    - 46 bits of random
    - 46 bits of hash of an identity key

- **Why have the two types?**

Device addresses

- Public and private address forms
- Public
  - 48 bits: 24-bits of company ID, 24-bits of company assigned number
  - Literally the same MAC address scheme as Ethernet and WiFi
- Private
  - Top two MSbs specify type
    - 46 bits of random
    - 46 bits of hash of an identity key

- **Why have the two types? Privacy**

Data whitening

- Avoid long series of repetitive bits (all zeros or all ones)
  - Would cause RF noise to be more focused in one direction
  - Radio hardware desires output to have zero DC-bias (or close to that)
  - Great example of the PHY and MAC being interwoven in wireless

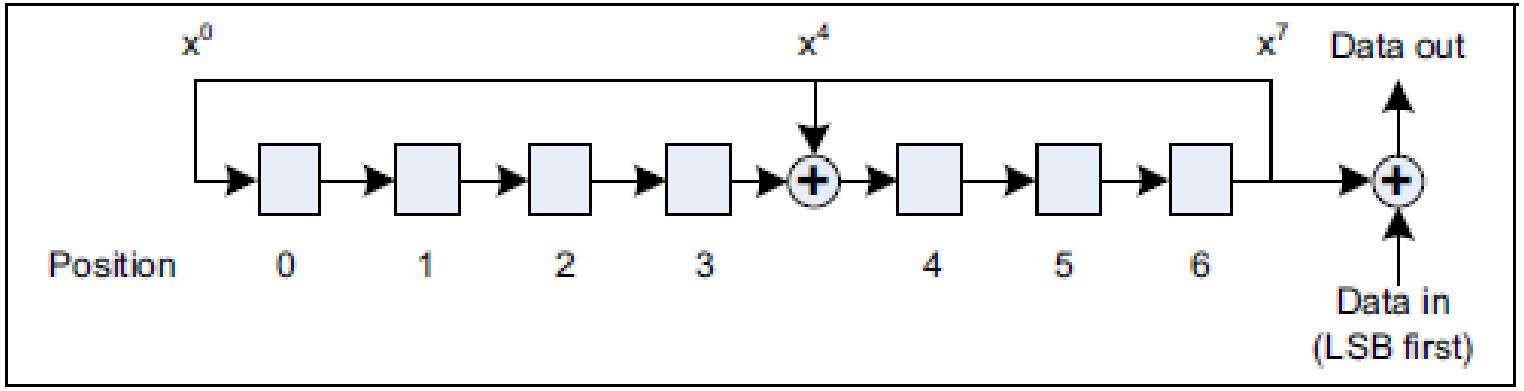


Figure 3.3: The LFSR circuit to generate data whitening

- I always forget this exists, since hardware usually handles it automatically

Bit processing pipeline

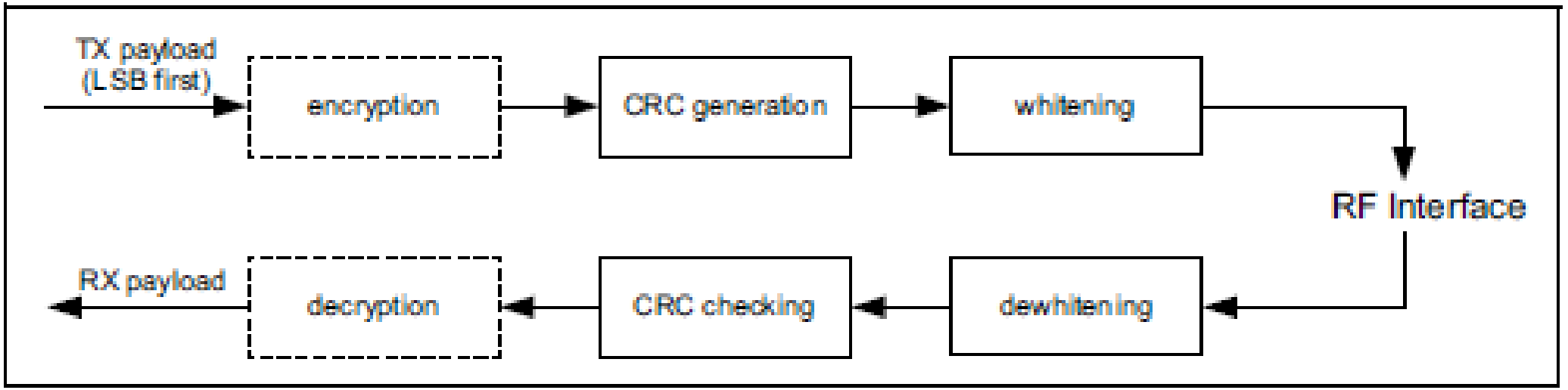


Figure 3.1: Payload bit processes for the LE Uncoded PHYs

Break + Question

- With enough scanners, could you track BLE devices as they move?

