Bluetooth®

Bluetooth Core Specification v5.1

Feature Overview

Bluetooth Core Specification v5.1 contains a series of updates to the *Bluetooth*® core specification. This document summarizes and explains each change.

Bluetooth Core Specification v5.1 should be consulted for full details.

Author: Martin Woolley

Version: 1.0

Revision Date: 28 January 2019

January 2019

Revision History

Version	Date	Author	Changes
1.0	28 January 2019	Martin Woolley	Initial Version

Bluetooth®

table of contents

back to contents

1.0 Direction Finding

Overview

蓝牙接近解决方案和定位系统目前使用信号强度来估计距离。 蓝牙核心规范v5.1中的新方向发现功能使蓝牙设备可以确定蓝牙信号传输的方向。

Bluetooth proximity solutions and positioning systems currently use signal strength to estimate distance. A new direction finding feature in Bluetooth Core Specification v5.1 makes it possible for Bluetooth devices to determine the direction of a Bluetooth signal transmission.

这一新功能提供了两种不同的方法，用于确定蓝牙信号从高精度传输的角度。 这两种方法称为到达角（aoa）和出发角（aod）。

This new feature offers two different methods for determining the angle that a Bluetooth signal is being transmitted from with a high degree of accuracy. The two methods are called Angle of Arrival (AoA) and Angle of Departure (AoD).

Each of the techniques requires one of the two communicating devices to have an array of multiple antennae, with the antenna array included in the receiving device when the AoA method is used and in the transmitting device when using AoD.

