Mesh Device Firmware Update Model

***Bluetooth*® Specification**

- **Version:** v1.0
- **Version Date:** 2023-09-12
- **Prepared By:** Mesh Working Group

Abstract:

The Mesh Device Firmware Update Model specification defines the requirements to enable a firmware upgrade of the Bluetooth mesh devices.

Version History

Version Number	Date	Comments
v1.0	2023-09-12	Adopted by the Bluetooth SIG Board of Directors.

Acknowledgments

Name	Company
Krzysztof Tkaczenko	Silvair, Inc.
Simon Slupik	Silvair, Inc.
Piotr Winiarczyk	Silvair, Inc.
Jori Rintahaka	Silicon Laboratories
Alok Mittal	STMicroelectronics
Omkar Kulkarni	Nordic Semiconductor ASA
Trond Einar Snekvik	Nordic Semiconductor ASA
Max Palumbo	Silicon Laboratories
Michał Budzoń	Silvair, Inc.
Max Palumbo	Katerra Inc.
Victor Zhodzishsky	Cypress Semiconductor Corporation

Contents

1 Introduction

This Mesh Device Firmware Update Model specification defines models (along with their required states and messages) that are used to provide device firmware upgrade functionality for the nodes that are part of a mesh network.

Terms, acronyms, and abbreviations that have a specific meaning in the context of this specification or the Bluetooth environment in general are defined or expanded upon on their first use. Defined terms that are used in this specification are listed in Section 1.6.

1.1 Language

1.1.1 Language conventions

In the development of a specification, the Bluetooth SIG has established the following conventions for use of the terms *"shall"*, ***"shall not"***, *"should"*, ***"should not"***, *"may"*, *"must"*, and *"can"*. In this Bluetooth specification, the terms in Table 1.1 have the specific meanings given in that table, irrespective of other meanings that exist.

Term	Definition
shall	—used to express what is required by the specification and is to be implemented exactly as written without deviation
shall not	—used to express what is forbidden by the specification
should	—used to express what is recommended by the specification without forbidding anything
should not	—used to indicate that something is discouraged but not forbidden by the specification
may	—used to indicate something that is permissible within the limits of the specification
must	—used to indicate either: 1. an indisputable statement of fact that is always true regardless of the circumstances 2. an implication or natural consequence if a separately-stated requirement is followed
can	—used to express a statement of possibility or capability

Table 1.1: Language conventions terms and definitions

1.1.1.1 Implementation alternatives

When specification content indicates that there are multiple alternatives to satisfy specification requirements, if one alternative is explained or illustrated in an example it is not intended to limit other alternatives that the specification requirements permit.

1.1.1.2 Discrepancies

It is the goal of Bluetooth SIG that specifications are clear, unambiguous, and do not contain discrepancies. However, members can report any perceived discrepancy by filing an erratum and can request a test case waiver as appropriate.

1.1.2 Reserved for Future Use

Where a field in a packet, Protocol Data Unit (PDU), or other data structure is described as "Reserved for Future Use" (irrespective of whether in uppercase or lowercase), the device creating the structure

shall set its value to zero unless otherwise specified. Any device receiving or interpreting the structure shall ignore that field; in particular, it shall not reject the structure because of the value of the field.

Where a field, parameter, or other variable object can take a range of values, and some values are described as "Reserved for Future Use," a device sending the object shall not set the object to those values. A device receiving an object with such a value should reject it, and any data structure containing it, as being erroneous; however, this does not apply in a context where the object is described as being ignored or it is specified to ignore unrecognized values.

When a field value is a bit field, unassigned bits can be marked as Reserved for Future Use and shall be set to 0. Implementations that receive a message that contains a Reserved for Future Use bit that is set to 1 shall process the message as if that bit was set to 0, except where specified otherwise.

The acronym RFU is equivalent to Reserved for Future Use.

1.1.3 Prohibited

When a field value is an enumeration, unassigned values can be marked as “Prohibited.” These values shall never be used by an implementation, and any message received that includes a Prohibited value shall be ignored and shall not be processed and shall not be responded to.

Where a field, parameter, or other variable object can take a range of values, and some values are described as “Prohibited,” devices shall not set the object to any of those Prohibited values. A device receiving an object with such a value should reject it, and any data structure containing it, as being erroneous.

“Prohibited” is never abbreviated.

1.2 Table conventions

This section describes conventions regarding the following:

- Requirements that are in a table
- Indicating a cell that has no value or content

1.2.1 Requirements that are in a table

Unless otherwise stated, requirements that are in a table in this specification can be defined as "Mandatory" (M), "Optional" (O), "Excluded" (X), or "Conditional" (C.n). Conditional statements (C.n) are listed directly below the table in which they appear.

1.2.2 Indicating a cell that has no value or content

Where a cell in a table indicates an intended absence of a value or other content, the cell will contain “none” or a hyphen (i.e., a “minus” sign). Examples of this are:

- In the “condition” column of the description of a finite state machine where a rule is unconditional
- In the “action” column of the description of a finite state machine where a rule has no action
- In a “restrictions” column where there are no applicable restrictions
- In an interface description where there are no parameters of a specific type

An empty cell in a table indicates that there is no useful information that can be placed in that cell. Examples of this are:

- In a “comments” column when there is no comment to make in a particular row
- In a column specifying properties when the relevant item is reserved for future use (and therefore does not have any properties)

- In a “units” column when the relevant item is unitless

1.3 Error handling

When a model processes an incoming message, the defined behavior may require the model to check one or more error conditions from a table of conditions (for example, see Table 6.4). These error conditions shall be checked in order, from top to bottom, starting from the first row of the table. The model shall stop checking after encountering a failed condition. Or, after not encountering a failed condition, the model shall stop checking after finishing the last row in the table.

1.4 Bluetooth specification release compatibility

This specification is compatible with Mesh Protocol v1.1 [1].

1.5 Conformance

Each capability of this specification shall be supported in the specified manner. This specification may provide options for design flexibility, because, for example, some products do not implement every portion of the specification. For each implementation option that is supported, it shall be supported as specified.

1.6 Terminology

Table 1.2 lists all of the defined terms used in this specification.

Term	Definition Location
Distributor	Section 2.1
firmware metadata	Section 2
firmware subsystem	Section 2
Initiator	Section 2.1
Standalone Updater	Section 2.1
Target	Section 2.1

Table 1.2: Terminology

2 Firmware update introduction

This specification defines requirements to enable updates of the firmware running on a node or its subsystems for use with Mesh Protocol v1.1 [1]. The Firmware Update models defined in this specification use the BLOB Transfer models defined in [10] for transport of the firmware images.

A firmware subsystem on a node is a binary image that can be updated independently of other firmware subsystems on the node. The firmware subsystems on the node should be partitioned so that each firmware image is standalone and can be installed and run after the firmware update for that subsystem is complete. For example, the networking stack and the application of a node may be packaged as separate binary images that can be downloaded separately and updated independently. If a firmware subsystem has any system dependencies, such as those related to installation of the subsystem, the system dependencies can be indicated in the firmware metadata.

The firmware metadata is vendor-specific binary data that is used to determine whether the incoming firmware image can be accepted by a node. For example, if the incoming firmware image requires installation of new versions of firmware for other subsystems on the node, the order of the subsystem firmware image installations may be identified in the vendor-specific metadata.

A firmware update requires the interaction of several models across several nodes in the network and, as a result, requires significant configuration of each node. The application keys and multicast addresses need to be bound to all the relevant models by the Mesh Manager, which provides the list of nodes in the network that support the Firmware Update models after the models are configured. The Mesh Manager need not participate in a firmware update.

2.1 Firmware update roles

Four roles participate in a firmware update over a Bluetooth mesh network:

- **Target role.** A Target role on a node identifies the firmware image running on the node or its subsystems and receives the firmware updates.
- **Initiator role.** An Initiator 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.** A Distributor delivers new firmware images to the Target nodes and monitors the progress of the firmware update. Firmware image delivery is performed using the BLOB transfer models (see Section 5 and Section 6 in [10]). 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.
- **Standalone Updater role.** A Standalone Updater checks for available updates for the firmware running on Target nodes and delivers the new firmware images directly to the Target nodes. The Standalone Updater need not support the Firmware Distribution Server model or BLOB Transfer Server model in order to receive a firmware image from an Initiator. For example, a mesh gateway device or a smartphone that can access the Internet while present on the mesh network might check for new firmware images available for the Target nodes and then, when a new firmware image is available, manage the delivery of the firmware image to the Target nodes without an intermediary Distributor.

Figure 2.1 shows an example of a mesh topology with an Initiator, a Distributor, and Target nodes. The Initiator (node I) uses a GATT Bearer to upload a new firmware image to a Distributor (node D). The Distributor transfers the firmware image to the Target nodes (nodes B, E, H, J, and G).

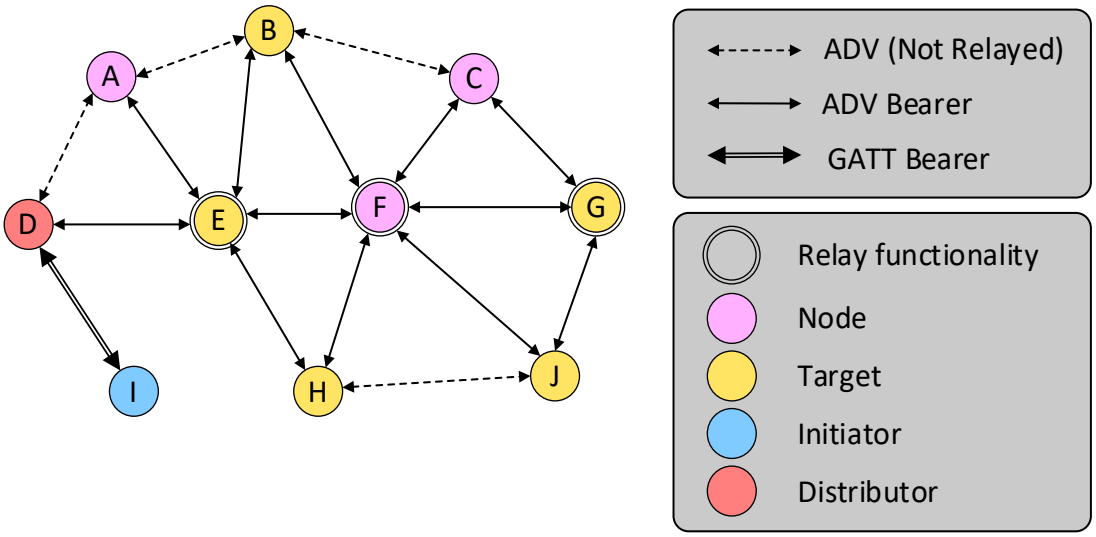

Figure 2.1: Example topology of a mesh network with Initiator, Distributor, and Target nodes

2.1.1 Models and model functionalities required for each firmware update role

Table 2.1 specifies the models required by each role that participates in a firmware update. Each role shall be implemented on its own element and shall not share the element with other DFU roles.

Role	Firmware Distribution Client	Firmware Distribution Server	Firmware Update Client	Firmware Update Server	BLOB Transfer Client	BLOB Transfer Server
Target	–	–	–	M	–	M
Initiator	M	–	M	–	M	–
Distributor	–	M	M	–	M	M
Standalone Updater	–	–	M	–	M	–

Table 2.1: Mapping of roles to models used in device firmware updates

A node may support multiple roles. For example, a Distributor may support the Target role. Table 2.2 shows examples of combined roles with X and Y representing different element indices.

	Element Indices for Model instances					
Device Role	Firmware Distribution Client	Firmware Distribution Server	Firmware Update Client	Firmware Update Server	BLOB Transfer Client	BLOB Transfer Server
Initiator and Target node	X	–	X	Y	X	Y
Distributor and Target node	–	X	X	Y	X	X, Y
Standalone Updater and Target node	–	–	X	Y	X	Y

Table 2.2: Element indices for combined DFU roles on a node

The Standalone Updater is not referred to in any of the requirements. For any message exchange between an Initiator and a Distributor (i.e., between a Firmware Distribution Client and Firmware Distribution Server models, and their corresponding BLOB transfer client and BLOB transfer server

models), a Standalone Updater exchanges these messages over an implementation-specific local interface. That is, from the perspective of the rest of the network, a Standalone Updater appears as an Initiator and a Distributor combined into one node.

A Standalone Updater may also support the Firmware Distribution models message exchange over the network interface. If a Standalone Updater can function as a Distributor for some other Initiator node in the network, then the Firmware Distribution Server model and BLOB Transfer Server model are supported. If a Standalone Updater can control some other Distributor node, then the Firmware Distribution Client model and BLOB Transfer Client model are supported.

Table 2.3 specifies the mandatory and optional models and procedures applicable for each role.

Model Features and Role Support	Target	Initiator	Distributor	Standalone Updater
BLOB Transfer Server	M	–	M	–
- Initialize And Receive BLOB procedure	M	–	M	–
- Get BLOB Reception Progress procedure	M	–	M	–
- Get BLOB Reception Progress procedure	M	–	M	–
- Push BLOB procedure	C.1	–	M	–
- Pull BLOB State procedure	C.2	–	M	–
BLOB Transfer Client	–	–	M	M
- Calculate Client Timeout procedure	–	–	M	M
- Retrieve Capabilities procedure	–	–	M	M
- Transfer BLOB procedure	–	–	M	M
- Send Block sub-procedure	–	–	M	M
- Send Data sub-procedure	–	–	M	M
- Determine Block Status sub-procedure	–	–	M	M
- Determine Transfer Status procedure	–	–	M	M
- Cancel Transfer procedure	–	–	M	M
- Get BLOB Receiver procedure	–	–	M	M
- Get Active BLOB Receivers procedure	–	–	M	M
- Get Transfer Progress procedure	–	–	M	M
Firmware Update Server	M	–	–	–
- Receive Firmware procedure	M	–	–	–
- Verify Firmware procedure	M	–	–	–
- Apply New Firmware procedure	M	–	–	–
Firmware Update Client		M	M	M
- Calculate Client Timeout procedure	–	–	M	M
- Retrieve Firmware Information procedure	–	M	M	M
- Firmware Compatibility Check procedure	–	M	M	M
- Start Firmware Update On Target Nodes procedure	–	–	M	M

Model Features and Role Support	Target	Initiator	Distributor	Standalone Updater
- Apply Firmware On Target Nodes procedure	–	–	M	M
- Cancel Firmware Update procedure	–	–	M	M
- Refresh Target Nodes Status procedure	–	–	M	M
- Confirm Update On Target Nodes procedure	–	–	M	M
- Get Update Receiver procedure	–	–	M	M
- Get Active Update Receivers procedure	–	–	M	M
Firmware Distribution Server	–	–	M	–
- Store Firmware procedure	–	–	M	–
- Store Firmware OOB procedure	–	–	C.3	–
- Firmware Retrieval Over HTTPS procedure	–	–	C.3	–
- Distribute Firmware procedure	–	–	M	–
Firmware Distribution Client	–	M	–	–
- Check For Current Firmware procedure	–	M	–	–
- Check For Updated Firmware procedure	–	M	–	–
- Retrieve Updated Firmware procedure	–	M	–	–
- Populate Distribution Receivers List procedure	–	M	–	–
- Distributor Capabilities procedure	–	M	–	–
- Initiate Distribution procedure	–	M	–	–
- Get Distribution Progress procedure	–	M	–	–
- Upload Firmware OOB procedure	–	O	–	–
- Upload Firmware procedure	–	M	–	–
- Firmware Images List Management procedure	–	M	–	–
- Firmware Retrieval Over HTTPS procedure	–	M	–	–
- Firmware Check Over HTTPS procedure	–	M	–	–

Table 2.3: Model features and role matrix

C.1: Mandatory if Low Power feature is not supported; otherwise optional.

C.2: Mandatory if Low Power feature is supported; otherwise optional.

C.3: At least one of the procedures shall be supported.

2.2 Security considerations

Security is critical during a mesh firmware update because the firmware update feature provides an interface for loading new firmware into a node to be executed.

The Mesh Protocol specification v1.1 [1] defines encryption and authentication of messages up to the access layer. In addition, the firmware image can be encrypted during transport by the higher-layer application. The method of encryption is vendor specific.

The Target node should use an authentication mechanism to check that the firmware image being transferred is from an authorized source. The choice of authentication mechanism is vendor specific. The authentication mechanism should follow security best practices such as those outlined in NIST SP 800-193 Section 3.5 [3].

To enable firmware updates to take place, each node on the network is configured with a set of application keys (AppKey) and multicast addresses to use for the Firmware Update models and the BLOB transfer models. All nodes that communicate with one another are provided the same set of security credentials and addresses. The Mesh Manager determines how the Firmware Update models on the network are configured.

2.3 Firmware update process overview

During the firmware update, 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. This is done by the Initiator (see Section 2.3.2) by adding all Target nodes that support the same firmware image to the Target nodes list that is transferred to the Distributor.

The mesh network shown in Figure 2.2 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 sets of nodes belong to the same network. Node D is distributing Firmware X v2.0.

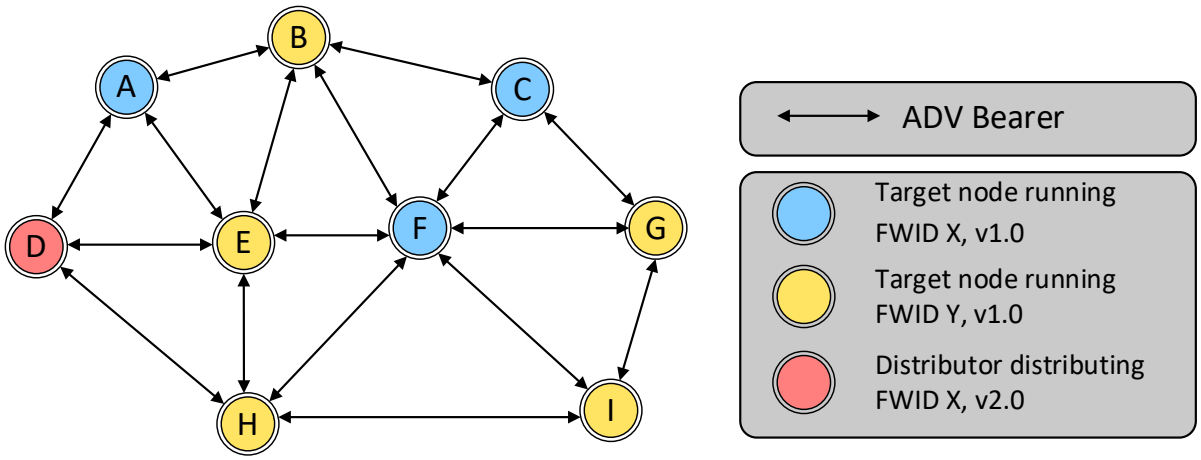

Figure 2.2: Example of a network with nodes running two different firmware images, and a Distributor updating FWID X to v2.0

Before initializing the Initiator, the Mesh Manager subscribes the models of the Target nodes running Firmware X v1.0 to a multicast address. This enables all Target nodes that are running Firmware X v1.0 to simultaneously update their firmware to version X v2.0. The yellow nodes do not participate in the firmware update because they did not accept the new firmware image during the compatibility check and were not added to the Target nodes list by the Initiator.

Process overview. A firmware update consists of the following steps:

1. The Initiator is initialized by the higher-layer application (see Section 2.3.1).
2. The Initiator checks the firmware running on the Target nodes, and checks for new firmware images available for the Target nodes (see Section 2.3.2).
3. If a new firmware image is available, the Initiator checks which Target nodes are compatible with the new firmware image (see Section 2.3.3).
4. The Initiator initializes the Distributor (see Section 2.3.4):
   - The Initiator transfers the list of Target nodes to the Distributor (see Section 2.3.4.1).
   - The Initiator transfers the firmware image to the Distributor (see Section 2.3.4.2).
   - The Initiator manages the list of firmware images on the Distributor as needed (see Section 2.3.4.3).
5. The Distributor begins firmware image distribution to the Target nodes (see Section 2.3.5).
6. The Initiator and Distributor may check the status of the firmware image distribution at any time, because this is a long-running process (see Section 2.3.6).
7. When firmware image distribution is complete the Target nodes verify the firmware image (see Section 2.3.7).
8. The new firmware image is applied to the Target nodes, and the Target nodes begin using the new firmware image (see Section 2.3.8).

2.3.1 Initiator initialization

The Initiator receives a list of Target nodes, a time to live (TTL) value, an AppKey Index, and a multicast address that Target nodes are subscribed to from the higher-layer application. For each Target node, the Initiator then retrieves firmware update information, such as the firmware version installed on the node, the location from which a new firmware image can be downloaded, and the availability of new firmware images.

2.3.2 Check for an available firmware update

The Initiator retrieves information about the firmware image installed on a Target node using the Check For Current Firmware procedure (see Section 7.2.2.1).

The Initiator then checks for new firmware images using the Check For Updated Firmware procedure (see Section 7.2.2.2). The Check For Updated Firmware procedure retrieves a Uniform Resource Identifier (URI) that is used to locate information about the most up-to-date firmware available for the Target node. The Initiator can download the new firmware image using an out-of-band (OOB) mechanism, such as an Internet connection. When the Initiator supports the https URI scheme, it uses the Firmware Retrieval Over HTTPS procedure (see Section 3.3) to download the new firmware image.

2.3.3 Firmware compatibility check on Target nodes

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 using the Firmware Compatibility Check procedure (see Section 7.1.2.4). This is an optional step to filter out Target nodes that cannot accept the new firmware image.

2.3.4 Distributor Initialization

The Initiator sends the list of Target nodes to the Distributor, and the Initiator either transfers the image to the Distributor or it provides instructions for how to retrieve the image and starts the OOB retrieval.

2.3.4.1 Transferring the Target nodes list to a Distributor

The Initiator optionally performs a compatibility check to determine whether the Distributor can store the entire list of Target nodes using the Distributor Capabilities procedure (see Section 7.2.2.5). In response, the Initiator gets the Distributor capabilities, which include the maximum number of Target nodes the Distributor can handle during a single firmware update. The Initiator optionally caches the Distributor capabilities, which do not change unless the firmware running on the Distributor is updated. The Initiator then transfers the list of Target nodes to the Distributor using the Populate Distribution Receivers List procedure (see Section 7.2.2.4).

2.3.4.2 Transferring a firmware image to a Distributor

The Initiator either uses the Upload Firmware procedure (see Section 7.2.2.9) to transfer the firmware image to the Distributor, or the Initiator uses the Upload Firmware OOB procedure (see Section 7.2.2.8) to instruct the Distributor to retrieve the firmware image using an OOB mechanism.

If the Initiator transfers the firmware image to the Distributor, then the Distributor uses the Store Firmware procedure (see Section 6.2.2.1) to receive the new firmware image.

If the Initiator triggers the Distributor to start firmware image retrieval, then the Distributor uses the Store Firmware OOB procedure (see Section 6.2.2.3) to receive the new firmware image. If the Distributor supports the https URI scheme, then the Distributor uses the Firmware Retrieval over HTTPS procedure (see Section 3.3). If the Distributor does not support this URI scheme, then firmware retrieval is performed using an implementation-specific OOB mechanism.

2.3.4.3 Firmware image management on a Distributor

A Distributor has finite resources for storing firmware images and firmware metadata. The Initiator optionally checks the distributor's capabilities using the Distributor Capabilities procedure. To manage firmware images stored on a Distributor, the Initiator uses the Firmware Images List Management procedure (see Section 7.2.2.10). The Initiator can remove a firmware image from the Distributor at any time.

2.3.5 Starting firmware image distribution to the Target nodes

To begin firmware image distribution from the Distributor to the Target nodes, the Initiator uses the Initiate Distribution procedure (see Section 7.2.2.6), which triggers the Distributor to start the Distribute Firmware procedure (see Section 6.2.2.4).

The Distributor uses the Start Firmware Update On Target Nodes procedure (see Section 7.1.2.5) to initiate the firmware image distribution. The Distributor then uses the Transfer BLOB procedure (see Section 6.2.3 in [10]) from the BLOB Transfer Client model to transfer the firmware image to each Target node. When the Transfer BLOB procedure completes successfully, the Target nodes automatically begin the Verify Firmware procedure (see Section 6.1.2.2); the Verify Firmware procedure is started with no external prompt from the Initiator or Distributor.

