CSE 122/22C ; WES 269

BLE Foundations

Pat Pannuto, UC San Diego

ppannuto@ucsd.edu

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]

- More boots-on-the-ground view:
  - https://inst.eecs.berkeley.edu/~ee290c/sp18/lec/Lecture7A.pdf
    - From a team that has implemented BLE HW several times
  - https://download.ni.com/evaluation/rf/intro_to_bluetooth_test.pdf

Outline

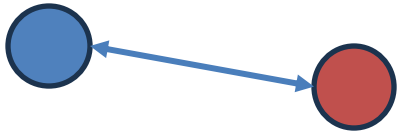
- **BLE Background**
- BLE Layers
  - Physical Layer
  - Link Layer
- BLE roles
  - Advertising
  - Scanning

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.

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
  - Primary/Secondary

BLE development

- 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 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

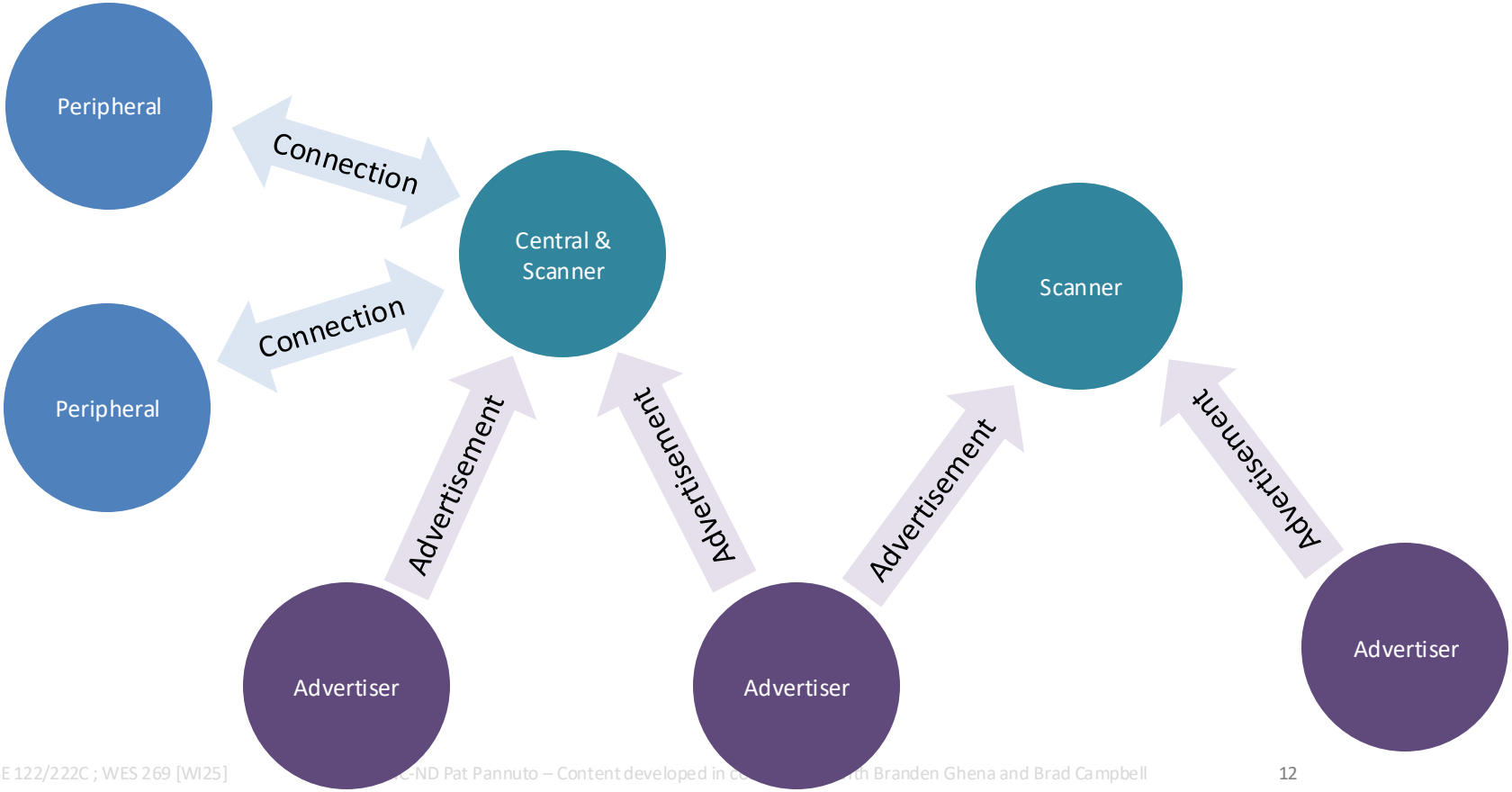
- Tip: be willing to just ignore things when skimming specs
  - 5.2 spec covers BLE and Bluetooth Classic and a bunch of upper layer stuff that we never have to care about

BLE 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

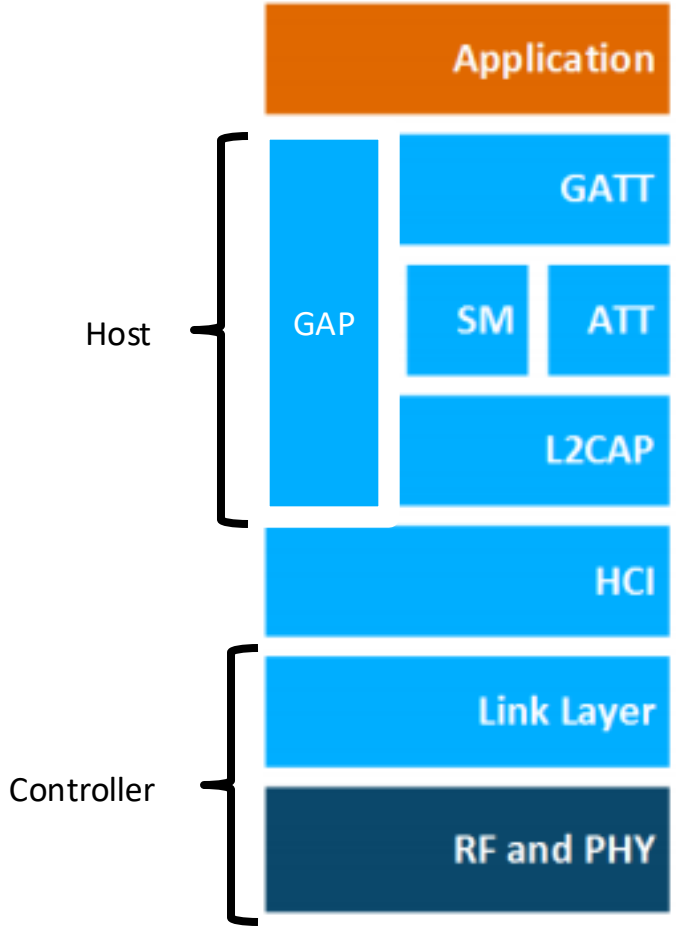
- Topology picture is a simplification of roles