Break + Question

- With enough scanners, could you track BLE devices as they move?
  - Depends on how long they use a device address for
  - You can do a scan of BLE transmissions to find device addresses
  - Scans at multiple locations can detect when a device moves throughout an area
  - But if the device re-randomizes between two scanners, you can’t follow it anymore
    - Re-randomizing at a scanner could be detectable…

Outline

- BLE Background
- BLE Layers
  - Physical Layer
  - Link Layer
- **BLE roles**
  - **Advertising**
  - Scanning

Advertising

- BLE discovery mechanism
  - Make nearby devices aware of advertiser's existence
  - Communicate some information from or about advertiser
  - Traditional purpose is to enable connections, but this is also useful for general communication

- Advertisements
  - Periodic broadcast messages with data
- Scan Requests/Responses
  - Scanner sends responses after getting a request
    - Only occurs when scanner is listening
  - Almost literally "bonus advertisement data"

Advertising packet layering

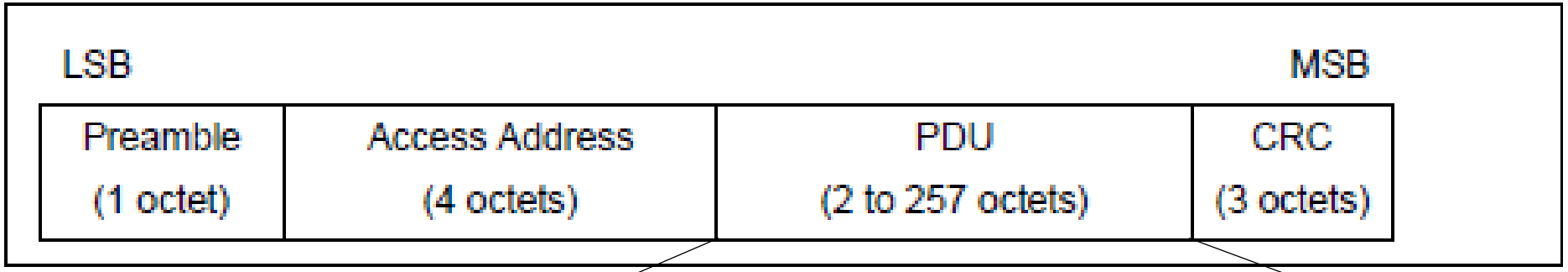


Figure 2.1: Link Layer packet format

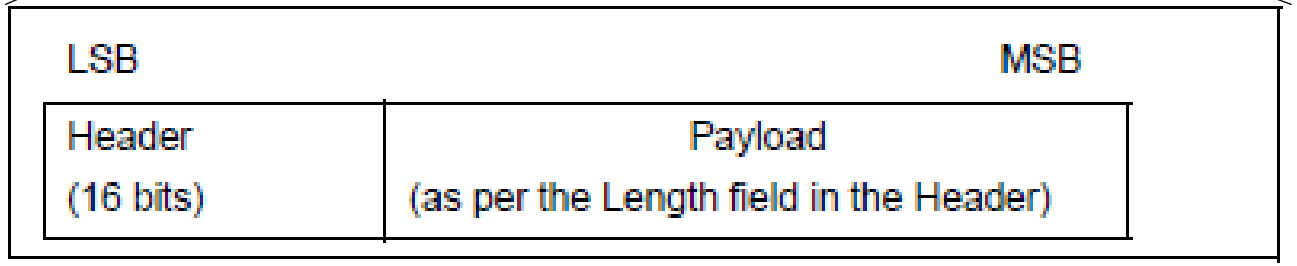


Figure 2.2: Advertising channel PDU

BLE advertising header

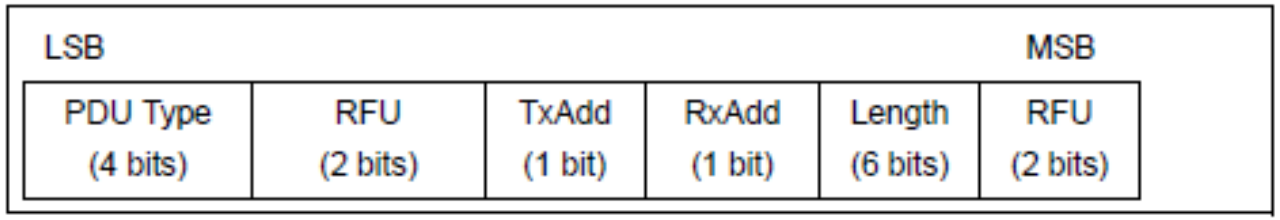


Figure 2.3: Advertising channel PDU Header

PDU Type $b_3b_2b_1b_0$	Packet Name
