Bluetooth blog

Bluetooth Mesh Networking: Friendship

Posted on August 24, 2017 by Kai Ren

Chapter 5 of the Bluetooth Mesh Networking Series

Bluetooth Low Energy (LE) is one of the world's most power-efficient, short-range wireless communications technologies. Its low power consumption is widely praised by developers and consumers. With the release of Bluetooth mesh networking, developers may be wondering whether Bluetooth mesh has also been designed with low power consumption in mind. Does it inherit the beauty of the low power consumption of Bluetooth LE?

The answer is yes, definitely! Bluetooth mesh networking includes various measures to optimize power consumption and, in particular, a feature called “friendship”.

Overview

The application of the friendship feature in Bluetooth mesh networking is likely to be very diverse. Some products, like lights, will be connected to the main electricity power source (the national power grid), and the power consumption of the Bluetooth mesh module, compared to the power consumption of the light itself, will be negligible. But other products, like smart sensors or locks, will be power constrained, which means they need to be powered by a small battery or energy harvesting technique. It’s products like these that are most likely to leverage the friendship concept of Bluetooth mesh.

If you’ve read earlier articles in our Bluetooth Mesh Networking Series, you already know that a node is a device which has been provisioned and is a member of the mesh network. Nodes have functionality related to the product type, but may also have functionality concerning the operation of the network itself and can take on special roles. This is determined by the mesh features they support. All nodes can transmit and receive mesh messages in the network. In addition, nodes can also optionally support one or more additional network features as listed below:

- **Relay feature**: The ability to receive and retransmit mesh messages over the advertising bearer to enable larger networks.

- **Proxy feature**: The ability to receive and retransmit mesh messages between GATT and advertising bearers.
- **Low-power feature**: The ability to operate within a mesh network at significantly reduced receiver duty cycles. Minimizing the time the radio receiver is on leads to lower power consumption with the node only enabling the receiver when strictly necessary. Low-Power nodes (LPN) achieve this through establishing a friendship with a Friend node.
- **Friend feature**: The ability to help an LPN operate by storing messages destined for the LPN and only forwarding them to it when the LPN explicitly requests messages from the Friend node.

To understand how friendship enables LPNs to reduce their power consumption, consider sensors. Sensors are a good example of a type of node likely to exploit friendship and act as LPNs. They usually spend the most significant proportion of their time transmitting data and only rarely need to receive it. Maybe a sensor only transmits a temperature reading whenever it falls outside of a set of configured limits, and perhaps this only happens twice a day. This infrequent transmission of data keeps the energy use for this type of device low.

But what if those temperature limits need to be modified to use different values according to the season and modification of these limits is achieved by sending a configuration message to the sensors? For a sensor to receive such messages directly, it needs to switch the radio on and listen. Most of the time it is listening it receives nothing, but energy is expended nevertheless.

So, working with a friend allows the LPN to schedule its use of the radio to receive messages to whatever frequency makes sense for that device, and at a much lower frequency than it would otherwise need if it had to listen for messages all the time. LPNs poll their friends for new messages, which the friend stores only occasionally. This is how power is saved.

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.

Friends and LPNs

An LPN must establish a friendship relationship with another node supporting the Friend feature in order to reduce its receiver duty cycles and save energy. Figure 1 is taken from the Bluetooth mesh profile specification. Amongst other things, it illustrates the relationship between LPNs and Friend Nodes. In particular it shows:

- Light blue: LPNs
- Dark grey: Friend nodes associated with and service specific LPNs
- Light grey: Friend nodes which do not have a relationship with an LPN

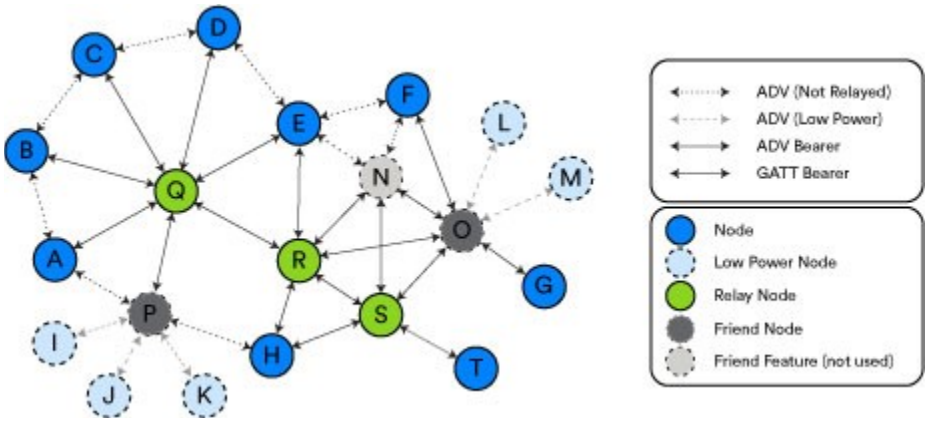

Figure 1 - example topology of a mesh network

Friend node **P** has a friendship relationship with LPNs **I**, **J** and **K.** Friend node **O** has a friendship relationship with LPN **L** and **M**. So, messages addressed to nodes **I**, **J** or **K** will be stored and forwarded by Friend P. Messages addressed to nodes **L** or **M** will be stored and forwarded by Friend **O**. Forwarding by the friend node only occurs when the LPN polls the friend for messages awaiting delivery.

