AN1319: *Bluetooth*® Mesh Device Firmware Update

This application note provides background information on the Bluetooth Mesh Device Firmware Update feature. It describes the BLOB transfer, the DFU roles in a Bluetooth mesh network, the models required for these roles, and the firmware update process.

KEY POINTS

- Bluetooth Mesh Device Firmware Update roles
- Bluetooth Mesh Firmware Distribution models
- Bluetooth Mesh Firmware Update models
- Bluetooth Mesh BLOB Transfer models
- Bluetooth Mesh Firmware Update Process

1 Introduction

Bluetooth Mesh Device Firmware Update is a new feature introduced in the Bluetooth Mesh specification v1.1. The Bluetooth Mesh Device Firmware Update feature provides a standard way to update device firmware over the air. The feature enables the Bluetooth mesh network to check for the availability of device firmware updates, update multiple devices simultaneously for the same firmware, and track the progress of an update.

This document briefly describes the different types of nodes that are required to run device firmware updates in a mesh network. These nodes play important roles in managing update images for the devices that need a firmware update and distributing the update images to the nodes. This document also describes the procedures to perform a firmware update in a mesh network. To run the DFU examples provided in the Bluetooth Mesh SDK, see *AN1370: Bluetooth® Mesh Device Firmware Update Example Walkthrough*.

2 Bluetooth Mesh DFU and BLOB Transfer

Bluetooth Mesh Device Firmware Update is designed as a two-layer architecture to run different protocols. The DFU layer runs the DFU protocol for higher-level decision making that manages and coordinates firmware updates among different types of nodes. The BLOB Transfer layer runs the underlying data transport protocol handling generic large binary object transfer. In principle, the BLOB Transfer component could be used standalone for transferring any data objects that are much larger than the maximum Access Layer PDU size between the network devices.

2.1 BLOB Transfer

The Bluetooth Mesh BLOB Transfer (MBT) is a component on a node transferring or receiving binary large objects (BLOBs) over a Bluetooth Mesh network. The MBT client sends a BLOB to one or multiple nodes and the MBT server receives the BLOB. Bluetooth Mesh Device Firmware Update uses the MBT component to transfer update images.

The following figure illustrates the concept of how the MBT component transfers an update image. The MBT client breaks an update image into suitably-sized blocks based on the capabilities of the MBT servers and transfers these blocks to the servers. Each block is composed of identically sized chunks of data, except for the last chunk which may be smaller than the other chunks if the block's size is not an integer multiple of the chunk's size. A single message carries only a single chunk.

