Bluetooth® Core 5.4

Technical Overview

Bluetooth® Core Specification v5.4 (Bluetooth® Core 5.4) includes several updates. This document summarizes and explains each change.

Author: Martin Woolley

Version: 1.1.1

Revision Date: 13 January 2025

1. Revision History

Version	Date	Author	Changes
1.0.0	7 February 2023	Martin Woolley	Initial Version
1.1.1	13 January 2025	Avi Negrin	Language Changes

Table of Contents

Table of Contents

Table of Contents

At a Glance

Periodic Advertising with Responses (PAwR)

PAwR is a new Bluetooth® Low Energy (LE) logical transport that provides a way to perform energy-efficient, bi-directional, communication in a large-scale one-to-many topology.

Encrypted Advertising Data

This new feature provides a standardized approach to the secure broadcasting of data in advertising packets.

LE GATT Security Levels Characteristic

Devices may now indicate the security mode and level required for all their GATT functionality to be available using a new GATT characteristic called *LE GATT Security Levels*.

Advertising Coding Selection

The Host can now specify which of two supported long range coding options are used with Extended Advertising.

1. Periodic Advertising with Responses

1.1 Background

1.1.1 Modes of Operation

The Bluetooth® Core Specification defines several concepts that collectively constitute the Bluetooth data transport architecture. Among these concepts are the Physical Transport, Physical Channel, Physical Link, Logical Link, and Logical Transport. Certain combinations and configurations are defined for use in support of different application types, each with particular characteristics relating to properties such as topology, timing, reliability, power, and channel use. The terms *operational mode* or *mode of operation* are sometimes used informally to refer to the various data transport architecture configurations.

Bluetooth LE supports several modes of operation, several of which provide ways of performing connectionless communication, with a transmitting device *advertising* and one or more receiver devices *scanning.* Advertising involves transmitting packets of data at regular or irregular intervals.

Two forms of scanning are defined: *passive scanning* and *active scanning*. Passive scanning involves the receiver scanning for and receiving advertising packets and not transmitting packets in response. When performing active scanning, the receiver may respond to *scannable advertising*[1] Protocol Data Units (PDUs) by transmitting *scan request* PDUs. Scan request PDUs represent a request for more information. The advertiser will respond to such requests with a scan response PDU containing additional application layer data.

1.1.2 Fundamental Properties of Communication Systems

The Bluetooth modes of operation can be compared using a number of the fundamental properties of communication systems. It is informative to consider these properties when evaluating the best way to use Bluetooth technology to meet the communication requirements of a new product or application.

A selection of significant properties appears in Table 1.

1 *See section 1.1.3 Advertising Modes and Basic Properties.*

Property	Description
Topology	Topology is concerned with the cardinality of the relationships which may be formed between communicating devices. Three distinct topologies are recognized; **one-to-one** (1:1), **one-to-many** (1:m) and **many-to-many** (m:n).
Direction - Packets	Communication involves the transmission and receipt of *packets*. With some modes of operation and configurations, packets travel in one direction only between communicating devices (unidirectional), while in others, there is a bidirectional exchange of packets.
Direction - Data	Some transmitted packets can contain data from the higher layers of the Bluetooth protocol stack (e.g., the application layer), and some cannot. Some modes of operation support the bidirectional communication of data, while in other cases, the transfer of data is unidirectional, even when bidirectional packet exchanges are taking place.
Communication Method	Connection-oriented communication involves initialization procedures that allow devices to prepare for communication by negotiating various parameter values. The agreed parameters then control certain aspects of communication, such as the timing of transmit and receive slots. In connectionless communication, there is no negotiation stage.
Data / Time Relationship	The Bluetooth Core Specification defines three different relationships between logical transports, the data to be transported, and time, and they are referred to with the terms *asynchronous*, *synchronous,* and *isochronous*. When performing *asynchronous* communication, the data to be transported has no special relationship with or dependency on time. Packets will be transmitted one or more times until received. It is important that all data is delivered, and retransmission schemes are used to support this, but it is not crucial *when* the data is delivered. Receivers may know the remote device's transport *packet* transmission schedule but not when to expect *data* from upper layers to be delivered by the transport in such packets. When performing *synchronous* communication, the data to be transported is time-dependent and passed to the link layer for transmission according to a schedule. Packets are of a fixed size, and data rates are constant. Packets not delivered within applicable time constraints are said to expire and are flushed. Isochronous communication is similar to synchronous communication. Data is time-bound and must be delivered within specific time constraints. Packets may vary in size, so that data rates may be variable. Packets not delivered within applicable time constraints are said to expire and are flushed.

<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td>Receiver Concurrency</td><td>A one-to-many topology is sometimes described or depicted as a hub-and-spoke arrangement of devices. The hub device can transmit data to each of the other devices, one device at a time in series, or it can transmit in such a way that some or all of the devices receive the same transmitted data at precisely the same time. In the first example, despite the one-to-many topology, the communication is effectively serialized in time across receivers. In contrast, in the other instance, the same data is delivered concurrently to a set of receivers.

Three receiver concurrency types are identified:

Single - one receiver at a time receives data in separate transmissions

Subset - a selection of all available receivers receive the same transmission

All - all receivers in range receive the same transmission</td></tr>
<tr><td>RF Channels</td><td>Bluetooth® LE divides the ISM band into forty 2 MHz-wide channels. Different operating modes and configurations use different subsets of these channels. Specific subsets are defined and referred to with the following names:

<table><tr><th>Channel Subset Name</th><th>Channels by Index</th></tr><tr><td>Primary advertising channels</td><td>37, 38 and 39</td></tr><tr><td>General-purpose channels</td><td>0 to 36</td></tr></table></td></tr>
<tr><td>Scalability</td><td>There are several possible interpretations of this term, which depend on the context. For example, when considering 1:m topologies, the maximum size of m may be the primary interest. In other scenarios, it may be that scaling up to some overall maximum data transfer rates or messages per second is the scalability concern.</td></tr>
<tr><td>Choice of PHY</td><td>Bluetooth® LE defines three physical layer variants known as PHYs.

LE 1M uses a symbol rate of 1 Msym/s with a required frequency deviation of at least 185 kHz. All devices must support LE 1M.

LE 2M is similar to LE 1M but uses a symbol rate of 2 Msym/s and has a required frequency deviation of at least 370 kHz. Support for LE 2M is optional.

LE Coded uses a symbol rate of 1 Msym/s. Packets are subject to a coding called Forward Error Correction (FEC) and, depending on configuration, a pattern mapping. This increases the effective range of transmissions but reduces the application data rate. Support for LE Coded is optional.

Support for each of the three PHYs varies according to the mode of operation and sometimes by the PDU type to be transmitted.</td></tr>
</table>

Table 1 - Fundamental properties of a communication system

1.1.3 Advertising Modes and Basic Properties

Advertising is a form of *connectionless* communication that, depending on how it is performed, has various properties that affect the behaviors of the advertising device and other devices that receive its transmitted advertising packets.

Property	Description
connectable vs. non-connectable	Connectable advertising means that a scanning device may respond to a received advertising packet by transmitting a request to form a connection with the advertising device.
scannable vs non-scannable	Scannable advertising means scanning devices may respond to an advertising packet by transmitting a *scan request*, asking for more application data from the advertiser.
directed vs. non-directed	When performing directed advertising, packets are addressed to a specific scanning device and will be ignored by other devices. Non-directed advertising packets are not addressed to a specific device and may be processed by any scanning device.
Irregular vs. fixed interval periodic	Advertising can be performed with a precise transmission schedule, and this is known as *periodic advertising*. Other forms of advertising operate to an irregular schedule. A random delay value between 0 and 10 ms is added to a fixed advertising interval to perturb advertising events in time.

Specific combinations of these properties are defined together with the circumstances in which they may be used, in the Bluetooth Core Specification. The selected combination is indicated by the type(s) of PDU transmitted or by the value of a field called AdvMode, which is present in some PDU types. Examples of defined combinations include *connectable undirected advertising* and *connectable and scannable advertising*.

1.1.4 GAP Roles

The Generic Access Profile (GAP) defines four roles.

A **Broadcaster** transmits advertising packets that are never connectable but may be scannable.

An **Observer** uses scanning to receive advertising packets from a Broadcaster.

A **Central** can initiate the creation of a connection with a Peripheral.

A **Peripheral** can accept a request to establish a connection sent by a Central.

The Bluetooth Core Specification allows a device to operate concurrently in multiple GAP roles. However, this is not mandatory; a specific device's ability to do this depends on the Controller's implemented features and capabilities.

In the context of Periodic Advertising with Responses (PAwR) in this paper, the term Broadcaster refers to an advertising device and Observer to any device which scans to receive advertising packets. The term *advertiser* may be used from time to time when advertising is being discussed in more general terms. The terms Central and Peripheral are used when the context relates to establishing connections.

1.1.5 The Evolution of Bluetooth Advertising

The first specification of Bluetooth LE technology[2] defined one form of advertising only. That original advertising mode is known as *legacy advertising*.

Legacy advertising is now accompanied in the Bluetooth Core Specification by a more sophisticated form of advertising called *extended advertising*.

1.1.5.1 Legacy Advertising

When performing legacy advertising, identical copies of legacy advertising packets are transmitted on up to three primary advertising channels, one channel at a time and in some pseudo-random sequence.

The transmission of legacy advertising packets takes place during *advertising events*. The scheduling of advertising events is primarily controlled by a link layer timing parameter, *advInterval.* But advertising events are made slightly irregular, so persistent collisions with other advertising devices are avoided. This is achieved by assigning a parameter known as *advDelay,* a pseudo-random value in the range of 0 - 10ms, and adding it to the fixed *advInterval* so that advertising events are perturbed in time. Figure 1 illustrates this.

