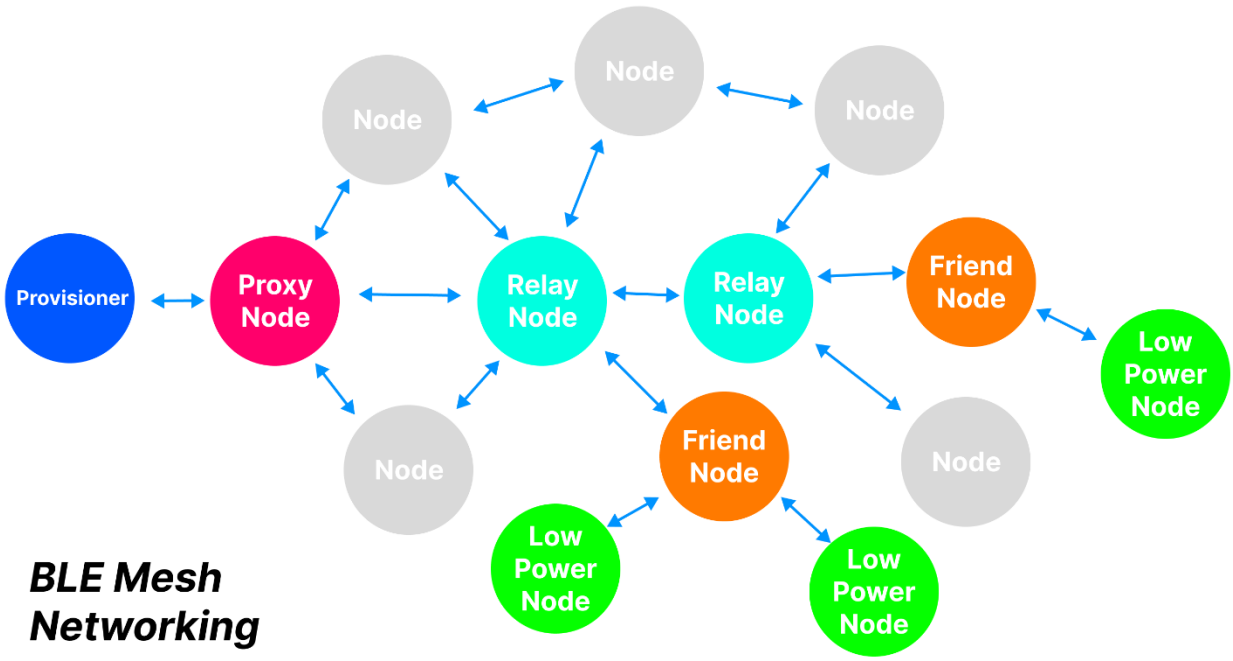
Bluetooth Mesh on Embedded Devices



Created by Sugiarto Wibowo

osugiartow@gmail.com

2024

1. Bluetooth Low Energy (BLE)

BLE is wireless protocol just like bluetooth classic, but the power consumption is relatively low and require less time and effort to pair evices. The drawback is lower speeds connection. BLE operates in the unlicensed Industrial, Scientific, and Medical (ISM) band at 2.4 to 2.485 GHz. It uses spread spectrum, frequency hoping, and full-duplex signal.

BLE Stack consists of **Controller** and **Host**. The **Controller** includes the **physical** (RF Physical) and **Link Layer** (LL). The **Host** consists of **Logical Link Control and Adaptation Protocol** (LP2CAP), the **Security Manager** (SM), the **Attribute protocol** (ATT), **Generic Attribute Protocol** (GATT), and **Generic Access Profile** (GAP). The interface between Controller and Host is called **Host Controller Interface** (HCI).

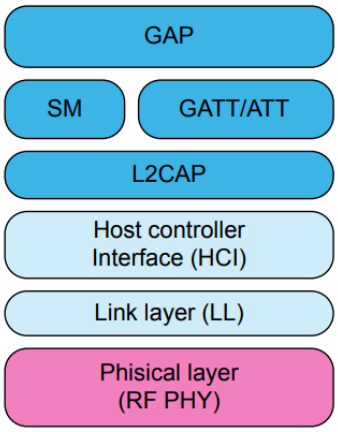


Figure 1 BLE Stack

1.1. Physical Layer

The physical layer is the lowest stack of the BLE. This layer uses **Adaptive Frequency-Hopping** (AFH) **Gaussian Frequency Shift Keying** (GFSK) radio for 1 Mbps and **2-level GFSK** for 2 Mbps. The AFH technology can avoids all frequencies used by other no-adaptive technologies, which allows moving from a bad channel to the next good channel. It has 40 RF channels (0-39) with 2 MHz spacing, there are two channels types:

- **Advertising channels** (channel 37, 38, 39) for advertising channel packets, packets for discoverability, and for broadcasting.
- **Data physical channel** uses other than advertising channels for bidirectional communication between connected devices.

