

Everything you need to know about Bluetooth LE advertising

Nordic Tech Webinar

Hung Bui / Senior Application Engineer

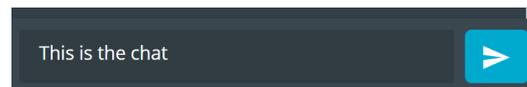
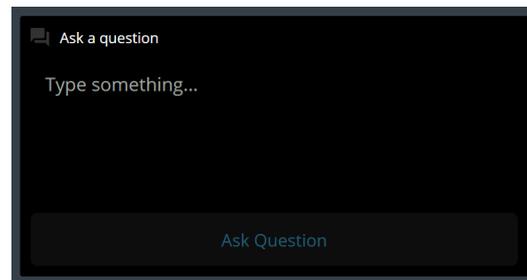
Håkon Helsing / Senior Application Engineer

September 2020



Practicalities

- Duration: 50-60 min
- Questions are encouraged!
- Please type questions in the top of the right sidebar
 - All questions are anonymous
 - Try to keep them relevant to the topic
- I will answer questions towards the end
- The chat is not anonymous, and should **not** be used for questions
- If you have more questions, please use DevZone
- A recording of the webinar will be available together with the presentation at webinars.nordicsemi.com



Today's hosts

Hung Bui



Senior Application
Engineer



Håkon Helsing

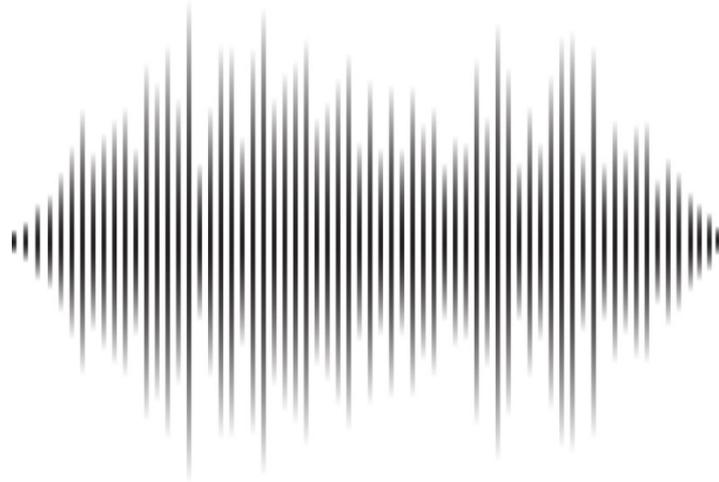


Senior Application Engineer



Agenda

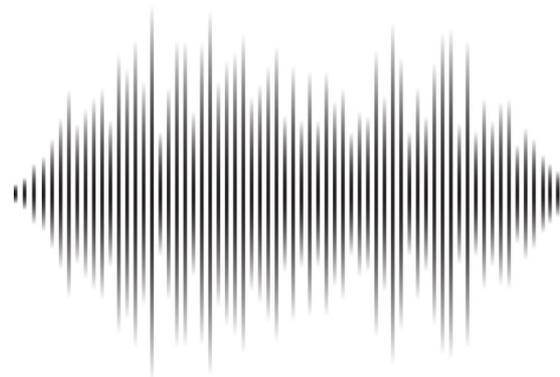
- Basics
- Advertising Extensions
- Advertising types
- Advertising data format
- nRF Connect SDK API and example walk through
- Advertisement analysis with nRF Sniffer for Bluetooth LE



Basics

What is advertising?

- A Broadcaster advertises to broadcast data
 - Data
 - RSSI
 - Direction Finding I/Q data
- A Peripheral advertises to broadcast data and/or to make itself discoverable for a connection
- Two types of advertising:
 - Legacy advertising
 - Extended advertising



GAP roles and Link Layer states

GAP role	Link Layer state
Broadcaster	Advertising
Observer	Scanning
Peripheral	Advertising Connection (Slave)
Central	Scanning Initiating Connection (Master)

All roles can also be in the standby state

Roles (GAP)



