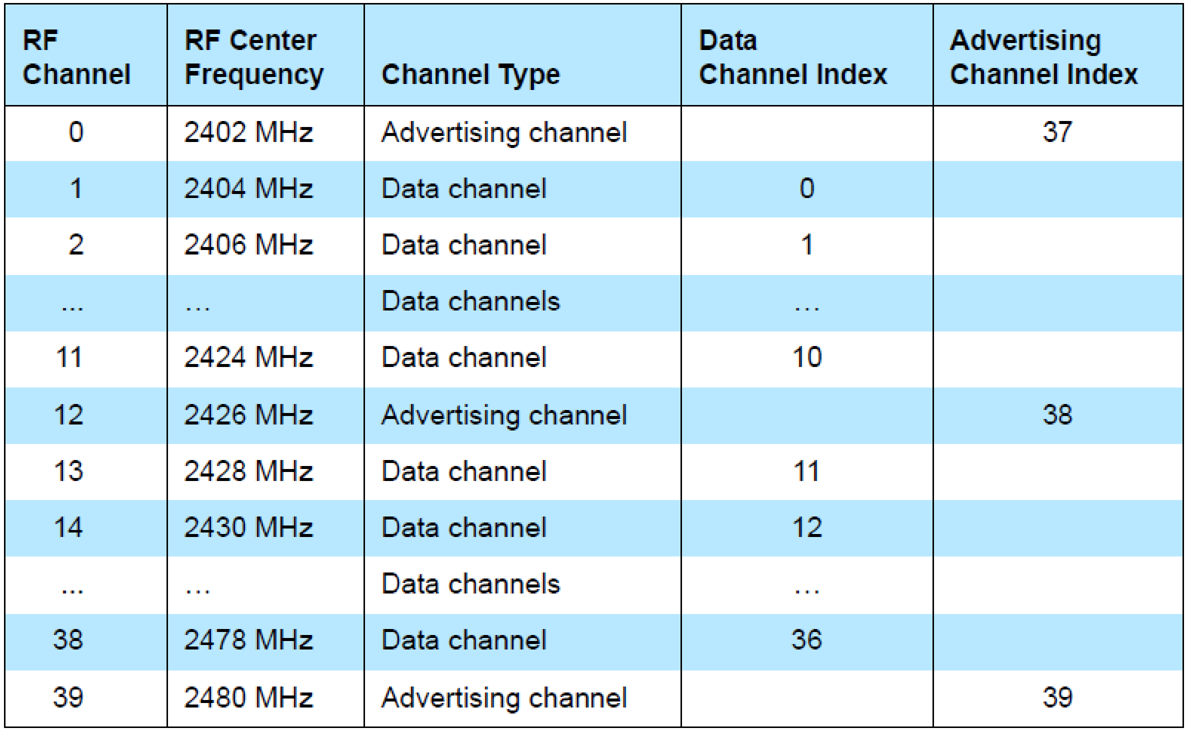
BLE: Bluetooth Low Energy

Frequency Channels

- ISM Band: 2.4 GHz
- 40 channels with BW of 2 MHz
  - 3 advertising channels for neighbor discovery: 37, 38, 39
  - 37 data channels (channel hopping)



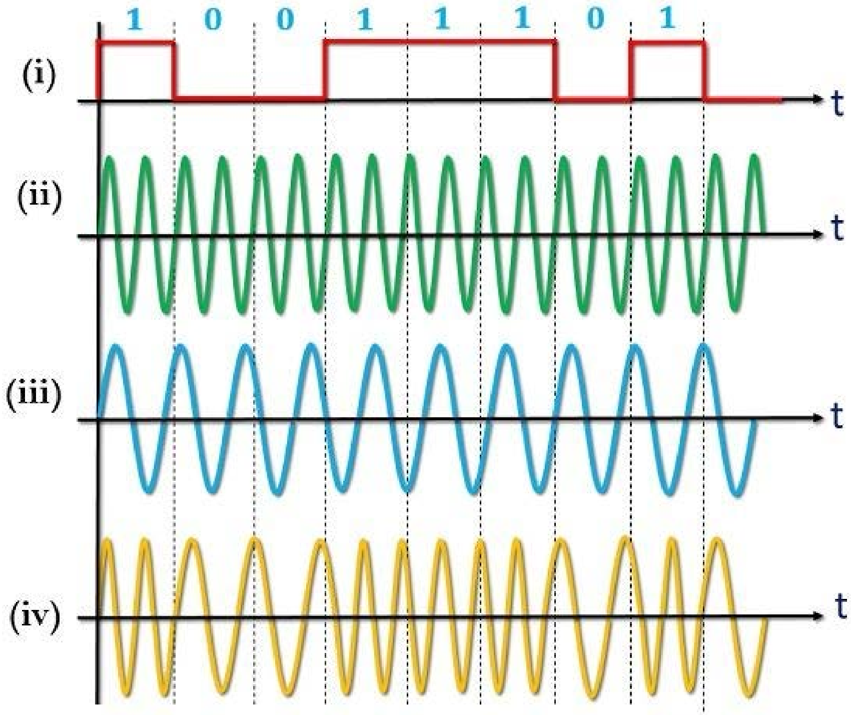
RF Channel	RF Center Frequency	Channel Type	Data Channel Index	Advertising Channel Index
0	2402 MHz	Advertising channel		37
1	2404 MHz	Data channel	0	
2	2406 MHz	Data channel	1	
...	…	Data channels	…	
11	2424 MHz	Data channel	10	
12	2426 MHz	Advertising channel		38
13	2428 MHz	Data channel	11	
14	2430 MHz	Data channel	12	
...	…	Data channels	…	
38	2478 MHz	Data channel	36	
39	2480 MHz	Advertising channel		39