1.2. Link Layer (LL)

The Link Layer defines a state machine which defines how each devices can use a radio to transmit information between devices. There are five states of the LL including:

- **Standby**: The device doesn't transmit or receive packets.

- **Advertising**: The device broadcasts advertisemenets in advertising channels.
- **Scanning**: The device looks for advertiser devices.
- **Initiating**: The device initiates connection to the advertiser device.
- **Connection**: The initiator device is in master role and communicates with the slave role, and defines timings of transmissions.
- **Advertiser device in slave role**: Communicates with the master role or the initiator device.

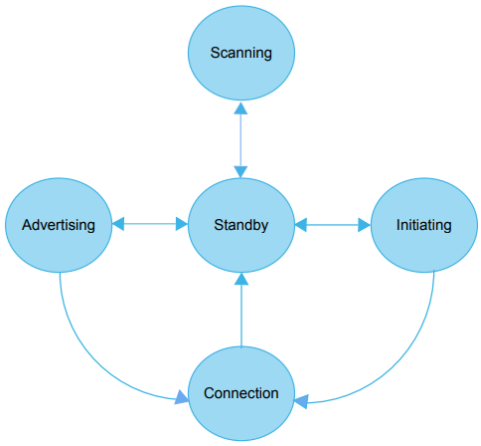


Figure 2 Link Layer state machine

1.2.1. BLE Packets

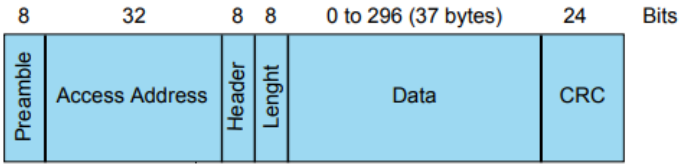


Figure 3 Standard BLE packet

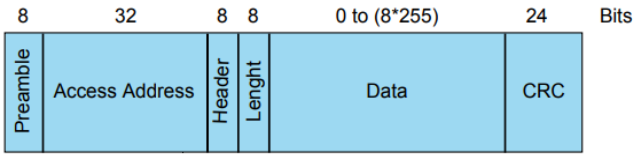


Figure 4 Extended Length BLE packet

A packet is consists of data that transmitted by one device and received by other device. There are two BLE packets structures, the standard and the extended length. The difference of the two packets is the data length only, and the packet structures are including:

- **Preamble:** RF synchronization sequence.

- **Access address:** For identifying the communication packets on physical layer channel.
- **Header:** Depends on packet type (advertising or data).

Advertising Packet Header Content				
Advertising Packet Type (4 bits)	Reserved (2 bits)	Tx Address type (1 bit)	Rx Address type (1 bit)	
Data Packet Header Content				
Link Layer identifier (2 bits)	Next sequence number (1 bit)	Sequence number (1 bit)	More Data (1 bit)	Reserved (3 bits)

Table 1 BLE Packet header contents

1.2.2. Advertising State

On advertising state, the Link Layer can transmits advertising packets and responds to scan requests from scanner. Advertising state can be moved to a standby state by stopping the advertising. While advertising, the three advertising channels (37, 38, 39) sends the same packet in sequence and known as **advertising event.**

Figure 5 Advertising packet with AD type flags

1.2.3. Scanning State

There are two types of scanning, which are **passive** and **active**. On passive scanning the advertisement data is receieved from an advertiser device. While active scanning, the scanner can send back a scan request packet to the advertiser which allows the scanner device get additional information from the advertiser.

1.2.4. Connection State

Connection state is established when the data to be transmitted are more complex. When the initiator device receieves an advertising packet from an advertising device to which it wants to connect, the initiator device can send a connect request packet to the advertiser device. The require informations for the connection establishments are including:

- Access address for identifying communications on physical link.
- CRC initialization value.
- Transmit window size.
- Connection interval
- Slave latency.
- Supervision timeout.
- Channel map 37 bits.
- Frequency-hop value (between 5 and 16).
- Sleep clock accuracy range.

A slave device can only be connected to one master device, but master device can be connected to several BLE slave devices. Bluetooth SIG says there is no limit on the number of slaves a master can connect to, but could be limited by the specific BLE stack.

1.3. Host Controller Interface (HCI)

HCI provides communication between the **Host** and **Controller** through software APIs or hardware interface such as SPI, UART, USB.

1.4. Logical Link Control and Adaptation Layer Protocol (L2CAP)

L2CAP supports higher level protocol multiplexing, packet segmentation, and reassembly operations, and the conveying of quality of service information.