每种技术都要求两个通信设备中的一个具有多个天线的阵列，当使用aoa方法时天线阵列包括在接收设备中，而当使用aod时，天线阵列包括在发送设备中。

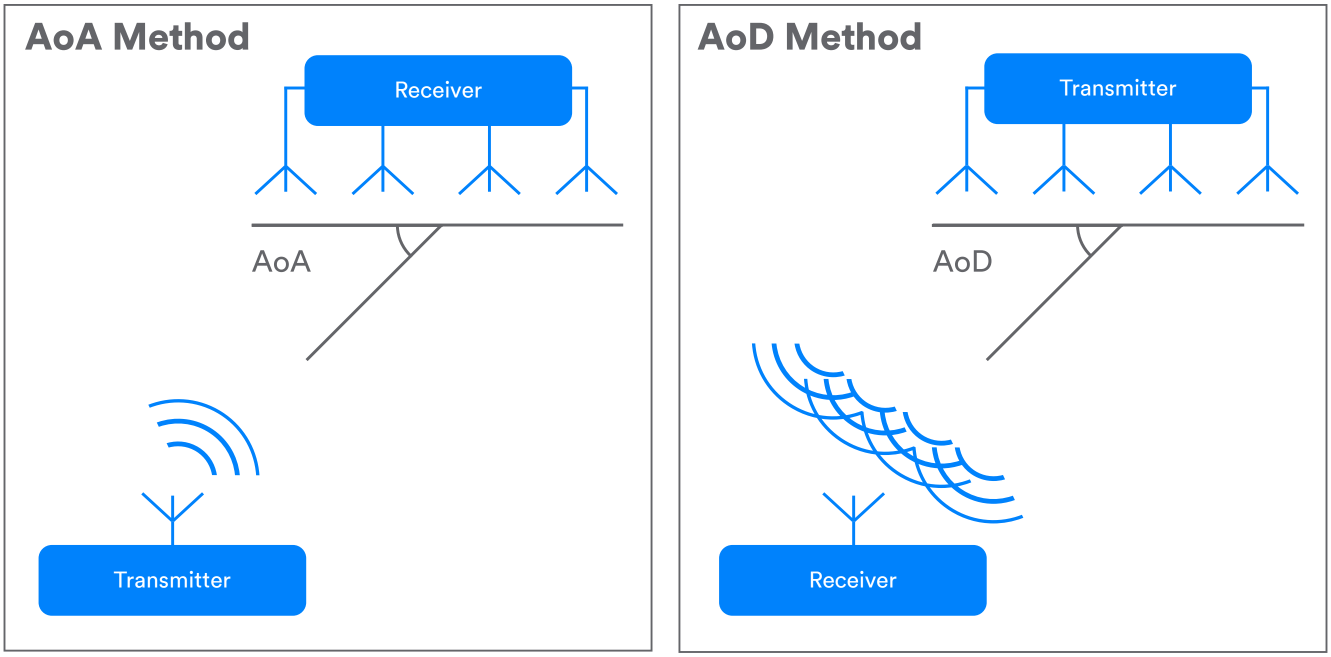

Figure 1 - Angle of Arrival (AoA) and Angle of Departure (AoD)

蓝牙核心规范v5.1为接收设备中的蓝牙低能量（le）控制器提供了生成数据的能力，然后可以使用该数据来计算发送设备的方向角。

Bluetooth Core Specification v5.1 gives the Bluetooth Low Energy (LE) controller in the receiving device the ability to generate data that can then be used to calculate the directional angle to the transmitting device.

The addition of direction finding in this release of the Bluetooth Core Specification is the first of several steps in the Bluetooth roadmap that will ultimately enable key enhancements to Bluetooth location services. When the associated profiles have been released, Bluetooth developers will be able to exploit the new direction finding controller capability to create high accuracy, interoperable positioning systems such as real-time locating systems (RTLS) and indoor positioning systems (IPS).

在这一版本的蓝牙核心规范中增加了寻向功能，这是蓝牙路线图中的几个步骤中的第一步，这些步骤将最终实现对蓝牙定位服务的关键增强。当相关的配置文件发布后，蓝牙开发者将能够利用新的寻向控制器功能来创建高精度、可互操作的定位系统，如实时定位系统（RTLS）和室内定位系统（IPS）。

back to contents

新的寻向功能还具有通过确定设备方向来增强蓝牙邻近解决方案的潜力，特别是在定向项目查找和兴趣点信息解决方案中。

The new direction finding feature also has the potential to enhance Bluetooth proximity solutions by determining device direction, particularly in directional item finding and point of interest information solutions.

Technical Details

蓝牙测向功能使用同相和正交（IQ）采样来测量特定时间入射到天线上的无线电波的相位。在AOA方法中，根据阵列的设计，将采样过程应用于阵列中的每个天线，一次一个，并以适当的顺序进行采样。

The Bluetooth direction finding feature uses In-Phase and Quadrature (IQ) sampling to measure the phase of radio waves incident upon an antenna at a specific time. In the AoA approach, the sampling process is applied to each antenna in the array, one at a time, and in some suitable sequence depending on the design of the array.

采样数据通过主机控制器接口（HCI）向上传递到堆栈，然后可以对采样数据应用适当的算法，以计算一个设备与另一个设备的方向。本核心规范版本中未定义从IQ样本计算角度的算法。一旦关联的概要文件可用，应用程序开发人员将有机会实现适合预期用例的算法。

Sampled data is passed up the stack via the Host Controller Interface (HCI) where it will then be possible to apply a suitable algorithm to the sampled data to calculate the direction of one device from the other. Algorithms for calculating angles from IQ samples are not defined in this core specification release. Once associated profiles are available, application developers will have the opportunity to implement algorithms suitable for the intended use case.

To support IQ sampling and the use of IQ samples by higher layers in the stack, the link layer (LL) and HCI have each changed.

为了支持IQ采样和堆栈中更高层使用IQ采样，链路层（ll）和HCI都发生了变化。

At the link layer, a new field called the Constant Tone Extension (CTE) has been defined (see Figure 2). The purpose of the CTE field is to provide constant frequency and wavelength signal material against which IQ sampling can be performed. This field contains a sequence of 1s, is not subject to the usual whitening process and is not included in the CRC calculation.