2 Version 4.0 of the Bluetooth Core Specification

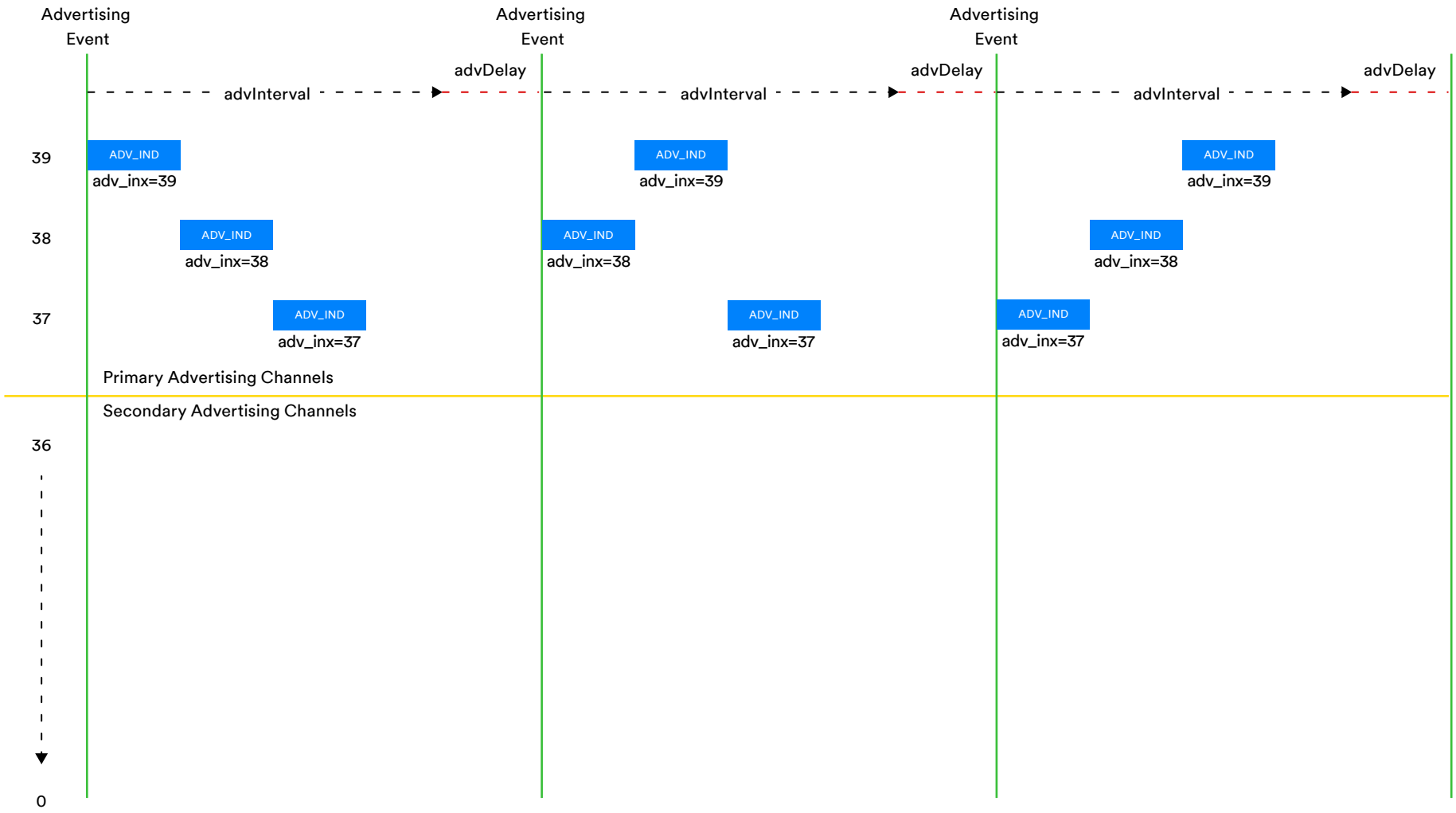

Figure 1 - Legacy advertising by channel index

Legacy advertising packets can contain application data in the *AdvData* field, but scan request packets cannot. Therefore the direction of packet transmissions can be bidirectional, but the transfer of application data using advertising and scan response PDUs is a strictly one-way capability.

1.1.5.2 Extended Advertising

Before Bluetooth® Core Specification v5.4 (Bluetooth® Core 5.4), two logical transports that involve advertising were defined. Figure 2 illustrates these logical transports and the advertising modes they support.

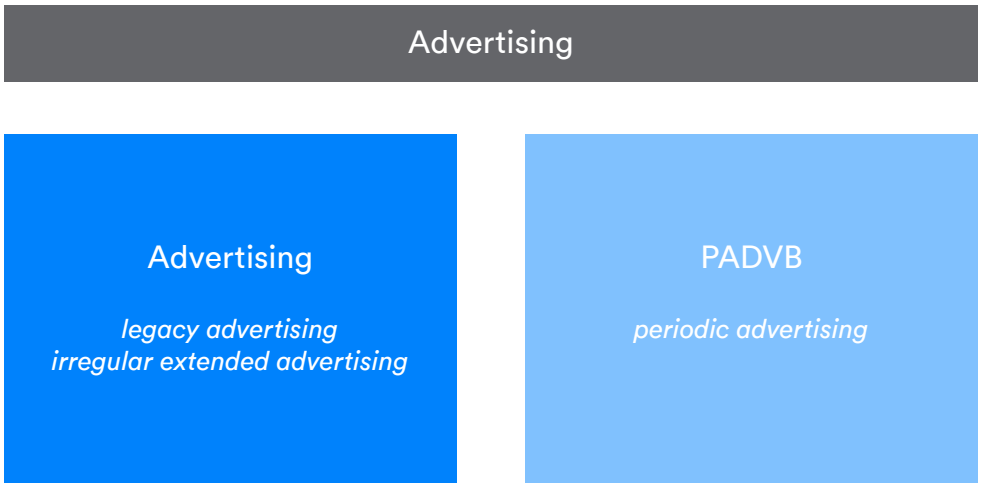

Figure 2 - logical transports and advertising modes

The *LE Advertising Broadcast* (ADVB) logical transport incorporates two advertising modes: legacy advertising and one of two ways of performing extended advertising. Both legacy advertising and the form of extended advertising associated with the ADVB logical transport involve irregular scheduling of events. This use of extended advertising is referred to here as *irregular extended advertising*.

The *LE Periodic Advertising Broadcast (PADVB)* logical transport is another form of extended advertising but is designated a distinct logical transport because it uses regular, fixed-rate timing for event and packet transmission scheduling.

Irregular extended advertising and *periodic advertising* both use the 37 general-purpose channels and the three primary advertising channels. In this context the general-purpose channels are sometimes referred to as the *secondary advertising channels*. A frequency hopping pattern, calculated using an algorithm known to both the Broadcaster and Observer devices, is used in both cases.

1.1.5.2.1 Irregular Extended Advertising

Irregular extended advertising is comparable with legacy advertising in that some extended advertising PDU types are only ever transmitted on the primary advertising channels. Their transmission scheduling is irregular due to the addition of the random advDelay value in the range of 0 to 10 ms in calculating advertising event times. It differs from legacy advertising in a number of ways including that distinct PDU types are used. Some of these PDU types are transmitted only on the primary advertising channels, but may be linked by a pointer field called *AuxPtr* to others that are transmitted only on the secondary advertising channels. Larger payloads may be handled by fragmenting the data and transmitting it in multiple PDUs linked together or *chained* using *AuxPtr* in certain PDU types.

ref core spec Link Layer Figure 4.25

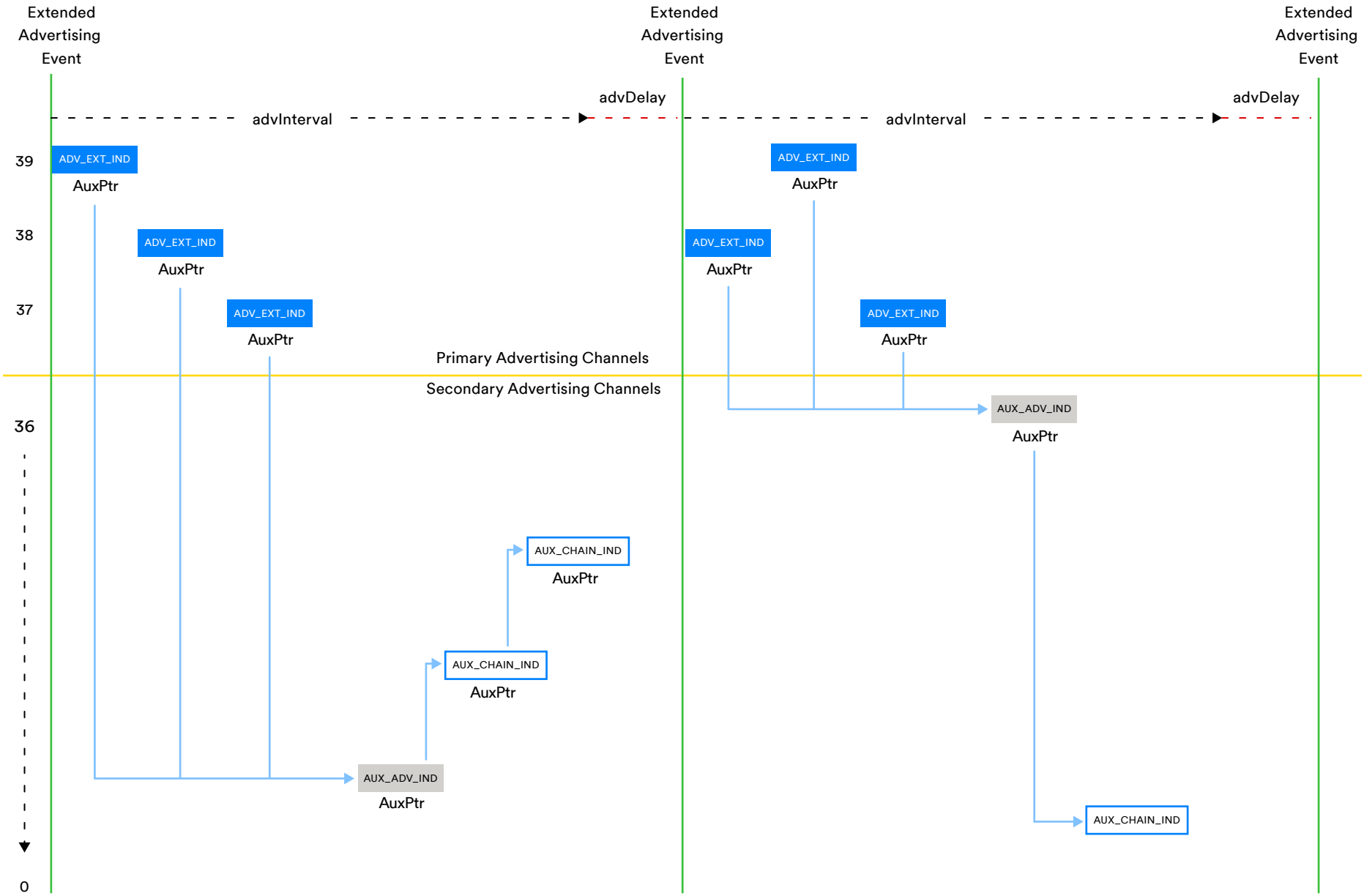

Figure 3 - Irregular extended advertising

1.1.5.2.2 Periodic Advertising

When the periodic advertising (PADVB) logical transport is used, the broadcasting device transmits packets to a fixed interval, deterministic schedule, and Observer devices can discover the Broadcaster's transmission schedule and synchronize their scanning to it. This can be accomplished by acquiring information from the SyncInfo field within an AUX_ADV_IND PDU or by using a procedure called Periodic Advertising Sync Transfer (PAST).

