13.9 Security Considerations

13.9.1 Authentication and Authorization Requirements

As shown in Figure 13.10, each attribute or characteristic defined by GATT profile has its own associated set of authentication and authorization requirements. These are defined independently for each of the attribute or characteristic and could be one of the following:

- Specified by the GATT profile.
- Specified by a higher layer profile.
- Specified by the application.

The authentication and authorization procedures will be explained in Chapter 14. As a thumb rule, the list of services and characteristics supported by a device is considered to have the following permissions:

- Read-Only.
- No Authentication required.
- No Authorization required.

This means that the service declaration, include definition and characteristic declaration are considered read-only and other devices can at least retrieve these without getting an error. The characteristics could be either read-only or write-only or read-write. If a client, for example, tries to access a write-only attribute using one of the read procedures, it gets an Error Response.

13.10 Summary

The GATT profile uses the attributes defined by the ATT protocol as building blocks and defines a hierarchy using those services. This hierarchy organizes the attributes into profiles which may contain services which may in turn contain characteristics. Based on the functionality, similar attributes are grouped together. This provides a very effective and powerful mechanism for devices to exchange information.

GATT provides the facility to discover the services of the remote devices (similar to the services provided by SDP in the BR/EDR world). It then provides facilities to discover the characteristics and included services that are contained in the service. Finally it provides mechanisms to read, write, notify and indicate characteristics.

The GATT based profile architecture abstracts all complexity related to the hierarchy and access of attributes so that the profiles on top are very simple.

Bibliography

Bluetooth Core Specification 4.0 http://www.bluetooth.org.

CHAPTER 14

Generic Access Profile

14.1 Introduction

The Generic Access Profile (GAP) defines the base functionality common to all Bluetooth devices. This includes modes of device and generic procedures related to discovery of the devices in vicinity, connecting to devices, and security. GAP is mandatory to be implemented for all devices that support Bluetooth. The position of GAP in the LE Protocol Stack is shown in Figure 14.1.

In general the profiles help to ensure interoperability between devices from different vendors. This is one of the strong points of Bluetooth technology compared to many other wireless technologies. Bluetooth devices from one vendor can seamlessly work with devices from other vendors provided they have some profiles in common.

The profiles also recommend the terminology to be used at the user interface level so that the end user gets a similar look and feel when using devices from different vendors. For example, GAP recommends the terminology to be used for representing various Bluetooth parameters (Bluetooth Device Address, Device Name, Passkey, etc.) at the user interface level. So, when asking for a user to enter the PIN key, GAP recommends the user interface to refer to it as "Bluetooth Passkey".

Before going further, it is highly recommended to read the sections related to GAP in Chapter 4 to get a good background. This chapter describes the LE specific parts of GAP.

GAP defines the basic requirements of an LE device to include at least the following:

- Physical Layer;
- Link Layer;
- L2CAP;
- Security Manager;
- ATT;
- GATT.

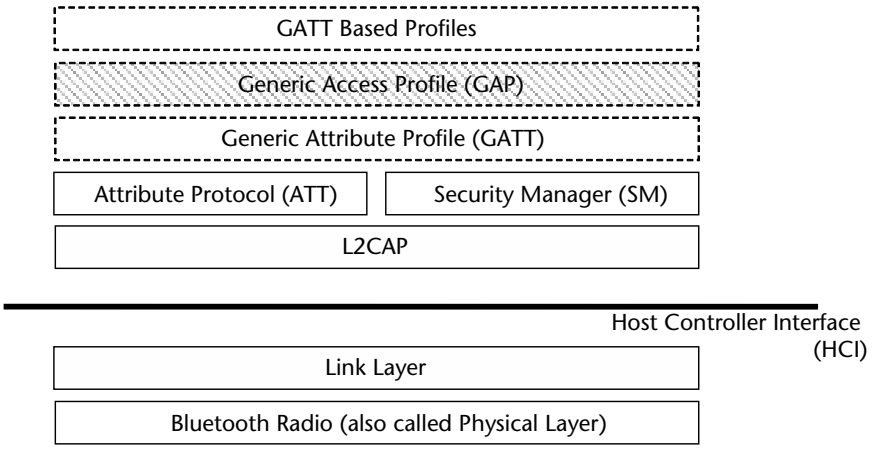


Figure 14.1 GAP in LE protocol stack.

14.2 Roles

GAP defines the following four roles for LE devices.

1. Broadcaster;
2. Observer;
3. Peripheral;
4. Central.

A device may operate in multiple GAP roles at the same time provided the link layer supports this.

14.2.1 Broadcaster Role

A device that is operating in the broadcaster role transmits advertising events periodically. This role is optimized for transmitter only applications. The device may not contain a receiver to optimize the memory footprint and cost. Devices supporting this role use the link layer advertising events to broadcast data. This is shown in Figure 14.2.

14.2.2 Observer Role

A device that is operating in the observer role receives the advertising events. The observer role is optimized for receiver only applications. The device may not contain a transmitter to optimize the memory footprint and cost. The devices supporting these roles receive the data that is sent in the link layer advertisement events. This is shown in Figure 14.2.

14.2.3 Peripheral Role

The device in the Peripheral role is the one that accepts a connection request from another device. This role is optimized for devices that support a single connection

Figure 14.2 Broadcaster and Observer roles.

and act as a Slave in a piconet. A device needs to have both a transmitter and a receiver to support this role. This is shown in Figure 14.3.

14.2.4 Central Role

The device in the Central role is the one that initiates establishment of a physical connection. The Central role supports multiple connections and is the initiator of connections. This device acts as a Master in a piconet and may connect to more than one Slave. This is shown in Figure 14.3.

14.3 Representation of Bluetooth Parameters

GAP states requirements about the generic terms that should be used on the user interface level. These are useful not only when designing user interfaces but also in user manuals, documentation, advertising, etc. This helps to ensure a uniform user experience irrespective of the vendor who builds the device or the application.

14.3.1 Bluetooth Device Address

A Bluetooth Device Address is used to identify a Bluetooth device. An LE-only device can either use a public address or a random address. Public Addresses and Random Addresses were explained in Chapter 8. The term to be used on the user interface level is defined by GAP as "Bluetooth Device Address". It is represented as 12 hexadecimal characters which may or may not be separated into subparts by

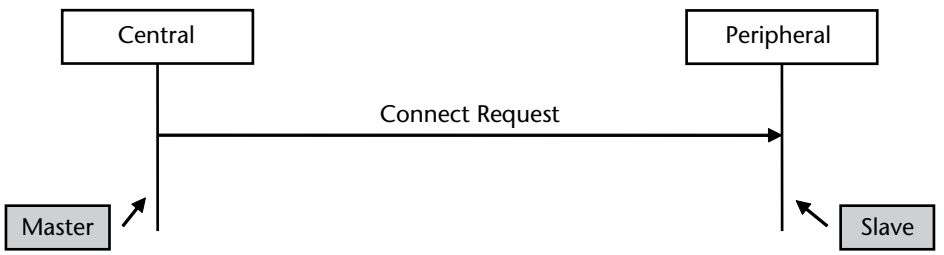


Figure 14.3 Peripheral and Central roles.

the colon symbol. So the Bluetooth Device Address could be represented as 00AABBCCDDEE or 00:AA:BB:CC:DD:EE. If a device supports dual mode (Both BR/EDR and LE), then it uses the same Bluetooth Device Address (also referred to as BD_ADDR) on both the BR/EDR physical channel and LE physical channel.

14.3.2 Bluetooth Device Name

Since it is difficult to refer to devices using a 48-bit address at a user level, the Bluetooth devices expose a user friendly name. This is a character string which is easier to remember, refer to, and distinguish one device from another. For Example 'BT Thermometer', 'SmartPhone', 'Deskjet Printer'.