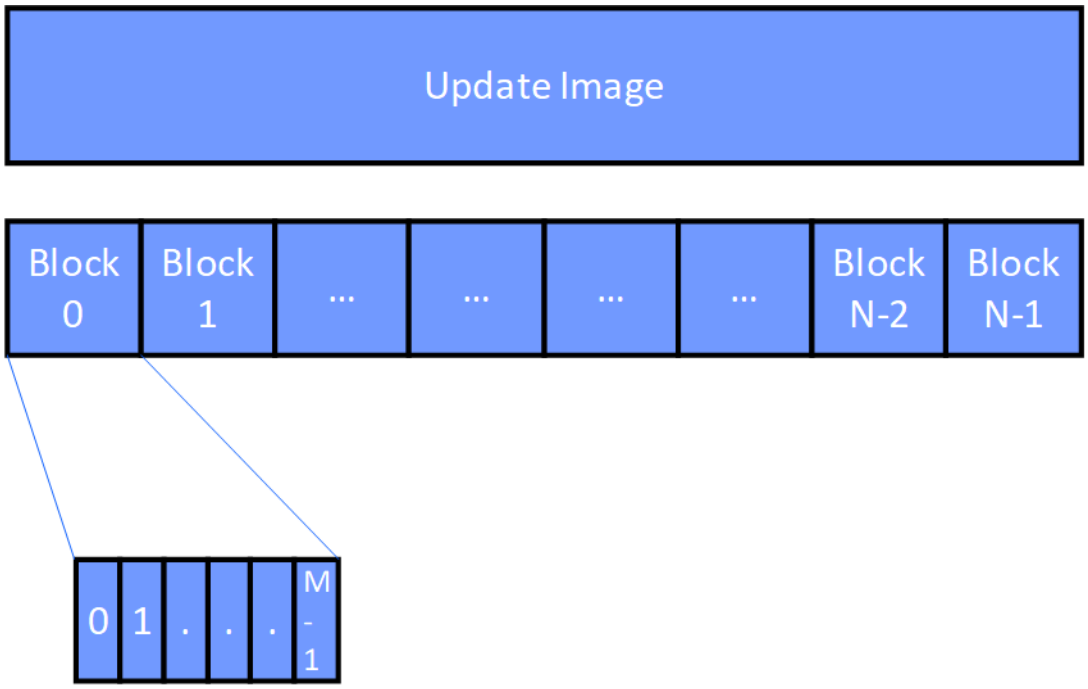

Figure 2-1. Bluetooth Mesh BLOB Transfer

The transfer may be performed in two modes: **Push BLOB Transfer mode** and **Pull BLOB Transfer mode**. The MBT client selects the transfer mode based on the capabilities supported by the servers. The MBT server may support both or only one of these modes.

In the Push BLOB Transfer mode, the client controls the flow of chunks to the servers, sending all the chunks from a block, then queries the servers to determine which chunks were received. All missing chunks are retransmitted.

In the Pull BLOB Transfer mode, the client transfers a BLOB typically to a single server at a time, although multiple simultaneous receivers can be supported as well. The server requests chunks from the client as it can process them, and the flow of chunks is controlled by the server.

A Low Power Node (LPN) typically only supports the Pull BLOB Transfer mode and therefore the Pull BLOB Transfer mode is typically used only if some of the MBT servers are Low Power Nodes. This mode makes it possible for the MBT server to throttle the transfer speed. This is necessary to ensure that no more chunks than the Friend node can store in its Friend Queue are transmitted at once.

The MBT client or server model is normally used as a transport in a higher-layer model such as the Firmware Update models.

2.2 Firmware Update Roles

Two to three roles participate in a firmware update over a Bluetooth mesh network: Initiator, Distributor, and Target Node, or Stand-alone Updater and Target Node. The Stand-alone Updater and Distributor as well as the Initiator can also be Target nodes. A firmware update is distributed simultaneously to the devices that have the same firmware. Sets of homogenous nodes are updated sequentially.

