UNIVERSIDAD
COMPLUTENSE
MADRID

Bluetooth Low Energy - 1

Networks and Protocols 1

Facultad de Informática

Bluetooth Standard

- Controlled by the Bluetooth SIG
  - Funded in 1998, currently 36000 members
- Technology for WPA networks
- Two independent stacks
  - *Traditional* bluetooth (BR/EDR)
  - Bluetooth low energy (BLE o smart)
    - Introduced in bluetooth 4.0
    - Previously known as Wibree, from Nokia
- Both stacks are incompatible
  - Many devices support both stacks

Bluetooth BR/EDR (*traditional*)

- Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR)
  - BR: 721kbps
  - BR/EDR: 2.1 Mbps
  - 802.11 AMP: 54Mbps
- Is *connection oriented:* devices establish a connection before sending data to each other
- Designed for specific applications
  - Audio transmission, phone, etc
- Has low power modes to extend battery life
- Maximum current about 25 mA
  - This current, although lower than other technologies like wifi, is not low enough for battery operated devices or energy harvesting systems

Bluetooth Low Energy

- New radio technology, open standard, designed for short reach and low power consumption
  - Small packets
  - Small RX and TX windows
  - Allows frequent radio power off
  - Can be used for battery operated devices
    - < 20mA maximum current
    - < 5 uA average current
- Low footprint (5.6 KB)
- Up to 1.4 Mbps and 1Km

BLE: typical use

23.2˚C

60.5 km/h

12:23 pm

Gate 10 BOARDING

3.2 kWh

PLAY >>

Network Available

- State publishing
  - Transferring small amount of data
  - A client can read the data at any moment
  - *Simple* interface (GATT)

BLE: general characteristics

Range:	~ 150 m without obstacles
Power (output):	~ 10 mW (10dBm)
Max current:	~ 15 mA
Latency:	3 ms
Bandwidth:	0.3 Mbit/s (application)
# Connections:	> 2 billion
Modulation:	GFSK @ 2.4 GHz
Reliability:	Adaptive Frequency Hopping, 24 bit CRC
Security:	128 bit AES CCM
Bias current:	~ 1µA
Topology	Star

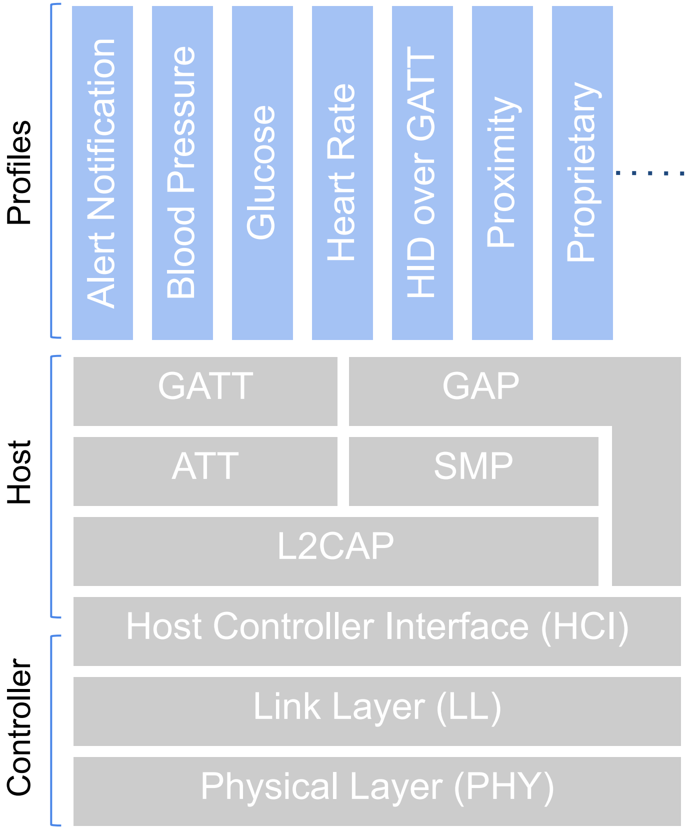
BLE Evolution

- 2010 - Bluetooth 4.0
- 2013 - Bluetooth 4.1
  - Concurrent Peripheral/Central
- 2014 - Bluetooth 4.2
  - LE Secure Connections
  - Data Length Extensions
- 2016 - Bluetooth 5
  - 2 Mbps
  - Long Range
  - Advertising Extensions
  - 10 -> 20 dBm max TX power
- 2017 - Bluetooth Mesh Profile
- 2019 - Bluetooth 5.1
  - Direction Finding
- 2020 - Bluetooth 5.2
  - Isochronous channels
  - LE Power Control
  - Enhanced Attribute Protocol
- Near future: LE Audio

BLE protocol stack

Profiles

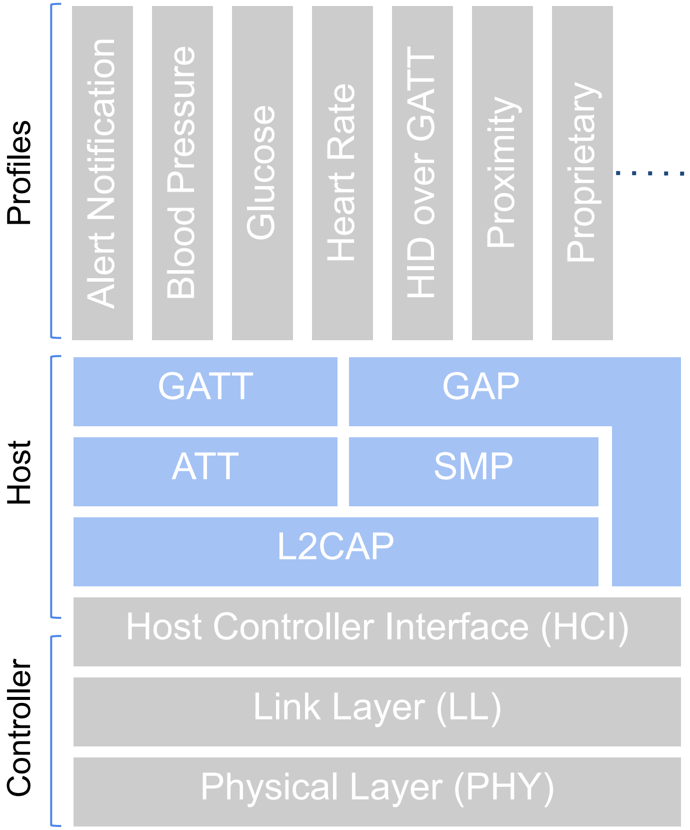
- Like the applications
- Define how the devices are going to communicate with each other, what will be their functionality, using
  - GAP roles, modes and procedures
  - GATT models and attribute interchange procedures
- Define the available data for interchange
- Standard and/or proprietary



BLE protocol stack

Host

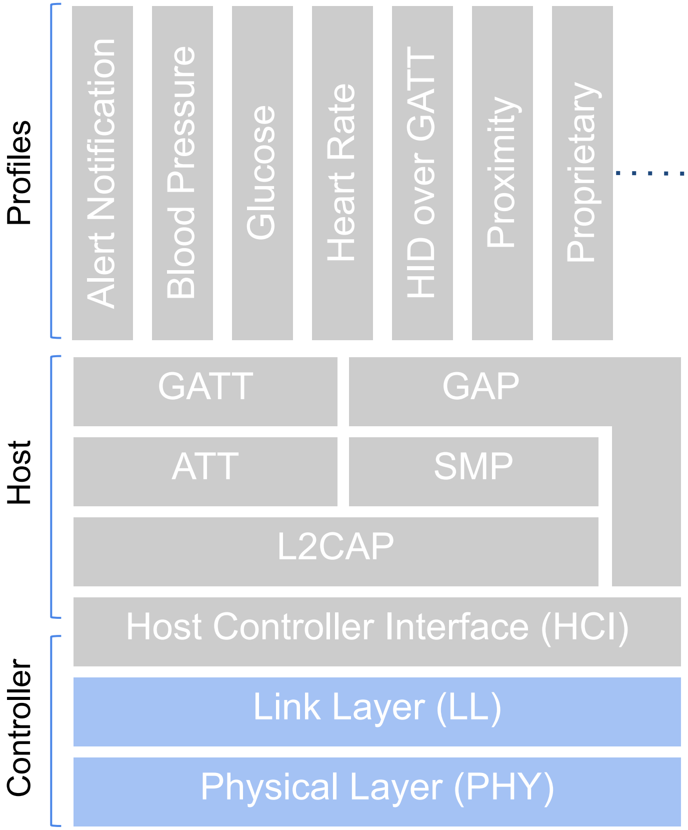
- High layers of the BLE protocol stack
- Logical Link Control and Adaptation Protocol (L2CAP)
  - Multiplexing layer
  - Fragmentation
  - Framing and data encapsulation
  - Error detection and correction
- Attribute Protocol (ATT)
  - Simple client-server model
  - Server serves attributes, clients can read them
- Security Manager Protocol (SMP)
  - Defines the authentication and encryption protocols and procedures
- Generic Attribute Profile (GATT)
  - Defines a hierarchical attribute structure
  - Offers services to discover and access server attributes, using the ATT protocol
- Generic Access Profile (GAP)
  - Defines the devices roles
  - Mechanisms for node discovering and connection establishment
  - Defines the security modes and procedures



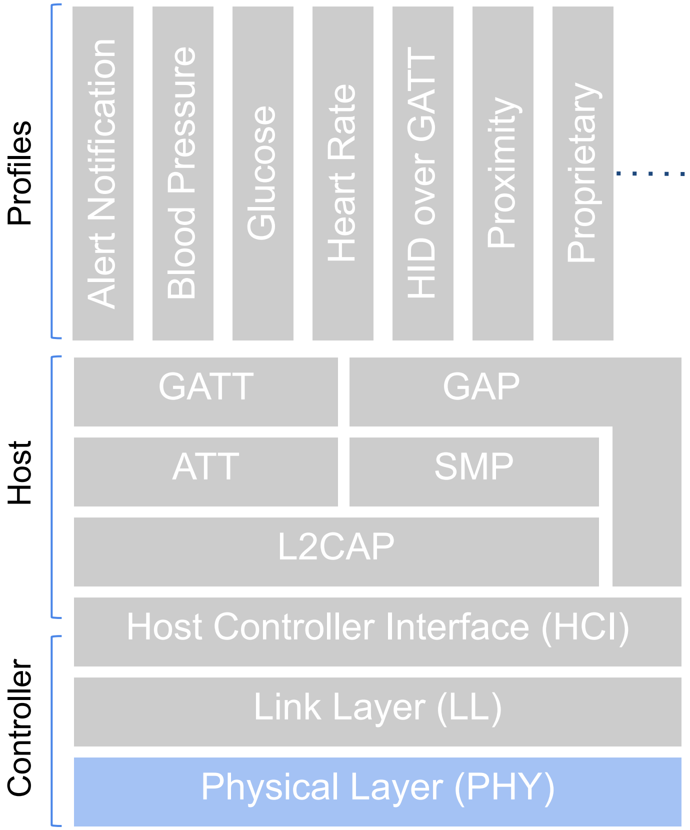
BLE protocol stack

Controller

- Pysical Layer (PHY)
  - Defines the way bits are transferred
  - Modulation, bands, transmission modes, rates
- Link Layer (LL)
  - States for the link control
  - Device addressing
  - Frame formats