1.5. Attribute Protocol (ATT)

ATT allows the device to expose some data to another device. The device **exposing attributes** is called **server** and the peer device **using the attributes** is called **client**. Attribute components are including:

- **Attribute handle:** Identifies an attribute on a server, allowing the client to reference attribute in read or write requests.
- **Attribute type:** UUID,
- **Attribute value.**
- **Attribute permissions:** Defined by each upper layer that uses the attribute such as security level for read/write access, notification and/or indication, and the undiscoverable permissions. The permissions are including **access, authentication, and authorization**.

In attribute server, a collection of attributes is called a database. Attribute protocol implements the peer-to-peer client-server protocol between an attribute server and an attribute client.

- **Server role**
  - Contains attribute database.

- Receivers requests, executes, responds commands.
- Indicates, notifies an attribute value when data change.

- **Client role**
  - Talks with server.
  - Send requests, waits for response (read), update (write) data.
  - Confirms indications.

1.6. Security Manager (SM)

The BLE Link Layer supports **encryption** and **authentication** by using counter mode with CBC-MAC (Cipher Block Chaining-Message Authentication Code) algorithm and 128-bit AES Block Cipher (AES-CCM). A 4-byte Message Integrity Check (MIC) is appended to the payload of the data channel PDU when encryption and authentication are used. When two devices want to encrypt the communication during the connection, the security manager uses the **pairing procedure** which allows both devices to **exchange their identity information** and create the **security keys for a trusted relationship**. The pairing procedures could prevent attacks from:

- **Mant-in-the-middle (MITM)** attacks: A device is able monitor and modify message to the communication channel between two devices.
- **Passive eavesdropping** attacks: A device listening on communication channel.

The pairing procedures are including **LE Legacy Pairing** for BLE v4.0 or v4.1 and **LE Secure Connection** on BLE v4.2. The LE Legacy Pairing uses and generates 2 keys, **Temporary Key** (TK) and **Short-term Key** (STK). The LE Secure Connections uses **Long-term Key** (LTK). The difference on the LE Secure Connections are using **Eliptical Curve Difffie-Helman** (ECDH) algorithm which allows the keys to be exchaged over an unsecured channel and protect against eavesdropping attacks; **Numeric comparison** method is added.

1.7. Privacy

The privacy feature on the advetising device makes the device untracked by scanners. There are two kinds of private addresses, **Non-resolvable** and **Resolveable.** Non-resolvable address are completely random (except for the two MSB) which cannot be recognized by unpaired devices. While resolvable address consists ofr 24 bit random dan hash (derived from the Identity Resolving Key) part, and only devices that know the IRK can recognize the device.

1.8. Generic Attribute (GATT)

GATT defines a framework for using the ATT protocol and it is used for services, characteristics, descriptors discovery, characters reading, writing, indication, and notification. When two devices are connected, there are two devices role:

- **GATT Client**: The device accesses data on the remote GATT server viar read, write, notify, or indicates.
- **GATT Server**: The device stores data locally and provides data access methods to a remote GATT client.

A device could possibly acts as the GATT client and server at the same time. A slave (peripheral) device has to be GATT server and a master (central) device hasn't to be the GATT client. Attributes are encapsulated with the **Characteristics** (with descriptors) and **Services** (primary, secondary, and include).

1.8.1. Characteristics Attribute

A characteristics exposes the type of data that the value represents, if the valuen can be read or written, how to configure the value to be indicated or notified, and it says what a value means. Charcteristics attributes consists of **characteristics declaration, value,** and **descriptors**.

1.8.2. Characteristics Decriptor

A descriptors are used to describe the characteristics value to add a specific "meaning" to the characteristics and making it understandable. The characteristics descriptors are including charcteristic extended properties, user description, client characteristic configuration, server characteristic configurations, characteristic presentation format, and characteristic aggregation format.

1.8.3. Service Attribute

A service is a collection of characteristics which operate together to provide global service to an applicative profile. A service has service declaration and include declaration.

1.8.4. GATT Procedures

GATT defines a standard set of procedures allowing services, characteristics, related descriptors to be discovered and how to use them. The procedures are including:

- Discovery
- Client-initiated
- Server-initiated

1.9. Generic Access Profile (GAP)

GAP enables the bluetooth devices to be visible and determines the function. In GAP, we need to define the device role, as the **Broadcaster** or **Obeserver** or **Central** or **Peripheral** device. When the device acts as the **Central**, the device usually have more processing power and memory

such as mobile phones; while **Peripheral**, the device are realively low power, and have sensors to read data and send it to the **Central** device.