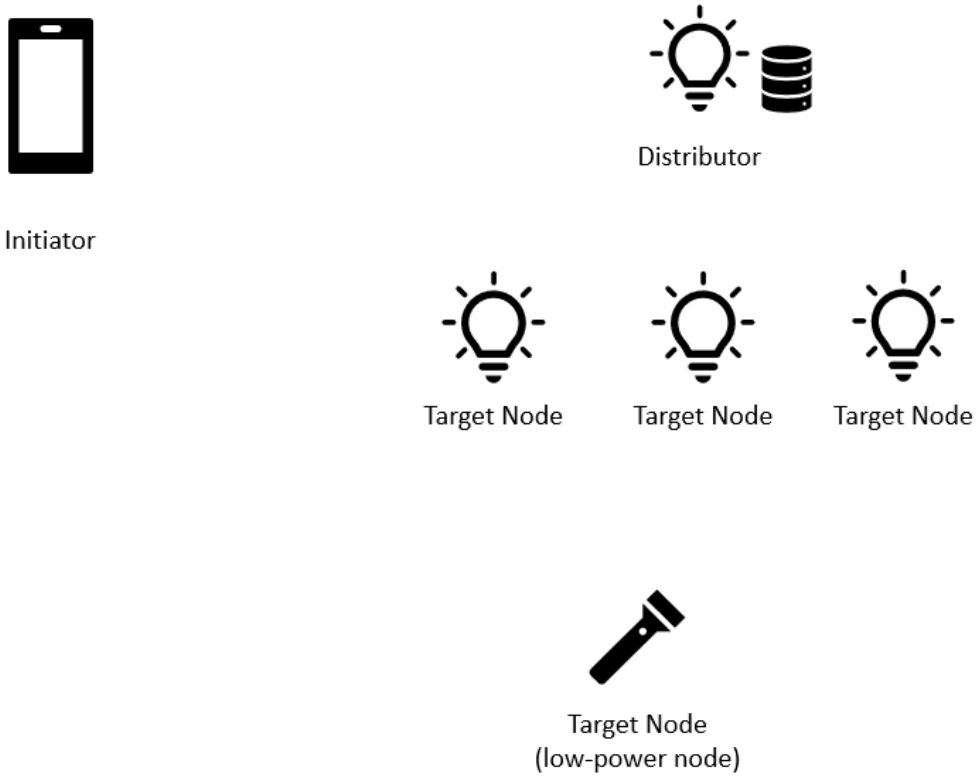

Figure 2-2. Firmware Update Roles

- **Initiator** role – checks for available updates for the firmware running on Target nodes that are included in a list provided by a higher-layer application and then sends the new firmware images to a Distributor. The procedures performed by an Initiator are controlled by a higher-layer application. An Initiator might be a smartphone or gateway device that periodically checks product websites for the availability of new firmware images.
- **Distributor** role – delivers new firmware images to the Target nodes and monitors the progress of the firmware update. The Distributor acts as an intermediary on behalf of the Initiator so that the Initiator does not always need to be present on the mesh network. The Distributor reports progress back to the Initiator when requested. The Distributor continues to have the configured functionality when distributing updates.
- **Stand-alone Updater** role – checks for available updates for the firmware running on Target nodes and delivers the new firmware images directly to the Target nodes. A Stand-alone Updater might be a smartphone or mesh gateway device that has access to the Internet while present on the mesh network to check for the availability of new firmware images for the Target nodes. When a new firmware image is available, it manages the delivery of the firmware image to the Target nodes without an intermediary Distributor.
- **Target Node** role – receives firmware updates and updates itself. The Target node stays in normal operation until a reboot with the new firmware.

2.3 Firmware Update Models

The Bluetooth Mesh Model specification v1.1 adds new models to support the Bluetooth Mesh Device Firmware Update feature. There are no changes in the Bluetooth Mesh Profile specification. The table below summarizes the models required by each role that participates in firmware updates. A node may support multiple roles. For example, a Distributor may support the Target node role.

Table 1. Mapping of Roles to Models Used in Device Firmware Updates

Role	Firmware Distribution Client	Firmware Distribution Server	Firmware Update Client	Firmware Update Server	BLOB Transfer Client	BLOB Transfer Server
Target node	–	–	–	M	–	M
Initiator	M	–	M	–	M	–
Distributor	–	M	M	–	M	M
Stand-alone Updater	C.1	C.2	M	–	M	C.2

M: Mandatory

C.1: Mandatory if Initiator role is supported, otherwise optional

C.2: Mandatory if Distributor role is supported, otherwise optional

- **Firmware Distribution Client** is the model used by the Initiator to send the firmware image and the firmware distribution parameters to the Distributor, and to start the firmware image transfer.
- **Firmware Distribution Server** is the model used by the Distributor to receive from the Initiator the firmware update parameters, the set of Target nodes to update, and the firmware image to transfer. This model can transfer one firmware image at a time.
- **Firmware Update Client** is the model used by the Distributor and Initiator to manage firmware updates. The Initiator uses this model to retrieve the information about the firmware subsystems installed on the Target node, and to get the download URIs of the new firmware images. The Distributor uses this model to start a firmware image transfer to the Target nodes.
- **Firmware Update Server** is the model used by the Target node to report the firmware images installed on the node and the download URI of new firmware images, and to initiate a firmware update to receive a new firmware image.
- **BLOB Transfer Client** is the model used to transfer BLOBs over a Bluetooth Mesh network. An MBT client can transfer a BLOB to one or more MBT servers, either unicasting or multicasting depending on the situation.
- **BLOB Transfer Server** is the model used to receive BLOBs over a Bluetooth mesh network from an MBT client.