0000	ADV_IND
0001	ADV_DIRECT_IND
0010	ADV_NONCONN_IND
0011	SCAN_REQ
0100	SCAN_RSP
0101	CONNECT_REQ
0110	ADV_SCAN_IND
0111-1111	Reserved

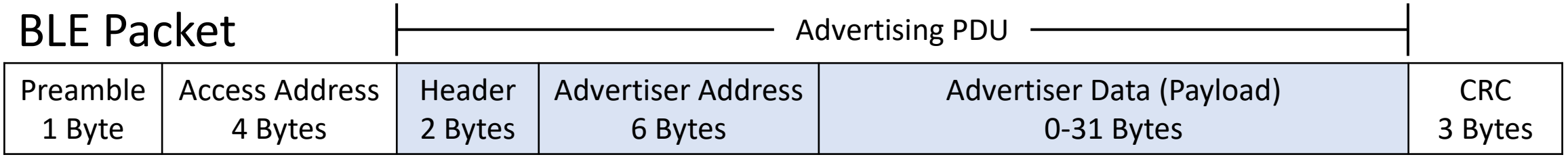
Table 2.1: Advertising channel PDU Header's PDU Type field encoding

- ADV_IND
  - Advertisement
  - Allows connections and scan requests
- ADV_NONCONN_IND
  - Advertisement
  - No connections or scan requests
- ADV_SCAN_IND
  - Advertisement
  - No connections but allows scan requests
- SCAN_REQ
  - Scan request
- SCAN_RSP
  - Scan response

Advertisement payloads

Payload	
AdvA (6 octets)	AdvData (0-31 octets)

- AdvA – address of the advertiser
  - TxAdd bit from header specifies if this is a “public” or “random” address
- Remaining up to 31 bytes are available for use
- Putting it all together, up to 47 bytes total:



Scan Requests and Responses

- Scan request
  - Just the two addresses: the scanner's and the advertiser's

Payload	
ScanA (6 octets)	AdvA (6 octets)

Figure 2.8: SCAN_REQ PDU Payload

- Scan response
  - Identical to an advertisement
  - But only occurs after a request

Payload	
AdvA (6 octets)	ScanRspData (0-31 octets)

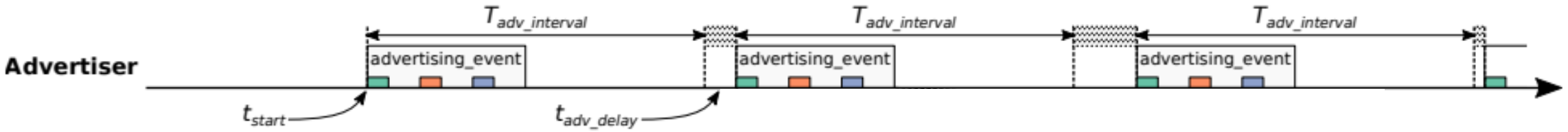
Figure 2.9: SCAN_RSP PDU payload

Advertising timing

- Advertising Events occur periodically [20ms – 10.24 s] (or longer)
  - Plus a random delay after each instance [0-10ms]
  - **Why?**