For a BR/EDR device, the name of the remote device can be fetched using HCI_Remote_Name_Request command. It may also be returned in the Extended Inquiry Result Event if an HCI_Inquiry was done after setting the Inquiry_Mode to Extended Inquiry Result Format.

For an LE only device the name is stored in the Device Name Characteristic. This Characteristic can be read using the GATT procedures. (Note that in the case of BR/EDR, the procedures to fetch the name of the remote device were defined at the link manager level. In the case of LE, they are placed much higher, at the GATT level. This is another step towards making the link layer much simpler in case of LE). The name is up to 248 octets in length and is UTF-8 encoded. It is Null terminated if the length is less than 248 octects. The term to be used on the UI level is "Bluetooth Device Name."

14.3.3 Bluetooth Passkey

The Bluetooth Passkey is used to authenticate two Bluetooth devices during the pairing process. For LE devices, the Bluetooth Passkey is a 6-digit numerical value and is represented in the integer range 000000 to 999999. The term to be used on the UI level is "Bluetooth Passkey". Similar to BR/EDR, a device could either have a fixed passkey or it may be entered by the user. The devices that have no mechanism to enter the passkey (no keypad) would use a fixed passkey.

14.3.4 Bluetooth Class of Device

The Class of Device is valid for only BR/EDR devices and not used for LE-only devices.

14.3.5 Pairing—Authentication and Bonding

Pairing is performed by the Security Manager as explained in Chapter 11. If the user initiates the pairing procedure to create a bond (trusted relationship) between the two devices, then the procedure is referred to as Bonding. If the user is requested to enter the Passkey as a part of the connection establishment procedure, then the procedure is referred to as Authentication.

14.4 Advertising and Scan Response Data Format

The Advertiser can send data to the devices that are scanning using the various types of advertising events. In addition if the Scanner requests for additional information then the Advertiser can provide additional data using the SCAN_RSP advertising packets. This is shown in Figure 14.4.

The Advertising Data is sent in the AdvData field of the following packets:

- Connectable Undirected (ADV_IND).
- Non-Connectable Directed (ADV_NONCONN_IND).
- Scannable Undirected (ADV_SCAN_IND).

The Scan Response Data is sent in the ScanRspData field of the SCAN_RSP packets. These packets were explained in Chapter 8. Note from Chapter 8 that the Connectable Directed (ADV_DIRECT_IND) packets do not contain any host data. Both the AdvData and ScanRspData are 0 to 31 octets in length. These contain a sequence of Advertising Data (AD) structures. The AD structure is shown in Figure 14.5.

The AD Structure contains three fields:

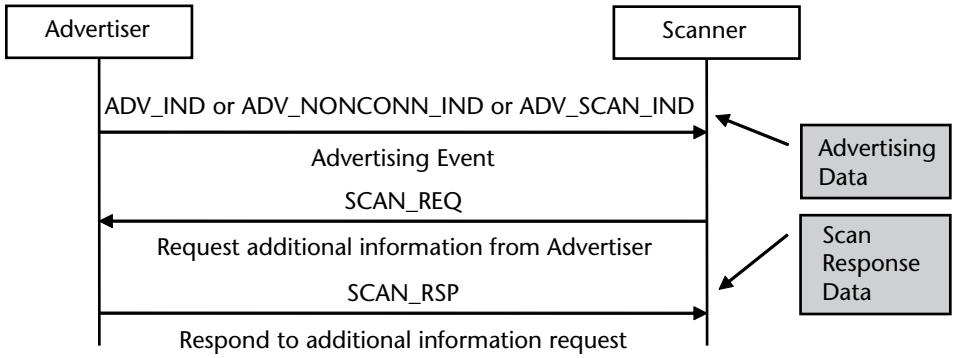


Figure 14.4 Advertising events containing advertising and scan response data.

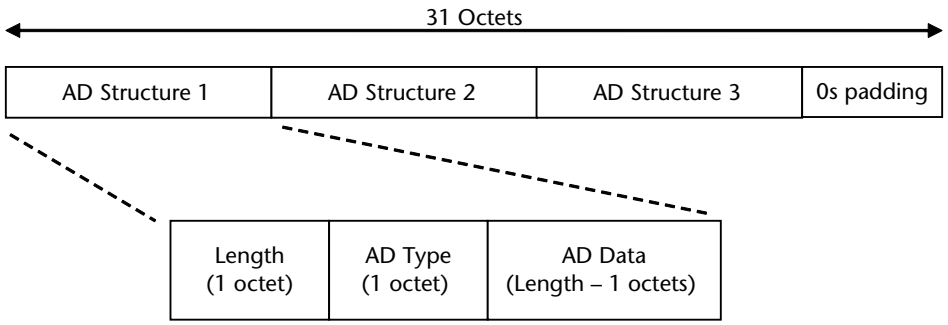


Figure 14.5 Format of advertising data and scan response data.

1. Length.
2. AD Type.
3. AD Data.

The AD Type field is used to specify the type of data contained in AD Data. The different values of AD Type field are listed in the Bluetooth Assigned Numbers Document. Some of the commonly used AD Type fields are explained below.

14.4.1 Local Name (AD Type = 0x08 or 0x09)

This is used to specify the device name. If the device name is too long, then a shortened version can be provided in this AD Type and the full name can be read from the device name characteristic. The AD Type field indicates the local name as follows:

- 0x08: Shortened local name.
- 0x09: Complete local name (If it can fit the AdvData).

14.4.2 Flags (AD Type = 0x01)

The Flag AD Type is a bit map used to provide information about the device. The various values are shown in Table 14.1. The value can be a combination of these values. For example a value of 0x06 would indicate:

- LE General Discoverable Mode.
- BR/EDR not supported.

14.4.3 Manufacturer Specific Data (AD Type = 0xFF)

This is used to specify the company identifier and any manufacturer specific information.

Figure 14.6 shows an example of ADV_IND packets that contain AdvData. In this example, the ADV_IND packet contains a payload of 23 bytes, out of which Advertiser Address is 6 bytes and AdvData is 17 bytes.

The AdvData contains two AD structures:

Table 14.1 Flags Contained in AdvData.

Value	*Information*
0x01	LE Limited Discoverable Mode
0x02	LE General Discoverable Mode
0x04	BR/EDR Not Supported
0x08	Simultaneous LE and BR/EDR to the same device capable (controller)
0x10	Simultaneous LE and BR/EDR to the same device capable (host)

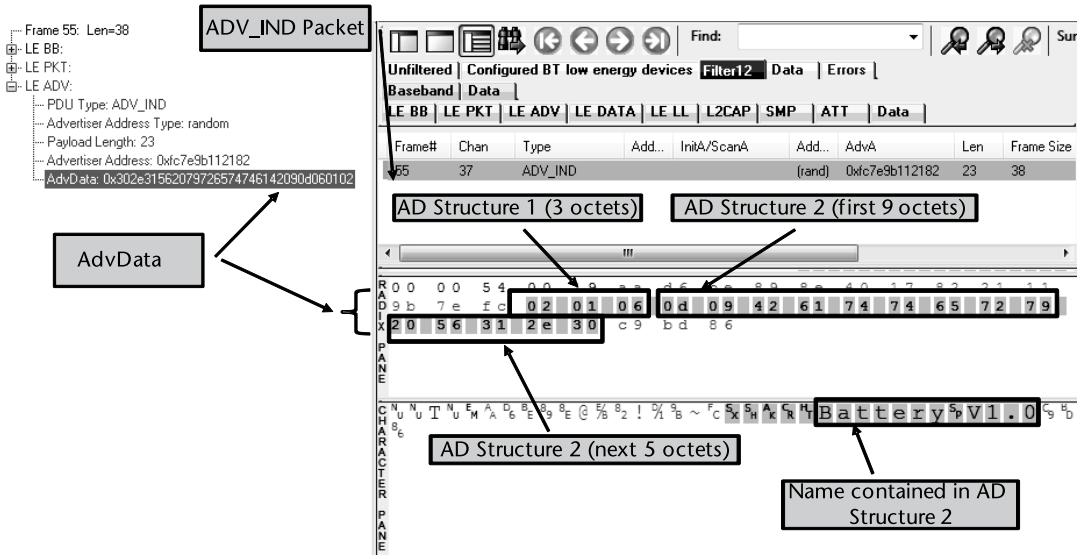


Figure 14.6 Example of AdvData contained in ADV_IND packets.