Broadcaster



Observer

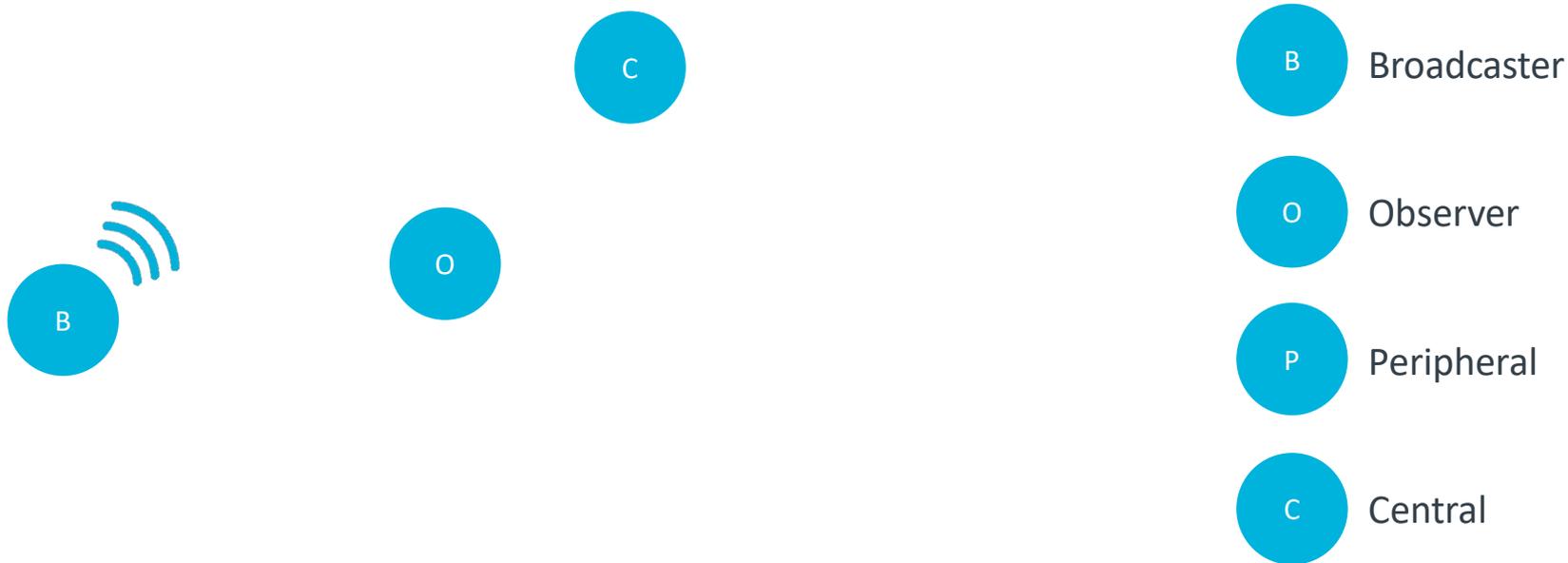


Peripheral

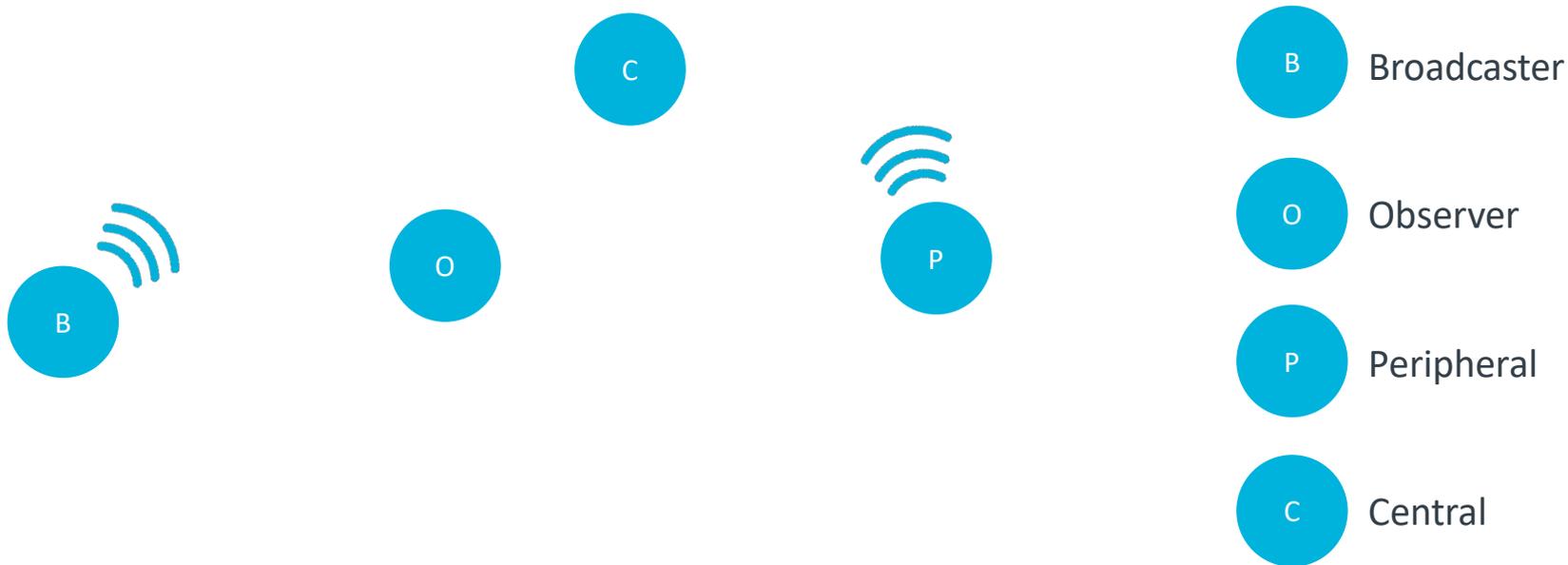


Central

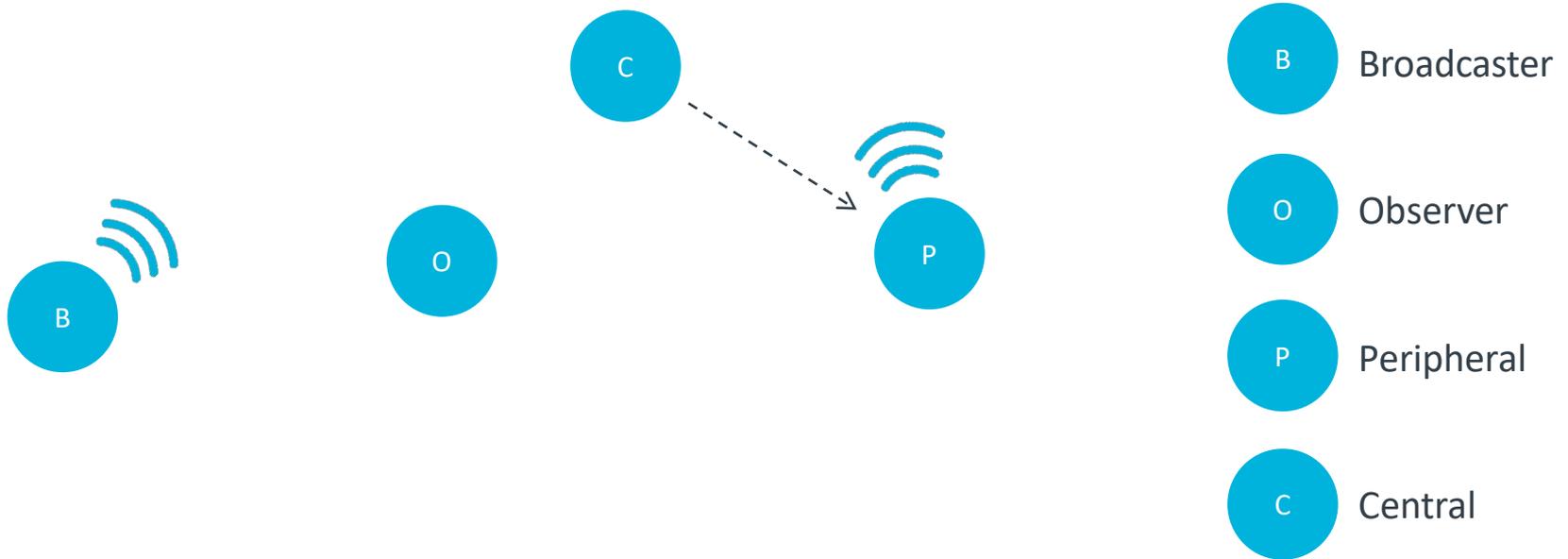
Roles (GAP)



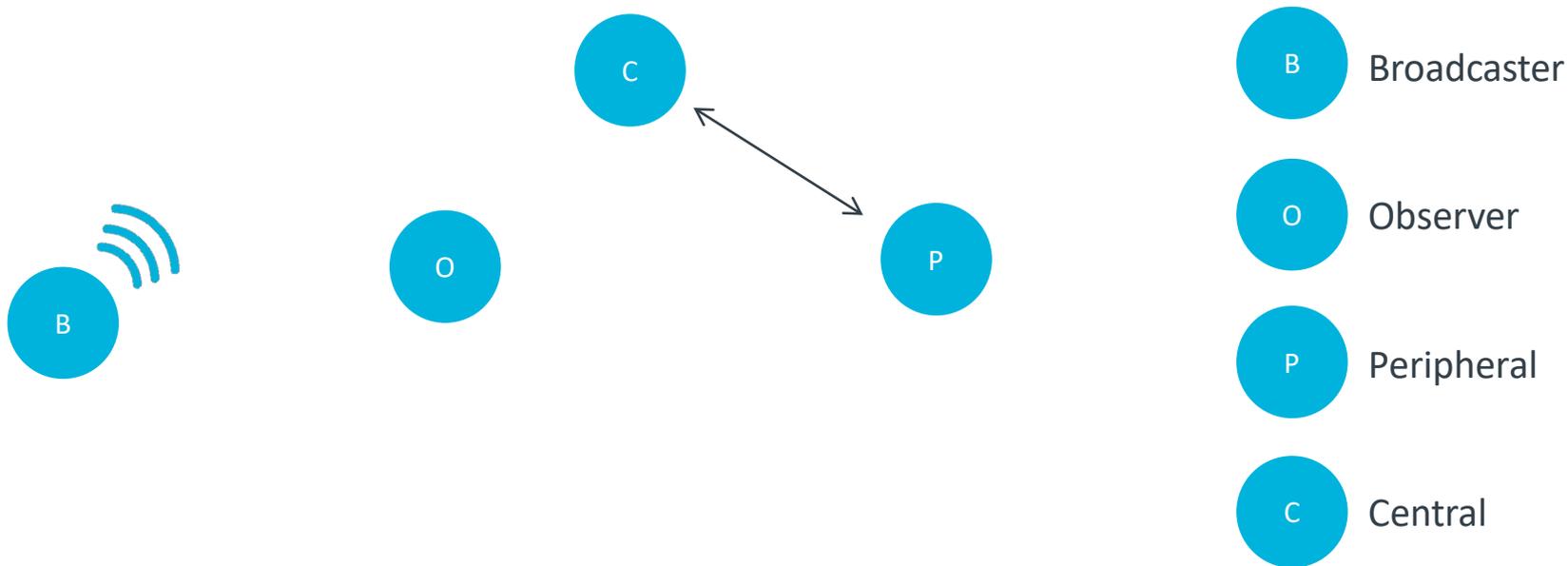
Roles (GAP)



