Bluetooth blog

The Fundamental Concepts of Bluetooth Mesh Networking, Part 2

Posted on August 14, 2017 by Bluetooth Developer Relations Team

Chapter 4 of the Bluetooth Mesh Networking Series

With Bluetooth® mesh, you can create large-scale networks capable of supporting secure, reliable communication between tens, hundreds, or thousands of devices. In part 1 of The Fundamental Concepts of Bluetooth Mesh Networking, we explored some of the basic concepts of a Bluetooth mesh network, including nodes, elements, models, and states. In this article, we examine addressing, messages, publications, subscriptions, and lists, and detail how these core concepts interweave to make a Bluetooth mesh network.

The Bluetooth Mesh Architecture

Bluetooth mesh runs on top of the Bluetooth Low Energy (LE) stack. Figure 1 below outlines the Bluetooth mesh stack and defines the functionality of each layer.

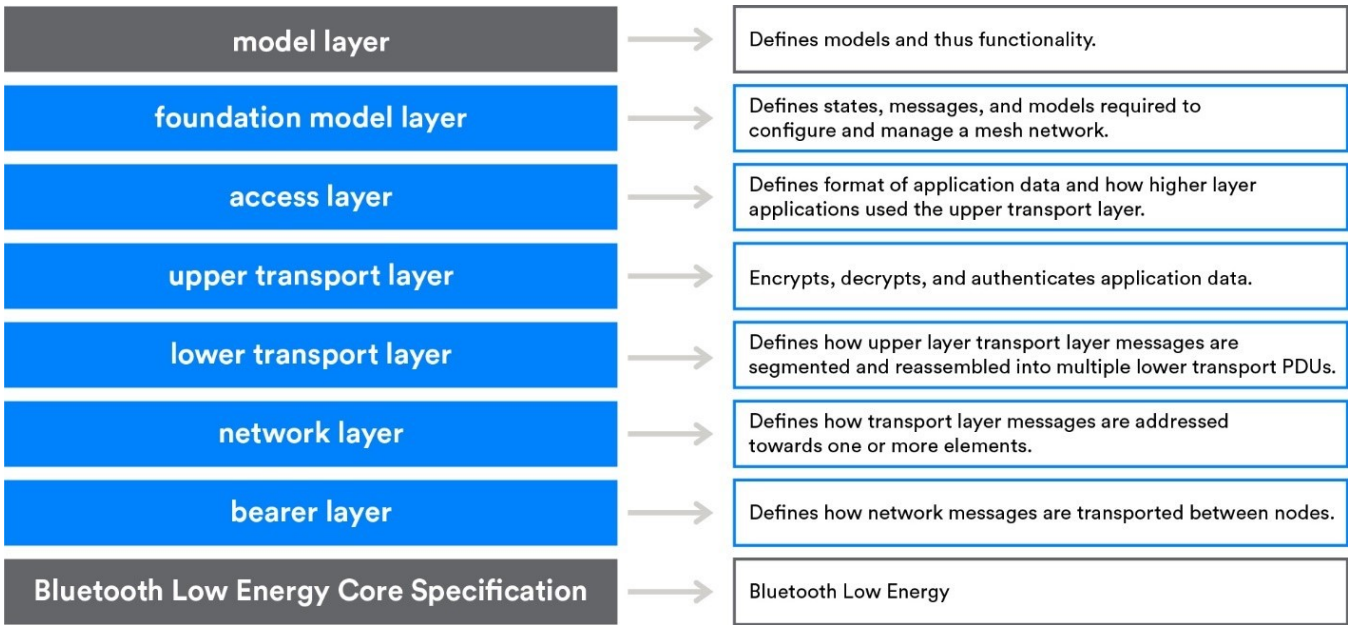

Figure 1 – Bluetooth mesh architecture.

As we discussed in part 1, nodes—such as light fixtures, temperature regulation equipment, manufacturing equipment, and electronic gates—are devices capable of sending, receiving, and/or relaying information within the Bluetooth mesh network. Messages are used to transfer data between nodes, and addresses are used to define where the messages come from (source) and go to (destination).

Addresses

There are four types of addresses; three of these types are used in messaging: unicast, virtual, and group addresses. The fourth is known as an unassigned address. Addresses are 16 bits in length and are encoded as defined below (Figure 2).

Values	Address Type
0b0000000000000000	Unassigned Address
0b0xxxxxxxxxxxxxxx (excluding 0b0000000000000000)	Unicast Address
0b10xxxxxxxxxxxxxx	Virtual Group
0b11xxxxxxxxxxxxxx	Group Address

Figure 2 – Mesh address encoding.

- **Unassigned Address** – Unconfigured elements, or elements without designated addresses, have an unassigned address. Given these elements do not have a unique address, they may not be used in messaging.
- **Unicast Address** – During provisioning, a provisioner allocates a unicast address to each element in a node for the lifetime of that node on the network. Unicast addresses may appear in the source address field and/or the destination address field of a message. Messages sent to unicast addresses are only processed by one element.
- **Virtual Address** – Virtual addresses are a set of elements associated with a specific Label UUID; these addresses may be published or subscribed to. The Label UUID is a 128-bit value associated with multiple elements that may come from one or more nodes.

  For virtual addresses, bits 15 and 14 are set to 1 and 0 respectively (Figure 2); bits 13 – 0 are set to a hash value (providing 16,384 hash values). The hash is derived from the Label UUID. Checking the full 128-bit UUID with a subscribing element is inefficient, especially as the UUID may span more than one message segment. The hash values provide a more efficient way of determining which messages go to which elements.