- 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
    - What devices might do all these of these (at once!) semi-regularly?

- 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

BLE Layers

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



Break + Check your understanding

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

Break + Check your understanding

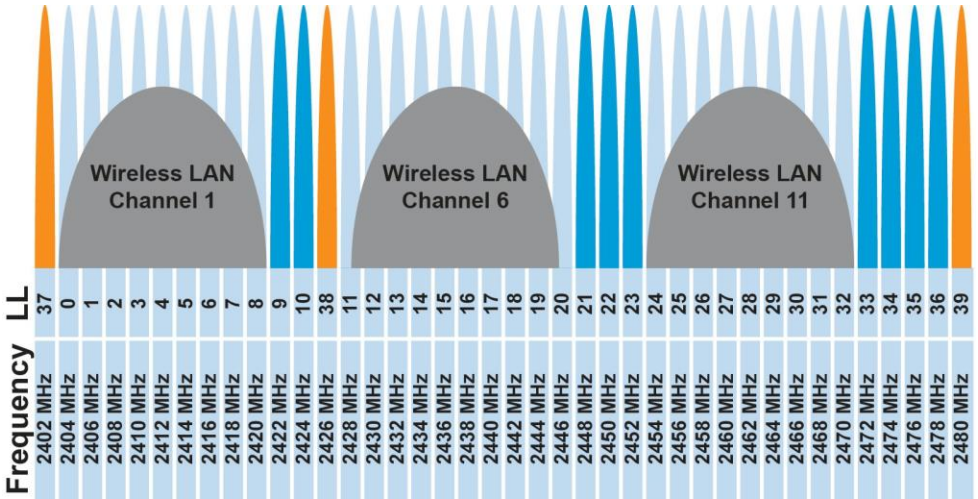
- Which roles is each device likely to have?
  - Keyboard
    - **Advertiser and Peripheral**
  - Laptop
    - **Scanner and Central**
  - Smartphone
    - **Advertiser, Peripheral, Scanner, and Central**

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)

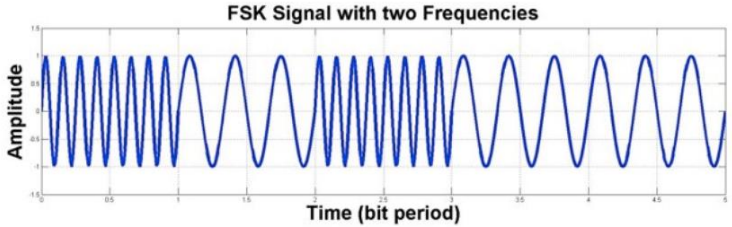


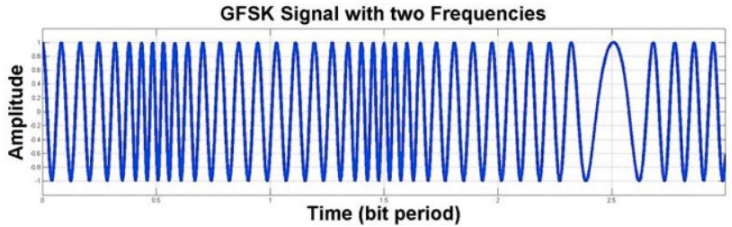
Why doesn't BLE avoid WiFi altogether?

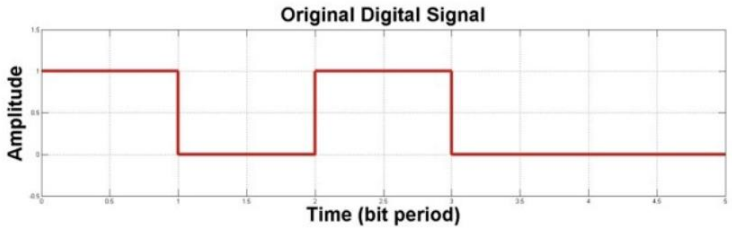
Can't on 2.4 GHz!