- User picks the rate as a tradeoff of energy and discovery latency

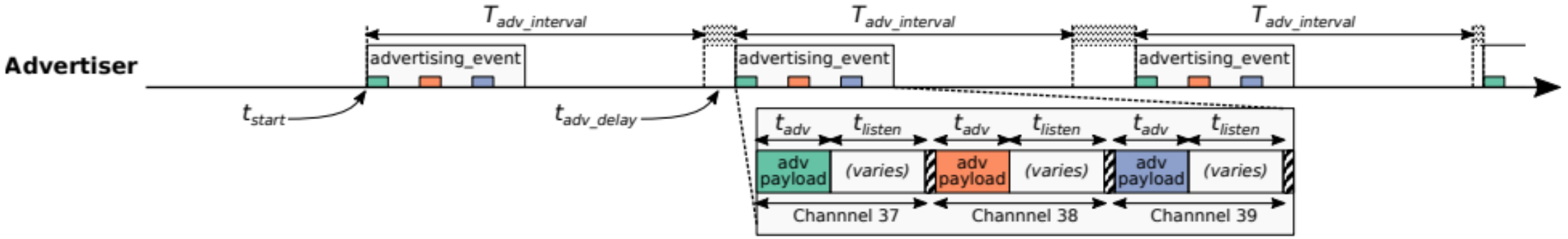
Advertising timing



- Advertising Events occur periodically [20ms – 10.24 s] (or longer)
  - Plus a random delay after each instance [0-10ms]
  - **Why? Avoid repeat collisions**

- User picks the rate as a tradeoff of energy and discovery latency

Advertising event



- Three transmissions, one on each advertising channel
  - Always in the same order

- Transmission, followed by listening window on that same channel
  - Requests will be sent >=150 us (Inter-Frame Spacing, IFS) after Tx
  - Followed by a retune to the next channel frequency
- This short listen window is the magic “low energy” part

Preserving energy in communication

- Most energy is spent listening
  - This is due primarily to how long listening durations are compared to transmissions
- Example: maximum-sized BLE transmission:
  - 8 bits/byte * 47 bytes = 376 bits at 1 Mbps = 0.376 ms transmitting
  - So listening for an entire second is >2500 times longer
  - But listening for only 0.376 ms requires sub-ms synchronization, which itself costs energy to manage…
  - Instead, when advertising, nRF radios listen for ~0.200 ms, only after a transmission

Payload of an advertisement

- What do you stick in the BLE payload anyways?
  - Theoretically whatever you want, but that isn't very compatible
  - Point is to specify capabilities of the advertiser

- Desire: specify payloads in such a way that all scanners can interpret what they mean about the device
  - This is different from traditional internet packets
  - Broadcasts are for _anyone_ to hear, not a specific server/application

- **Ideas?**

TLV Format

- Type – Length – Value
  - Actually, BLE does the length part first
  - Scanner can hop through length/type pairs to find what interests it