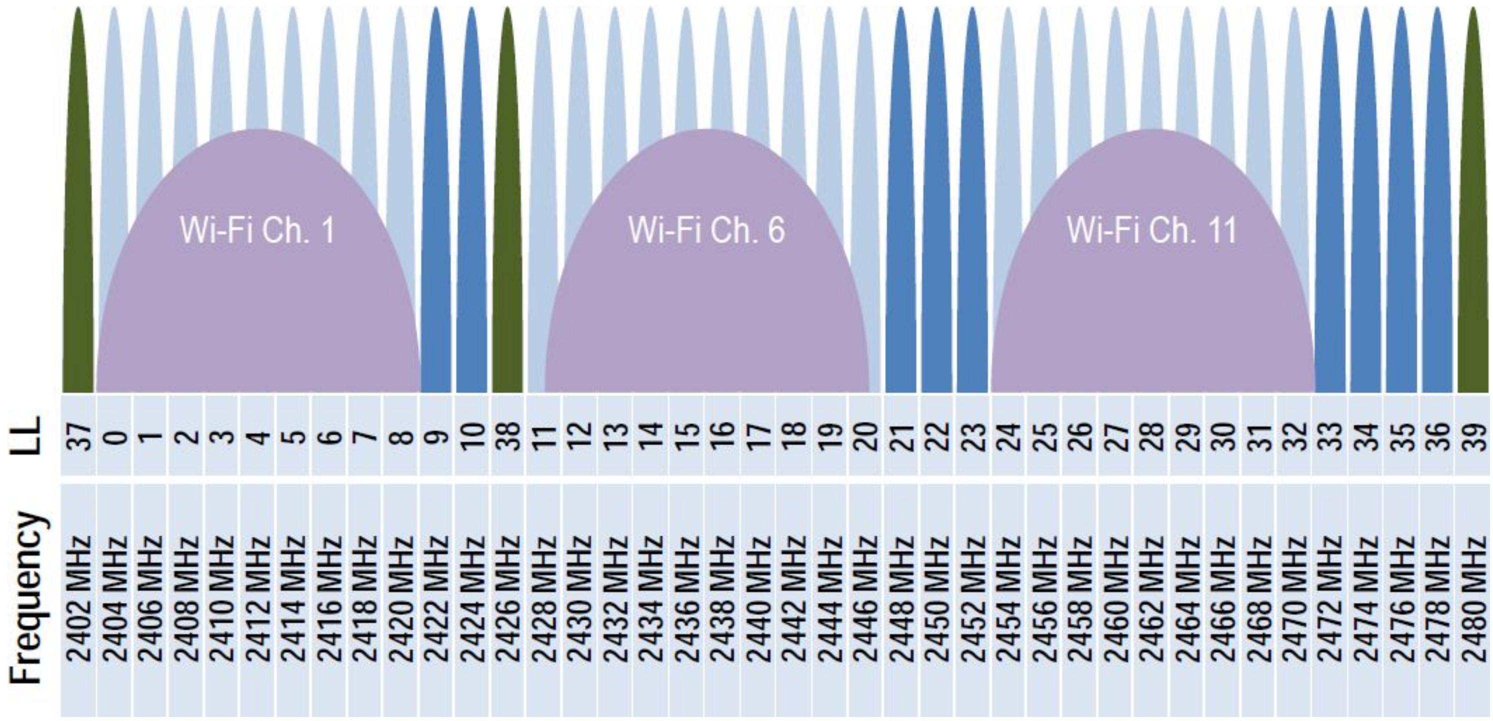
PHY layer


Profiles
Alert Notification
Blood Pressure
Glucose
Heart Rate
HID over GATT
Proximity
Proprietary
.....
Host
GATT
GAP
ATT
SMP
L2CAP
Host Controller Interface (HCI)
Controller
Link Layer (LL)
Physical Layer (PHY)

Physical layer (PHY)

- 2.4 GHz ISM band
- 40 Channels with 2 MHz spacing
  - 3 Advertisement channels
  - 37 Data/Secondary Advertisement channels

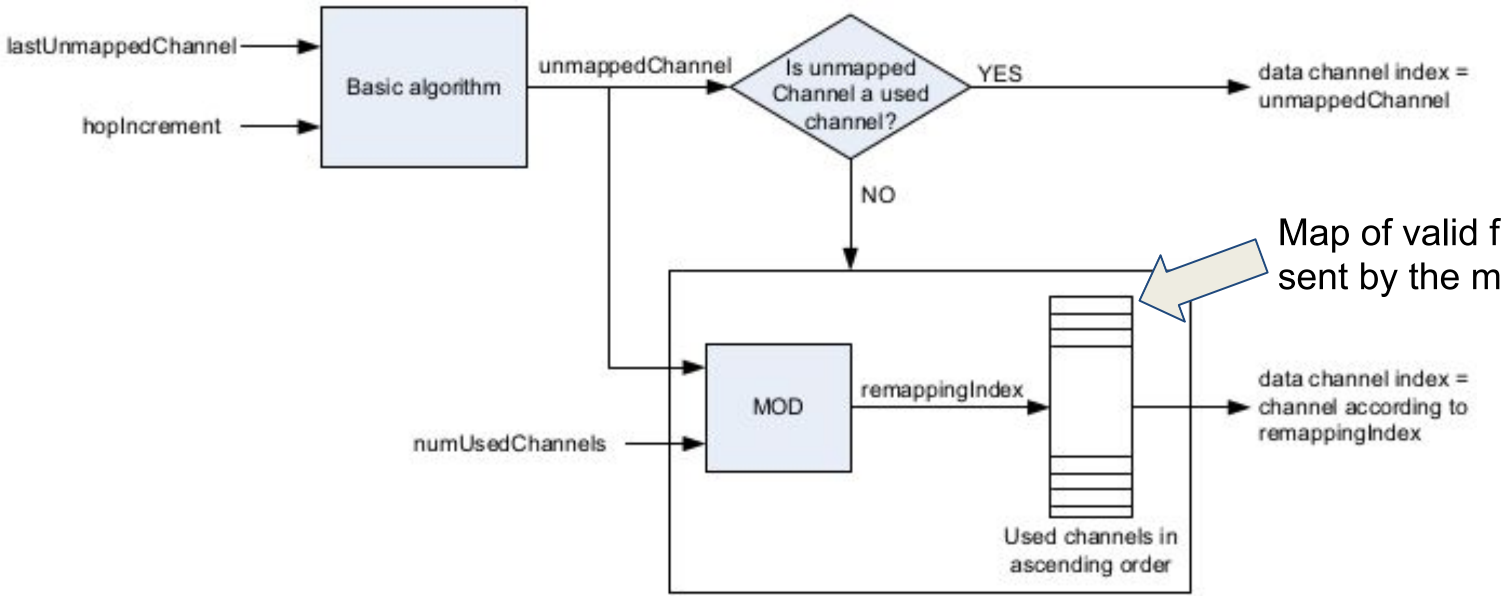
- Max TX power of 20dBm
- Modulation GFSK
  - 1 Mbps
  - 2 Mbps (from BLE 5.0)
  - S=2,8 -> 500kbps, 125 kbps



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

Physical layer (PHY)

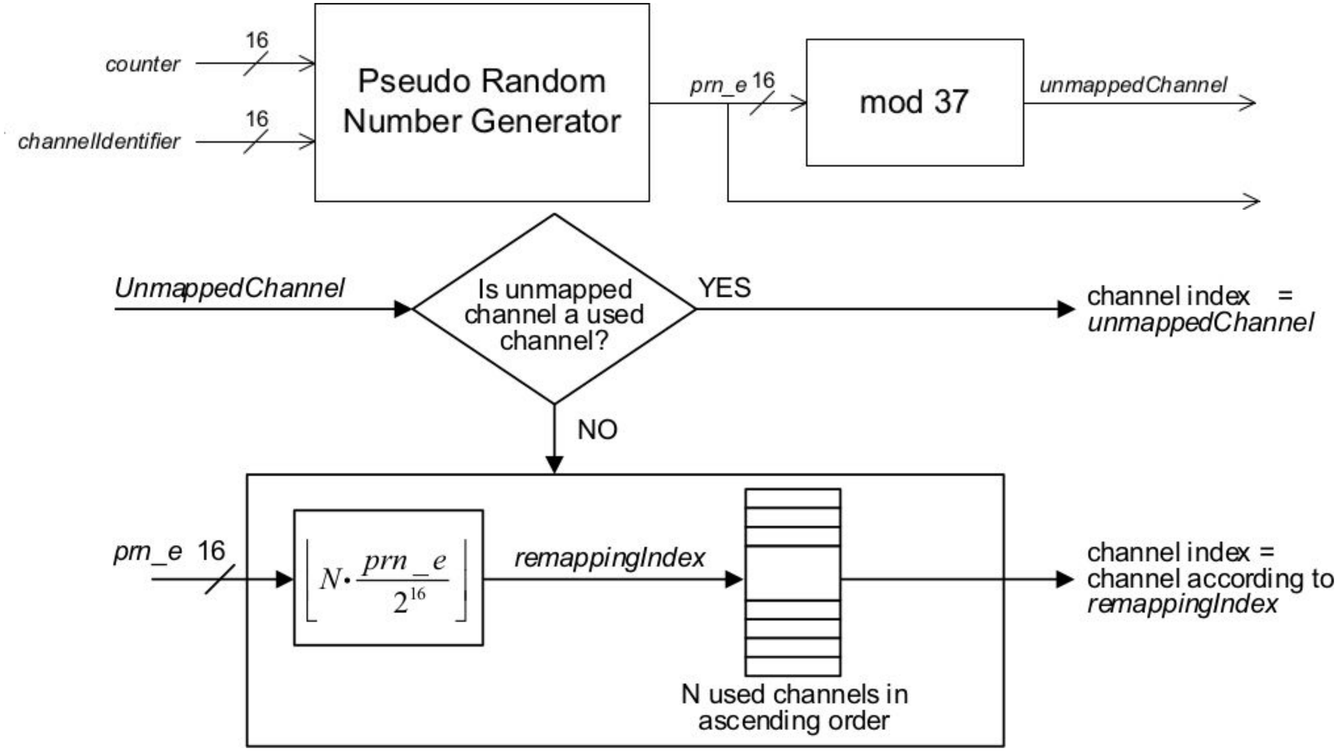
- *FHSS* in connections
  - The master sends a map of valid frequencies in the connection setup
    - Bad/Noisy channels are not included in the map
  - Two algorithms for frequency selection (Vol 6, Part B, 4.5.8)
    - Alg #1, basic algorithm:
      - unmappedChannel = (lastUnmappedChannel + hop_increment) mod 37


lastUnmappedChannel
hopIncrement
Basic algorithm
unmappedChannel
Is unmapped Channel a used channel?
YES
data channel index = unmappedChannel
NO
MOD
numUsedChannels
remappingIndex
data channel index = channel according to remappingIndex
Used channels in ascending order

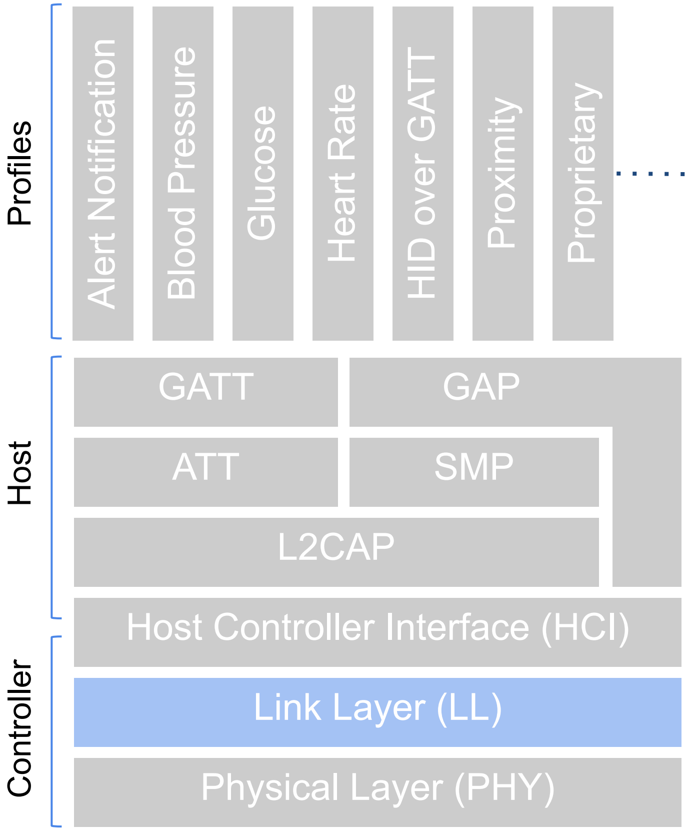
Map of valid frequencies sent by the master