PAST involves a device passing periodic advertising synchronization parameter values over an LE-ACL connection to the Observer. The device passing these details may be the Broadcaster itself or a third device that acquires the synchronization parameters by scanning for AUX_ADV_IND PDUs transmitted by the Broadcaster. The procedure avoids the need for the Observer, which may be a small, power-constrained device, to scan for synchronization data itself, a potentially expensive operation.

By precisely synchronizing with the Broadcaster's transmission schedule, the Observer can scan in the most energy-efficient way.

Periodic advertising involves multiple extended advertising PDU types and all 40 radio channels, as depicted in Figure 4. Here we see ADV_EXT_IND PDUs transmitted on the primary advertising channel, with AuxPtr pointing to an associated AUX_ADV_IND PDU transmitted on the secondary channels. This PDU contains the SyncInfo field which contains information that allows an Observer to synchronize its scanning with the periodic transmission of AUX_SYNC_IND PDUs. Note that the Observer only needs to receive one ADV_EXT_IND PDU to be able to acquire the periodic advertising synchronization data in the SyncInfo field of a AUX_ADV_IND PDU. Once this has been achieved, it can proceed to only scan for AUX_SYNC_IND PDUs

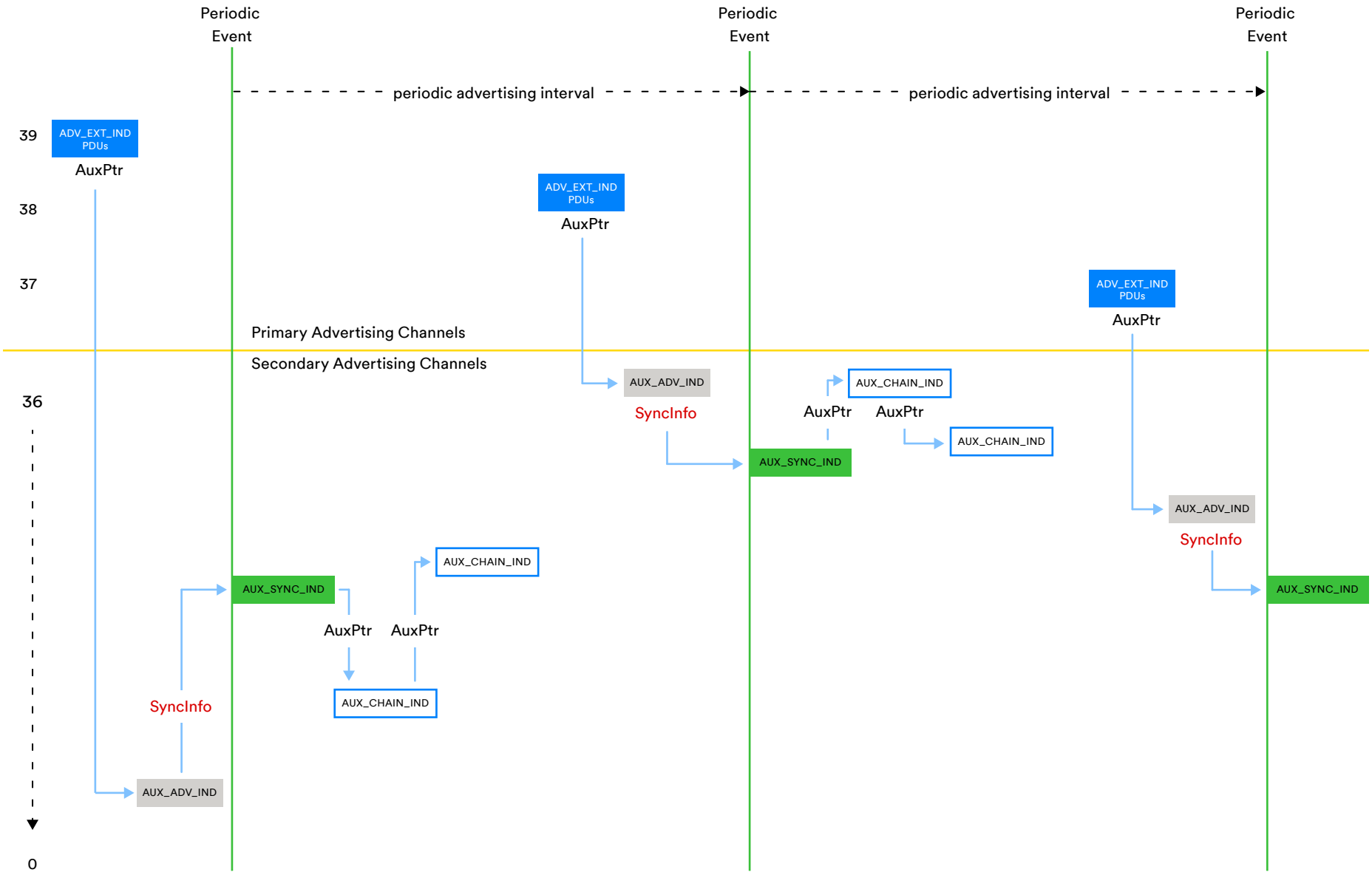

Figure 4 - Periodic Advertising (PADVB) PDUs and Channel Use

The AUX_SYNC_IND advertising PDU type is transmitted at fixed intervals. It is not possible for Observers to respond to PADVB periodic advertising PDUs and hence this logical transport only supports unidirectional communication of application data.

1.1.5.3 Comparing Legacy Advertising and Extended Advertising

Table 2 summarizes some of the important differences between legacy advertising and extended advertising.

	Legacy Advertising	Extended Advertising	
Max. host advertising data size	31 bytes	1,650 bytes	Extended Advertising supports *fragmentation*, enabling a 50x larger maximum host advertising data size to be supported.
Max. host advertising data per packet	31 bytes	254 bytes	Extended Advertising PDUs use the *Common Extended Advertising Payload Format,* which supports an 8x larger advertising data field.
TX channels	37,38,39	0-39	Extended Advertising uses the 37 general-purpose channels as secondary advertising channels. However, the ADV_EXT_IND PDU type may only be transmitted on the primary advertising channels.
PHY support	LE 1M	LE 1M LE 2M (excluding ADV_EXT_IND PDUs) LE Coded	All Extended Advertising PDUs may be transmitted using any of the three LE PHYs except for the ADV_EXT_IND PDU, which may be transmitted using LE 1M or LE Coded.
Max. active advertising configurations	1	16	Extended Advertising includes *Advertising Sets* that enable advertising devices to support up to 16 different advertising configurations at a time and to interleave advertising for each advertising set according to time intervals defined in the sets.
Transmission timing	Irregular	Irregular and Regular (periodic)	Extended Advertising includes *Periodic Advertising*, enabling time-synchronized communication of advertising data between transmitters and receivers.

Table 2 - Comparing legacy and extended advertising

1.2 About Periodic Advertising with Responses (PAwR)

1.2.1 Overview

PAwR is similar to periodic advertising (PADVB) in several ways:

- PADVB allows application data to be transmitted by one device (the Broadcaster) to one or more receiving devices (the *Observers*), forming a one-to-many communication topology. The same is true of PAwR.
- PAwR and PADVB both use a connectionless communication method.
- Transmission of advertising packets is periodic with a fixed interval and no random perturbation of the schedule in both cases.
- Observers can establish the periodic transmission schedule used by the Broadcaster from AUX_ADV_IND PDUs or by using the Periodic Advertising Sync Transfer (PAST) procedure.

PAwR differs from PADVB as follows:

- PADVB supports the unidirectional communication of data from a Broadcaster to Observers only. PAwR Observers can transmit response packets back to the Broadcaster. PAwR provides a *bidirectional*, connectionless communication mechanism.
- Synchronization information for periodic advertising *without* responses (PADVB) is contained within the SyncInfo field of AUX_ADV_IND PDUs. Synchronization information for periodic advertising *with* responses (PAwR) is contained within the SyncInfo field and in the ACAD field of AUX_ADV_IND PDUs.
- The PADVB Broadcaster schedules transmissions within advertising *events*. The PAwR Broadcaster schedules transmissions in a series of events *and subevents,* and Observers are expected to have synchronized in such a way as to listen during a specific subevent or subevents only.
- The PAwR Broadcaster may use a transmission time slot to send a connection request (AUX_CONNECT_REQ) to a specific device and establish an LE-ACL connection with it. PADVB does not have this capability.
- With periodic advertising without responses (PADVB), application data tends to only change from time to time. PAwR is designed with the expectation that application data will change frequently.
- With PADVB, the same application data is delivered to all Observer devices synchronized to the same advertising set. With PAwR, different data can be delivered to each Observer device or set of Observer devices.
- Support for the Periodic Advertising Sync Transfer (PAST) procedure is optional with PADVB but mandatory with PAwR.

1.2.2 Benefits

1.2.2.1 Bidirectional Connectionless Communication

PAwR supports the *bidirectional* exchange of application data using connectionless communication, which was not previously possible with Bluetooth LE.

1.2.2.2 Scalability

PAwR offers a more scalable way to create a one-to-many topology capable of bidirectional application data communication than a collection of LE-ACL connections. It is common for no more than a handful of concurrent LE-ACL connections to be supported by a Central device[3] , whereas bidirectional communication with thousands of devices is possible using PAwR.

1.2.2.3 Energy Efficiency

Synchronization between the Broadcaster and each Observer takes place at subevent level, meaning that Observers only scan during a small subset of all Broadcaster transmissions. The subevent synchronization process involves application logic, so packets received will usually contain data of relevance to the Observer. This energy-efficient approach means Observer devices could run off batteries for several years. Section *1.2.3.6 Battery Life* illustrates.

1.2.2.4 Flexible Topologies and Receiver Concurrency

PaWR offers flexibility in terms of the topologies supported. When a PAwR Broadcaster transmits a packet, it does so in a subevent. The packet is received by all devices synchronized to that subevent, and this may be a single device, a subset of the complete set of the Observer devices, or all such devices, depending on the application-layer rules for subevent synchronization.

With PADVB, each advertising set forms a one-to-many topology, and all devices synchronized to an advertising set receive data at each broadcast.

1.2.2.5 Applications

PAwR is well-suited to applications that need to send and receive *messages* between a central hub device and a large number of other devices in a network. Messages could contain commands or sensor data values, or other data, as defined by the application layer. The Electronic Shelf Label (ESL) profile uses PAwR to transport messages containing commands and responses and serves as a good example of PAwR in use.

PAwR is not intended to be used for products that require a real-time messaging capability. As will be explained in the *Technical Highlights* section, PAwR works by periodically transmitting a series of packets, one after the other in specific time slots known as *subevents*. Devices are configured so that they listen during certain subevents only and therefore it is common for there be a delay between the start of a use case which delivers commands or data to devices across the network, and the transmission time slot for each device arising. Consequently, a noticeable elapsed time will

3 LE-ACL scalability limits are largely an implementation concern rather than inherent in the specification details

sometimes be experienced when sending messages to multiple, unrelated devices. The elapsed time will vary in magnitude from milliseconds to tens of seconds, depending on details of the use case and system configuration.