Roles (GAP)

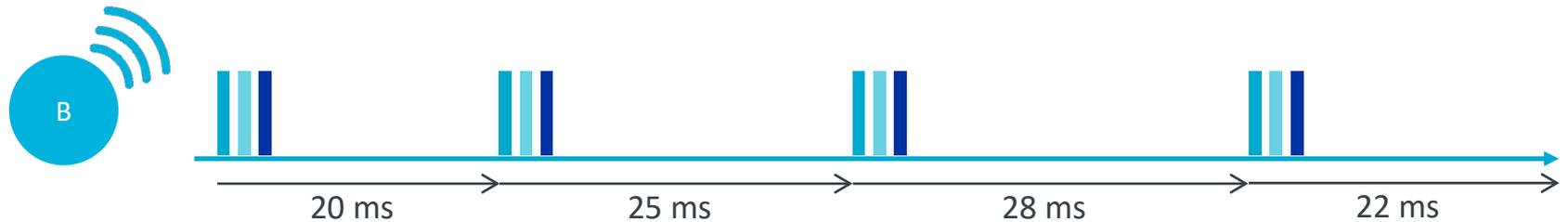


Roles (GAP)

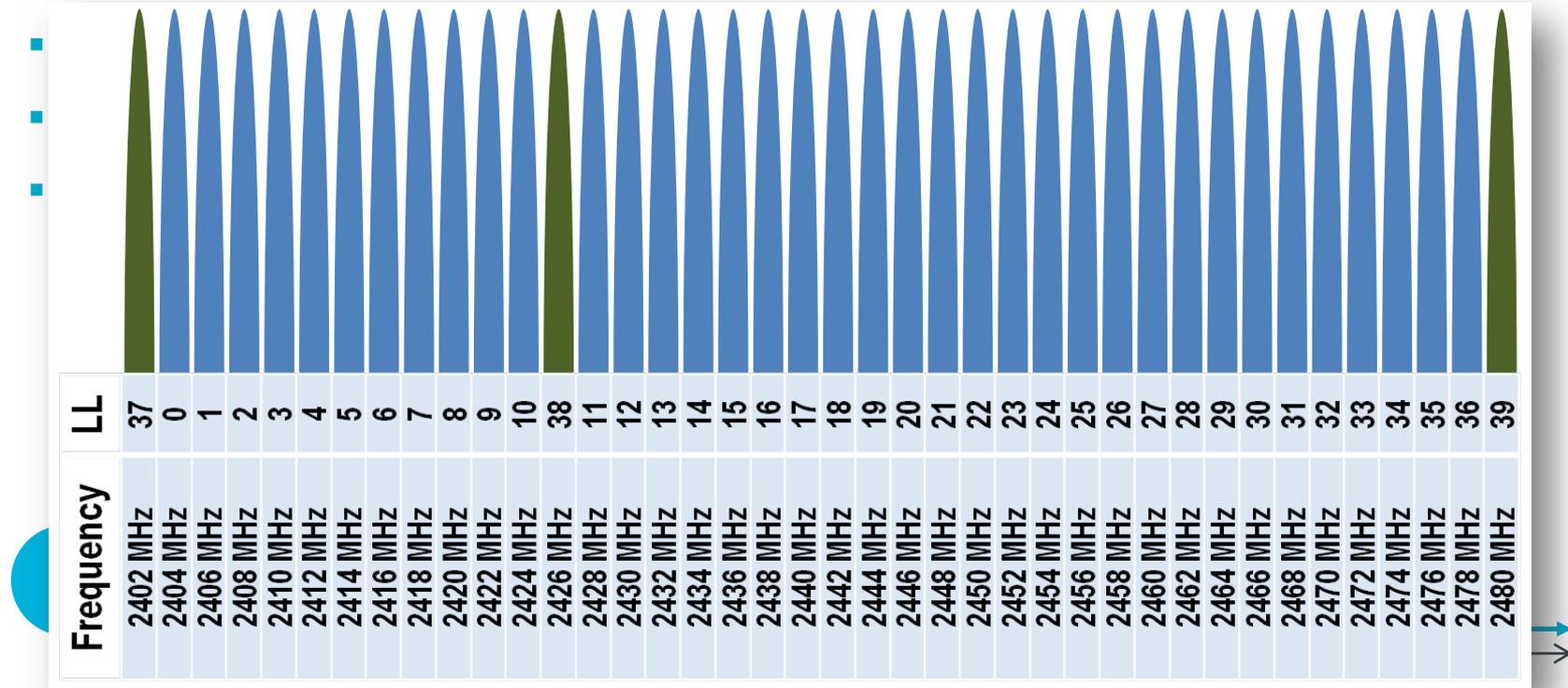


Advertising and scanning

- Advertising interval = 20 ms
- 0-10 ms random delay
- Advertising on channel

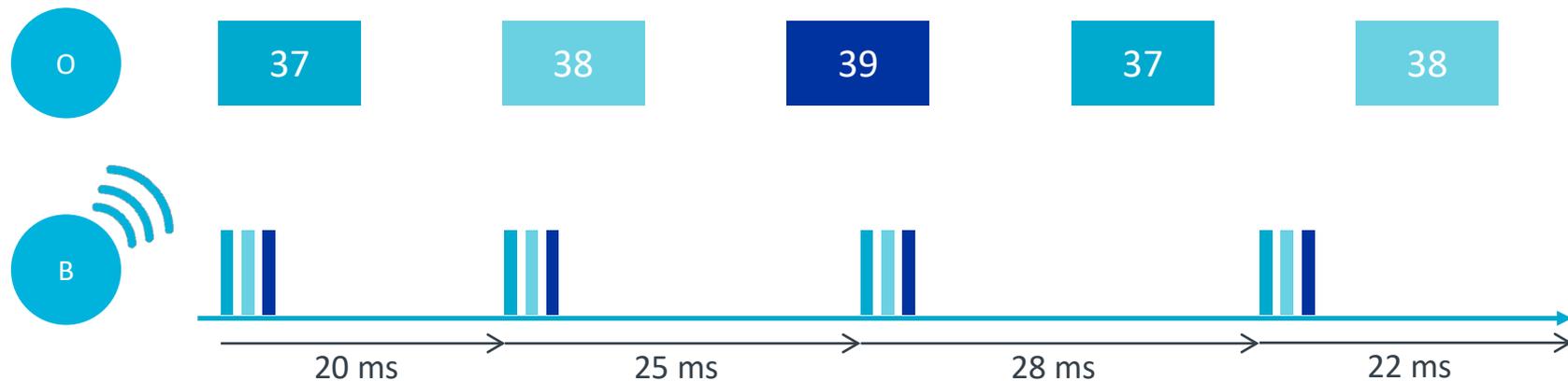


Advertising and scanning



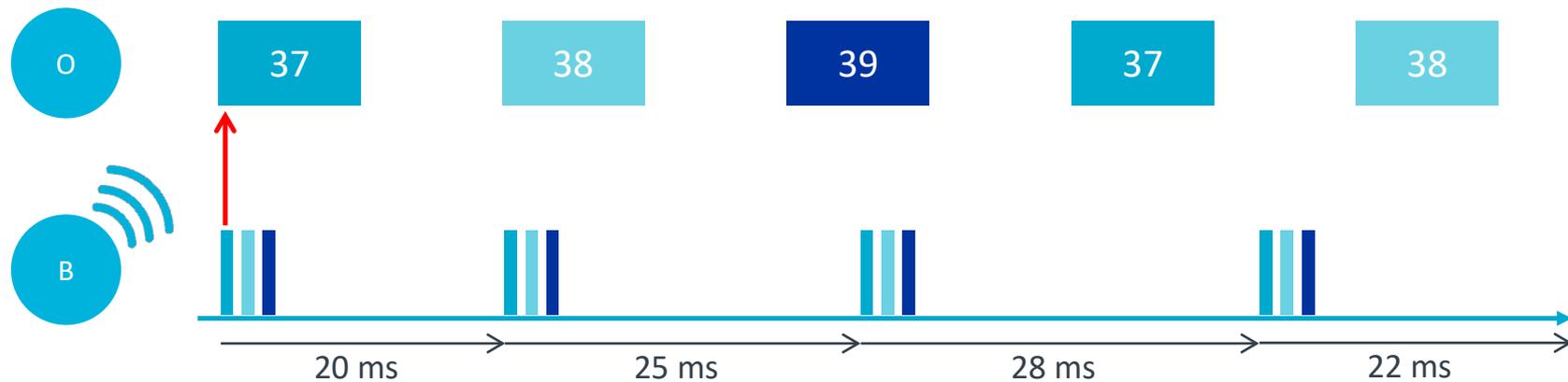
Advertising and scanning

- Scan interval = 20 ms
- Scan window = 10 ms



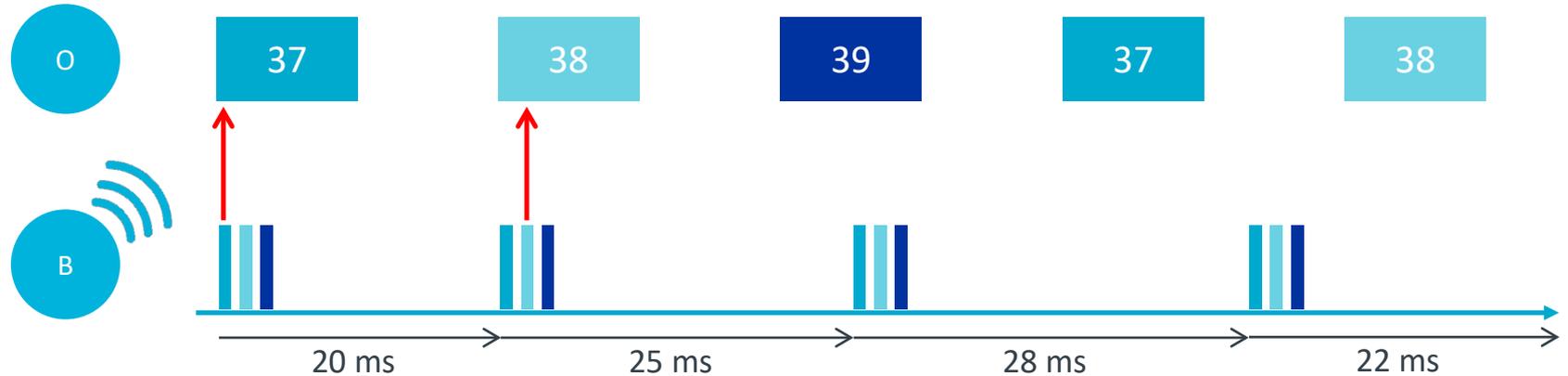
Advertising and scanning

- Scan interval = 20 ms
- Scan window = 10 ms



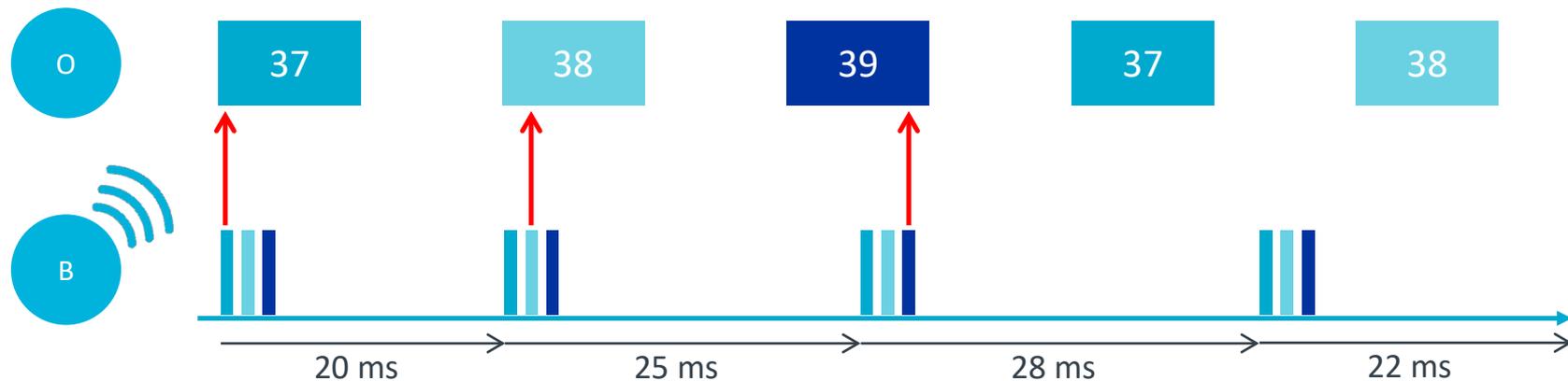
Advertising and scanning

- Scan interval = 20 ms
- Scan window = 10 ms



Advertising and scanning

- Scan interval = 20 ms
- Scan window = 10 ms



Advertising types

A photograph of a white lighthouse at night. The lighthouse is illuminated from within, and its light is projected outwards in several beams, creating a starburst effect against the dark blue sky. The lighthouse has a cylindrical body with a small window and a glass-enclosed lantern room at the top. The overall scene is serene and atmospheric.

Legacy advertising types

Type	Scannable	Connectable
ADV_IND	Yes	Yes
ADV_SCAN_IND	Yes	
ADV_NONCONN_IND		
ADV_DIRECT_IND		Yes

Legacy advertising types

Type	Scannable	Connectable
ADV_IND	Yes	Yes
ADV_SCAN_IND	Yes	
ADV_NONCONN_IND		
ADV_DIRECT_IND		Yes

- Extended advertising types
 - ADV_EXT_IND
 - AUX_ADV_IND
 - AUX_SYNC_IND
 - AUX_CHAIN_IND

GAP roles and advertising types

GAP role	Advertising types
Broadcaster	ADV_SCAN_IND ADV_NONCONN_IND
Peripheral	ADV_IND ADV_SCAN_IND ADV_NONCONN_IND ADV_DIRECT_IND