Role[1]	Description	Transmitter	Receiver	Typical example
Broadcaster	Sends advertising events	M	O	Temperature sensor which sends temperature values
Observer	Receives advertising events	O	M	Temperature display which just receives and displays temperature values
Peripheral	Always a slave. It is on connectable advertising mode. Supports all LL control procedures; encryption is optional	M	M	Watch
Central	Always a master. It never advertises. It supports active or passive scan. It supports all LL control procedures; encryption is optional	M	M	Mobile phone

1. *1. M = Mandatory; O = Optional*

Figure 6 GAP Roles

There are two fundamental concepts in GAP

- **GAP modes**: It configures a device to act in a specific way for a long time. GAP modes are including **broadcast, discoverable, connectable**, and **bondable** type.
- **GAP procedures**: It configres a device to perform a single action for a specific limited time. The procedures types are **observer, discovery, connection**, and **bonding**.

2. Bluetooth Mesh

Bluetooth Mesh is not a wireless communication technology. It is a networking technology that built upon Bluetooth Low Energy (BLE).

2.1. Key Concept

- Mesh profile = Fundamental requirements for BLE Mesh networking.
- Mesh model = Define basic nodes functionality on a mesh network.

2.2. Flooding Protocol

In mesh network, flooding protocol enables the node to recognized whether the received messages are new or old, if old then the message won't be relayed, therefore the network usage is optimized and secure against replay attacks.

2.3. System Architecture

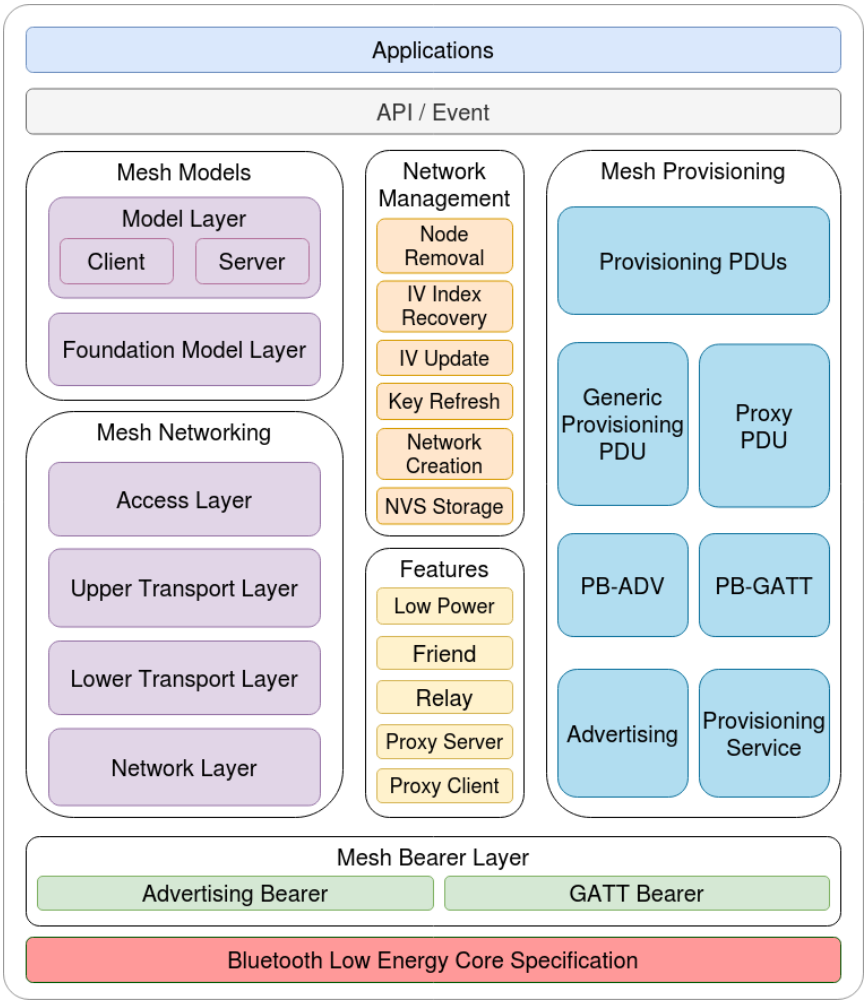


Figure 7 BLE Mesh Architecture

Mesh Networking consists of five main components, including:

2.3.1. Mesh Protocol Stack

2.3.1.1. Mesh Networking

It is responsible for processing of messages of BLE-MESH nodes It implements several functions, including:

- Facilitate communication between nodes.
- Encrypt and decrypt messages.
- Manage Mesh network resources.
- Segment and reassembly of Mesh network messages.
- Model mapping of messages between different models.