- Bit rate: 1 Mbps (v 5.0: 2Mbps, 500 kbps, 125 kbps: optional)
- Modulation: GFSK

Range (x2) Range (x4)

Modulation

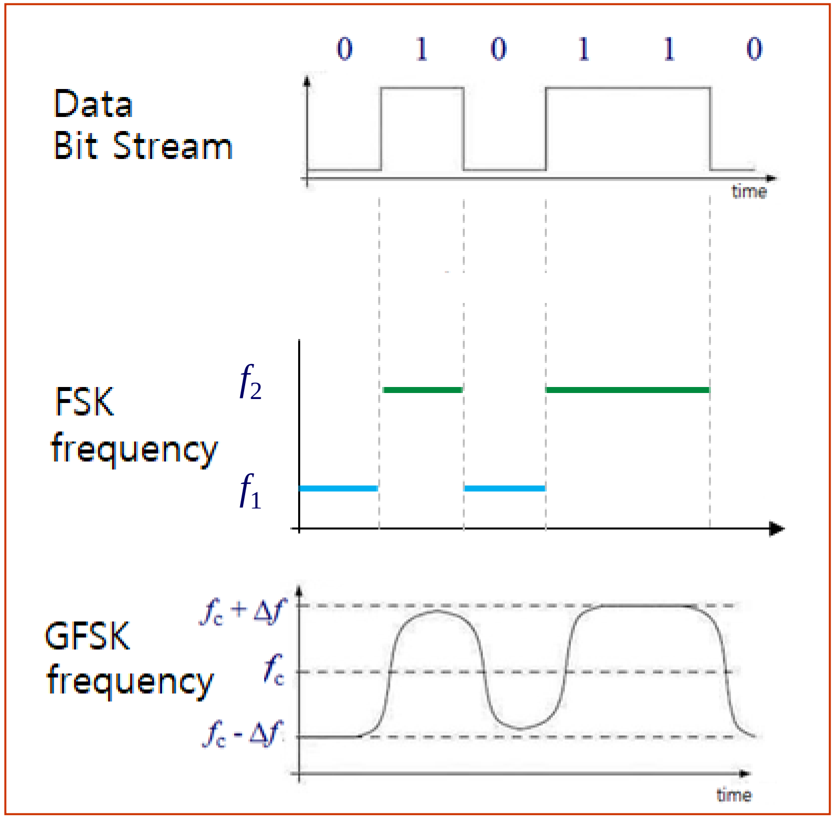
- Frequency Shift Keying (FSK)



(i) Digital bitstream
(ii) High frequency carrier wave
(iii) Low frequency carrier wave
(iv) FSK modulated wave

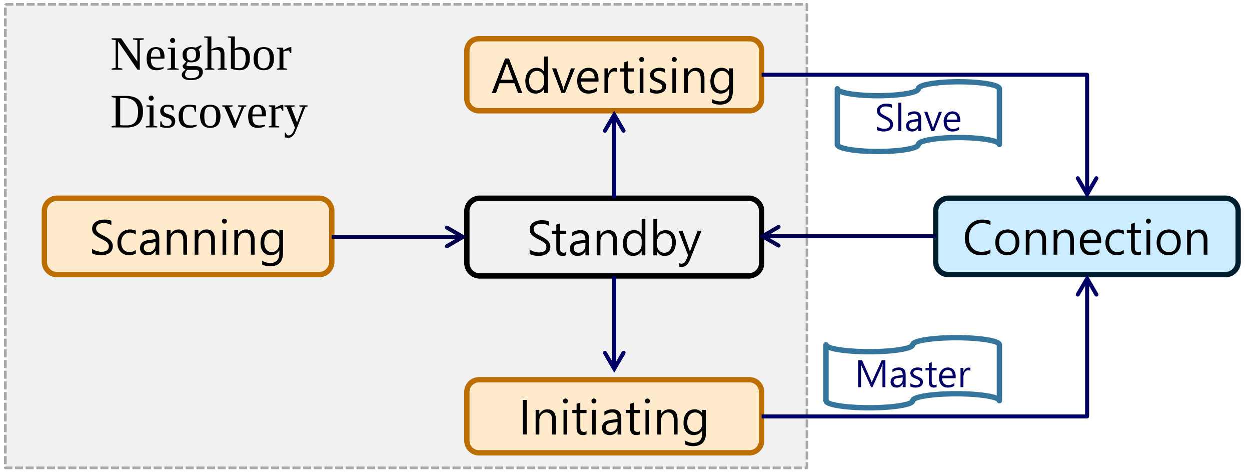
Electronics Coach

- Gaussian FSK (GFSK)