Scan request and scan response

- ADV_SCAN_IND and ADV_IND are scannable



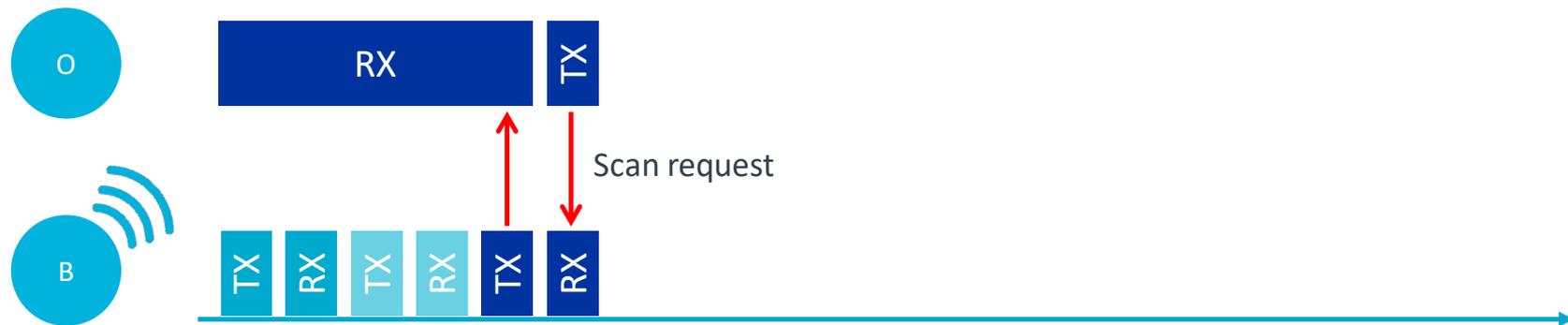
Scan request and scan response

- ADV_SCAN_IND and ADV_IND are scannable



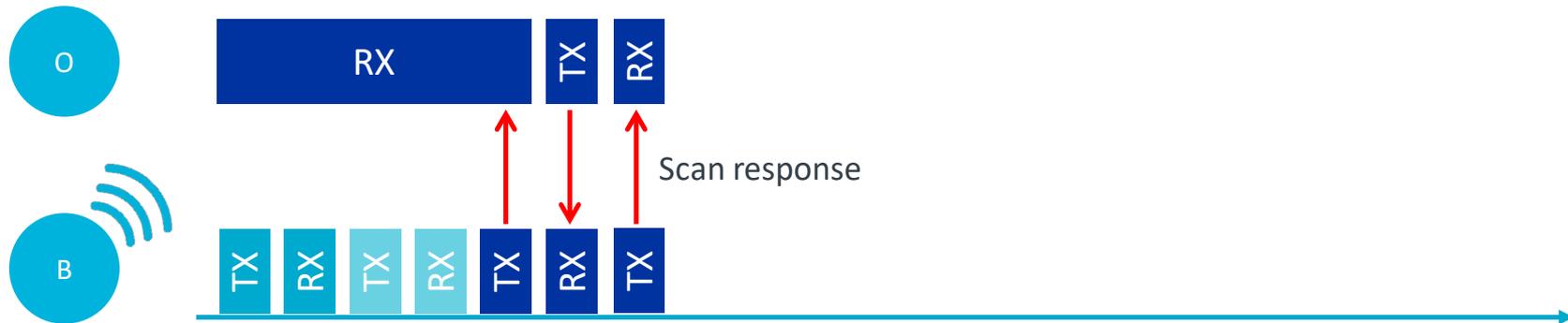
Scan request and scan response

- ADV_SCAN_IND and ADV_IND are scannable



Scan request and scan response

- ADV_SCAN_IND and ADV_IND are scannable



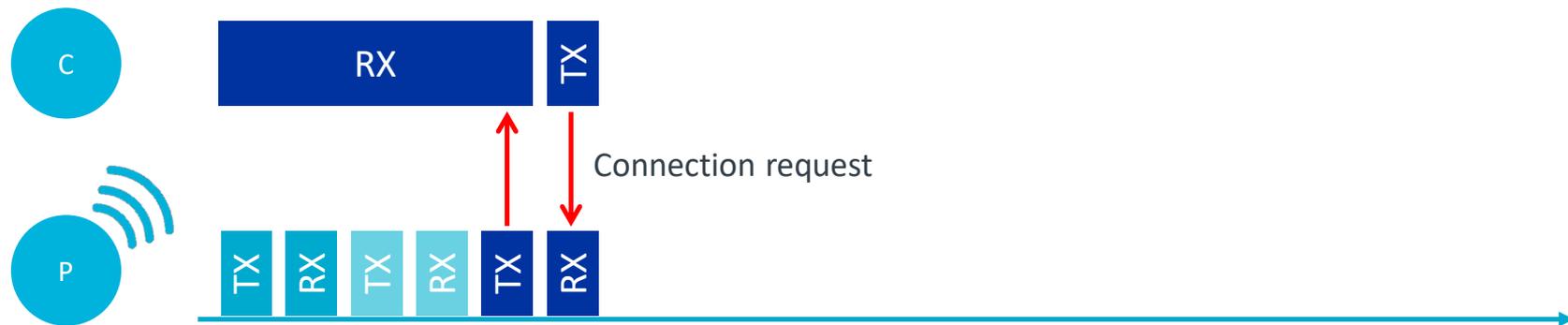
Connection request

- ADV_IND and ADV_DIRECT_IND are connectable



Connection request

- ADV_IND and ADV_DIRECT_IND are connectable



Non-connectable advertising

- ADV_NONCONN_IND
- Not connectable or scannable
- No RX -> Lower power consumption



Advertising interval

- ADV_NONCONN_IND, ADV_SCAN_IND, ADV_IND
 - 20 ms to 10.24 seconds, in steps of 0.625 ms
 - 0-10 ms random delay
- Trade-off between power consumption and device discovery time
- Scan interval and scan window
- Leverage scanner if possible

Advertising interval - ADV_DIRECT_IND

- Directed advertising
- Used to quickly reconnect to a known Central
- Two duty cycle options:
 - Low duty cycle
 - The time between two advertisements shall be less or equal to 10 ms
 - High duty cycle
 - The time between two advertisements shall be less or equal to 3.75 ms
 - Maximum duration of 1.28 s

White list

- An advertiser can use a white list to filter out unwanted scan requests and/or connection requests
- The white list contains device addresses and/or Identity Resolving Keys (IRKs) that the advertiser shall accept scan requests and/or connection requests from, filtering away all other.
 - These typically belong to devices it has bonded with
 - IRKs are shared after bonding, and are used to resolve private resolveable addresses



Advertising data

Advertising PDU

- Advertising PDU:



- Advertising PDU header:



Advertising PDU header

- Advertising PDU Header:

PDU type 4 bits	RFU 1 bit	ChSel 1 bit	TXAdd 1 bit	RXAdd 1 bit	Length 8 bits
---------------------------	---------------------	-----------------------	-----------------------	-----------------------	-------------------------

PDU type	PDU name
0b0000	ADV_IND
0b0001	ADV_DIRECT_IND
0b0010	ADV_NONCONN_IND
0b0110	ADV_SCAN_IND

Advertising PDU header

- Advertising PDU header:

PDU type 4 bits	RFU 1 bit	ChSel 1 bit	TXAdd 1 bit	RXAdd 1 bit	Length 8 bits
---------------------------	---------------------	-----------------------	-----------------------	-----------------------	-------------------------

- RFU: Reserved for future use
- ChSel: 1 if LE Channel Selection Algorithm #2 is supported
- TXAdd: 0 if address is public, 1 if random
- RXAdd: 0 if the target's address is public, 1 if random
- Length: Payload length

Advertising PDU payload

- Advertising PDU:

Header
16 bits

Payload
(1-255 octets)

ADV_IND payload

AdvA
(6 octets)

AdvData
(0-31 octets)

ADV_NONCONN_IND payload

AdvA
(6 octets)

AdvData
(0-31 octets)

ADV_DIRECT_IND payload

AdvA
(6 octets)

TargetA
(6 octets)

ADV_SCAN_IND payload

AdvA
(6 octets)

AdvData
(0-31 octets)