3 Bluetooth Mesh Firmware Update Process

Multiple Target nodes can be updated either simultaneously using a multicast address or individually using unicast addresses. For efficiency, a multicast address should be used when multiple Target nodes indicate that they can accept the same firmware image. The Initiator does this by adding all Target nodes that support the same firmware image to the Target nodes list that is transferred to the Distributor. The Initiator transfers the firmware image to the Distributor and the Distributor distributes the firmware image to the nodes in the Target nodes list. Finally, the Target nodes apply the firmware image when the Distributor instructs them to do so. The Initiator can order the Distributor to either trigger applying the update immediately after the transfer is complete, or to wait for another message from the Initiator before applying the new firmware.

The Initiator optionally can transfer multiple firmware images to the Distributor, but the Distributor can distribute only one firmware image to Target nodes in a firmware update. This means that some nodes may not participate in the firmware update. The Initiator manages firmware images that are identified by the Firmware ID and determines what nodes are to be included in the Target nodes list.

The following figure illustrates an example of the case that only the nodes running the same firmware accept the firmware update. The mesh network has two sets of Target nodes running different firmware, one running Firmware X v1.0 (blue nodes: A, F, and C) and another running Firmware Y v1.0 (yellow nodes: B, E, H, I, and G). Both belong to the same network. The Distributor Node D is distributing Firmware X v2.0.

The blue nodes accepted the Firmware X v2.0. They therefore have subscribed to a multicast address and are simultaneously updating their firmware to version X v2.0. The yellow nodes are not participating in the firmware update because they did not accept the Firmware X v2.0 during the compatibility check and were not added to the Target nodes list by the Initiator.

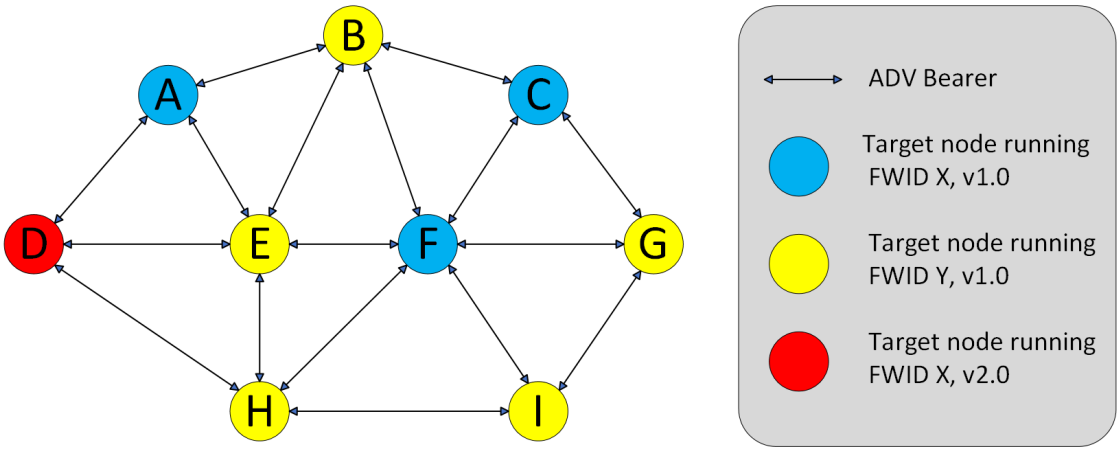

Figure 3-1. Distributor Updates Only Nodes that Accept the Firmware ID and Version

The following subsections briefly review the firmware update process.

3.1 Setting Up an Update

The first step is to set up an update. The Initiator polls the Target nodes for their update URIs and current firmware. The Initiator uses the information to get a newer firmware image and its metadata. The Initiator transfers a Target nodes list and an update image to the Distributor. The Initiator then starts the firmware distribution process. When the distribution is started, the Initiator specifies to the Distributor whether the update should be applied immediately after the transfer completes, or whether the update should be applied only after the Initiator sends a message to the Distributor to trigger applying the update. This mechanism can be used to defer rebooting the devices to a suitable time.

- The Initiator receives a list of Target nodes from the higher-layer application.
- When a new firmware image is available, the Initiator optionally may check whether nodes in the Target nodes list can accept the new firmware image. The nodes also inform the Initiator whether they will become unprovisioned or will have some changes to their composition data. The same information will be provided to the Distributor later on in the process.
- The Initiator optionally may check the information of the firmware images stored on a Distributor and can remove a firmware image from the Distributor at any time.
- The Initiator is not necessarily present on the mesh network after the distribution has started.
- The Initiator may check the status of the firmware image distribution via the Distributor at any time.