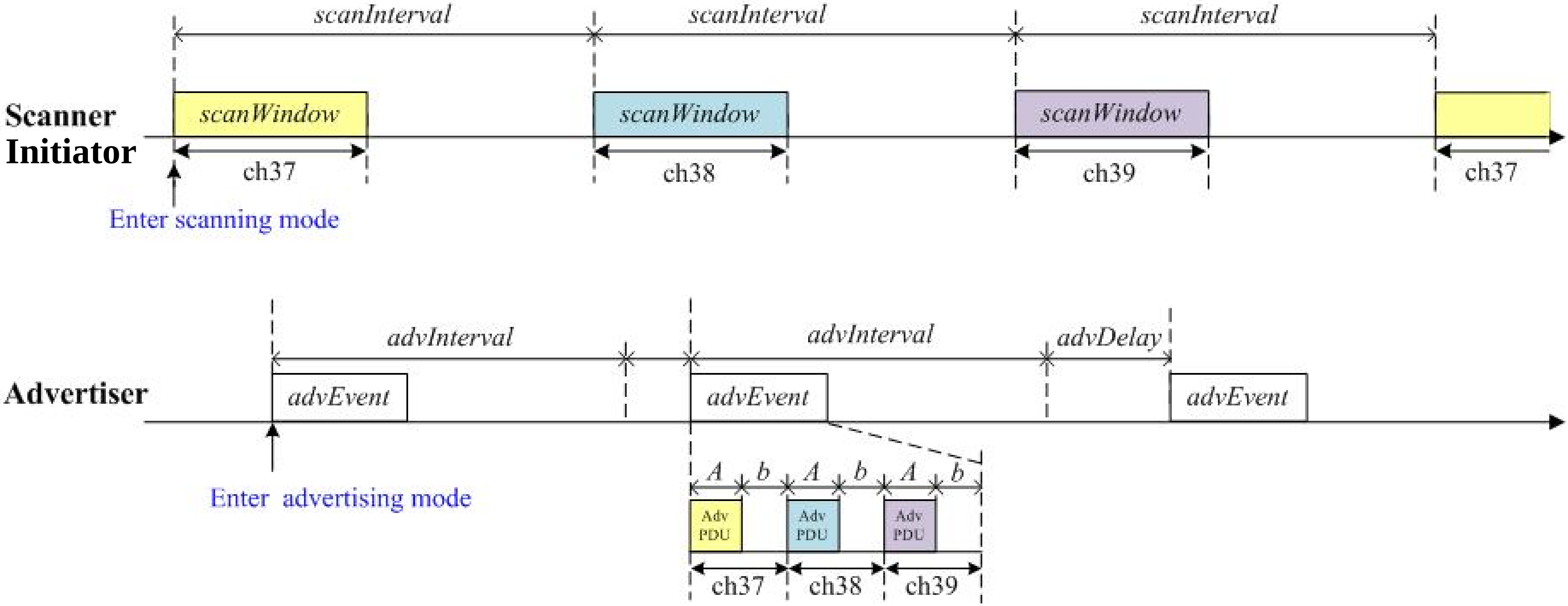
BLE Link Layer States

- **Standby State**
  - does not transmit and receive any packets
- **Advertising State**
  - transmitting advertising packet, then listening to and responding to responses
- **Scanning State**
  - listening for advertising packets
- **Initiating State**
  - listening for advertising packets from specific device(s) and responding to initiate connection
- **Connection State**
  - connected and ready for communication