By way of contrast, Bluetooth mesh networking also makes use of a system of messaging for commands and sensor data. However, Bluetooth mesh offers a near to real-time messaging solution, with messages sent and responded to more or less immediately. To achieve this though, mesh devices perform passive scanning with a duty cycle as close to 100 percent as possible and this has consequences for energy consumption. PAwR devices such as electronic shelf labels operate to a low duty cycle and therefore perform well in terms of energy efficiency.

1.2.3 Technical Highlights

1.2.3.1 Events, Sub-events, and Response Slots

Understanding how PAwR partitions and uses time is key to understanding this logical transport.

As with other advertising modes, activity takes place in events which in the case of PAwR are known as *Periodic advertising with responses events* (PAwR events). These events occur at fixed intervals, with no random perturbation in scheduling. An event starts every *periodic advertising interval* ms.

Each PAwR event consists of several subevents, and it is during subevents that advertising packets are transmitted. The Host configures the number of subevents per event up to a maximum of 128. A subevent starts every *periodic advertising subevent interval* ms. The Host configures the number of subevents per event and the periodic advertising subevent interval using a Host Controller Interface (HCI) command called HCI_LE_Set_Periodic_Advertising_Parameters V2 (or later).

See *Figure 5 - PAwR events and subevents*.

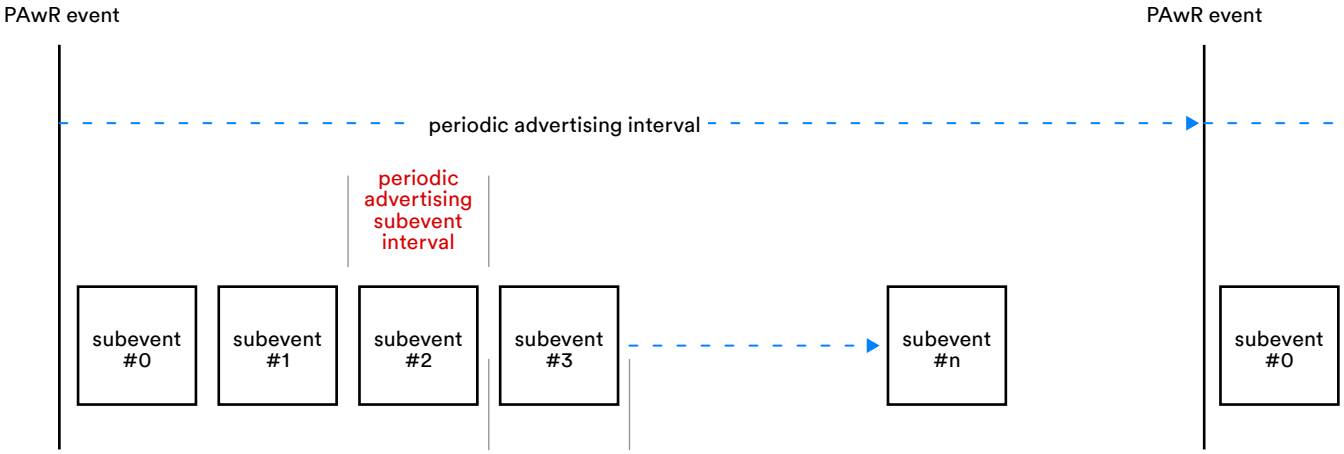

Figure 5 - PAwR events and subevents[4]

In each subevent, the Broadcaster transmits one packet, which usually contains an AUX_SYNC_SUBEVENT_IND PDU but may instead contain an AUX_CONNECT_REQ PDU. After a delay, known as the Periodic Advertising Response Slot Delay, a series of time slots are reserved within the same

4 #nse - 1 means Number of Subevents minus one

subevent for receiving responses from Observer devices. Responses to AUX_SYNC_SUBEVENT_IND PDUs are sent in AUX_SYNC_SUBEVENT_RSP PDUs. The Host configures the number of response slots required by the HCI command HCI_LE_Set_Periodic_Advertising_Parameters. Figure 6 illustrates the structure of a PAwR subevent.

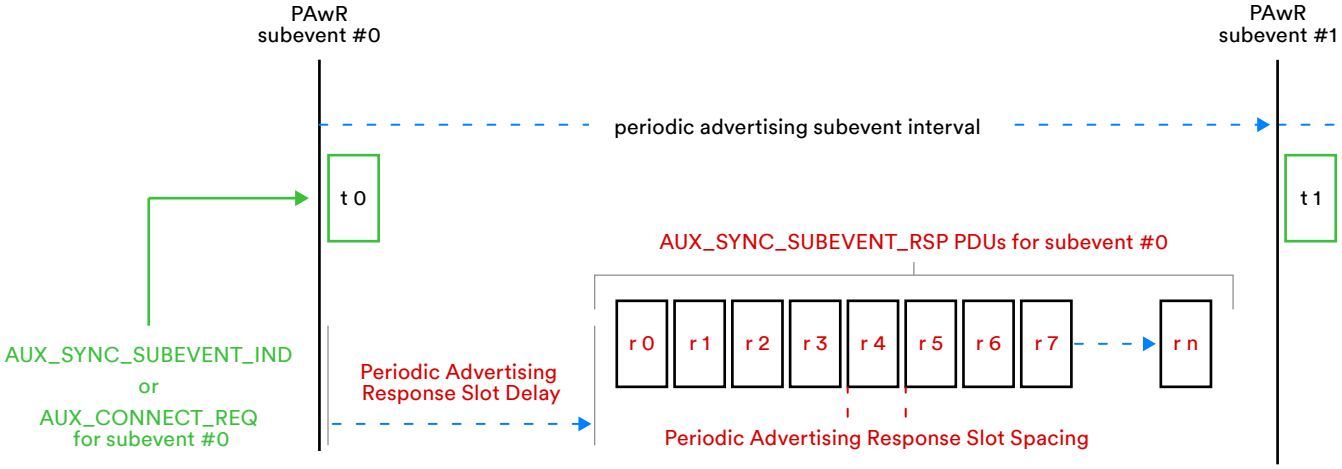

Figure 6 - A PAwR subevent with response slots

1.2.3.2 Channel Selection

Channel selection is accomplished using Channel Selection Algorithm #2, and takes place at each periodic advertising subevent. Responses to PDUs transmitted in a subevent use the same channel. This includes AUX_SYNC_SUBEVENT_RSP PDUs sent in response to a AUX_SYNC_SUBEVENT_IND PDU and AUX_CONNECT_RSP PDUs which are sent in response to AUX_CONNECT_REQ PDUs.

1.2.3.3 Synchronizing

1.2.3.3.1 General

The process of *synchronizing* provides the Observer device with the information it needs to efficiently scan for and receive relevant packets transmitted by the advertising device. In the case of PAwR, there are three aspects to this:

1. The Observer needs to know how often *periodic advertising with responses event*s will occur and when the next such event will occur. This information is provided in a parameter called the *periodic advertising interval* and a calculated value known as *syncPacketWindowOffset*.
2. The Observer needs information about *subevents,* including how often they occur and how many subevents each *periodic advertising with responses event* accommodates. It also needs to know certain details relating to the time slots within each subevent reserved for the transmission of responses. This information is contained within parameters known as *Subevent_Interval, Num_Subevents*, *Response_Slot_Delay*, *Response_Slot_Spacing*, and *Num_Response_Slots*.

3. Finally, an Observer needs to know which subevent number it should scan for, which particular response slot it should use, and the access address to use in response packets transmitted.

Having acquired the event timing information described in (1) and the subevents information in (2), the Observer has a complete description of the timing parameters and structure of the events and subevents of the PAwR advertising train. But it is only when it has the information in (3) that it can schedule its scanning such that it receives only those packets that are expected to contain data of relevance and can schedule the transmission of response packets.

(1) and (2) are dealt with by the PAwR logical transport, as defined in the Bluetooth Core Specification. There is a choice of two procedures that may be used to obtain this level of synchronization information. The two procedures are covered in this paper in sections *1.2.3.3.2 Scanning for Periodic Advertising Synchronization Information* and *1.2.3.3.3 Periodic Advertising Sync Transfer (PAST)*.

(3) must be addressed by the application layer and may be defined in an applicable Bluetooth profile specification. This is covered in section *1.2.3.3.4 Subevent Synchronization and Response Slot Allocation*.

1.2.3.3.2 Scanning for Periodic Advertising Synchronization Information

PAwR and PADVB each use a similar procedure for acquiring periodic advertising synchronization information by scanning.

With both PAwR and PADVB, an Observer scans for AUX_ADV_IND packets transmitted on the secondary advertising channels. These PDUs are pointed to by the channel index, offset and PHY information in the AuxPtr field in ADV_EXT_IND PDUs that are transmitted on the primary channels. AUX_ADV_IND includes the SyncInfo field, which contains the *periodic advertising interval* value and some data items from which to calculate a variable called *syncPacketWindowOffset*. Having acquired these two values, the Observer can calculate when *periodic advertising with responses events* will occur, per (1) in *1.2.3.3.1 General*.

PAwR also requires information about subevents and response slots, per (2) in *1.2.3.3.1 General,* before it can complete the synchronization procedure. This information is to be found in the same AUX_ADV_IND PDU from which the *periodic advertising interval* was obtained but in a new AD type called Periodic Advertising Response Timing Information. The new AD type is transmitted in the Additional Controller Advertising Information (ACAD) field of the AUX_ADV_IND PDU. See Table 3.

Field	Length (octets)	Description
RspAA	4	The access address to be used by the Observer when it transmits a response packet back to the Broadcaster.
numSubevents	1	The number of subevents per PAwR event.
subeventInterval	1	The time from the start of one subevent to the beginning of the next subevent. Expressed as a multiple of 1.25 ms and supporting a range of 7.5 ms to 318.75 ms.
responseSlotDelay	1	Time from the start of a subevent to the first response slot. Expressed as a multiple of 1.25 ms and supporting a range of 1.25 ms to 317.5 ms.
responseSlotSpacing	1	Time from the start of one response slot to the beginning of the next response slot. Expressed as a multiple of 0.125 ms and supporting a range of 0.25 ms to 31.875 ms.

Table 3 - PAwR data in Periodic Advertising Response Timing Information AD type in AUX_ADV_IND PDUs

1.2.3.3.3 Periodic Advertising Sync Transfer (PAST)

When using the PAST procedure, sometimes the device passing the synchronization parameters over the connection will first acquire it by scanning on behalf of the other device. In the case of PAwR, however, support for PAST is mandatory and so the PAwR Broadcaster can pass the required synchronization data over an LE ACL connection to the Observer. If this approach is taken, no scanning for AUX_ADV_IND PDUs is necessary by either device.

The same data items presented in Table 3 are passed over the LE ACL connection in a new PDU type called LL_PERIODIC_SYNC_WR_IND.

