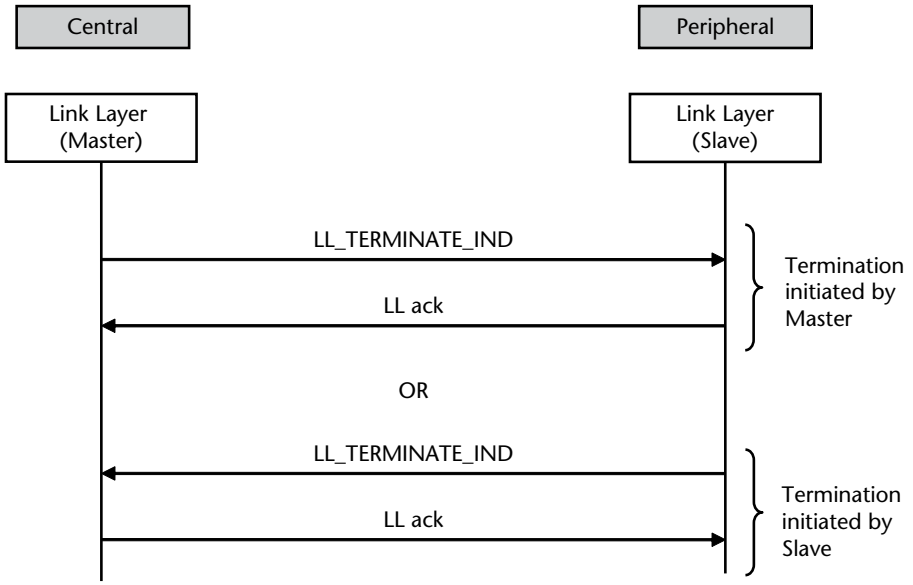

Figure 14.18 Terminate connection procedure.

Table 14.7 Bonding Modes

S. No.	*Mode*	*Mandatory/Optional*
1	Non-Bondable Mode	Mandatory for Peripheral and Central
2	Bondable Mode	Optional for Peripheral and Central

Table 14.8 Bonding Procedure

S. No.	*Procedure*	*Mandatory/Optional*
1	Bonding Procedure	Optional for Peripheral and Central

14.6.4.3 Bonding Procedure

The Bonding Procedure is performed, for example, when a device needs to access a service on a peer device that requires bonding. It is initiated by the Security Manager of the Central. If the peer device is in Bondable Mode, the two devices exchange and store the bonding information in the security database. This is shown in Figure 14.19.

14.7 Security

Implementation of security is optional for LE devices. It is implemented if the access to any of the services is to be protected.

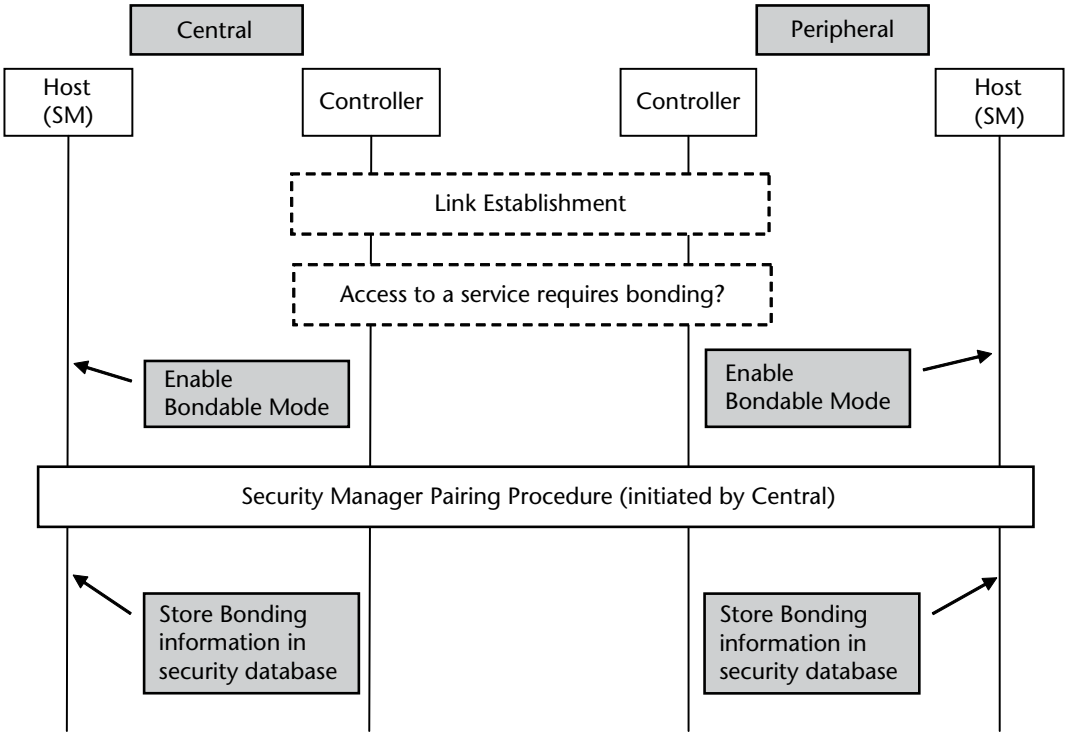

Figure 14.19 Bonding procedure.

Once the LE connection is done, the Security Manager of each device will specify the security mode to be used for further transactions. A device may impose security requirements either at a device level or at a service level. In general, if an application requires security, it will specify the requirements to the Security Manager so that the Security Manager can enforce the correct level of security.

The LE specification defines two modes and four procedures with respect to security.

The two security modes are shown in Table 14.9.

The four security procedures are shown in Table 14.10.

14.7.1 LE Security Mode 1

LE Security Mode 1 has three security levels with increasing security levels:

1. Level 1: No security (No authentication and no encryption).
2. Level 2: Unauthenticated pairing with encryption.
3. Level 3: Authenticated pairing with encryption
4. Level 4: Authenticated LE Secure Connections pairing with encryption.

A connection with higher level of security also satisfies the security requirements for lower levels. So a connection in Level 2 satisfies the requirements for Level 1 and a connection in Level 3 satisfies the requirements for Level 2 and Level 1. Similarly, a connection in Level 4 satisfies the requirements of Level 1, Level 2,

Table 14.9 Security Modes (All Modes are Excluded for Broadcaster and Observer)

S. No.	*Mode*	*Mandatory/Optional*
1	LE Security Mode 1	Optional for Peripheral and Central
2	LE Security Mode 2	Optional for Peripheral and Central

Table 14.10 Security Procedures (All Procedures are Excluded for Broadcaster and Observer)

S. No.	*Mode*	*Mandatory/Optional*
1	Authentication Procedure	Optional for Peripheral and Central
2	Authorization Procedure	Optional for Peripheral and Central
3	Connection Data Signing	Optional for Peripheral and Central
4	Authenticate Signed Data Procedure	Optional for Peripheral and Central

and Level 3. Besides, this connection in Level 3 or Level 4 also satisfies the requirements of LE security mode 2 (as explained in the next section).

14.7.2 LE Security Mode 2

LE Security Mode 2 is used for connection based data signing. It has two security levels:

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

Data signing is not used when a connection is operating in Level 2, Level 3, or Level 4 of security mode 1. Data signing is explained later in this chapter.