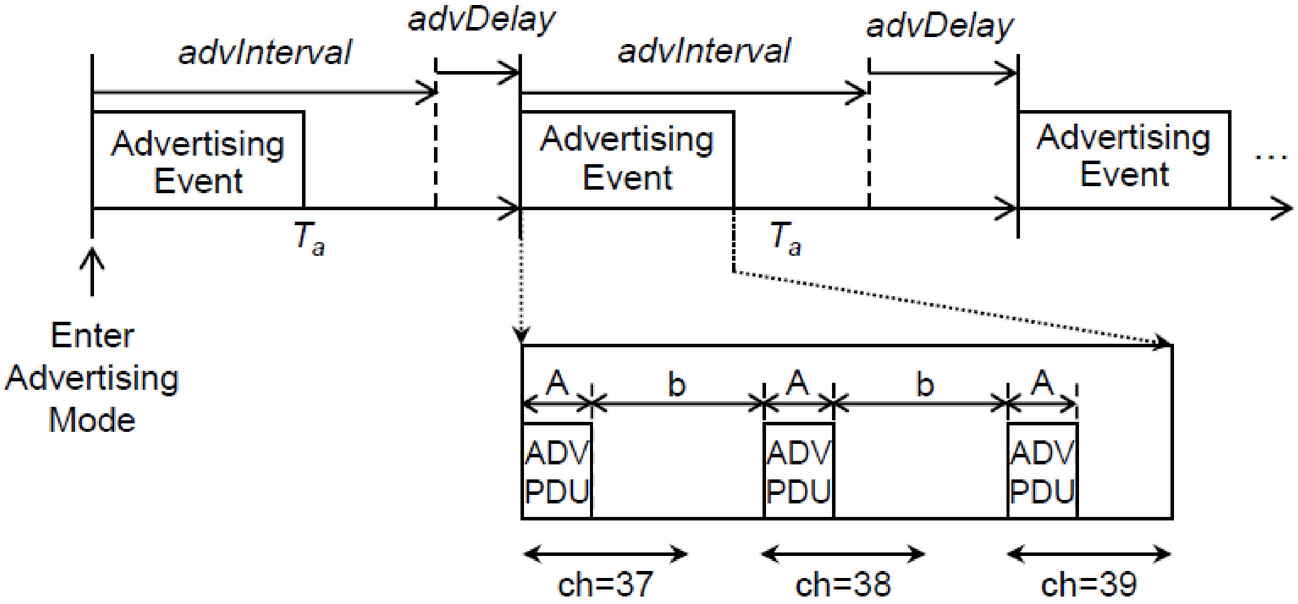
Neighbor Discovery: Overview

- ## Neighbor Discovery (ND) Process



ND: Advertising State (1)

General Advertising Process



- *advInterval* : an integer multiple of 0.625ms (in range 20ms ~ 10.24s)
- *advDelay* : a pseudo-random value with a range of 0ms ~ 10ms
- A+b $\leq$ 10ms

Advertising State (2)

- **4 Advertising Event Types**

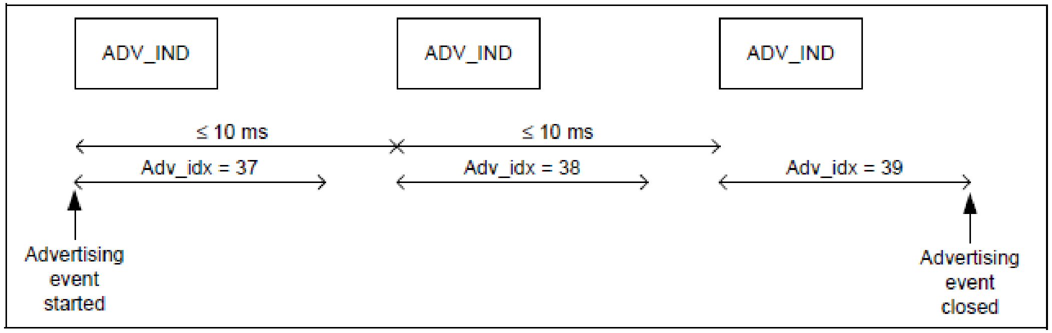
Advertising Event Type	PDU for advertising event type	Allowable response PDUs for advertising event	
		SCAN_REQ	CONNECT_REQ
Connectable undirected event	ADV_IND	Yes	Yes
Connectable directed event	ADV_DIRECT_IND	No	Yes
Non-connectable undirected event	ADV_NONCONN_IND	No	No
Scannable undirected event	ADV_SCAN_IND	Yes	No

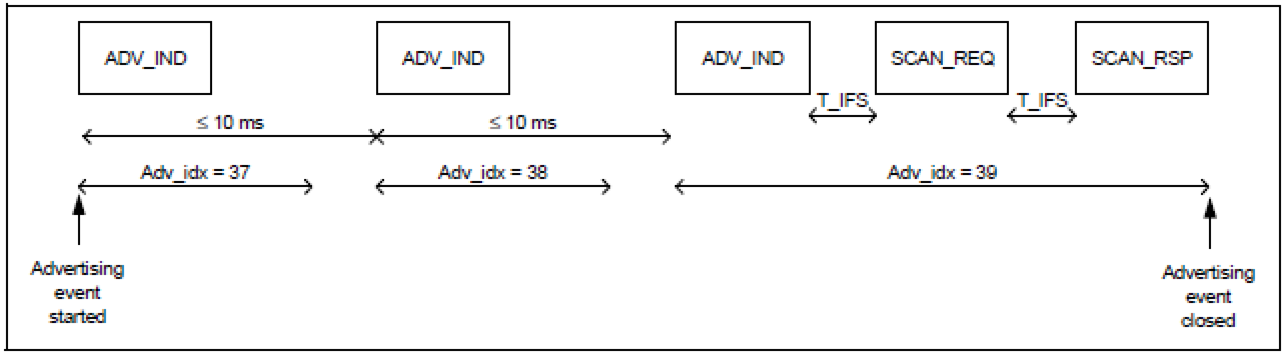
- SCAN_REQ: a scanner sends SCAN_REQ to the advertiser to ask for more information
- CONNECT_REQ: an initiator sends CONN_REQ to establish a connection