These functions are span in four hierarchy layers:

Access Layer	Defines application formats, encryption and decryption of data packets.
Upper Transport Layer	-- Encrypts, decrypts, and authenticates application data to and from the access layer. -- Handles transport control messages, including friendship and heartbeat messages.
Lower Transport Layer	Handles segmentation and reassembly of PDU.
Network Layer	Defines the address type and format of the network messages, and implements the relay function of the device.

2.3.1.2. Mesh Provisioning

It is responsible for provisioning flow of BLE-MESH devices. Several functions include:

- Provision of unprovision devices.
- Allocate Mesh network resources/
- Four authentication methods support during provisioning.

Mesh Provisioning consist four hierarchy layers:

Provisioning PUDs	PDUs from different layers are handled using provisioning protocol
Generic Provisioning PDU/Proxy PDU	The Provisioning PDUs are transmitted to an unprovisioned device using a Generic Provisioning layer or Proxy protocol layer.

PB-ADV/PB-GATT	Defines how the Provisioning PDUs are transmitted as transactions that can be segmented and reassembled.
Advertising/Provisioning Service	Defines how sessions are established, i.e., the transactions from the generic provisioning layer can be delivered to a single device.

2.3.1.3. Mesh Elements and Models

It is responsible for the implementation of SIG-defined models. The monitoring and the configuration of the mesh devices are done through elements and models, a device could have multiple roles which called as element. The **elements** are supported **up to 8** and each element could be supported **up to 7 models.**

2.3.1.3.1. Elements

Part of a single node, which can be **independently controlled** and each element must contain required models. The number and the structure of elements of a node is static and doesn't change throughout the lifetime of a node, any changes will require the node to be reprovisioned. For instance, LED Lighting product is associated as a single node which consist of three elements of lamp.

2.3.1.3.2. Models

Represents **a specific service** and has **unique sets of messages** to process, and states that may be shared with other elements. An element within a node must support at least one model because a model help to define a product behaviour as a mesh device. The model types are defined by the **Bluetooth SIG**. There are mandatory models in mesh network configuration.

2.3.1.3.2.1. Configuration server model

It represents a mesh device network configuration of a device, its responsible for maintaining configuration-related states, including: NetKey List, AppKey List, Model to AppKey List, Node Identity, Key Refresh Phase, Heartbeat Publish, Heartbeat Subscription, Network Transmit, Relay Retransmit etc. This model must only be supported by a primary element. It also uses the device key for application layer security.

2.3.1.3.2.2. Configuration client model

Represents an element that can control and monitor the configuration of a node.

2.3.1.3.2.3. Health server model

It represents a mesh device network diagnostic. The Configuration Client Model uses messages to control the state maintained by the Configuration Server Model. The Provisioner must contain the Configuration Client Model, with which the configuration messages, like Configuration Composition Data Get can be sent. It also uses application key for application layer security.

2.3.1.3.2.4. Health client model

Represents an element that can monitor the health messages of a node. The Health Client Model uses messages to control the state maintained by the Health Server Model. The model can get the self-test information of other nodes through the message "Health Fault Get".

2.3.1.3.2.5. Generic client/server models

2.3.1.3.2.6. Sensor client/server models

2.3.1.3.2.7. Time and scenes client/server models

2.3.1.3.2.8. Lighting client/server models

2.3.1.3.2.9. Remote Provisioning Server model

The model is used to support the functionality of provisioning a remote device over the mesh network and to perform the Node Provisioning Protocol Interface procedures.

2.3.1.3.2.10. Remote Provisioning Client model

The model is used to support the functionality of provisioning devices into a mesh network by interacting with a mesh node that supports the Remote Provisioning Server model.

2.3.2. Network management

Implements several network management procedures, including:

- Node removal procedure to remove a node from the network.
- IV update procedure is used to update the nodes' IV Index.
- Key refresh procedure is used to update the nodes' NetKey, AppKey, etc.
- Network creation procedure is used to create a mesh network.
- NVS storage is used to store node's networking information.