14.7.3 Secure Connections-Only Mode

A device can configure itself to be in secure connections-only mode. In this mode, it permits only LE security mode Level 4 (authenticated LE secure connections pairing with encryption) or services that require security mode 1 Level 1 (no security).

14.7.4 Authentication Procedure

The authentication procedure is initiated after a connection has been established. It covers LE security mode 1.

Authentication is achieved by enabling encryption and the security of that encryption depends on the type of pairing performed. There are two types of pairing:

1. Authenticated pairing: This requires pairing performed with authentication set to “MITM protection”.
2. Unauthenticated pairing: This requires pairing performed with authentication set to “No MITM protection”.

These were explained in Chapter 11.

The authentication procedure describes the step to be performed in two scenarios:

1. Receiving a service request: Each service on the server side has security settings associated with it. Before replying to a service request, the server checks to see if the current level of security is sufficient to allow access to that particular service. If it is not, the server replies with the appropriate error code.
2. Initiating a service request: While initiating a service request, the client compares the level of security required to access a service and the current level of security. If the current level of security is insufficient, it initiates the appropriate procedures like pairing or encryption.

14.7.5 Authorization Procedure

Authorization procedure involves getting a confirmation from the user to continue with a security procedure. This may be done after a successful authentication.

14.7.6 Encryption Procedure

A connection may be encrypted by either the Central (Master) or the Peripheral (Slave). This involves invoking the appropriate Security Manager procedures from the Master or Slave side:

1. Initiation from the Slave side: The Slave device may initiate security by sending a Security Request command to the Master. This was explained in Chapter 11.
2. Initiation from the Master side: The Master device may initiate the setup of encrypted session. This was explained in Chapter 11.

In order to enhance the security, specifications 4.2 require that if encryption fails and any of the devices is using a resolvable private address, then the device should immediately discard that address and generate a new address.

14.7.7 Data Signing

Data signing is used in LE Security Mode 2 when there is a requirement to send authenticated data between two devices on an unencrypted connection. The data in the data PDU is signed and the signature is appended to the data PDU itself using the Connection Data Signing Procedure. On the peer device, the signature is verified using the Authenticate Signed Data procedure. This is shown in Figure 14.20.

The data signing method is useful when there is a requirement for fast connection setup and data transfers. This is because the whole data does not need to be encrypted, a process which would otherwise take considerable time and battery power.

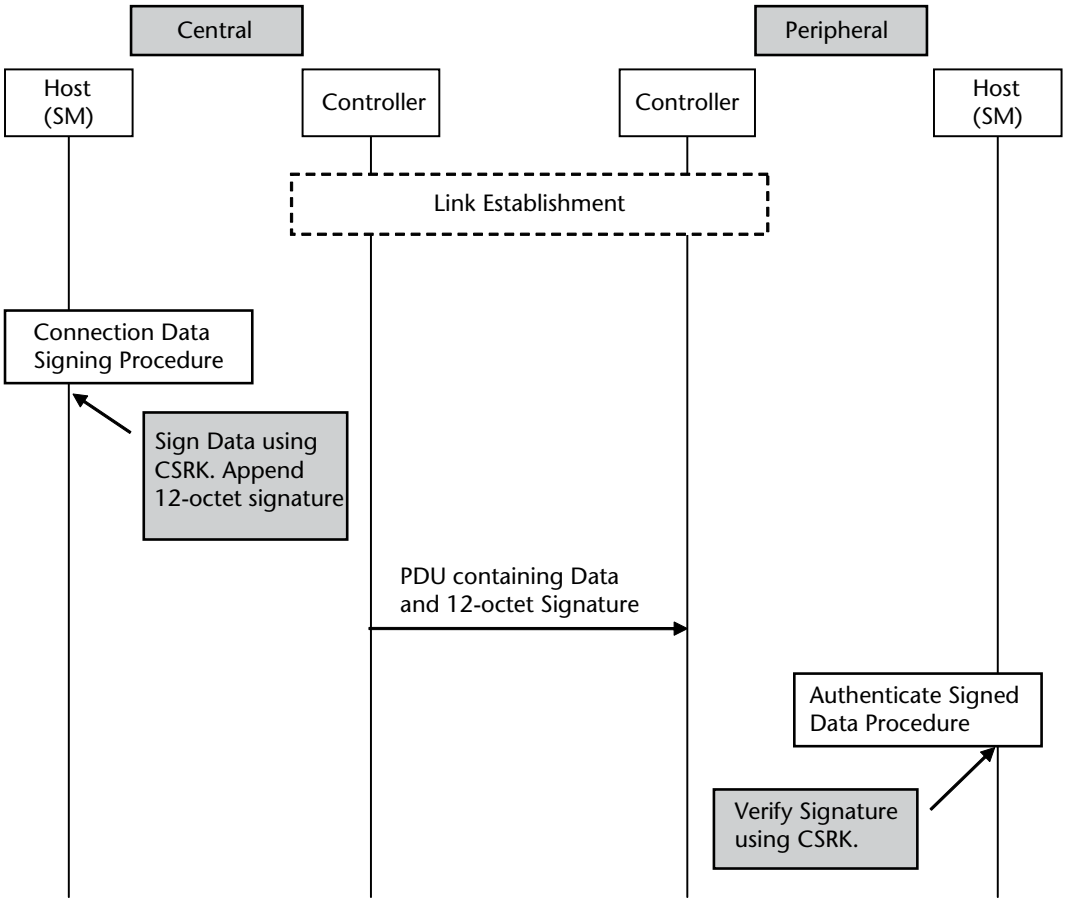

Figure 14.20 Data signing.

14.7.8 Privacy Feature

Since many of the LE devices are supposed to be carried by people (for example, shoes, watch, heart rate sensor), tracking those devices would allow tracking a person by tracking the transmissions from these devices. This could compromise the privacy of a person.

The privacy feature is used to prevent tracking of devices over a period of time. This is done by changing the Bluetooth device address frequently. It is optional to implement this feature in LE devices.

If a Peripheral device supports the privacy feature, then it exposes two characteristics:

1. Peripheral Privacy Flag (Mandatory to support privacy): Privacy is enabled when this flag is set to 1.
2. Reconnection Address Characteristic (Optional): If this characteristic is present, then it can be used in the directed connectable mode to include the address of the Central device in the directed connectable advertising packets (ADV_DIRECT_IND). It may also allow only packets from the device whose address matches the reconnection address. This address can be

stored in the White List to reduce power consumption by filtering packets from other devices.

When a Central device connects to a Peripheral which supports privacy and exposes the Reconnection Address Characteristic, then it writes a new reconnection address in this characteristic every time it makes a connection to it. It may write the same reconnection address to its own White List to allow packets from only this device the next time.

As explained in Chapter 8, specifications 4.2 added the support for resolving lists. The Host may provide a resolving list to the Controller. The Controller may do the address resolution on its own before informing the Host.

14.7.9 Random Device Address

The Random Address was explained in Chapter 8. It is a privacy feature of LE where the device can hide its real address and use a random address which can change over time. So the real address is not revealed at any time. This helps to ensure that a device cannot be tracked.