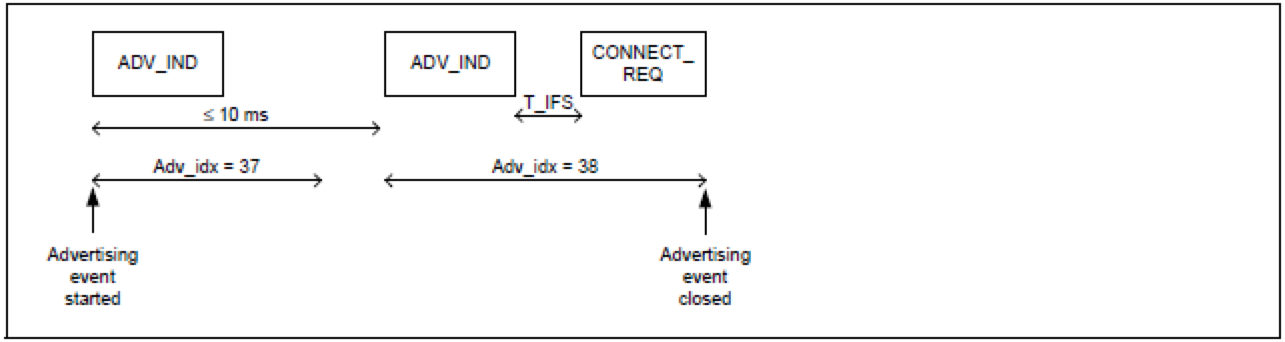
- For scannable undirected event or non-connectable undirected event type: *advInterval* < 100ms

Advertising State (3)

- ## Connectable Undirected Event



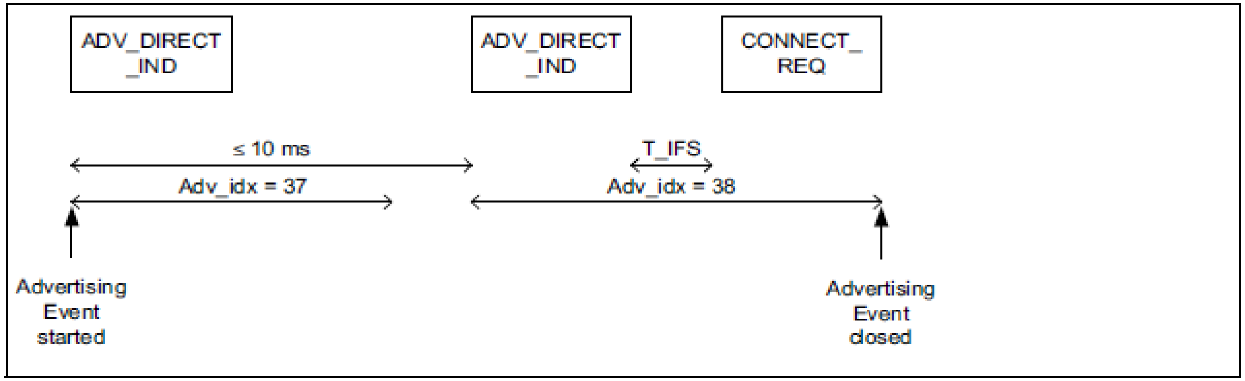




Advertising State (4)

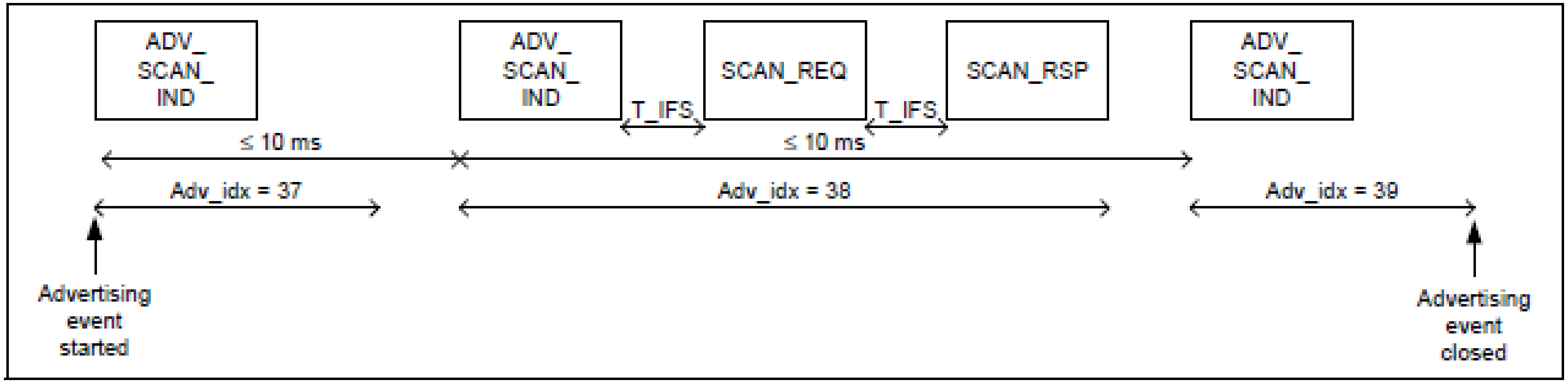
Connectable Directed Event

- Only specific device(s) are allowed to connect without requesting more scanning information
- Advertiser sends ADV_DIRECT_IND PDUs
    - contains both the initiator's and the advertiser's device address
- Initiator replies with CONNECT_REQ PDU