在链路层，定义了一个称为恒定音调扩展（cte）的新字段（见图2）。 cte字段的目的是提供恒定的频率和波长信号材料，可以对其进行iq采样。 此字段包含1的序列，不受通常的白化过程影响，并且不包含在crc计算中。

LSB MSB

Preamble (1 or 2 octets)	Access-Address (4 octets)	PDU (2-258 octets)	CRC (3 octets)	Constant Tone Extension (16 to 160 μs)

Figure 2 - Constant Tone Extension

CTE can be used in both connectionless and connection-oriented scenarios. For connectionless use, the periodic advertising feature is required (since deterministic timing in the sampling process is important) and CTE is appended to AUX_SYNC_IND PDUs. For connection-oriented use, new PDUs LL_CTE_REQ and LL_CTE_RSP have been defined. In either case, there are new HCI PDUs that allow the configuration of various aspects of CTE PDUs, such as the CTE length, length of the antenna switching pattern, and antenna IDs.

CTE可用于无连接和面向连接的场景。对于无连接使用，需要定期广告功能（因为采样过程中的确定时间很重要），并将CTE附加到辅助同步指示PDU。对于面向连接的使用，已经定义了新的PDU llu-cte-req和llu-cte-rsp。在这两种情况下，都有新的HCI PDU，允许配置CTE PDU的各个方面，例如CTE长度、天线切换模式的长度和天线ID。

back to contents

2.0 GATT Caching Enhancements

Background

所有蓝牙低能量连接设备都使用通用属性配置文件（gatt）。因此，GATT缓存的主题与广泛的设备类型相关。

All Bluetooth Low Energy connected devices use the Generic Attribute Profile (GATT). As such, the subject of GATT caching is of relevance to a wide range of device types.

GATT设备包含一个称为属性表的数据库。属性表包含GATT服务、特征和描述符结构细节和值，是基于GATT的蓝牙低能设备工作方式的核心。属性表中的条目由属性句柄标识。

GATT devices contain a database known as the *attribute table*. The attribute table contains GATT service, characteristic, and descriptor structural details and values, and is central to how GATT-based Bluetooth Low Energy devices work. Entries in the attribute table are identified by *attribute handles*.

GATT clients must perform a procedure known as *service discovery* to acquire details of the attribute table on the remote GATT server device that the client has connected to. The client can then use these details, including the identifying attribute handles in subsequent Attribute Protocol (ATT) interactions with the server.

GATT设备包含一个称为属性表的数据库。属性表包含GATT服务、特征和描述符结构细节和值，是基于GATT的蓝牙低能设备工作方式的核心。属性表中的条目由属性句柄标识。

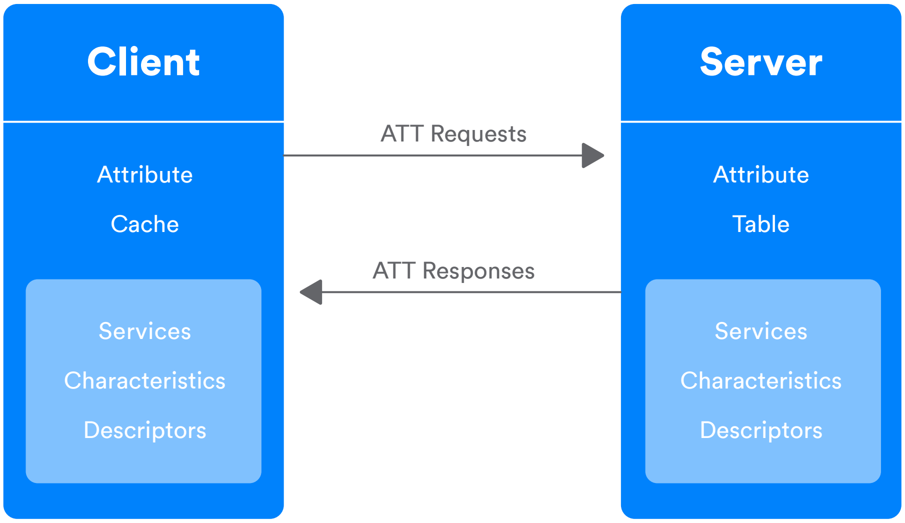

Figure 3 - Service Discovery and Attribute Caching

Some devices do not change their attribute table structure throughout their life. The GATT services, characteristics, and descriptors present in the table will always be the same, and only values of characteristics or descriptors will change. Other devices do change their attribute table from time to time.