2.3.3. Nodes Features

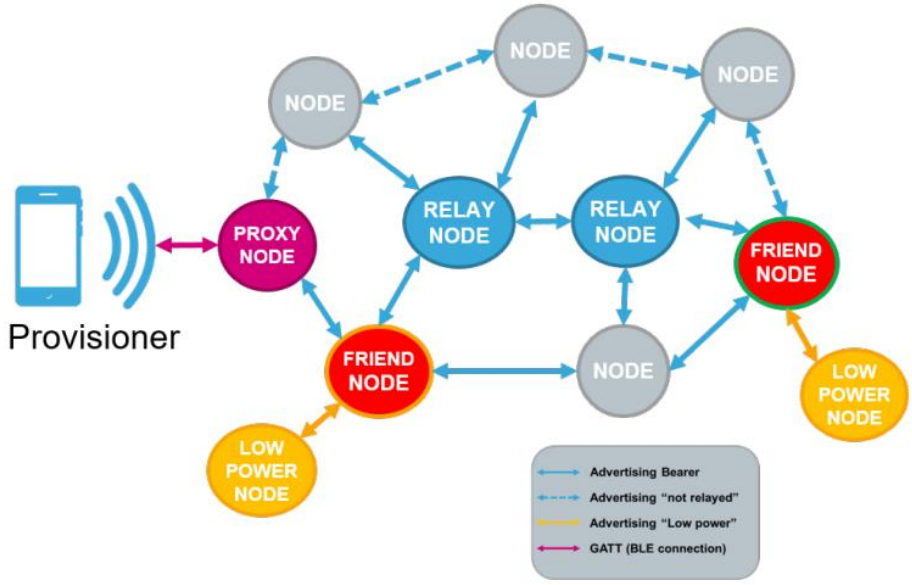


Figure 8 Mesh Topology

- **Proxy Server/Client**

  Expose BLE devices that don't support mesh networking to interact with nodes in a mesh network. It is two node roles in proxy protocol, which enable nodes to send and receive Network PDUs, mesh beacons, proxy configuration messages and Provisioning PDUs over a connection-oriented bearer.

- **Simple**

  Leaf nodes are unusable nodes because they can't relay messages and usually contains legacy nodes or resource constrained devices.

- **Relay**

  Retransmit received message which then traverse the entire mesh network, making multiple "hops" between devices by being relayed.

- **Low power nodes (LPN)**

  Battery operated devices that primarily send but barely receive messages, and do not need 100% duty-cycle.

- **Friend nodes**

  Devices that store messages for LPN and deliver them whenever the LPNs wake up or poll for "waiting messages"

2.3.4. Mesh Bearer Layer

Mesh-Protocol Stack utilized the bearer layer to transmit data via the BLE Advertising channel and connection channel, including Advertising Bearer and GATT Bearer. To simply put, it is responsible to pass data between the Mesh-Protocol Stack and BLE.

- **Advertising Bearer (PB-ADV)**

Nodes use the BLE GAP **advertising and scanning channels** to send and receive the messages. This way can only be used for provisioning when provisioner and unprovisioned device both support PB-ADV.

- **GATT Bearer (PB-GATT)**

  The smartphone and the Proxy node use the **BLE GATT connection channels** to send/receive messages in a point-to-point topology. It **uses a standard GATT service** and **provides two characteristics**, one for handling **value notifications**, and one for **writing commands**. If an unprovisioned device wants to be provisioned through this method, it needs to implement the related Mesh Provisioning Service. Unprovisioned devices which don't implement such service cannot be provisioned into mesh network through PB-GATT bearer.

2.3.5. Applications

This layer utilizes Mesh Protocol Stack and Mesh Models by calling API and handling event to intercat with Mesh Networking, Mesh Models, and Mesh Provisioning. The interaction between application layer and API are including:

- Call the provisioning-related API for provisioning.
- Call the model-related API to send messages.
- Call the device-attributes-related API to get local information about the device.

The application layer is designed based on events, which take parameters to the application layers. It consists of two categories, including:

- The events completed by calling API, such as nodes sending messages.
- The events that protocol stack actively reports to the application layer
  - The Event that the protocol stack actively reports.
  - The Event that Model actively reports.

2.4. Security

In BLE Mesh, security is mandatory and **all messages** that transported are **encrypted.** A unique address is assigned to ecah node by the provisioner. There are three security keys:

[1] **Network Key (NetKey)**

A membership key which shared across all the nodes on the network.

[2] **Flooding Security Material**

It is derived from NetKey and can be used by other nodes in the same network. Messages encrypted with flooding security material can be decoded by any node in the same network.

[3] **Directed Security Material**

The directed security material is derived from the NetKey and can be used by other nodes in the directed forwarding path.

[4] **Application Key (AppKey)**

Used to secure communications at the upper transport layer by encrypting/decrypting application messages, and differentiate applications from each other. Application key is used for decryption of application data before delivering application data to application layer and encryption of them during the delivery of application layer. Some nodes in the network have a specific purpose and can restrict access to potentially sensitive data based on the needs of the application. **AppKey is bound to NetKey, which means it only can be used on the attached NetKey and it should be only bound to a single NetKey**

[5] **Device Key (DevKey)**

Unique device key that was visible only to the provisioner and the device itself, it is used for provisioning, configuration, and key management. The device key is also used to encrypt Configuration Messages, i.e., the message transferred between the Provisioner and the node when the device is configured.