1.2.3.3.4 Subevent Synchronization and Response Slot Allocation

Subevent synchronization is concerned with indicating to an Observer device the subevent it should perform scanning for. One or more Observer devices may be synchronized to the same subevent. An individual Observer may be synchronized to receive during one or more subevents.

In addition, for an Observer to be able to send a response PDU, it must have some basis for determining which subevent response slot to use.

Both of these concerns are the responsibility of the application layer. Section 1.2.4 Electronic Shelf Labels and PAwR explains how the Electronic Shelf Label profile deals with these concerns by example.

1.2.3.4 Host Controller Interface

Several changes have been made to the Host Controller Interface (HCI) to support the use of the PAwR logical transport.

1.2.3.4.1 Periodic Advertising Configuration

Version 2 of the **LE Set Periodic Advertising Parameters command** adds parameters Num_Subevents, Subevent_Interval, Response_Slot_Delay, Response_Slot_Spacing, and Num_Response_Slots. These parameters are required when configuring the Controller for periodic advertising with responses.

1.2.3.4.2 Setting Application Data

Two new commands and one event, each concerned with application data transported using PAwR, have been added to the HCI.

The Broadcaster uses the **LE Periodic Advertising Subevent Data Request event**. The Controller sends the event to the Host whenever it needs to request data to be included in AUX_SYNC_SUBEVENT_IND PDUs to be transmitted in a number of consecutive subevents in the future. It is recommended that the Controller requests data for as many future subevents as it has memory to accommodate. This strategy minimizes the number of HCI events required.

The Controller drives the process of acquiring application data for transmission rather than the Host because the Host and Controller are typically independent components, linked by a physical transport such as UART or USB. The Host knows nothing about Controller scheduling or memory availability in the Controller for storing this data. Application data to be transmitted in PAwR subevents will typically be quite dynamic, and the Host needs to keep the Controller supplied with the correct data for each upcoming subevent. But this can only be done when the Controller is ready for that data, which only it can determine.

The **LE Set Periodic Advertising Subevent Data command** allows the Broadcaster's Host to send the Controller data to be used in a series of one or more future subevent transmissions.

The Observer's Host uses the **LE Set Periodic Advertising Response Data command** to supply the Controller with application data to include in an AUX_SYNC_SUBEVENT_RSP PDU sent in a specific subevent and response slot.

1.2.3.4.3 Receiving Application Data

The **LE Periodic Advertising Response Report event** is a new event that the Broadcaster's Controller uses to pass PAwR response data that it has received to the Host.

The Observer's Controller uses the **LE Periodic Advertising Report event** to pass received advertising packet details to the Host. Version 2 of this HCI event adds a subevent field that indicates the PAwR subevent number of the received packet.

1.2.3.4.4 Synchronization

The **LE Set Periodic Sync Subevent command** is a new command that the Observer uses to instruct its Controller to synchronize its scanning schedule with one or more PAwR subevents belonging to a particular PAwR advertising train.

The **LE Periodic Advertising Sync Established event** and **LE Periodic Advertising Sync Transfer Received event** have each been updated to version 2 to add Num_Subevents, Subevent_Interval, Response_Slot_Delay, and Response_Slot_Spacing parameters.

1.2.3.5 Comparison with Other Logical Transports

The following table uses *Table 1 - Fundamental properties of a communication system* to compare PAwR with the other Bluetooth LE logical transports.

	Logical Transports						
Property	PAwR	ADVB legacy	ADVB extended	PADVB	LE ACL	LE BIS	LE CIS
Topologies	1:m	1:1[1], 1:m	1:1[1], 1:m	1:1, 1:m	1:1, 1:m[2]	1:m	1:1, 1:m[2]
Direction - Packets	bidirectional	bidirectional[3]	bidirectional[3]	unidirectional	bidirectional	unidirectional	bidirectional
Direction - Data	bidirectional	unidirectional	unidirectional	unidirectional	bidirectional	unidirectional	unidirectional or bidirectional
Communication Method	connectionless	connectionless	connectionless	connectionless	connection-oriented	connectionless	connection-oriented[6]
Data / Time Relationship	asynchronous	asynchronous	asynchronous	asynchronous	asynchronous	isochronous	isochronous
Receiver Concurrency	single/subset/all	all	all/subset[4]	all/subset[4]	single	all/subset[5]	single
RF Channels	primary during synchronization, secondary during synchronized advertising.	primary	primary secondary	primary during synchronization, secondary during synchronized advertising.	general-purpose	primary during synchronization, secondary during synchronized broadcast isochronous operation.	general-purpose
Scalability	m in 1:m topology is very large[7].	m in 1:m topology is very large[7].	m in 1:m topology is very large[7].	m in 1:m topology is very large[7].	limited by implementation issues but generally small[8]	m in 1:m topology is very large[7].	limited by implementation issues but generally small[8]

	Logical Transports						
Choice of PHY	Synchronized - LE 1M, LE 2M LE Coded Synchronizing - LE 1M or LE Coded	LE 1M only	LE 1M, LE 2M LE Coded except for the ADV_EXT_IND PDU (LE 1M or LE Coded)	Synchronized - LE 1M, LE 2M LE Coded Synchronizing - LE 1M or LE Coded	LE 1M, LE 2M LE Coded	Synchronized - LE 1M, LE 2M LE Coded Synchronizing - LE 1M or LE Coded	LE 1M, LE 2M LE Coded
Notes	1- *directed advertising* can be used to transfer data to a single target device 2 - A topology similar to 1:m can be created using multiple LE ACL connections or LE CIS streams from a single Central device to multiple Peripheral devices but it should be noted that technically, these are multiple, separate 1:1 relationships. 3 - packet flow is bidirectional when performing scannable advertising and with receivers performing active scanning 4 - advertising sets can be used to partition receivers into subsets 5 - A device can listen to one BIS stream from a choice of many streams within a broadcast isochronous group 6 - LE CIS has an associated LE ACL connection 7 - e.g., thousands of devices 8 - e.g., eight or less						

Table 4 - Comparing PAwR with other LE logical transports

As Table 4 shows, PAwR's key strengths include the fact that application data communication is bidirectional, flexibility is provided in the choices of topology and receiver concurrency available, and the number of devices that can be communicated with from one Broadcaster is very large (i.e., thousands of devices).

1.2.3.6 Battery Life

PAwR allows Observer devices to synchronize on a single subevent. This can result in devices being able to run off a coin cell battery for several years. The power requirements of products will vary and depend not only on how Bluetooth communication is performed. Table 5 presents an example energy consumption calculation for a device requiring a battery life of five years.

- Using LE 1M with PAwR involves a transmission rate of 1,000,000 bits per second.
- Assuming an advertising interval of 1.6 seconds.
- Assuming the Observer device has synchronized to a single subevent within that interval.
- A typical packet size of around 300 bits requires the receiver to be active for 300µs every 1600000µs.
- 157,680,000 (seconds in five years) / 1.6 seconds * 0.000300 seconds = 29,565 seconds of receiver activity over five years.
- This equates to just over 8 hours of energy consumption over five years due to receiver activity.
- A typical measure of current consumption due to Bluetooth receiver activity is 10 mAh. Therefore in this scenario, about 80 mA will be consumed.
- A typical coin cell battery has a capacity of 150 mAh.

This leaves sufficient energy for other aspects of the device, such as standby mode and running a display.

Note that using LE 2M would reduce energy consumption further still but at the expense of reduced range.

Table 5 - An illustration of energy consumption resulting from PAwR use

1.2.4 Electronic Shelf Labels and PAwR

1.2.4.1 Overview of the Electronic Shelf Label Profile

The Electronic Shelf Label (ESL) profile defines the standardized use of Bluetooth LE in the control of and communication with *electronic shelf labels*.

An electronic shelf label, as the name suggests, is an electronic device designed to be attached to shelves in settings like large stores. It has a display, which provides information such as the name and price of the items on that shelf. This usually takes the form of an image. Typically, an electronic shelf label can store multiple images, with one at a time selected for active display. Devices often have colored LEDs and sensors for collecting data like the local ambient temperature or the current battery level.

Figure 7 - An example electronic shelf label

Amongst the use cases which feature electronic shelf labels are the following examples:

1. Switching on the LED of a specific label and setting its color to red so that a shop worker can quickly locate a product for a customer.
2. Reducing the prices of all freshly baked bread an hour before closing.
3. Collecting the current temperature of the inside of a refrigerator.

The ESL profile uses both PAwR and connection-oriented interactions to meet its complete set of requirements. Images, for example, are written to devices over LE ACL connections. But most commands and responses involve ESL messages transported using PAwR PDUs in subevents.

ESL uses a device addressing scheme which consists of an 8-bit ESL ID and a 7-bit Group ID. The ESL ID is unique within the group of devices identified by a Group ID. Therefore a network of ESL devices can include up to 128 groups, each with a maximum of 255 unique ESL devices that are members of that group. In total, there may be 32,640 ESL devices in a network.

The ESL profile deals with subevent synchronization and response slot allocation as follows:

- The PAwR Broadcaster, known as an Access Point (AP) in the ESL profile specification, configures electronic shelf label devices by writing to various GATT characteristics over an LE ACL connection. The data written includes the assignment of an ESL Address consisting of an ESL ID and a Group ID. *Group* is an ESL profile concept, but its value is also used to indicate the number of the subevent during which the ESL device should scan.
- Response slot allocation happens dynamically. ESL devices receive an array of one or more commands from the AP in PAwR AUX_SYNC_SUBEVENT_IND PDUs. All commands in a request packet are addressed to the same ESL Group_ID. But each is addressed to a specific ESL in the group using its ESL_ID[5]. The index of the command in the array, counting from 1 for the first command, determines the response slot to be used. So, for example, having executed the 3rd command in the request PDU's array, response slot #3 will be used.

1.2.4.2 ESL and PAwR Illustration

1.2.4.2.1 ESL and 1:1 Device Communication

Figure 8 shows the transmission of PDUs that occur when the AP issues a command addressed to a single electronic shelf label. The diagram illustrates how PAwR acts as a transport for ESL commands and responses, as defined by the profile.