AdvA – Advertiser address

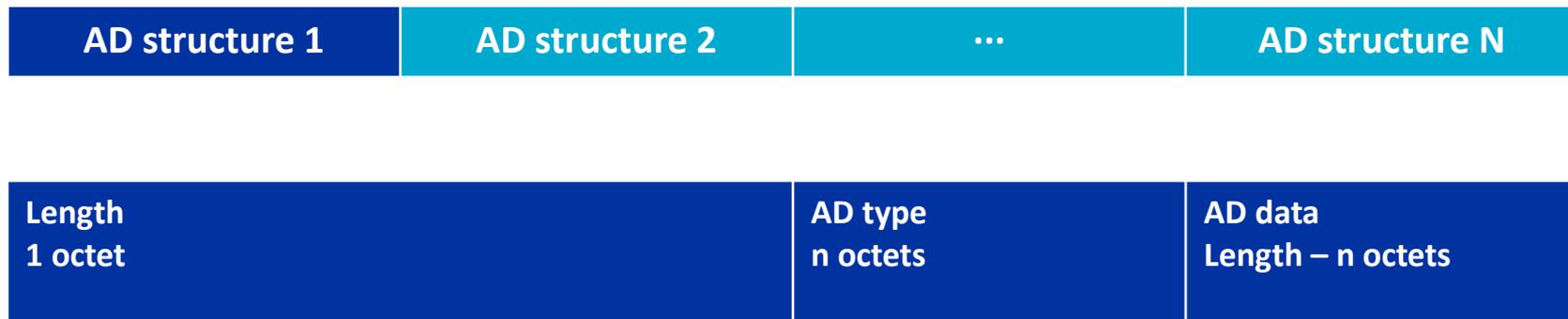
- Public (TXAdd = 0)
 - Unique address that can be recognized
- Random (TXAdd = 1)
 - Static
 - Private
 - Non-resolvable
 - Resolvable

Advertising data format

- Advertising PDU:



Advertising data format



- With ADV_IND, ADV_NONCONN_IND and ADV_SCAN_IND the sum of the structures can be maximum 31 bytes, **with overhead**

Advertising data format – AD type

- AD type data format is defined in Part A in Bluetooth Core Specification Supplement (CSS)
- Most common AD types:
 - Service UUID
 - Local Name
 - Flags
 - Manufacturer Specific Data

Advertising data format – AD type

- Service UUID
 - Typically included in connectable advertisements so that Centrals know supported service(s) without connecting. For example should a Heart Rate Sensor include the Heart Rate Service UUID
- Local Name
 - Typically included in connectable advertisements so that a human user can select to connect. For example «Heart rate sensor».
 - Can be shortened (0x08) or complete (0x09)

Advertising data format – AD type

- Flags (0x01)
 - Shall be included when any of the Flag bits are non-zero and the advertising packet is connectable (ADV_IND)
 - 5 bits:
 - Limited Discoverable Mode
 - General Discoverable Mode
 - BR/EDR Not Supported
 - Simultaneous LE and BR/EDR to Same Device Capable (Controller)
 - Simultaneous LE and BR/EDR to Same Device Capable (Host)

Advertising data format – AD type

- Manufacturer Specific Data (0xFF)
 - Typically used to include custom data
 - The two first octets shall contain a company identifier from the company identifier assigned numbers (free to obtain for Bluetooth SIG members)

Advertising Extensions

A large, white, parabolic satellite dish antenna is shown from a low angle, looking up against a clear, bright blue sky. The dish is supported by a complex metal structure with various beams and ladders. The lighting is bright, suggesting a sunny day. The overall composition is clean and modern.

Why use Advertising Extensions?

- Increases advertising data length
- Allows advertising on data channels
- Enables long range connection establishment
- Chaining
- Periodic advertising



Increases Advertising data length

- Legacy Advertising

- 2 bytes header
- 37 bytes payload
- 31 bytes advertising data



- Advertising Extensions

- 2 bytes header
- 255 bytes payload
- 254 bytes advertising data
- 0-63 extended header



Advertising on data channels

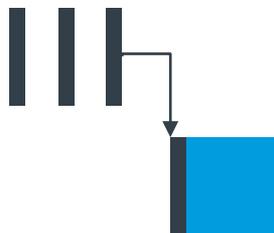
- Legacy Advertising

- Complete payload repeated on the advertising channels



- Advertising extensions

- Header is repeated on the advertising channels
- Payload is only transmitted once



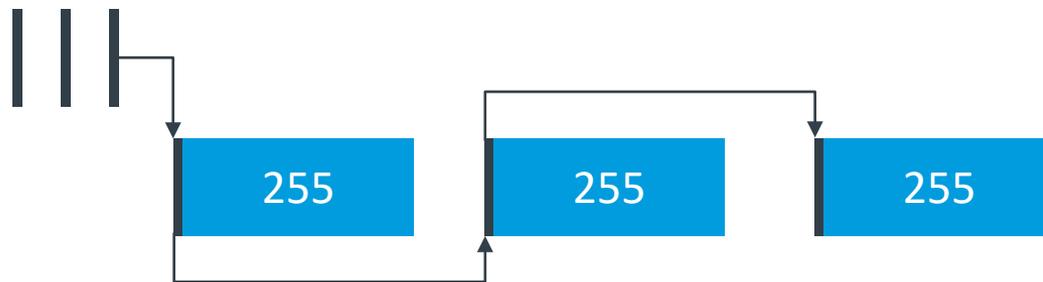
Advertising on data channels

- Longer packets and coding
- Congested advertising channels
- Reduces contention and duty cycle



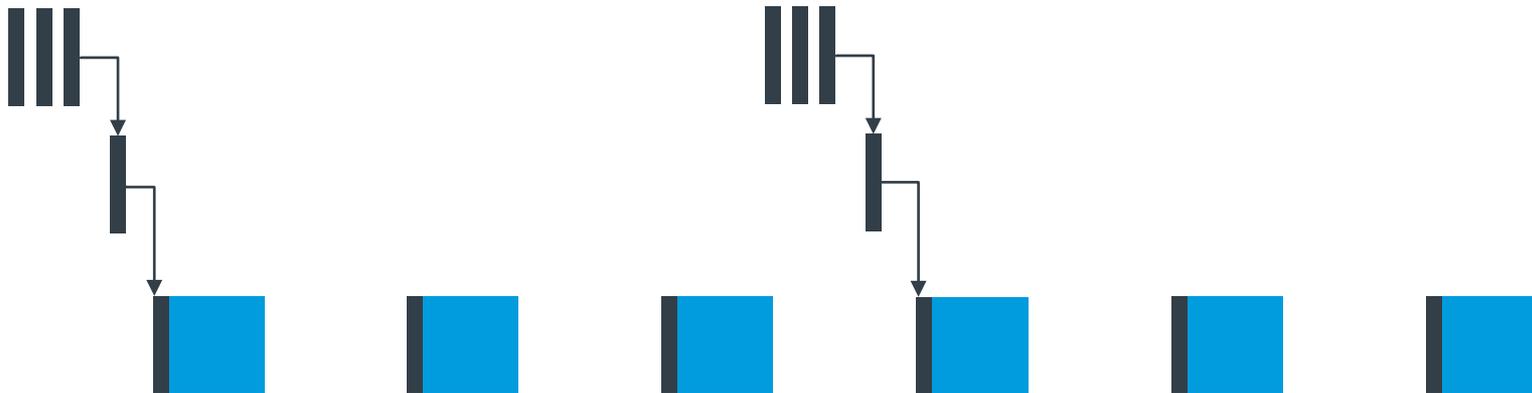
Chaining

- Advertisements can be chained together to extend the amount of advertising data