- AD Structure 1: 3 octets: It contains [02 01 06]. This is decoded as follows:
  - Length = 02.
  - AD Type = 01 (Flags).
- AD Data = 06. This means:
  - The device is currently in LE Generate Discoverable Mode.
  - The devices does not support BR/EDR.
- AD Structure 2: 14 octets. It contains [0d 09 42 61 74 74 65 72 79 20 56 31 2e 30]. This is decoded as follows:
  - Length = 0d.
  - AD Type = 09 (complete local name).
  - AD Data = 42 61 74 74 65 72 79 20 56 31 2e 30.
- When converted to ascii characters, this means "Battery V1.0."

14.5 GAP Characteristics

GAP defines certain characteristics to provide further information about the device using the GAP service. There can be only one instance of the GAP service on a device. The Service UUID of the GAP service is <<Generic Access Profile>>. Some of the commonly used GAP characteristics are explained below.

14.5.1 Device Name Characteristic

The Device Name Characteristic contains the name of the device as a UTF-8 encoded string. It can be 0 to 248 octets long. It has the following fields:

- Attribute Type: 0x2A00 – UUID for <<Device Name>>.
- Attribute Value: Device Name: 0 to 248 octets in length.

The air log for ATT Read Response for reading the Device Name Characteristic is shown in Figure 14.7. As shown in the figure, the Master executes the following steps to read the device name:

1. Frame #283: The Master sends a Read Request to read the Characteristic for Attribute Handle 2. (UUID = Characteristic).
2. Frame #286: The Slave sends a Read Response indicating that the Characteristic contains Device Name and the Attribute Handle is 3.
3. Frame #289: The Master sends a Read Request to read the Device Name (UUID = Device Name).
4. Frame #292: The Slave sends a Read Response which contains the Device Name.

14.5.2 Appearance Characteristic

The appearance characteristic is used by the discovering devices to associate an icon (or string) to this device. The various values of the appearance characteristic are defined in the Bluetooth Assigned Numbers document. This can be compared to the Class of Device (CoD) in the case of BR/EDR. The remote devices use the CoD parameter to display an icon for the device (for example, icon of a headset, mobile phone or printer). In the case of LE, the remote devices could fetch the appearance characteristic to find the type of the device and then display an icon or any other user interface that is appropriate for that device. It has the following fields:

- Attribute Type: 0x2A01 – UUID for <<Appearance>>
- Attribute Value: Appearance (as per values defined in Bluetooth Assigned Numbers): 2 octets in length.

The air log for ATT Read Response for reading the Appearance Characteristic is shown in Figure 14.8. As shown in the figure, the Master executes the following steps to read the Appearance:

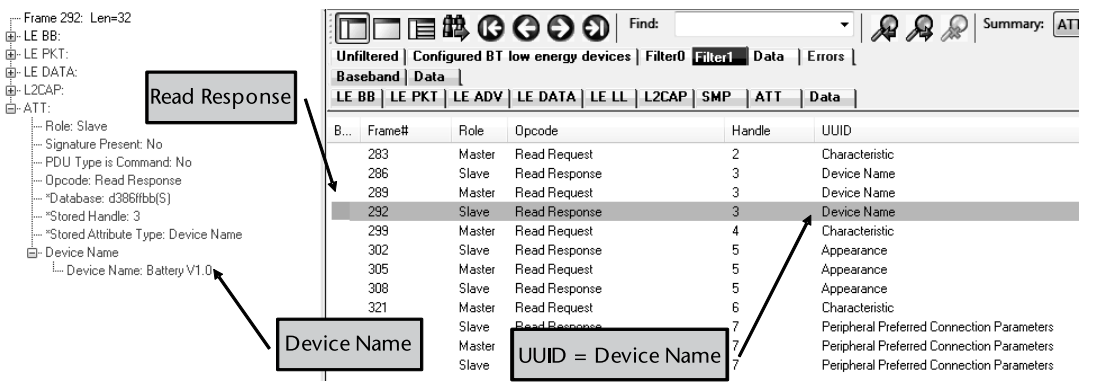


Figure 14.7 Device name characteristic.

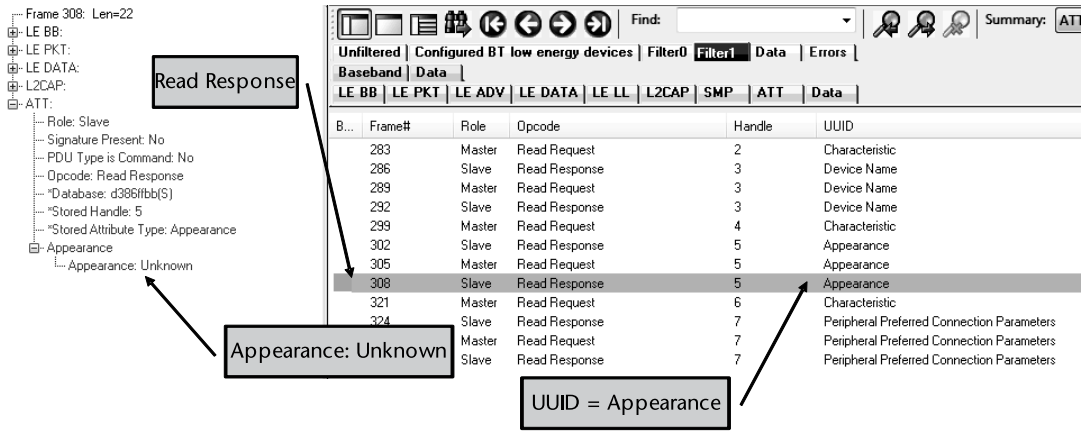


Figure 14.8 Appearance characteristic.

1. Frame #299: The Master sends a Read Request to read the Characteristic for Attribute Handle 4. (UUID = Characteristic).
2. Frame #302: The Slave sends a Read Response indicating that the Characteristic contains Appearance and the Attribute Handle is 5.
3. Frame #305: The Master sends a Read Request to read the Appearance (UUID = Appearance).
4. Frame #308: The Slave responds with the Appearance characteristic. It contains the value "Unknown Appearance".

14.5.3 Peripheral Privacy Flag Characteristic

The Peripheral Privacy Flag Characteristic defines whether privacy is currently in use within this device or not.

Setting this flag to 1 indicates that privacy is enabled in this device. It has the following fields:

- Attribute Type: 0x2A02 – UUID for <<Peripheral Privacy Flag>>
- Attribute Value: Peripheral Privacy Flag: 1 octet in length
  - 0x00: Indicates that privacy is disabled in this device.
  - 0x01: Indicates that privacy is enabled in this device.
  - Remaining values are reserved.