Physical layer (PHY)

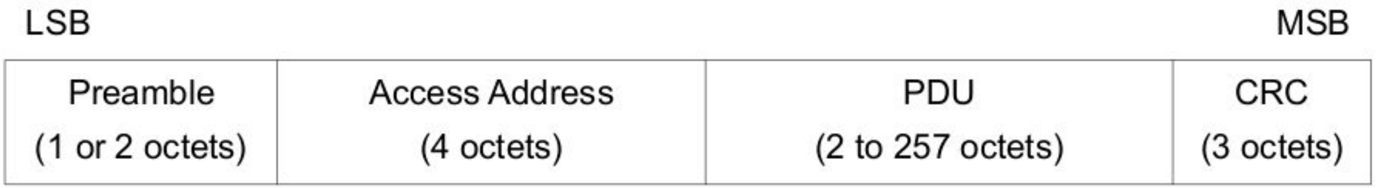
- *UFHSS* in connections
  - The master sends a map of valid frequencies in the connection setup
    - Bad/Noisy channels are not included in the map
  - Two algorithms for frequency selection (Vol 6, Part B, 4.5.8)
    - Alg #2, similar to #1 but with a pseudo random generator
      - The slave address is used as seed


counter
16
channelIdentifier
16
Pseudo Random Number Generator
prn_e 16
mod 37
unmappedChannel
UnmappedChannel
Is unmapped channel a used channel?
YES
channel index = unmappedChannel
NO
prn_e 16
N· prn_e / 2^16
remappingIndex
N used channels in ascending order
channel index = channel according to remappingIndex

Link Layer (LL)



Frame format (5.0): LE 1M y 2M

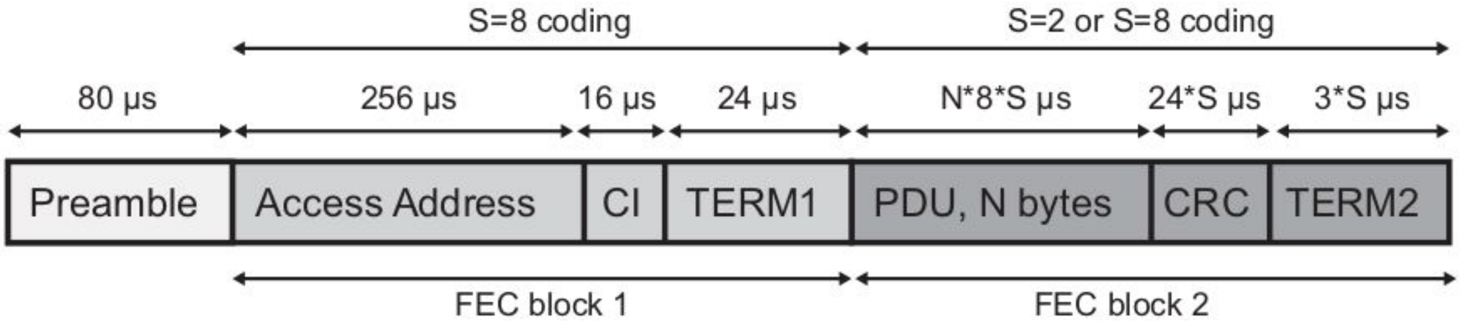


LSB … MSB

Preamble (1 or 2 octets)	Access Address (4 octets)	PDU (2 to 257 octets)	CRC (3 octets)

- Preamble
  - 1 byte for LE 1M and 2 bytes for LE 2M (same duration)
  - Frequency synchronization
  - Estimation of symbol duration
  - Automatic gain control
- Access Address
  - Fixed for advertisements (0x8E89BED6)
  - New for each connection or periodic advertisement
- PDU:
  - The internal format depends on the type of frame and channel
- CRC de 24 bits

Frame format (5.0): LE Coded

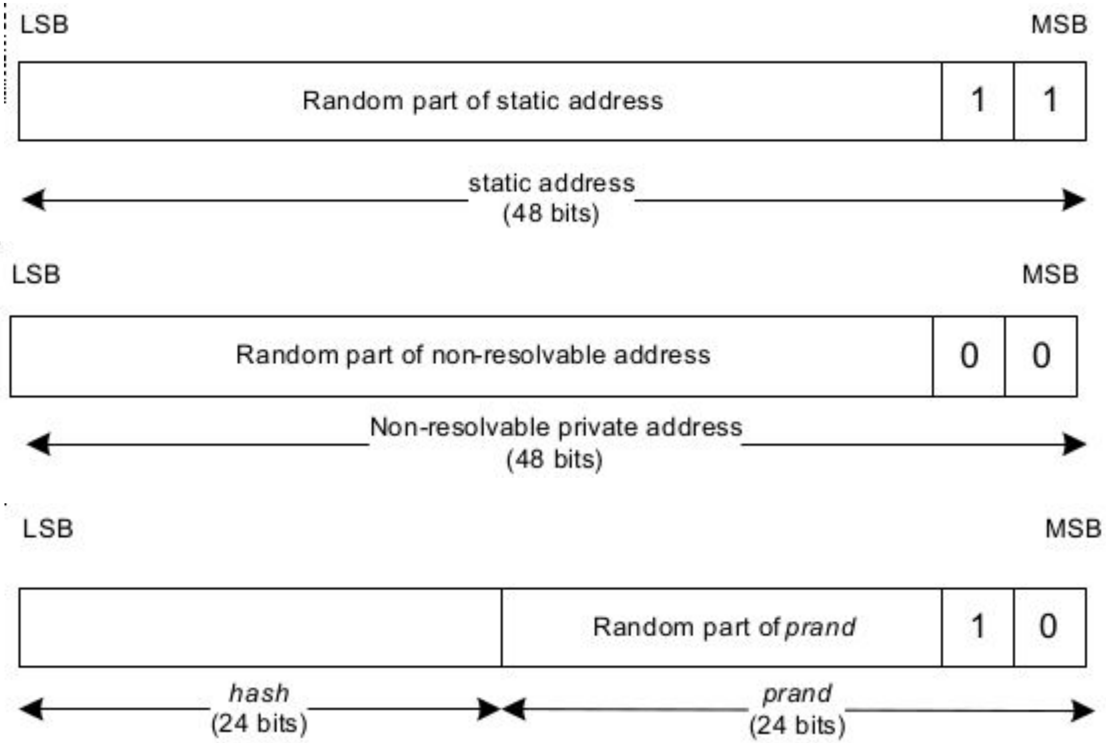


	Fields						
	Preamble	Access Address	CI	TERM1	PDU	CRC	TERM2
Number of Bits	Uncoded	32	2	3	16 – 2056	24	3
Duration when using S=8 coding (µs)	80	256	16	24	128 – 16448	192	24
Duration when using S=2 coding (µs)	80	256	16	24	32 – 4112	48	6

Devices addresses

- 6 bytes, IEEE format
  - It is not the same as the *Access Address*
  - Sent as part of the payload (PDU)
- Public
  - Registered with IEEE
- Random
  - Static
  - Private unsolvable
  - Private solvable
    - hash + random num

hash = ah (IRK, prand)



LSB MSB

Random part of static address | 1 | 1

static address (48 bits)

LSB MSB

Random part of non-resolvable address | 0 | 0

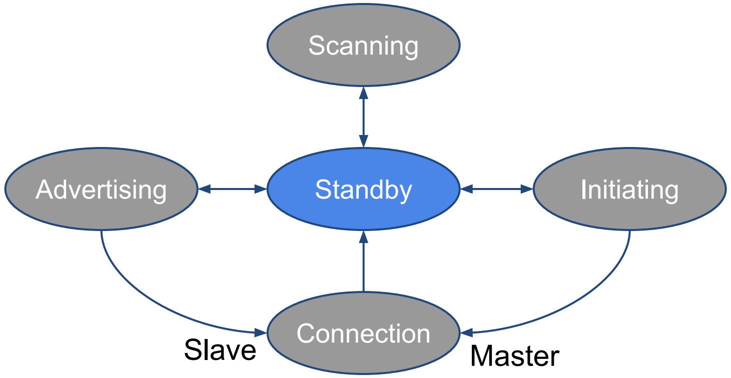
Non-resolvable private address (48 bits)

LSB MSB

| Random part of *prand* | 1 | 0

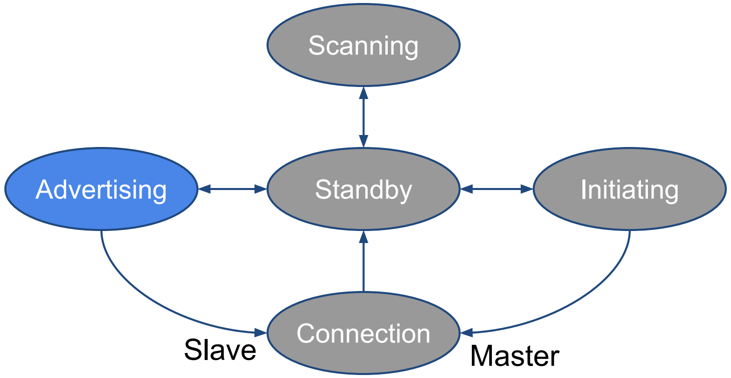
hash (24 bits) *prand* (24 bits)

Link Layer FSM



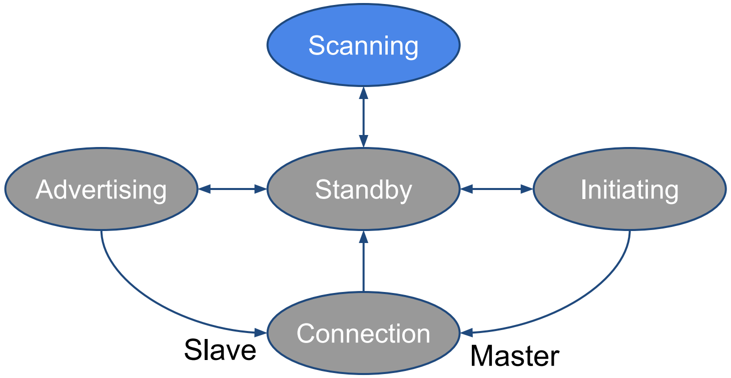
- Standby:
  - Initial state and standby
  - Radio is powered off
  - State changes only when an upper layer requests it

Link Layer FSM



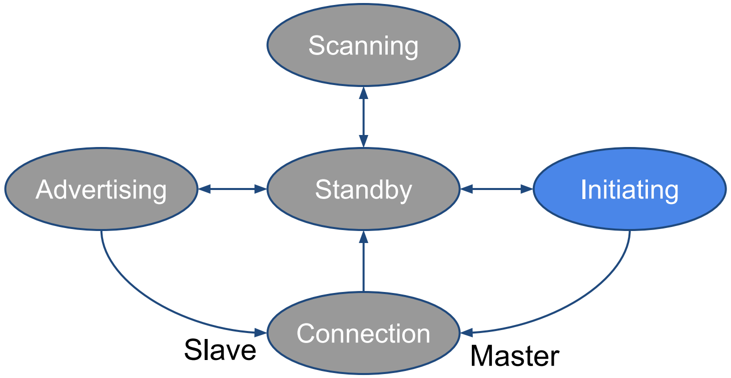
- Advertising - Advertiser
  - Advertising events repeated periodically
    - Advertising interval
  - Send advertising packets
    - Each advertising packet is sent to the three adv. channels
    - Transmit information about the advertising device
    - Can be *scannable,* the devices will respond to Scan Requests received on the same channel
    - Can be *connectable,* the devices is willing to accept connections and will respond to a Connection Request (acting after as the slave)