2.3.6 Checking the progress of a firmware image transfer

To retrieve the progress of the firmware update for each Target node at any time during an image transfer, the Initiator uses the Get Distribution Progress procedure (see Section 7.2.2.7). The Get Distribution Progress procedure provides a list of Target nodes and the progress of the firmware image transfer, the update phase of the transfer, and any potential errors.

2.3.7 Verification of downloaded firmware

To verify the integrity of the complete firmware image (the reconstituted chunks and blocks of a BLOB after the Transfer BLOB procedure completes), the Target node can use any implementation- specific verification method. This check is used to verify the integrity and authenticity of the firmware image transferred, and the Target node discards any image that fails this verification check.

The Distributor uses the Refresh Target Nodes Status procedure (see Section 7.1.2.8) to determine whether the firmware image was verified successfully on each Target node.

2.3.8 Applying the firmware image to the Target nodes

The Distributor applies the new firmware image to the Target nodes after verification completes. The Distributor uses the Apply Firmware On Target Nodes procedure (see Section 7.1.2.6) to instruct the Target nodes to apply the new firmware image.

The Distributor checks whether the new firmware image was applied successfully and records the result for each Target node using the Refresh Target Nodes Status procedure (see Section 7.1.2.8).

The Distributor either applies the firmware image immediately after verification is complete or waits for a prompt from the Initiator to apply the new firmware image. The update policy, which is sent to the Distributor at the start of firmware image distribution, determines when the firmware image is applied.

2.3.8.1 Update policies

There are two policies that determine when the Distributor triggers installation of the new firmware image on the Target nodes (see Section 4.3.3.2):

- The Distributor triggers installation of a firmware image immediately after the Target node successfully verifies the image.
- The Distributor waits for a prompt from the Initiator before it triggers installation of a verified firmware image on the Target nodes.

2.4 Interactions among models during a firmware update

Figure 2.3 summarizes the interactions and information exchange between all models that participate in a firmware update.

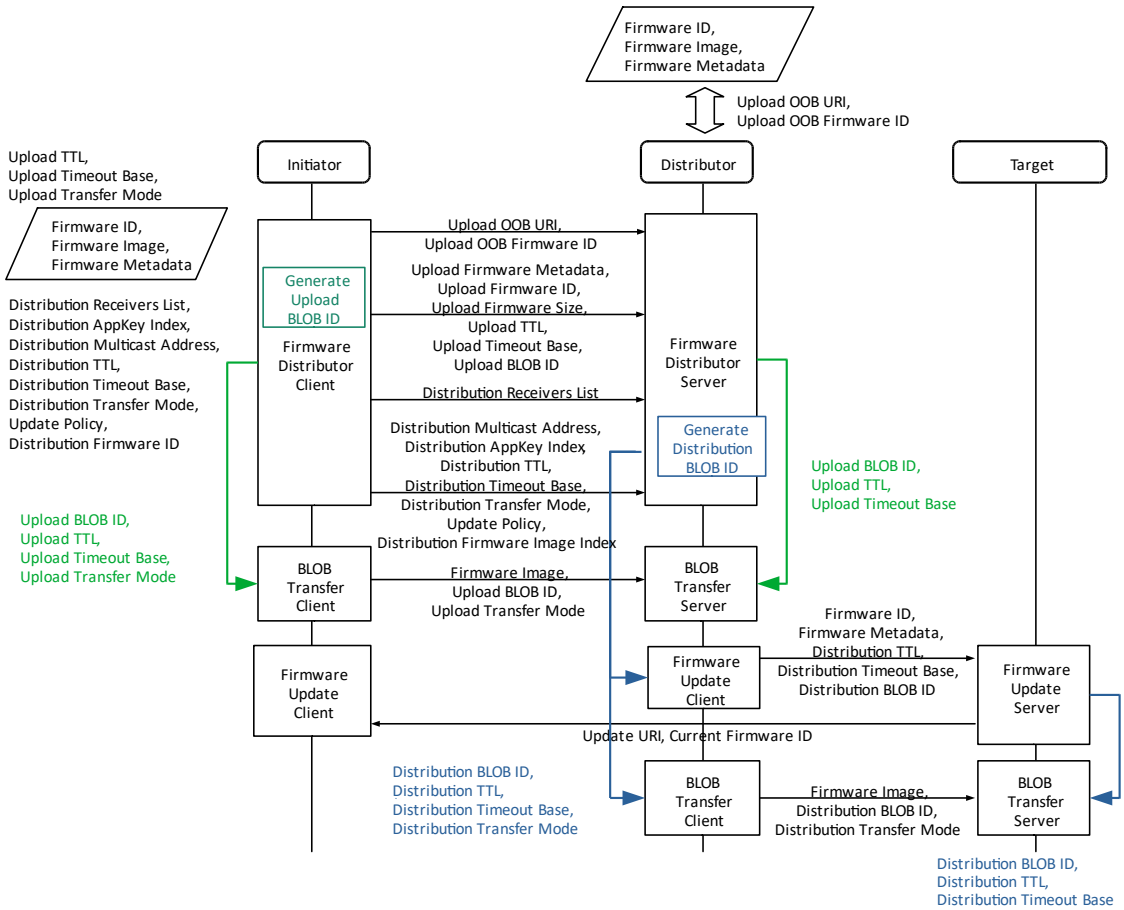

Figure 2.3: Firmware update roles, models, and information flow overview

3 Out-of-band firmware retrieval

This section defines the recommended firmware archive file format and the procedures used to retrieve a new firmware image from an HTTP server OOB.

3.1 Firmware archive file format

The new firmware image, with accompanying metadata, should be stored in a firmware archive file. The firmware archive file shall use the structure defined in Table 3.1. The firmware archive file uses the gzip archive file format [5].

Item	Description
Manifest File	A file that contains a description of the firmware package, including the name of the Firmware Image File and an optional Metadata File for the update
Firmware Image File	The file containing the firmware image, as identified in the Manifest File
Metadata File	An optional file, provided by the vendor and identified in the Manifest File, that may provide metadata for the firmware image

Table 3.1: Firmware archive file structure

3.1.1 Manifest File

The Manifest File item shall be present in the firmware archive file. The Manifest File shall be named manifest.json. The format of the Manifest File shall be JavaScript Object Notation (JSON) [6]. The Manifest File shall be in the format defined in Section 3.1.1.1. This JSON schema uses the format defined in JSON Schema Draft 7 [7].

The manifest/firmware/firmware_image_file member of the manifest.json file contains the Firmware Image File name. If present, the manifest/firmware/metadata_file member contains the Metadata File name.

The manifest/firmware/firmware_id member shall contain the Firmware ID of the firmware stored in the firmware archive file identified by the manifest/firmware/firmware_image_file member. The format of the manifest/firmware/firmware_id member shall be Base16, as defined in [4].

3.1.1.1 Manifest file JSON schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://www.bluetooth.com/specifications/specs/mesh-dfu-1-0-firmware-archive-schema.json#",
  "type": "object",
  "title": "The Mesh Firmware Package Schema",
  "description": "This JSON schema is part of the Mesh Device Firmware Update Model Bluetooth specification and may be made available separately for the convenience of members of Bluetooth SIG. Use of this JSON schema by members of Bluetooth SIG is governed by the membership and other related agreements between Bluetooth SIG and its members, including those agreements posted on Bluetooth SIG's website located at www.bluetooth.com. Use of this JSON schema by anyone who is not a member of Bluetooth SIG is prohibited and is an infringement of the intellectual property rights of Bluetooth SIG and its members.",
  "required": [
    "manifest"
  ],
  "properties": {
    "manifest": {
      "$id": "#/properties/manifest",
      "type": "object",
```

```
      "required": [
        "firmware"
      ],
      "properties": {
        "firmware": {
          "$id": "#/properties/manifest/properties/firmware",
          "type": "object",
          "required": [
            "firmware_image_file",
            "firmware_id"
          ],
          "properties": {
            "firmware_image_file": {
              "$id":
"#/properties/manifest/properties/firmware/properties/firmware_image_file",
              "type": "string",
              "examples": [
                "firmware.bin"
              ],
              "pattern": "^(.*)$"
            },
            "metadata_file": {
              "$id":
"#/properties/manifest/properties/firmware/properties/metadata_file",
              "type": "string",
              "examples": [
                "metadata.bin"
              ],
              "pattern": "^(.*)$"
            },
            "firmware_id": {
              "$id":
"#/properties/manifest/properties/firmware/properties/firmware_id",
              "type": "string",
              "examples": [
                "010246573A312E332E35"
              ],
              "pattern": "^(?:[0-9A-F]{2}){2,108}$"
            }
          }
        }
      }
    }
  }
}
```

3.1.2 Firmware Image File

The Firmware Image File item shall be present in the firmware archive file and shall have the file name defined in the manifest/firmware/firmware_image_file member of the Manifest File.

3.1.3 Metadata File

The Metadata File item shall be present in the firmware archive file only when the manifest/firmware/metadata_file member is present in manifest.json. If the Metadata File member is

present, then the metadata file shall have the file name defined in the manifest/firmware/manifest_file member of the Manifest File.

3.1.4 Example of a firmware archive file

An example of the structure of a firmware archive file is presented below:

```
   Date      Time     File Name
------------------- --------------
2019-12-24 05:53:00 manifest.json
2019-12-24 16:11:56 firmware.bin
2018-12-24 07:54:20 metadata.bin
```

The content of the manifest.json file for the example firmware archive file is presented below.

```
{
    "manifest": {
        "firmware": {
            "firmware_image_file": "firmware.bin",
            "metadata_file": "metadata.bin",
            "firmware_id": "010246573A312E332E35"
        }
    }
}
```

3.2 Firmware Check Over HTTPS procedure

The Firmware Check Over HTTPS procedure is used to check the availability of a new firmware image over HTTP. Firmware availability checks using protocols other than HTTP are implementation specific.

The Firmware Check Over HTTPS procedure can be canceled at any time by a higher-layer model or application. When the procedure is canceled, the server shall cancel the HTTP Request that is in progress.

As inputs, the Firmware Check Over HTTPS procedure takes an Update URI and the Current Firmware ID.

The Update URI input shall use the https scheme. The Current Firmware ID shall contain an array of octets identifying firmware to be checked for a new version.

The procedure consists of the following steps:

1. The procedure shall construct a Request URI (see Section 3.2.1).
2. The HTTP client shall send an HTTP Request to the HTTP server (see Section 3.2.2) and shall wait for an HTTP Response (see Section 3.2.3).
3. When the HTTP Response is received from the HTTP server, the client shall take either of the following actions:
    - If the status code is 200 OK, then the body of the response shall contain the new firmware description file, and the procedure completes successfully.
    - If the status code is 404 Not Found, indicating that a new firmware image is not available, then the procedure completes successfully.

The procedure fails if any of the following conditions is met:

- A transmission error occurs.
- The HTTP Request does not complete with a valid HTTP Response.

- The HTTP Response status code is not either 200 OK or 404 Not Found.

3.2.1 Construction of the Request URI

The Request URI is constructed from the Update URI input and Current Firmware ID input values.

The Request URI is the concatenation of the Update URI input, the /check path suffix, and a query string, and shall have the following format:

- The Request URI scheme shall be the Update URI input scheme.
- The Request URI authority shall be the Update URI input authority.
- The Request URI path shall be constructed by concatenating the Update URI input path with the /check path suffix.
- The Request URI query string identifies the firmware installed on the node and shall have the following format:
  - The Request URI query string shall contain one key. The query key shall appear one time in the query string.
  - The query key for the query shall be “cfwid”.
  - The query key value shall be the Current Firmware ID input in Base16 encoding [4].

For example, if the Current Firmware ID input is 0x02 0xFF 0x10 0x00 0x02 0x03, and the Update URI input is “https://mesh.example.com/check-for-updates”, then the Request URI is the following:

```
https://mesh.example.com/check-for-updates/check?cfwid=02FF10000203
```

3.2.2 Request

The HTTP client shall use the Request URI to make an HTTP GET request.

3.2.3 Response

The HTTP client shall interpret the following status codes of the HTTP GET response to have the following semantics:

- **200 OK**: Request succeeded, and the requested data is in the response body.
- **404 Not Found**: The Current Firmware ID is not valid, or new firmware is not available for the requested Current Firmware ID.
- **405 Method Not Allowed**: The request method was something other than GET.
- **501 Not Implemented**: The request contains a query key or query value not known by the server.

When the response status code is 200 OK, the response headers shall contain a Content-Type header set to “application/json”. The body of the response shall contain a new firmware description file.

3.2.3.1 New firmware description file

The new firmware description file shall be in the JSON format and shall follow the schema defined in Section 3.2.3.1.1. The new firmware description file contains information about the new firmware version and the number of available new firmware images.

The manifest/firmware/firmware_id member of the new firmware description file shall contain the new firmware version.

The manifest/firmware/dfu_chain_size member of the new firmware description file shall contain the number of firmware images needed to update the firmware specified by the Current Firmware ID to

the newest version. This value can be displayed to the user to show progress of the multistage update.

The manifest/firmware/firmare_image_file_size member of the new firmware description file shall contain the size in bytes of the firmware image file of the new firmware.

3.2.3.1.1 New firmware description file JSON schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://www.bluetooth.com/specifications/specs/mesh-dfu-1-0-new-firmware-check-schema.json#",
  "type": "object",
  "title": "The New Firmware Description File Schema",
  "description": "This JSON schema is part of the Mesh Device Firmware Update Model Bluetooth specification and may be made available separately for the convenience of members of Bluetooth SIG. Use of this JSON schema by members of Bluetooth SIG is governed by the membership and other related agreements between Bluetooth SIG and its members, including those agreements posted on Bluetooth SIG's website located at www.bluetooth.com. Use of this JSON schema by anyone who is not a member of Bluetooth SIG is prohibited and is an infringement of the intellectual property rights of Bluetooth SIG and its members.",
  "required": [
    "manifest"
  ],
  "properties": {
    "manifest": {
      "$id": "#/properties/manifest",
      "type": "object",
      "required": [
        "firmware"
      ],
      "properties": {
        "firmware": {
          "$id": "#/properties/manifest/properties/firmware",
          "type": "object",
          "required": [
            "firmware_id",
            "dfu_chain_size"
          ],
          "properties": {
            "firmware_id": {
              "$id":
"#/properties/manifest/properties/firmware/properties/firmware_id",
              "type": "string",
              "examples": [
                "010246573A312E332E35"
              ],
              "pattern": "^(?:[0-9A-F]{2}){2,108}$"
            },
            "dfu_chain_size": {
              "$id":
"#/properties/manifest/properties/firmware/properties/dfu_chain_size",
              "type": "integer",
              "default": 0
```

```
            },
            "firmware_image_file_size": {
              "$id":
"#/properties/manifest/properties/firmware/properties/firmware_image_file_size",
              "type": "integer"
            }
          }
        }
      }
    }
  }
}
```

3.2.3.2 Example of new firmware description file

An example of the content of the new firmware description file is presented below.

```
{
  "manifest": {
    "firmware": {
      "firmware_id": "010246573A312E332E35",
      "dfu_chain_size": 2,
      "firmware_image_file_size": 196160
    }
  }
}
```

3.2.3.3 Example of firmware update chain

Figure 3.1 illustrates an example firmware update that involves a firmware update chain for which the Firmware Retrieval Over HTTPS procedure is invoked three times. Firmware images A, B, and C form the firmware update chain. The firmware image installed on the node is firmware image A. The newest available firmware for the node is firmware image C. Firmware image A can only be updated to image B, and image B can be updated to image C.

The first invocation of the Firmware Retrieval Over HTTPS procedure uses firmware image A as the Current Firmware ID input. The response contains a new firmware description file in which manifest/firmware/firmware_id identifies firmware image B and the manifest/firmware/dfu_chain_size value is 2. The application can inform the user that two steps are needed to update from firmware image A to the newest available version (firmware image C).

The second invocation of the Firmware Retrieval Over HTTPS procedure uses firmware image B as the Current Firmware ID input. The third invocation of the Firmware Retrieval Over HTTPS procedure uses firmware image C as the Current Firmware ID input. The Update URI input is the same for each procedure invocation and is omitted for brevity.

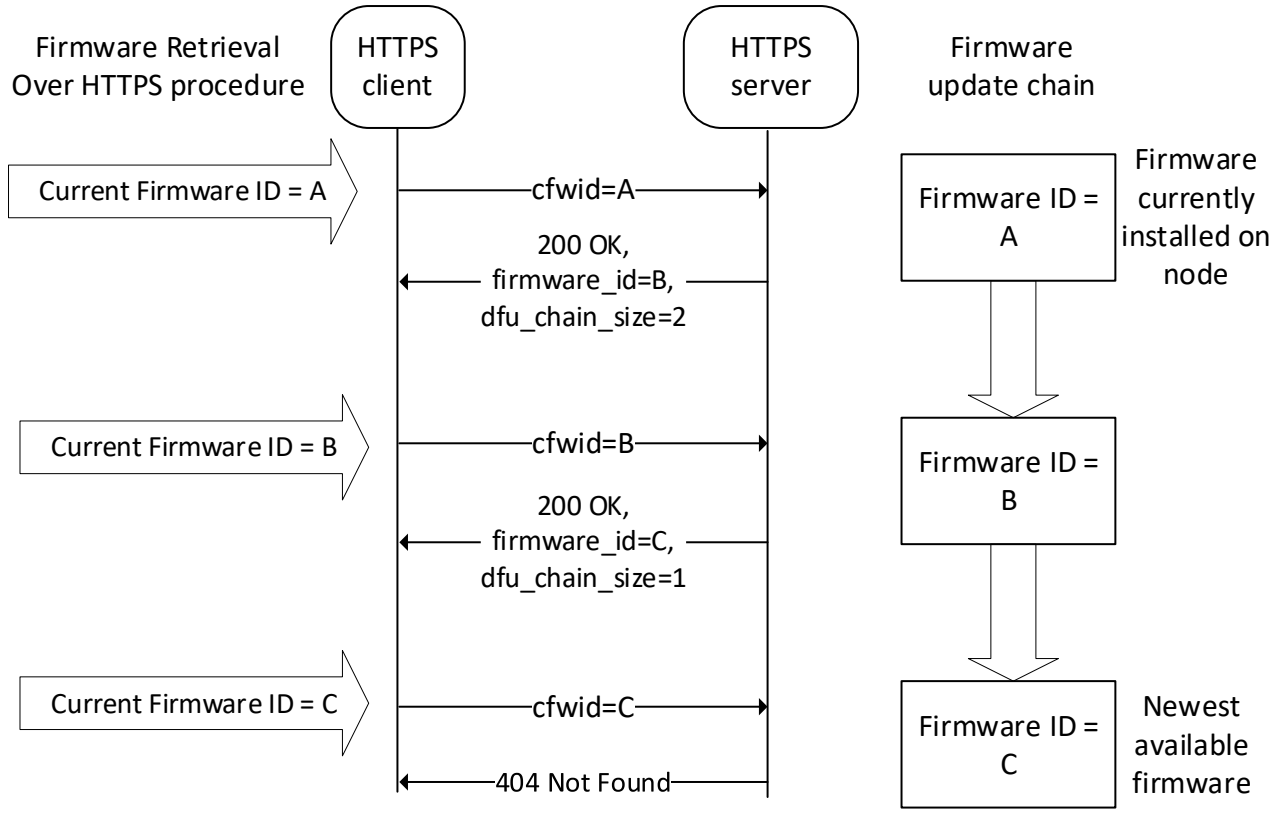

Figure 3.1: Example of a firmware update that involves a firmware update chain and requires three invocations of the Firmware Retrieval Over HTTPS procedure

3.3 Firmware Retrieval Over HTTPS procedure

The Firmware Retrieval Over HTTPS procedure is used to retrieve a new firmware image with accompanying metadata and firmware ID using HTTP. Retrieval using different protocols is implementation specific. The Firmware Retrieval Over HTTPS procedure is used by the Initiator or Distributor to retrieve a new firmware image, the new image metadata, and the new image Firmware ID.

This procedure can be canceled at any time by a higher-layer model or application. When the procedure is canceled, the server shall cancel the HTTP Request that is in progress.

As inputs, the Firmware Retrieval Over HTTPS procedure takes an Update URI and the Current Firmware ID.

The Update URI input shall use the https scheme. The Current Firmware ID shall contain an array of octets identifying a firmware image to be updated.

The procedure consists of the following steps:

1. The procedure shall construct a Request URI (see Section 3.3.1).
2. The HTTP client shall send an HTTP Request to the HTTP server (see Section 3.3.2) and shall wait for an HTTP Response (see Section 3.3.3).
3. When the HTTP Response is received from the HTTP server, the client shall take either of the following actions:
    - If the status code is 200 OK, then the body of the response shall contain the firmware archive file for the firmware image, and the procedure completes successfully.

- If the status code is 404 Not Found, indicating that a new firmware image is not available, then the procedure completes successfully.

The procedure fails if any of the following conditions is met:

- A transmission error occurs.
- The HTTP Request does not complete with a valid HTTP Response.
- The HTTP Response status code is not either 200 OK or 404 Not Found.

3.3.1 Construction of the Request URI

The Request URI is constructed from the Update URI input and Current Firmware ID input values.

The Request URI is the concatenation of the Update URI input, the /get path suffix, and a query string, and shall have the following format:

- The Request URI scheme shall be the Update URI input scheme.
- The Request URI authority shall be the Update URI input authority.
- The Request URI path shall be constructed by concatenating the Update URI input path with the /get path suffix.
- The Request URI query string identifies the firmware installed on the node and shall have the following format:
  - The Request URI query string shall contain one key. The query key shall appear one time in the query string.
  - The query key for the query shall be “cfwid”.
  - The query key value shall be the Current Firmware ID input in Base16 encoding [4].

For example, if the Current Firmware ID input is 0x02 0xFF 0x10 0x00 0x02 0x03, and the Update URI input is “https://mesh.example.com/check-for-updates”, then the Request URI is the following:

```
https://mesh.example.com/check-for-updates/get?cfwid=02FF10000203
```

3.3.2 Request

The HTTP client shall use the Request URI to make an HTTP GET request.

3.3.3 Response

The HTTP server responds to the HTTP client using the HTTP protocol.

The HTTP client shall interpret the following status codes of the HTTP GET response to have the following semantics:

- **200 OK**: Request succeeded, and the requested data is in the response body.
- **404 Not Found**: Either the Current Firmware ID is not valid, or new firmware is not available for the requested Current Firmware ID.
- **405 Method Not Allowed**: The request method was not GET.
- **501 Not Implemented**: The request contained a query key or query value not known by the server.

When the response status code is 200 OK, the response headers shall contain all headers defined in Table 3.2, with values defined in the Required Header Value column.

Header	Required Header Value	Description
Content-Type	application/gzip	–
Content-Disposition	attachment; filename="FILE_NAME"	FILE_NAME is the firmware archive file name.

Table 3.2: Response headers for a response with status code 200 OK

The body of the response shall contain a firmware archive file (see Section 3.1).

4 Firmware update states

This section defines the states used by the Firmware Update models.

Firmware ID. Firmware images shall be identified by a Firmware ID. The Firmware ID format is defined in Table 4.1.

Field	Size (octets)	Description	Req.
Company ID	2	A 16-bit Company ID, assigned by the Bluetooth SIG, that identifies the firmware vendor	M
Version Information	0...106	Vendor-specific information describing the firmware binary package	M

Table 4.1: Firmware ID format

The Company ID field shall contain a valid Bluetooth SIG-assigned Company ID [9] that identifies the firmware vendor. The Company ID may be the Company ID from Composition Data Page 0, or it may be any other valid, assigned Company ID.

If present, the Version Information field shall contain vendor-specific information describing the firmware binary package.

TTL values. The values used with the Update TTL, Upload TTL, and Distribution TTL states are defined in Table 4.2.

Value	Description
0x00–0x7F	TTL field value of a Network PDU
0x80–0xFE	Prohibited
0xFF	TTL field value used in a Network PDU is the Default TTL state value.

Table 4.2: TTL values

AppKey Index values. The values used with the Update AppKey Index and Distribution AppKey Index states are defined in Table 4.3.