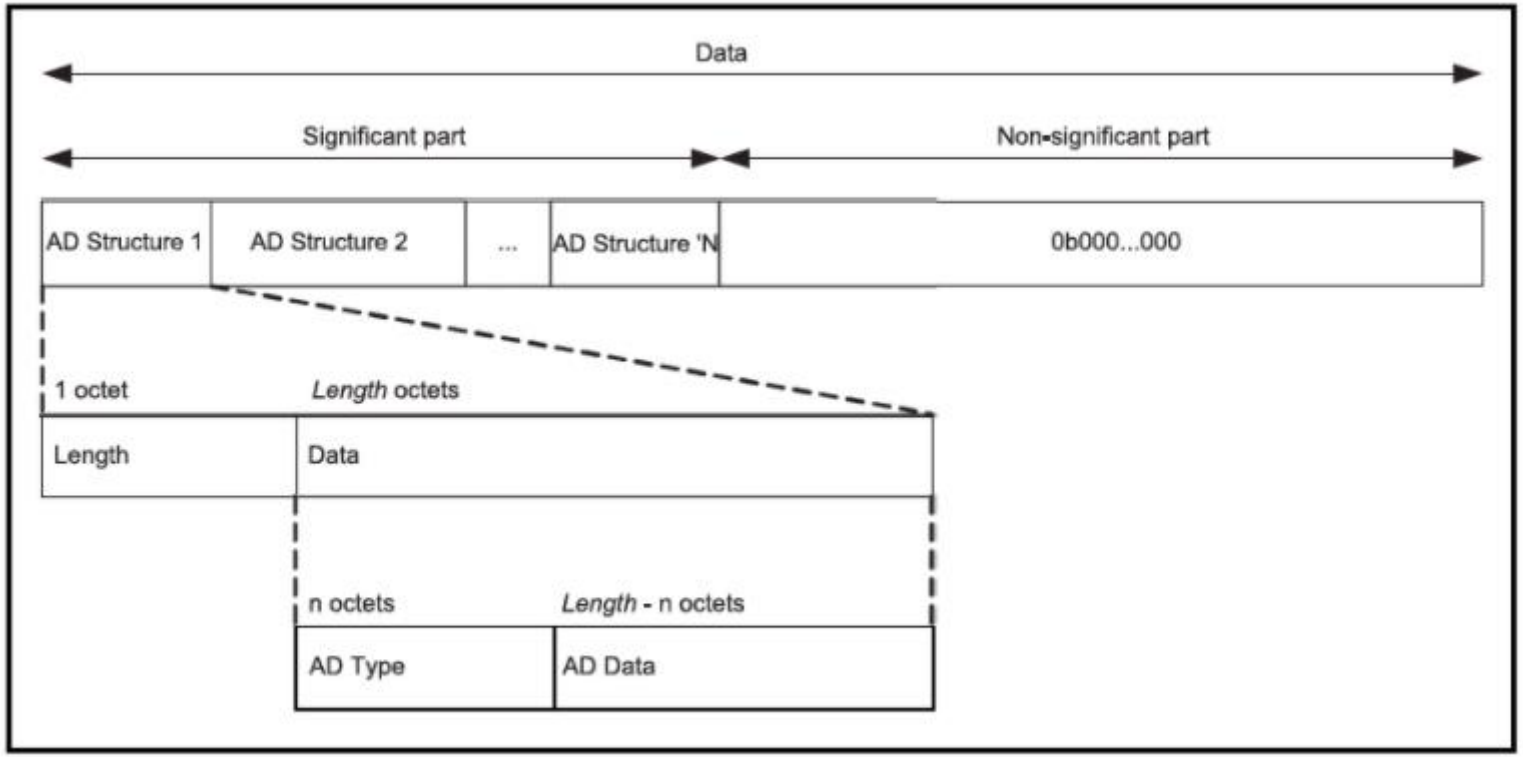


Figure 11.1: Advertising and Scan Response data format

Payload types

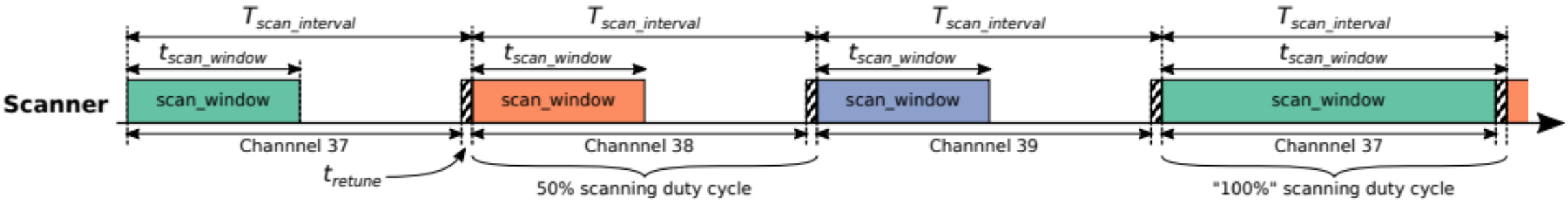
- Listed in the Core Specification Supplement [Supplement v9]
  - Each might have their own considerations about AD Data format

- Flags (supported modes: BLE and Bluetooth) required by Apple?
- Name
- Service UUID
- TX Power Level
- Manufacturer-specific data
- And about twenty others

Outline

- BLE Background
- BLE Layers
  - Physical Layer
  - Link Layer
- **BLE roles**
  - Advertising
  - **Scanning**