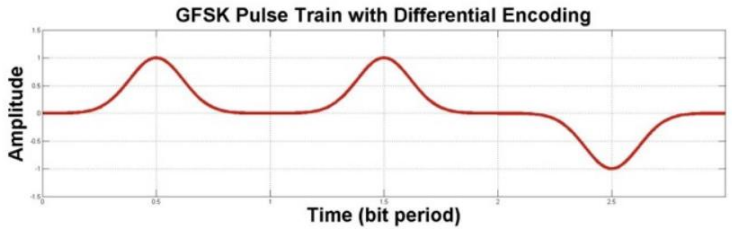
BLE modulation

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



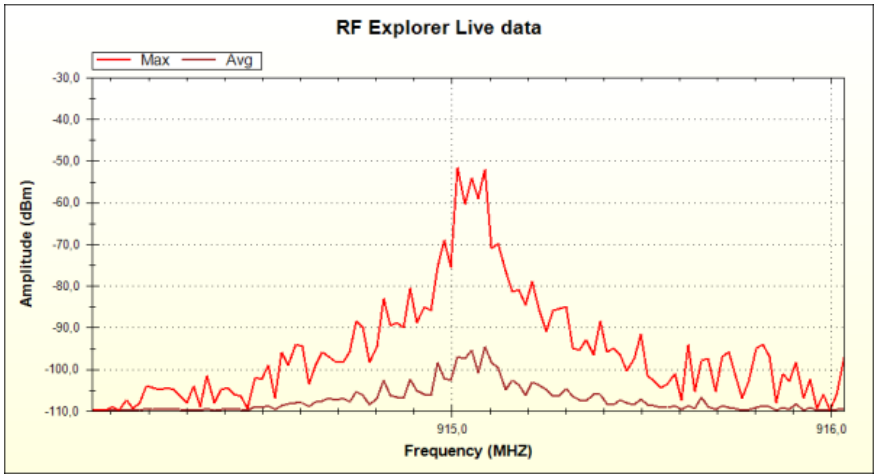




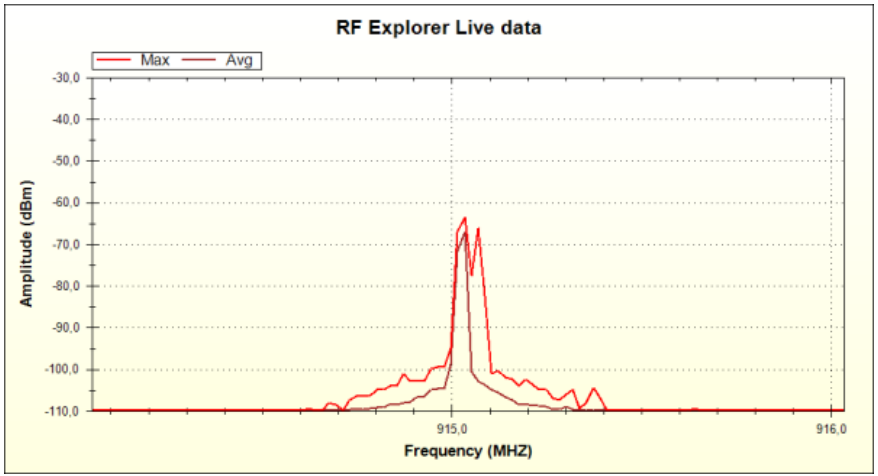


Gaussian FSK lessens spectral leakage at the expense of some loss in intersymbol discriminability

- Translation: GFSK reduces bandwidth at the cost of bit errors



FSK



GFSK

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

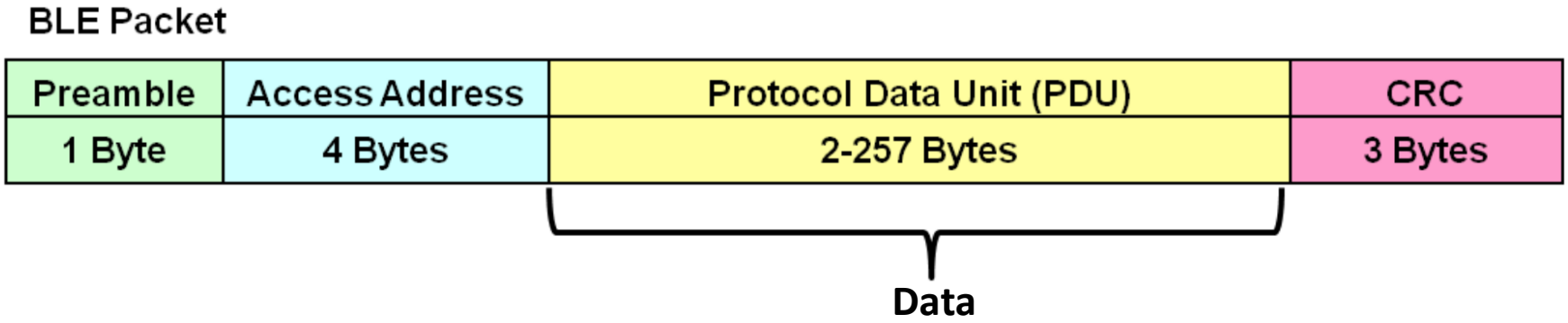
The range to a smart phone is typically 0-50 meters due to limited RF performance of the phones.

- 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



- Same packet structure for both advertisements and connections
  - Fields are filled in little endian (BLE is not the internet!)
- Advertisement packets use fixed Access Address: `0x8E89BED6`
- Established connections use a (randomly chosen) unique Access Address

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 layers being interwoven in wireless**

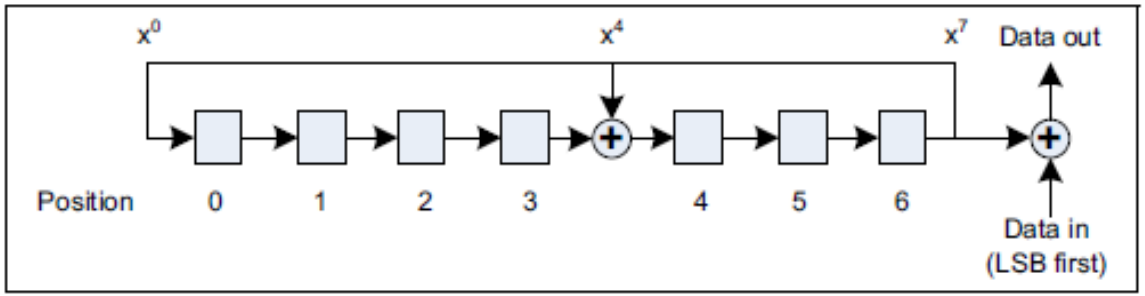
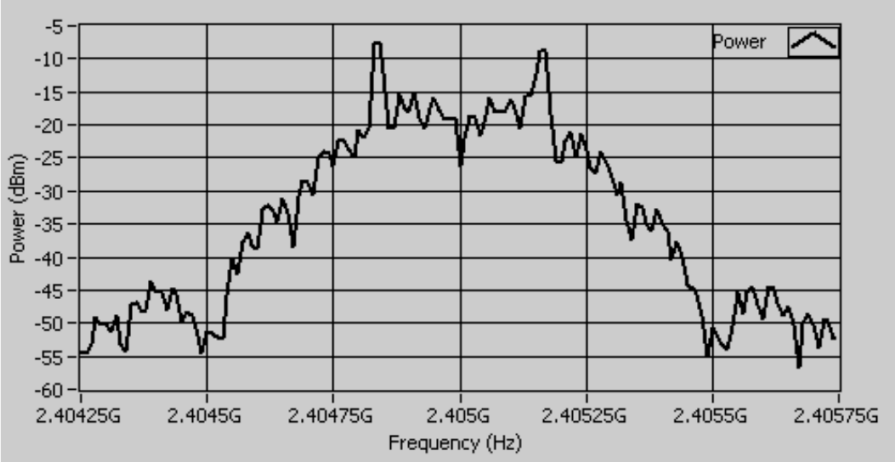