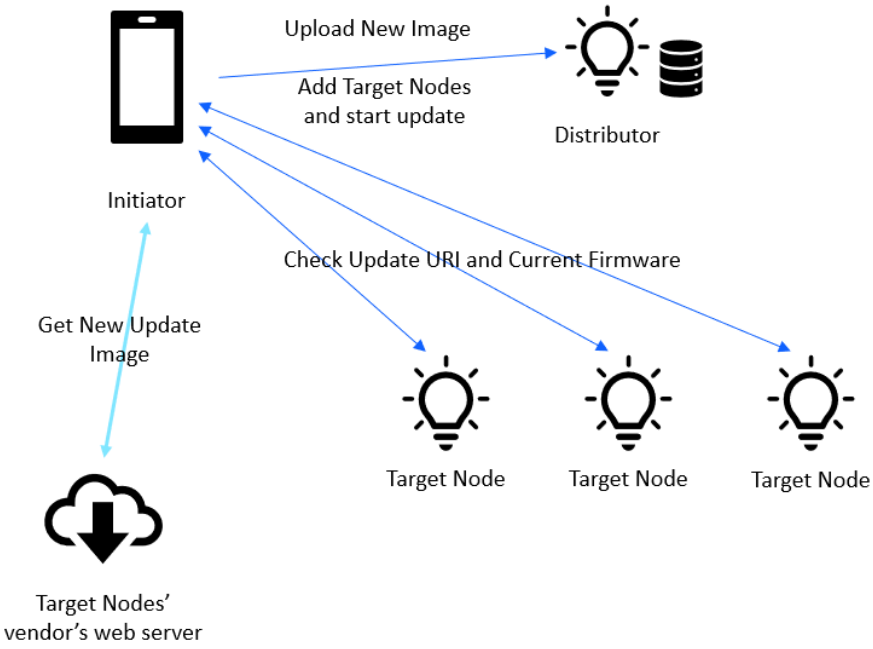

Figure 3-2. Firmware Update Process: Setting Up an Update

3.2 Starting an Update

The Distributor has received an update image, a Target nodes list, and the Distribution Start command from the Initiator. The Distributor notifies the Target nodes to prepare for an update and sends the metadata of the update image to the Target nodes. The Distributor then negotiates suitable BLOB Transfer parameters with the Target nodes and starts the BLOB Transfer.

- The Target nodes check the metadata and may reject the update.
- The Target nodes inform the Distributor whether they will become unprovisioned after the update or not, and whether they will have changes to their composition data, based on the metadata.
- Upon a request from the Initiator to retrieve the progress of the firmware update for each Target node, the Distributor provides a list of Target nodes and the progress of the firmware image transfer, the update phase of the transfer, and any potential errors.

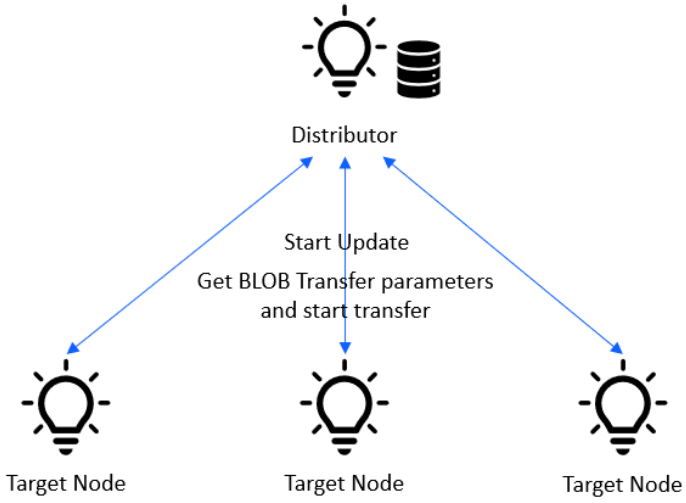

Figure 3-3. Firmware Update Process: Starting an Update

3.3 Sending the Update Image

The MBT client sends the update image to the MBT servers. Section 2.1 BLOB Transfer explains how an update image is transferred.