Messages are encrypted twice, once with an application or device key and the second time with a network key. The mesh profile **security model uses a privacy** mechanism called **obfuscation** through AES to encrypt the source address, sequence numbers, and other header information, which make the message difficult to track. Other security also added such as ECDH (**Elliptic Curve Public Private Key Pair**) which enable to **share secret over an insecure channel**. Also, MIC (**Message Integrity Check**) for authentication and confidentiality.

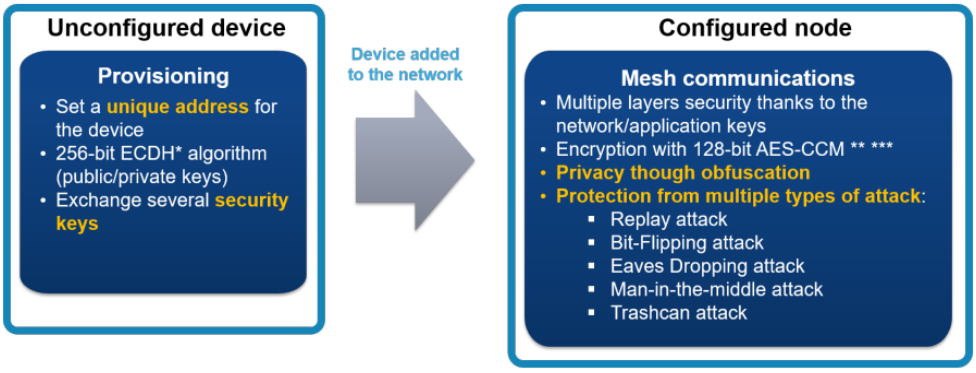


Figure 9 Security done by provisioning

2.5. Mesh Packet Data Unit

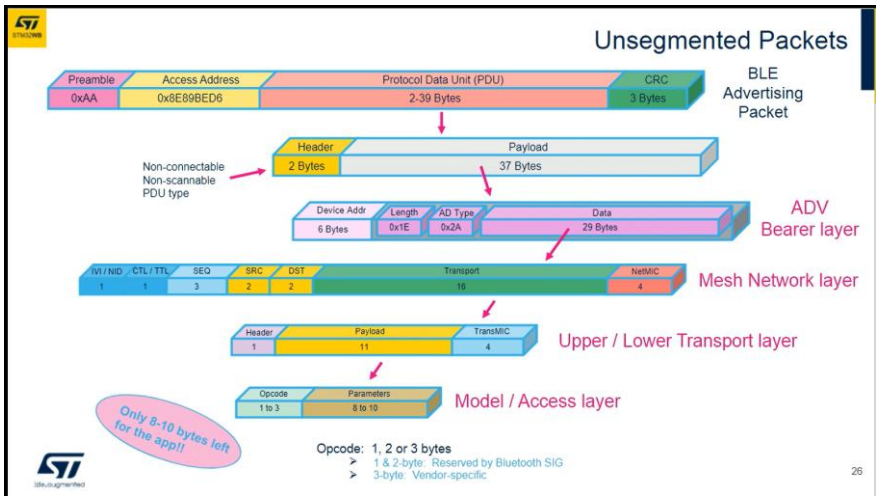


Figure 10 Mesh Packet Data Unit

2.6. Address

a. Unassigned

Its indicates that an Element hasn't yet been configured or had a Unicast Address assigned to it.

b. Unicast

Assigned to a device during provisioning process which **represent a single element** in a node. A unicast address may appear in the source/destination address field of a message. Messages sent to a unicast address can only be processed by the element that owns the unicast address.

c. **Group**

A multicast address representing **at least one element in one or more nodes**.

d. Virtual

A set of destination addresses that represented as a Label UUID, which is a 128-bit value.

2.7. Publish and Subscribe

Nodes can publish and subscribe messages using unicast, virtual, or group addresses.

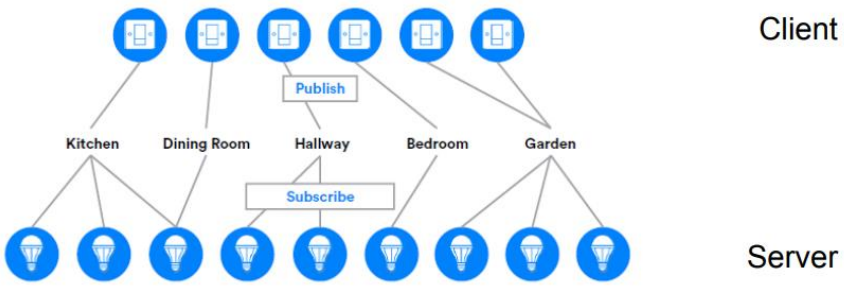


Figure 11 Publish and Subscribe Illustration

2.8. Messages

Mesh network uses message-oriented communication which acknowledgement with a response reliable message, conversely, unreliable message won't receive any acknowledgement.