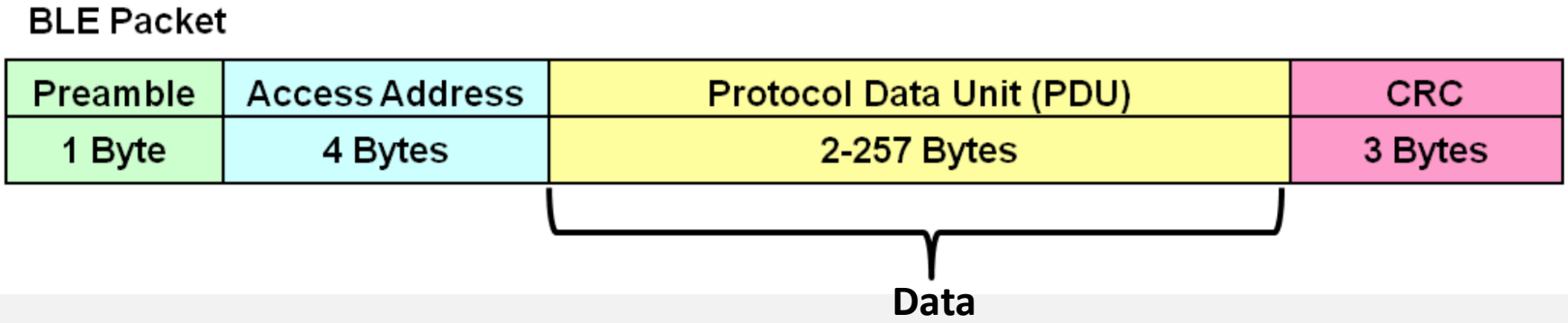
Figure 3.3: The LFSR circuit to generate data whitening

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)



Aside: Another example of PHY/MAC co-design



- Established **connections use a (randomly chosen) unique Access Address**

Not actually random…

Spec has rules regarding the bit pattern that “help” the short preamble “work well” (don’t look too much like noise / silence; sync-friendly; etc)

Bit processing pipeline

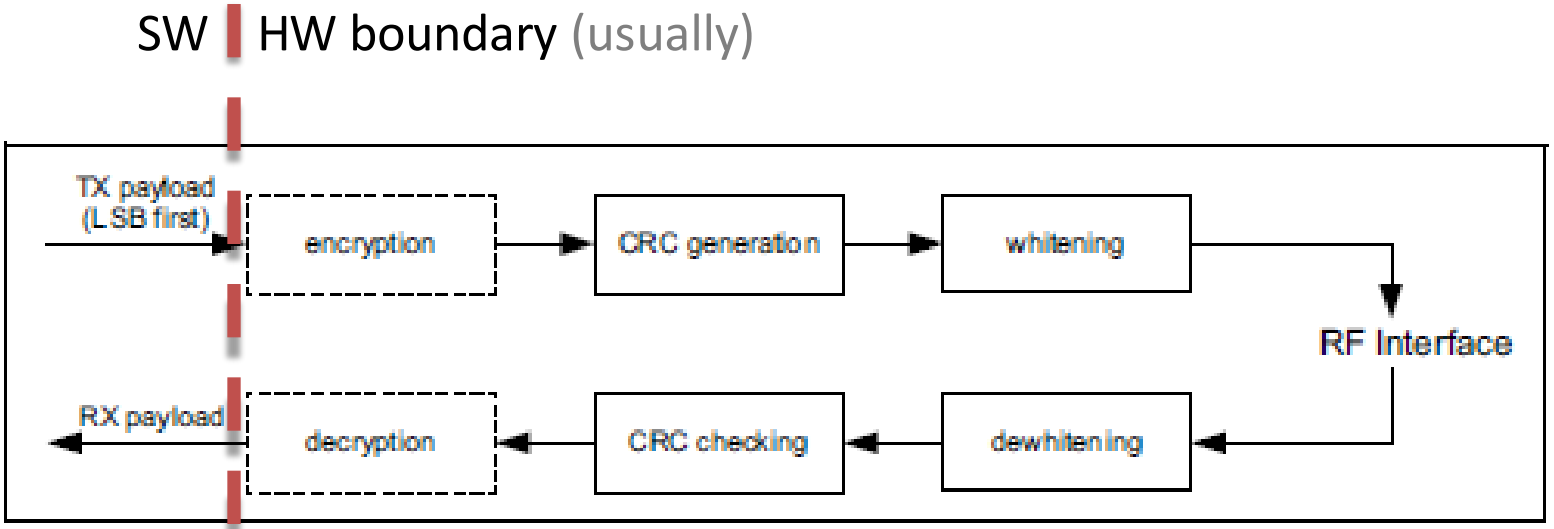


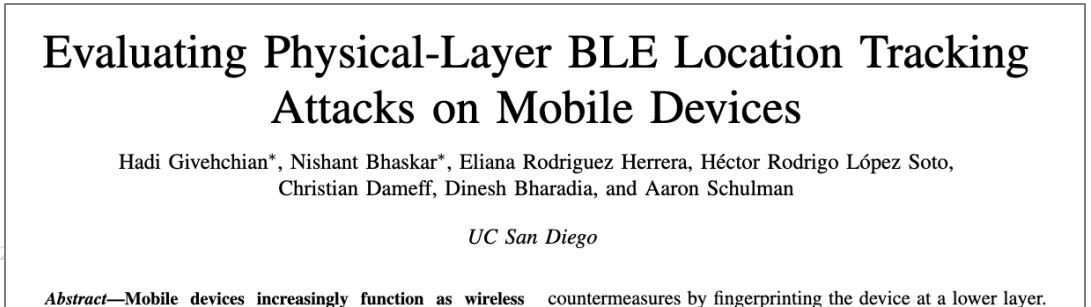
Figure 3.1: Payload bit processes for the LE Uncoded PHYs