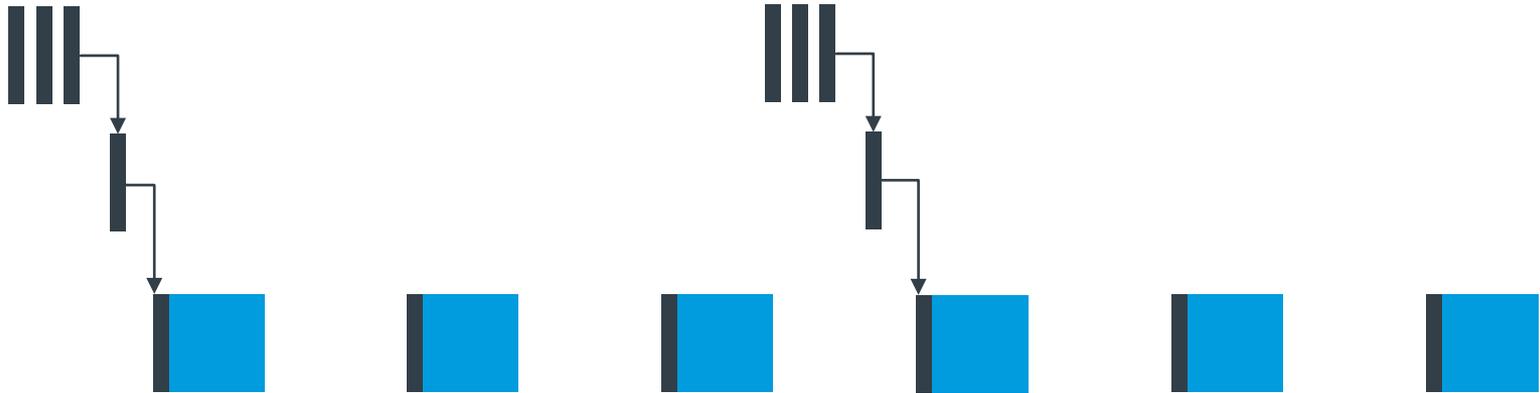
Periodic Advertising

- Enables synchronized broadcasting of advertising data
- Happens at a deterministic interval allowing true connectionless broadcasting



Periodic Advertising

- Connectionless Direction Finding



A lighthouse at night with its light shining out against a dark blue sky. The lighthouse is white with a black top section. The light is bright yellow and orange, and it is shining out in several beams. The sky is dark blue with some faint clouds. The lighthouse is the central focus of the image.

nRF Connect SDK

API, example walkthrough and demo

Demo setup

nRF Connect SDK

The image is a screenshot of the Nordic Semiconductor website. At the top left is the Nordic Semiconductor logo. The navigation bar includes links for Products, Software and tools, News, Support, About us, and Investors, along with a search icon. Below the navigation bar is a banner with the text "A suite of tools for all your development needs". On the left side of the banner, there is a section titled "Your development partner" with the subtitle "Advanced development tools for wireless developers". A dropdown menu is open under "Software and tools", listing various categories: Development kits, Prototyping platforms, Software, Cellular IoT, Bluetooth Low Energy (highlighted with a blue arrow), Thread, Zigbee, ANT, and 802.15.4. To the right of the dropdown menu, there is a section titled "Featured Bluetooth LE SDKs" which lists: nRF5 SDK, nRF Connect SDK (highlighted in blue), nRF5 SDK for Mesh, and nRF5 SDK for HomeKit. Below this is another section titled "Featured Bluetooth LE protocol stacks" which lists: SoftDevice S112, SoftDevice S113, and SoftDevice S122.

Products ▾ Software and tools ▾ News ▾ Support ▾ About us ▾ Investors ▾

A suite of tools for all your development needs

Your development partner
Advanced development tools for wireless developers

Software and tools

Development kits	+
Prototyping platforms	+
Software	+
Cellular IoT	+
Bluetooth Low Energy	-
Thread	+
Zigbee	+
ANT	+
802.15.4	+

Featured Bluetooth LE SDKs

- nRF5 SDK
- nRF Connect SDK**
- nRF5 SDK for Mesh
- nRF5 SDK for HomeKit

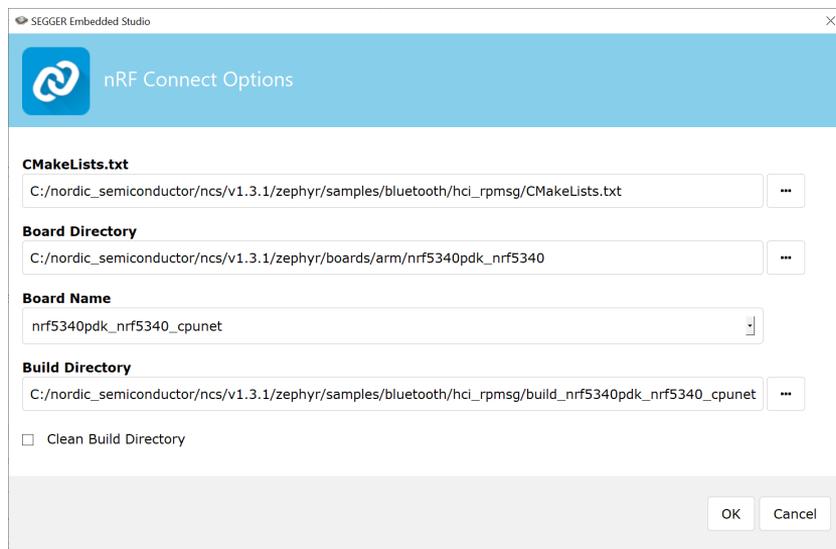
Featured Bluetooth LE protocol stacks

- SoftDevice S112
- SoftDevice S113
- SoftDevice S122

SEGGER Embedded Studio

Preparing the network core of nRF5340

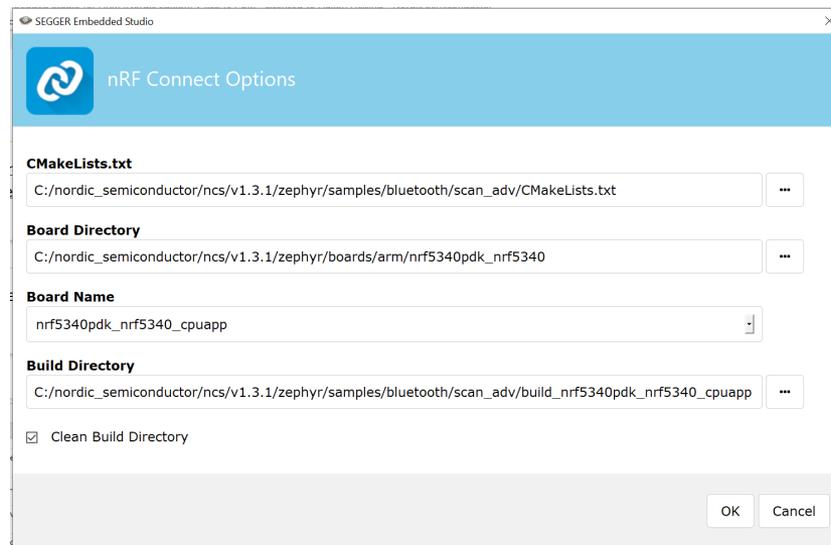
- nRF 52 series devices may skip this step
- Prepare nRF5340 PDK with the *hci_rpmsg* sample
- Select the Network core as shown
- Build and run the sample



SEGGER Embedded Studio

Project setup

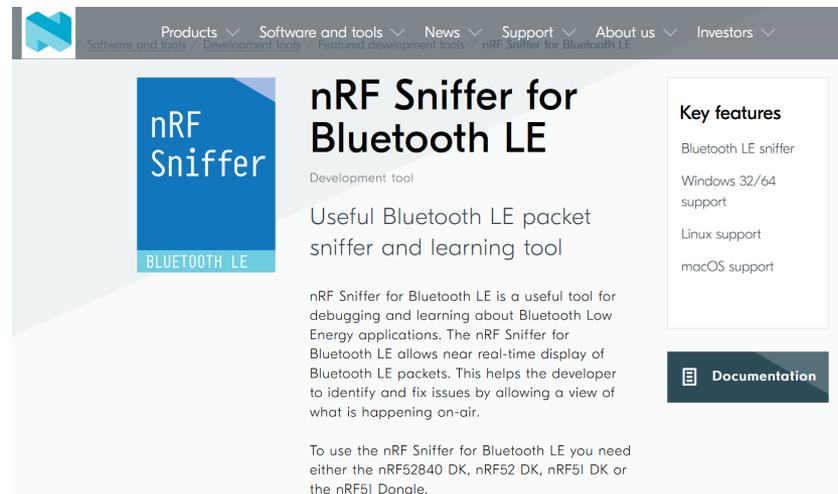
- Locate the *scan_adv* sample
- Select your board directory
- Select the Application core
- Click 'Build and run' to flash the sample



Demo

nRF Sniffer for Bluetooth LE

- Find the download and instructions at nordicsemi.com



The screenshot shows the product page for the nRF Sniffer for Bluetooth LE. The page features a navigation bar with links for Products, Software and tools, News, Support, About us, and Investors. The main content area includes a product image, a title, a description, and a list of key features. A documentation button is also visible.