The Generic Access Profile defines the random address to be of two types:

1. Static Address: A device may choose to initialize its static address to a new value after each power cycle but cannot change it while it is still powered. If the device changes its static address, the peer device will not be able to connect to it with the old address that they may have stored.
2. Private Address: The private address may further be of following two types:
   a. Non-resolvable private address: The peer device can never discover the real address.
   b. Resolvable private address: The peer device can derive the real address using the random address and the link key of the connection.

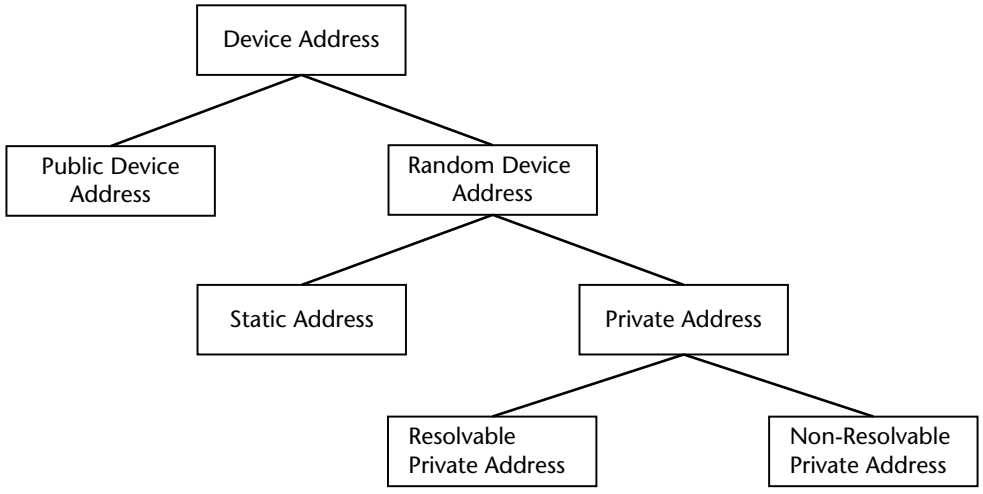

Figure 14.21 Types of Bluetooth device addresses for LE devices.

The different types of device addresses for LE devices are shown in Figure 14.21.

14.8 Summary

The Generic Access Profile defines the modes that a device can be in as well as the generic procedures related to discovering devices, discovering the names of devices, connecting to devices, and security.

GAP tries to maintain cohesiveness in terms of naming conventions and modes as far as possible between BR/EDR and LE. This helps to ensure a uniform user experience when the user is connecting to either an LE device or a BR/EDR device. Of course the internal procedures behind the scenes are different between BR/EDR and LE. So even though the procedure to fetch the name from the remote device is different for BR/EDR and LE, GAP defines the same term "Bluetooth Device Name" to be used at the UI level.

This chapter completed the explanation of the LE protocol stack and mandatory profiles. The next chapter will focus on GATT-based profiles which use the services provided by GAP and GATT to support the use cases that were discussed in Chapter 1.

Bibliography

Bluetooth Core Specification 4.0 http://www.bluetooth.org.
Bluetooth Assigned Numbers, https://www.bluetooth.org/assigned-numbers.

CHAPTER 15

GATT-Based Profiles

15.1 Introduction

The LE Architecture was introduced in Chapter 6. Though it looks similar to the BR/EDR architecture, the LE profiles are much simpler. A major reason for this is that LE introduced the concept of GATT-based profiles. Most of the common functionality needed by all the LE profiles is moved into the ATT protocol and GATT profile. The profiles on top of GATT use the services provided by GATT and only need to implement the minimum items needed to support that specific function. The simplicity of the GATT-based profiles and services can also be seen in the brevity of some of the specification documents (Some are only ten pages and several of the documents are around twenty pages.)

As an example, the battery profile just defines the characteristics that are exposed by a device to provide information about the battery. The generic procedures to access those characteristics are defined by the ATT protocol and GATT profile. This makes the battery profile very simple.

This chapter introduces some of the commonly used GATT-based services and profiles. The location of GATT-based profiles in the LE Protocol Stack is shown in Figure 15.1.

The Bluetooth SIG has defined a versatile list of GATT-based services and profiles to address different scenarios where LE devices can be used. There are currently more than 20 profiles already defined.

One of the major enhancements done after introduction of specifications 4.0 was towards IoT space by enabling the support of IPv6 for Bluetooth. This was done by the IPSP, which will be covered later in this chapter.

15.2 Profile, Services, and Characteristics

The GATT-based profile architecture was explained in detail in Chapter 13. LE introduces the concept of services and profiles that are defined independently of each other in separate specifications.

A service can be considered to be a data structure used to describe a particular function or feature. It is a collection of characteristics and describes what a device

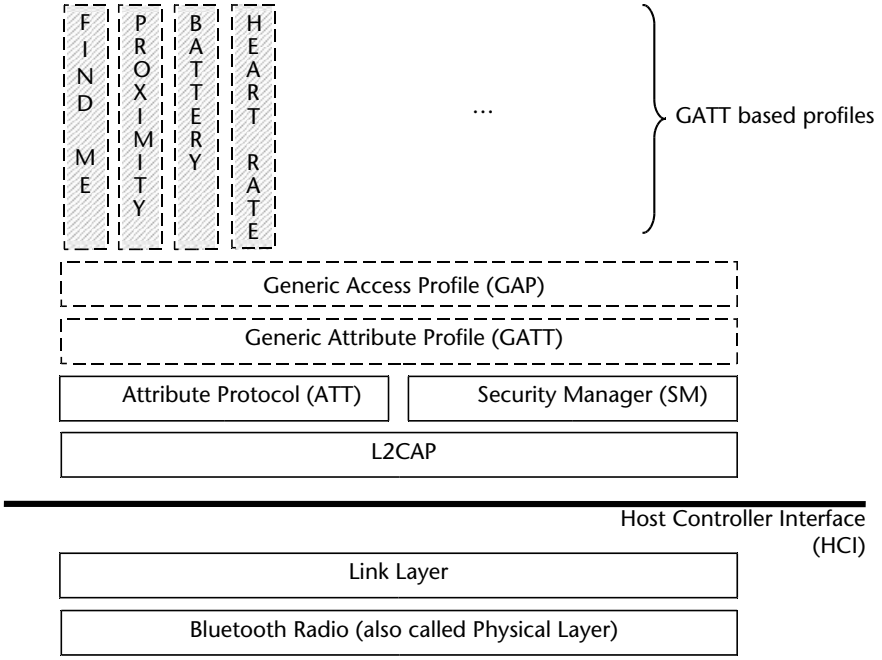

Figure 15.1 GATT-based profiles in LE protocol stack.

does. All services are implemented by a server and accessed by one or more remote devices (acting as clients).

A profile can include one or more services. It is possible for several profiles to use the same service. For example the Device Information Service may be used by several profiles to provide useful information about the device (like manufacturer name, model number, etc.). Figure 15.2 shows the relationship between Profiles, Services and Characteristics.

- A Device may support one or more profiles.
  - A Profile may support zero or more services. (It may not contain any services as well for some of the profile roles.)
    - Each service may contain one or more characteristics.
- The characteristics are the data values that can be read, written, indicated or notified.