某些设备在其整个生命周期中不会更改其属性表结构。 表中存在的gatt服务，特征和描述符将始终相同，并且只有特征或描述符的值才会发生变化。 其他设备会不时更改其属性表。

Service discovery takes time and consumes energy. Therefore, Bluetooth Core Specification v5.1 defines an *attribute caching* strategy aimed at allowing clients to skip service discovery when nothing has changed.服务发现需要时间并消耗能量。 因此，蓝牙核心规范v5.1定义了一种属性缓存策略，旨在允许客户端在没有任何变化时跳过服务发现。

Previously, caching and client/server attribute table synchronization was controlled solely using the *Service Changed* characteristic that might be present in the *Generic Attribute Service*. The GATT server could inform a connected client that its attribute table had changed by sending an ATT indication to the client. The client replied with an ATT confirmation and performed service discovery to synchronize its attribute cache with that of the server.

以前，仅使用可能存在于通用属性服务中的服务更改特征来控制高速缓存和客户端/服务器属性表同步。 gatt服务器可以通过向客户端发送att指示来通知连接的客户端其属性表已经改变。 客户端回复了att确认并执行了服务发现，以使其属性缓存与服务器的属性缓存同步。

To avoid the GATT server needing to keep track of every client that ever connected to it, and whether or not each client had been informed of the latest attribute table change, previously the core

为了避免gatt服务器需要跟踪每个连接到它的客户端，以及每个客户端是否都被告知最新的属性表更改，以前核心规范规定客户端和没有信任关系的服务器（即 每次连接时都需要扌行服务发现。 此规则可能会导致某些类型产品的能源效率和用户体验问题。

specification stipulated that clients and servers that have no trusted relationship (i.e. are not bonded) were required to perform service discovery every time they connect. This rule can cause energy efficiency and user experience issues for some types of products.

In addition, beyond making a single attempt to inform the client that the attribute table had changed using the ATT *Service Changed* indication, there was no further state management carried out with respect to the client's view of the attribute table vs the server's. The approach allowed a race condition in the communication between client and server, with respect to attribute table changes and general ATT interactions to exist, whereby it was possible for a client to time-out whilst waiting for a *Service Changed* indication after connecting to the server, proceed to send general ATT PDUs, and then receive a Service Changed indication.

Improved Caching Strategy

此版本更改了gatt客户端和服务器如何接近属性缓存和缓存同步。 它允许没有与服务器建立信任关系的客户端跨连接保留其属性缓存并解决上述竞争条件问题，从而提供显着的用户体验和能效提升。

This release makes changes to how attribute caching and cache synchronization is approached by GATT clients and servers. It offers significant user-experience and energy-efficiency improvements by allowing clients without a trusted relationship with a server to retain their attribute cache across connections and resolves the race condition issue described above.

Two new characteristics, each a member of the *Generic Attribute Service,* have been introduced: *Database Hash* and *Client Supported Features*. Clients which do not have a trusted relationship with the server may now cache the attribute table across connections if the client supports the new Database Hash characteristic, as indicated by the client updating a flag in the server's *Client Supported Features* characteristic.