Link Layer FSM



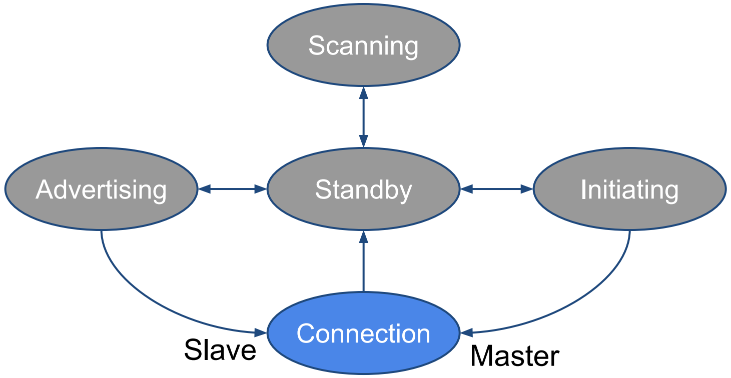
- Scanning - *Scanner*
  - Listens for advertisement packets sent on the adv. channels
  - Used to discover devices that are sending their advertisements
  - If a *scannable* advertisement is received, the device can send a Scan Request to obtain additional information on the same channel

Link Layer FSM



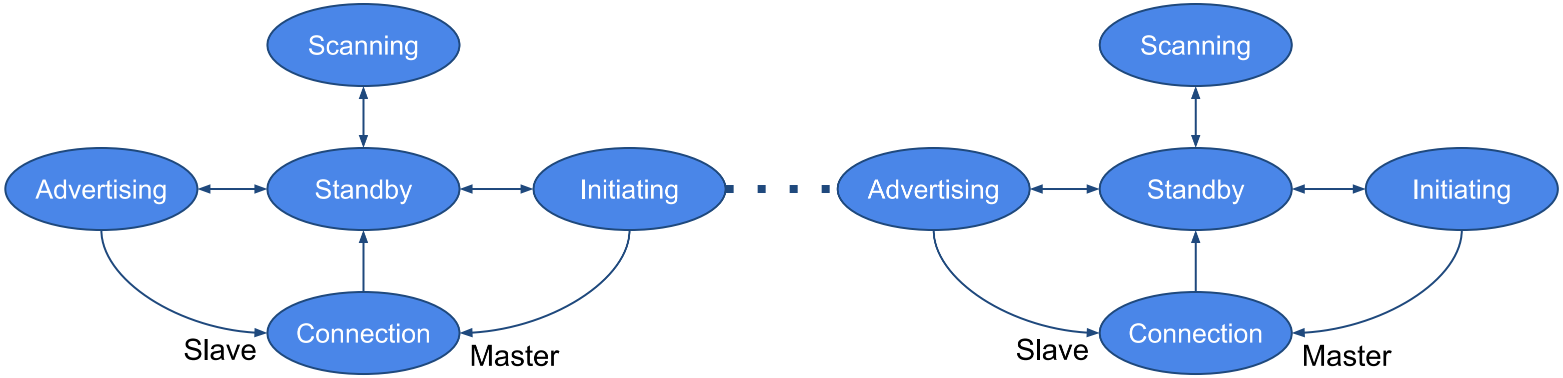
- Initiating - *Initiator*
  - Listens for *connectable* advertisements
  - Can initiate a connection sending a connection request on the same channel
    - It will then become the Master of the connection

Link Layer FSM



- Connection
  - When a connection request was received as a response to an advertisement
    - The device is the slave in the connection
  - When the device sended a connection request in response to a received advertisement
    - The device is the master in the connection
  - The master device can read attributes from the slave
  - The slave will respond to the requests of the master

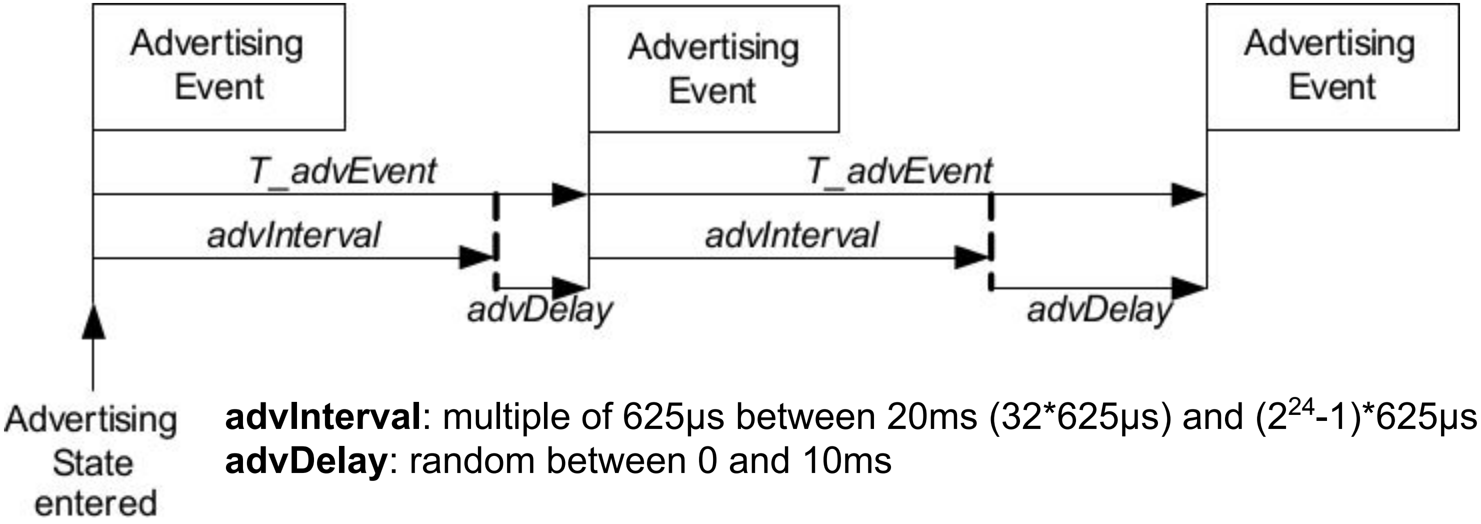
Multi-FSM



- Starting from Bluetooth 4.1 the LL supports multiple FSMs
  - Can maintain connections as master with some devices at the same time
  - At the same time can be advertiser, and accept new connections as slave
  - At the same time can be scanner, and request new connections as master with new discover devices

Advertising Events

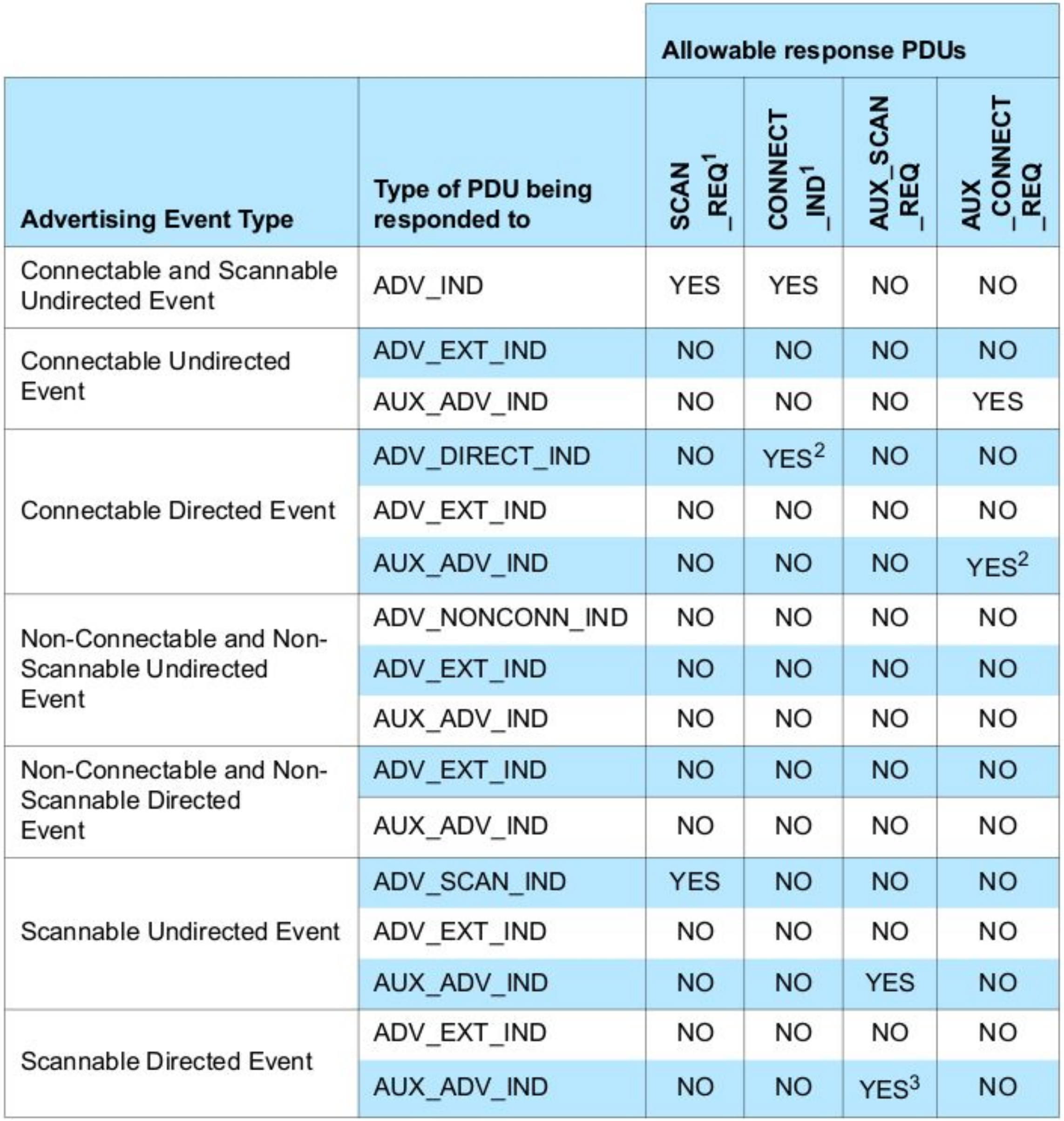
- The device sends advertising packets in *Advertising Events*
  - They repeat periodically



advInterval: multiple of 625μs between 20ms (32*625μs) and $(2^{24}$-1)*625μs (2.917h)
advDelay: random between 0 and 10ms

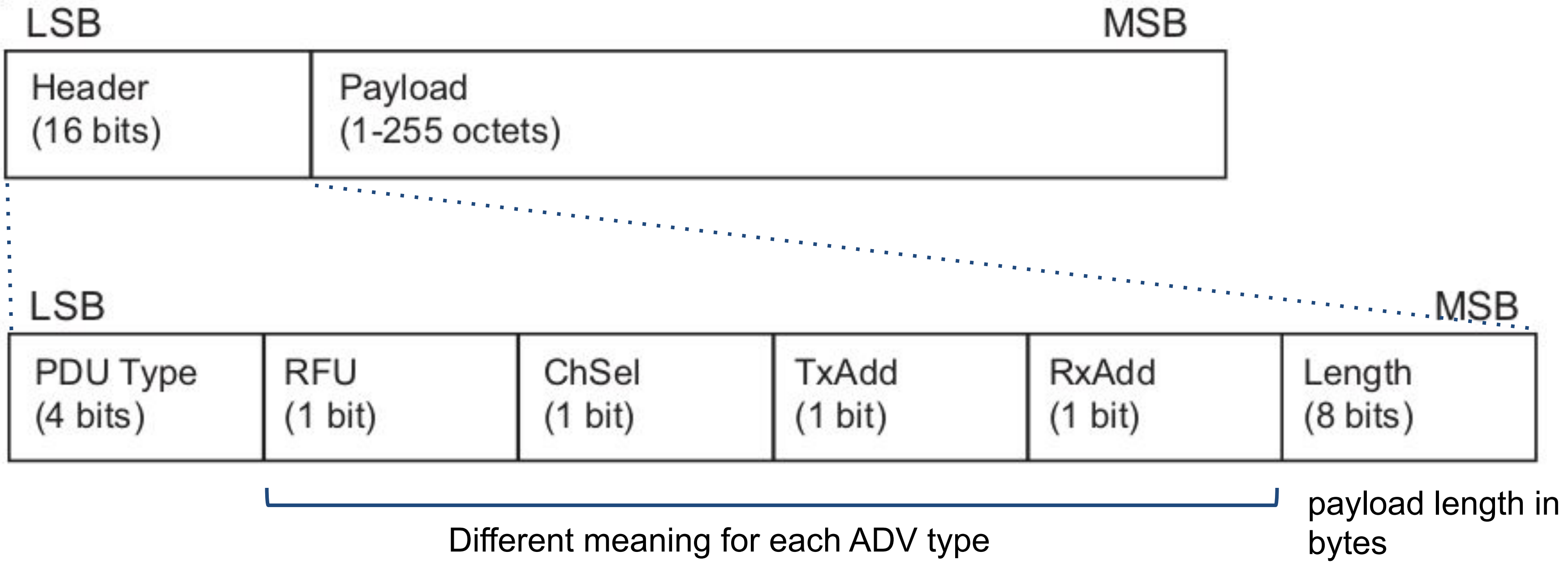
- One advertisement per event: channels 37-39
  - The advertiser can interrupt the event before its end
- Three classes: normal, extended y periodic
  - Several types in each class