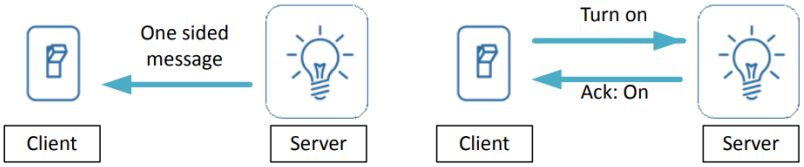


Figure 12 Message Oriented illustration

2.9. Friendship and Low Power Node (LPN) Messaging

LPN is **power sensitive** device because powered using a battery, therefore they have little duty-cycle to save power and **barely receive message**, but still send **message sometimes**. LPN depends on friend node; **a friend node caches** all incoming messages and **waits** for the polling message from the LPN to provide it with any message.

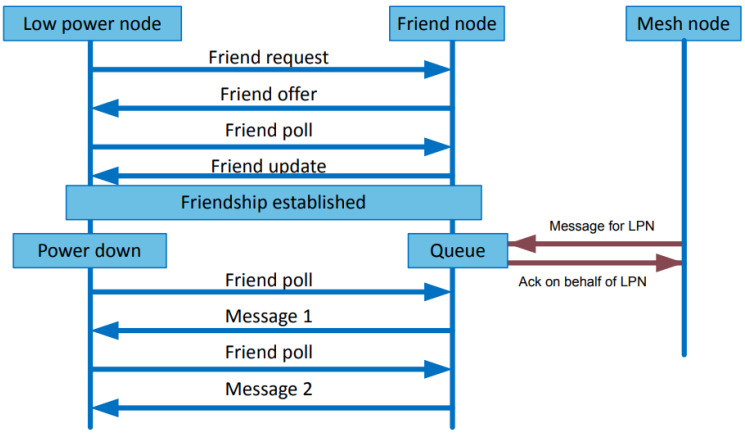


Figure 13 Friendship and LPN Messaging Communication

2.10. Provisioning Process

Provisioning is a process of configuring a device on a network. The process is initialized by a **provisioner** which usually placed on a **smartphone** or could be MCU that supports BLE.

a. **Beaconing**

   A new GAP AD type introduced, including mesh beacon AD type.

b. **Invitation**

   The provisioner sends an invitation to the unprovisioned device.

c. **Public Key exchange**

The provisioner and the unprovisioned device exchange their public keys, either directly or OOB (Out of Band) method.

d. **Authentication**

   A cryptographic exchange takes place.

e. **Distribution of the provisioning data**

   A session key is combination from both keys and peer public keys, and then used to secure the subsequent data exchange to complete the provisioning process, including NetKey. After the provisioning completed, the provisioned device owns the NetKey, the IV index, and a unicast address which allocated by the provisioner.

f. The device is known as **node** and could be assigned to a group.

3. Mesh Terminology

a. Unprovisioned = Unregistered device in the Mesh network.
b. Node = A device or sensor that being provisioned.
c. Relay Node = A node that supports and able to relay message to other node.
d. Proxy Node = A node that supports and uses the Proxy features.
e. Friend Node = A node that stores the Low Power Node data.
f. Low Power Node = A node that has Low Power features (usually battery-powered device) and befriend with Friend Node.
g. Provisioner = A node that can register unprovisioned device to the Mesh network.
h. State = A value representing a condition of an element that is exposed by an element of a node. I.e. brightness, brightness, on/off, etc.
i. Model = Device basic functionality.
j. Element = An addressable entity.
k. Composition Data State = Information about the node, including the supported models and its elements.
l. PB-ADV = Provisioning bearer used to provision a device using **Generic Provisioning PDUs** over the advertising channels.
m. PB-GATT = Provisioning bearer used to provision a device using **Proxy PDU**s to encapsulate Provisioning PDUs within the Mes Provisioning Service.
n. PB-Remote = Uses the existing mesh network to provision an unprovisioned device that isn’t within immediate radio range of the Provisioner.
o. Reassembly and Segmentation (SAR) = Communication network method, which divide data into small units before the transmitter device transmit the packets and reassembled the received packets in a proper order at the receiver device. This is automatically completed by the lower transport layer.

References

[1] ESP-BLE-MESH - ESP32 - — ESP-IDF Programming Guide v5.2.2 documentation (espressif.com)

[2] Bluetooth® LE mesh overview - stm32mcu

[3] NUCLEO-WB55RG - STM32 Nucleo-64 development board with STM32WB55RG MCU, supports Arduino, ST Morpho connectivity - STMicroelectronics