引入了两个新特征，每个特征都是通用属性服务的成员：数据库哈希和客户端支持的功能。 如果客户端支持新的数据库哈希特征，那么与服务器没有可信关系的客户端现在可以跨连接缓存属性表，如客户端更新服务器的客户端支持的功能特征中的标志所指示的。

The *Database Hash* characteristic allows the client to ask the server if anything has changed, rather than relying on the server telling it using a Service Changed indication. The server is responsible for maintaining the value of the *Database Hash* characteristic, which is a hash value, calculated from pertinent aspects of the attribute table. The client reads its value immediately after establishing a connection. The client may cache the *Database Hash* value and subsequently use it to determine whether or not the remote attribute table has changed. If it has changed, the client performs service discovery again. If it has not, it does not need to. This offers a major user-experience and energy efficiency benefit to some device types.

数据库哈希特性允许客户端询问服务器是否有任何更改，而不是依赖于服务器使用服务更改指示告诉它。 服务器负责维护数据库哈希特征的值，该值是根据属性表的相关方面计算的哈希值。 客户端在建立连接后立即读取其值。 客户端可以缓存数据库哈希值，并随后使用它来确定远程属性表是否已更改。 如果已更改，则客户端再次执行服务发现。 如果没有，则不需要。 这为某些设备类型提供了重要的用户体验和能效优势。

Furthermore, a client may now deduce that a device it is connecting to is the same type of device as one previously connected to and whose attribute table has already been cached by the client. If the database hash from the connected device is the same as the one associated with the client's attribute cache, and other details such as the device manufacturer are the same, the client may conclude that there is no need to perform service discovery for the connected device since the attribute cache obtained from another device contains equivalent data already.

此外，客户端现在可以推断出已所连接的设备是与先前连接到的设备相同类型的设备，并且其属性表已经被客户端高速缓存。 如果来自连接设备的数据库哈希与与客户端属性缓存关联的数据库哈希相同，并且其他详细信息（例如设备制造商）相同，则客户端可以断定不需要为连接的设备执行服务发现 因为从另一个设备获得的属性缓存已经包含等效数据。

For some applications, this change has considerable value. For example, consider Bluetooth smart locks, where a smartphone or other client device interacts with doors in a building to authenticate and open the door for a user when they approach. Service discovery need only be performed the first time the user attempts to pass through a door with a smart lock. The user may perceive a delay

对于某些应用，这种变化具有相当大的价值。 例如，考虑蓝牙智能锁，其中智能手机或其他客户端设备与建筑物中的门进行交互以对用户进行身份验证并打开门。 服务发现只需在用户第一次尝试通过带智能锁的门时执行。 在第一次使用期间，用户可能感觉到门解锁的延迟，但是不需要用户在建筑物服务发现中接近任何门的所有后续时间，并且用户将经历来自智能锁的近乎瞬时的响应。

in the door unlocking during this first occasion, but all subsequent times the user approaches any of the doors in building service discovery will not be required, and the user will experience a near instantaneous response from the smart lock.

Better State Management

状态机定义属性表的客户端视图和其属性表的服务器视图是否同步，并且因此，客户端是否需要执行服务发现。 修订后的属性缓存规范引入了严格定义的强健缓存概念，该规范将此状态机形式化并引入了使用它的机制

A state machine defines whether or not the client view of the attribute table and the server view of its attribute table are in sync and, as such, whether or not the client needs to perform service discovery. The revised specification for attribute caching introduces the rigorously defined concept of *Robust Caching* that formalizes this state machine and introduces mechanisms for using it.

Clients are said to be in the *change-aware* state or are *change-unaware*. The specification lays out the precise rules for transitioning to the appropriate state and how to behave when in each of the two states.

状态机定义属性表的客户端视图和其属性表的服务器视图是否同步，并且因此，客户端是否需要执行服务发现。 修订后的属性缓存规范引入了严格定义的强健缓存概念，该规范将此状态机形式化并引入了使用它的机制

Of particular note is the new «Database Out Of Sync» ATT error response that the server may return if it believes the client attribute table cache is out of sync with the server's. The server will ignore all ATT commands received from the client while it is in the change-unaware state. A number of events can transition the client's state to *change-aware*, including the server receiving an ATT confirmation to a *Service Changed* indication it had previously sent or the server having notified the client using the <<Database Out Of Sync>> error and subsequently receiving some other ATT PDU from the client. From the client's point of view, if it moves to the change-unaware state it will not use its attribute cache, regarding it as invalid. It will continue to be treated as invalid until the client's attribute cache and the servers are in sync once again.