Value	Description
0x0000–0x0FFF	The index of the application key
0x1000–0xFFFE	Prohibited
0xFFFF	The index of the application key is not set

Table 4.3: AppKey Index values

4.1 Firmware Update Server states

This section defines the states used by the Firmware Update Server model.

4.1.1 Firmware Information List

The Firmware Information List state is a list of Update URIs and Firmware IDs for each firmware subsystem installed on a Target node. This state changes after the node applies a new firmware update successfully.

The format of an entry in the Firmware Information List state is defined in Table 4.4.

Field	Description	Req.
Current Firmware ID	Identifies a firmware subsystem installed on a Target node	M
Update URI	Indicates the location of a new firmware archive file	M

Table 4.4: Firmware Information List entry format

The format of the Current Firmware ID field is defined in Section 4.1.1.1.

The format of the Update URI field is defined in Section 4.1.1.2.

The index of an entry in the Firmware Information List state may change throughout the lifetime of the node. The Current Firmware ID and Update URI fields are used by applications to identify the new index of an entry in an updated Firmware Information List state.

For example, if a Target node provides firmware images from a Bluetooth mesh network to a set of devices on a wired bus, then the order in which devices on the bus are listed in the Firmware Information List state can change when a device is plugged into or unplugged from the bus. When a device on the bus is updated, the new firmware information replaces the old entry in the list and this can change the index of the entry.

4.1.1.1 Current Firmware ID

The Current Firmware ID state identifies a firmware image on the node or on any subsystem within the node. The format of this state is defined in Table 4.1.

4.1.1.2 Update URI

The Update URI state indicates the location of the new firmware archive file. The Update URI state shall be either a URI, as defined in [2], or it shall be empty. If the Update URI state is not empty, then it shall be formatted as the URI data type is defined in [8] and shall use the https scheme.

The Update URI state is used as part of a URI of an HTTP request used in the Firmware Retrieval Over HTTPS procedure (see Section 3.3) or the Firmware Check Over HTTPS procedure (see Section 3.2).

4.1.2 Update Phase

The Update Phase state is a 3-bit value that identifies the firmware update phase of the Firmware Update Server. The values for this state are defined in Table 4.5. The default value of the Update Phase state shall be Idle.

Phase Code	Phase	Description
0x0	Idle	Ready to start a Receive Firmware procedure (see Section 6.1.2.1).
0x1	Transfer Error	The Transfer BLOB procedure failed (see Section 6.2.3 in [10]).
0x2	Transfer Active	The Receive Firmware procedure is being executed
0x3	Verifying Update	The Verify Firmware procedure is being executed (see Section 6.1.2.2).
0x4	Verification Succeeded	The Verify Firmware procedure completed successfully.
0x5	Verification Failed	The Verify Firmware procedure failed.
0x6	Applying Update	The Apply New Firmware procedure is being executed (see Section 6.1.2.3).
0x7	–	RFU

Table 4.5: Update Phase state values

4.1.3 Firmware Update Additional Information

The Firmware Update Additional Information state is a 5-bit value that identifies the node state after successful application of a verified firmware image. The values for this state are defined in Table 4.6. The default value of the Firmware Update Additional Information state shall be 0x0.

Value	Name	Description
0x0	CD Unchanged	Node's Composition Data state will not change.
0x1	CD Changed and RPR Unsupported	Node's Composition Data state will change, and remote provisioning is not supported. The new Composition Data state value is effective after the node is reprovisioned.
0x2	CD Changed and RPR Supported	Node's Composition Data state will change, and remote provisioning is supported. The node supports remote provisioning and Composition Data Page 128. The Composition Data Page 128 contains different information than Composition Data Page 0.
0x3	Device Unprovisioned	The node will become unprovisioned after successful application of a verified firmware image.
0x4–0x1F	RFU	Reserved for Future Use.

Table 4.6: Firmware Update Additional Information state values

4.1.4 Update Firmware Image Index

The Update Firmware Image Index state is a uint8 value that identifies the firmware image in the Firmware Information List that is being updated.

The default value of the Update Firmware Image Index state shall be 0xFF.

4.1.5 New Firmware Image

The New Firmware Image state is an internal state that contains the binary image for the firmware update.

4.1.6 Update BLOB ID

The Update BLOB ID state is an 8-octet value that contains the BLOB ID (see Section 3.1.1.1 in [10]) to be used by the BLOB Transfer Server model during a firmware update.

The default value of the Update BLOB ID state shall be Unknown.

4.1.7 Update Firmware Metadata

The Update Firmware Metadata state is a variable-length value that contains the firmware metadata for the incoming firmware image.

The default value of the Update Firmware Metadata state shall be Unknown.

4.1.8 Update TTL

The Update TTL state indicates the TTL value to be used by the BLOB Transfer Server model during a firmware update.

The values for the Update TTL state are defined in Table 4.2. The default value of the Update TTL state shall be 0xFF.

4.1.9 Update Server Timeout Base

The Update Server Timeout Base state is a uint16 value that indicates the timeout after which the Firmware Update Server suspends firmware image transfer reception.

The default value of the Update Server Timeout Base state shall be Unknown.

4.2 Firmware Update Client states

This section defines the states used by the Firmware Update Client model.

4.2.1 Update Receivers

The Update Receivers state is an internal state that lists the Target nodes to receive a firmware update. The value of this state can be empty. Each entry in the list has the format defined in Table 4.7.

Field	Description
Address	Unicast address of the Target node
Firmware Image Index	Index of the firmware image on the Target node to update
Retrieved Update Phase	The retrieved Update Phase state of the Target node
Status	Status of the last operation performed by the Target node
Update Additional Information	The retrieved Firmware Update Additional Information state of the Target node

Table 4.7: Entry format for an Update Receivers state entry

The Address field identifies the unicast address of a Target node.

The Firmware Image Index field identifies the firmware image in the Firmware Information List state of the Target node that is being updated.

The Retrieved Update Phase field identifies the phase of the firmware update on the Firmware Update Server. The value of the Retrieved Update Phase field is either the retrieved value of the Update Phase state or a value set by the client.

The values of the Retrieved Update Phase field are defined in Table 4.8. The Source column indicates whether the value comes from the Firmware Update Server or the Firmware Update Client. The default value for the Retrieved Update Phase field shall be Unknown.

Phase Code	Phase	Description	Source
0x0	Idle	No firmware transfer is in progress	Server
0x1	Transfer Error	Firmware transfer was not completed	Server
0x2	Transfer Active	Firmware transfer is in progress	Server
0x3	Verifying Update	Verification of the firmware image is in progress	Server
0x4	Verification Succeeded	Firmware image verification succeeded	Server
0x5	Verification Failed	Firmware image verification failed	Server
0x6	Applying Update	Firmware applying is in progress	Server
0x7	Transfer Canceled	Firmware transfer has been canceled	Client
0x8	Apply Success	Firmware applying succeeded	Client
0x9	Apply Failed	Firmware applying failed	Client
0xA	Unknown	Phase of a node was not yet retrieved	Client
0xB–0xF	–	RFU	–

Table 4.8: Retrieved Update Phase field values

The Status field shall identify the Status Code for the last operation performed on the Firmware Update Server states. The values of the Status field are defined in Section 4.5. The default value for this field shall be Success.

The Update Additional Information field indicates the value of the Firmware Update Additional Information state on the Target node. The default value for this field shall be Unknown.

The default value of the Update Receivers state shall be empty.

4.2.2 Active Update Receivers

The Active Update Receivers state is an implicit state that identifies those Firmware Update Servers that are actively participating in a firmware update (e.g., have not timed out of the firmware update). This state contains a subset of items from the Update Receivers list. The internal representation of this state is not defined by this specification.

The default value of the Active Update Receivers state is Unknown.

4.2.3 Update Multicast Address

The Update Multicast Address state is a state that contains a group address, the Label UUID of a virtual address, or the Unassigned address. If the value of this state is a group address or a virtual address, then the client sends messages to the Firmware Update Servers at that address.

If the value is the Unassigned address, then messages are not sent to a multicast address. The default value of the Update Multicast Address state shall be the Unassigned address.

4.2.4 Update AppKey Index

The Update AppKey Index state is a uint16 value that indicates the index of the application key that the Firmware Update Client shall use to exchange messages with the Firmware Update Server.

The values of the Update AppKey Index state are defined in Table 4.3. The default value of the Update AppKey Index state shall be 0xFFFF.

4.2.5 Update TTL

The Update TTL state indicates the TTL value to be used by the BLOB Transfer Client model (see Section 6 in [10]) during a firmware update.

The values for the Update TTL state are defined in Table 4.2. The default value of the Update TTL state shall be 0xFF.

4.2.6 Update Client Timeout Base

The Update Client Timeout Base state is a uint16 value that is used to compute the timeout for the response for the request sent to a Firmware Update Server.

The default value of the Update Client Timeout Base state shall be Unknown.

4.3 Firmware Distribution Server states

This section defines the states used by the Firmware Distribution Server model.

4.3.1 Distribution Receivers

The Distribution Receivers state is a composite state that indicates the list of Target nodes (receivers) that are the target of a firmware update from the Firmware Distribution Server.

The Distribution Receivers state includes the following states:

- Distribution Receivers List state (see Section 4.3.1.1)
- Distribution Receivers List Count state (see Section 4.3.1.2)

4.3.1.1 Distribution Receivers List

The Distribution Receivers List state is a list identifying Target nodes and the firmware image that is the target of a firmware update.

The structure of an entry in the Distribution Receivers List state is defined in Table 4.9.

Field	Size (octets)	Description	Req.
Target Node Address	2	Unicast address of a Target node.	M
Update Firmware Image Index	1	Index of the firmware image in the Firmware Information List state of the Target node to be updated.	M

Table 4.9: Structure of a Distribution Receivers List state entry

The Target Node Address field shall contain the unicast address of a Target node.

The Update Firmware Image Index field shall contain the index of the firmware image in the Firmware Information List state of the Target node that will be updated.

The default value of the Distribution Receivers List state shall be empty.

4.3.1.2 Distribution Receivers List Count

The Distribution Receivers List Count state is a uint16 value that indicates the number of entries in the Distribution Receivers List state.

The value of the Distribution Receivers List Count cannot exceed the value of the Max Distribution Receivers List Size from the Distributor Capabilities state (see Section 4.3.2.1).

The default value of the Distribution Receivers List Count shall be 0.

4.3.2 Distributor Capabilities

The Distributor Capabilities state is a composite state that indicates firmware update capabilities supported by the Firmware Distribution Server model.

The Distributor Capabilities state includes the following states:

- Max Distribution Receivers List Size state (see Section 4.3.2.1)
- Max Firmware Images List Size state (see Section 4.3.2.2)
- Max Firmware Image Size state (see Section 4.3.2.3)
- Max Upload Space state (see Section 4.3.2.4)
- Remaining Upload Space state (see Section 4.3.2.5)
- Out-of-Band Retrieval Supported state (see Section 4.3.2.6)
- Supported URI Scheme Names state (see Section 4.3.2.7)

4.3.2.1 Max Distribution Receivers List Size

The Max Distribution Receivers List Size state is a uint16 value that indicates the maximum number of addresses that a Firmware Distribution Server can store in the Distribution Receivers List state (see Section 4.3.1.1).

The default value of the Max Distribution Receivers List Size state is implementation specific.

4.3.2.2 Max Firmware Images List Size

The Max Firmware Images List Size state is a uint16 value that determines the maximum number of firmware images that the Firmware Distribution Server can store in the Firmware Images List state.

The default value of the Max Firmware Images List Size state is implementation specific.

4.3.2.3 Max Firmware Image Size

The Max Firmware Image Size state is a uint32 value that indicates the maximum size of a single firmware image (in octets) that can be stored on the Firmware Distribution Server.

The default value of the Max Firmware Image Size state is implementation specific.

4.3.2.4 Max Upload Space

The Max Upload Space state is a uint32 value that indicates the total storage dedicated to the storage of firmware images by Firmware Distribution Servers on a node.

The values for the Max Upload Space state are defined in Table 4.10.

Value	Description
0x00000000–0xFFFFFFFE	Total size of storage dedicated to firmware images (in octets)
0xFFFFFFFF	Size of storage dedicated to firmware images is 0xFFFFFFFF octets or greater.

Table 4.10: Max Upload Space state format

The default value of the Max Upload Space state is implementation specific.

4.3.2.5 Remaining Upload Space

The Remaining Upload Space state is a uint32 value that indicates the available space (in octets) in storage that is dedicated to storage of firmware images on Firmware Distribution Servers. The value of the Remaining Upload Space state shall not be greater than the value of the Max Upload Space state.

The values for the Remaining Upload Space state are defined in Table 4.11.

Value	Description
0x00000000–0xFFFFFFFE	Total available space in storage dedicated to firmware images (in octets)
0xFFFFFFFF	Available space in storage dedicated to firmware images is 0xFFFFFFFF octets or greater.

Table 4.11: Remaining Upload Space state format

The default value of the Remaining Upload Space state is implementation specific.

4.3.2.6 Out-of-Band Retrieval Supported

The Out-of-Band Retrieval Supported state is a uint8 value that indicates whether the Firmware Distribution Server supports the capability to retrieve firmware images by using OOB mechanisms.

The values for the Out-of-Band Retrieval Supported state are defined in Table 4.12.

Value	Description
0x00	OOB Not Supported
0x01	OOB Supported
0x02–0xFF	Prohibited

Table 4.12: Out-of-Band Retrieval Supported state values

The default value of the Out-of-Band Retrieval Supported state is implementation specific.

4.3.2.7 Supported URI Scheme Names

The Supported URI Scheme Names state is a list of supported URI schemes.

Each scheme name shall be a code point defined in the Bluetooth SIG Assigned Numbers [9]. The value of the Supported URI Scheme Names state shall be an array of UTF-8 encoded code points.

When the Out-of-Band Retrieval Supported state is OOB Supported (0x01), the Supported URI Scheme Names shall indicate at least one supported URI scheme.

When the Supported URI Scheme Names state indicates support for the https scheme, the Firmware Retrieval Over HTTPS procedure (see Section 3.3) shall be supported.

For example, if both the file and https URI schemes are supported, then the U+0010 and U+0017 code points indicate their support. The Supported URI Scheme Names state value for those two code points is an array containing two octets: 0x10, 0x17.

The default value of the Supported URI Scheme Names state is implementation specific.

4.3.3 Distribution Parameters

The Distribution Parameters state is a composite state that contains parameters used by the Firmware Distribution Server model during distribution of a firmware image.

The Distribution Parameters state includes the following states:

- Distribution Phase state (see Section 4.3.3.1)
- Update Policy state (see Section 4.3.3.2)
- Distribution Multicast Address state (see Section 4.3.3.3)
- Distribution AppKey Index state (see Section 4.3.3.4)
- Distribution TTL state (see Section 4.3.3.5)
- Distribution Timeout Base state (see Section 4.3.3.6)
- Distribution Transfer Mode state (see Section 4.3.3.7)
- Distribution Firmware Image Index state (see Section 4.3.3.8)

4.3.3.1 Distribution Phase

The Distribution Phase state is a uint8 value that indicates the phase of a firmware image distribution being performed by the Firmware Distribution Server. The values for this state are defined in Table 4.13.

Value	Name	Description
0x00	Idle	No firmware distribution is in progress.
0x01	Transfer Active	Firmware distribution is in progress.
0x02	Transfer Success	The Transfer BLOB procedure has completed successfully.
0x03	Applying Update	The Apply Firmware On Target Nodes procedure is being executed.
0x04	Completed	The Distribute Firmware procedure has completed successfully.
0x05	Failed	The Distribute Firmware procedure has failed.
0x06	Canceling Update	The Cancel Firmware Update procedure is being executed.

Value	Name	Description
0x07	Transfer Suspended	The Transfer BLOB procedure is suspended.
0x08–0xFF	–	RFU

Table 4.13: Distribution Phase state values

The default value of the Distribution Phase state shall be Idle.

4.3.3.2 Update Policy

The Update Policy state is a 1-bit value that indicates when to apply a new firmware image. The values for this state are defined in Table 4.14.

Value	Update Policy	Description
0x00	Verify Only	The Firmware Distribution Server verifies that firmware image distribution completed successfully but does not apply the update. The Initiator (the Firmware Distribution Client) initiates firmware image application.
0x01	Verify And Apply	The Firmware Distribution Server verifies that firmware image distribution completed successfully and then applies the firmware update.

Table 4.14: Update Policy state values

The default value of the Update Policy state shall be Unknown.

4.3.3.3 Distribution Multicast Address

The Distribution Multicast Address state indicates the destination address to use when transferring a firmware image to Target nodes.

The Distribution Multicast Address state shall contain a group address, the Label UUID of a virtual address, or the Unassigned address.

If the value of the Distribution Multicast Address state is the Unassigned address, then messages are not sent to a multicast address. The default value of this state shall be the Unassigned address.

4.3.3.4 Distribution AppKey Index

The Distribution AppKey Index state is a uint16 value that indicates the index of the application key that the Firmware Distribution Server shall use when executing Firmware Update Client procedures (see Section 7.1.2).

The values of the Distribution AppKey Index state are defined in Table 4.3. The default value of the Distribution AppKey Index state shall be 0xFFFF.

4.3.3.5 Distribution TTL

The Distribution TTL state indicates the TTL value the Firmware Distribution Server shall use when executing Firmware Update Client procedures (see Section 7.1.2).

The values for the Distribution TTL state are defined in Table 4.2. The default value of the Distribution TTL state shall be 0xFF.

4.3.3.6 Distribution Timeout Base

The Distribution Timeout Base state is a uint16 value that is used by the procedures of the Firmware Update Client model and BLOB Transfer Client model to calculate timeout values during firmware image distribution.

The default value of the Distribution Timeout Base state shall be Unknown.

4.3.3.7 Distribution Transfer Mode

The Distribution Transfer Mode state is a 2-bit value that indicates the transfer mode used during firmware image distribution.

The values for the Distribution Transfer Mode state are defined in Section 3.1.4 in [10].

The default value of the Distribution Transfer Mode state shall be Unknown.

4.3.3.8 Distribution Firmware Image Index

The Distribution Firmware Image Index state identifies the firmware image being distributed.

The value of the Distribution Firmware Image Index state identifies the firmware image in the Firmware Images List state to transfer during firmware image distribution.

The default value of the Distribution Firmware ID state shall be Unknown.

4.3.4 Upload Parameters

The Upload Parameters state is a composite state that contains parameters used by the Firmware Distribution Server model during a firmware image upload to the Distributor.

The Upload Parameters state includes the following states:

- Upload Type state (see Section 4.3.4.1)
- Upload Firmware ID state (see Section 4.3.4.2)
- Upload Phase state (see Section 4.3.4.3)
- Upload Firmware Size state (see Section 4.3.4.4)
- Upload BLOB ID state (see Section 4.3.4.5)
- Upload Firmware Metadata state (see Section 4.3.4.6)
- Upload OOB Firmware ID state (see Section 4.3.4.7)
- Upload OOB URI state (see Section 4.3.4.8)
- Upload TTL state (see Section 4.3.4.9)
- Upload Timeout Base state (see Section 4.3.4.10)

4.3.4.1 Upload Type

The Upload Type state is a 1-bit value that indicates the type of the upload. The values for the Upload Type state are defined in Table 4.15.

Value	Name	Description
0x0	In-band	The firmware upload is executed in-band.
0x1	Out-of-band	The firmware upload is executed OOB.

Table 4.15: Upload Type values

The default value of the Upload Type state shall be In-band.

4.3.4.2 Upload Firmware ID

The Upload Firmware ID state identifies the firmware image being uploaded to the Distributor.

The format of the Upload Firmware ID field is defined in Table 4.1.

The default value of the Upload Firmware ID state shall be Unknown.

4.3.4.3 Upload Phase

The Upload Phase state is a uint8 value that indicates the phase of a firmware image delivery to a Firmware Distribution Server.

The values for the Upload Phase state are defined in Table 4.16.

Value	Name	Description
0x00	Idle	No firmware upload is in progress.
0x01	Transfer Active	The Store Firmware or Store Firmware OOB procedure is being executed.
0x02	Transfer Error	The Store Firmware procedure or Store Firmware OOB procedure failed.
0x03	Transfer Success	The Store Firmware procedure or the Store Firmware OOB procedure completed successfully.
0x04–0xFF	–	RFU

Table 4.16: Upload Phase state values

The default value of the Upload Phase state shall be Idle.

4.3.4.4 Upload Firmware Size

The Upload Firmware Size state is a uint32 value that indicates the size of the firmware image being uploaded to the Firmware Distribution Server.

The default value of the Upload Firmware Size state shall be Unknown.

4.3.4.5 Upload BLOB ID

The Upload BLOB ID state is an 8-octet value contains the BLOB ID (see Section 3.1.1.1 in [10]) to be used by the BLOB Transfer Server model during a firmware upload to the Firmware Distribution Server.

The default value of the Upload BLOB ID state shall be Unknown.

4.3.4.6 Upload Firmware Metadata

The Upload Firmware Metadata state is a variable-length value that contains the firmware metadata of the firmware image being uploaded to the Firmware Distribution Server.

The default value of the Upload Firmware Metadata state shall be Unknown.

4.3.4.7 Upload OOB Firmware ID

The Upload OOB Firmware ID state identifies the Firmware ID that the Store Firmware OOB procedure uses to retrieve a new firmware image.

The format of the Upload OOB Firmware ID field is defined in Table 4.1.

The default value of the Upload OOB Firmware ID state shall be Unknown.

4.3.4.8 Upload OOB URI

The Upload OOB URI state is a string value that indicates the URI that the Store Firmware OOB procedure uses to retrieve a new firmware image. The format of this state is defined in [2].

The default value of the Upload OOB URI state shall be Unknown.

4.3.4.9 Upload TTL

The Upload TTL state indicates the TTL value the Firmware Distribution Server shall use when executing the Initialize And Receive BLOB procedures (see Section 5.2.1 in [10]) during a firmware image upload.

The values for the Upload TTL state are defined in Table 4.2. The default value of the Upload TTL state shall be 0xFF.

4.3.4.10 Upload Timeout Base

The Upload Timeout Base state is a uint16 value used by the Firmware Distribution Server model to calculate the timeout value during a firmware upload to the Firmware Distribution Server.

The default value of the Upload Timeout Base state shall be Unknown.

4.3.5 Firmware Images List

The Firmware Images List state contains information about each firmware image that is stored on a Firmware Distribution Server.

The list entries have the format defined in Table 4.17. The default value of the Firmware Images List state is implementation specific.

Field	Description
Index	Indicates the position of the entry in the list.
Firmware ID	Identifies the firmware image that is stored on the Firmware Distribution Server.
Firmware Metadata	The metadata of the firmware image identified by the Firmware ID field.

Table 4.17: Format of the Firmware Images List state entry

The index value of the first entry shall be 0. The indexes are positive consecutive integers.

The format of the Firmware ID field is defined in Table 4.1.

The Firmware Metadata field is a variable-length value that shall contain the firmware metadata for the firmware image identified by the Firmware ID field.

4.4 Firmware Distribution Client states

This section defines the states used by the Firmware Distribution Client model.

4.4.1 Update Information From Target Nodes

The Update Information From Target Nodes state is a list identifying the firmware images installed on each Target node that the Firmware Distribution Client provides updates for.

The format of the Update Information From Target Nodes state's entry is defined in Table 4.18.

Field	Size (in octets)	Description
Address	2	Unicast address of the Target node
Firmware Images List	variable	List with entries

Table 4.18: Update Information From Target Nodes state format

The Address field is the unicast address of a Target node.

The Firmware Images List field is a list of firmware images installed on the Target node identified by the Address field. The format of the Firmware Images List field is defined in Table 4.19.

The default value of the Update Information From Target Nodes state shall be empty.

Field	Size (in octets)	Description
Firmware Image Entry	variable	First entry
Firmware Image Entry	variable	Second entry
…	…	…
Firmware Image Entry	variable	Last entry

Table 4.19: Firmware Images List field format

The Firmware Images Entry format is defined in Table 4.20.