Advertising State (5)

- **Scannable Undirected Event**
  - No connection request is allowed
  - Advertiser sends ADV_SCAN_IND PDUs
  - Scanner sends SCAN_REQ PDU
  - Advertiser replies with SCAN_RSP PDU

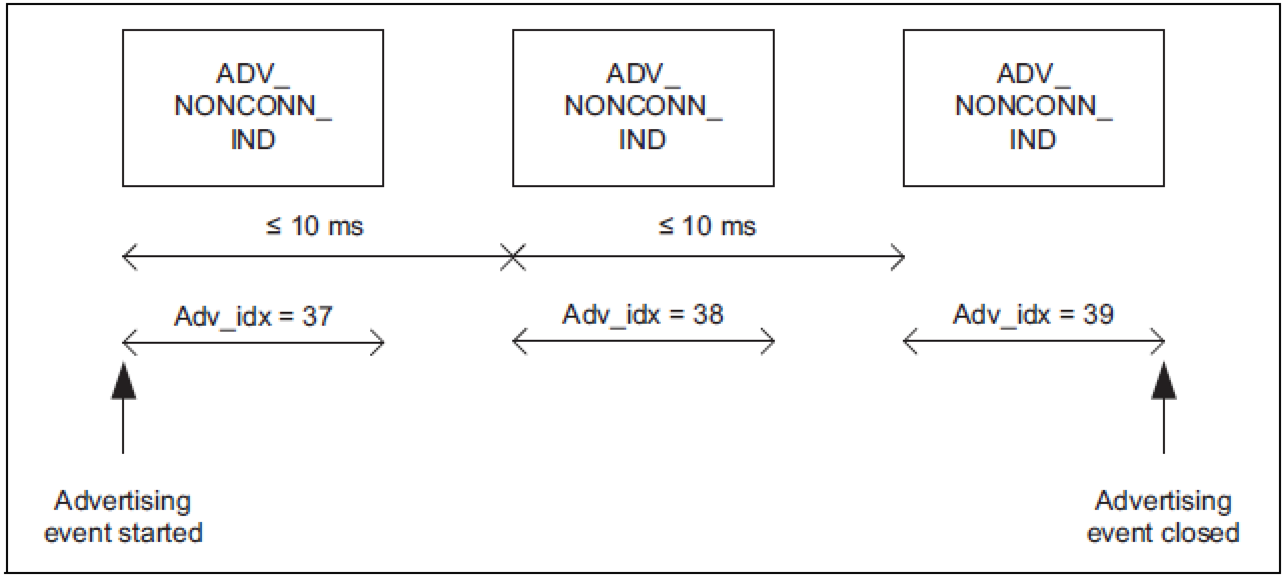


The process is similar with Connectable Undirected Event Type where the SCAN_REQ is sent by the scanner

Advertising State (6)

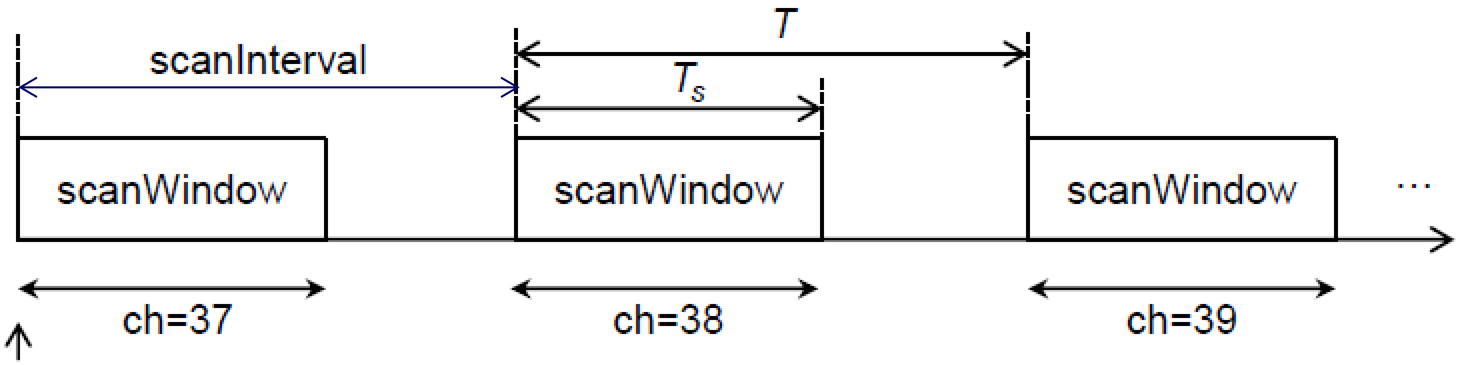
Non-connectable Undirected Event

- Scanner only listens the information sent by the advertiser
- Advertiser sends ADV_NONCONN_IND PDU