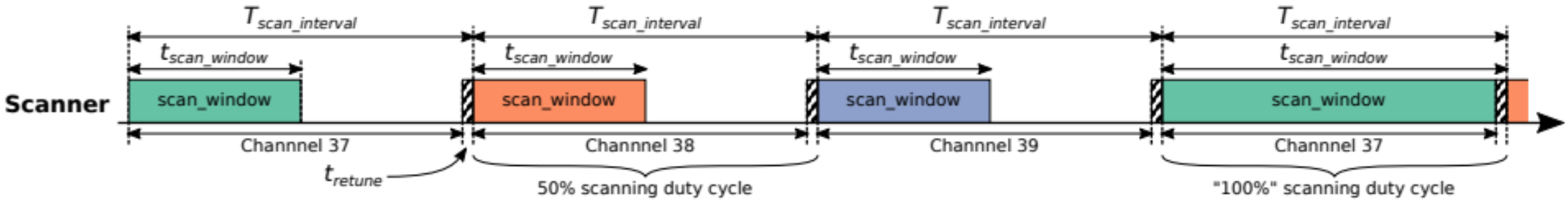
Scanning Pattern



- Iterate through channels, listening for advertisements
  - $T_{scan\_interval}$ controls rate at which channels are changes
  - T_{scan_window} controls duty cycle of listening

- **Why listen at a low duty cycle?**

Scanning Pattern

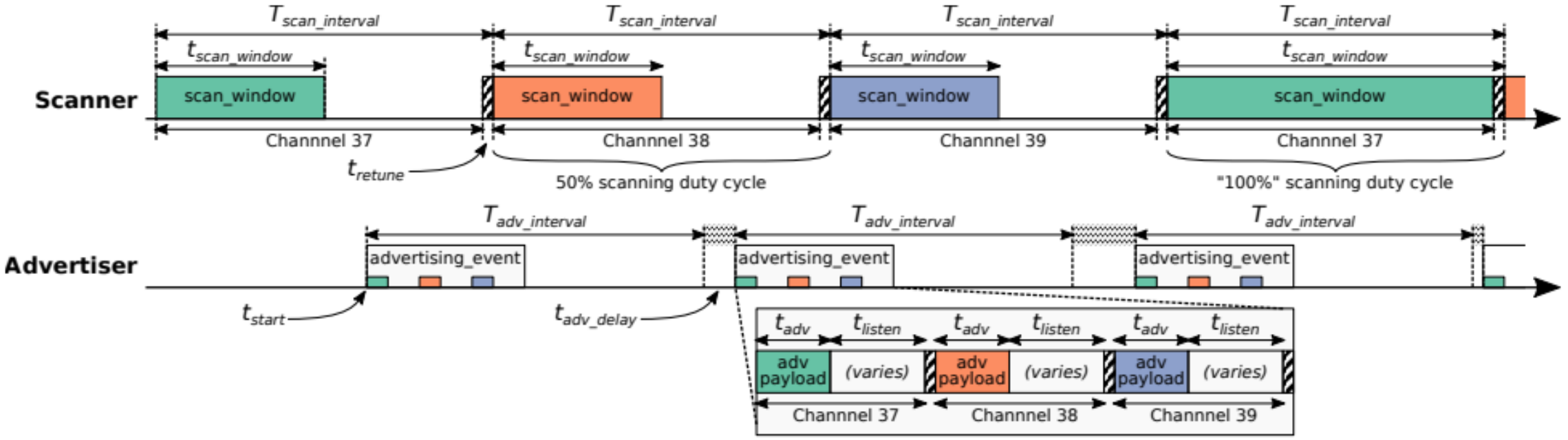


- Iterate through channels, listening for advertisements
  - $T_{scan\_interval}$ controls rate at which channels are changes
  - T_{scan_window} controls duty cycle of listening

- **Why listen at a low duty cycle?** **Save energy**

Putting it all together

- Advertisements are received when the channel of the scan window and the channel of the advertisement overlap in time (and space)



A warning about scanning expectations

- Scanners will NOT receive 100% of packets sent
  - Even ignoring range issues

- Packets are lost due to (in roughly descending order):
  - Duty cycle
  - Sharing 2.4 GHz antenna with WiFi
  - Retune period after each scanning interval
  - Dropped packets in the receive software
  - Packet collisions

Outline

- BLE Background
- BLE Layers
  - Physical Layer
  - Link Layer
- BLE roles
  - Advertising
  - Scanning