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?
  - Link Layer…
    - Depends on how long a device uses the same address
    - Scan all the devices in my office … learn ‘my devices’
    - Scanners throughout the building could watch me move?
    - But if a device re-randomizes between two scanners, can’t follow it
      - Could probably detect this re-randomization though… [also, “a” device, in 2022?]
  - Physical Layer…
    -



Evaluating Physical-Layer BLE Location Tracking Attacks on Mobile Devices

Hadi Givehchian*, Nishant Bhaskar*, Eliana Rodriguez Herrera, Héctor Rodrigo López Soto, Christian Dameff, Dinesh Bharadia, and Aaron Schulman

UC San Diego

***Abstract*—Mobile devices increasingly function as wireless** countermeasures by fingerprinting the device at a lower layer.

Outline

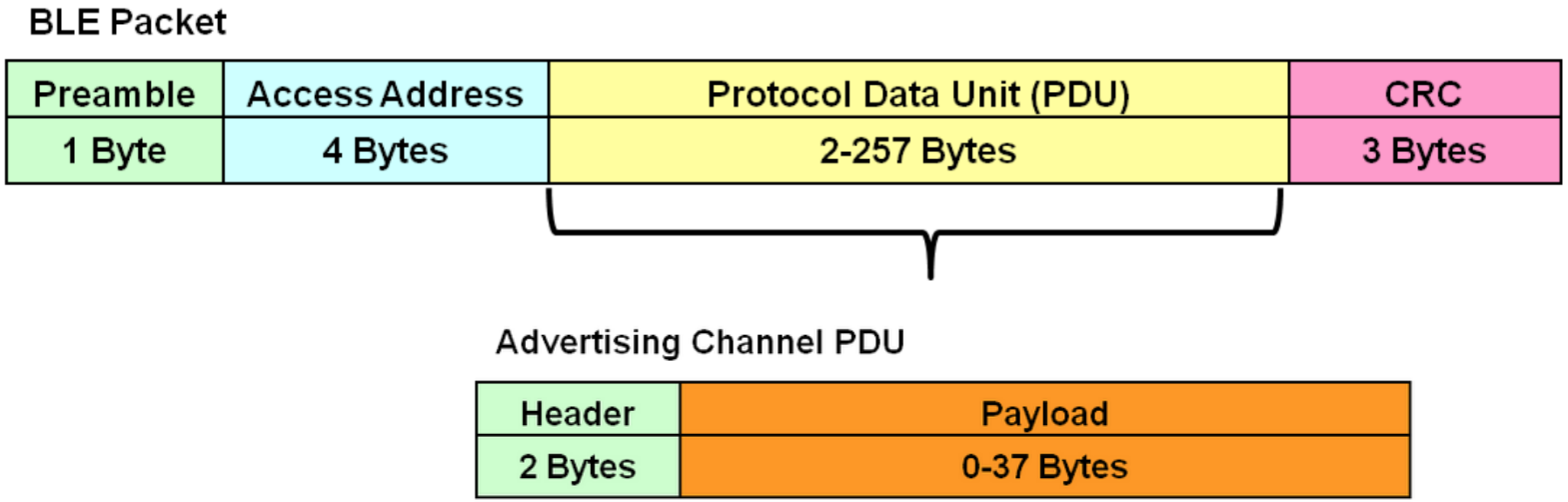
- BLE Background
- BLE Layers
  - Physical Layer
  - Link Layer
- **BLE roles**
  - **Advertising**
  - Scanning

Advertising

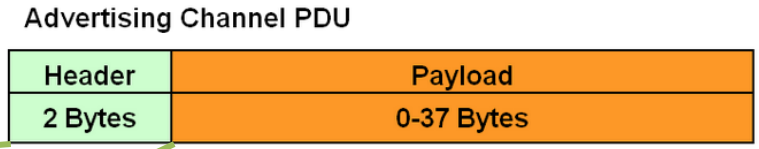
- BLE discovery mechanism
  - Make nearby devices aware of advertiser's existence
  - Communicate some information from or about advertiser
  - Traditional purpose is to enable connections, but this is also useful for general communication

- 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



BLE advertising header



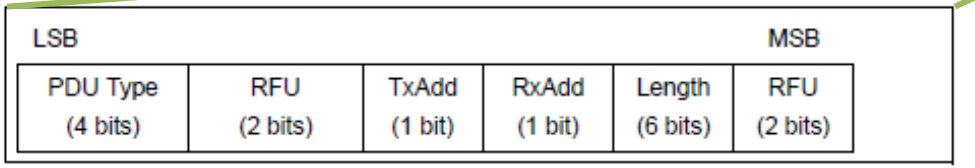


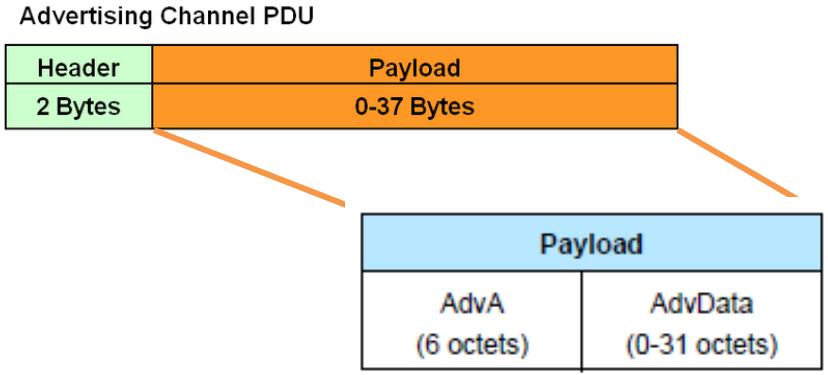
Figure 2.3: Advertising channel PDU Header