ND: Scanning and Initiating State

General scanning/initiating process



Enter scanning

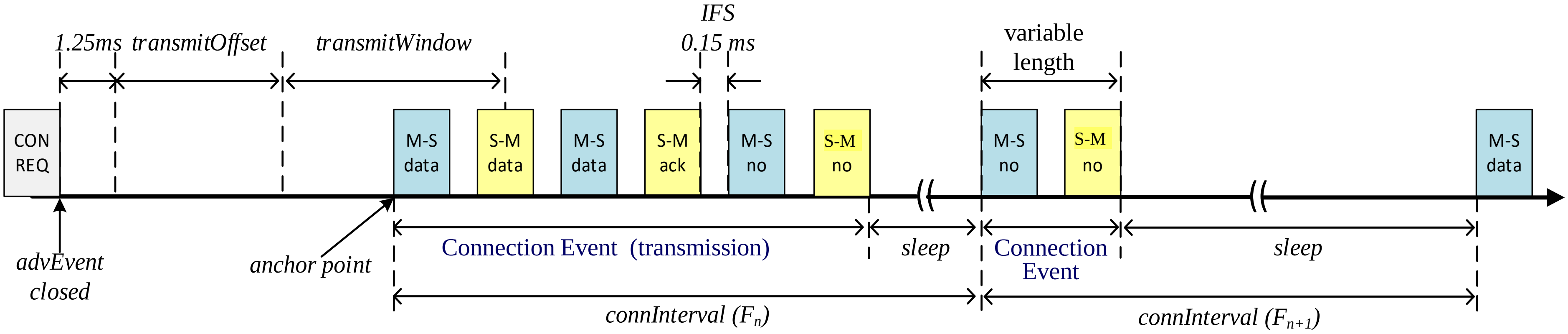
- $sanWindow \le scanInterval \le 10.24s$
- On every scan window, different channel should be scanned

- Initiating state
  - Similar with scanning state.

Connection State (1)

Master-Slave Communication

- Master and slave alternate a packet transmission
- All information for transmission in connection state is specified by CONNECT-REQ PDU

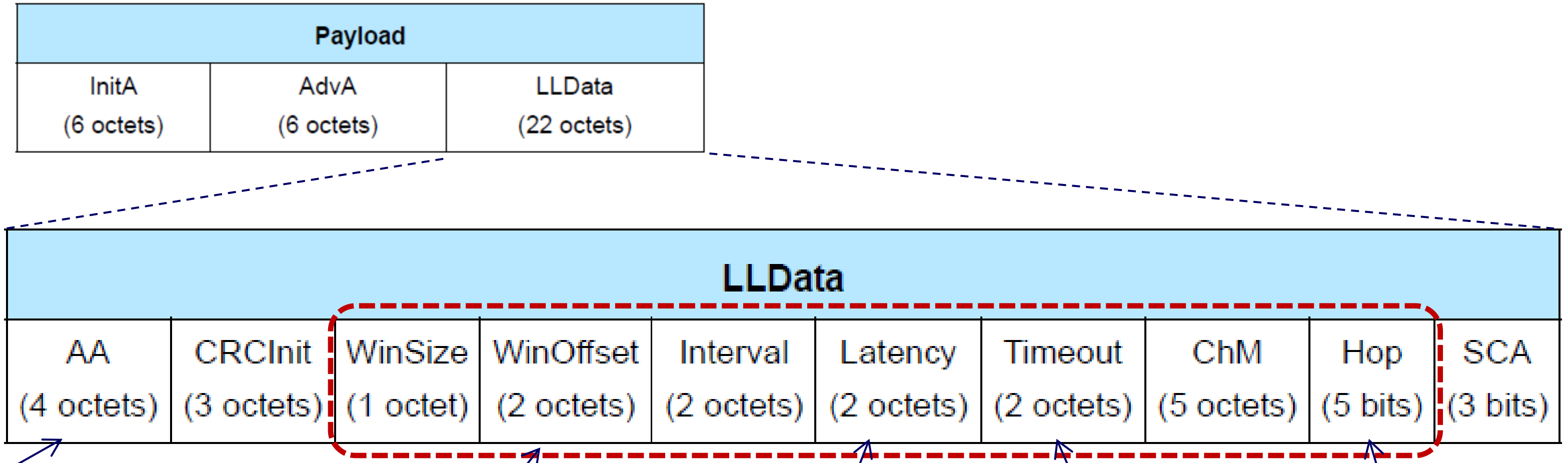


Anchor point : the transmission start time of the first message by Master received by Slave.

- If the slave misses the first message, it should wait until the next connection event and set the anchor point.