- **Group Address** – Group addresses are another type of multicast address found in Bluetooth mesh networking. Representing multiple elements from one or more nodes, there are two types of group addresses:
    - *Dynamically assigned* -> 0xC000-0xFEFF
    - *Fixed addresses* - Assigned by Bluetooth SIG and divided into five segments:
        - *Reserved for Future Use (RFU)* –> 0xFF00-0xFFFB
        - *All-proxies* -> 0xFFFC
            - Sent to all nodes with proxy functionality enabled.
            - *All-friends* -> 0xFFFD
                - Sent to all nodes with friend functionality enabled.
            - *All-relays* -> 0xFFFE
                - Sent to all nodes with relay functionality enabled.
            - *All-nodes* -> 0xFFFF
                - Sent to all nodes.
            - All messages sent to fixed nodes are processed by the primary element of the node.

FEATURED RESOURCE

Bluetooth Mesh Networking - An Introduction for Developers

Download this comprehensive technology overview to learn more about the key concepts and terminology, system architecture, and security mechanisms, as well as the unique message publication and delivery technique behind Bluetooth mesh networking.

Messages

Bluetooth mesh networks communicate via messages. A message may be termed a control message or an access message.

- *Control Messages* – Messages concerned with the operation of the Bluetooth mesh network. Examples include heartbeat and friend request messages.

- *Access Messages* – Allow client models to retrieve or set the value of state values in server models, or they are used to report state values by the server.

Models implement and define the functionality of nodes. Elements are uniquely addressable entities within nodes containing one or more models and states define the status of elements. For every state, there is set of messages that a server model supports. Examples include a client model requesting the value of a state or requesting to change a state and a server model sending messages about states and/or a state change.

Messages are identified by opcodes and have associated parameters. An opcode identifies the operation of the message. Examples include:

- Generic OnOff Get – Used to identify the state; OnOff state for a generic model.
    - There are no parameters for Generic OnOff Get.
- Generic OnOff Set is used to set the OnOff state of a generic model.
    - Parameters:
        - OnOff – The target value (on or off).
        - TID – Transaction Identifier – Is the message new or a re-transmission.
        - Transition Time – How long an element should take to transition from one state to another.
        - Delay – Message execution delay.

There are two categories of access message, acknowledged and unacknowledged. Acknowledged messages are transmitted to and acknowledged by each receiving element. The response is typically a status message. No response is sent to an unacknowledged message. Bluetooth mesh network status messages are an example of unacknowledged messages.

Message Security

Every Bluetooth mesh network message is secured using NetKeys and AppKeys to encrypt and authenticate messages. NetKeys are used for the network layer communication. Assuming a Bluetooth mesh network has no subnets, all communication within that mesh network uses the same network key.

AppKeys are used for application data. Some nodes within the network may have specific applications and, within these applications, potentially sensitive data requiring limited access. Such nodes have a specific AppKey and are associated with specific applications. Examples of areas that

may use different AppKeys include security (building access control, equipment room access, and CEO office access), lighting (manufacturing floor, exterior building lights, and walkways), and HVAC systems.

Relay nodes, such as light bulbs or wall switches, typically have valid NetKey and can relay sensitive messages within the network. However, they would not have access to the specific AppKeys for the various restricted areas, such as building control or HVAC Systems, and couldn't decrypt the application data.

Message Exchange

Bluetooth mesh networking uses a publish/subscribe model for message transport. Nodes generating messages are said to publish messages. Nodes interested in receiving messages subscribe to addresses they are interested in. Messages may be published to unicast, group, or virtual addresses.

Messages may be sent as replies to other messages, or they may be unsolicited messages. When a model sends a reply message, it uses the message originator's source address as the destination address. When sending an unsolicited message, it uses the model's publish address as the destination address. Every model in a node has a single publish address.

When receiving messages, each instance of a model within a node (there may be multiple models in a node) may subscribe to receive messages from one or more group or virtual addresses.

Models subscribing to messages use a model's subscription lists to define valid addresses that they can receive messages from. When messages are received by a model, the model checks its subscription list. It's considered a match when the address on the subscription list is set to the model's element unicast address or a fixed group address that belongs to the node. Figure 3 shows valid source and destination addresses for access messages.

Address Type	Valid Source Address	Valid Destination Address for Access Messages
Unassigned Address	No	No
Unicast Address	Yes	Yes
Virtual Group	No	Yes
Group Address	No	Yes

Figure 3 – Valid source and destination addresses for access messages.

As Bluetooth mesh entities publish the status of various nodes, systems throughout the whole Bluetooth mesh network can subscribe to this data regardless of proximity to a transmitting node's location. This allows equipment on one side of the network to talk to administrators elsewhere in the facility via low-power wireless messages, regardless of distance.

Learn More About Mesh

Bluetooth mesh combines the proven, global interoperability and mature, trusted ecosystem associated with Bluetooth technology to support the creation of industrial-grade device networks. Now that you have a basic understanding of the fundamental concepts behind Bluetooth mesh networking, you're ready to take a deeper dive into the intricacies of the topology. For an in-depth look at Bluetooth mesh, download the Bluetooth Mesh Technical Overview. Later in the Bluetooth Mesh Networking Series, we'll explore Bluetooth mesh security, provisioning, proxy nodes, and more.

ON-DEMAND WEBINAR

What Makes Bluetooth Mesh so Disruptive?

The behind-the-scenes story of the making of Bluetooth mesh

Watch our free on-demand webinar to discover how Bluetooth mesh is disrupting building automation, wireless sensor networks, asset tracking, and more.

SHARE:

This entry was posted in Bluetooth Mesh, Bluetooth Developer, and tagged Bluetooth Developer, Bluetooth Mesh Networking Series, Mesh Technology by Bluetooth Developer Relations Team