Field	Size (in octets)	Description
Current Firmware ID Length	1	Length of the Current Firmware ID field
Current Firmware ID	variable	Identifies the most recently reported firmware version on the Target node
New Firmware ID Length	1	Length of the New Firmware ID field
New Firmware ID	variable	Identifies the new firmware available for the Target node. This field may be empty
Update URI Length	1	Length of the Update URI field
Update URI	1…255	URI used to locate new firmware
Firmware Image Length	4	Length of the Firmware Image field
Firmware Image	variable	Binary data that comprises the firmware image

Table 4.20: Firmware Image Entry field format

The format of the Current Firmware ID field is defined in Table 4.1.

The format of the New Firmware ID field is defined in Table 4.1. The default value of the New Firmware ID field is empty.

The format of the URI in the Update URI field is defined in [8]. The default value of the Update URI field is empty.

The format of the Firmware Image is binary data. The default value of the Firmware Image field is empty.

4.5 Summary of status codes

The status codes for the Firmware Update Server model and the Firmware Update Client model are defined in Table 4.21.

Status Code	Status Code Name	Description
0x00	Success	The message was processed successfully.
0x01	Insufficient Resources	Insufficient resources on the node
0x02	Wrong Phase	The operation cannot be performed while the server is in the current phase.
0x03	Internal Error	An internal error occurred on the node.
0x04	Wrong Firmware Index	The message contains a firmware index value that is not expected.
0x05	Metadata Check Failed	The metadata check failed.

Status Code	Status Code Name	Description
0x06	Temporarily Unavailable	The server cannot start a firmware update.
0x07	BLOB Transfer Busy	Another BLOB transfer is in progress.

Table 4.21: Status codes for the Firmware Update Server model and Firmware Update Client model

The status codes for the Firmware Distribution Server model and the Firmware Distribution Client model are defined in Table 4.22.

Status Code	Status Code Name	Description
0x00	Success	The message was processed successfully.
0x01	Insufficient Resources	Insufficient resources on the node
0x02	Wrong Phase	The operation cannot be performed while the server is in the current phase.
0x03	Internal Error	An internal error occurred on the node.
0x04	Firmware Not Found	The requested firmware image is not stored on the Distributor.
0x05	Invalid AppKey Index	The AppKey identified by the AppKey Index is not known to the node.
0x06	Receivers List Empty	There are no Target nodes in the Distribution Receivers List state.
0x07	Busy With Distribution	Another firmware image distribution is in progress.
0x08	Busy With Upload	Another upload is in progress.
0x09	URI Not Supported	The URI scheme name indicated by the Update URI is not supported.
0x0A	URI Malformed	The format of the Update URI is invalid.
0x0B	URI Unreachable	The URI is unreachable.
0x0C	New Firmware Not Available	The Check Firmware OOB procedure did not find any new firmware.
0x0D	Suspend Failed	The suspension of the Distribute Firmware procedure failed.
0x0E–0xFF	Reserved For Future Use	Reserved for future use

Table 4.22: Status codes for the Firmware Distribution Server model and Firmware Distribution Client model

5 Firmware update messages

This section defines the messages used by the Firmware Update models.

5.1 Firmware Update model messages

This section defines the format of messages exchanged between the Firmware Update Client and Firmware Update Server models.

5.1.1 Firmware Update Information Get

The Firmware Update Information Get message is an acknowledged message used to get information about the firmware images installed on a node.

The response to the Firmware Update Information Get message is a Firmware Update Information Status message.

The structure of the Firmware Update Information Get message is defined in Table 5.1.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
First Index	1	Index of the first requested entry from the Firmware Information List state	M
Entries Limit	1	Maximum number of entries that the server includes in a Firmware Update Information Status message	M

Table 5.1: Firmware Update Information Get message structure

The Opcode field shall contain the opcode value for the Firmware Update Information Get message defined in the Assigned Numbers document [9].

The First Index field shall indicate the first entry on the Firmware Information List state of the Firmware Update Server (see Section 4.1.1) to return in the Firmware Update Information Status message.

The Entries Limit field shall indicate the maximum number of Firmware Information Entry fields to return in the Firmware Update Information Status message.

5.1.2 Firmware Update Information Status

The Firmware Update Information Status message is an unacknowledged message used to report information about firmware images installed on a node.

The Firmware Update Information Status message is sent in response to a Firmware Update Information Get message.

The structure of the Firmware Update Information Status message is defined in Table 5.2.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Firmware Information List Count	1	The number of entries in the Firmware Information List state	M

Field	Size (octets)	Description	Req.
First Index	1	Index of the first requested entry from the Firmware Information List state	M
Firmware Information List	variable	List of entries	M

Table 5.2: Firmware Update Information Status message structure

The Opcode field shall contain the opcode value for the Firmware Update Information Status message defined in the Assigned Numbers document [9].

The Firmware Information List Count field shall indicate the number of entries in the Firmware Information List state.

The First Index field shall indicate the first entry on the Firmware Information List state of the Firmware Update Server (see Section 4.1.1) that was returned.

The Firmware Information List field shall contain the selected entries from the Firmware Information List state. The structure of the Firmware Information List field is defined in Table 5.3.

Field	Size (octets)	Description
Firmware Information Entry	variable	First entry
Firmware Information Entry	variable	Second entry
…	…	…
Firmware Information Entry	variable	Last entry

Table 5.3: Firmware Information Entry field format

The Firmware Information Entry field shall identify the information for a firmware subsystem on the node from the Firmware Information List state.

The structure of the Firmware Information Entry field is defined in Table 5.4.

Field	Size (octets)	Description	Req.
Current Firmware ID Length	1	Length of the Current Firmware ID field	M
Current Firmware ID	variable	Identifies the firmware image on the node or any subsystem on the node.	M
Update URI Length	1	Length of the Update URI field	M
Update URI	1 to 255	URI used to retrieve a new firmware image	C.1

Table 5.4: Firmware Information Entry field format

C.1: If the value of the Update URI Length field is greater than 0, then the Update URI field shall be present.

The Current Firmware ID Length field shall indicate the length of the Current Firmware ID field.

The Current Firmware ID field shall contain the Firmware ID of a firmware image installed on the node. The format of this field is defined in Table 4.1.

The Update URI Length field shall indicate the length of the Update URI field.

If present, the Update URI field shall indicate the URI link used to retrieve the firmware image.

5.1.3 Firmware Update Firmware Metadata Check

The Firmware Update Firmware Metadata Check message is an acknowledged message, sent to a Firmware Update Server, to check whether the node can accept a firmware update.

The response to the Firmware Update Firmware Metadata Check message is a Firmware Update Firmware Metadata Status message.

The structure of the Firmware Update Firmware Metadata Check message is defined in Table 5.5.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Update Firmware Image Index	1	Index of the firmware image in the Firmware Information List state to check	M
Incoming Firmware Metadata	1 to 255	Vendor-specific metadata	O

Table 5.5: Firmware Update Firmware Metadata Check message structure

The Opcode field shall contain the opcode value for the Firmware Update Firmware Metadata Check message defined in the Assigned Numbers document [9].

The Update Firmware Image Index field shall identify the firmware image in the Firmware Information List state on the Firmware Update Server that the metadata is checked against.

If present, the Incoming Firmware Metadata field shall contain the custom data from the firmware vendor. The firmware metadata can be used to check whether the installed firmware image identified by the Firmware Image Index field will accept an update based on the metadata provided for the new firmware image.

5.1.4 Firmware Update Firmware Metadata Status

The Firmware Update Firmware Metadata Status message is an unacknowledged message sent to a Firmware Update Client that is used to report whether a Firmware Update Server can accept a firmware update.

The Firmware Update Firmware Metadata Status message is sent in response to a Firmware Update Firmware Metadata Check message.

The structure of the Firmware Update Firmware Metadata Status message is defined in Table 5.6.

Field	Size (bits)	Description	Req.
Opcode	16	The message opcode	M
Status	3	Status Code from the firmware metadata check	M
Additional Information	5	The Firmware Update Additional Information state from the Firmware Update Server (see Section 4.1.3)	M
Update Firmware Image Index	8	Index of the firmware image in the Firmware Information List state that was checked	M

Table 5.6: Firmware Update Firmware Metadata Status message structure

The Opcode field shall contain the opcode value for the Firmware Update Firmware Metadata Status message defined in the Assigned Numbers document [9].

The Status field shall report whether the firmware metadata check against the image by the Firmware Image Index indicated that the firmware image can be updated. The status codes for the Status field are defined in Table 4.21.

The Additional Information field shall indicate the Firmware Update Additional Information state (see Section 4.1.3).

The Firmware Image Index field shall identify the firmware image in the Firmware Information List state on the Firmware Update Server that was checked.

5.1.5 Firmware Update Get

The Firmware Update Get message is an acknowledged message used to get the current status of the Firmware Update Server.

The response to the Firmware Update Get message is a Firmware Update Status message.

The structure of the Firmware Update Get message is defined in Table 5.7.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.7: Firmware Update Get message structure

The Opcode field shall contain the opcode value for the Firmware Update Get message defined in the Assigned Numbers document [9].

5.1.6 Firmware Update Start

The Firmware Update Start message is an acknowledged message used to start a firmware update on a Firmware Update Server.

The response to the Firmware Update Start message is a Firmware Update Status message.

The structure of the Firmware Update Start message is defined in Table 5.8.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Update TTL	1	TTL value to use during firmware image transfer	M
Update Timeout Base	2	Used to compute the timeout of the firmware image transfer	M
Update BLOB ID	8	BLOB identifier for the firmware image	M
Update Firmware Image Index	1	Index of the firmware image in the Firmware Information List state to be updated	M
Incoming Firmware Metadata	1 to 255	Vendor-specific firmware metadata	O

Table 5.8: Firmware Update Start message structure

The Opcode field shall contain the opcode value for the Firmware Update Start message defined in the Assigned Numbers document [9].

The Update TTL field shall indicate the TTL value that is used during firmware image transfer. The values for the Update TTL field are defined in Table 4.2.

The Update Timeout Base field shall contain the value that the Firmware Update Server uses to calculate when firmware image transfer will be suspended.

The Update BLOB ID field shall contain the BLOB identifier to use during firmware image transfer.

The Firmware Image Index field shall identify the firmware image in the Firmware Information List state to be updated.

If present, the Incoming Firmware Metadata field shall contain the custom data from the firmware vendor that is used to check whether the firmware image can be updated.

5.1.7 Firmware Update Cancel

The Firmware Update Cancel message is an acknowledged message used to cancel a firmware update and delete any stored information about the update on a Firmware Update Server.

The response to a Firmware Update Cancel message is a Firmware Update Status message.

The structure of the Firmware Update Cancel message is defined in Table 5.9.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.9: Firmware Update Cancel message structure

The Opcode field shall contain the opcode value for the Firmware Update Cancel message defined in the Assigned Numbers document [9].

5.1.8 Firmware Update Apply

The Firmware Update Apply message is an acknowledged message used to apply a firmware image that has been transferred to a Firmware Update Server.

The response to a Firmware Update Apply message is a Firmware Update Status message.

The structure of the Firmware Update Apply message is defined in Table 5.10.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.10: Firmware Update Apply message structure

The Opcode field shall contain the opcode value for the Firmware Update Apply message defined in the Assigned Numbers document [9].

5.1.9 Firmware Update Status

The Firmware Update Status message is an unacknowledged message sent by a Firmware Update Server to report the status of a firmware update.

A Firmware Updates Status message is sent in response to a Firmware Update Get message, a Firmware Update Start message, a Firmware Update Cancel message, or a Firmware Update Apply message.

The structure of the Firmware Update Status message is defined in Table 5.11.

Field	Size (bits)	Description	Req.
Opcode	16	The message opcode	M
Status	3	Status Code for the requesting message	M
RFU1	2	Reserved for Future Use	M
Update Phase	3	The Update Phase state of the Firmware Update Server	M
Update TTL	8	TTL value to use during firmware image transfer	O
Additional Information	5	The Firmware Update Additional Information state from the Firmware Update Server (see Section 4.1.3)	C.1
RFU2	3	Reserved for Future Use	C.1
Update Timeout Base	16	Used to compute the timeout of the firmware image transfer	C.1

Field	Size (bits)	Description	Req.
Update BLOB ID	64	BLOB identifier for the firmware image	C.1
Update Firmware Image Index	8	The index of the firmware image in the Firmware Information List state being updated	C.1

Table 5.11: Firmware Update Status message structure

C.1: If the Update TTL field is present, then the Additional Information field, Update Timeout Base field, Update BLOB ID field, and Firmware Image Index field shall be present; otherwise, the Additional Information field, RFU2 field, Update Timeout Base field, Update BLOB ID field, and Update Firmware Image Index field shall not be present.

The Opcode field shall contain the opcode value for the Firmware Update Status message defined in the Assigned Numbers document [9].

The Status field shall identify the Status Code for the last operation performed on the Firmware Update Server state. The values of the Status field are defined in Table 4.21.

The Update Phase field shall indicate the Update Phase state (see Section 4.1.2).

If present, the Update TTL field shall indicate the Update TTL state. The Update TTL values are defined in Table 4.2.

If present, the Additional Information field shall indicate the Firmware Update Additional Information state (see Section 4.1.3).

If present, the Update Timeout Base field shall indicate the Update Server Timeout Base state.

If present, the Update BLOB ID field shall indicate the Update BLOB ID state.

If present, the Update Firmware Image Index field shall identify the firmware image in the Firmware Information List state being updated.

5.2 Firmware Distribution model messages

This section defines the format of messages exchanged between the Firmware Distribution Client and Firmware Distribution Server models.

5.2.1 Firmware Distribution Receivers Add

The Firmware Distribution Receivers Add message is an acknowledged message sent by a Firmware Distribution Client to add new entries to the Distribution Receivers List state of a Firmware Distribution Server.

The response to a Firmware Distribution Receivers Add message is a Firmware Distribution Receivers Status message.

The structure of the Firmware Distribution Receivers Add message is defined in Table 5.12.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Receivers List	variable	List of Receiver Entry fields	M

Table 5.12: Firmware Distribution Receivers Add message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Receivers Add message defined in the Assigned Numbers document [9].

The structure of the Receivers List field is defined in Table 5.13.

Field	Size (octets)	Description
Receiver Entry	3	First entry
Receiver Entry	3	Second entry
…	…	…
Receiver Entry	3	Last entry

Table 5.13: Receivers List field format

The Receivers List field shall contain at least one Receiver Entry field. For each Receiver Entry field in the Receivers List field, the value of the Address field shall be unique.

The structure of the Receiver Entry field is defined in Table 5.14.

Field	Size (bits)	Description	Req.
Address	16	The unicast address of the Target node	M
Update Firmware Image Index	8	The index of the firmware image in the Firmware Information List state to be updated	M

Table 5.14: The format of the Receiver Entry field

The Address field shall contain the unicast address of a Target node.

The Update Firmware Image Index field shall identify the index of a firmware image in the Firmware Information List state of the Target node to be updated.

5.2.2 Firmware Distribution Receivers Delete All

The Firmware Distribution Receivers Delete All message is an acknowledged message sent by a Firmware Distribution Client to remove all entries from the Distribution Receivers List state of a Firmware Distribution Server.

The response to a Firmware Distribution Receivers Delete All message is a Firmware Distribution Receivers Status message.

The structure of the Firmware Distribution Receivers Delete All message is defined in Table 5.15.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.15: Firmware Distribution Receivers Delete All message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Receivers Delete All message defined in the Assigned Numbers document [9].

5.2.3 Firmware Distribution Receivers Status

The Firmware Distribution Receivers Status message is an unacknowledged message sent by a Firmware Distribution Server to report the size of the Distribution Receivers List state.

A Firmware Distribution Receivers Status message is sent as a response to a Firmware Distribution Receivers Add message or a Firmware Distribution Receivers Delete All message.

The structure of the Firmware Distribution Receivers Status message is defined in Table 5.16.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Status	1	Status Code for the requesting message	M
Receivers List Count	2	The number of entries in the Distribution Receivers List state	M

Table 5.16: Firmware Distribution Receivers Status message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Receivers Status message defined in the Assigned Numbers document [9].

The Status field shall identify the Status Code for the last operation performed on the Firmware Distribution Server states. The values for the Status field are defined in Table 4.22.

The Receivers List Count field shall indicate the Distribution Receivers List Count state.

5.2.4 Firmware Distribution Receivers Get

The Firmware Distribution Receivers Get message is an acknowledged message sent by the Firmware Distribution Client to get the firmware distribution status of each Target node.

The response to a Firmware Distribution Receivers Get message is a Firmware Distribution Receivers List message.

The structure of the Firmware Distribution Receivers Get message is defined in Table 5.17.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
First Index	2	Index of the first requested entry from the Distribution Receivers List state	M
Entries Limit	2	Maximum number of entries that the server includes in a Firmware Distribution Receivers List message	M

Table 5.17: Firmware Distribution Receivers Get message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Receivers Get message defined in the Assigned Numbers document [9].

The First Index field shall indicate the first entry of the Distribution Receivers List state of the Firmware Distribution Server to return in the Firmware Distribution Receivers List message.

The Entries Limit field shall set the maximum number of Target Node Entries to return in the Firmware Distribution Receivers List message. The value of the Entries Limit field shall be greater than 0.

5.2.5 Firmware Distribution Receivers List

The Firmware Distribution Receivers List message is an unacknowledged message sent by the Firmware Distribution Server to report the firmware distribution status of each receiver.

A Firmware Distribution Receivers List message is sent as a response to a Firmware Distribution Receivers Get message.

The structure of the Firmware Distribution Receivers List message is defined in Table 5.18.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Receivers List Count	2	The number of entries in the Distribution Receivers List state	M
First Index	2	Index of the first requested entry from the Distribution Receivers List state	M
Receivers List	variable	List of entries	O

Table 5.18: Firmware Distribution Receivers List message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Receivers List message defined in the Assigned Numbers document [9].

The Receivers List Size field shall indicate the Distribution Receivers List Size state.

The First Index field shall indicate the index of the first entry in the Distribution Receivers List state of the Firmware Distribution Server that was returned.

If present, the Receivers List field shall contain entries from the Distribution Receivers List state. The structure of the Receivers List field is defined in Table 5.19.

Field	Size (octets)	Description
Target Node Entry	5	First entry
Target Node Entry	5	Second entry
…	…	…
Target Node Entry	5	Last entry

Table 5.19: Receivers List field format

The structure of the Target Node Entry field is defined in Table 5.20.

Field	Size (bits)	Description	Req.
Address	15	15 least significant bits of the unicast address of the Target node	M
Retrieved Update Phase	4	Retrieved Update Phase state of the Target node	M
Update Status	3	Status of the last operation with the Firmware Update Server	M
Transfer Status	4	Status of the last operation with the BLOB Transfer Server	M
Transfer Progress	6	Progress of the BLOB transfer in 2 percent increments	M
Update Firmware Image Index	8	Index of the firmware image on the Firmware Information List state that is being updated.	M

Table 5.20: The Target Node Entry field format

The Address field shall indicate the address of the Target node.

The Retrieved Update Phase field shall indicate the Retrieved Update Phase state of the Firmware Update Server identified by the Address field. The values for the Retrieved Update Phase field are defined in Table 4.8.

The Update Status field shall indicate the status of the last operation between the Firmware Update Client and the Firmware Update Server identified by the Address field. The values for the Update Status field are defined in Table 4.21.

The Transfer Status field shall indicate the status of the last operation between the BLOB Transfer Client and the BLOB Transfer Server identified by the Address field. The values for the Transfer Status field are defined in Section 3.4 in [10].

The Transfer Progress field shall indicate the progress of the firmware image transfer. The value is multiplied by 2 to calculate the percentage of the firmware image that has been delivered and confirmed. The range of allowed values for the Transfer Progress field is 0x0 to 0x32. The values 0x33 to 0x3F are Prohibited.

The Update Firmware Image Index field shall identify the index of the firmware image on the Firmware Information List state of the Target node that is being updated.

5.2.6 Firmware Distribution Capabilities Get

The Firmware Distribution Capabilities Get message is an acknowledged message sent by a Firmware Distribution Client to get the capabilities of a Firmware Distribution Server.

The response to a Firmware Distribution Capabilities Get message is a Firmware Distribution Capabilities Status message.

The structure of the Firmware Distribution Capabilities Get message is defined in Table 5.21.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.21: Firmware Distribution Capabilities Get message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Capabilities Get message defined in the Assigned Numbers document [9].

5.2.7 Firmware Distribution Capabilities Status

The Firmware Distribution Capabilities Status message is an unacknowledged message sent by a Firmware Distribution Server to report Distributor capabilities.

A Firmware Distribution Capabilities Status message is sent as a response to a Firmware Distributor Capabilities Get message.

The structure of the Firmware Distribution Capabilities Status message is defined in Table 5.22.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Max Distribution Receivers List Size	2	Maximum number of entries in the Distribution Receivers List state	M
Max Firmware Images List Size	2	Maximum number of entries in the Firmware Images List state	M
Max Firmware Image Size	4	Maximum size of one firmware image (in octets)	M

Field	Size (octets)	Description	Req.
Max Upload Space	4	Total space dedicated to storage of firmware images (in octets)	M
Remaining Upload Space	4	Remaining available space in firmware image storage (in octets)	M
Out-of-Band Retrieval Supported	1	Value of the Out-of-Band Retrieval Supported state	M
Supported URI Scheme Names	variable	Value of the Supported URI Scheme Names state	C.1

Table 5.22: Firmware Distribution Capabilities Status message structure

C.1: If the value of the Out-of-Band Retrieval Supported field is OOB Supported, then the Supported URI Scheme Names field shall be present; otherwise, the Supported URI Scheme Names field shall not be present.

The Opcode field shall contain the opcode value for the Firmware Distribution Capabilities Status message defined in the Assigned Numbers document [9].

The Max Distribution Receivers List Size field shall indicate the value of the Max Distribution Receivers List Size state.

The Max Firmware Images List Size field shall indicate the value of the Max Firmware Images List Size state.

The Max Firmware Image Size field shall indicate the value of the Max Firmware Image Size state.

The Max Upload Space field shall indicate the value of the Max Upload Space state.

The Remaining Upload Space field shall indicate the value of the Remaining Upload Space state.

The Out-of-Band Retrieval Supported field shall indicate the value of the Out-of-Band Retrieval Supported state.

If present, the Supported URI Scheme Names field shall indicate the value of the Supported URI Scheme Names state.

5.2.8 Firmware Distribution Get

The Firmware Distribution Get message is an acknowledged message sent by a Firmware Distribution Client to get the state of the firmware image distribution on a Firmware Distribution Server.

The response to a Firmware Distribution Get message is a Firmware Distribution Status message.

The structure of the Firmware Distribution Get message is defined in Table 5.23.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.23: Firmware Distribution Get message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Get message defined in the Assigned Numbers document [9].

5.2.9 Firmware Distribution Start

The Firmware Distribution Start message is an acknowledged message sent by a Firmware Distribution Client to start the firmware image distribution to the Target nodes in the Distribution Receivers List.

The response to a Firmware Distribution Start message is a Firmware Distribution Status message.

The structure of the Firmware Distribution Start message is defined Table 5.24.

Field	Size (bits)	Description	Req.
Opcode	16	The message opcode	M
Distribution AppKey Index	16	Index of the application key used in a firmware image distribution	M
Distribution TTL	8	TTL value used in a firmware image distribution	M
Distribution Timeout Base	16	Used to compute the timeout of the firmware image distribution	M
Distribution Transfer Mode	2	Mode of the transfer	M
Update Policy	1	Firmware update policy	M
RFU	5	Reserved for Future Use	M
Distribution Firmware Image Index	16	Index of the firmware image in the Firmware Images List state to use during firmware image distribution	M
Distribution Multicast Address	16 or 128	Multicast address used in a firmware image distribution	M

Table 5.24: Firmware Distribution Start message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Start message defined in the Assigned Numbers document [9].

The Distribution AppKey Index field shall indicate the application key used for the firmware image distribution.

The Distribution TTL field shall indicate the TTL value used for a firmware image distribution. The values for the Distribution TTL field are defined in Table 4.2.

The Distribution Timeout Base field shall contain the value that is used to calculate when firmware image distribution will be suspended.

The Distribution Transfer Mode field shall indicate the mode used to transfer the BLOB to the Target node (see Section 3.1.4 in [10]).

The Update Policy field shall indicate the update policy that the Firmware Distribution Server will use for this firmware image distribution.