特别要注意的是，如果服务器认为客户端属性表缓存与服务器不同步，则可能返回新的«数据库不同步»ATT错误响应。服务器将忽略从客户端接收的所有ATT命令，而此时它处于“未意识到更改”状态。许多事件可以将客户机的状态转换为可识别更改，包括服务器接收到ATT确认，而服务器接收到之前发送的服务更改指示，或服务器使用<<database out of sync>误通知客户机，然后从客户机接收到一些其他ATT PDU。从客户机的角度来看，如果它移动到更改未意识到的状态，它将不会使用其属性缓存，将其视为无效。在客户端的属性缓存和服务器再次同步之前，它将继续被视为无效。

back to contents

3.0 Advertising Enhancement 1: Randomized Advertising Channel Indexing

Background

在蓝牙核心规范v5.0中，广告事件被定义为“在主要广告频道上发送的一个或多个广告pdus，从第一个使用的广告频道索引开始并以最后使用的广告频道索引结束”。

In Bluetooth Core Specification v5.0, advertising events are defined as "*one or more advertising PDUs sent on the primary advertising channel beginning with the first used advertising channel index and ending with the last used advertising channel index*".

实际上，这意味着当所有三个频道都在使用时（通常是这种情况），广告使用序列37中的频道然后按照严格的顺序使用38然后是39。

In practice, this means that when all three channels are in use, as is often the case, advertising uses channels in the sequence 37 then 38 then 39, in strict order.

To lessen the possibility of persistent packet collisions, where two or more devices advertise on the same channel in an overlapping time period, Bluetooth Core Specification v5.0 stipulates that the time between consecutive advertising events must include a random delay of between 0 and 10ms.

为了减少持续数据包冲突的可能性，当两个或多个设备在重叠的时间段内在同一个通道上进行广告时，蓝牙核心规范v5.0规定，连续广告事件之间的时间必须包括0到10毫秒之间的随机延迟。

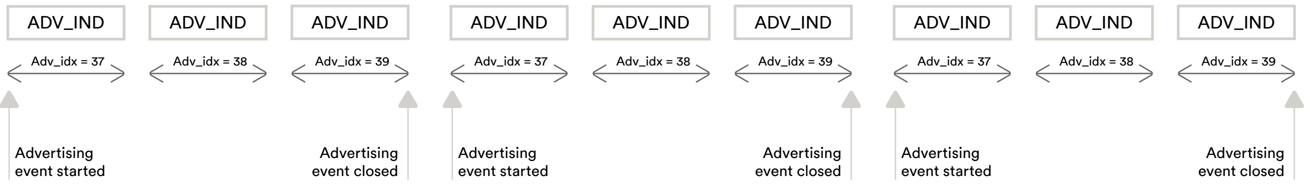

Figure 4 - Advertising channel use per the Bluetooth Core Specification 5.0 with the fixed sequence of 37, 38 then 39

Improved Packet Collision Avoidance

在此版本中，处于广告状态的设备不再需要以严格且不变的顺序选择广告频道，从最低使用频道索引开始到以最高频道索引结束。 现在允许随机选择信道索引。 广告信道索引的随机化进一步降低了广告分组冲突发生的可能性。

In this release, devices in the advertising state are no longer required to select advertising channels in a strict and unchanging sequence, starting with the lowest used channel index and ending with the highest. It is now permissible to select channel indices at random. The randomization of advertising channel indices further reduces the potential for advertising packet collisions occurring.

Applications that use advertising to perform connectionless communication will benefit from improved scalability and reliability in busy radio environments by implementing this change to advertising channel index selection.