The Peripheral may implement this characteristic as only readable or readable/writable. If this characteristic is writable, then the central device can remotely enable or disable privacy on this device.

14.5.4 Reconnection Address Characteristic

The Reconnection Address is used to store the address to connect to next time. It may be exchanged between the two devices at each connection. If the device changes its address, then it can provide the new reconnection address to the peer

device so that the peer device knows which address to connect to the next time. It has the following fields:

- Attribute Type: 0x2A03 – UUID for <<Reconnection Address>>
- Attribute Value: Reconnection Address.

14.5.5 Peripheral Preferred Connection Parameters Characteristic

The peripheral preferred connection parameters (PPCP) characteristic contains the various preferred connection parameters of a peripheral. This includes parameters like minimum and maximum connection interval, Slave latency, and connection supervision timeout multiplier. It has the following fields:

- Attribute Type: 0x2A04 – UUID for <<Peripheral Preferred Connection Parameters>>
- Attribute Value: Peripheral Preferred Connection Parameter: 8 octets in length.

The air log for ATT Read Response for reading the PPCP Characteristic is shown in Figure 14.9. As shown in the figure, the Master executes the following steps to read the PPCP:

1. Frame #321: The Master sends a Read Request to read the Characteristic for Attribute Handle 6. (UUID = Characteristic).
2. Frame #324: The Slave sends a Read Response indicating that the Characteristic contains PPCP and the Attribute Handle is 7.
3. Frame #325: The Master sends a Read Request to read the Appearance (UUID = PPCP).
4. Frame #328: The Slave responds with the PPCP characteristic. It contains the 8-octets:"ff ff ff ff 00 00 ff ff". The meaning of these octets is shown in Table 14.2.

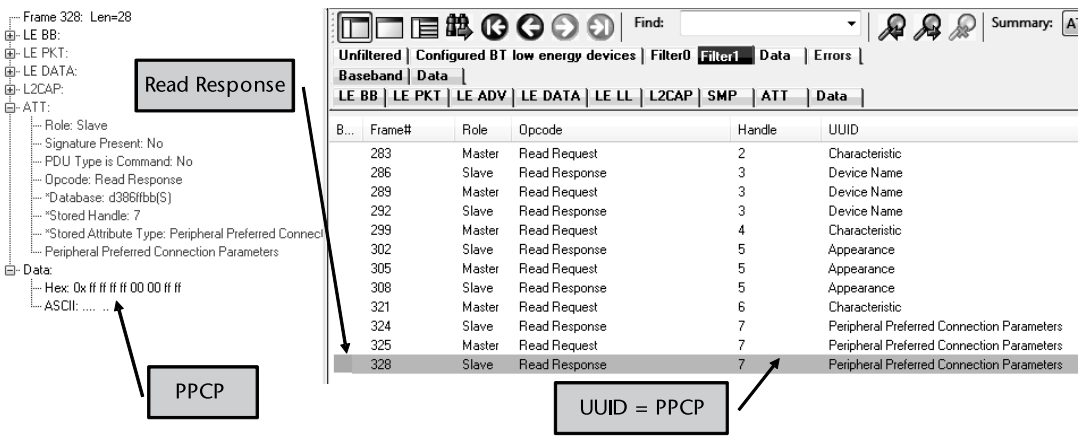


Figure 14.9 Peripheral preferred connection parameters characteristic.

Table 14.2 Peripheral Preferred Connection Parameters Data

Name	*Size*	*Value*	*Meaning*
Minimum Connection Interval	2	ff ff	No specific minimum
Maximum Connection Interval	2	ff ff	No specific maximum
Slave Latency	2	00 00	Slave latency of the connection in terms of number of connection events
Connection Supervision Timeout Multiplier	2	ff ff	No specific value requested

14.6 Operational Modes and Procedures

GAP defines the various modes in which an LE device can be in as well as the procedures that can be carried out. It broadly defines the following four modes and procedures:

1. Broadcast mode and observation procedure.
2. Discovery modes and procedures.
3. Connection modes and procedures.
4. Bonding modes and procedures.

14.6.1 Broadcast Mode and Observation Procedure

The broadcast mode is the simplest mode which allows an LE device to send data to other LE devices using advertising events. The device that sends the data is termed as broadcaster and the device that receives the data is termed as observer.

14.6.1.1 Broadcast Mode

The broadcaster sends data in either nonconnectable undirected (ADV_NONCONN_IND) or scannable undirected advertising events (ADV_SCAN_IND). These events were explained in Chapter 8. Both these events permit sending data from the Advertiser to the Scanner but do not permit the Scanner to connect to the Advertiser. Since there is no acknowledgement from any device, the data sent in broadcast mode is considered unreliable.

14.6.1.2 Observation Procedure

The observation procedure is used by the observer to receive data from the broadcaster. The device can use either passive scanning or active scanning to receive the advertising events. As explained in Chapter 8, in the passive scanning mode the device just receives the advertising data while in the active scanning mode the device may also request some additional information from the Advertiser using SCAN_REQ.

14.6.2 Discovery Modes and Procedures

The discovery modes and procedures are used to discover other devices in the vicinity. There are three discovery modes and three discovery procedures in this category. The discovery modes are implemented by the device acting in the Peripheral role. (These are excluded for device acting in the Central role since the Central device does the discovery and is not the one that is to be discovered) The three discovery modes are shown in Table 14.3.

The discovery procedures are implemented by the device to discover the devices in the vicinity. Out of these the first two are implemented by the device acting in the Central role while the third can be implemented by both the Central and Peripheral device. The three discovery procedures are shown in Table 14.4.

14.6.2.1 Nondiscoverable Mode

A device in nondiscoverable mode cannot be discovered by other devices. In this mode either the device does not send any advertising packets or if it sends advertising packets then it does not set the LE General Discoverable Mode and LE Limited Discoverable Mode flags in the AD type.

14.6.2.2 Limited Discoverable Mode

A device configured in limited discoverable mode can be discovered by other devices for only a limited period of time. An example of this could be a device which sends advertisements for a limited period of time when the user presses a button on it. A device in limited discoverable mode can send three types of events:

- Nonconnectable advertising events (ADV_NONCONN_IND).
- Scannable undirected advertising events (ADV_SCAN_IND).
- Connectable undirected advertising events (ADV_IND).

Table 14.3 Discovery Modes

S. No.	*Mode*	*Mandatory/Optional*
1	Nondiscoverable Mode	Mandatory for Peripheral. Excluded for Central.
2	Limited Discoverable Mode	Optional for Peripheral. Excluded for Central.
3	General Discoverable Mode	Mandatory if Limited Discoverable mode is not implemented, else optional. Excluded for Central.

Table 14.4 Discovery Procedures

S. No.	*Procedure*	*Mandatory/Optional*
1	Limited Discovery Procedure	Optional for Central. Excluded for Peripheral
2	General Discovery Procedure	Mandatory for Central. Excluded for Peripheral
3	Name Discovery Procedure	Optional for both Peripheral and Central. Can be invoked by either of these.

A device in limited discoverable mode sets the LE Limited Discoverable Mode flag to one and LE General Discoverable Mode flag to zero in the AD Type. The Limited Discoverable Mode is shown in Figure 14.10.

14.6.2.3 Limited Discovery Procedure

A device that is performing limited discovery procedure receives the data from the devices in limited discoverable mode. The host receives the advertising packets from all the devices and checks the AD Type field in the advertisement packets. If the AD Type field is present and has the LE Limited Discoverable flag set to one then it adds that device to the list of devices found during the limited discovery procedure.

Besides the flag, the advertising data may contain other AD Types like local name, service UUIDs, etc. This data may also be used by the host for other procedures. The Limited Discovery Procedure is shown in Figure 14.10.