Advertising Events Types



		Allowable response PDUs			
Advertising Event Type	Type of PDU being responded to	SCAN_REQ[1]	CONNECT_IND[1]	AUX_SCAN_REQ	AUX_CONNECT_REQ
Connectable and Scannable Undirected Event	ADV_IND	YES	YES	NO	NO
Connectable Undirected Event	ADV_EXT_IND	NO	NO	NO	NO
	AUX_ADV_IND	NO	NO	NO	YES
Connectable Directed Event	ADV_DIRECT_IND	NO	YES[2]	NO	NO
	ADV_EXT_IND	NO	NO	NO	NO
	AUX_ADV_IND	NO	NO	NO	YES[2]
Non-Connectable and Non-Scannable Undirected Event	ADV_NONCONN_IND	NO	NO	NO	NO
	ADV_EXT_IND	NO	NO	NO	NO
	AUX_ADV_IND	NO	NO	NO	NO
Non-Connectable and Non-Scannable Directed Event	ADV_EXT_IND	NO	NO	NO	NO
	AUX_ADV_IND	NO	NO	NO	NO
Scannable Undirected Event	ADV_SCAN_IND	YES	NO	NO	NO
	ADV_EXT_IND	NO	NO	NO	NO
	AUX_ADV_IND	NO	NO	YES	NO
Scannable Directed Event	ADV_EXT_IND	NO	NO	NO	NO
	AUX_ADV_IND	NO	NO	YES[3]	NO

PDU format: advertisements



LSB MSB

Header (16 bits)	Payload (1-255 octets)

LSB MSB

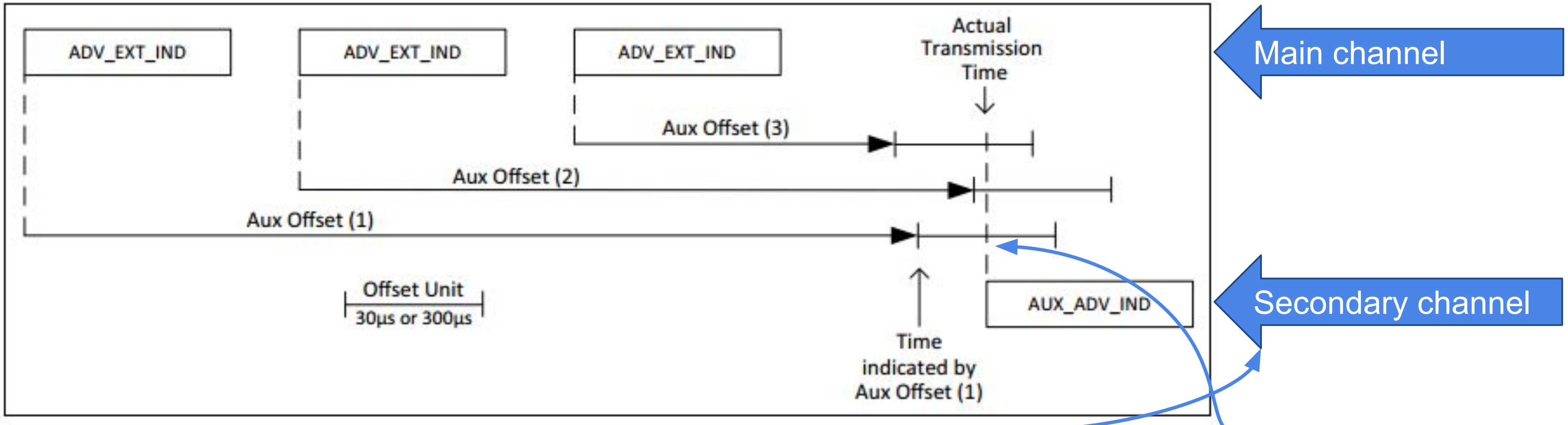
PDU Type (4 bits)	RFU (1 bit)	ChSel (1 bit)	TxAdd (1 bit)	RxAdd (1 bit)	Length (8 bits)

Different meaning for each ADV type

payload length in bytes

Type	TxAdd (0/1)	RxAdd (0/1)	Payload	
ADV_IND	pub/rand	-	Adv Address (6 bytes)	Adv Data (0-31 bytes)
ADV_DIRECT_IND	pub/rand	pub/rand	Adv Address (6 bytes)	Target Address (6 bytes)
ADV_NONCONN_IND	pub/rand	-	Adv Address (6 bytes)	Adv Data (0-31 bytes)
ADV_SCAN_IND	pub/rand	-	Adv Address (6 bytes)	Adv Data (0-31 bytes)

PDU format: extended advertisements



LSB MSB

Header	Payload
(16 bits)	(1-255 octets)

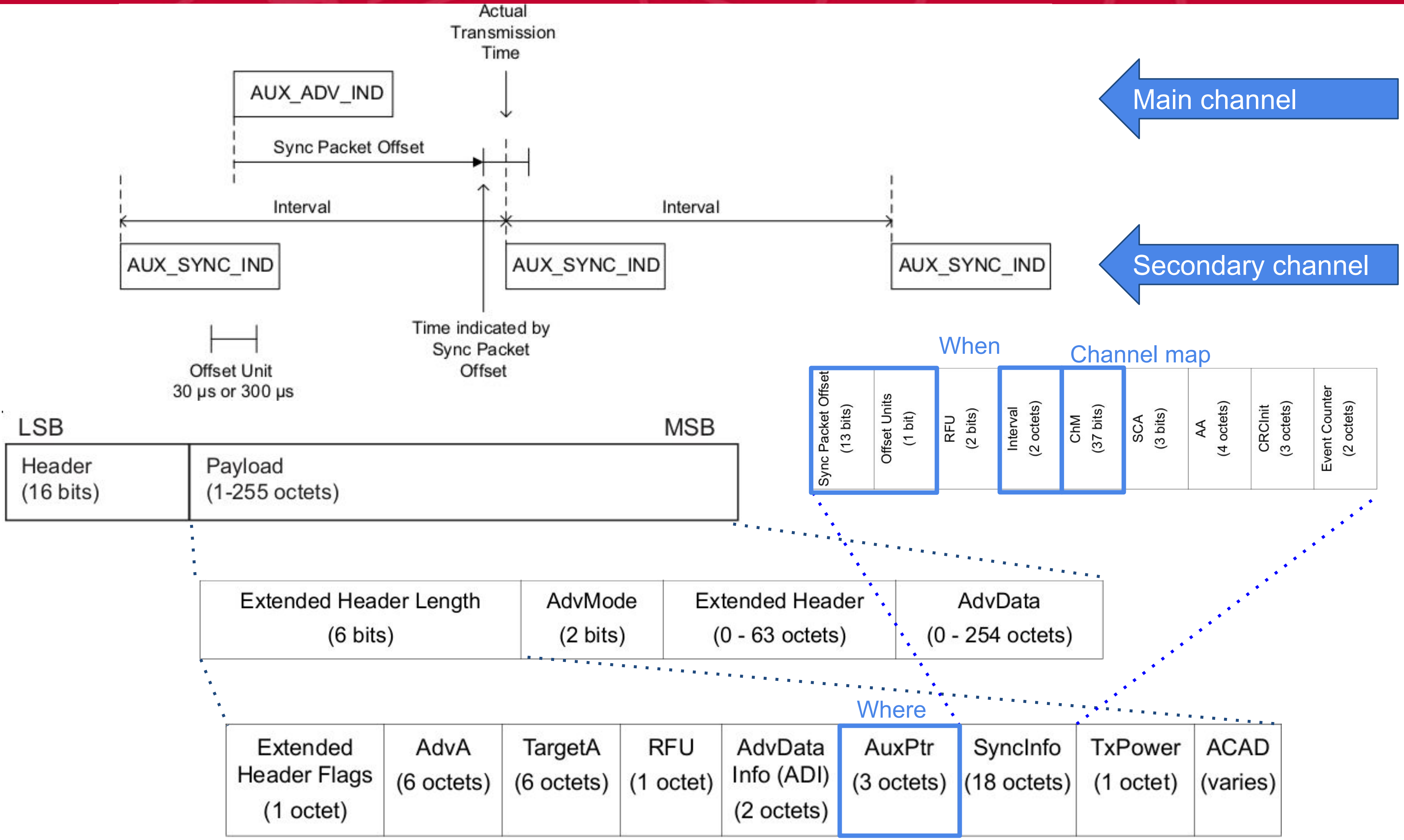
Where When

Channel Index	CA	Offset Units	AUX Offset	AUX PHY
(6 bits)	(1 bits)	(1 bits)	(13 bits)	(3 bits)

Extended Header Length	AdvMode	Extended Header	AdvData
(6 bits)	(2 bits)	(0 - 63 octets)	(0 - 254 octets)

Extended Header Flags	AdvA	TargetA	RFU	AdvData Info (ADI)	AuxPtr	SyncInfo	TxPower	ACAD
(1 octet)	(6 octets)	(6 octets)	(1 octet)	(2 octets)	(3 octets)	(18 octets)	(1 octet)	(varies)

PDU format: periodic advertisements



LSB	MSB
Header (16 bits)	Payload (1-255 octets)

Extended Header Length (6 bits)	AdvMode (2 bits)	Extended Header (0 - 63 octets)	AdvData (0 - 254 octets)

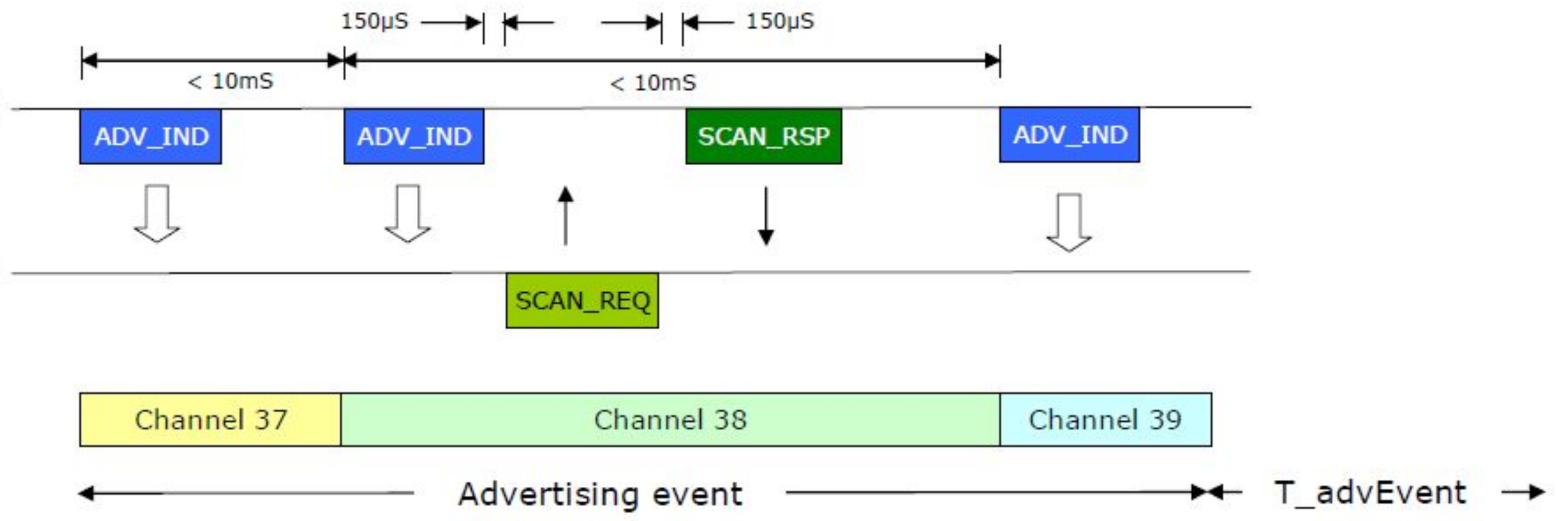
Extended Header Flags (1 octet)	AdvA (6 octets)	TargetA (6 octets)	RFU (1 octet)	AdvData Info (ADI) (2 octets)	AuxPtr (3 octets)	SyncInfo (18 octets)	TxPower (1 octet)	ACAD (varies)