Friendship Parameters

LPNs need to find Friend nodes and establish a friendship relationship with them. The procedure involved is called friend establishment. We'll examine this procedure shortly, but before we do let's introduce some of the key parameters governing the behavior of LPNs, since these are set during friend establishment.

1. *ReceiveDelay* is the time which elapses between the LPN, sending a request to the Friend node and it starting to listen for a response. This allows the Friend node time to prepare its response and send it back.

2. *ReceiveWindow* is the time that the LPN spends listening for a response. Figure 2 illustrates the timing sequence involving both ReceiveDelay and ReceiveWindow.

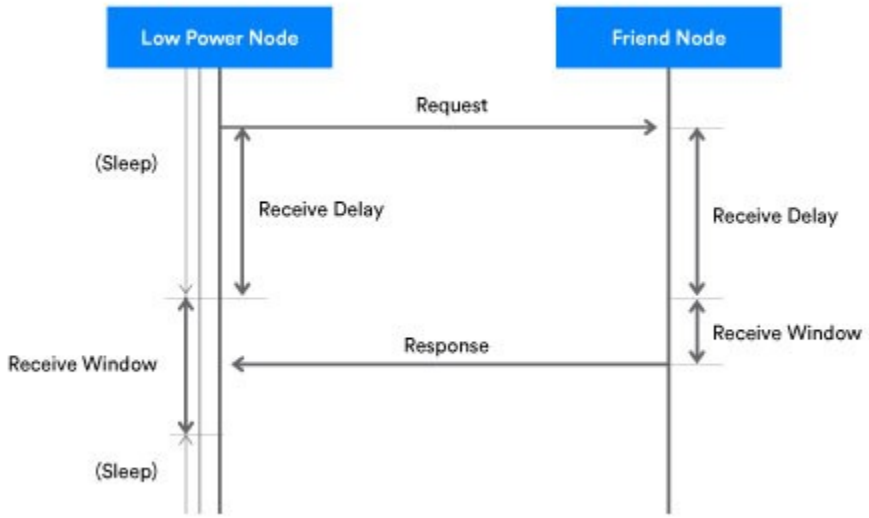

Figure 2 - ReceiveDelay and ReceiveWindow timing sequence

1. *PollTimeout* establishes a maximum time which may elapse between two consecutive requests sent by the LPN to its Friend node. If no requests from an LPN are received by the Friend node before the *PollTimeout* timer expires, then the friendship is terminated.

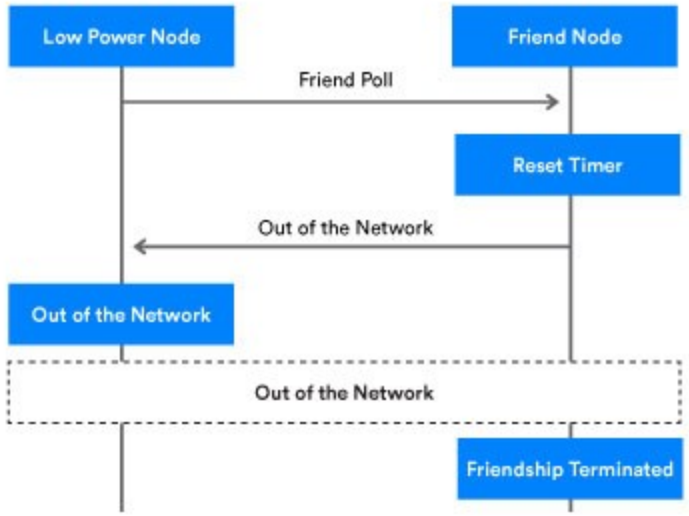

Figure 3 - PollTimeout timing sequence

Friend establishment

If two people want to build a friendship with each other, just one look can be enough! For friendship in a Bluetooth mesh network to be established, there are a few more steps to go through.

1. The LPN publishes a *Friend Request* message. This message is not relayed and therefore only Friend nodes in direct radio range process it. Nodes which do not have the Friend feature discard it. The Friend Request message includes the LPN's required *ReceiveDelay, ReceiveWindow* and *PollTimeout* parameters.

2. Each Friend node nearby which can support the requirements specified in the Friend Request message prepare and transmit a Friend Offer message back to the LPN. This message includes various parameters, including the *ReceiveWindow* size supported, message queue size available, subscription list size available and the RSSI value measured by the Friend node.

3. On receiving a Friend Offer message, the LPN selects a suitable Friend node by applying an implementation-specific algorithm. The algorithm is likely to consider various points. Some devices may place priority on the *ReceiveWindow* size to reduce power use as much as possible, while some may be more concerned about the RSSI value in order to ensure they can maintain a good quality link with the Friend node. The precise algorithm used is left to the product developer to determine.