The most important thing to note here is that this framework allows a remote device to read or write data or register notifications or indications for that data. This data is contained in the characteristics. The LE profiles are broadly focused on providing access to some particular data. Each profile finally exposes certain data in the form of characteristics which can be accessed by remote devices. This data could be:

- The temperature that the LE temperature sensor wants to report.
- The alert level that has been set so that the device can be alerted on some specific condition.

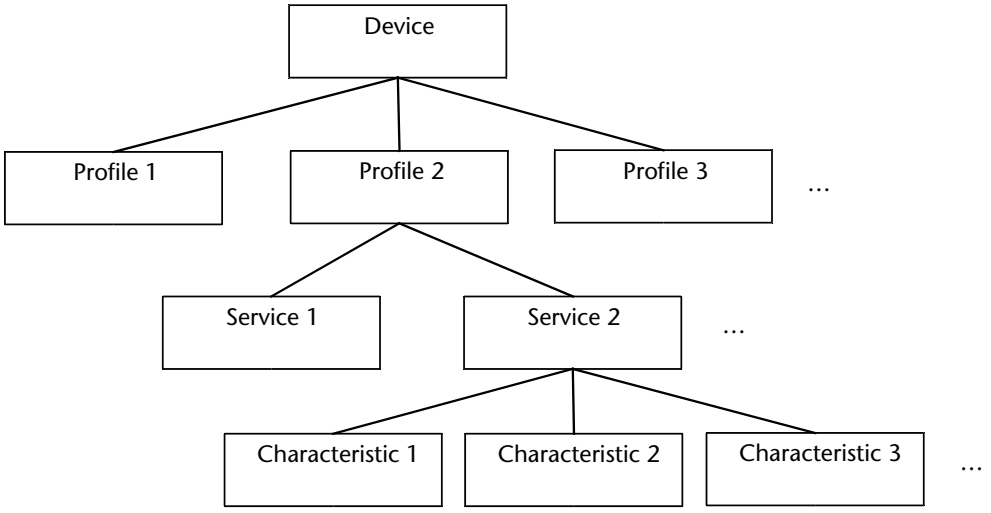

Figure 15.2 Profiles, services, and characteristics.

- The heartbeat data.
- The status of a battery.

Note that this is in contrast to BR/EDR profiles. For example the Handsfree or A2DP profiles cannot really be considered to be data centric. They enable voice and audio to be transferred on top of a Bluetooth connection.

15.3 Immediate Alert Service (IAS)

The Immediate Alert Service is a very simple service which allows a remote device to write an alert into it. When the remote device writes an alert into it, the device may take some specific action like flashing an LED, sounding a buzzer, etc.

One example might be finding a misplaced key fob. The user may press a button on the mobile phone which would lead to an alert being generated by the key fob. The key fob could, for example, start buzzing. This could be used to locate the key fob.

15.3.1 Service Declaration

The Link Loss Service is instantiated as a <<Primary Service>>. The Service UUID is set to <<Immediate Alert>>.

15.3.2 Service Characteristics

The Immediate Alert Service exposes only one characteristic which is mandatory.

1. Alert Level (Write Without Response): The remote devices can write the alert level using the GATT Write Without Response procedure.

The Alert Level is a control point characteristic. (Note from Chapter 12 that control point characteristics are the ones which cannot be read. These attributes can only be written, notified, or indicated).

The remote device can set the Alert Level to one of the following:

1. No Alert: The device does not alert.
2. Mild Alert: The device alerts.
3. High Alert: The device alerts in the strongest possible way.

On writing a Mild Alert or High Alert, the device continues to alert until one of the following conditions occur:

1. An implementation specific timeout occurs
2. User takes some action on the device like pressing a button to acknowledge and stop the alert.
3. A new alert level is written.
4. The physical link is disconnected.

The Immediate Alert Service is shown in Figure 15.3.

In summary, from a use case scenario perspective, this service can be used to set an alert in the remote LE device so that the LE device can take appropriate action.

15.4 Find Me Profile (FMP)

The Find Me Profile supports the function to allow users to find misplaced devices. A button is pressed on a device which causes an alert to be raised on a peer device.

15.4.1 Roles

Find Me Profile defines the following two roles:

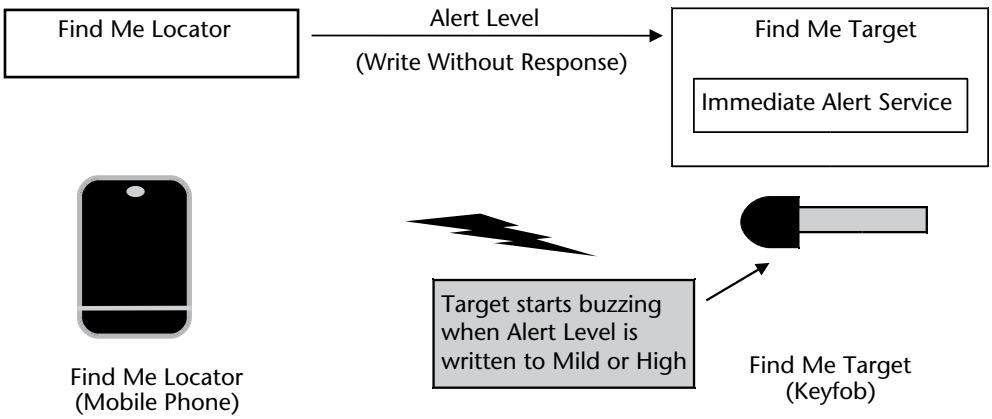

Figure 15.3 Immediate alert service and find me profile roles.

1. Find Me Locator (GATT Client): This is the device on which the button is pressed. When a button is pressed, it writes the specific Alert Level into the Alert Level characteristic of the Find Me Target.
2. Find Me Target (GATT Server): This is the device on which an alert is raised.

The Find Me Locator uses the GATT service discovery procedures to discover the Immediate Alert Service on the Target.

The profile does not impose any restrictions on which of the two devices should act as GAP Central or Peripheral. Either of the devices can act in the GAP Central role and the peer device acts in the GAP Peripheral role. The Central device does the discovery and connection establishment with the Peripheral device. The Find Me Profile roles are shown in Figure 15.3.

15.5 Link Loss Service (LLS)

The Link Loss Service is a very simple service which allows an alert to be raised when the connection to a remote device is lost. When the link is lost, the device may take some specific action like flashing an LED, sounding a buzzer, locking the device, etc. This is useful when an alert is needed if a device moves out of range or comes into range.

One example of this service is when a user may use it between a watch and a mobile phone. If the user forgets a mobile phone and walks away (or if the mobile phone is stolen), the watch would raise an alert.

Another example could be a playground or a shopping mall where a child and parent may wear a watch (or any other device) supporting this service. If the child moves out of the range, then the connection would be lost and the parent would get an alert on their watch.

15.5.1 Service Declaration

The Link Loss Service is instantiated as a <<Primary Service>>. The Service UUID is set to <<Link Loss>>.

15.5.2 Service Characteristics

The Link Loss Service exposes only one characteristic which is mandatory:

1. Alert Level (Read, Write): The remote devices can read or write the alert level using the GATT Read Characteristic Value and Write Characteristic Value procedures.

The Alert Level can be set to one of the following:

1. No Alert: The device does not alert.
2. Mild Alert: The device alerts on loss of a link.
3. High Alert: The device alerts in the strongest possible way on loss of a link.