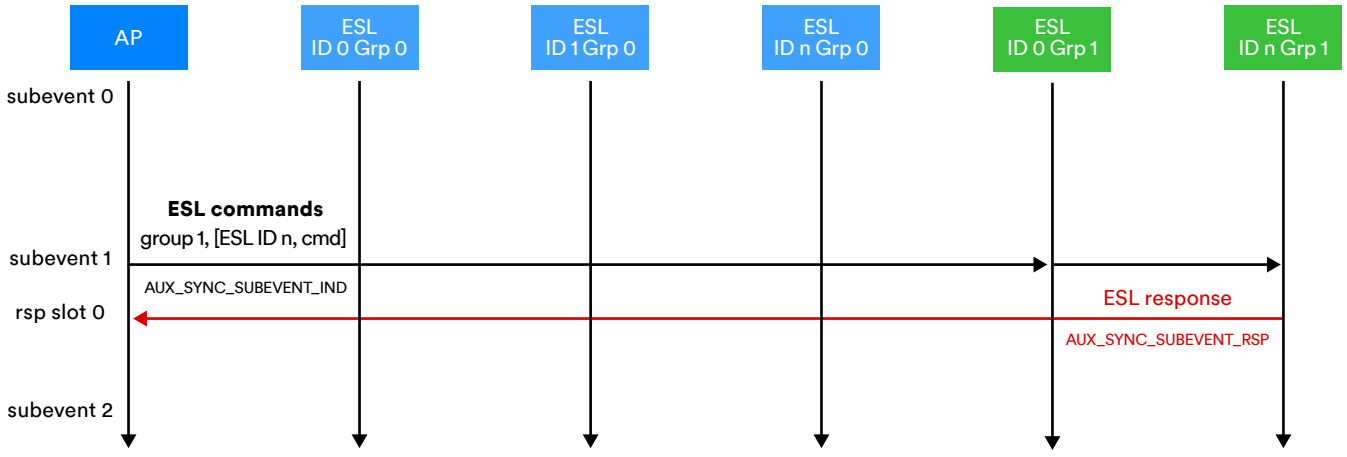

Figure 8 - An ESL command sent to an individual device

5 *ESL also supports the concept of broadcast messages but these do not elicit a response and so are not relevant here*

The shelf label to which the ESL command is addressed is a member of ESL group 1. This means that it is synchronized to PAwR subevent #1. The AP, therefore, formulates the ESL Payload, which can include an array of one or more commands, each addressed to a specific ESL ID within that same group, and transmits it as the payload of a PAwR AUX_SYNC_SUBEVENT_IND PDU during PAwR subevent #1.

The transmitted packet is received simultaneously by all shelf labels that are members of group 1 since they have all synchronized on and are listening during subevent #1. The single command in this PDU is addressed to ESL ID #n, so all shelf labels that receive the message discard it except for the device with the address of ESL ID #n and Group ID #1. This device processes the command per the ESL profile specification and then formulates and transmits a response in an AUX_SYNC_SUBEVENT_RSP PDU during response slot #0. Response slot #0 was used because the command being responded to was the command array's first (and only) member in the request.

Note that the ESL Profile specification numbers response slots starting at 1 whereas the Bluetooth Core Specification uses a numbering scheme that starts at 0. Figures 8 and 9 use this scheme.

1.2.4.2.2 ESL and 1:m Device Communication

Figure 9 shows the transmission of PDUs that occur when the AP issues a command addressed to several shelf labels, each of which is a member of ESL group #1. This is followed by transmitting a single command addressed to a single device that belongs to ESL group #2.

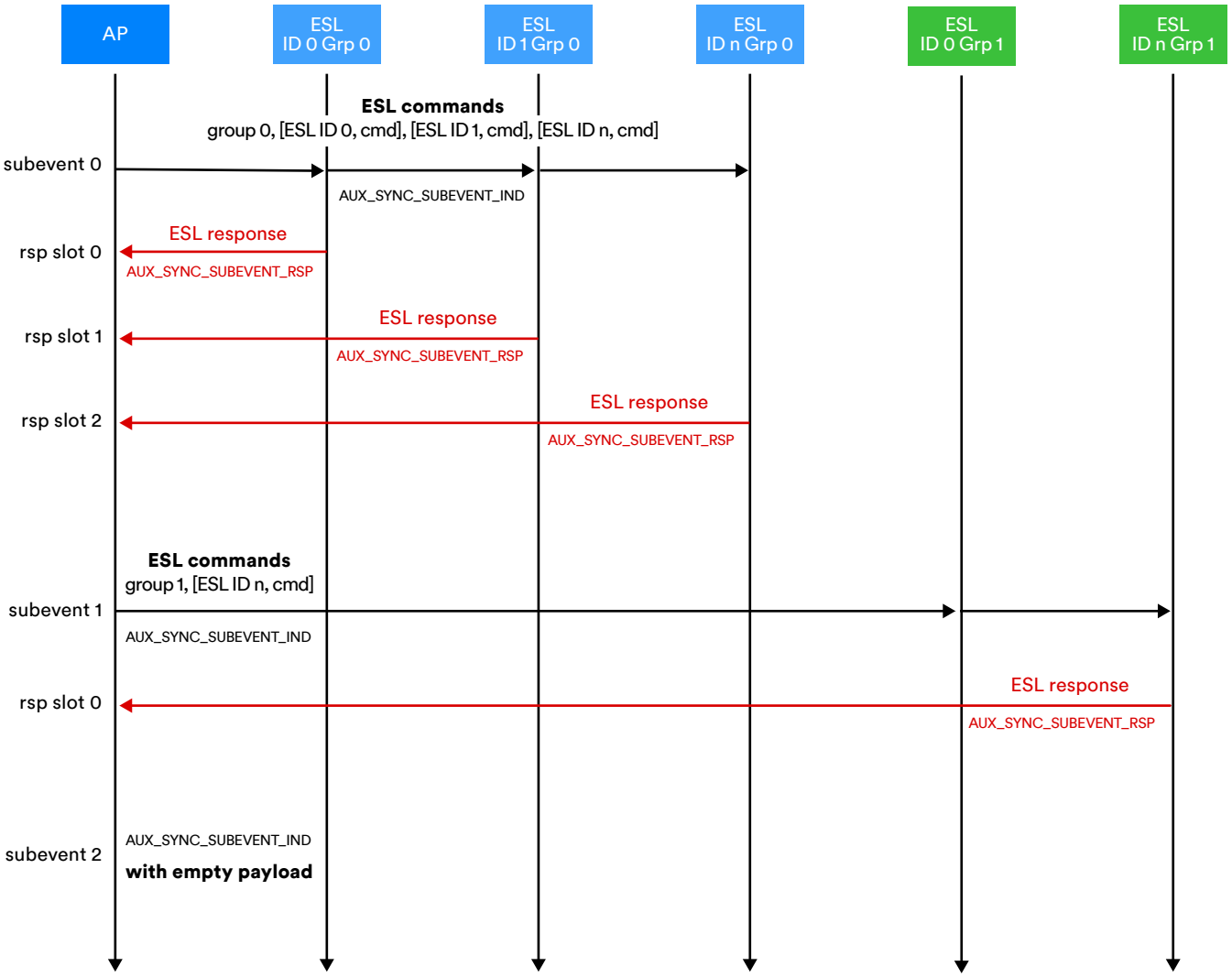

Figure 9 - An ESL request containing commands addressed to multiple devices in the group

The first ESL request contains three commands. The request targets three shelf labels belonging to ESL group #0, so it is formatted and set as the payload of an AUX_SYNC_SUBEVENT_IND PDU and transmitted in PAwR subevent #0.

All ESL shelf labels that are members of group #0 receive the PDU simultaneously since they are all synchronized on PAwR subevent #0. The ESL command array contains commands addressed to those shelf labels in the group that have ID #0, #1, and #n. These three devices process their respective commands. The device with ID #0 responds with an AUX_SYNC_SUBEVENT_RSP PDU in response slot 0. The device with ID #1 responds with an AUX_SYNC_SUBEVENT_RSP PDU in response slot 1. Finally, the device with ID #n responds with an AUX_SYNC_SUBEVENT_RSP PDU in response slot #2 since the command responded to was the third in the ESL command array. Other devices with different IDs ignore the request.

In PAwR subevent #1, the transmitted AUX_SYNC_SUBEVENT_IND PDU contains a command addressed to a single ESL which has ESL_ID=n. All ESLs that are members of ESL Group #1 are synchronized on PAwR subevent #1 and so receive this PDU. The ESL with ESL_ID=n processes the sole command in the payload and responds in PAwR response slot #0.

The AP has no commands to send to other ESLs in other groups and so in the remaining subevents, AUX_SYNC_SUBEVENT_IND PDUs with an empty payload are transmitted.

2. Encrypted Advertising Data

2.1 Background

2.1.1 Advertising

Background relating to Bluetooth® *advertising* was presented in section *1. Periodic Advertising with Responses*.

2.1.2 Structures and Types

The Bluetooth Core Specification defines the Advertising Data (AD) structure. It is a general container for application data to be included in advertising and scan response packets by Bluetooth LE and in Extended Inquiry Response (EIR) packets by Bluetooth BR/EDR. Data packaged within an AD structure may also appear in the Additional Controller Advertising Data (ACAD) field and in Out Of Band (OOB) exchanges.
The AD structure is divided into a length field, a type identifier, and the application data.
See Figure 10.

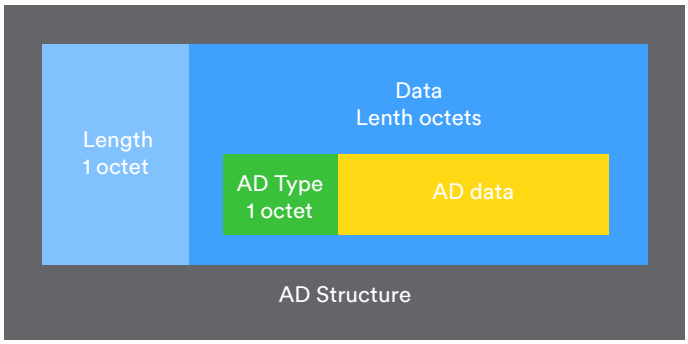

Figure 10 - AD Structure

The Bluetooth Core Specification Supplement (CSS) defines a series of AD *types*, examples of which include Flags, Complete Local Name, and Service Data.

2.1.3 Encryption

Sometimes the confidential nature of the data to be transmitted in advertising, scan response, or EIR packets may make it necessary for that data to be encrypted. Before Bluetooth® Core 5.4, there was no standardized way to meet this requirement. Encryption and authentication procedures were defined for connection-oriented communication but not for connectionless scenarios.

Encryption is a process that securely encodes information such that an unauthorized third party coming into possession of the encoded data cannot access the original *plain text* information. Encryption addresses the need for *confidentiality* in the presence of eavesdroppers.

Encryption algorithms use one or more keys in the encryption and decryption of data. Some algorithms also use an initialization vector (IV) as input. Collectively this data is known as *key material*. The intended recipient(s) of encrypted data must somehow be provided with the associated key material so that received data can be decrypted. The sharing of key material needs to be accomplished securely so that unauthorized parties cannot come into possession of it.

CCM is the *Counter with CBC-MAC* block cipher mode. It is used with a 128-bit block cipher such as AES to encrypt and authenticate messages. Authentication is achieved by including a calculated message authentication code (MAC) which the Bluetooth Core Specification calls a MIC (Message Integrity Check).