Connection State (2)

- **CONNECT_REQ PDU**



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

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

Connection ID

transmit Offset = WinOffset x 1.25 ms
TransmitWindow = WinSize x 1.25 ms
connInterval = Interval x 1.25 ms

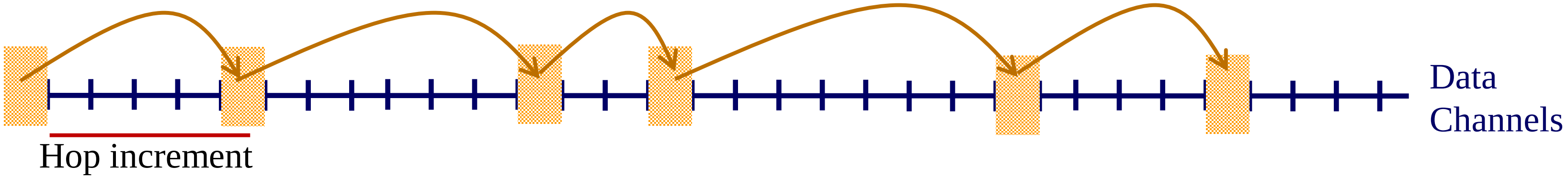
the first hopping data channel

the number of consecutive connection events that the slave can skip, i.e., sleep

If the device does not receive data for a duration of *connSupervisionTimeout*, it exits the connection state, i.e., the connection is considered lost

Connection state (3)

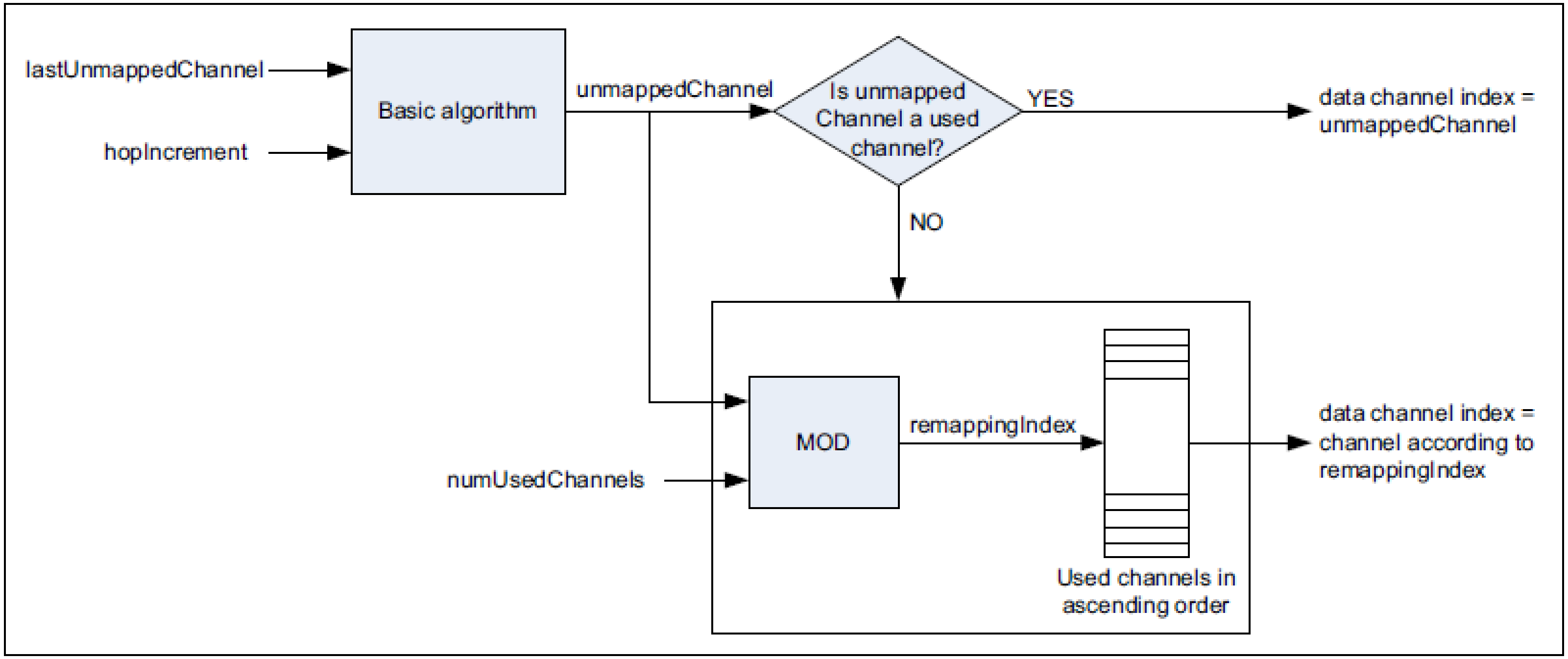
- **Channel Classification (37 data channels)**
  - Used channels : used for connection
  - Unused channels : not used for connection
- **Channel Hopping**
  - BLE uses different channel for different connection event



- **Channel Selection Algorithm**
  - Parameters
    - Unmapped channel (current connection event)
    - Hop increment
    - Last unmapped channel : unmapped channel of previous event
  - Selection step
    - Calculation of the unmapped channel index
    - Mapping this index to data channel

Connection State (4)

- **Channel Selection in Frequency Hopping (FH)**
  - Randomly selected for each connection event


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

$$F_{n+1} = (F_n + \underbrace{hopIncrement}_{\text{randomly generated}}) \bmod 37$$

$$remappingIndex = F_{n+1} \bmod numUsedChannels$$

Connection State (5)

- **Closing connection event**
  - When neither devices have data to send
    - MD (more data) bit of header: indicates whether a device has data to send
  - Two consecutive packets with invalid CRC (transmission failures) close connection event
  - When *connSupervisionTimeout* is reached.