14.6.2.4 General Discoverable Mode

A device configured in general discoverable mode can be discovered by other devices that are performing the general discovery procedure. This is typically used when the device wishes to remain in the discoverable mode for a long period of time. A device in general discoverable mode can send three types of events:

- Nonconnectable advertising events (ADV_NONCONN_IND).
- Scannable undirected advertising events (ADV_SCAN_IND).
- Connectable undirected advertising events (ADV_IND).

A device in general discoverable mode sets the LE Limited Discoverable Mode flag to zero and LE General Discoverable Mode flag to one in the AD Type. The General Discoverable Mode is shown in Figure 14.11.

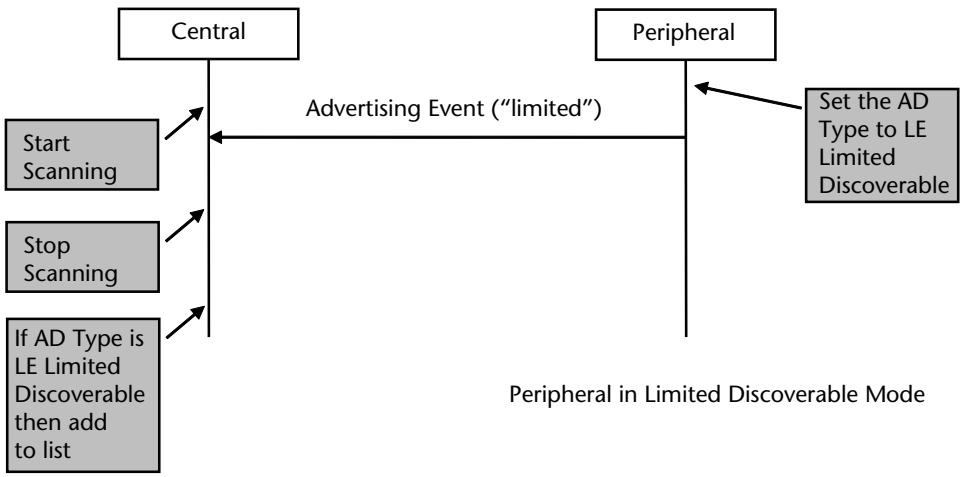


Figure 14.10 Limited discoverable mode and limited discovery procedure.

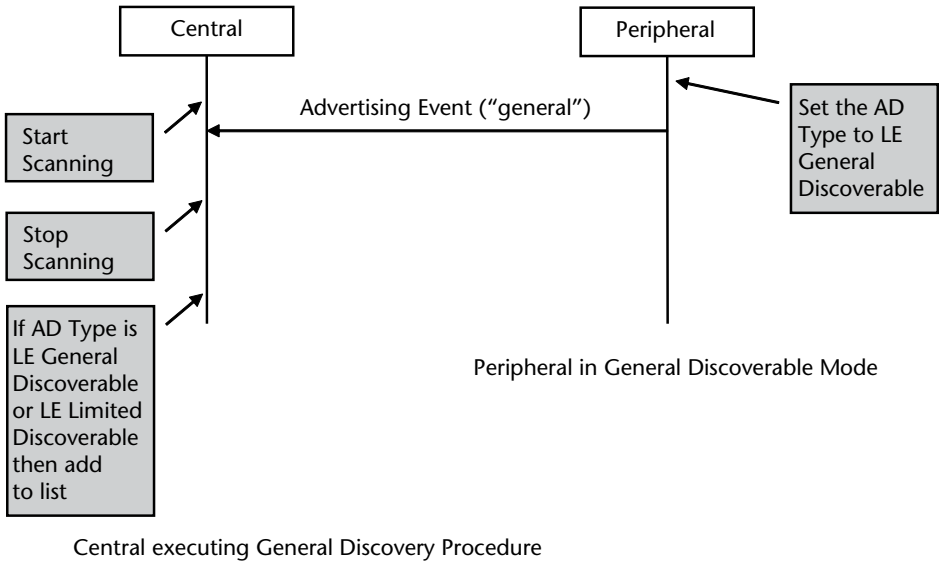


Figure 14.11 General discoverable mode and general discovery procedure.

14.6.2.5 General Discovery Procedure

A device that is performing general discovery procedure receives the data from the devices in general discoverable mode or limited discoverable mode. The host receives the advertising packets from all the devices and checks the AD Type field in the advertisement packets. If the AD Type field is present and has either the LE Limited Discoverable flag or LE General Discoverable flag set to one then it adds that device to the list of devices found during the general discovery procedure.

Besides the flag, the advertising data may contain other AD Types like local name, service UUIDs etc. This data may also be used by the host for other procedures. The General Discovery Procedure is shown in Figure 14.11.

14.6.2.6 Name Discovery Procedure

The name discovery procedure is used to obtain the Bluetooth Device Name of a remote device. It can be invoked by either the device in Central role or the device in Peripheral role. The Bluetooth name of the LE device is present at two places:

1. In the Advertising Event if the AD Type in the AD structure is set to Local Name.
   - This name is acquired during the general discovery procedure or limited discovery procedure.
   - It is possible that the full name may not be provided in the advertising events since the advertising events are limited to 31 octets. In such cases a shortened name may be provided in the advertising event.
2. Device Name Characteristic.

If the complete device name is not acquired using step (1) above, then the name discovery procedure may be performed.

The name discovery procedure uses the following steps:

1. Establish a connection using the connection establishment procedures.
2. Read the Device Name Characteristic using the GATT procedure Read Using Characteristic UUID.
3. Terminate the connection if it's not needed any longer.

The Name Discovery Procedure is shown in Figure 14.12

14.6.3 Connection Modes and Procedures

The connection modes and procedures are used to establish a connection. There are three connection modes and six connection procedures in this category. The connection modes are entered by the device in the Peripheral role and the discovery procedures are initiated by the device in Central role. The terminate connection