Sync Packet Offset (13 bits)	Offset Units (1 bit)	RFU (2 bits)	Interval (2 octets)	ChM (37 bits)	SCA (3 bits)	AA (4 octets)	CRCInit (3 octets)	Event Counter (2 octets)

Scan Events

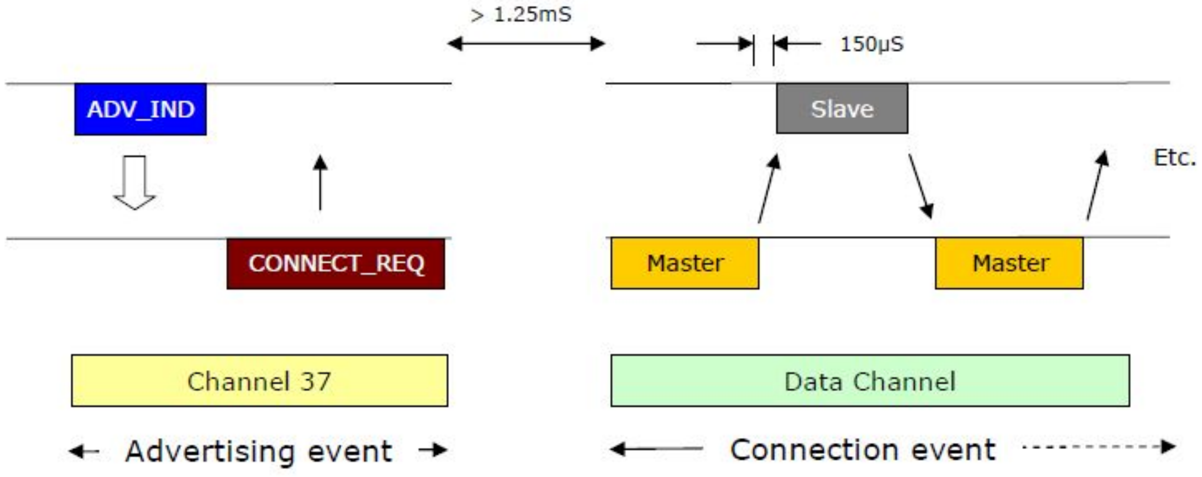
- In scanning state
- The device scans the advertising channels
  - *scanWindow*: time for which the channel is listened
  - *scanInterval*: between scanning events
  - Both <= 40.96s and *scanWindow* < *scanInterval*
- If an advertisement packet has AuxPtr, the device listens also in the announced secondary channel
- Two types
  - passive: only receives the advertisements
  - active: can send connection requests if it receives connectable messages or scan request for scannable messages

Scan Request - Scan Response



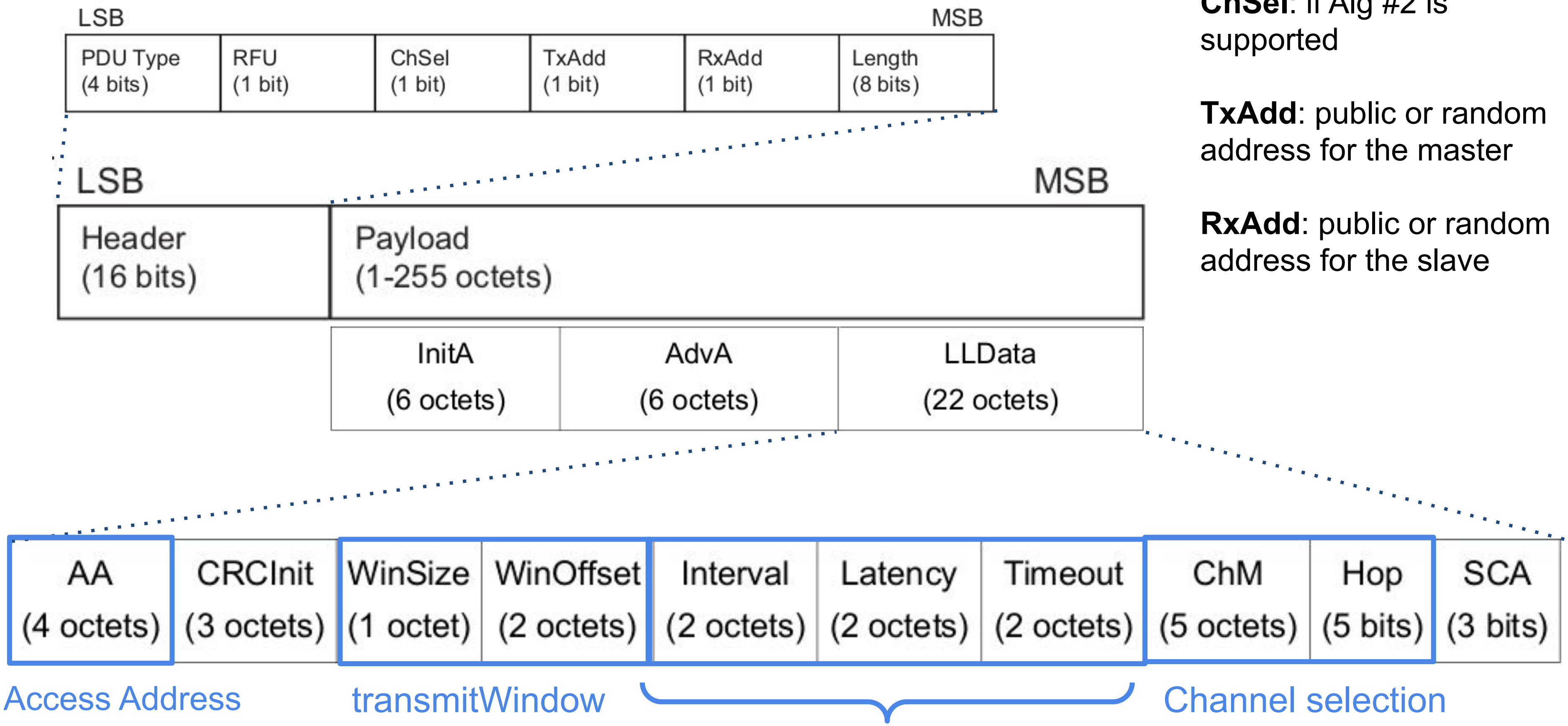
- A scannable advertisement admits *Scan Request*
  - The scanner sends the Scan request on the same channel
    - Backoff process to avoid collisions
  - *Scan Response* uses also the same channel
    - Can interrupt the *adv. event*

Initiating a connection



- In state *initiating*
- The device scans the advertisement channels as in the scan event
- If the advertisement is *connectable* the device sends a *Connection Request*
  - Generally in the same channel
  - In the case of LE coded a secondary channel is used

PDU format: Connection Request



LSB					MSB
PDU Type (4 bits)	RFU (1 bit)	ChSel (1 bit)	TxAdd (1 bit)	RxAdd (1 bit)	Length (8 bits)

LSB	MSB
Header (16 bits)	Payload (1-255 octets)

InitA	AdvA	LLData
(6 octets)	(6 octets)	(22 octets)

AA	CRCInit	WinSize	WinOffset	Interval	Latency	Timeout	ChM	Hop	SCA
(4 octets)	(3 octets)	(1 octet)	(2 octets)	(2 octets)	(2 octets)	(2 octets)	(5 octets)	(5 bits)	(3 bits)

ChSel: if Alg #2 is supported

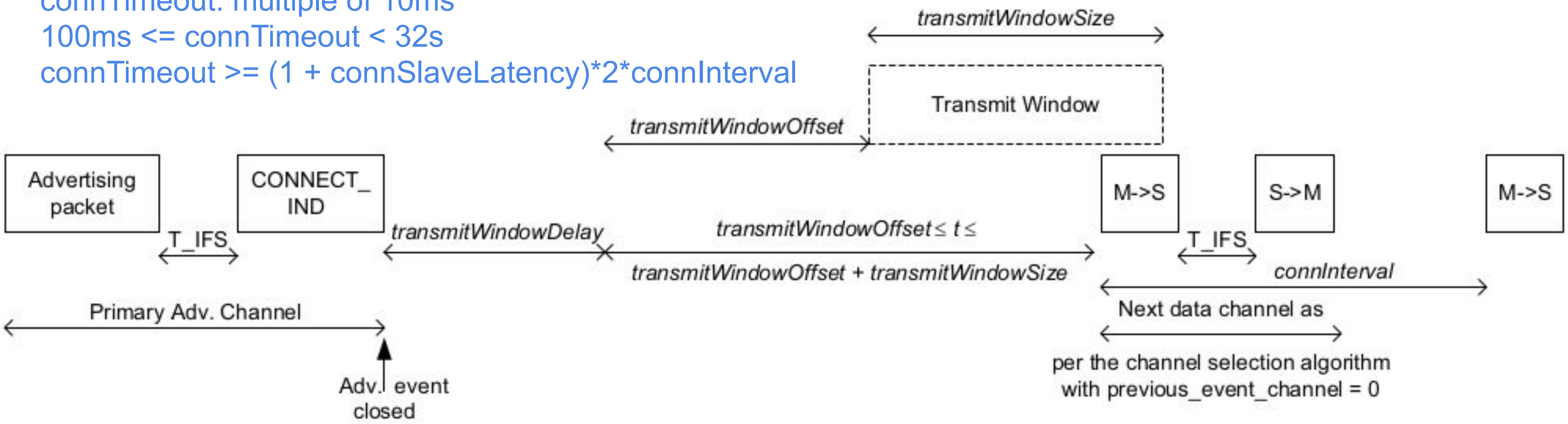
TxAdd: public or random address for the master

RxAdd: public or random address for the slave

connInterval = Interval *1.25ms (entre 1.25ms y 4s)
connSlaveLatency = Latency
connSupervisionTimeout = Timeout * 10ms

