

Simplifying BLE Mesh Integration

Bluetooth Low Energy (BLE) Mesh technology allows smart devices to maintain a cooperative wireless network that can be managed remotely, leading to the development of smart homes, building automation, and sensor networks.

Bluetooth Low Energy (BLE) is a wireless technology available on Android-, iOS-, Windows-, and Linux-based devices. It's designed to provide connectivity between a master controller, typically a smartphone or tablet, and smart devices.

Smart devices are everywhere and include everyday Internet of Things (IoT) objects such as lighting, heating systems, home appliances, access alarms, sensors, location tags, and wearables. Using BLE Mesh technology, these smart devices can connect to a wireless network and be managed remotely, enabling the development of smart homes, building automation, sensor networks, and smart industries and cities.

OVERVIEW

A BLE mesh is a network of as many as 32,766 smart devices, although most networks are much smaller. Messages sent and received within this network initiate predefined events that occur in the participating devices. Any BLE device running

BT4.0 or higher has the potential to support BLE meshes, provided the proper firmware is installed.

The BLE mesh uses a managed flood technique for transmitting messages to connected devices, which are called nodes (*Fig. 1*). It sends a message, called a BLE broadcast or advertisement, to those nodes that are within range. The receiving nodes rebroadcast, or relay, this message to other nodes within their range. Once the proper node receives the message, the predefined event is initiated.

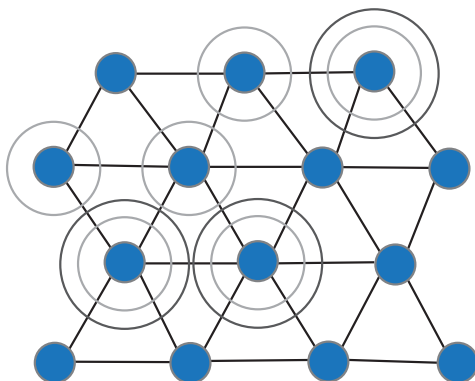
Messages aren't rebroadcast unconditionally. The nodes rebroadcast until the message time-to-live (TTL) has expired, where the TTL defines the number of relays, or hops, that a message can make. The TTL is decremented by one as it reaches each relay node. Relay nodes will only forward messages that have a TTL greater than one, to avoid an endless cascade of messages. Also, each node in the mesh maintains a message cache of its most recent messages, and incoming messages that match one already received are discarded without rebroadcasting.

BLE mesh networks using relay nodes are far more reliable than other types of networks that use nodes as routers. Since relay nodes receive broadcasts from any node within range, this redundancy keeps the network up even if an intermediate node fails or is removed from the network, while router nodes can be a single point of failure. Relay nodes also provide better scalability and improved performance.

Periodically, a heartbeat message can be sent from each node to indicate that it's still active. Using the heartbeat message to see