使用广告来执行无连接通信的应用程序将受益于在繁忙的无线电环境中通过实现对广告频道索引选择的更改而提高的可扩展性和可靠性。

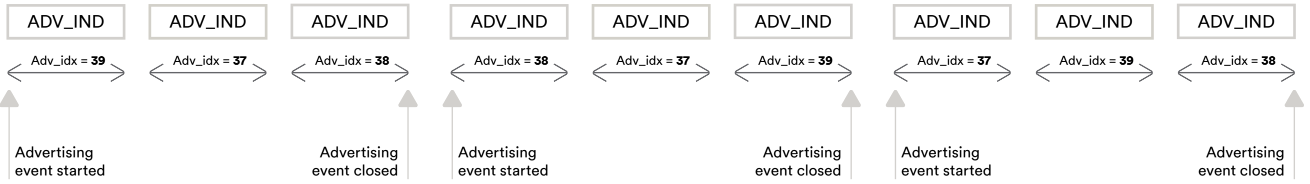

Figure 5 - Advertising channel use per the Bluetooth Core Specification 5.1 with a randomized channel index sequence

-广告频道使用符合蓝牙核心规范5.1的随机频道索引序列

back to contents

4.0 Advertising Enhancement 2: Periodic Advertising Sync Transfer

Background

Bluetooth Core Specification v5.0 introduced periodic advertising that uses deterministic scheduling of advertising events and provides a procedure that devices can use to synchronize their scanning with the advertising schedule of another device. Synchronization of the timing of scanning and advertising can make the scanning device more energy efficient and can make possible some use cases that require precise timing in the exchange of data.

To allow synchronization with the periodic advertising of a remote device, the remote device advertises AUX_ADV_IND PDUs which contain a field called SyncInfo. SyncInfo contains everything the receiving device needs to know to synchronize with the periodic advertising of AUX_SYNC_IND PDUs performed by the remote device from that point on. This periodic advertising synchronization procedure can be a relatively expensive operation, however.

The Power of Two

Some device types, with limited power, may not be able to afford the energy cost associated with the periodic advertising synchronization procedure or may have limitations in duty cycle or scan time that prevent it from working.

The new Periodic Advertising Sync Transfer (PAST) feature allows another, less constrained device to perform the synchronization procedure and then pass the acquired synchronization details over a point-to-point Bluetooth Low Energy connection to the other, constrained device. For example, a smartphone could scan for AUX_SYNC_IND packets from a TV and then pass them over a connection to an associated smart watch so that the watch can then benefit from using periodic advertising and scanning to acquire data from the TV.