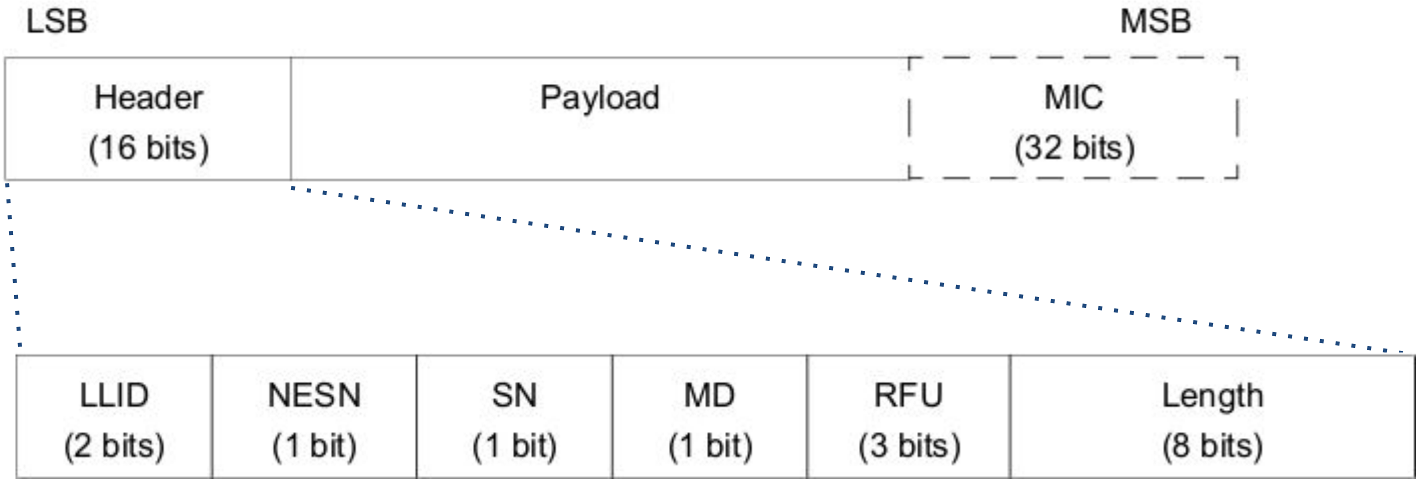
Connection Events

connTimeout: multiple of 10ms
100ms <= connTimeout < 32s
connTimeout >= (1 + connSlaveLatency)*2*connInterval



- Starts with the connection request packet M -> S
- Each event uses a different channel
  - FHSS using the ChMap sent in the conn request
  - Hop and Alg #2 (or #1 if ChSel = 0)
- They repeat periodically (connInterval)
- The slave can ignore *connSlaveLatency* events
- *connTimeout*: timeout interval to cancel the connection if there is no answer from the slave

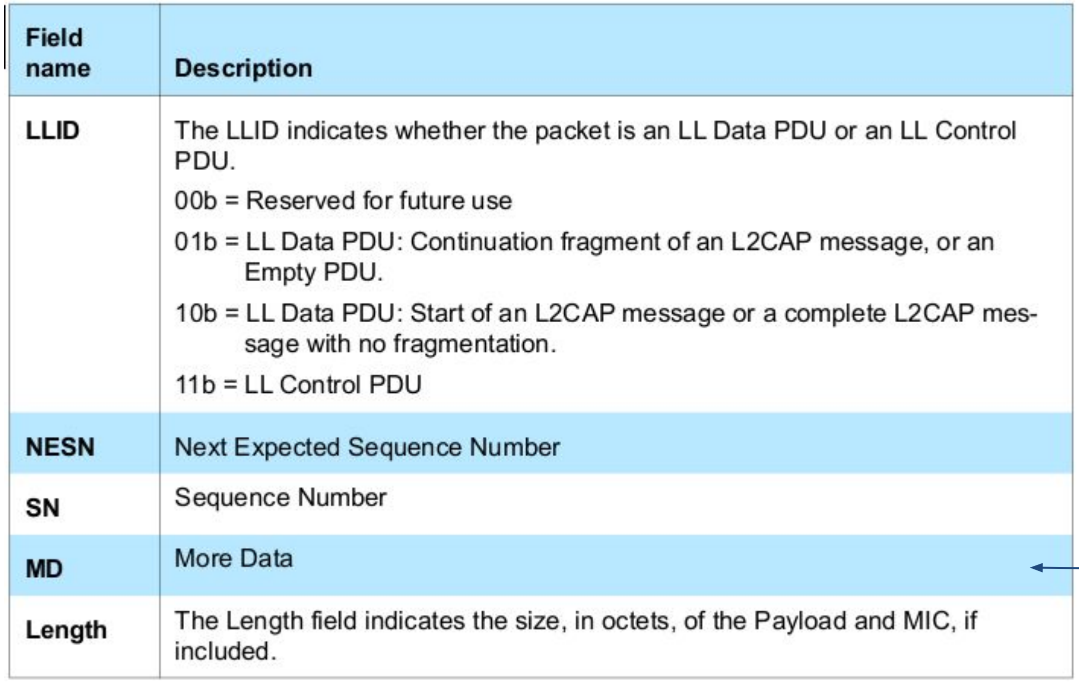
PDU format: Data Channel/Connections



LSB		MSB
Header (16 bits)	Payload	MIC (32 bits)

LLID	NESN	SN	MD	RFU	Length
(2 bits)	(1 bit)	(1 bit)	(1 bit)	(3 bits)	(8 bits)

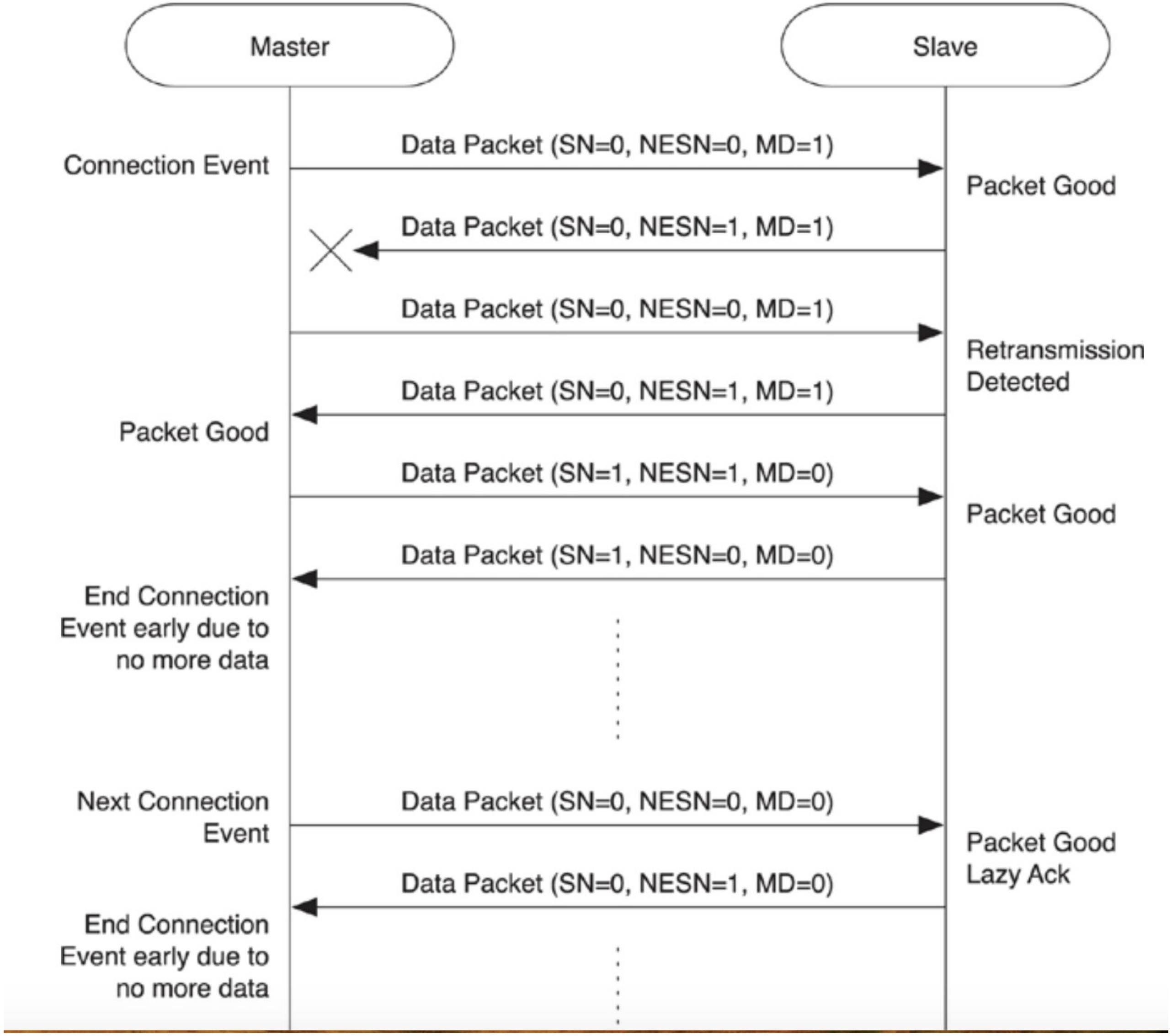
Long del payload en bytes



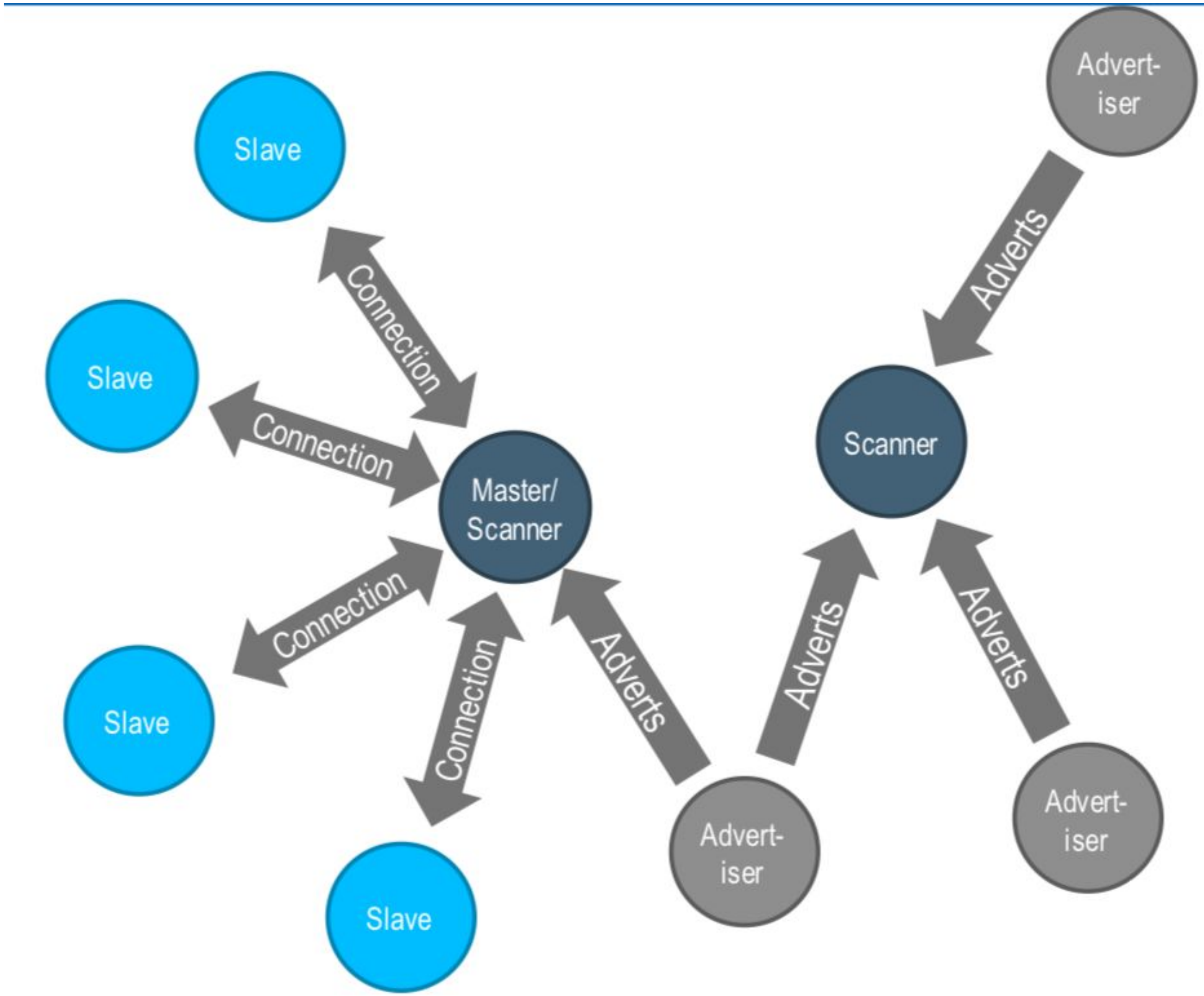
Field name	Description
LLID	The LLID indicates whether the packet is an LL Data PDU or an LL Control PDU. 00b = Reserved for future use 01b = LL Data PDU: Continuation fragment of an L2CAP message, or an Empty PDU. 10b = LL Data PDU: Start of an L2CAP message or a complete L2CAP message with no fragmentation. 11b = LL Control PDU
NESN	Next Expected Sequence Number
SN	Sequence Number
MD	More Data
Length	The Length field indicates the size, in octets, of the Payload and MIC, if included.