When the service is implemented in a device and the connection is lost, the device starts alerting at the specified alert level. The alert continues till one of the following occurs:

1. An implementation specific timeout occurs.
2. The user takes some action on the device like pressing a button to acknowledge and stop the alert.
3. The physical link is reconnected.

The Link Loss Service is shown in Figure 15.4.

15.6 Transmit Power Service (TPS)

The Transmit Power Service is a very simple service which exposes the device's transmit power level when the device is in connected mode.

One example of this service is when the user wishes to know the distance between two devices. The greater the distance, the greater would be the transmit power level. So depending on the transmit power level, the user can correlate the distance between the two devices.

Another example of this service could be a connection between the user's watch and a computer. As soon as the user comes near the computer, the computer automatically unlocks itself and when the user moves away from the computer, the computer locks itself. This can be done based on the transmit power level between the watch and the computer to calculate the distance between the two.

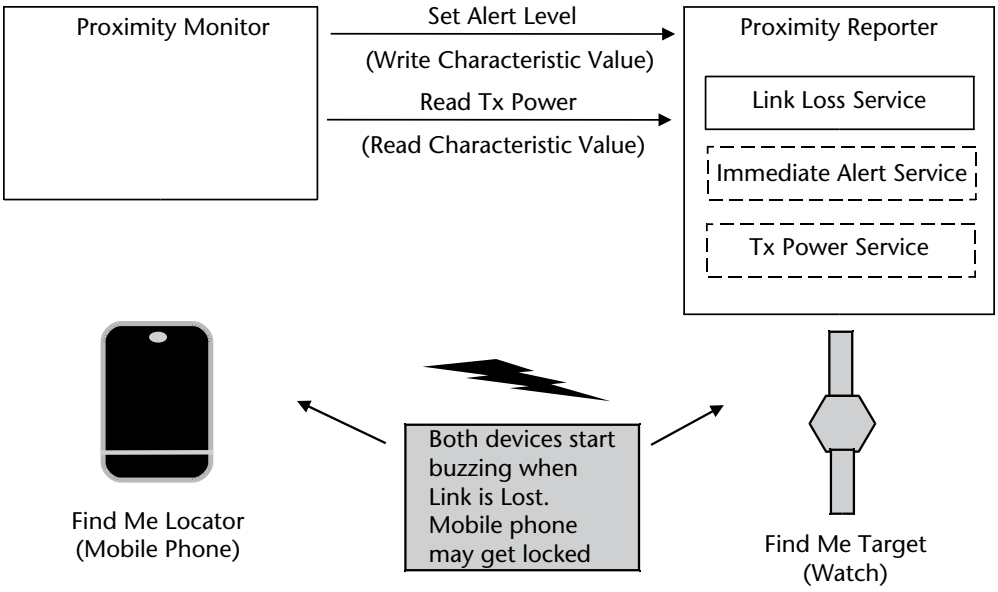

Figure 15.4 Immediate alert service, link loss service, Tx power service, and proximity profile roles.

15.6.1 Service Declaration

The Transmit Power Service is instantiated as a <<Primary Service>>. The Service UUID is set to <<Tx Power>>.

15.6.2 Service Characteristics

The Transmit Power Service exposes only one characteristic which is mandatory:

1. Tx Power Level (Read): The remote devices can read the Tx Power Level using the GATT Read Characteristic Value procedure.

The Tx Power Service is shown in Figure 15.4.

15.7 Proximity Profile (PXP)

The Proximity Profile supports the use of monitoring the proximity (distance) between two devices. It supports the following scenarios:

1. If a device moves far away and the connection drops, then an alert is generated.
2. If a device moves far away and the path loss increases above a certain value, then an alert is generated.

15.7.1 Roles

Proximity Profile defines the following two roles:

1. Proximity Monitor (GATT Client): The Monitor can read the Tx Power of the remote device and set the alert levels for immediate alert and link loss using GATT procedures.
2. Proximity Reporter (GATT Server): The Reporter exposes the characteristics of Immediate Alert Service, Link Loss Service and Tx Power Service.

Generally the Proximity Reporter is an LE-only device which acts as a GAP Peripheral. The Proximity Monitor could be a dual-mode or LE-only device which acts as GAP Central. The Proximity Profile roles are shown in Figure 15.4. The services shown in dashed boxes are optional while the services shown in solid boxes are mandatory.

15.8 Battery Service (BAS)

The Battery Service is a very simple service to provide information about the battery to remote devices. The remote devices may either read the battery level or be notified when the battery level has changed. One example of this service would be to get a notification on the mobile phone if the battery of a thermometer is about to run down.

15.8.1 Service Declaration

The Service UUID is set to <<Battery Service>>.

15.8.2 Service Characteristics

The Battery Service exposes only one characteristic which is mandatory:

1. Battery Level (Read, Notify): The remote devices can read the level using the GATT Read Characteristic Value or be notified when the battery level changes. The notification property is optional.

The Battery Level is denoted as a percentage from 0% to 100%. 0% represents a battery that is fully discharged and 100% denotes a battery that is fully charged. In order to save battery power, instead of polling the battery level periodically, the client can also configure the server to send a notification when the battery level changes.

15.9 Device Information Service (DIS)

The Device Information Service is used to provide manufacturer information about a device. For example, this service can be used to provide the manufacturer name, model number, serial number, etc.

15.9.1 Service Declaration

The Device Information Service is instantiated as a <<Primary Service>>. The Service UUID is set to <<Device Information >>.

15.9.2 Service Characteristics

The Device Information Service exposes the following characteristics. It is mandatory to support any one of these characteristics.

1. Manufacturer Name String: The name of the manufacturer of the device.
2. Model Number String: Model Number.
3. Serial Number String: Serial Number of the particular device.
4. Hardware Revision String: Revision of hardware in the device.
5. Firmware Revision String: Revision of firmware in the device.
6. Software Revision String: Revision of software in the device.
7. System ID: This represents a structure that contains an Organizationally Unique Identifier (OUI) followed by a manufacturer-defined identifier.
8. IEEE 11073-20601 Regulatory Certification Data List: Regulatory and Certification information about the product. The IEEE 11073-20601 specification defines a common framework for making an abstract model of personal health data so that this data can be exchanged and interpreted between health devices and computer systems.

All these characteristics are read only and can be read using the GATT Characteristic Value Read procedure.

15.10 Current Time Service (CTS)

The Current Time Service is used to provide the current time to remote devices. One example of this service would be to update the time in a watch automatically when a user travels to a new time zone or when daylight savings time changes. The watch could be connected to a mobile phone and be notified whenever the user travels to a new time zone.

15.10.1 Service Declaration

The Current Time Service is instantiated as a <<Primary Service>>. The Service UUID is set to <<Current Time Service>>.

15.10.2 Service Characteristics

The Current Time Service exposes the following three characteristics. Out of these, the first characteristic is mandatory and the remaining two are optional.

1. Current Time (Read, Notify): This characteristic provides the current data and time of the server device. This characteristic can be read or can be configured to be notified.
2. Local Time Information (Read): This characteristic provides the information like time zone and day light saving offset.
3. Reference Time Information (Read): This characteristic provides information about the reference time source from which the time was obtained. For example, this provides information on how accurate the time source is.