The Distribution Firmware Image Index field shall identify the index of the firmware image from the Firmware Images List state to transfer during firmware image distribution.

The Distribution Multicast Address field shall identify the multicast address of the Target nodes that the firmware image is intended for. The value of the Distribution Multicast Address field shall be a group address, the Label UUID of a virtual address, or the Unassigned address.

5.2.10 Firmware Distribution Suspend

The Firmware Distribution Suspend message is an acknowledged message sent by a Firmware Distribution Client to suspend the firmware image distribution from a Firmware Distribution Server.

The response to a Firmware Distribution Suspend message is a Firmware Distribution Status message.

The structure of the Firmware Distribution Suspend message is defined in Table 5.25.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.25: Firmware Distribution Suspend message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Suspend message defined in the Assigned Numbers document [9].

5.2.11 Firmware Distribution Cancel

The Firmware Distribution Cancel message is an acknowledged message sent by a Firmware Distribution Client to stop the firmware image distribution from a Firmware Distribution Server.

The response to a Firmware Distribution Cancel message is a Firmware Distribution Status message.

The structure of the Firmware Distribution Cancel message is defined in Table 5.26.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.26: Firmware Distribution Cancel message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Cancel message defined in the Assigned Numbers document [9].

5.2.12 Firmware Distribution Apply

The Firmware Distribution Apply message is an acknowledged message sent from a Firmware Distribution Client to a Firmware Distribution Server to apply the firmware image on the Target nodes.

The response to a Firmware Distribution Apply message is a Firmware Distribution Status message.

The structure of the Firmware Distribution Apply message is defined in Table 5.27.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.27: Firmware Distribution Apply message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Apply message defined in the Assigned Numbers document [9].

5.2.13 Firmware Distribution Status

The Firmware Distribution Status message is an unacknowledged message sent by a Firmware Distribution Server to report the status of a firmware image distribution.

A Firmware Distribution Status message is sent as a response to a Firmware Distribution Get message, Firmware Distribution Start message, Firmware Distribution Suspend message, Firmware Distribution Cancel message, or Firmware Distribution Apply message.

The structure of the Firmware Distribution Status message is defined in Table 5.28.

Field	Size (bits)	Description	Req.
Opcode	16	The message opcode	M
Status	8	Status Code for the requesting message	M
Distribution Phase	8	Phase of the firmware image distribution	M

Field	Size (bits)	Description	Req.
Distribution Multicast Address	16	Multicast address used in firmware image distribution	O
Distribution AppKey Index	16	Index of an application key used in a firmware image distribution	C.1
Distribution TTL	8	TTL value used in a firmware image distribution	C.1
Distribution Timeout Base	16	Used to compute the timeout of the firmware image distribution	C.1
Distribution Transfer Mode	2	Mode of the transfer	C.1
Update Policy	1	Firmware update policy	C.1
RFU	5	Reserved for Future Use	C.1
Distribution Firmware Image Index	16	The index of the firmware image in the Firmware Images List state used during firmware image distribution	C.1

Table 5.28: Firmware Distribution Status message structure

C.1: If the Distribution Multicast Address field is present, then the Distribution AppKey Index field, Distribution TTL field, Distribution Timeout Base field, Distribution Transfer Mode field, Update Policy field, RFU field, and the Distribution Firmware Image Index field shall also be present; otherwise, the Distribution AppKey Index field, Distribution TTL field, Distribution Timeout Base field, Distribution Transfer Mode field, Update Policy field, RFU field and the Distribution Firmware Image Index field shall not be present.

The Opcode field shall contain the opcode value for the Firmware Distribution Status message defined in the Assigned Numbers document [9].

The Status field shall identify the Status Code for the last operation performed on the Firmware Distribution Server state. The status codes are defined in Table 4.22.

The Distribution Phase field shall indicate the Distribution Phase state.

If present, the Distribution Multicast Address field shall indicate the Distribution Multicast Address state. When using a Label UUID, the status message shall provide this value as a virtual address, as defined in the Mesh Protocol specification v1.1 [1].

If present, the Distribution AppKey Index field shall indicate the Distribution AppKey Index state.

If present, the Distribution TTL field shall indicate the Distribution TTL state.

If present, the Distribution Timeout Base field shall indicate the Distribution Timeout Base state.

If present, the Distribution Transfer Mode field shall indicate the Distribution Transfer Mode state.

If present, the Update Policy field shall indicate the Update Policy state.

If present, the Distribution Firmware Image Index field shall indicate Distribution Firmware Image Index state.

5.2.14 Firmware Distribution Upload Get

The Firmware Distribution Upload Get message is an acknowledged message sent by a Firmware Distribution Client to check the status of a firmware image upload to a Firmware Distribution Server.

The response to a Firmware Distribution Upload Get message is a Firmware Distribution Upload Status message.

The structure of the Firmware Distribution Upload Get message is defined in Table 5.29.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.29: Firmware Distribution Upload Get message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Upload Get message defined in the Assigned Numbers document [9].

5.2.15 Firmware Distribution Upload Start

The Firmware Distribution Upload Start message is an acknowledged message sent by a Firmware Distribution Client to start a firmware image upload to a Firmware Distribution Server.

The response to a Firmware Distribution Upload Start message is a Firmware Distribution Upload Status message.

The structure of the Firmware Distribution Upload Start message is defined in Table 5.30.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Upload TTL	1	TTL value used in a firmware image upload	M
Upload Timeout Base	2	Used to compute the timeout of the firmware image upload	M
Upload BLOB ID	8	BLOB identifier for the firmware image	M
Upload Firmware Size	4	Firmware image size (in octets)	M
Upload Firmware Metadata Length	1	Size of the Upload Firmware Metadata field	M
Upload Firmware Metadata	1 to 255	Vendor-specific firmware metadata	C.1
Upload Firmware ID	variable	The Firmware ID identifying the firmware image being uploaded	M

Table 5.30: Firmware Distribution Upload Start message structure

C.1: If the value of the Upload Firmware Metadata Length field is greater than 0, then the Upload Firmware Metadata field shall be present; otherwise, the Upload Firmware Metadata field shall be omitted.

The Opcode field shall contain the opcode value for the Firmware Distribution Upload Start message defined in the Assigned Numbers document [9].

The Upload TTL field shall indicate the TTL value used for a firmware image upload. The values for the Upload TTL field are defined in Table 4.2.

The Upload Timeout Base field shall contain the value that is used to calculate when firmware image upload will be suspended.

The Upload BLOB ID field shall field shall contain the BLOB identifier to use during firmware image upload.

The Upload Firmware Size field shall indicate the size of the firmware image being uploaded.

The Upload Firmware Metadata Length field shall indicate the length of the Upload Firmware Metadata field.

If present, the Upload Firmware Metadata field shall contain the custom data from the firmware vendor that will be used to check whether the firmware image can be updated.

The Upload Firmware ID field shall identify the firmware image being uploaded.

5.2.16 Firmware Distribution Upload OOB Start

The Firmware Distribution Upload OOB Start message is an acknowledged message sent by a Firmware Distribution Client to start a firmware image upload to a Firmware Distribution Server using an OOB mechanism.

The response to a Firmware Distribution Upload OOB Start message is a Firmware Distribution Upload Status message.

The structure of the Firmware Distribution Upload OOB Start message is defined in Table 5.31.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Upload OOB URI Length	1	Length of the Upload OOB URI field	M
Upload OOB URI	1 to 255	URI for the firmware image source	M
Upload OOB Firmware ID	variable	The Firmware ID value used to generate the URI query string	M

Table 5.31: Firmware Distribution Upload OOB Start message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Upload OOB Start message defined in the Assigned Numbers document [9].

The Upload OOB URI Length field shall indicate the length of the Upload OOB URI field. The value 0 is Prohibited.

The Upload OOB URI field shall contain the URI used to retrieve the firmware image.

The Upload OOB Firmware ID field shall contain the value used to generate the URI query string (see Section 3.2.1 and Section 3.3.1).

5.2.17 Firmware Distribution Upload Cancel

The Firmware Distribution Upload Cancel message is an acknowledged message sent by a Firmware Distribution Client to stop a firmware image upload to a Firmware Distribution Server.

The response to a Firmware Distribution Upload Cancel message is a Firmware Distribution Upload Status message.

The structure of the Firmware Distribution Upload Cancel message is defined in Table 5.32.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.32: Firmware Distribution Upload Cancel message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Upload Cancel message defined in the Assigned Numbers document [9].

5.2.18 Firmware Distribution Upload Status

The Firmware Distribution Upload Status message is an unacknowledged message sent by a Firmware Distribution Server to report the status of a firmware image upload.

A Firmware Distribution Upload Status message is sent as a response to a Firmware Distribution Upload Get message, Firmware Distribution Upload Start message, Firmware Distribution Upload OOB Start message, or Firmware Distribution Upload Cancel message.

The structure of the Firmware Distribution Upload Status message is defined in Table 5.33.

Field	Size (bits)	Description	Req.
Opcode	16	The message opcode	M
Status	8	Status Code for the requesting message	M
Upload Phase	8	Phase of the firmware image upload to a Firmware Distribution Server	M
Upload Progress	7	A percentage indicating the progress of the firmware image upload	O
Upload Type	1	A bit indicating whether the Upload is done in-band or OOB	C.1
Upload Firmware ID	variable	The Firmware ID identifying the firmware image being uploaded	C.2
Upload OOB Firmware ID	variable	The Firmware ID used to generate the URI query string	C.3

Table 5.33: Firmware Distribution Upload Status message structure

C.1: If the Upload Progress field is present, then the Upload Type field shall be present; otherwise, the Upload Type field shall be omitted.

C.2: If the Upload Type field is present, and its value is In-band, then the Upload Firmware ID field shall be present; otherwise, the Upload Firmware ID field shall be omitted.

C.3: If the Upload Type field is present, and its value is OOB, then the Upload OOB Firmware ID field shall be present; otherwise, the Upload OOB Firmware ID field shall be omitted.

The Opcode field shall contain the opcode value for the Firmware Distribution Upload Status message defined in the Assigned Numbers document [9].

The Status field shall identify the Status Code for the last operation performed on the Firmware Distribution Server state. The values for the Status field are defined in Table 4.22.

The Upload Phase field shall indicate the Upload Phase state.

If present, the Upload Progress field shall indicate the progress of the firmware upload as a percentage. The values 0x65 to 0xFF are Prohibited.

If present, the Upload Type field shall indicate the Upload Type state.

If present, the Upload Firmware ID field shall indicate the Upload Firmware ID state.

If present, the Upload OOB Firmware ID field shall indicate the Upload OOB Firmware ID state.

5.2.19 Firmware Distribution Firmware Get

The Firmware Distribution Firmware Get message is an acknowledged message sent by a Firmware Distribution Client to check whether a specific firmware image is stored on a Firmware Distribution Server.

The response to a Firmware Distribution Firmware Get message is a Firmware Distribution Firmware Status message.

The structure of the Firmware Distribution Firmware Get message is defined in Table 5.34.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Firmware ID	variable	The Firmware ID identifying the firmware image to check	M

Table 5.34: Firmware Distribution Firmware Get message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Firmware Get message defined in the Assigned Numbers document [9].

The Firmware ID field shall identify the firmware image to check.

5.2.20 Firmware Distribution Firmware Get By Index

The Firmware Distribution Firmware Get By Index message is an acknowledged message sent by a Firmware Distribution Client to check which firmware image is stored in a particular entry in the Firmware Images List state on a Firmware Distribution Server.

The response to a Firmware Distribution Firmware Get By Index message is a Firmware Distribution Firmware Status message.

The structure of the Firmware Distribution Firmware Get By Index message is defined in Table 5.35.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Distribution Firmware Image Index	2	Index of the entry in the Firmware Images List state	M

Table 5.35: Firmware Distribution Firmware Get By Index message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Firmware Get By Index message defined in the Assigned Numbers document [9].

The Distribution Firmware Image Index field shall indicate the index of the entry in the Firmware Images List state.

5.2.21 Firmware Distribution Firmware Delete

The Firmware Distribution Firmware Delete message is an acknowledged message sent by a Firmware Distribution Client to delete a stored firmware image on a Firmware Distribution Server.

The response to a Firmware Distribution Firmware Delete message is a Firmware Distribution Firmware Status message.

The structure of the Firmware Distribution Firmware Delete message is defined in Table 5.36.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Firmware ID	variable	Identifies the firmware image to delete	M

Table 5.36: Firmware Distribution Firmware Delete message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Firmware Delete message defined in the Assigned Numbers document [9].

The Firmware ID field shall identify the firmware image to be deleted from the Firmware Images List state.

5.2.22 Firmware Distribution Firmware Delete All

The Firmware Distribution Firmware Delete All message is an acknowledged message sent by a Firmware Distribution Client to delete all firmware images stored on a Firmware Distribution Server.

The response to a Firmware Distribution Firmware Delete All message is a Firmware Distribution Firmware Status message.

The structure of the Firmware Distribution Firmware Delete All message is defined in Table 5.37.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M

Table 5.37: Firmware Distribution Firmware Delete All message structure

The Opcode field shall contain the opcode value for the Firmware Distribution Firmware Delete All message defined in the Assigned Numbers document [9].

5.2.23 Firmware Distribution Firmware Status

The Firmware Distribution Firmware Status message is an unacknowledged message sent by a Firmware Distribution Server to report the status of an operation on a stored firmware image.

A Firmware Distribution Firmware Status message is sent in response to a Firmware Distribution Firmware Get message, a Firmware Distribution Firmware Get By Index message, a Firmware Distribution Firmware Delete message, or a Firmware Distribution Firmware Delete All message.

The structure of the Firmware Distribution Firmware Status message is defined in Table 5.38.

Field	Size (octets)	Description	Req.
Opcode	2	The message opcode	M
Status	1	Status Code for the requesting message	M
Entry Count	2	The number of firmware images stored on the Firmware Distribution Server	M
Distribution Firmware Image Index	2	Index of the firmware image in the Firmware Images List state	M
Firmware ID	variable	Identifies associated firmware image	C.1

Table 5.38: Firmware Distribution Firmware Status message structure

C.1: If 1) the Firmware Images List state lists the firmware image identified by the Distribution Firmware Image Index field; 2) the message is sent in response to a Firmware Distribution Firmware Get message; or 3) the message is sent in response to a successfully processed Firmware Distribution Firmware Delete message, then the Firmware ID field shall be present; otherwise, the Firmware ID field shall be omitted.

The Opcode field shall contain the opcode value for the Firmware Distribution Firmware Status message defined in the Assigned Numbers document [9].

The Status field shall identify the Status Code for the last operation performed on the Firmware Distribution Server state. The Status codes are defined in Table 4.22.

The Entry Count field shall indicate the number of firmware images listed in the Firmware Images List state.

The Distribution Firmware Image Index field shall indicate the index of the firmware image in the Firmware Images List state. The values from 0x0000 to 0xFFFE identify a firmware image in the Firmware Images List state. The value 0xFFFF indicates that the firmware image is not listed in the Firmware Images List state.

If present, the Firmware ID field shall contain the Firmware ID of the firmware image at the index in the Firmware Images List state that is identified by the Distribution Firmware Image Index field.

6 Firmware Update server models

6.1 Firmware Update Server model

6.1.1 Description

The Firmware Update Server model is used by the Target node to report the firmware images installed on the node and the location of new firmware images, and to initiate a firmware update in order to receive the new firmware. The firmware image will be transferred using the BLOB Transfer Server model (see Section 5 in [10]).

The Firmware Update Server model is a main model that extends the BLOB Transfer Server model (see Section 5 in [10]).

The access layer security on the Firmware Update Server model uses application keys.

This model shall support model subscription, as defined in Section 4.2.3 of the Mesh Protocol specification v1.1 [1].

The Firmware Update Server model adds the state instances listed in Table 6.1 and Table 6.2 and the messages listed in Table 6.3 to the model it extends, and requires one element: the Firmware Update Main element. The Firmware Update Main element contains the Firmware Update Server main model and all the models that the main model extends.

Table 6.1 defines whether the states from the Firmware Update Server model are stored with a scene.

State	Stored with Scene
Firmware Information List	No
Update Phase	No
Firmware Update Additional Information	No
Update Firmware Image Index	No
New Firmware Image	No
Update BLOB ID	No
Update Firmware Metadata	No
Update TTL	No
Update Server Timeout Base	No

Table 6.1: Whether Firmware Update Server states are stored with a scene

Table 6.2 illustrates the state bindings between the Firmware Update Server model states and the states of the models that the Firmware Update Server extends.

State	Bound State	
	Model	State
Firmware Information List	–	–
Update Phase	–	–
Firmware Update Additional Information	–	–

State	Bound State	
	Model	State
Update Firmware Image Index	–	–
New Firmware Image	–	–
Update BLOB ID	–	–
Update Firmware Metadata	–	–
Update TTL	–	–
Update Server Timeout Base	–	–

Table 6.2: Firmware Update Server states and bindings

Table 6.3 lists the Firmware Update Server model messages.

Element	Model Name	State	Message	Rx	Tx
Firmware Update Main	Firmware Update Server	Firmware Information List	Firmware Update Information Get	M	–
			Firmware Update Information Status	–	M
		Update Firmware Metadata	Firmware Update Firmware Metadata Check	M	–
			Firmware Update Firmware Metadata Status	–	M
		Update Phase, Firmware Update Additional Information, Update Firmware Image Index, Update BLOB ID, Update Firmware Metadata, Update TTL, Update Server Timeout Base	Firmware Update Get	M	–
			Firmware Update Start	M	–
			Firmware Update Cancel	M	–
			Firmware Update Apply	M	–
			Firmware Update Status	–	M

Table 6.3: Firmware Update Server messages

6.1.2 Procedures

Figure 6.1 shows a diagram of the Update Phase transitions of the Firmware Update Server model during the procedures.

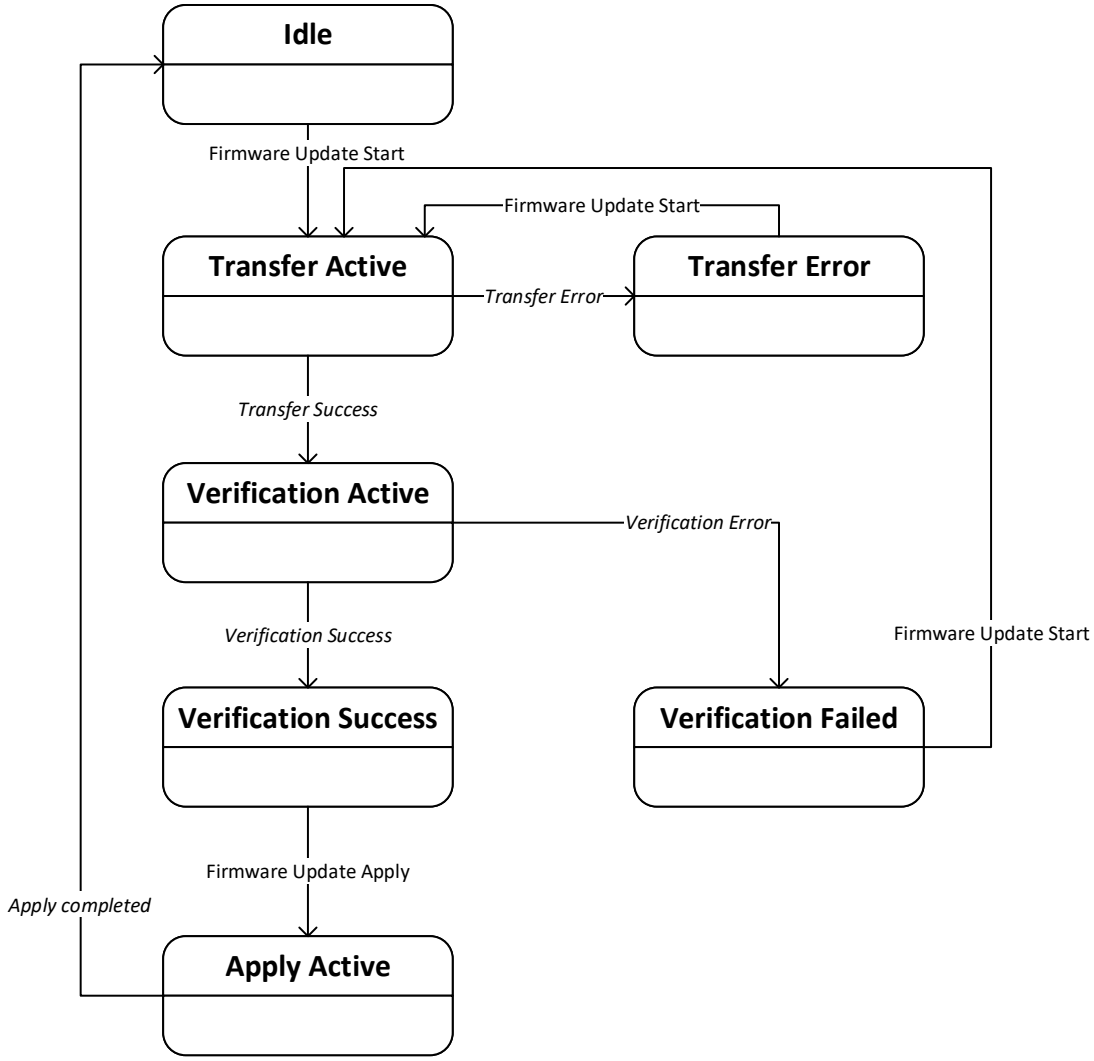

Figure 6.1: The Update Phase transitions

6.1.2.1 Receive Firmware procedure

The Receive Firmware procedure is executed on a Firmware Update Server to start the firmware image transfer during a firmware update.

As inputs, the Receive Firmware procedure takes an Update BLOB ID, an Update Timeout, and an Update TTL.

This procedure can be canceled at any time by a higher-layer model or application. When the procedure is canceled while the Initialize And Receive BLOB procedure is being executed or is stopped, the server shall cancel the Initialize And Receive BLOB procedure.

The Receive Firmware procedure consists of the following steps:

1. The Firmware Update Server shall set the Update BLOB ID state to the value of Update BLOB ID input.
2. The server shall start the Initialize And Receive BLOB procedure from the BLOB Transfer Server model with the following inputs, and shall wait for the Initialize And Receive BLOB procedure to complete or stop:
    - The Update BLOB ID state shall be the Update BLOB ID input.

- The Update Timeout input of the Receive Firmware procedure shall be the Timeout input for the Initialize And Receive BLOB procedure.
- The Update TTL of the Receive Firmware procedure shall be the Transfer TTL input for the Initialize And Receive BLOB Procedure.

3. The server shall perform the following actions:
   a. The server shall store the BLOB output from the Initialize And Receive BLOB procedure in the New Firmware Image state.
   b. The Update Phase state shall be set to Verifying Update.
   c. The server shall start the Verify Firmware procedure.

The Receive Firmware procedure completes if any of the following conditions are met:

- If the Verify Firmware procedure is started, then the Receive Firmware procedure completes successfully.
- If the Initialize And Receive BLOB procedure does not complete successfully or is stopped, then the Firmware Update Server model shall set the Update Phase state to Transfer Error, and the Receive Firmware procedure fails.

6.1.2.2 Verify Firmware procedure

The Verify Firmware procedure is executed on a Firmware Update Server to check the integrity of the firmware image stored in the New Firmware Image state.

The Verify Firmware procedure can be canceled at any time by a higher-layer model or application.

The Firmware Update Server model shall verify the firmware image stored in the New Firmware state. The verification method is implementation specific.

If the verification completes successfully, then the server shall set the Update Phase state to Verification Succeeded, and the procedure completes successfully.

If the verification does not complete successfully, then the server shall set the Update Phase state to Verification Failed, and the procedure fails.

6.1.2.3 Apply New Firmware procedure

The Apply New Firmware procedure is executed on a Firmware Update Server to install the firmware image stored in the New Firmware Image state.

The server shall set the Update Phase state to Applying Update and shall begin installing the new firmware image.