2.2 About Encrypted Advertising Data

2.2.1 Capabilities and Benefits

The new *Encrypted Advertising Data* feature provides a standardized, generally applicable mechanism for communicating encrypted data in advertising, scan response and EIR packets, and securely sharing associated encryption key material. This means that connectionless communication can now be used as a secure application data delivery mechanism in one-to-many or one-to-one topologies.

2.2.2 Technical Highlights

This section describes the main technical aspects of the Encrypted Advertising Data feature when used with the Bluetooth LE transport. The general approach when using Bluetooth BR/EDR is similar or identical. The Bluetooth Core Specification should be checked for details.

2.2.2.1 Sharing key material

The standard procedure for sharing key material for use with the Encrypted Advertising Data feature requires the device which is to transmit that data to adopt the GAP Peripheral role. This means that the device can advertise and accept a connection request from another device acting in the GAP Central role.

The GAP Peripheral must act as a GATT server and implement certain mandatory GATT services, including the GAP (generic access profile) service.

A new characteristic called ***Encrypted Data Key Material*** has been defined, and a GAP Peripheral may include it in the GAP service. The new characteristic provides the basis by which key material can be shared with devices that are the intended recipients of encrypted advertising data.

The Encrypted Data Key Material characteristic contains a 24-octet value which is made up of a 16-octet session key and an 8-octet IV value. A GATT client can read this value over an encrypted and authenticated ACL connection only; therefore, the advertising device and all devices intended to be recipients of encrypted advertising data must have been paired. Writing to the characteristic is not permitted. The characteristic may also support GATT indications but only over a secure link.

When supported, the GAP Peripheral/GATT server can use indications to inform a connected GAP Central device whenever the key material value changes.

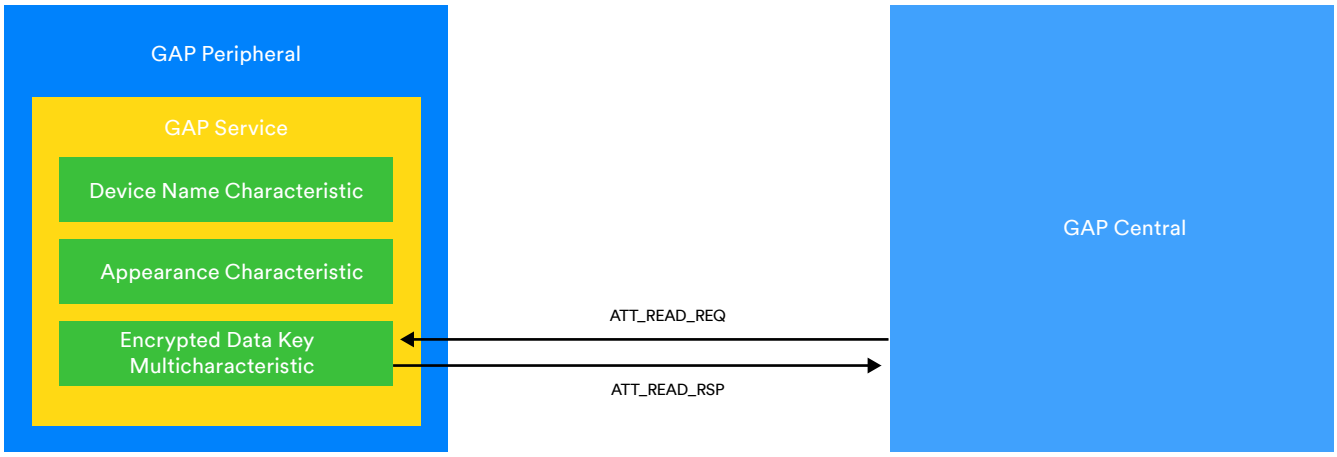

Figure 11 - Client initiated read of Encrypted Data Key Material

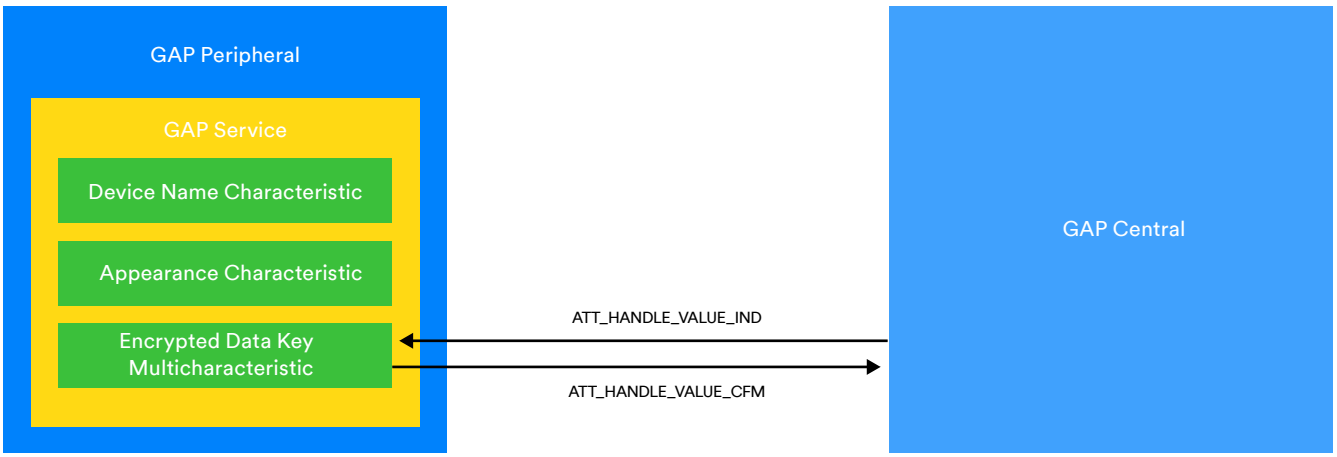

Figure 12 - Server initiated indication of Encrypted Data Key Material

Should there be a requirement for a device to accommodate multiple encryption key material values, the Encrypted Data Key Material characteristic can be included in services other than the GAP service.

2.2.2.2 Encryption of Data

Any data to be securely transmitted must first be encapsulated within an appropriate AD structure.

More than one AD structure can be encrypted. This is accomplished by first concatenating the collection of AD structures requiring encryption into a sequence of AD structures. It is this sequence of one or more AD structures that is encrypted.

The CCM algorithm is used to encrypt and authenticate the data. Consult the Bluetooth Core Specification Volume 6, Part E, section 2 for details of the use of CCM.

2.2.2.3 Transmission of encrypted data

A new AD type called *Encrypted Data* has been defined for use as the container for the ciphertext produced by encrypting the sequence of one or more AD types that need to be secured. The Encrypted Data AD type is then included within appropriate packets. Those AD types that have been encrypted are not included in these packets in their original plain-text form. Other, unencrypted AD types may be included in packets which contain the Encrypted Data AD type.

In addition to the ciphertext payload, the Encrypted Data AD structure's data field contains a 40-bit *Randomizer* field and a 32-bit *Message Integrity Check* (MIC). Figure 13 shows an example advertising payload which contains two AD types (ESL and Local Name) that have been encrypted and encapsulated within the Encrypted Data AD type and one AD type (Flags) which is included unencrypted.

The Randomizer field contains a 5-octet random number generated per the requirements for random numbers stated in the Bluetooth Core Specification. A new Randomizer value must be generated every time the payload value changes. The Randomizer value is used in formulating a *nonce* which the CCM algorithm requires.

The Randomizer field must also change whenever the device changes its address, assuming a random device address is in use. This causes the contents of an advertising packet to change when the device address changes, reducing the ability for somebody to track the device.

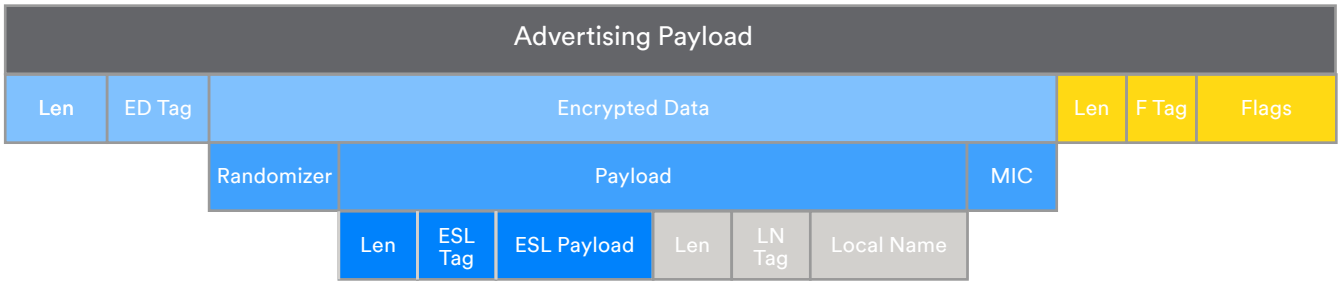

Figure 13 - The Encrypted Data AD Type

2.2.3 Profiles using Encrypted Advertising Data

Profiles that use the Encrypted Advertising Data feature are responsible for defining how encryption key material consisting of a session key and IV is to be pre-shared. As described, the Encrypted Data Key Material characteristic is provided for this purpose, and it may either be included within the generic access profile GATT service or in some other service depending on other profile-related considerations, such as device roles.

3. The LE GATT Security Levels Characteristic

3.1 Background

3.1.1 The Generic Attribute Profile (GATT)

The Generic Attribute Profile (GATT) provides a means by which device data and capabilities may be represented in a hierarchical structure consisting of GATT services, characteristics, and descriptors. Each of these three types of GATT construct is an instance of something more general, called an *attribute*. Attributes are declared and defined in an *attribute table,* which flattens the hierarchical structure and assigns each entry in the table a unique identifier known as a *handle*.

Devices access the attributes in the attribute table of a connected, remote device using a protocol called the Attribute Protocol (ATT), following rules defined by various GATT procedures, such as *characteristic value read* and *characteristic value write*.

GATT defines two roles. The GATT client sends ATT commands (which do not need to be responded to) and requests (for which a response is required) to the GATT server. The GATT server accepts and processes commands and requests that are received from a GATT client. The GATT server may also send ATT PDUs of various types to the GATT client, informally known as *notifications*, *indications,* and *responses*.

Each attribute in the *attribute table* includes a set of *attribute permissions*. Attribute permissions define rules regarding the kind of access a connected client may or may not have to that attribute (e.g., the ability to read its value) and any conditions that might apply before that access is granted[6]. For example, an attribute's permissions might indicate that clients can read its value but only over an authenticated and encrypted link. Attribute permissions also apply to ATT servers and their communication with clients using notifications and indications.

ATT PDUs are transported over an LE-ACL[7] connection, so before any GATT procedures may be executed and ATT PDUs exchanged, devices must first connect. Generally, after establishing a connection, a GATT client will proceed by performing a series of procedures known as the *discovery procedures*. Discovery is concerned with determining the content of the remote device's attribute table in terms of services, characteristics, and descriptors and their associated properties, such as handle values, types, and permissions. Note that attribute permissions do not restrict the ability of a client to perform the discovery procedures.