15.11 Health Thermometer Service (HTS)

The Health Thermometer Service is used to provide information from the thermometer like temperature, temperature type, etc. This service can be used in healthcare and fitness applications.

15.11.1 Service Declaration

The Device Information Service is instantiated as a <<Primary Service>>. The Service UUID is set to <<Health Thermometer Service >>.

15.11.2 Service Characteristics

The Health Thermometer Service exposes the following characteristics. Some of the characteristics also expose a corresponding Client Characteristic Configuration

Descriptor or a Valid Range Descriptor. Out of these the first service (Temperature Measurement) along with the corresponding Client Characteristic Configuration Descriptor are mandatory.

1. Temperature Measurement (Indicate): Used to indicate a temperature measurement to the peer device. The least significant bit is used to indicate whether the temperature is in Celsius or Fahrenheit.
   a. Client Characteristic Configuration Descriptor: Used to configure the Temperature Measurement Characteristic (for example, to enable indication).
2. Temperature Type (Read): This is used to specify the location of the human body where the temperature is measured.
3. Intermediate Temperature (Notify): This is used to send intermediate values to a device while the temperature measurement is still in progress. The interval at which this is sent could typically vary from 0.25 seconds to 2 seconds.
   a. Client Characteristic Configuration Descriptor: Used to configure the Intermediate Temperature Characteristic.
4. Measurement Interval (Read): This is used to set the interval between two successive measurements.
   a. Client Characteristic Configuration Descriptor: Used to configure the Measurement Interval Characteristic.
   b. Valid Range Descriptor: This provides the supported range of measurement interval values.

This service is shown in Figure 15.5.

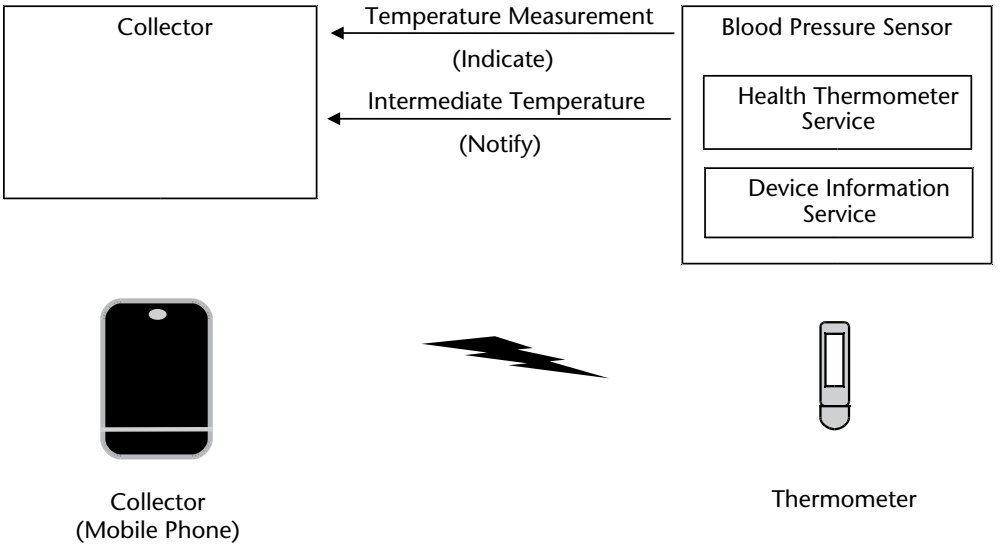

Figure 15.5 Health thermometer service and health thermometer profile.

15.12 Health Thermometer Profile (HTP)

The Health Thermometer Profile allows a device to interact with a thermometer sensor that exposes the Health Thermometer Service.

15.12.1 Roles

The Health Thermometer Profile defines the following two roles:

1. Thermometer (GATT Server): The thermometer is the device that performs the temperature measurement and informs the collector.
2. Collector (GATT Client): The collector is the device that receives the data from the thermometer.

The Thermometer implements the GAP Peripheral role. The collector implements the GAP Central role. The Thermometer Profile roles are shown in Figure 15.5. The Thermometer role includes two services—Health Thermometer Service and Device Information Service. Both the services are mandatory.

15.13 Blood Pressure Service (BPS)

The Blood Pressure Service is used to provide information about the blood pressure and other related information. This service can be used in healthcare applications.

15.13.1 Service Declaration

The Service UUID is set to <<Blood Pressure Service >>.

15.13.2 Service Characteristics

The Blood Pressure Service exposes the following characteristics. Some of the characteristics also expose a corresponding Client Characteristic Configuration Descriptor. Out of these the first service (Blood Pressure Measurement) along with the corresponding Client Characteristic Configuration Descriptor and the Blood Pressure Feature are mandatory.

1. Blood Pressure Measurement (Indicate): Used to indicate a blood pressure measurement to the peer device. It may have additional fields like the pulse rate, Systolic and Diastolic pressure, etc.
   a. Client Characteristic Configuration Descriptor: Used to configure the Blood Pressure Measurement Characteristic.
2. Intermediate Cuff Pressure (Notify): This is used to send the intermediate values while the measurement is still in progress.
   a. Client Characteristic Configuration Descriptor: Used to configure the Intermediate Cuff Pressure Characteristic.
3. Blood Pressure Feature (Read): This is used to describe the supported features of the Blood Pressure Sensor.

The Blood Pressure Service is shown in Figure 15.6.

15.14 Blood Pressure Profile (BLP)

The Blood Pressure Profile allows a device to interact with a blood pressure sensor that exposes the Blood Pressure Service.

15.14.1 Roles

The Blood Pressure Profile defines the following two roles:

1. Blood Pressure Sensor (GATT Server): The blood pressure sensor is the device that performs the blood pressure measurement and informs the collector.
2. Collector (GATT Client): The collector is the device that receives the data from the blood pressure sensor.

The Blood Pressure Sensor implements the GAP Peripheral role. The collector implements the GAP Central role. The Blood Pressure Profile roles are shown in Figure 15.6. The Blood Pressure Sensor role includes two services—Blood Pressure Service and Device Information Service. Both the services are mandatory.

15.15 Health, Sports and Fitness Profiles

Similar to the services and profiles described above, LE includes several other profiles for health, sports and fitness applications. These include the following:

1. Glucose Service (GLS): Exposes the glucose and other data from a glucose sensor.

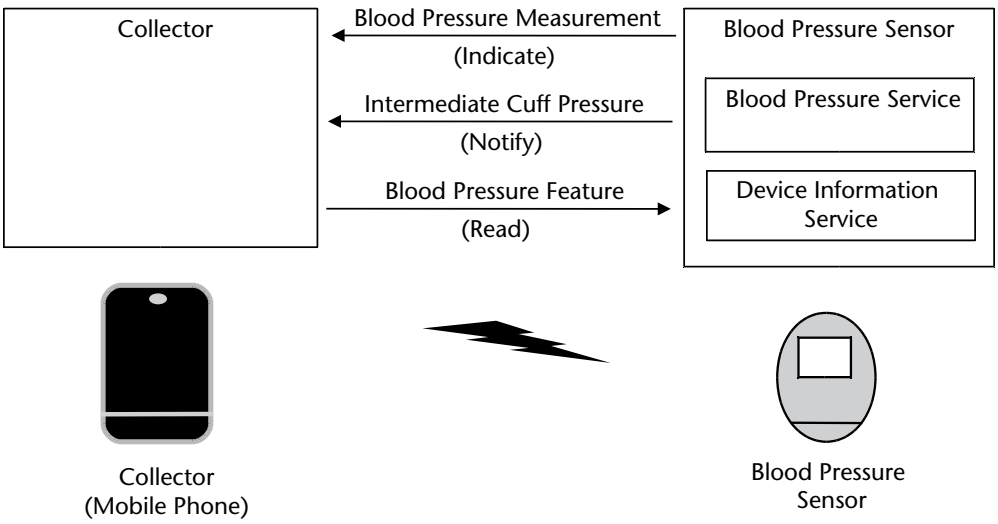

Figure 15.6 Blood pressure service and blood pressure profile.