When firmware image installation completes successfully, the Firmware Update Server performs one of the following actions based on the value of the Firmware Update Additional Information state:

- If the Firmware Update Additional Information state value is CD Unchanged (0x0), then no additional actions are taken.
- If the Firmware Update Additional Information state value is CD Changed and RPR Unsupported (0x1), then the node shall use the composition data from the installed firmware image on Composition Data Page 0. The composition data from the old firmware image is discarded.
- If the Firmware Update Additional Information state value is CD Changed and RPR Supported (0x2), then the node shall use the composition data from the old firmware image on Composition Data Page 0 and shall use the composition data from the installed firmware image on Composition Data Page 128.

- If the Firmware Update Additional Information state value is Device Unprovisioned (0x3), then the node shall perform the implementation-specific actions needed to become unprovisioned.

Whether the firmware image is installed successfully or the installation fails, the server shall set the Update Phase to Idle after firmware image installation is completed.

6.1.3 Behavior

When the Firmware Update Server checks error conditions while processing a received message, the server shall check the error conditions one by one starting with the first item. This applies to the error handling behavior in Sections 6.1.3.1 through 6.1.3.9.

6.1.3.1 Receiving a Firmware Update Information Get message

When a Firmware Update Server receives a Firmware Update Information Get message, the Firmware Update Server shall respond with a Firmware Update Information Status message with the First Index field set to the value of the First Index field in the Firmware Update Information Get message.

6.1.3.2 Sending a Firmware Update Information Status message

A Firmware Update Server sends a Firmware Update Information Status message in response to a Firmware Update Information Get message.

The Firmware Information List Count field shall be set to the number of entries in the Firmware Information List state.

The Firmware Information List field shall be formatted as follows based on the value of the First Index field of the received message:

- If the First Index field identifies an existing entry in the Firmware Information List state, then the server shall select entries from the Firmware Information List state consecutively starting with the entry indicated by the value of the First Index field. The number of entries that are returned shall not exceed the number indicated by the Entries Limit field of the received message.
- If the First Index field indicates an entry that does not exist in the Firmware Information List state, then the Firmware Information List field shall be omitted.

Each Firmware Information Entry field shall be set to the corresponding entry in the Firmware Information List state as follows:

- The Update URI field of the message shall be set to the value of the Update URI state.
- The Update URI Length field of the message shall indicate the length of the Update URI.
- The Current Firmware ID field of the message shall be set to the value of the Current Firmware ID state.
- The Current Firmware ID Length field of the message shall indicate the length of the Current Firmware ID.

6.1.3.3 Receiving a Firmware Update Get message

Get firmware info. When a Firmware Update Server receives a Firmware Update Get message, the server shall respond with a Firmware Update Status message with the Status field set to Success.

Error handling. When a Firmware Update Server receives a Firmware Update Get message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Update Server shall respond with a Firmware Update Status message with the Status field set to Internal Error.

6.1.3.4 Receiving a Firmware Update Start message

When a Firmware Update Server receives a Firmware Update Start message, and the Update Phase state is Idle, Transfer Error, or Verification Failed, and the message is successfully processed (i.e., it does not result in any error condition listed in Table 6.4), then the server shall perform the following steps.

New firmware image. If either the Incoming Firmware Metadata field is not present in the Firmware Update Start message, or the Incoming Firmware Metadata field is present and the Firmware Update Server does not have a copy of the firmware image (i.e., the metadata and firmware image index combination does not indicate a firmware image that has been received and cached previously), then the server shall perform the following steps:

1. The server shall set the following states:
    - The Update Phase state shall be set to Transfer Active.
    - The Update Firmware Metadata state shall be set to the value of the Incoming Firmware Metadata field.
    - The Update TTL state, the Update Server Timeout Base state, the Update BLOB ID state, and the Update Firmware Image Index state shall be set to the value of the corresponding fields.
2. The server shall respond with a Firmware Update Status message with the Status field set to Success.
3. The server shall start the Receive Firmware procedure with the following inputs: Update BLOB ID state, Update Server Timeout Base state, and Update TTL state.

Firmware image partially received. If all of the following conditions are true:

- The Incoming Firmware Metadata field is not present in the Firmware Update Start message or the Incoming Firmware Metadata field is equal to the Update Firmware Metadata state.
- The Update Phase state is equal to Transfer Error.
- The Update TTL state, the Update Server Timeout Base state, the Update BLOB ID state, and the Update Firmware Image Index state are equal to the corresponding fields of the message.

(i.e., the message fields indicate that a previously suspended transfer is being resumed), then the server shall perform the following steps:

1. The server shall set the Update Phase state to Transfer Active.
2. The server shall respond with a Firmware Update Status message with the Status field set to Success.

The BLOB Transfer Server resumes receiving the firmware image.

Firmware image already received. If the Incoming Firmware Metadata field is present in the Firmware Update Start message, and the Firmware Update server has a copy of the firmware image (i.e., the metadata and firmware image index combination indicates a firmware image that has been received and cached previously, or the Distributor is updating itself and a copy of the firmware image is already available), then the server shall perform the following steps:

1. The server shall set the following states:
    - The Update Phase state shall be set to Verifying Update.
    - The Update Firmware Metadata state shall be set to the value of the Incoming Firmware Metadata field.

- The Update TTL state, the Update Server Timeout Base state, the Update BLOB ID state, and the Update Firmware Image Index state shall be set to the value of the corresponding fields.

2. The server sends a Firmware Update Status message with the Status field set to Success.
3. The server stores the identified firmware image in the New Firmware Image state.
4. The server executes the Verify Firmware procedure (see Section 6.1.2.2).

Error handling. When the Firmware Update server receives a Firmware Update Start message, and the Update Phase state of the server is Idle, Transfer Error, or Verification Failed, and the message is not successfully processed (i.e., it results in any error condition listed in Table 6.4), the server shall respond with a Firmware Update Status message with the Status field set to the corresponding status code defined in Table 6.4.

Error Condition	Status Code Name (see Table 4.21)
The BLOB Transfer Server is executing some other procedure	BLOB Transfer Busy
The firmware metadata check failed	Metadata Check Failed
The server cannot allocate necessary resources to receive the firmware image	Insufficient Resources
A condition on the server is preventing the update (e.g., low battery level)	Temporarily Unavailable
The Firmware Information List state does not contain an entry corresponding to the value of the Update Firmware Image Index field	Wrong Firmware Index

Table 6.4: Error conditions for the Firmware Update Start message

When a Firmware Update Server receives a Firmware Update Start message, and the message is not processed successfully because an implementation-specific error (e.g., a memory allocation error, or flash memory error) has occurred, the Firmware Update Server shall respond with the Firmware Update Status message with the Status field set to Internal Error.

Idempotency. When the server receives a Firmware Update Start message, and the Update Phase state of the server is Transfer Active, Verifying Update, Verification Succeeded, or Applying Update, and the message is successfully processed (i.e., it does not result in any error condition specified in Table 6.5), the server shall respond with a Firmware Update Status message with the Status field set to Success.

When the server receives a Firmware Update Start message, and the Update Phase state of the server is Transfer Active, Verifying Update, Verification Succeeded, or Applying Update, and the message is not successfully processed (i.e., it results in any error condition specified in Table 6.5), the server shall respond with the Firmware Update Status message with the Status field set to the corresponding status code defined in Table 6.5.

Error Condition	Status Code Name (see Table 4.21)
The Update TTL field does not match the Update TTL state	Wrong Phase
The Update Timeout Base field does not match the Update Server Timeout Base state	Wrong Phase
The Update BLOB ID field does not match the Update BLOB ID state	Wrong Phase

Error Condition	Status Code Name (see Table 4.21)
The Incoming Firmware Metadata field does not match the Update Firmware Metadata state	Wrong Phase
The Update Firmware Image Index field does not match the Update Firmware Image Index state	Wrong Phase

Table 6.5: Conditions used in idempotency checks when a Firmware Update Start message is received

6.1.3.5 Receiving a Firmware Update Cancel message

Cancel update. When a Firmware Update Server receives a Firmware Update Cancel message, the server shall cancel the Receive Firmware procedure, shall cancel the Verify Firmware procedure, and shall set the Update Phase state to Idle. The server shall respond with a Firmware Update Status message with the Status field set to Success.

Error handling. When a Firmware Update Server receives a Firmware Update Cancel message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Update Server shall respond with a Firmware Update Status message with the Status field set to Internal Error.

6.1.3.6 Receiving a Firmware Update Apply message

Apply firmware. When a Firmware Update Server receives a Firmware Update Apply message, and the Update Phase state is Verification Succeeded, and the new firmware can be applied immediately, the server shall set the Update Phase state to Applying Update. Then, the server shall respond with a Firmware Update Status message with the Status field set to Success and should wait until the Firmware Update Status has been fully transmitted (including all its retransmissions, if any) and shall start the Apply New Firmware procedure.

Error handling. When the Firmware Update Server receives a Firmware Update Apply message, and the Update Phase state is Verification Succeeded, and the new firmware cannot be applied (e.g., the battery level is too low), the server shall respond with a Firmware Update Status message with the Status field set to Temporarily Unavailable.

When the Firmware Update Server receives a Firmware Update Apply message, and the Update Phase state is not either Verification Succeeded or Applying Update, the server shall respond with a Firmware Update Status message with the Status field set to Wrong Phase.

When a Firmware Update Server receives a Firmware Update Apply message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Update Server shall respond with a Firmware Update Status message with the Status field set to Internal Error.

Idempotency. When the Firmware Update Server receives a Firmware Update Apply message, and the Update Phase state is Applying Update, the server shall respond with a Firmware Update Status message with the Status field set to Success.

6.1.3.7 Sending a Firmware Update Status message

A Firmware Update Server sends a Firmware Update Status message in response to a Firmware Update Get message, a Firmware Update Start message, a Firmware Update Cancel message, or a Firmware Update Apply message.

When the Update Phase state is Idle, the Firmware Update Server shall set the Update Phase field of the Firmware Update Status message to the value of the Update Phase state, and shall omit the Update TTL field, the Additional Information field, the Update Timeout Base field, the Update BLOB ID field, and the Update Firmware Image Index field.

When the Update Phase state is not Idle, the Firmware Update Server shall set the Update Phase field, the Additional Information field, the Update TTL field, the Update Timeout Base field, the Update BLOB ID field, and the Update Firmware Image Index field to the value of the corresponding states.

6.1.3.8 Receiving a Firmware Update Firmware Metadata Check message

Check metadata. When the Firmware Update Server receives a Firmware Update Firmware Metadata Check message, and the message is processed successfully (i.e., it does not result in any error condition listed in Table 6.6), the server shall respond with a Firmware Update Firmware Metadata Status message with the Status field set to Success.

Error handling. When the Firmware Update Server receives a Firmware Update Firmware Metadata Check message, and the message is not processed successfully (i.e., it results in an error condition listed in Table 6.6), the server shall respond with a Firmware Update Firmware Metadata Status message with the Status field set to the corresponding status code defined in Table 6.6.

Error Condition	Status Code Name (see Table 4.21)
The Firmware Metadata field is not present, and the implementation-specific validation algorithm requires that this field be present	Metadata Check Failed
The Firmware Metadata field is present, and the firmware is not accepted	Metadata Check Failed
The Firmware Information List state does not contain an entry corresponding to the value of the Update Firmware Image Index field	Wrong Firmware Index

Table 6.6: Error conditions for the Firmware Update Firmware Metadata Check message

6.1.3.9 Sending a Firmware Update Firmware Metadata Status message

A Firmware Update Server sends a Firmware Update Firmware Metadata Status message in response to a Firmware Update Firmware Metadata Check message.

The Firmware Update Server shall set the Update Firmware Image Index field to the value of the corresponding field in the received message and shall set the Additional Information field to the Firmware Update Additional Information state.

6.2 Firmware Distribution Server model

6.2.1 Description

The Firmware Distribution Server model is 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 uses the Firmware Update Client model and the BLOB Transfer Client model to manage the firmware update. The Firmware Distribution Server can transfer one firmware image at a time.

The Firmware Distribution Server model is a main model that extends the BLOB Transfer Server model (see Section 5 in [10]). The Firmware Distribution Server also has the corresponding Firmware Update Client model (see Section 7.1) and the BLOB Transfer Client model (see Section 6 in [10]) that shall also be present on the same element as the main model.

The access layer security on the Firmware Distribution Server model uses application keys.

This model shall support model subscription, as defined in Section 4.2.3 of the Mesh Protocol specification v1.1 [1].

The Firmware Distribution Server model adds the state instances listed in Table 6.7 and Table 6.8 and the messages listed in Table 6.9 to the model it extends, and requires one element: the Firmware Distribution Main element. The Firmware Distribution Main element contains the Firmware Distribution Server main model and all the models that the main model extends.

Table 6.7 defines whether the states from the Firmware Distribution Server model are stored with a scene.

State	Stored with Scene
Distribution Receivers	No
Distributor Capabilities	No
Distribution Parameters	No
Upload Parameters	No
Firmware Images List	No

Table 6.7: Whether Firmware Distribution Server states are stored with a scene

Table 6.8 illustrates the state bindings between the Firmware Distribution Server model states and the states of the models that the Firmware Distribution Server extends.

State	Bound State	
	Model	State
Distribution Receivers	–	–
Distributor Capabilities	–	–
Distribution Parameters	–	–
Upload Parameters	–	–
Firmware Images List	–	–

Table 6.8: Firmware Distribution Server states and bindings

Table 6.9 lists the Firmware Distribution Server model messages.

Element	Model Name	State	Message	Rx	Tx
Firmware Distribution Main	Firmware Distribution Server	Distribution Receivers	Firmware Distribution Receivers Add	M	–
			Firmware Distribution Receivers Delete All	M	–
			Firmware Distribution Receivers Status	–	M
			Firmware Distribution Receivers Get	M	–
			Firmware Distribution Receivers List	–	M
		Distributor Capabilities	Firmware Distribution Capabilities Get	M	–
			Firmware Distribution Capabilities Status	–	M
		Distribution Parameters	Firmware Distribution Get	M	–
			Firmware Distribution Start	M	–
			Firmware Distribution Suspend	M	–
			Firmware Distribution Cancel	M	–
			Firmware Distribution Apply	M	–
			Firmware Distribution Status	–	M
		Upload Parameters	Firmware Distribution Upload Get	M	–
			Firmware Distribution Upload Start	M	–
			Firmware Distribution Upload Start OOB	M	–
			Firmware Distribution Upload Cancel	M	–

Element	Model Name	State	Message	Rx	Tx
			Firmware Distribution Upload Status	–	M
		Firmware List	Firmware Distribution Firmware Get	M	–
			Firmware Distribution Firmware Get By Index	M	–
			Firmware Distribution Firmware Delete	M	–
			Firmware Distribution Firmware Delete All	M	–
			Firmware Distribution Firmware Status	–	M

Table 6.9: Firmware Distribution Server messages

6.2.2 Procedures

6.2.2.1 Store Firmware procedure

The Store Firmware procedure is executed on a Firmware Distribution Server to store the new firmware image. The firmware image is transferred using the Initialize And Receive BLOB procedure defined in Section 5.2.1 in [10].

The Store Firmware procedure can be canceled at any time by a higher-layer model or application. When the procedure is canceled and the Initialize And Receive BLOB procedure is being executed or is paused, the server shall cancel the Initialize And Receive BLOB procedure.

The Store Firmware procedure consists of the following steps:

1. If the BLOB Transfer Server is not executing any other procedure, the Firmware Distribution Server shall start the Initialize And Receive BLOB procedure from the BLOB Transfer Server model with the following inputs and wait for it to complete:

   - The BLOB ID input is set to the Upload BLOB ID state.
   - The Timeout input is set to the Upload Timeout state.
   - The Transfer TTL input is set to the Upload TTL state.

   If the BLOB Transfer Server is executing any procedure when the Store Firmware procedure starts, then the Firmware Distribution Server shall set the Upload Phase state to Transfer Error and complete the Store Firmware procedure.

2. If the Initialize And Receive BLOB procedure completes successfully, then the Firmware Distribution Server shall add a new entry in the Firmware List state. The Firmware Distribution Server shall set the following states:

   - The server shall set the fields in the new Firmware List entry as follows:
     - › The Index field shall be set to the smallest available value, based on the position of the new firmware image in the Firmware List state. New firmware images are added to the end of the list.
     - › The Firmware ID field shall be set to the value of the Upload Firmware ID state.
     - › The Firmware Metadata field shall be set to the value of the Upload Firmware Metadata state.
   - The server shall decrease the Remaining Upload Space state by the amount of space that was used to store the firmware image.
   - The server shall set the Upload Phase state to Transfer Success.

When the Initialize And Receive BLOB procedure fails or is stopped, the Firmware Distribution Server shall set the Upload Phase state to Transfer Error and the Store Firmware procedure shall fail.

6.2.2.2 Check Firmware OOB procedure

The Check Firmware OOB procedure is executed on the Firmware Distribution Server to check if a new firmware image is available OOB.

As inputs, the procedure takes an Update URI and a Firmware ID.

As output, the procedure gives the Firmware ID of a new firmware image, if one is available.

The Check Firmware OOB procedure consists of the following steps:

The Update URI input is used to check for availability of a new firmware image. The check is performed based on the URI scheme that is used in the Update URI input:

- If the Update URI input uses the https scheme, then the Firmware Distribution Server shall execute the Firmware Check Over HTTPS procedure, described in Section 3.2, using the Update URI and the Firmware ID as inputs.
- If the Update URI input is not using the https scheme, then the check is implementation specific, and the retrieved new firmware description file should use the file format defined in Section 3.1.

The Check Firmware OOB procedure completes if any of the following conditions are met:

- If the Firmware Check Over HTTPS procedure returns code 200 OK, then the Check Firmware OOB procedure completes successfully, and returns the new image's Firmware ID from the firmware description file.
- If the Firmware Check Over HTTPS procedure returns code 404 Not Found, then the Check Firmware OOB procedure completes successfully, and returns no new Firmware ID.
- If the Firmware Check Over HTTPS procedure fails, then the Check Firmware OOB procedure fails.
- If the vendor-specific check completes successfully and a new firmware image is available, then the procedure completes successfully and returns a new Firmware ID.
- If the vendor-specific check completes successfully and a new firmware image is not available, then the procedure completes successfully and returns no new Firmware ID.
- If the vendor-specific check fails, then the Check Firmware OOB procedure fails.

6.2.2.3 Store Firmware OOB procedure

The Store Firmware OOB procedure is executed on the Firmware Distribution Server to download a new firmware image by using an OOB mechanism.

The Store Firmware OOB procedure can be canceled at any time by a higher-layer model or application. When the procedure is canceled, the server shall cancel the OOB download process that is in progress. When the Upload OOB URI is using the https schema, the Firmware Retrieval Over HTTPS procedure is canceled as described in Section 3.3. When the Upload OOB URI is not using the https schema, the cancel behavior is implementation specific.

The Store Firmware OOB procedure consists of the following steps:

1. The Firmware Distribution Server shall set the Upload Phase state to Transfer Active and the Upload Type state to Out-of-band.

2. The server shall perform either of the following actions based on the value of the Upload OOB URI:
    - If the Upload OOB URI uses the https scheme (the URI schemes are defined in [9]), then the server shall download a firmware image using the Firmware Retrieval Over HTTPS procedure (see Section 3.3).
    - If the Upload OOB URI does not use the https scheme, then the download process is implementation specific, and the retrieved firmware image should use the firmware archive file format (see Section 3.1).
3. When the OOB download process completes successfully, the server shall add a new entry in the Firmware List state. The Firmware Distribution Server shall set the following states:
    - The server shall set the fields in the new Firmware List entry as follows:
        - › The Index field shall be set to the smallest available value, based on the position of the new firmware image in the Firmware List state. New firmware images are added to the end of the list.
        - › The Firmware ID field shall be set to the value of the Upload Firmware ID state.
        - › The Firmware Metadata field shall be set to the value of the Upload Firmware Metadata state.
    - The server shall decrease the Remaining Upload Space state by the amount of space that was used to store the firmware image.
    - The server shall set the Upload Phase state to Transfer Success.

When the OOB download process fails, the Firmware Distribution server shall set the Upload Phase state to Transfer Error, and the Store Firmware OOB procedure fails.

6.2.2.4 Distribute Firmware procedure

The Distribute Firmware procedure is executed on a Firmware Distribution Server to distribute a firmware image to one or more Target nodes. The firmware image is transferred using the Transfer BLOB procedure defined in Section 6.2.3 in [10].

The Distribute Firmware procedure can be canceled at any time by a higher-layer model or application. When the procedure is canceled, the server shall cancel all procedures started by the Distribute Firmware procedure shown in the following list below:

- Start Firmware Update On Target Nodes procedure from the Firmware Update Client model
- Transfer BLOB procedure from the BLOB Transfer Client model
- Get Active Update Receivers procedure from the Firmware Update Client model
- Apply Firmware On Target Nodes procedure from the Firmware Update Client model
- Retrieve Firmware Information procedure from the Firmware Update Client model

This procedure consists of the following consecutive steps:

1. Initiate
2. Transfer
3. Refresh
4. Apply

5. Confirm

The Apply and Confirm steps are executed immediately by the Firmware Distribution Server when the Update Policy state is equal to Verify And Apply.

Additionally, the Distribute Firmware procedure has a Resume step, which is executed if the Transfer step is suspended and then later is resumed. After the Resume step is executed, the rest of the procedure continues from the Transfer step.

Initiate step. During the Initiate step, the Firmware Distribution Server shall set the Distribution Phase state to Transfer Active and shall generate an 8-octet BLOB ID value to use as an input in the Transfer BLOB procedure. The BLOB ID should be unique, and the method of generating the BLOB ID value is implementation specific.

Then, the server shall start the Start Firmware Update On Target Nodes procedure with the following inputs and shall wait for the procedure to complete:

- The Update Multicast Address input is set to the Distribution Multicast Address state.
- The Receivers List input is set to Distribution Receivers List state.
- The Update AppKey Index input is set to the Distribution AppKey Index state.
- The Update TTL input is set to the Distribution TTL state.
- The Update Timeout Base input is set to the Distribution Timeout Base state.
- The Update BLOB ID input is set to the BLOB ID value.
- The Firmware Metadata input is set to the Firmware Metadata from the entry in the Firmware List state identified by the Distribution Firmware Image Index state.

The result of the Start Firmware Update On Target Nodes procedure is one of the following:

- If the Start Firmware Update On Target Nodes procedure completes successfully, then the Distribute Firmware procedure proceeds to the Transfer step.
- If the Start Firmware Update On Target Nodes procedure fails, then the server shall set the Distribution Phase state to Failed, and the Distribute Firmware procedure fails.

Transfer step. During the Transfer step, the Firmware Distribution Server shall start the Transfer BLOB procedure from the BLOB Transfer Client model with the following inputs and shall wait for the procedure to complete:

- The Unicast Addresses input is set to the list of unicast addresses of the receivers retrieved from the Get Active Update Receivers procedure.
- The BLOB Multicast Address input is set to the Distribution Multicast Address state.
- The AppKey Index input is set to the Distribution AppKey Index state.
- The Transfer TTL input is set to the Distribution TTL state.
- The BLOB ID input is set to the BLOB ID value.
- The BLOB Data input is set to the firmware image identified by the Distribution Firmware Image Index state.
- The Transfer Mode input is set to the Distribution Transfer Mode state.
- The Timeout Base input is set to the Distribution Timeout Base state.

The Transfer step may be suspended when the Transfer BLOB procedure from the BLOB Transfer Client model can be suspended. The suspended Transfer step may be resumed by executing the Resume step. When the Transfer BLOB procedure from the BLOB Transfer Client model is suspended, the Transfer step shall be suspended, and the Distribution Phase state shall be set to the Transfer Suspended value.

The result of the Transfer BLOB procedure is one of the following:

- If the Transfer BLOB procedure completes successfully, then the Distribute Firmware procedure proceeds to the Refresh step.
- If the Transfer BLOB procedure fails, then the server shall set the Distribution Phase state to Failed, and the Distribute Firmware procedure fails.

Refresh step. During the Refresh step, the Firmware Distribution Server shall start the Refresh Target Nodes Status procedure with the following inputs and shall wait for the procedure to complete:

- The Update Multicast Address input is set to the Distribution Multicast Address state.
- The Nodes List input is set to the list of unicast addresses of the receivers retrieved from the Get Active BLOB Receivers procedure.
- The Update AppKey Index input is set to the Distribution AppKey Index state.
- The Update Timeout Base input is set to the Distribution Timeout Base state.
- The Update TTL input is set to the Distribution TTL state.