4. After selecting a Friend node, the LPN sends a *Friend Poll* message to this Friend node.

5. After receiving a *Friend Poll* message from the LPN, the Friend node replies with a *Friend Update* message, which concludes the friend establishment procedure and provides security parameters. At this point, friendship has been established.

Friendship messaging

After friendship establishment, the Friend node stores all messages for the LPN in the Friend Queue. Collectively, these are known as stored messages. Figure 4 below illustrates the exchange of messages between a Friend node and associated LPN.

- When the Friend node receives a message addressed to the Friend node's LPN, the Friend node buffers this message, storing it in an area called the Friend Queue. In Figure 4, we can see that message 1 and 2 are stored in the Friend node on behalf of the LPN.

- Periodically, the LPN enables its transceiver and sends a *Friend Poll* to the Friend node, asking for any buffered messages stored for it.

- The Friend node begins by sending one stored message back to the LPN as a response to the *Friend Poll*.

- The LPN continues sending *Friend Poll* messages after each received message from the Friend node until it receives a *Friend Update* message with the MD (MD=More Data) field set to 0. This means there are no more messages buffered for the LPN. At this point, the LPN stops polling the Friend node.

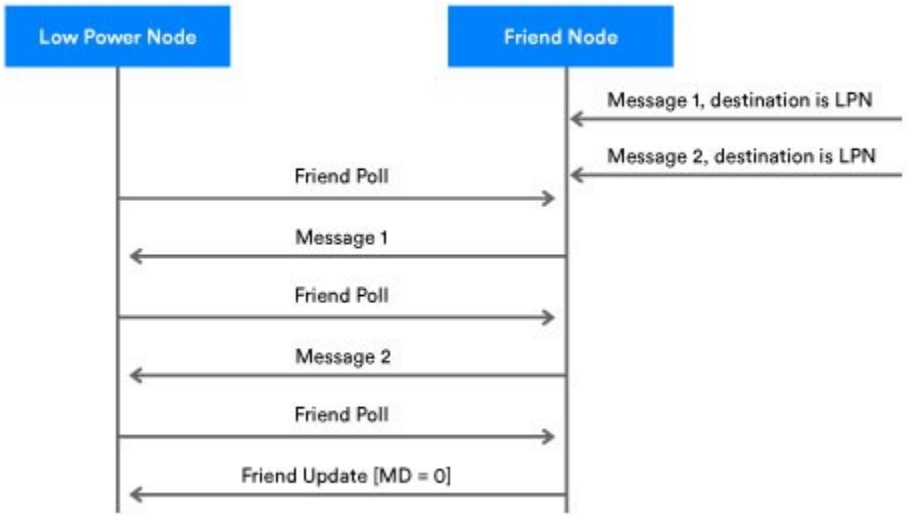

Figure 4 – Friendship messaging

Security

Security is everywhere in Bluetooth mesh. Friendship is no different and it uses two special security credentials:

- **Master security material**: Derived from the NetKey, it can also be used by the other nodes in the same network. Messages encrypted with the master security material can be decrypted by any node in the same network.
- **Friend security material**: Derived from the NetKey and some additional counter numbers which are generated by the LPN and the Friend node. Messages encrypted with the friend security material can only be decrypted by the Friend and LPNs which possess it.

How are the two security materials used by the LPN and Friend node? Here is a summary:

The corresponding friendship messages encrypted with the friend security material are:

- *Friend Poll*
- *Friend Update*
- *Friend Subscription List Add/Remove/Confirm*

- Stored messages that the Friend node delivers to the LPN

The corresponding friendship messages encrypted with the master security material are:

- *Friend Clear*
- *Friend Clear Confirm*

The messages sent from the LPN to the Friend node are encrypted with the master or friend security material depending on an application setting.

Friendship Termination

Friendship can be terminated in some conditions:

- If no *Friend Poll*, *Friend Subscription List Add* or *Friend Subscription List Remove* messages are
  received by the Friend node before the *PollTimeout* timer expires, the friendship is terminated.
- An LPN can initiate a friendship termination procedure by sending *Friend Clear* message to its Friend node, causing the friendship to be terminated by the Friend node.

Platform Selection Tips

Developers should consider the following guidelines when selecting platforms upon which to implement Friend and LPNs:

- **RAM capacity**. The amount of RAM available directly affects how many LPNs a Friend node can support and how many messages it can buffer for associated LPNs.
- **LPNs**. General power consumption performance of the selected MCU and module is key with LPNs. In addition, the wakeup/warmup time from sleep mode to running mode affects the responsiveness and latency on a LPN.

As a developer, I'm sure I share your keen anticipation of Bluetooth mesh SDKs becoming available. Then we can all get a taste of Bluetooth mesh "Friendship" together!

ON-DEMAND WEBINAR

What Makes Bluetooth Mesh so Disruptive?

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