2. Glucose Profile (GLP): Allows a device to interact with a glucose sensor device.
3. Heart Rate Service (HRS): Exposes the heart rate and other data from a heart rate sensor.
4. Heart Rate Profile (HRP): Allows a collector to interact with a heart rate sensor device.
5. Cycling Speed and Cadence Service (Service): Exposes the cycling speed and other information to be used in sports and fitness applications.
6. Cycling Speed and Cadence Profile (CSCP): Allows a collector to interact with a cycling speed and cadence sensor.

15.16 Internet Protocol Support Profile (IPSP)

One of the major enhancements brought in by specifications 4.1 is the enhanced support for the Internet of Things. This was done by introducing IPSP while specifications 4.2 were being ratified. IPSP requires compliance to specifications 4.1 or higher.

IPSP provides the support for exchanging IPv6 packets between two devices over BLE transport. This means that a BLE sensor can exchange packets with the Internet through a router-like home gateway or mobile phone. The router does not need to do too much packet processing or conversion since the packets are already IPv6 and can be routed to or from the internet easily.

The complete details of how IPv6 communication happens over BLE transport are specified by RFC 7668 (IPv6 over Bluetooth(R) low energy). This profile also uses RFC4861 (neighbor discovery for IP version 6 (IPv6)) and RFC6775 (neighbor discovery optimizations for 6LoWPAN).

What is IPv6?

Each device on the Internet is assigned a unique address called the IP address. This address is used to identify the device and to send packets to that device. Historically, the IP addresses have been 4 bytes (i.e., 32-bits) long. An example of an IP address is 192.168.3.55.

The digits denote the four bytes of the IP address in decimal form (instead of hexadecimal form for ease of use). This is called IPv4 or Internet protocol version 4 address format.

Theoretically this can address 2^{32} or approximately 4.3 billion devices. The real number is much smaller because some of these addresses have specific meanings and are not assigned to devices, while some other addresses may have been allocated to certain organizations but not really assigned to devices. With the rapid growth of Internet and the connected devices, these addresses started getting exhausted rapidly.

IPv6 addresses are 128 bits. This can theoretically address 2^{128} devices or approximately 3.4×10^{38} devices (or three hundred and forty trillion-trillion) unique IP addresses. A typical IPv6 address looks like 2001:0DB8:D3F0:0123:4567:89AB.

This is very useful in the exponentially growing Internet of Things space where each and every device or sensor may need a way to uniquely identify itself, which could be an IPv6 address.

Besides adding the support for more addresses, IPv6 introduced several other networking advantages and is widely used in IoT space, amongst other things.

What is 6LoWPAN?

6LoWPAN is an acronym for IPv6 over low-power wireless personal area networks (WPANs). As the name suggests, this finds applications in scenarios where internet connectivity is required by low-power radio communication devices.

6LoWPAN uses mechanisms like header compression and fragmentation in order to support the WPAN devices. It was originally defined for IEEE 802.15.4-based networks, though it can be used for other networks, such as BLE.

IPSP defines two roles: the Node (this role is used for devices that can produce or consume IPv6 packets) and the Router (this role is used for devices that can route IPv6 packets).

A typical IPSP configuration is depicted in Figure 15.7, which shows a blood pressure sensor acting as a Node and sending the data to the Internet via a Smartphone that is acting as a router. The blood pressure sensor sends IPv6 packets to the Smartphone. The Smartphone routes these packets to the internet. Since the packets are already IP packets, the Smartphone has to perform minimum processing while routing the packets.

6LoWPAN defines two terms:

- 6LN: This refers to a 6LoWPAN node entity. This is also referred to as node by IPSP.
- 6LBR: This refers to a 6LoWPAN border router. This is also referred to as router by IPSP.

Both peripheral and central devices can act as either 6LN or 6LBR, though it's more likely that a peripheral would acts as a 6LN and a central would acts as a 6LBR.

The complete protocol stack architecture for 6LoWPAN and BLE to support IPSP is shown in Figure 15.8. The main points to note are:

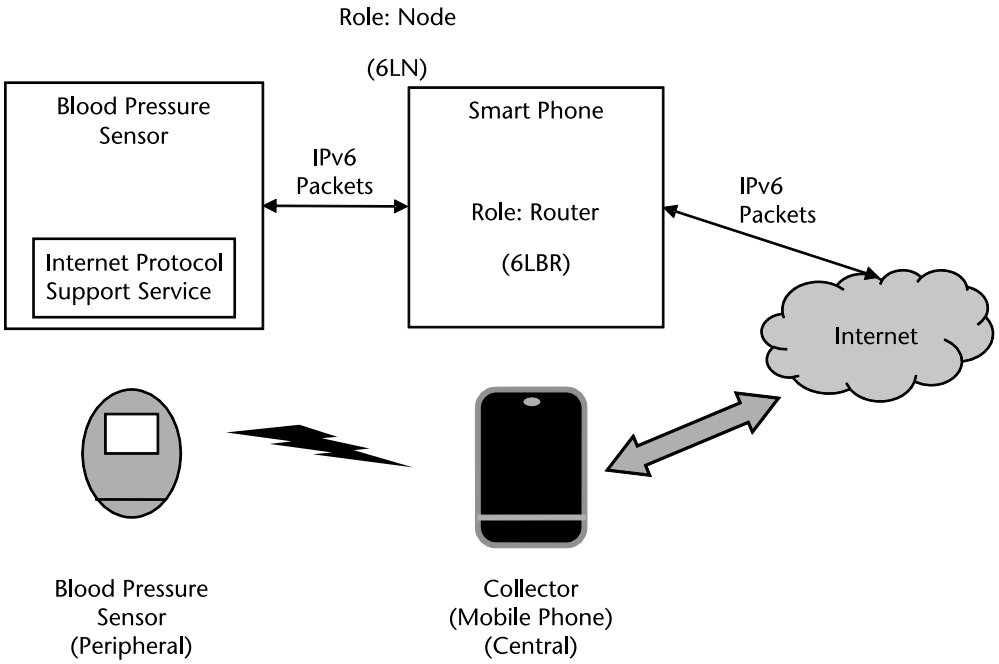

Figure 15.7 IPSP typical configuration.