The result of the Refresh Target Nodes Status procedure is one of the following:

- If the Refresh Target Nodes Status procedure completes successfully, and the Update Policy state is Verify Only, then the Firmware Distribution Server shall set the Distribution Phase state to Transfer Success, and the Distribute Firmware procedure completes successfully.
- If the Refresh Target Nodes Status procedure completes successfully, and the Update Policy state is Verify And Apply, then the Firmware Distribution server shall set the Distribution Phase state to Applying Update, and the Distribute Firmware procedure proceeds to the Apply step.
- If the Refresh Target Nodes Status procedure fails, then the server shall set the Distribution Phase state to Failed, and the Distribute Firmware procedure fails.

Apply step. During the Apply step, the Firmware Distribution Server shall start the Apply Firmware On Target Nodes procedure with the following inputs and shall wait for the procedure to complete:

- The Update Multicast Address input is set to the Distribution Multicast Address state.
- The Nodes List input is set to the list of unicast addresses of the receivers retrieved from the Get Active Update Receivers procedure.
- The Update AppKey Index input is set to the Distribution AppKey Index state.
- The Update Timeout Base input is set to the Distribution Timeout Base state.
- The Update TTL input is set to the Distribution TTL state.

The result of the Apply Firmware On Target Nodes procedure is one of the following:

- If the Apply Firmware On Target Nodes procedure completes successfully, then the Distribute Firmware procedure shall proceed to the Confirm step.

- If the Apply Firmware On Target Nodes procedure fails, then the server shall set the Distribution Phase state to Failed and the Distribute Firmware procedure fails.

Confirm step. During the Confirm step, the Firmware Distribution Server shall wait until the Target nodes have applied the new firmware update and then shall start the Retrieve Firmware Information procedure with the following inputs and shall wait for the procedure to complete:

- The Update Multicast Address input is set to the Distribution Multicast Address state.
- The Nodes List input is set to the list of unicast addresses of the receivers retrieved from the Get Active Update Receivers procedure.
- The Update AppKey Index input is set to the Distribution AppKey Index state.
- The Update Timeout Base input is set to the Distribution Timeout Base state.
- The Update TTL input is set to the Distribution TTL state.

The server shall start the Confirm Update On Target Nodes procedure with the following inputs and wait for the procedure to complete:

- The Firmware ID input is set to the Firmware ID of the entry in the Firmware List state identified by the Distribution Firmware Image Index state.
- The Retrieved Information List input is set to the output (the Result List) of the Retrieve Firmware Information procedure.

The result of the Confirm Update On Target Nodes procedure is one of the following:

- If the Confirm Update On Target Nodes procedure completes successfully, then the server shall set the Distribution Phase to Completed, and the Distribute Firmware procedure completes successfully.
- If the Confirm Update On Target Nodes procedure fails, then the server shall set the Distribution Phase state to Failed, and the Distribute Firmware procedure fails.

Resume step. During the Resume step, the Firmware Distribution Server shall start the Start Firmware Update On Target Nodes procedure with the following inputs and shall wait for the procedure to complete:

- The Update Multicast Address input is set to the Distribution Multicast Address state.
- The Receivers List input is set to Distribution Receivers List state.
- The Update AppKey Index input is set to the Distribution AppKey Index state.
- The Update TTL input is set to the Distribution TTL state.
- The Update Timeout Base input is set to the Distribution Timeout Base state.
- The Update BLOB ID input is set to the BLOB ID value.
- The Firmware Metadata input is set to the Firmware Metadata from the entry in the Firmware List state identified by the Distribution Firmware Image Index state.

The result of the Start Firmware Update On Target Nodes procedure is one of the following:

- If the Start Firmware Update On Target Nodes procedure completes successfully, then the Transfer step shall be resumed by resuming the Transfer BLOB procedure.

- If the Start Firmware Update On Target Nodes procedure fails, then the server shall set the Distribution Phase state to Failed, and the Distribute Firmware procedure fails.

6.2.3 Behavior

When the Firmware Distribution Server checks the error conditions while processing a received message, the server shall check the error conditions one by one starting with the first item. This applies to the error handling behavior in Sections 6.2.3.1 through 6.2.3.22.

6.2.3.1 Receiving a Firmware Distribution Receivers Add message

When a Firmware Distribution Server receives a Firmware Distribution Receivers Add message, and the Distribution Phase state of the server is Idle, the server shall determine the set of receivers that are contained in the message and that are not already contained in the Distribution Receivers List state (i.e., New Receivers), then perform one of the following steps.

- **Adding receivers.** If the sum the Distribution Receivers List Count state value and the number of New Receivers is less than or equal to the value of the Max Distribution Receivers List Size state, then the server shall add the New Receivers to the Distribution Receivers List state, and shall respond with a Firmware Distribution Receivers Status message with the Status field set to Success.

- **Cannot add receivers.** If the sum of the Distribution Receivers List Count state value and the number of New Receivers is greater than the value of the Max Distribution Receivers List Size state, then the server shall respond with a Firmware Distribution Receivers Status message with the Status field set to Insufficient Resources.

Busy with distribution. When a Firmware Distribution Server receives a Firmware Distribution Receivers Add message, and the Distribution Phase state is not Idle, the server shall respond with a Firmware Distribution Receivers Status message with the Status field set to Busy With Distribution.

6.2.3.2 Receiving a Firmware Distribution Receivers Delete All message

Deleting all nodes. When a Firmware Distribution Server receives a Firmware Distribution Receivers Delete All message, and the Distribution Phase state of the server is Idle, the server shall clear the Distribution Receivers List state and shall respond with a Firmware Distribution Receivers Status message with the Status field set to Success.

When a Firmware Distribution Server receives a Firmware Distribution Receivers Delete All message, and the Distribution Phase state of the server is Completed or Failed, the server shall clear the Distribution Receivers List state, shall set the Distribution Phase state to Idle, and shall respond with a Firmware Distribution Receivers Status message with the Status field set to Success.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Receivers Delete All message, and the Distribution Phase state of the server is not Idle and not Completed and not Failed, the server shall respond with a Firmware Distribution Receivers Status message with the Status field set to Busy With Distribution.

6.2.3.3 Sending a Firmware Distribution Receivers Status message

A Firmware Distribution Server sends a Firmware Distribution Receivers Status message in response to a Firmware Distribution Receivers Add message or a Firmware Distribution Receivers Delete All message.

When a Firmware Distribution Server sends a Firmware Distribution Receivers Status message, the Firmware Distribution Server shall set the Receivers List Count field to the value of the Distribution Receivers List Count state.

6.2.3.4 Receiving a Firmware Distribution Receivers Get message

When a Firmware Distribution Server receives a Firmware Distribution Receivers Get message, the Firmware Distribution Server shall respond with a Firmware Distribution Receivers List message with

the First Index field set to the value of the First Index field in the Firmware Distribution Receivers Get message.

6.2.3.5 Sending a Firmware Distribution Receivers List message

A Firmware Distribution Server sends a Firmware Distribution Receivers List message in response to a Firmware Distribution Receivers Get message.

When a Firmware Distribution Server sends a Firmware Distribution Receivers List message, the Firmware Distribution Server shall set the Receivers List Count field to the value of the Distribution Receivers List Count state.

When the First Index field from the Firmware Distribution Receivers Get message indicates an entry that does not exist in the Distribution Receivers List state, the Receivers List field shall be omitted.

When the First Index field from the Firmware Distribution Receivers Get message indicates an entry that exists in the Distribution Receivers List state, the Firmware Distribution Server shall select entries from the Distribution Receivers List state consecutively, starting with the entry indicated by the value of the First Index field. The number of selected entries shall be limited by the following:

- The number of entries shall not exceed the value of the Entries Limit field from the Firmware Distribution Receivers Get message.
- The number of entries shall not cause the message payload to exceed the maximum Access PDU size.

Then, the server shall set the fields in every Target Node Entry as follows:

- The Address field of the message shall be set to the Target Node Address field of the corresponding entry in the Distribution Receivers List state.
- The Transfer Status field shall be set to the value of the output from the Get BLOB Receivers procedure from the BLOB Transfer Client model.
- The Transfer Progress field shall be set to the value of the output from the Get BLOB Transfer Progress procedure from the BLOB Transfer Client model.
- The Retrieved Update Phase field and the Update Status field shall be set to the values of the outputs from the Get Update Receiver procedure from the Firmware Update Client model.
- The Update Firmware Image Index field of the message shall be set to the Update Firmware Image Index field of the corresponding entry in the Distribution Receivers List state.

6.2.3.6 Receiving a Firmware Distribution Capabilities Get message

When a Firmware Distribution Server receives a Firmware Distribution Capabilities Get message, the Firmware Distribution Server shall respond with a Firmware Distribution Capabilities Status message.

6.2.3.7 Sending Firmware Distribution Capabilities Status message

A Firmware Distribution Server sends a Firmware Distribution Capabilities Status message in response to a Firmware Distribution Capabilities Get message.

When a Firmware Distribution Server sends a Firmware Distribution Capabilities Status message, the Firmware Distribution Server shall set the Max Distribution Receivers List Size field, the Max Firmware List Size field, the Max Firmware Image Size field, the Max Upload Space field, the Remaining Upload State field, the Out-of-Band Retrieval Supported field, and the Supported URI Scheme Names field to the values of the corresponding states.

6.2.3.8 Receiving a Firmware Distribution Get message

When a Firmware Distribution Server receives a Firmware Distribution Get message, the Firmware Distribution Server shall respond with a Firmware Distribution Status message with the Status field set to Success.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Get message, and the message is not processed successfully because an implementation-specific error (e.g., a memory allocation error, or flash memory error) has occurred, the Firmware Distribution Server shall respond with the Firmware Distribution Status message with the Status field set to Internal Error.

6.2.3.9 Receiving a Firmware Distribution Start message

Start firmware distribution. When a Firmware Distribution Server receives a Firmware Distribution Start message, and the Distribution Phase state is Idle, Completed, or Failed, and the message is processed successfully (i.e., it does not result in any error condition specified in Table 6.10), then the server shall set the Distribution Parameters state to the corresponding field values defined in Table 6.11, shall start the Distribute Firmware procedure, and shall respond with a Firmware Distribution Status message with the Status field set to Success.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Start message, and the Distribution Phase state is Idle, Completed, or Failed, and the message is not processed successfully (i.e., it results in an error condition specified in Table 6.10), the server shall respond with a Firmware Distribution Status message with the Status field set to the corresponding status code defined in Table 6.10.

Error Condition	Status Code Name (see Table 4.22)
The Distribution Receivers List state is empty	Receivers List Empty
The Application Key identified by the AppKey Index field is unknown to the server	Invalid AppKey Index
The firmware image identified by the Distribution Firmware Image Index field is not known to the server	Firmware Not Found

Table 6.10: Error conditions for the Firmware Distribution Start message

When a Firmware Distribution Server receives a Firmware Distribution Start message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Distribution Server shall respond with the Firmware Distribution Status message with the Status field set to Internal Error.

Busy. When a Firmware Distribution Server receives a Firmware Distribution Start message, and the Distribution Phase state is Transfer Active, Transfer Suspended, Transfer Success, or Applying Update, and one or more values of the fields of the received message do not match the corresponding states defined in Table 6.11, then the server shall respond with a Firmware Distribution Status message with the Status field set to Busy With Distribution.

When a Firmware Distribution Server receives a Firmware Distribution Start message, and the Distribution Phase state is Canceling Update, the server shall respond with a Firmware Distribution Status message with the Status field set to Busy With Distribution.

Resume distribution. When a Firmware Distribution Server receives a Firmware Distribution Start message, and the Distribution Phase state is Transfer Suspended, and all values of the fields of the received message match the corresponding states defined in Table 6.11, then the server shall resume the Distribute Firmware procedure by executing the Resume step, and shall respond with a Firmware Distribution Status message with the Status field set to Success.

Idempotency. When a Firmware Distribution Server receives a Firmware Distribution Start message, and the Distribution Phase state is Transfer Active, Transfer Success, Applying Update, or

Completed, and all values of the fields of the received message match the corresponding states defined in Table 6.11, then the server shall respond with a Firmware Distribution Status message with the Status field set to Success.

Field	State
Distribution Multicast Address	Distribution Multicast Address
Distribution AppKey Index	Distribution AppKey Index
Distribution TTL	Distribution TTL
Distribution Timeout Base	Distribution Timeout Base
Distribution Transfer Mode	Distribution Transfer Mode
Update Policy	Update Policy
Distribution Firmware Image Index	Distribution Firmware Image Index

Table 6.11: Mapping of the fields of the Firmware Distribution Start message to the Distribution Parameters state

6.2.3.9.1 Receiving a Firmware Distribution Suspend message

Suspend distribution. When a Firmware Distribution Server receives a Firmware Distribution Suspend message, and the Distribution Phase state of the server is Transfer Active, the server shall try to suspend the Distribute Firmware procedure. When the suspension of the Distribute Firmware procedure succeeds, the server shall set the Distribution Phase state to Transfer Suspended and shall respond with a Firmware Distribution Status message with the Status field set to Success. When the suspension of the Distribute Firmware procedure fails, the server shall respond with a Firmware Distribution Status message with the Status field set to Suspend Failed.

When a Firmware Distribution Server receives a Firmware Distribution Suspend message, and the Distribution Phase state of the server is not Transfer Active and not Transfer Suspended, the server shall respond with a Firmware Distribution Status message with the Status field set to Wrong Phase.

Idempotency. When a Firmware Distribution Server receives a Firmware Distribution Suspend message, and the Distribution Phase state of the server is Transfer Suspended, the server shall respond with a Firmware Distribution Status message with the Status field set to Success.

6.2.3.10 Receiving a Firmware Distribution Cancel message

Cancel distribution. When a Firmware Distribution Server receives a Firmware Distribution Cancel message, and the Distribution Phase state of the server is Transfer Active, Transfer Suspended, or Transfer Success, the server shall set the Distribution Phase state to Canceling Update, shall cancel the Distribute Firmware procedure, and shall respond with a Firmware Distribution Status message with the Status field set to Success. Then the server shall start the Cancel Firmware Update procedure from the Firmware Update Client model and shall wait for the procedure to complete. Finally, the Firmware Distribution Server shall set the Distribution Phase state to Idle.

When a Firmware Distribution Server receives a Firmware Distribution Cancel message, and the Distribution Phase state is Applying Update, the server shall set the Distribution Phase state to Canceling Update, shall cancel the Distribute Firmware procedure, and shall respond with a Firmware Distribution Status message with the Status field set to Success. Then the Firmware Distribution Server shall set the Distribution Phase state to Idle and shall respond with a Firmware Distribution Status message with the Status field set to Success.

When a Firmware Distribution Server receives a Firmware Distribution Cancel message, and the Distribution Phase state of the server is either Completed or Failed, the server shall set the

Distribution Phase state to Idle and shall respond with a Firmware Distribution Status message with the Status field set to Success.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Cancel message, and the message is not processed successfully because an implementation-specific error (e.g., a memory allocation error, or flash memory error) has occurred, the Firmware Distribution Server shall respond with the Firmware Distribution Status message with the Status field set to Internal Error.

Idempotency. When a Firmware Distribution Server receives a Firmware Distribution Cancel message, and the Distribution Phase state of the server is either Idle or Canceling Update, the server shall respond with a Firmware Distribution Status message with the Status field set to Success.

6.2.3.11 Receiving a Firmware Distribution Apply message

Apply firmware update. When a Firmware Distribution Server receives a Firmware Distribution Apply message, and the Distribution Phase state of the server is Transfer Success, the server shall start the Apply Firmware On Target Nodes procedure from the Firmware Update Client model, shall set the Distribution Phase state to Applying Update, and shall respond with a Firmware Distribution Status message with the Status field set to Success.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Apply message, and the Distribution Phase state of the server is Idle, Canceling Update, Transfer Active, Transfer Suspended, or Failed, the server shall respond with a Firmware Distribution Status message with the Status field set to Wrong Phase.

Idempotency. When a Firmware Distribution Server receives a Firmware Distribution Apply message, and the Distribution Phase state of the server is either Applying Update or Completed, the server shall respond with a Firmware Distribution Status message with the Status field set to Success.

6.2.3.12 Sending a Firmware Distribution Status message

A Firmware Distribution Server sends a Firmware Distribution Status message in response to a Firmware Distribution Get message, a Firmware Distribution Start message, a Firmware Distribution Suspend message, a Firmware Distribution Cancel message, or a Firmware Distribution Apply message.

When a Firmware Distribution Server sends a Firmware Distribution Status message, and the Distribution Phase state of the server is not Idle, the server shall set the Distribution Phase field, the Distribution Multicast Address field, the Distribution AppKey Index field, the Distribution TTL field, the Distribution Timeout Base field, the Distribution Transfer Mode field, the Update Policy field, and the Distribution Firmware Image Index field to the values of the corresponding states.

When a Firmware Distribution Server sends a Firmware Distribution Status message, and the Distribution Phase state of the server is Idle, the server shall set the Distribution Phase field to the value of the Distribution Phase state and shall omit the following fields from the message: the Distribution Multicast Address field, the Distribution AppKey Index field, the Distribution TTL field, the Distribution Timeout Base field, the Distribution Transfer Mode field, the Update Policy field, and the Distribution Firmware Image Index field.

6.2.3.13 Receiving a Firmware Distribution Upload Get message

When a Firmware Distribution Server receives a Firmware Distribution Upload Get message, the Firmware Distribution Server shall respond with a Firmware Distribution Upload Status message with the Status field set to Success.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Upload Get message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Distribution Server shall respond with the Firmware Distribution Upload Status message with the Status field set to Internal Error.

6.2.3.14 Receiving a Firmware Distribution Upload Start message

Start upload. When a Firmware Distribution Server receives a Firmware Distribution Upload Start message, and the Upload Phase state of the server is Idle, Transfer Success, or Transfer Error, and the Upload Firmware ID field of the message does not match the Firmware ID of any entry in the Firmware Images List state, and the message is processed successfully (i.e., it does not result in any error condition listed in Table 6.12), then the server shall set the Upload Phase state to Transfer Active, shall set the Upload Type state to In-band, and shall set the Upload TTL state, the Upload Timeout Base state, the Upload BLOB ID state, the Upload Firmware Size state, the Upload Firmware Metadata state, and the Upload Firmware ID state to the values of the corresponding fields in the message. Then, the server shall respond with a Firmware Distribution Upload Status message with the Status field set to Success and shall start the Store Firmware procedure.

Firmware image already exists. When a Firmware Distribution Server receives a Firmware Distribution Upload Start message, and the Upload Phase state of the server is Idle, Transfer Success, or Transfer Error, and the Upload Firmware ID field of the message matches the Firmware ID of any entry in the Firmware Images List state, the server shall set the Upload Phase state to Transfer Success, and shall respond with a Firmware Distribution Upload Status message with the Status field set to Success.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Upload Start message, and the Upload Phase state of the server is Idle, Transfer Success, or Transfer Error, and the message is not processed successfully (i.e., it results in any of the error conditions listed in Table 6.12), then the server shall respond with a Firmware Distribution Upload Status message with the Status field set to the corresponding status code defined in Table 6.12.

Error Condition	Status Code Name (see Table 4.22)
The number of entries in the Firmware List state is equal to the Max Firmware List Size state	Insufficient Resources
The Upload Firmware Size field is greater than the Max Firmware Image Size state	Insufficient Resources
The Upload Firmware Size field is greater than the Remaining Upload Space state	Insufficient Resources

Table 6.12: Error conditions for the Firmware Distribution Upload Start message

When a Firmware Distribution Server receives a Firmware Distribution Upload Start message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Distribution Server shall respond with the Firmware Distribution Upload Status message with the Status field set to Internal Error.

Busy. When a Firmware Distribution Server receives a Firmware Distribution Upload Start message, and the Upload Phase state of the server is Transfer Active, and the Upload Type state of the server is Out-of-band, then the server shall respond with a Firmware Distribution Upload Status message with the Status field set to Busy With Upload.

When a Firmware Distribution Server receives a Firmware Distribution Upload Start message, and the Upload Phase state of the server is Transfer Active, and the Upload Type state of the server is In-band, and the message is not processed successfully (i.e., it results in any of the error conditions listed in Table 6.13), then the server shall respond with a Firmware Distribution Upload Status message with the Status field set to Busy With Upload.

Idempotency. When a Firmware Distribution Server receives a Firmware Distribution Upload Start message, and the Upload Phase state of the server is Transfer Active, and the Upload Type state of the server is In-band, and the message is processed successfully (i.e., it does not result in any error condition listed in Table 6.13), the server shall respond with a Firmware Distribution Upload Status message with the Status field set to Success.

Error Condition
The Upload Timeout Base field does not match the Upload Timeout Base state
The Upload TTL field does not match the Upload TTL state
The Upload BLOB ID field does not match the Upload BLOB ID state
The Upload Firmware Size field does not match the Upload Firmware Size state
The Upload Firmware Metadata field does not match the Upload Firmware Metadata state
The value of the Upload Firmware Metadata Length field does not match the length of Upload Firmware Metadata state
The Upload Firmware ID field does not match the Upload Firmware ID state

Table 6.13: Conditions used in idempotency checks when a Firmware Distribution Upload Start message is received

6.2.3.15 Receiving a Firmware Distribution Upload OOB Start message

Upload start. When a Firmware Distribution Server receives a Firmware Distribution Upload OOB Start message, and the Upload Phase state of the server is Idle, Transfer Success, or Transfer Error, and the message is processed successfully (i.e., it does not result in any error condition listed in Table 6.14), then the server shall execute the Check Firmware OOB procedure with the Upload OOB URI field and Upload OOB Firmware ID field as inputs. If the new Firmware ID returned by the procedure does not match the Firmware ID of any entry in the Firmware Images List state, then the server shall set the Upload Phase state to Transfer Active, shall set the Upload Type to Out-of-band, shall set the Upload OOB URI and the Upload OOB Firmware ID states to the values of the corresponding fields, shall set the Upload Type state to Out-of-band, shall start the Store Firmware OOB procedure, and shall respond with a Firmware Distribution Upload Status message with the Status field set to Success.

Firmware image already exists. When a Firmware Distribution Server receives a Firmware Distribution Upload OOB Start message, and the Upload Phase state of the server is Idle, Transfer Success, or Transfer Error, and the message is processed successfully (i.e., it does not result in any error condition listed in Table 6.14), then the server shall execute the Check Firmware OOB procedure with the Upload OOB URI field and Upload OOB Firmware ID field as inputs. If the new Firmware ID returned by the procedure matches the Firmware ID of any entry in the Firmware Images List state, the server shall set the Upload Phase state to Transfer Success, shall set the Upload OOB URI and the Upload OOB Firmware ID states to the values of the corresponding fields, shall set the Upload Type state to Out-of-band, and shall respond with a Firmware Distribution Upload Status message with the Status field set to Success.

No new image available. When a Firmware Distribution Server receives a Firmware Distribution Upload OOB Start message, and the Upload Phase state of the server is Idle, Transfer Success, or Transfer Error, and the message is processed successfully (i.e., it does not result in any error condition listed in Table 6.14), then the server shall execute the Check Firmware OOB procedure with the Upload OOB URI field and Upload OOB Firmware ID field as inputs. If the Check Firmware OOB procedure completes successfully but does not yield a new Firmware ID (i.e., no new firmware image is available), then the server shall respond with a Firmware Distribution Upload Status message with the Status field set to New Firmware Not Available.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Upload OOB Start message, and the Upload Phase state is Idle, Transfer Success, or Transfer Error, and the message is not processed successfully (i.e., it results in any error condition listed in Table 6.14), then the server shall respond with a Firmware Distribution Upload Status message with the Status field set to the corresponding status code defined in Table 6.14.

Error Condition	Status Code Name (see Table 4.22)
The number of entries in the Firmware List state is equal to the Max Firmware List Size state	Insufficient Resources
The Upload OOB URI field is not formatted as defined in Section 4.3.4.8	URI Malformed
The Upload URI scheme name of the Upload OOB URI field does not match any entry in the Supported URI Scheme Names state	URI Not Supported
The Upload OOB URI could not be reached (e.g., an HTTPS connection could not be formed)	URI Unreachable

Table 6.14: Error conditions for the Firmware Distribution Upload OOB Start message

When a Firmware Distribution Server receives a Firmware Distribution Upload OOB Start message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Distribution Server shall respond with the Firmware Distribution Upload Status message with the Status field set to Internal Error.