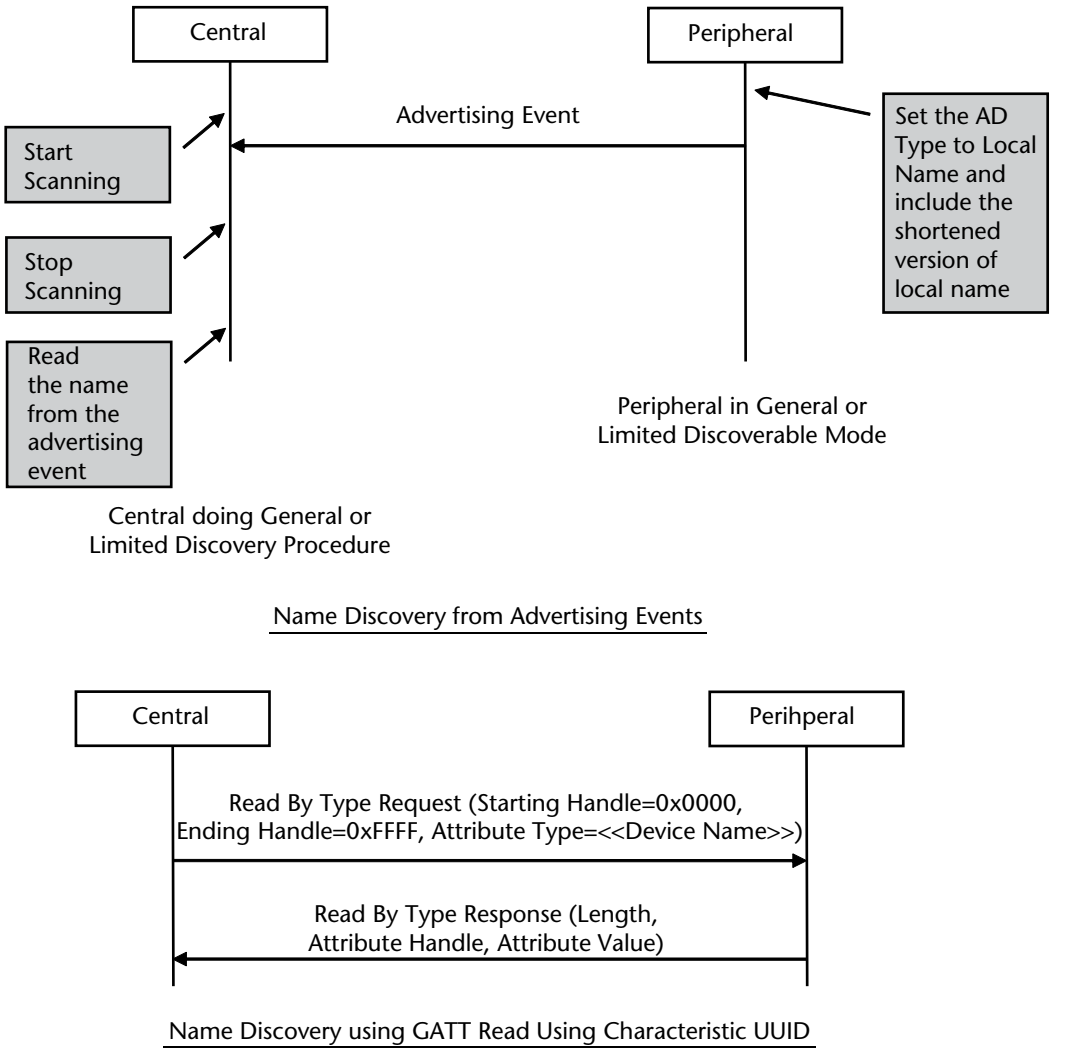


Figure 14.12 Name discovery procedure.

procedure and update connection parameter procedure can be initiated by both the Peripheral and Central. The three connection modes are shown in Table 14.5.

The six discovery procedures are shown in Table 14.6.

14.6.3.1 Nonconnectable Mode

In the Nonconnectable Mode, the device does not allow connection establishment. This mode is entered by the Peripheral device when it does not intend to allow the other devices to connect to it.

The Peripheral can send following two types of events in this mode:

1. Nonconnectable undirected advertising events.
2. Scannable undirected advertising events.

As explained in Chapter 8, both these events don't allow the remote device to establish a connection. The Scannable undirected advertising event allows the remote device to get additional information from the Peripheral.

14.6.3.2 Directed Connectable Mode

In the Directed Connectable Mode, the device allows connections from only a specified peer device. In this mode the Peripheral devices uses the directed connectable advertising events (ADV_DIRECT_IND). It specifies the address of the peer device that is allowed to connect to it in the directed connectable advertising event.

The peer device which is acting as a Central can establish a connection with it. After a connection is established, the Central device becomes the Master and the Peripheral device becomes the Slave. Once the connection is established, the device exits the Directed Connectable Mode and enters the Nonconnectable Mode. This mode is shown in Figure 14.13.

Table 14.5 Connection Modes (All Modes Excluded for Broadcaster and Observer except Nonconnectable Mode)

S. No.	*Mode*	*Mandatory/Optional*
1	Nonconnectable Mode	Mandatory for Peripheral, Broadcaster and Observer. Excluded for Central.
2	Directed Connectable Mode	Optional for Peripheral. Excluded for all other roles.
3	Undirected Connectable Mode	Mandatory for Peripheral. Excluded for all other roles.

Table 14.6 Discovery Procedures (All Procedures Excluded for Broadcaster and Observer)

S. No.	*Procedure*	*Mandatory/Optional*
1	Auto Connection Establishment Procedure	Optional for Central. Excluded for Peripheral
2	Direct Connection Establishment Procedure	Mandatory for Central. Excluded for Peripheral
3	General Connection Establishment Procedure	Mandatory for Central if Privacy feature is supported, else optional. Excluded for Peripheral
4	Selective Connection Establishment Procedure	Optional for Central. Excluded for Peripheral
5	Connection Parameter Update Procedure	Mandatory for Central. Optional for Peripheral
6	Terminate Connection Procedure	Mandatory for Central and Peripheral

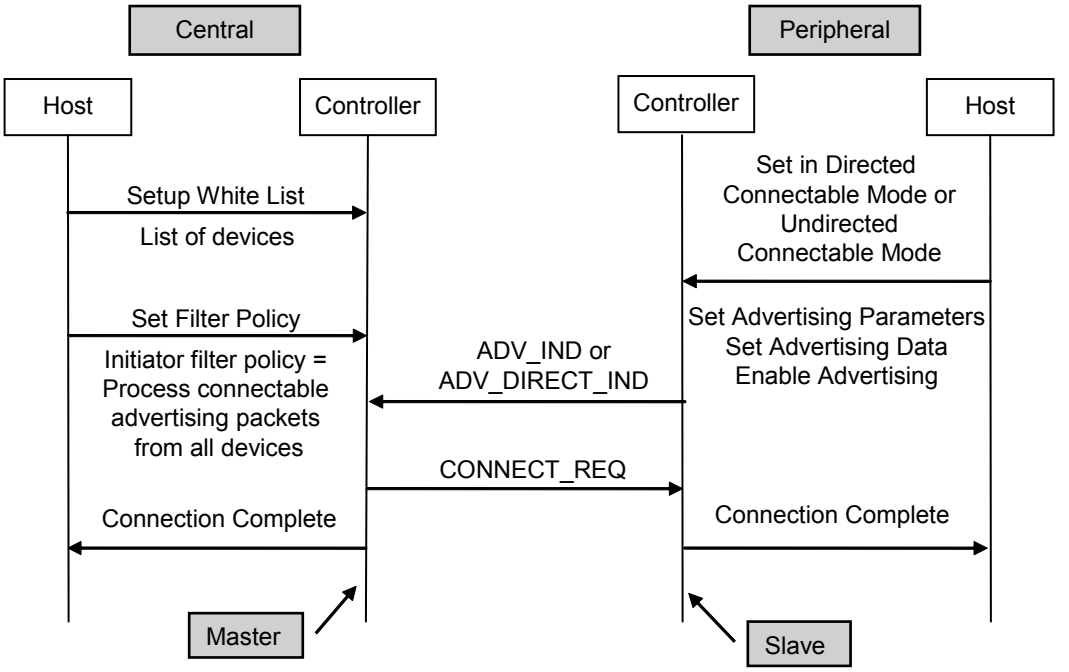


Figure 14.13 Directed connectable mode, undirected connectable mode, and auto connection establishment procedure.

14.6.3.3 Undirected Connectable Mode

In the Undirected Connectable Mode, the device allows connections from any peer device. In this mode the Peripheral devices uses the undirected connectable advertising events (ADV_IND).

The peer device which is acting as a Central can establish a connection with it. After a connection is established, the Central device becomes the Master and the Peripheral device becomes the Slave. Once the connection is established, the device exits the Undirected Connectable Mode and enters the Nonconnectable Mode. This mode is shown in Figure 14.13.

14.6.3.4 Auto Connection Establishment Procedure

In Auto Connection Establishment Procedure the host of the Central configures the controller to autonomously establish a connection with a device in the directed connectable mode or undirected connectable mode.

The list of devices to which the controller can establish a connection autonomously is specified using a White List. The initiator White List is used for this purpose. If the peer device address matches one of the addresses stored in the White List, then the controller establishes a connection autonomously. (White List was explained in Chapter 8). This is shown in Figure 14.13.