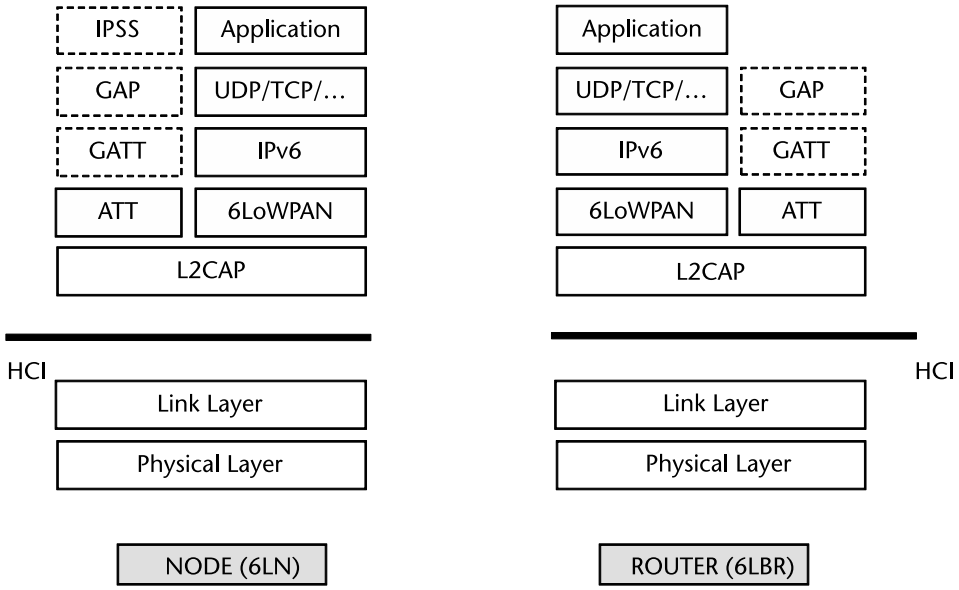

Figure 15.8 6LoWPAN and BLE protocol stack.

1. The node runs an instance of Internet protocol support service (IPSS) that allows the router device to discover it using GATT procedures.
2. The application can be any application that is capable of generating or consuming data that is transported over IPv6 packets.
3. The initial connection establishment happens over the link layer and then L2CAP layer.

4. Application, UDP/TCP, IPv6, and 6LoWPAN represent the various layers of the protocol stack that generate the IPv6 packet and provide it to L2CAP for transmission to the remote entity.

15.16.1 Service Declaration

The IPSS runs on the device acting in the node role.

The service UUID is set to <<Internet Protocol Support Service>>.

This service does not define any characteristics.

15.16.2 Configuration

This profile uses the LE connection-oriented channels feature with LE credit-based flow control.

An MTU size of 1,280 or higher is required at L2CAP level. This is in line with IPv6 requirements, which require the host to to be capable of processing datagrams of at least 1,280 bytes.

The 128-bit IPv6 address is formed based on the 48-bit BD_ADDR, using the private address of the device. The remaining bits are stuffed with predefined values.

One of the typical configurations of a BLE network connected to the internet is shown in Figure 15.9. The blood pressure sensor, thermometer, and watch act as nodes (6LN) and the Smartphone acts as a router (6LBR). The 6LNs cannot talk to each other directly, rather, each of the 6LNs are connected to the 6LBR in star

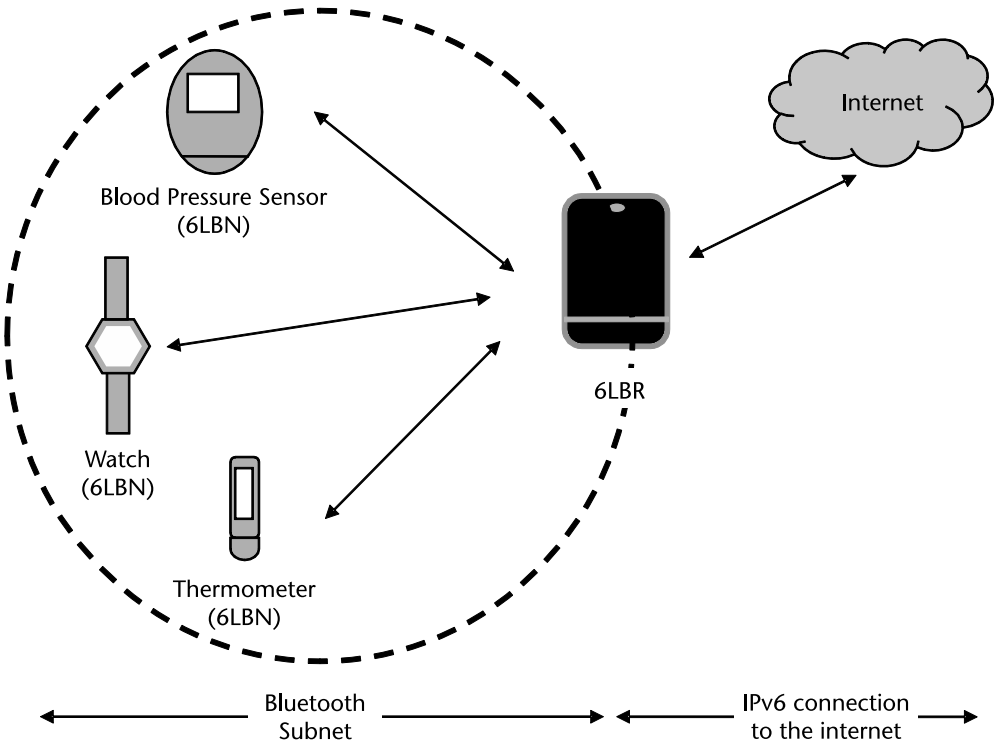

Figure 15.9 Typical configuration of a BLE network connected to the Internet.

topology. The 6LBR is, in turn, connected to the Internet and provides Internet connectivity to the 6LNs.

It's not mandatory that the 6LBR be connected to the Internet. The 6LN and 6LBR may form a subnet and communicate with each other as well. This forms an isolated BLE network. For example, the set of wearable devices that the person is carrying may not require Internet connectivity at all times, but the smart shoes that the person is wearing may need to be connected to the smart watch. This can be done by creating a subnet comprising the smart shoes, smart watch, and the Smartphone.

15.16.3 Profile Stack Requirements

Devices implementing IPSS need the following components: IPSS (Internet profile support service); IPSS, GATT, ATT (for the service discovery); and GAP (for device discovery, connection setup, and security).

15.16.4 Typical IPv6 operations

An IPv6 Host or router can use the IPv6 discovery process to obtain information about its local environment. IPv6 messages comprise the following:

- Router Solicitation: This message is used to request the local routers to transmit information. The information gets transmitted in router advertisement messages. The router advertisement messages contain various link and Internet parameters.
- Router Advertisement: This message is used by the routers to transmit information to the local hosts. Generally, the routers transmit the information periodically.
- Neighbor Solicitation: This message is used to request the local hosts to send information. The hosts send information in the neighbor advertisement message.
- Neighbor Advertisement: This message is used to transmit host information.
- Redirect: This message is used by the routers to notify a host of a better route or destination.

The ICMPv6 information messages echo request (Ping) and echo response can be used to test connectivity with a router or another host. Figures 15.10, 15.11 and 15.12 show a typical sequence of IPv6 messages.

Figure 15.10 shows a neighbor advertisement message that is sent by the host to inform the other devices about its presence.

Figure 15.11 shows a router solicitation message that is sent by the host to request the router to send information about the router.

Figure 15.12 show an echo request (also known as Ping). The response to this message is an echo response.