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

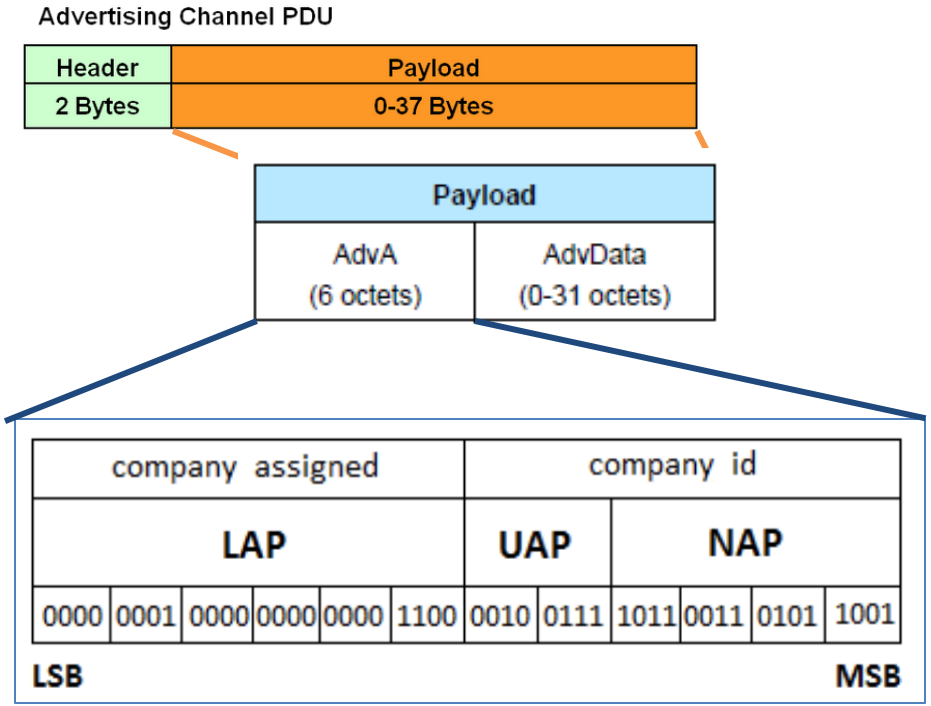


- **AdvA** – address of the advertiser
- Remaining up to 31 bytes are available for use

Advertiser 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

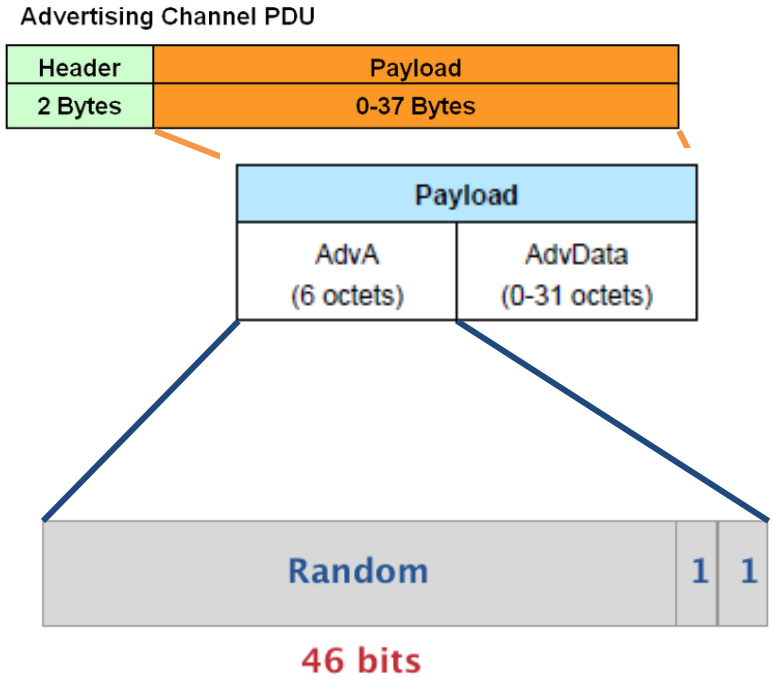
- **Why have the two types?**



Advertiser device addresses

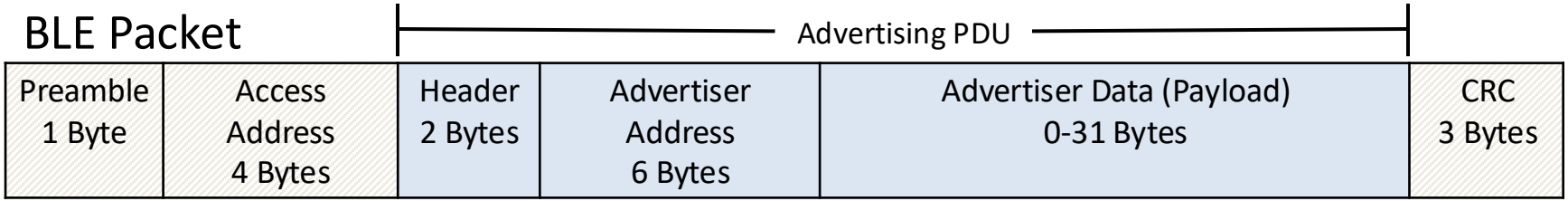
- Public and private address forms
- Public
  - 48 bits: 24-bits of company ID, 24-bits of company assigned number
  - Literally the same MAC address scheme as Ethernet and WiFi
- Private
  - Top two MSbs specify type
    - 46 bits of random
    - 46 bits of hash of an identity key