14.6.3.5 Direct Connection Establishment Procedure

In Direct Connection Establishment Procedure the host of the Central establishes a connection to a particular peer device. White Lists are not used in this procedure.

The host simply specifies the address of the peer device to connect to when initiating the connection establishment. This is shown in Figure 14.14.

14.6.3.6 General Connection Establishment Procedure

In General Connection Establishment Procedure the host of the Central configures the controller to connect to a particular device that may be in either directed connectable mode or undirected connectable mode.

In this procedure, the host of the Central broadly performs the following steps:

1. The host first scans for advertising packets to find out the list of devices that are present in the vicinity and can be connected to.
2. After getting the list of devices that are present, the host selects one of the devices to which the connection is to be made (for example, by displaying a UI to the user to select the device to connect to).
3. The host then uses the direct connection establishment procedure to establish a connection to the selected device. (The direct connection establishment procedure was explained earlier).

White Lists are not used in this procedure since the host selects the device to connect to instead of giving a list of device to connect to autonomously. This is shown in Figure 14.15.

14.6.3.7 Selective Connection Establishment Procedure

In Selective Connection Establishment Procedure the host of the Central configures the controller to connect to one of the devices in a set of devices specified in a White List.

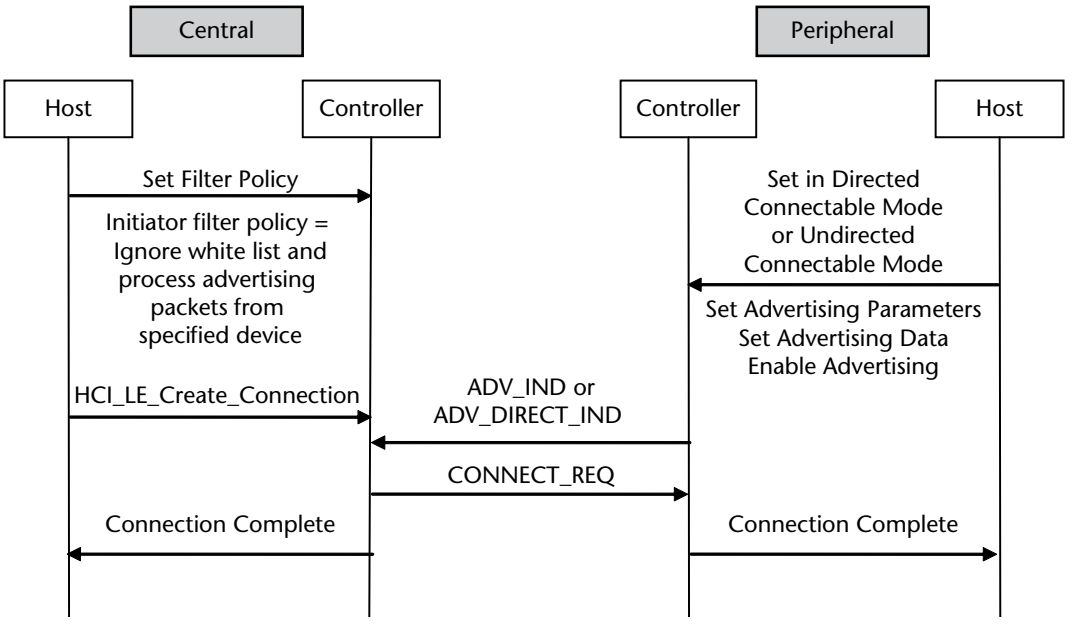


Figure 14.14 Direct connection establishment procedure.

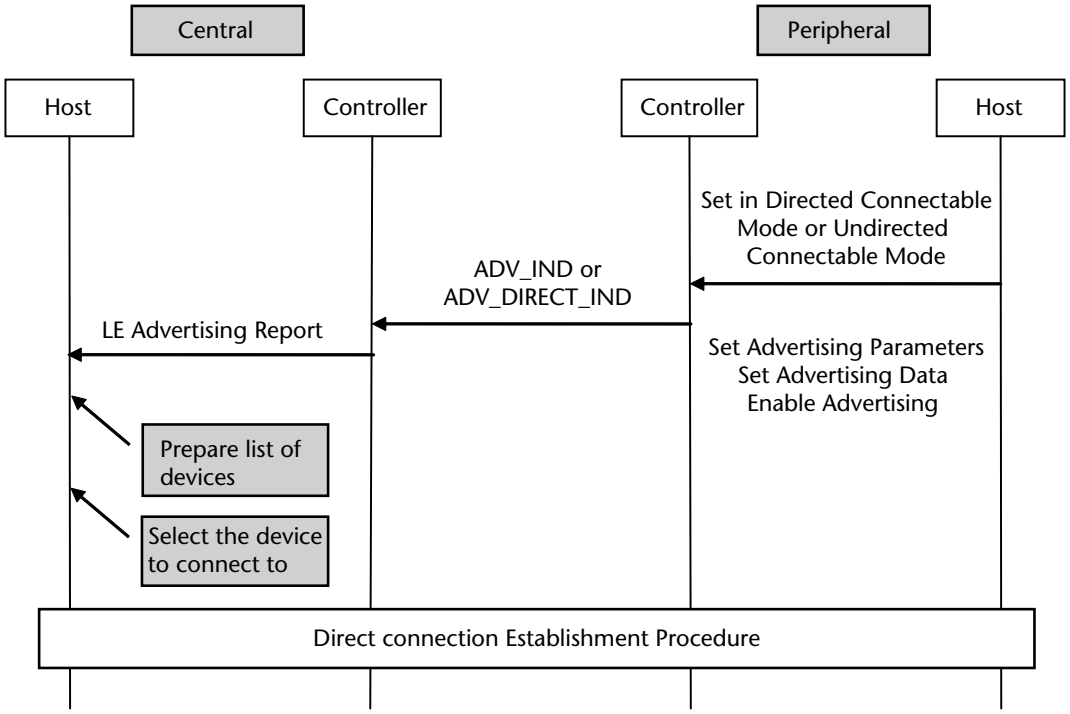


Figure 14.15 General connection establishment procedure.

In this procedure, the host of the Central broadly performs the following steps:

1. The host writes the list of devices that it may connect to into the White List.
2. The host sets the scanner filter policy so that it gets the advertising packets from only the devices in the white list.
3. The host starts scanning.
4. When one of the devices in the white list is discovered, the host stops scanning and initiates a connection to the discovered device using direct connection establishment procedure. (The direct connection establishment procedure was explained earlier).

This is shown in Figure 14.16.

14.6.3.8 Connection Parameter Update Procedure

The Connection Parameter Update Procedure is used by either the Peripheral or Central to update the link layer parameters of the current connection.

There are two possible scenarios:

1. Connection Parameter Update initiated by the Central. In this scenario the link layer connection update procedure is used.
2. Connection Parameter Update initiated by the Peripheral. In this scenario the L2CAP connection parameter update procedure is used.

Both the scenarios are shown in Figure 14.17.