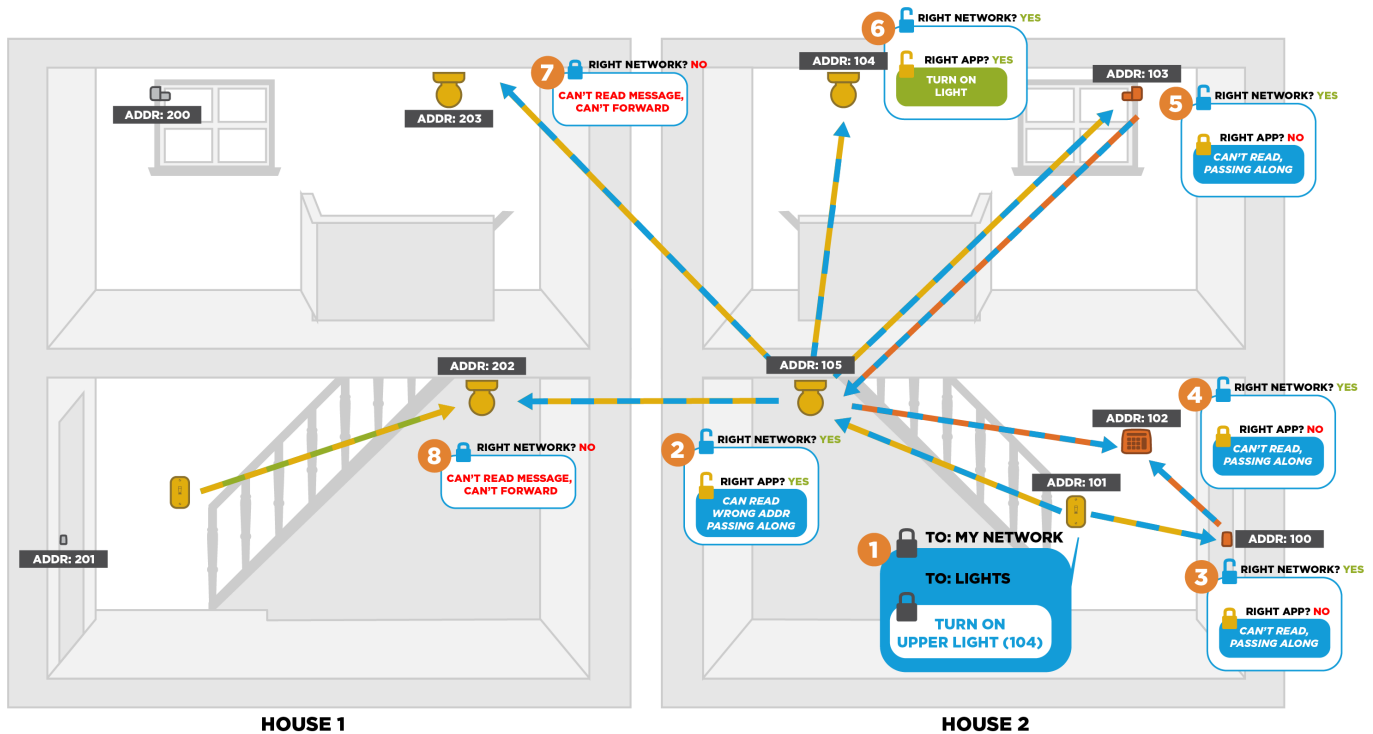
BLE mesh is a managed flood network

Message relay and managed flooding



- Resending received packets
- BLE advertising & scanning
- Allows range extension
- Limited by TTL
- Message cache
- Optional relay

1. The BLE Mesh takes a managed flood approach to distributing a system through the network.



2. This is an example of the BLE Mesh in action with neighboring networks.

how far each node is from other nodes, in terms of the number of hops needed to reach it, allows the TTL to be optimized and set to a value no higher than necessary.

TERMINOLOGY

While all nodes in a BLE mesh can transmit and receive messages, some nodes have additional specific purposes. A “relay” node can rebroadcast a message that it receives. A “low-power” node spends most of its time in a low-power state with its radio turned off to save battery life. A “friend” node works in conjunction with a low-power node by storing and then forwarding messages when it’s polled by that low-power node.

A “proxy” node is a device that uses legacy Bluetooth connectivity to interact with other devices in the BLE mesh; in turn, it can forward messages to other nodes via other connections or adverts, if the relay feature is also enabled. Smartphones are expected to be the type of nodes that will use proxy nodes to convey messages simply because a phone can’t use relay mode. A relay mode requires the radio to be switched on 100% of the time, which isn’t going to be allowed by the phone manufacturer.

The Mesh spec allows a node to be in multiple modes, like relay, proxy, and friend. It’s unlikely a node using low-power mode will use any other mode.