Extends the current event

Flow control: SN, NESN and MD

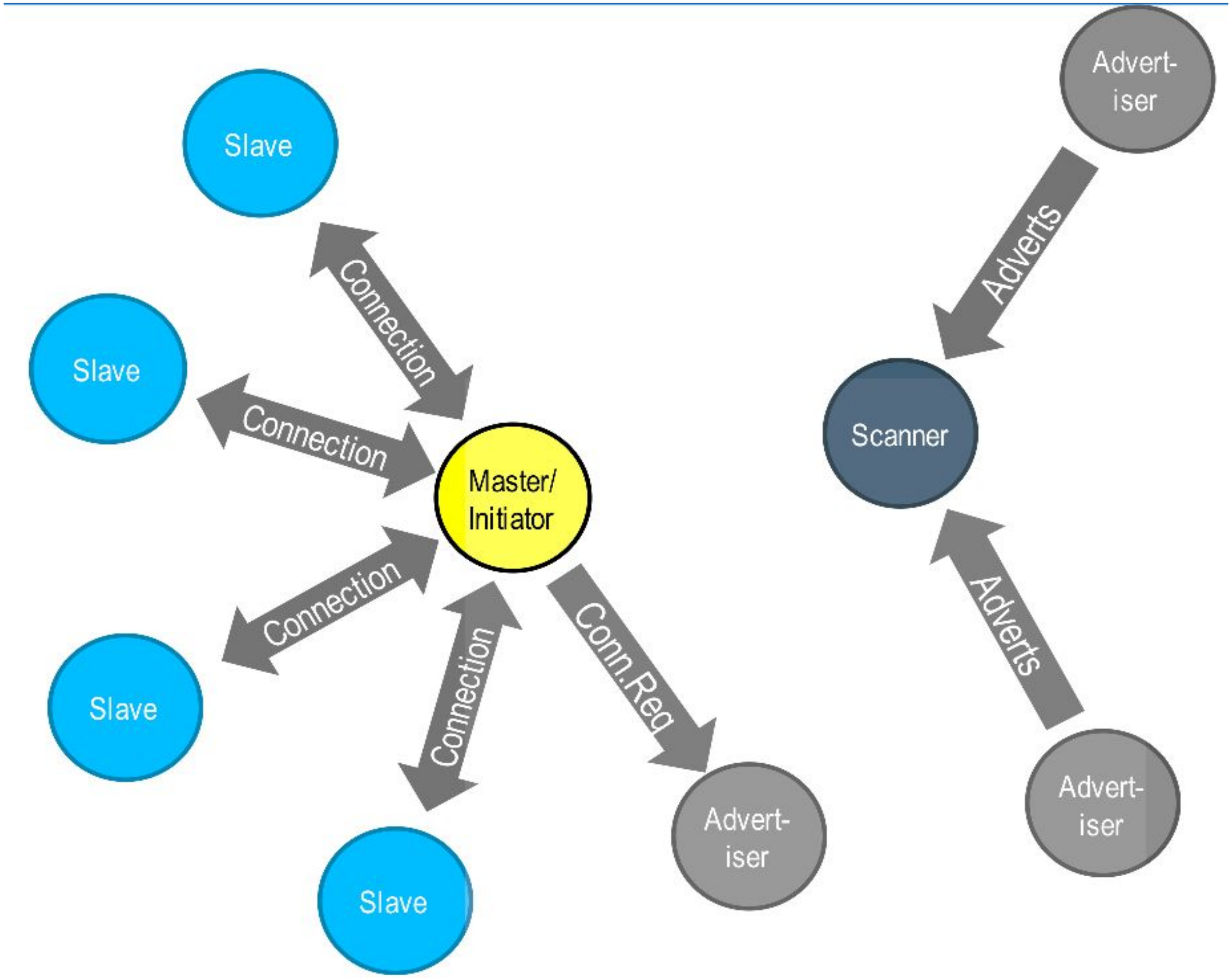

Master
Slave
Connection Event
Data Packet (SN=0, NESN=0, MD=1)
Packet Good
Data Packet (SN=0, NESN=1, MD=1)
Data Packet (SN=0, NESN=0, MD=1)
Retransmission Detected
Data Packet (SN=0, NESN=1, MD=1)
Packet Good
Data Packet (SN=1, NESN=1, MD=0)
Packet Good
Data Packet (SN=1, NESN=0, MD=0)
End Connection Event early due to no more data
Next Connection Event
Data Packet (SN=0, NESN=0, MD=0)
Packet Good Lazy Ack
Data Packet (SN=0, NESN=1, MD=0)
End Connection Event early due to no more data

Piconet

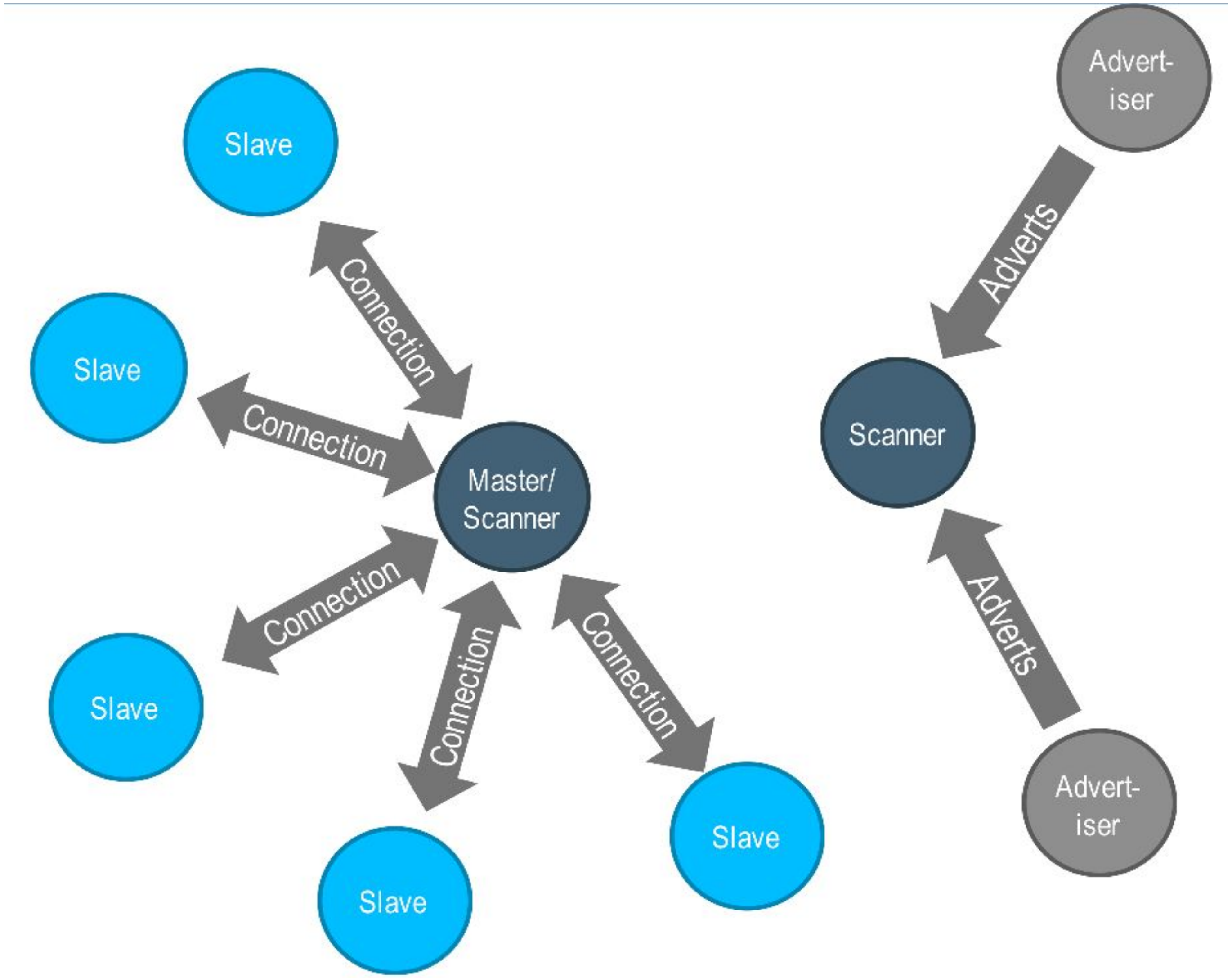


Advert- iser
Slave
Adverts
Connection
Slave
Connection
Scanner
Master/ Scanner
Connection
Connection
Adverts
Adverts
Adverts
Slave
Advert- iser
Advert- iser
Slave

Piconet

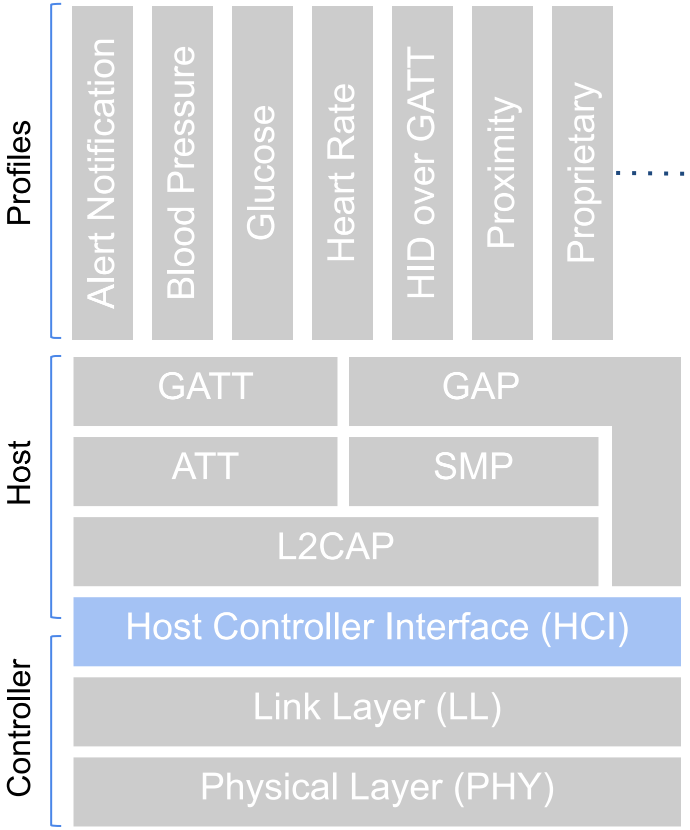

Slave
Slave
Slave
Slave
Connection
Connection
Connection
Connection
Master/ Initiator
Conn.Req
Advert- iser
Adverts
Scanner
Adverts
Advert- iser
Advert- iser

Piconet


Advert-iser
Slave
Adverts
Connection
Slave
Connection
Scanner
Master/ Scanner
Connection
Slave
Connection
Connection
Adverts
Advert-iser
Slave
Slave

Host Controller Interface (HCI)

- Standarises the communication between the Host and the Controller
  - Uses a serial interface
- Commands host->controller
- Events controller->host
- Two configurations
  - All in a single SoC
  - Host + Applications in one chip, the controller on a different chip
    - Used on smartphones



References

- Bluetooth core specification
  - https://www.bluetooth.com/specifications/bluetooth-core-specification/
- Kevin Townsed, Carles Cufí, Akiba & Robert Davidson, “Getting Started with Bluetooth Low Energy”, 2014, O’Reilly.
- Robin Heydon, “Bluetooth Low Energy: The Developer's Handbook”, 2013, Prentice Hall