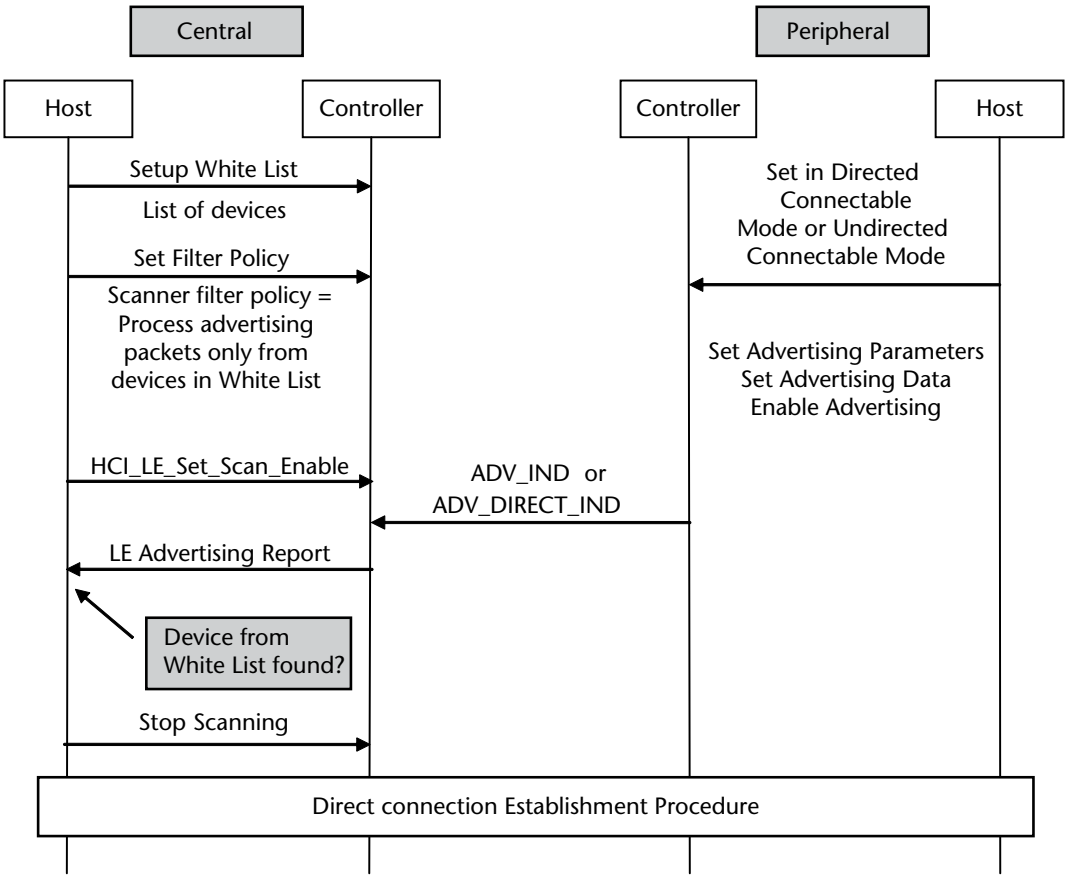


Figure 14.16 Selective connection establishment procedure.

14.6.3.9 Terminate Connection Procedure

The Terminate Connection Procedure is used by either the Peripheral or Central to terminate the current connection. This procedure is shown in Figure 14.18.

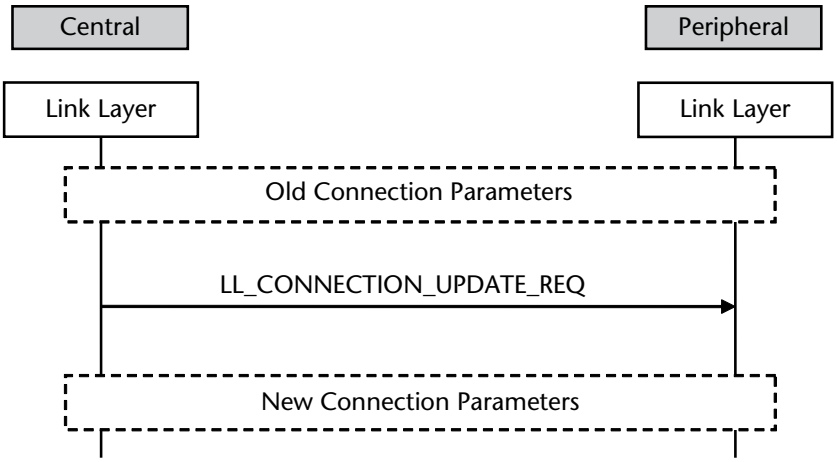
14.6.4 Bonding Modes and Procedures

The bonding procedure allows two devices to create a trusted relationship between them. There are two bondable modes and one bonding procedure in this category. The bondable modes are implemented by both the device acting in Central role and Peripheral role. The two bondable modes are shown in Table 14.7.

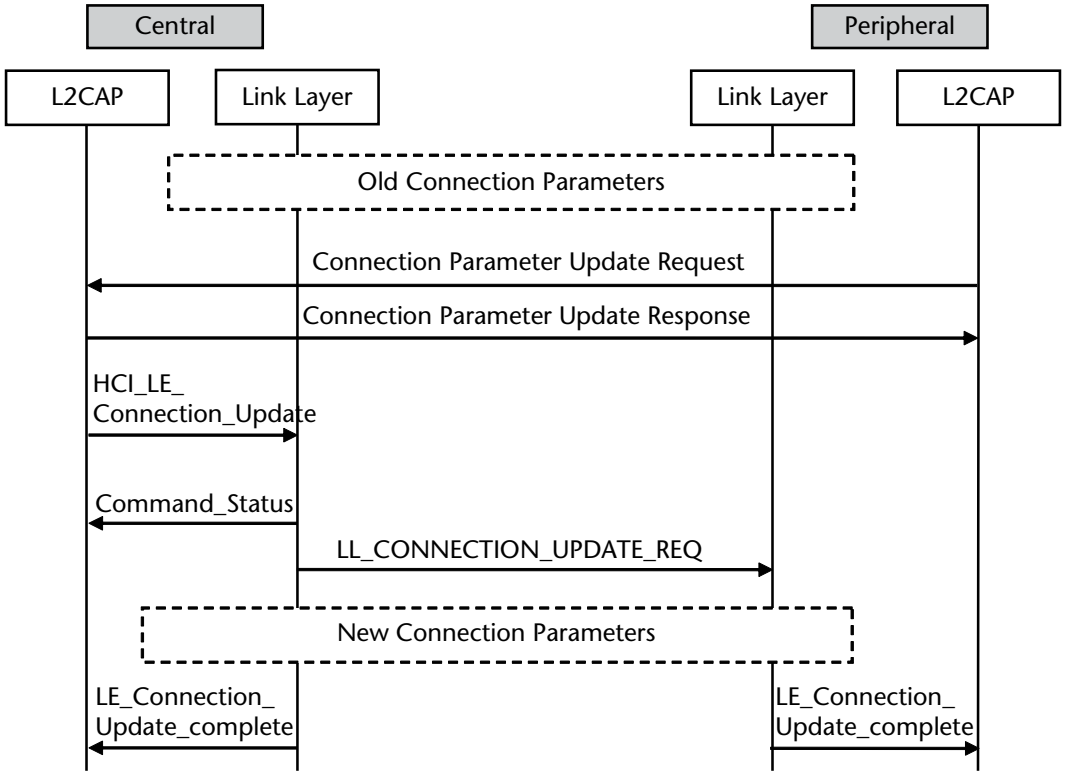
The bonding procedure is implemented by both the device acting in Central role and Peripheral role. The one bonding procedure is shown in Table 14.8.

14.6.4.1 Nonbondable Mode

In this mode, a device does not allow a bond to be created with a peer device.



Scenario 1: Connection Parameter Update Procedure Initiated by Central



Scenario 2: Connection Parameter Update Procedure Initiated by Peripheral

Figure 14.17 Connection parameter update procedure.

14.6.4.2 Bondable Mode

In this mode, a device allows a bond to be created with a peer device. This is shown in Figure 14.19.