Messages are sent and received using a publish/subscribe paradigm (Fig. 2). Publishing involves sending messages from one node to a group of one or more other nodes. Subscribing is configuring a node so that it can receive messages from 0

or more nodes. A node has a subscription list that contains the publish addresses to which it has subscribed, so that it can select or filter out messages that it receives.

Each message consists of an opcode and a payload, or content data. The opcode dictates the behavior at the receiving end. Outgoing messages on a BLE mesh are encrypted with both a network key and an application key. The network key provides security for all communication within the BLE mesh network. The application key delivers separation and authentication to ensure the correct model in the intended device is the one that’s activated—for example, a message sent to turn on a light bulb should not be able to unlock a door. By the time the application receives its message, it has been completely decrypted and is given as plaintext.

PROVISIONING A NODE

When a BLE mesh device is powered up for the first time, it’s not “provisioned.” That is, it doesn’t have a node address or any other configuration information. To become part of the mesh, it must be given this information. Once the device is provisioned, typically by a smartphone, it’s called a node, and has a unicast address, publish information, a subscription list, and network and applications keys, among other things. Provisioning updates the records in the device’s persistent configuration database. Each node will have one or more elements, which in turn have one or more models, that expedite a particular behavior for that node.

The provisioner is the only entity in the mesh that’s aware of all members in the network. It’s also the only device that

knows how to program the publication address and the subscription lists of every node. No individual node is aware of the entire mesh. The provisioner isn't needed for the network to function, but it is needed to add or remove nodes from the network. In this way, there's no central point of failure that can bring down the entire network. However, if the provisioner device is damaged or lost, then no one else knows all of the nodes in the network; they must all be re-provisioned with another device.

As an illustration, consider an office with 50 lights and 20 switches. Only the provisioner is aware of the node addresses of all 70 devices. The provisioner can create group addresses for each area of the office, and then set the publication addresses in the switches and the subscription addresses in the lights based on the group addresses. When a light switch's state is changed from off to on, for example, it publishes a group address and the application-subscribed lights turn on.

IMPLEMENTING A BLE MESH

Implementing a BLE mesh may sound overwhelming, but the process can be greatly simplified. Manufacturers like Laird Connectivity offer BLE modules that jumpstart the mesh-development process. BLE modules have an embedded event-driven scripting language, smartBASIC, built into their firmware that's used by product developers to simplify BLE module integration.

By using only a handful of built-in functions and events, an application can be written for implementing a BLE mesh using a Laird BLE module. The first time the BLE mesh is started, the underlying stack will see that the device isn't provisioned, which starts the beaconing process. This alerts the user, who will provision the device, making it a node. Otherwise, the start function lets the node participate in the mesh network and listen for incoming messages, which it will either act on or relay onwards.

When an incoming message is fully decrypted by the underlying stack, it triggers a predefined event and the application script will have a user-written handler that defines the behavior for that event. Such a script examines the opcode in the message and determines what action should be taken, such as switching on a light, starting a washing machine, or setting the house alarm or even send the information to the cloud.

Programmable BLE modules, like the BL652 and BL654 from Laird Connectivity, provide secure wireless engines for IoT networks. They have a wide range of configurable interfaces and an industrial temperature rating, and there's a fully featured development kit to start BLE development. The power-management-efficient BL652 and BL654, which measure 10 x 14 mm and 10 x 15 mm, respectively, provide hostless operation for automated use cases.

CONCLUSION

Implementing a BLE mesh network allows a user to connect IoT devices in a smart wireless network. The network is flexible, scalable, robust, and reliable. Using commercially available BLE modules greatly simplifies the implementation process, enabling these smart networks to be up and running more quickly.

Mahendra Tailor is a Technology Leader for Laird Connectivity, where he has worked for 21 years. For nearly 15 years, he has been involved mostly in Bluetooth, which has resulted in many patents. Mahendra has worked in embedded and automation systems for over 35 years, after getting a degree in Control and Instrumentation Engineering from The City University in London. His goal is to make technology easy for others to embed into end products.