Products ▾ Software and tools ▾ News ▾ Support ▾ About us ▾ Investors ▾
Software and tools / Development tools / Featured development tools / nRF Sniffer for Bluetooth LE

nRF Sniffer for Bluetooth LE

Development tool

Useful Bluetooth LE packet sniffer and learning tool

nRF Sniffer for Bluetooth LE is a useful tool for debugging and learning about Bluetooth Low Energy applications. The nRF Sniffer for Bluetooth LE allows near real-time display of Bluetooth LE packets. This helps the developer to identify and fix issues by allowing a view of what is happening on-air.

To use the nRF Sniffer for Bluetooth LE you need either the nRF52840 DK, nRF52 DK, nRF51 DK or the nRF51 Dongle.

Key features

- Bluetooth LE sniffer
- Windows 32/64 support
- Linux support
- macOS support

[Documentation](#)

Demo

Examples

- Example 1: scan_adv sample, unmodified from Zephyr Project
- Example 2: Static address, name, Company ID
- Example 3: Advertising interval, set address
- Example 4: Scan response, send more data(128-bit UUID)

Example 1

scan_adv

Example 1 - scan_adv sample

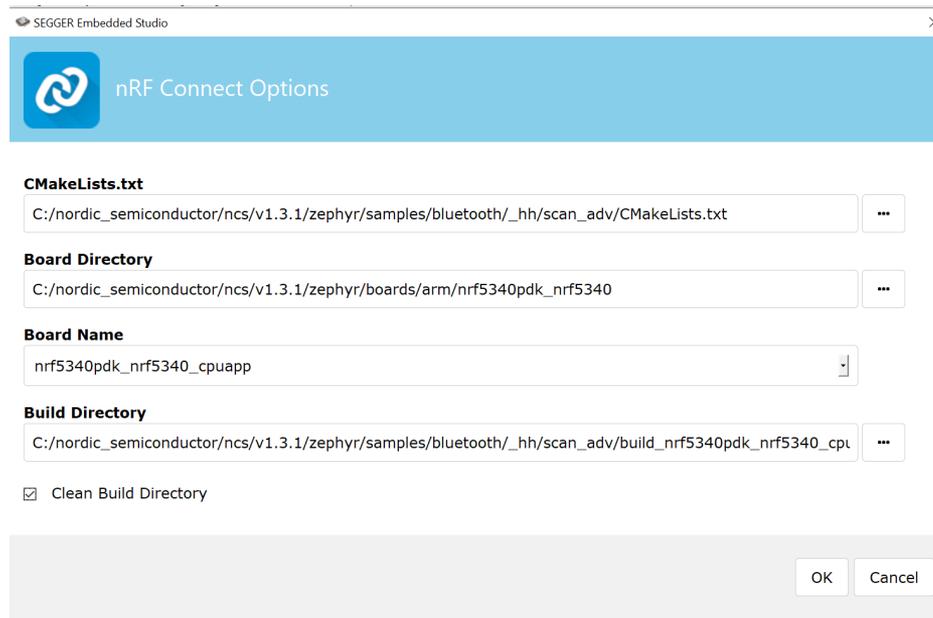
- [nRF Sniffer for Bluetooth LE window]

Example 2

Static address, name, Company ID

Example 2

- [SES-window]



Example 2

- [Wireshark window]

Filtering

btle.length != 0

PACKET LIST

No.	Time	Source	PHY	Protocol	Length	Delta time (µs end to start)	SN	NESN	More Data	Event counter	Info
16921	439.077	44:95:6f:dc:2a:bc	LE 1M	LE LL	35		266				0 ADV
16922	439.079	44:95:6f:dc:2a:bc	LE 1M	LE LL	35		266				0 ADV
16923	439.080	4b:f4:22:c0:c6:1d	LE 1M	LE LL	12		150				0 SCAL
16924	439.082	44:95:6f:dc:2a:bc	LE 1M	LE LL	37		150				0 SCAL
16925	439.182	44:95:6f:dc:2a:bc	LE 1M	LE LL	35		107004				0 ADV
16926	439.183	4b:f4:22:c0:c6:1d	LE 1M	LE LL	34		150				0 COMI
16927	439.196	Master_0xaf9a9d63	LE 1M	LE LL	6		13251	0	0	False	0 Con
16928	439.226	Master_0xaf9a9d63	LE 1M	LE LL	6		29872	0	0	False	1 Con
16929	439.227	Slave_0xaf9a9d63	LE 1M	LE LL	9		150	0	1	True	1 Con
16931	439.228	Slave_0xaf9a9d63	LE 1M	LE LL	6		150	1	0	True	1 Con

> Frame 16926: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

PACKET DETAILS

Nordic BLE Sniffer

- Board: 18
- Header Version: 2, Packet counter: 29749
- Length of packet: 10
- Flags: 0x01
- Channel: 37
- RSSI (dBm): -52
- Event counter: 0
- Delta time (µs end to start): 150
- [Delta time (µs start to start): 510]
- Bluetooth Low Energy Link Layer
 - Access Address: 0x8e89bed6
 - Packet Header: 0x22c5 (PDU Type: CONNECT_REQ, ChSel: #1, TxAdd: Random, RxAdd: Random)
 - Initiator Address: 4b:f4:22:c0:c6:1d (4b:f4:22:c0:c6:1d)
 - Advertising Address: 44:95:6f:dc:2a:bc (44:95:6f:dc:2a:bc)
 - Link Layer Data
 - Access Address: 0xaf9a9d63
 - CRC Init: 0xcf76b0
 - Window Size: 3 (3.75 msec)
 - Window Offset: 9 (11.25 msec)
 - Interval: 24 (30 msec)
 - Latency: 0
 - Timeout: 72 (720 msec)
 - Channel Map: ffffffff
 - ...0 0101 = Hop: 5
 - 001. = Sleep Clock Accuracy: 151 ppm to 250 ppm (1)
 - CRC: 0x01b585

PACKET BYTES

Packet info as:
 - hexadecimal (left)
 - ASCII (right)

```

0000 12 35 00 02 35 74 06 0a 01 25 34 00 00 96 00 00 5 - 5t...%4....
0010 00 d6 be 89 8e c5 22 1d c6 c0 22 f4 4b bc 2a dc .....K...
0020 6f 95 44 63 9d 9a af b0 76 cf 03 09 00 18 00 00 oDc...v...L.
0030 00 48 00 ff ff ff ff 1f 25 80 ad a1 ..H.....%*..
    
```

Wireshark filter for connection interval
 btle.connect.interval

Interval (btle.link_layer_data.interval), 2 bytes

Packets: 20560 - Dis

Example 3

Advertising interval and set address

Example 3

- [SES-window]

Example 3

- [Wireshark window]

Example 4

Scan response, send more data(128-bit UUID)

Example 4

- [SES-window]

Example 4

- [Wireshark window]

Q&A