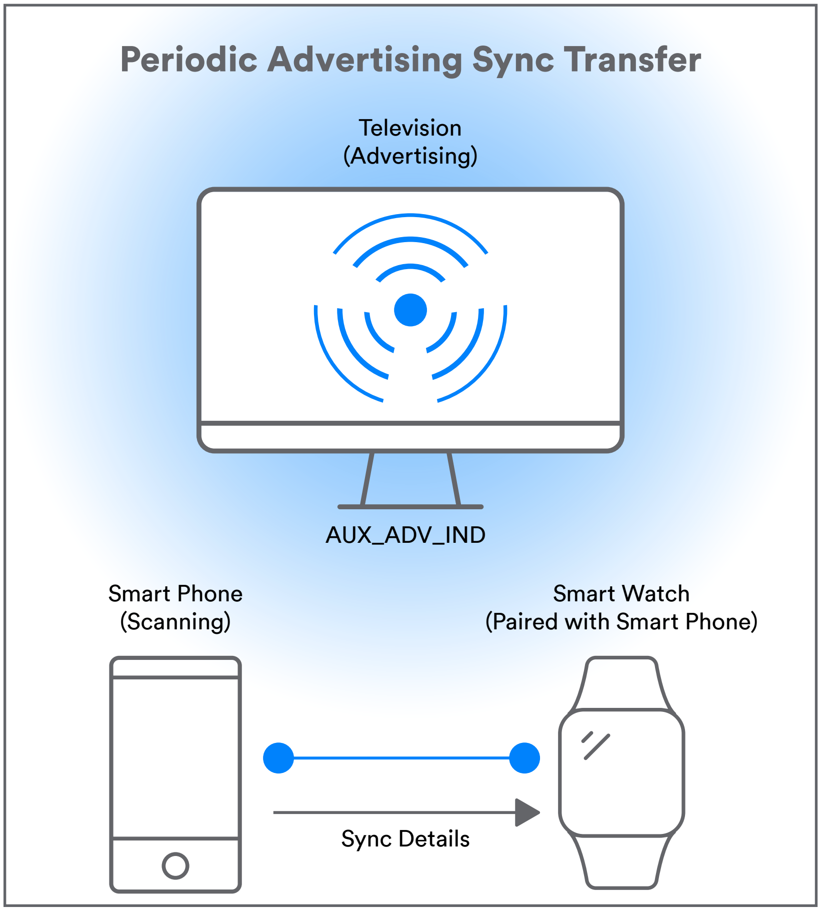

Figure 6 - Periodic Advertising Sync Transfer usage example

Minor Enhancements

A number of minor enhancements are included in this release of the core specification.

HCI Support for Debug Keys in LE Secure Connections

Enhancement

LE secure connections is a Bluetooth pairing procedure that uses the Diffie Hellman key agreement protocol to secure the exchange of shared security keys during pairing. Diffie Hellman uses asymmetric, elliptic curve cryptography with a public and a private key. This makes it impossible to obtain the shared keys and use them for tracing and debugging connections during developing and testing.

In Bluetooth Core Specification v4.2, hard-coded key values for testing purposes were defined. But in cases where the Elliptic Curve algorithms are implemented in the controller, there was no way for the host to indicate it wanted to use them. The latest version of the core specification adds an HCI command that lets the host tell the controller to use the debug key values. Cases where the host implements the Elliptic Curve algorithms itself are not affected by this change.

Sleep Clock Accuracy Update Mechanism

Enhancement

Currently, when establishing an LE connection, the master device informs the slave how accurate its clock is using the Sleep Clock Accuracy (SCA) field. But the accuracy requirement might change depending on the concurrent use cases handled by the controller. For example, it might start at one value but need to be stepped up when another connection with higher clock-accuracy requirements is established.

Bluetooth Core Specification v5.1 provides a new link layer PDU, LL_CLOCK_ACCURACY_REQ, that can be used to inform connected slaves of new clock accuracy values. This PDU may be transmitted either by the master to the slave or by the slave to the master so that slaves may use it to inform masters in a connection of their clock accuracy.

This feature may result in lower power consumption in some cases.

ADI Field in Scan Response Data

Error Correction

The AdvDataInfo (ADI) field is used in extended advertising packets. Previously, this field was not allowed in scan response packets. In the latest core specification release it has become permissible to include ADI in scan response packets.

Interaction Between QoS and Flow Specification

Informative

This change is a clarification of the rules relating to Quality of Service (QoS) and Flow as they relate to Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR).

back to contents

Host Channel Classification for Secondary Advertising

Error Correction

The HCI command LE_Set_Host_Channel_Classification allows the classification of radio channels as “bad”. Previously its use applied only to connections, but now it applies to secondary advertising channels too.

Allow the SID to Appear in Scan Response Reports

Error Correction

The Advertising Set ID (SID) field is used in extended advertising packets. Previously this field was not allowed in scan response packets. In Bluetooth Core Specification v5.1 it has become permissible to include SID in scan response reports.

Specify the behavior when rules are violated

Informative

A new section, “Responding to Invalid behavior” has been added to the latest core specification release to clarify the rules which can be followed when dealing with a badly behaved Bluetooth device.