The hierarchical structure of service, characteristics, and descriptors is shown in Figure 14.

---

6 See the Bluetooth LE Security Study Guide for more information about Bluetooth LE security including attribute permissions
7 ATT can be used with Bluetooth BR/EDR as well as Bluetooth LE but we're only concerned with LE here

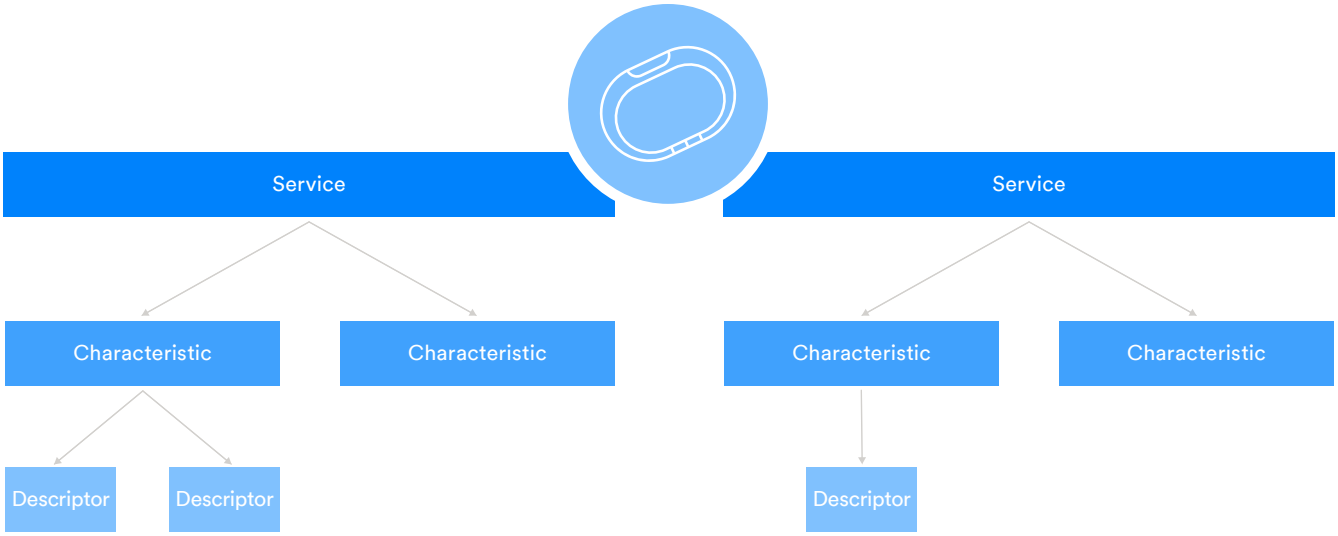

Figure 14 - Services, Characteristics, and Descriptors

If an attempt to access an attribute is made, and the conditions of the associated attribute permissions are not met, the attribute protocol defines several error codes to be returned to indicate to the client device that the access request was denied and for what reason. Examples include *insufficient encryption*, *insufficient authentication,* and *insufficient encryption key size*.

Two special services are mandatory in all GATT servers. These are the *generic access service* and the *generic attribute service*.

3.1.3 GATT Security and User Experience

An attribute's permissions are checked whenever an attempt to access that attribute is made. If the security conditions mandated by an attribute's permissions are not satisfied, access will be denied, and a response containing an error code will be returned in an ATT_ERROR_RSP PDU.

If the server denies access to an attribute, the connection will not usually be closed. This allows the client to deal with the error by upgrading security so that subsequent access attempts will succeed. For example, if an attempt to read a characteristic value results in the *insufficient encryption* error being returned, the client might handle this by initiating the pairing procedure and, when complete, upgrading the connection to use encryption. The user might then need to interact with the device again to retry the failed operation.

Handling errors due to unsatisfied security permissions at the time that they occur has the disadvantage that an application's normal flow is interrupted, and the user experience is, therefore, sub-optimal. However, Bluetooth® Core Specification v5.3 (Bluetooth® Core 5.3) offers no alternative to this security error-handling strategy.

3.2 About the LE Gatt Security Levels Characteristic

3.2.1 Overview

Bluetooth® Core 5.4 defines a new characteristic called the *LE Gatt Security Levels* characteristic (SLC)[8]. The SLC characteristic allows clients to determine the GATT server security conditions, which must be satisfied if access to all GATT functionality is to be granted. Importantly, it allows this to be determined before accessing attributes that the application uses. Checking access requirements in advance allows a better user experience to be created without ad hoc interruptions to application flow caused by security-level problems.

3.2.2 Technical Highlights

Devices *may* include the SLC characteristic in the mandatory generic access profile service. Its inclusion is therefore optional but recommended, given the improved user experience its use can make possible.

The SLC characteristic allows read access (only) to its value without further security restrictions, such as the need for the link to be encrypted.

Bluetooth LE security levels are expressed in terms of a *mode* and a *level*.

LE security mode 1 has the following security levels:

1. No security (No authentication and no encryption)
2. Unauthenticated pairing with encryption
3. Authenticated pairing with encryption
4. Authenticated LE Secure Connections pairing with encryption using a 128-bit strength encryption key

LE security mode 2 has two security levels:

1. Unauthenticated pairing with data signing
2. Authenticated pairing with data signing

LE security mode 3 has three security levels:

1. No security (no authentication and no encryption)
2. Use of unauthenticated Broadcast_Code
3. Use of authenticated Broadcast_Code

LE Secure Connections Only mode (LE security mode 1 level 4)

Table 6 - Bluetooth LE security modes and levels

8 *See Bluetooth Core Specification Volume 3 Part C Section 12.7.*

There may be more than one security mode and level combination that will satisfy the security requirements of all attributes of the server. Consequently, the SLC characteristic's attribute value consists of an array of one or more Security Level Requirements fields. The Security Level Requirements field is of type uint8[2], with the first uint8 value containing a direct representation of a security mode (e.g., 0×01 for security mode 1) and the second, a representation of a security level (e.g., 0×04 for security level 4).

Clients use the SLC characteristic by reading its value and evaluating the current security mode and level against the values indicated by the Security Level Requirement field(s). If it is found to be the case that the current security mode and level are insufficient to allow all GATT functionality supported by the server, the client application, at this point, takes steps to remedy this, typically by invoking procedures to upgrade the link security.

4. Advertising Coding Selection

4.1 Background

4.1.1 Bluetooth® LE and PHYs

The Bluetooth LE physical layer defines three variants, referred to collectively as *PHYs*. The three PHYs are called LE 1M, LE 2M, and LE Coded.

When LE Coded is used, a Forward Error Correction (FEC) algorithm and, depending on configuration, a pattern mapper is applied to payloads before transmission. This results in additional error correction data being included in the transmitted packet. Using FEC allows data to be received correctly at significantly longer distances from the transmitter (i.e., with lower signal-to-noise ratios). An FEC parameter called *S* takes one of two values, S=2 or S=8, and controls how much error correction data is generated and to what extent the communication range might be increased.

The three PHYs are presented and compared in Table 7.

	LE 1M	LE Coded S=2	LE Coded S=8	LE 2M
Symbol Rate	1 Msym/s	1 Msym/s	1 Msym/s	2 Msym/s
Protocol Data Rate	1 Mb/s	500 kb/s	125 kb/s	2 Mb/s
Symbols per Bit	1	2	8	1
Bits per symbol	1	0.5	0.125	1
Error Detection	CRC	CRC	CRC	CRC
Error Correction	NONE	FEC	FEC	NONE
Range Multiplier (approx.)	1	2	4	0.8
Requirement	Mandatory	Optional	Optional	Optional
Comment	The default PHY, which all Bluetooth LE devices must support.	Increases range by a factor of around 2 compared to the range achieved with LE 1M but reduces protocol data rate to 500 kbps. If LE Coded with S=2 is supported then LE Coded with S=8 must also be supported.	Increases range by a factor of around 4 compared to the range achieved with LE 1M but reduces protocol data rate to 125 kbps. If LE Coded with S=8 is supported then LE Coded with S=2 must also be supported.	Offers superior spectral efficiency compared to LE 1M. May improve application data rates if used appropriately at the link layer.

Table 7 - The Bluetooth® LE PHYs

4.1.2 Host Controller Interface (HCI) and PHY Parameters

The Host may select the PHY to be used or express a preference using HCI commands and events. At Bluetooth® Core 5.3, however, where LE Coded has been selected for use with extended advertising[9], it is not possible to specify the value of the coding parameter *S* to use (2 or 8).

9 *Legacy advertising can only be used with LE 1M*

4.2 About the Coding Scheme Selection on Advertising (CSSA) Change

4.2.1 Overview

Bluetooth® Core 5.4 changes various HCI commands to allow the value for the FEC parameter S to be specified when using LE Coded.

4.2.3 Technical Highlights

Table 8 lists the HCI commands and events modified to accommodate the LE Coded S parameter value.

HCI command or event	Change
LE Set Extended Advertising Parameters command	Version 2 of this command adds parameters *Primary_Advertising_PHY_Options* and *Secondary_Advertising_PHY_Options*. In each case, the following values are defined: 0×00 The Host has no preferred or required coding when transmitting on LE Coded 0×01 The Host prefers that S=2 coding be used when transmitting on LE Coded 0×02 The Host prefers that S=8 coding be used when transmitting on LE Coded 0×03 The Host requires that S=2 coding be used when transmitting on LE Coded 0×04 The Host requires that S=8 coding be used when transmitting on LE Coded
LE Extended Advertising Report event	The Primary_PHY[i] and Secondary_PHY[i] parameters may each use a value of 0×03 to indicate S=8 or 0×04 to indicate S=2.

Table 8 - CSSA HCI changes

New link layer feature support bits have been allocated to indicate support (or otherwise) for CSSA by the Host and the Controller.

5. Conclusion

Bluetooth® Core 5.4 adds a significant new bidirectional connectionless capability in PAwR and makes it possible to broadcast confidential data in advertising packets securely. In addition to these considerable enhancements, applications that use GATT can now offer a better user experience when dealing with attribute security requirements than before, and devices can exercise control over an important parameter (S) when using LE Coded for extended advertising.

References

Item	Location
Bluetooth® Core Specification v5.4	https://www.bluetooth.com/specifications/specs/
The Bluetooth® LE Security Study Guide	https://www.bluetooth.com/bluetooth-resources/le-security-study-guide/
The Bluetooth® Security and Privacy Best Practices Guide (SIG members only)	https://www.bluetooth.com/bluetooth-resources/bluetooth-security-and-privacy-best-practices-guide/
Bluetooth® Core 5.0 Feature Overview	https://www.bluetooth.com/bluetooth-resources/bluetooth-5-go-faster-go-further/