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



The full advertisement packet

- 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)

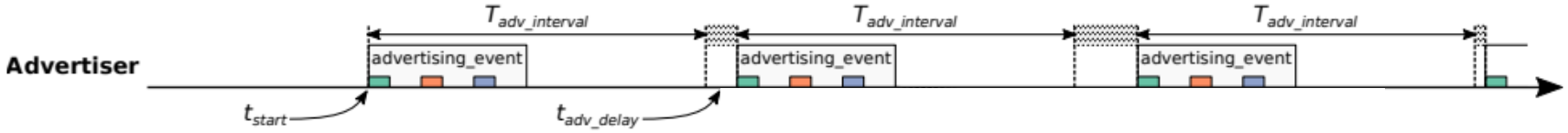
Figure 2.8: SCAN_REQ PDU Payload

- 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

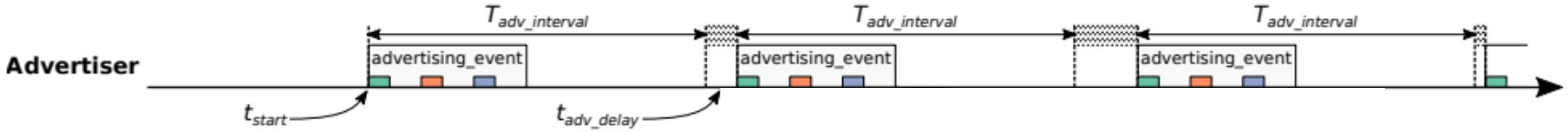
Advertising timing



- 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

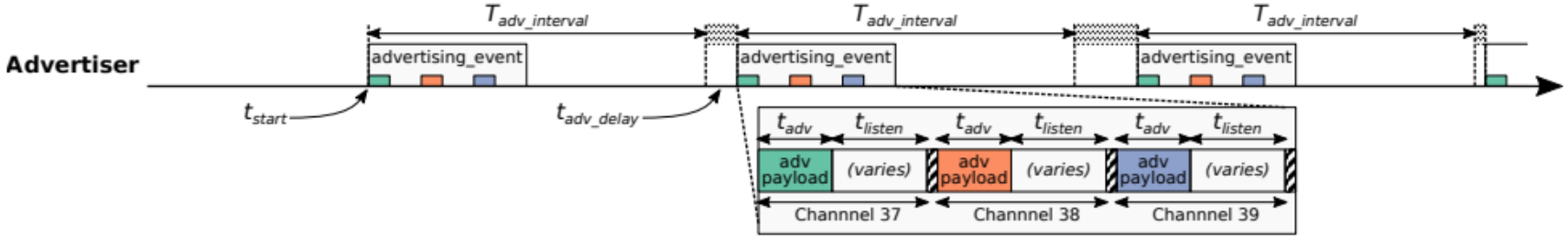
Advertising timing



- 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

- **Question: Advertising vs. Listening in BLE**
  - Which uses more energy…
    Listening for 1 second or transmitting 30 bytes three times?
    - Radio uses 60 mW when active (TX or RX)
    - 1 Mbps data rate
  - By how much?

Example: Maximum-sized transmission

- 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 between two device, 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

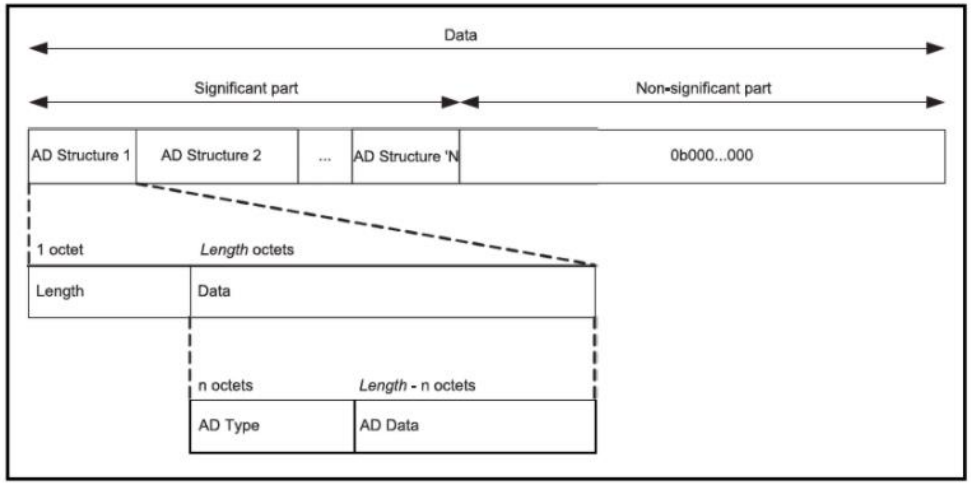


Figure 11.1: Advertising and Scan Response data format

Payload types

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

- Flags
- Name
- Service UUID
- TX Power Level
- Manufacturer-specific data
- And about twenty others

Payload types

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

- Flags
- Name
- Service UUID
- TX Power Level
- Manufacturer-specific data
- And about twenty others

1.3.1 Description

The Flags data type contains one bit Boolean flags. The Flags data type shall be included when any of the Flag bits are non-zero and the advertising packet is connectable, otherwise the Flags data type may be omitted. All 0x00 octets after the last non-zero octet shall be omitted from the value transmitted.

Note: If the Flags AD type is not present in a non-connectable advertisement, the Flags should be considered as unknown and no assumptions should be made by the scanner.

Payload types

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

- Flags
- Name
- Service UUID
- TX Power Level
- Manufacturer-specific data
- And about twenty others

1.3.2 Format

The Flags field may be zero or more octets long. This allows the Flags field to be extended while using the minimum number of octets within the data packet.