In the Push BLOB Transfer Mode, the Distributor transfers the chunks from a block to the Target nodes. After the Distributor has sent all chunks of a block, the Distributor queries the Target nodes to report missing chunks and transfers each missing chunk to the Target nodes, until no missing chunks are reported. The Distributor then moves on to next block, until all blocks are sent.

In the Pull BLOB Transfer Mode, the Distributor receives an initial list of chunks from the Target nodes and transfers the chunks from the initial list to the Target nodes. The Distributor then waits for the BLOB Partial Block Report from the Target nodes before transferring the next set of chunks. The Distributor repeats the step transferring the requested chunks to the Target nodes until all blocks are sent.

- The higher-layer application can cancel the BLOB Transfer at any time.

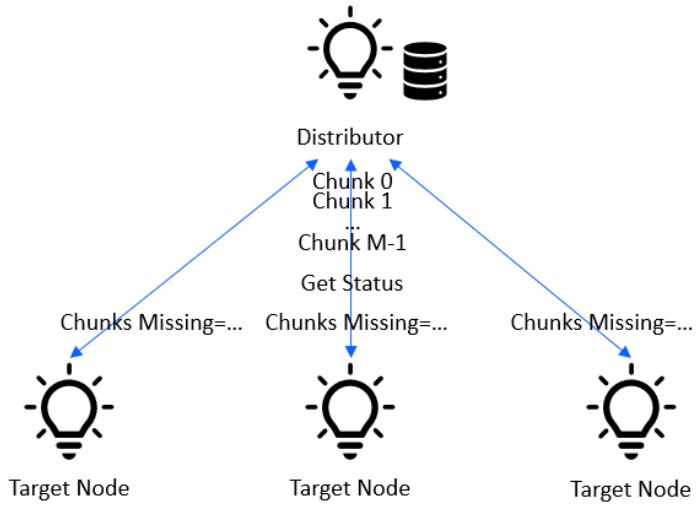

Figure 3-4. Firmware Update Process: Sending the Update Image

3.4 Applying an Update

The firmware image distribution is complete when the Target nodes have received the update image. The Target nodes check the integrity of the update image, and the Distributor polls them until all Target nodes complete. Then, if the Initiator had instructed the Distributor to start applying the update immediately in the Distribution Start message, the Distributor instructs the Target nodes to apply the update image. Otherwise, the Distributor waits for the applying signal from the Initiator before triggering applying the new firmware. The Target nodes reboot with the new firmware.

- The higher-layer application can cancel the verify firmware procedure at any time.
- The Target nodes can remain provisioned or become unprovisioned after an update image is applied. The states of the Target nodes are reported to the Distributor and Initiator.

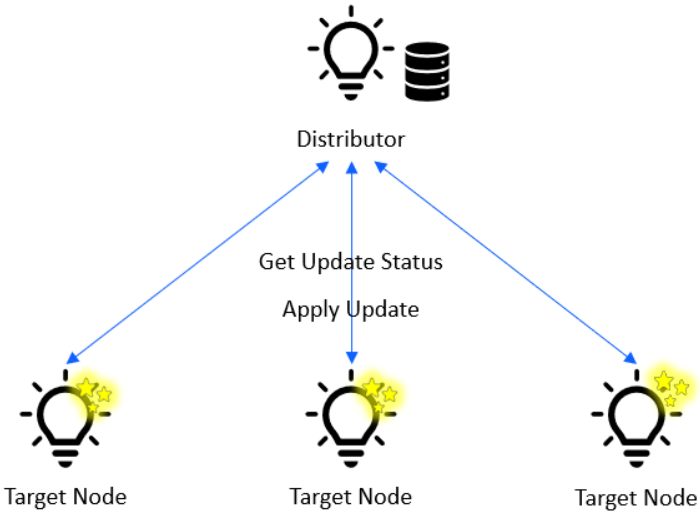

Figure 3-5. Firmware Update Process: Applying an Update

Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

www.silabs.com