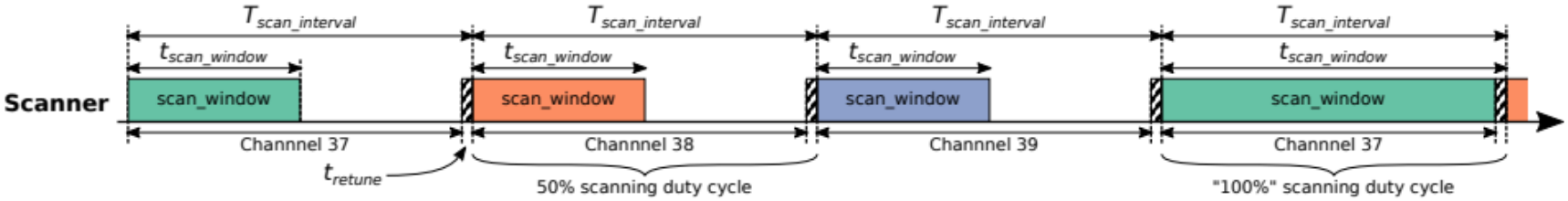
Data Type	Octet	Bit	Description
«Flags»	0	0	LE Limited Discoverable Mode
	0	1	LE General Discoverable Mode
	0	2	BR/EDR Not Supported. Bit 37 of LMP Feature Mask Definitions (Page 0)
	0	3	Simultaneous LE and BR/EDR to Same Device Capable (Controller). Bit 49 of LMP Feature Mask Definitions (Page 0)
	0	4	Simultaneous LE and BR/EDR to Same Device Capable (Host). Bit 66 of LMP Feature Mask Definitions (Page 1)
	0	5..7	Reserved for future use

Table 1.4: Flags data types

Outline

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

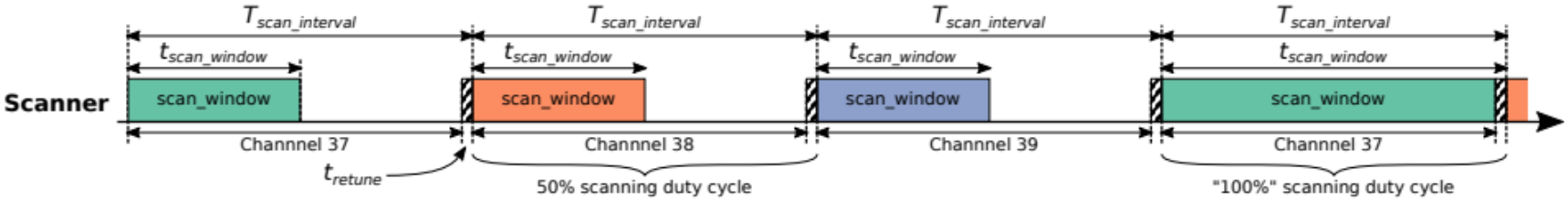
Scanning Pattern



- 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

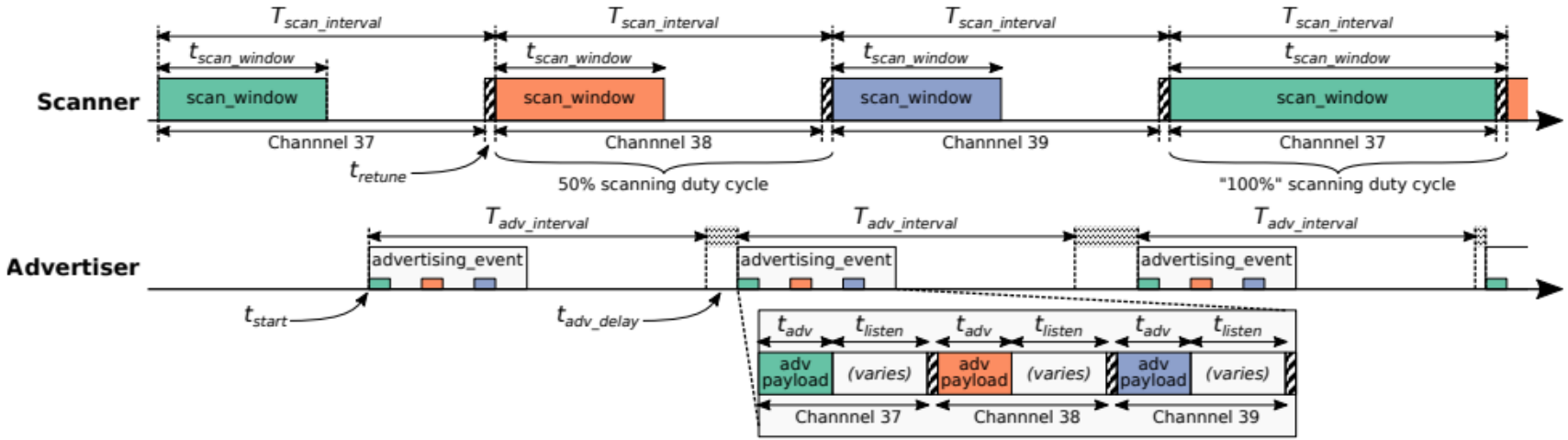


- Iterate through channels, listening for advertisements
  - $T_{scan\_interval}$ controls rate at which channels are changes
  - T_{scan_window} controls duty cycle of listening

- **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

Next Time: Going deeper on advertisements

For those wanting more on BLE energy use

Schrader, Raphael, et al. "Advertising power consumption of bluetooth low energy systems." 2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS). IEEE, 2016.

Administrivia

Active Assignments:

- Due: "Before {class/lab 2.i}" [~+ 2 days]
  - Post-Lab 1
  - Pre-Lab 2
    - **Note: Everyone must submit their own Pre-Lab**
      - *Collaboration??* Is okay, but it's in your own best interest that everyone in your group is comfortable with all of the pre-lab questions.
- Due: "Before {class/lab 3.i}" [~+ 2 weeks]
  - Post-Lab 2
    - **This is a group assignment, one copy / group**
    - **Check-offs must happen in-person — Effective deadline is last TA OH!**
  - [222C, 269 only]: BLE Paper Responses
  - [Released Mon, Jan 27]: Pre-Lab 3