Busy. When a Firmware Distribution Server receives a Firmware Distribution Upload OOB Start message, and the Upload Phase state of the server is Transfer Active, and the Upload Type state of the server is In-band, then the server shall respond with a Firmware Distribution Upload Status message with the Status field set to Busy With Upload.

When a Firmware Distribution Server receives a Firmware Distribution Upload OOB Start message, and the Upload Phase state of the server is Transfer Active, and the Upload Type state of the server is Out-of-band, and the message is not processed successfully (i.e., it results in any error condition listed in Table 6.15), then the server shall respond with a Firmware Distribution Upload Status message with the Status field set to Busy With Upload.

Idempotency. When a Firmware Distribution Server receives a Firmware Distribution Upload OOB Start message, and the Upload Phase state of the server is Transfer Active, and the Upload Type state of the server is Out-of-band, and the message is processed successfully (i.e., it does not result in any error condition listed in Table 6.15), then the server shall respond with a Firmware Distribution Upload Status message with the Status field set to Success.

Error Condition
The Upload OOB URI field does not match the Upload OOB URI state
The Upload OOB Firmware ID field does not match the Upload OOB Firmware ID state

Table 6.15: Conditions used in idempotency checks when a Firmware Distribution Upload OOB Start message is received

6.2.3.16 Receiving a Firmware Distribution Upload Cancel message

Cancel Distribution. When a Firmware Distribution Server receives a Firmware Distribution Upload Cancel message, and the Upload Phase state of the server is Idle, then the server shall respond with a Firmware Distribution Upload Status message with the Status field set to Success.

When a Firmware Distribution Server receives a Firmware Distribution Upload Cancel message, and the Upload Phase state of the server is either Transfer Error or Transfer Success, then the server shall set the Upload Phase state to Idle and shall respond with a Firmware Distribution Upload Status message with the Status field set to Success.

When a Firmware Distribution Server receives a Firmware Distribution Upload Cancel message, and the Upload Phase state of the server is Transfer Active, and the Upload Type state of the server is In-band, then the server shall cancel the Store Firmware procedure, shall set the Upload Phase state to Idle, and shall respond with a Firmware Distribution Upload Status message with the Status field set to Success.

When a Firmware Distribution Server receives a Firmware Distribution Upload Cancel message, and the Upload Phase state of the server is Transfer Active, and the Upload Type state of the server is Out-of-band, then the server shall cancel the Store Firmware OOB procedure, shall set the Upload Phase state to Idle, and shall respond with a Firmware Distribution Upload Status message with the Status field set to Success.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Upload Cancel message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Distribution Server shall respond with the Firmware Distribution Upload Status message with the Status field set to Internal Error.

6.2.3.17 Sending a Firmware Distribution Upload Status message

A Firmware Distribution Server sends a Firmware Distribution Upload Status message in response to a Firmware Distribution Upload Get message, a Firmware Distribution Upload Start message, a Firmware Distribution Upload OOB Start message, or a Firmware Distribution Upload Cancel message.

The Firmware Distribution Server formats the Firmware Distribution Upload Status message based on the value of the Upload Phase state.

- **Transfer Active.** If the Upload Phase state is Transfer Active, then the server shall set the Upload Phase field of the message to Transfer Active, and then the server shall do one of the following:
  - If the Upload Type state is In-band, then the server shall set the Upload Progress field to the output value of the Get BLOB Reception Progress procedure from the BLOB Transfer Server model, shall set the Upload Type field to In-band, and shall set the Upload Firmware ID field to the value of the Upload Firmware ID state.
  - If the Upload Type state is Out-of-band, then the server shall set the Upload Progress field to indicate the progress of the Store Firmware OOB procedure (in the range 0 to 99), shall set the Upload Type field to Out-of-band, and shall set the Upload OOB Firmware ID field to the value of the Upload OOB Firmware ID state.
- **Idle.** If the Upload Phase state is Idle, then the server shall set the Upload Phase field of the message to Idle and shall omit the Upload Progress field, Upload Type field, and Upload Firmware ID field.
- **Transfer Success.** If the Upload Phase state is Transfer Success, then the server shall set the Upload Phase field of the message to Transfer Success, shall set the Upload Progress field of the message to 100, and then the server shall do one of the following:
  - If the Upload Type state is In-band, then the server shall set the Upload Firmware ID field of the message to the value of the Upload Firmware ID state.
  - If the Upload Type state is Out-of-band, then the server shall set the Upload OOB Firmware ID field of the message to the value of the Upload OOB Firmware ID state.
- **Transfer Error.** If the Upload Phase state is Transfer Error, then the server shall set the Upload Phase field of the message to Transfer Error, shall set the Upload Progress field of the message to 0, and then the server shall do one of the following:
  - If the Upload Type state is In-band, then the server shall set the Upload Type field to In-band, and shall set the Upload Firmware ID field of the message to the value of the Upload Firmware ID state.

- If the Upload Type state is Out-of-band, then the server shall set the Upload Type field to Out-of-band, and shall set the Upload OOB Firmware ID field of the message to the value of the Upload OOB Firmware ID state.

Figure 6.2 shows the transitions between the phases of the Upload Phase state.

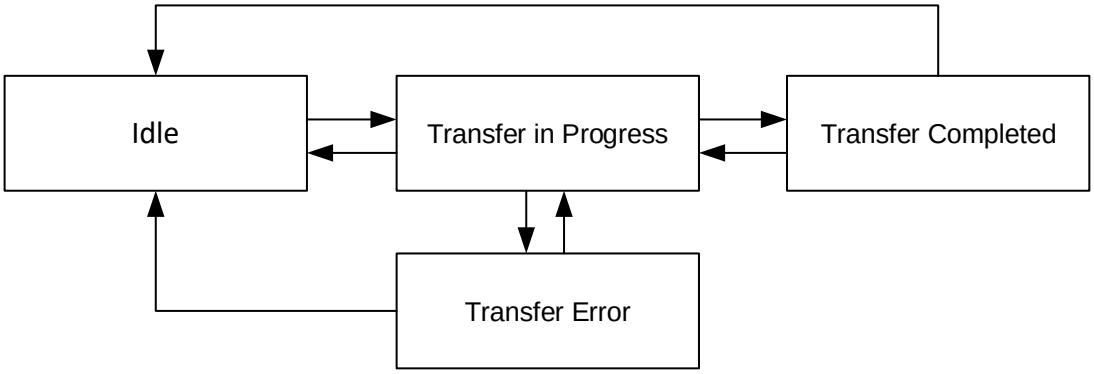

Figure 6.2: Upload Phase state transitions

6.2.3.18 Receiving a Firmware Distribution Firmware Get message

Getting firmware image. When a Firmware Distribution Server receives a Firmware Distribution Firmware Get message, and the firmware image indicated by the Firmware ID field is present in the Firmware List state (i.e., the Entry), then the server shall respond with a Firmware Distribution Firmware Status message with the Status field set to Success, the Firmware ID field of the message set to the value of the Firmware ID field of the Entry, and the Distribution Firmware Image Index field of the message set to the index of the Entry.

Firmware images not present. When a Firmware Distribution Server receives a Firmware Distribution Firmware Get message, and the firmware indicated by the Firmware ID field is not present in the Firmware List state, then the server shall respond with a Firmware Distribution Firmware Status message with the Status field set to Firmware Not Found, the Distribution Firmware Image Index field set to 0xFFFF, and the Firmware ID field set to the value of the Firmware ID field in the Firmware Distribution Firmware Get message that was received.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Firmware Get message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Distribution Server shall respond with the Firmware Distribution Firmware Status message with the Status field set to Internal Error, the Distribution Firmware Image Index field set to 0xFFFF, and the Firmware ID field omitted.

6.2.3.19 Receiving a Firmware Distribution Firmware Get By Index message

Getting firmware image. When a Firmware Distribution Server receives a Firmware Distribution Firmware Get By Index message, and the firmware image indicated by the Index field is present in the Firmware List state (i.e., the Entry), then the server shall respond with a Firmware Distribution Firmware Status message with the Status field set to Success, the Distribution Firmware Image Index field of the message set to the index of the Entry, and the Firmware ID field of the message set to the Firmware ID field of the Entry.

Index does not exist. When a Firmware Distribution Server receives a Firmware Distribution Firmware Get By Index message, and the firmware image identified by the Distribution Firmware Image Index field is not present in the Firmware List state, then the server shall respond with a Firmware Distribution Firmware Status message with the Status field set to Firmware Not Found, the Distribution Firmware Image Index field of the Firmware Distribution Firmware Status message set to

the value of the Distribution Firmware Image Index field in the Firmware Distribution Firmware Get By Index message that was received, and the Firmware ID field shall be omitted.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Firmware Get By Index message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Distribution Server shall respond with the Firmware Distribution Firmware Status message with the Status field set to Internal Error, the Distribution Firmware Image Index field set to 0xFFFF, and the Firmware ID field omitted.

6.2.3.20 Receiving a Firmware Distribution Firmware Delete message

Deleting firmware image. When a Firmware Distribution Server receives a Firmware Distribution Firmware Delete message, and the Distribution Phase state is Idle, and the firmware identified by the Firmware ID field is present in the Firmware List state, then the server removes the firmware image from internal storage and shall remove the entry in the Firmware List state indicated by the value of the Firmware ID field of the message. In the Firmware List state, all indexes of all entries that are greater than the removed entry shall be decreased by 1 to keep the list contiguous. The server shall respond with a Firmware Distribution Firmware Status message with the Status field set to Success, the Distribution Firmware Image Index field set to 0xFFFF, and the Firmware ID field of the Firmware Distribution Firmware Status message set to the value of the Firmware ID field in the Firmware Distribution Firmware Delete message that was received.

Busy. When a Firmware Distribution Server receives a Firmware Distribution Firmware Delete message, and the Distribution Phase state is not Idle, then the server shall respond with a Firmware Distribution Firmware Status message with the Status field set to Busy With Distribution. The Distribution Firmware Image Index field shall be set to 0xFFFF. The Firmware ID field shall be omitted.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Firmware Delete message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Distribution Server shall respond with the Firmware Distribution Firmware Status message with the Status field set to Internal Error, the Distribution Firmware Image Index field set to 0xFFFF, and the Firmware ID field omitted.

Idempotency. When a Firmware Distribution Server receives a Firmware Distribution Firmware Delete message, and the Distribution Phase state is Idle, and the firmware identified by the Firmware ID field is not present in the Firmware List state, then the server shall respond with a Firmware Distribution Firmware Status message with the Status field set to Success, the Distribution Firmware Image Index field set to 0xFFFF, and the Firmware ID field of the Firmware Distribution Firmware Status message set to the value of the Firmware ID field in the Firmware Distribution Firmware Delete message that was received.

6.2.3.21 Receiving a Firmware Distribution Firmware Delete All message

Deleting all firmware images. When a Firmware Distribution Server receives a Firmware Distribution Firmware Delete All message, and the Distribution Phase state is Idle, then the server shall remove all firmware images from internal storage, shall remove all entries from the Firmware List state, shall set the Remaining Upload Space state to the value of the Max Upload Space state, and shall respond with a Firmware Distribution Firmware Status message with the Status field set to Success, the Distribution Firmware Image Index field set to 0xFFFF and the Firmware ID field shall be omitted.

Busy. When a Firmware Distribution Server receives a Firmware Distribution Firmware Delete All message, and the Distribution Phase state is not Idle, then the server shall respond with a Firmware Distribution Firmware Status message with the Status field set to Busy With Distribution, the Distribution Firmware Image Index field set to 0xFFFF, and the Firmware ID field shall be omitted.

Error handling. When a Firmware Distribution Server receives a Firmware Distribution Firmware Delete All message, and the message is not processed successfully because an implementation-specific error has occurred (e.g., a memory allocation error or flash memory error), then the Firmware Distribution Server shall respond with the Firmware Distribution Firmware Status message with the Status field set to Internal Error, the Distribution Firmware Image Index field set to 0xFFFF, and the Firmware ID field omitted.

6.2.3.22 Sending a Firmware Distribution Firmware Status message

A Firmware Distribution Server sends a Firmware Distribution Firmware Status message in response to a Firmware Distribution Firmware Get message, a Firmware Distribution Firmware Get By Index message, a Firmware Distribution Firmware Delete message, or a Firmware Distribution Firmware Delete All message.

When a Firmware Distribution Server sends a Firmware Distribution Firmware Status message in response to a Firmware Distribution Firmware Get message, a Firmware Distribution Firmware Get By Index message, a Firmware Distribution Firmware Delete message, or a Firmware Distribution Firmware Delete All message, the server shall set the Entry Count field in the message to the number of the entries in the Firmware List state.

7 Firmware Update client models

7.1 Firmware Update Client model

7.1.1 Description

The Firmware Update Client model is used by the Distributor and Initiator.

The Initiator uses this model to retrieve the information about the firmware subsystems installed on the Target node, and to get the location of the new firmware images. The Initiator receives the list of Target nodes from the higher-layer application.

The Distributor uses this model to start a firmware image transfer to the Target nodes. The Distributor receives the list of Target nodes from the Initiator. The Distributor uses the procedures on the Firmware Update Client model with the inputs to these procedures being chosen by the Firmware Distribution Server.

The Firmware Update Client is a main model that extends the BLOB Transfer Client model (see Section 6 in [10]).

The Firmware Update Client model requires one element: the Firmware Update Client Main element. The Firmware Update Client Main element contains the Firmware Update Client main model and all the models that the main model extends.

The access layer security on the Firmware Update Client model uses application keys.

Table 7.1 illustrates the structure of the element, procedures, and messages used by the Firmware Update Client model.

Element	Model Name	Procedure	Message	Rx	Tx
Firmware Update Client Main	Firmware Update Client	Retrieve Firmware Information	Firmware Update Information Get	–	C.1
			Firmware Update Information Status	C.1	–
		Firmware Compatibility Check	Firmware Update Firmware Metadata Check	–	C.1
			Firmware Update Firmware Metadata Status	C.1	–
		Start Firmware Update On Target Nodes, Apply Firmware On Target Nodes, Cancel Firmware Update, Refresh Target Nodes Status	Firmware Update Get	–	C.2
			Firmware Update Start	–	C.2
			Firmware Update Cancel	–	C.2
			Firmware Update Apply	–	C.2
			Firmware Update Status	C.2	–

Table 7.1: Firmware Update Client elements and procedures

C.1: Mandatory if Initiator is supported or Standalone Updater is supported; otherwise, Optional.

C.2: Mandatory if Distributor is supported or Standalone Updater is supported; otherwise, Optional.

7.1.2 Behavior

If a Firmware Update Client executes a procedure that sets the Update TTL state, then, when originating a message during that procedure, the client shall set the TTL field of the Network PDU to the value indicated by the Update TTL state. If the procedure does not set the Update TTL state, then the client shall use the Update TTL input in the Network PDU.

If a Firmware Update Client executes a procedure that sets the Update AppKey Index state, then, when originating a message during that procedure, the client shall use the key indicated by the Update AppKey Index state to encrypt the message. If the procedure does not set the Update AppKey Index state, then the client shall use the key indicated by the Update AppKey Index input to encrypt the message.

7.1.2.1 Multicast versus unicast addressing while executing the procedures

If a Firmware Update Client executes a procedure that sets the Update Multicast Address state, then, when originating a message during that procedure, the client shall use the multicast address indicated by the Update Multicast Address state. Otherwise, the client shall use the multicast address input as the multicast address:

- If the multicast address is the Unassigned address, then the message shall be sent to each unicast address in the Address List input.
- If the multicast address is not the Unassigned address, then the message should be sent to the address indicated by the multicast address input.

The client could use a mix of multicast and unicast addressing. For example, the client can set rules to determine when to use unicast addressing or multicast addressing based on the number of intended recipients.

7.1.2.2 Calculate Client Timeout procedure

The Calculate Client Timeout procedure is executed on a Firmware Update Client to calculate the Client Timeout value.

The client calculates a Client Timeout value, in milliseconds, from the following formula:

$$Client\ Timeout=(10{,}000*(Update\ Client\ Timeout\ Base+2))+(100*Update\ TTL)\ [milliseconds]$$

7.1.2.3 Retrieve Firmware Information procedure

The Retrieve Firmware Information procedure is executed on a Firmware Update Client to retrieve information about the firmware images installed on a set of Target nodes.

As inputs, the procedure takes an Update Multicast Address, a list of the Target nodes for a given firmware image (i.e., the Nodes List), an Update AppKey Index, an Update Timeout Base, and an Update TTL.

As output, the procedure gives the Result List. Each entry in the Result List contains a unicast address (Address) and a list of entries (Firmware Information List) defined in Table 4.4.

The Retrieve Firmware Information procedure can be canceled at any time by a higher-layer model or application.

The Retrieve Firmware Information procedure consists of the following steps:

1. The client shall set the Update Client Timeout Base state to the value of the Update Timeout Base input.
2. The client shall send a Firmware Update Information Get message (see Section 7.1.2.1) to the nodes in the Nodes List input, with the First Index field set to 0 and the Entries Limit field set to an implementation-specific value greater than 0.

   The client shall attempt to gather the Firmware Update Information Status messages for each entry in the Nodes List input.
3. The client shall initialize a timer with a timeout set to the output value of the Calculate Client Timeout procedure (see Section 7.1.2.2).

While the timer is running, the client may repeat the request to the Target nodes that have not responded. The number of request repetitions is implementation specific.

4. For each Firmware Update Information Status message received from the Source Address, the client stores the firmware subsystem information in the Result List as follows:
   - The Address field of the entry in the Result List is set to the value of the Source Address.
   - The entries from the Firmware Information List field of the Firmware Update Information Status message are added to the Firmware Information List field of the corresponding entry in the Result List.

If the Firmware Information List field does not contain information about all firmware subsystems on the Target node (i.e., the sum of the First Index field and the number of entries in the Firmware Information List field is less than the value of the Firmware Information List Count field), then the client shall send additional Firmware Update Information Get messages to retrieve the information for the remaining subsystems.

7.1.2.3.1 Sending a Firmware Update Information Get message

To retrieve the information about the firmware subsystems installed on a Target node, a Firmware Update Client shall send a Firmware Update Information Get message. The response to a Firmware Update Information Get message is a Firmware Update Information Status message.

7.1.2.3.2 Receiving a Firmware Update Information Status message

When a Firmware Update Client receives a Firmware Update Information Status message from a Firmware Update Server, the client shall store the information for the firmware subsystem.

7.1.2.4 Firmware Compatibility Check procedure

The Firmware Compatibility Check procedure is executed on a Firmware Update Client to check whether a set of Target nodes can accept a firmware update.

As inputs, the procedure takes an Update Multicast Address, a set of Target nodes and the index of the firmware subsystem on the node to update (the Receivers List), a Firmware Metadata, an Update AppKey Index, an Update Timeout Base, and an Update TTL.

The Firmware Compatibility Check procedure can be canceled at any time by a higher-layer model or application.

The Firmware Compatibility Check procedure consists of the following steps:

1. The client shall set the Update Client Timeout Base state to the value of the Update Timeout Base input.
2. The client shall send a Firmware Update Firmware Metadata Check message (see Section 7.1.2.1) with the following field values:
   - The Firmware Image Index field shall be set to the Update Firmware Image Index field in the Receiver Entry.
   - The Firmware Metadata field shall be set to the Firmware Metadata input.

   And then the client shall wait for the Firmware Update Firmware Metadata Status message. The client shall attempt to gather the Firmware Update Firmware Metadata Status messages for each entry in the Receivers List input (i.e., the Receiver Entry).
3. The client shall initialize a timer with a timeout set to the output value of the Calculate Client Timeout procedure (see Section 7.1.2.2).

   While the timer is running, the client may repeat the request to the Target nodes that have not responded. The number of request repetitions is implementation specific.

The output of the procedure is the list of Target nodes that responded to the request with the Status field equal to Success, and the list of nodes that responded with the Status field not equal to Success or did not respond.

7.1.2.4.1 Sending a Firmware Update Firmware Metadata Check message

To determine if a new firmware image is suitable for a Target node, a Firmware Update Client shall send a Firmware Update Firmware Metadata Check message. The response is a Firmware Update Firmware Metadata Status message (see Section 5.1.4).

7.1.2.4.2 Receiving a Firmware Update Firmware Metadata Status message

When a Firmware Update Client receives a Firmware Update Firmware Metadata Status message and the Status field is Success, the firmware subsystem on the Target node identified by the Update Firmware Image Index field accepted the metadata and can be updated with the new firmware image.

7.1.2.5 Start Firmware Update On Target Nodes procedure

The Start Firmware Update On Target Nodes procedure is executed on a Firmware Update Client to initiate or resume a firmware update to a set of Target nodes.

As inputs, the procedure takes an Update Multicast Address, a set of Target nodes and the index of the firmware subsystem on the node to update (i.e., the Receivers List), an Update AppKey Index, an Update TTL, an Update Client Timeout Base, an Update BLOB ID, and a Firmware Metadata.

The Start Firmware Update On Target Nodes procedure can be canceled at any time by a higher-layer model or application.

The Start Firmware Update On Target Nodes procedure consists of the following steps:

1. The client performs the following operations:
    a. The client removes all entries from the Update Receivers state.
    b. For each unicast address on the Receivers List input, the client adds an entry in the Update Receivers state and sets the following fields:
        - The Address field is set to the value of that unicast address.
        - The Firmware Image Index field is set to the value of that Firmware Image Index.
        - The Retrieved Update Phase field is set to Unknown.
        - The Update Status field is set to Success.
        - The Update Additional Information field is set to CD Unchanged (0x0).
    c. The client sets the Update Multicast Address state, the Update AppKey Index state, the Update Client Timeout Base state, and the Update TTL state to the values of the corresponding inputs.
    d. The client copies the Update Receivers state to the Active Update Receivers state.
2. The client shall send a Firmware Update Start message (see Section 7.1.2.1) with the Update TTL field, the Update Client Timeout Base field, the Update BLOB ID field, the Update Firmware Image Index field, and the Firmware Metadata field set to the values of the corresponding inputs, and then the client shall wait for the Firmware Update Status message. The client shall attempt to gather Firmware Update Status messages for each unicast address in the Receivers List input.
3. The client shall initialize a timer with a timeout set to the output value of the Calculate Client Timeout procedure (see Section 7.1.2.2).

    While the timer is running, the client may repeat the request to the Target nodes that have not responded. The number of request repetitions is implementation specific. When the timer expires,

the client shall update the Active Update Receivers state by removing the Target nodes that have not responded or that have responded with the Status field not equal to Success.

The Start Firmware Update On Target Nodes procedure completes when either of the following conditions is met:

- If the Active Update Receivers state is not empty, then the Start Firmware Update On Target Nodes procedure completes successfully.
- If the Active Update Receivers state is empty, then the Start Firmware Update On Target Nodes procedure fails.

7.1.2.5.1 Sending a Firmware Update Start message

To start a firmware image transfer, a Firmware Update Client shall send a Firmware Update Start message. The response is a Firmware Update Status message (see Section 5.1.9).

7.1.2.6 Apply Firmware On Target Nodes procedure

The Apply Firmware On Target Nodes procedure is executed on a Firmware Update Client to trigger the installation of a firmware image on a set of Target nodes.

As inputs, the procedure takes an Update Multicast Address, a list of the Target nodes for a given firmware image (i.e., the Nodes List), an Update AppKey Index, an Update TTL, and an Update Client Timeout Base.

The Apply Firmware On Target Nodes procedure can be canceled at any time by a higher-layer model or application.

The Apply Firmware On Target Nodes procedure consists of the following steps:

1. The client shall set the Update Client Timeout Base state to the value of the Update Client Timeout Base input.
2. The client shall send a Firmware Update Apply message (see Section 7.1.2.1) and shall wait for a Firmware Update Status message.
3. The client shall initialize a timer with a timeout set to the output value of the Calculate Client Timeout procedure (see Section 7.1.2.2).

   While the timer is running, the client may repeat the request to the Target nodes that have not responded with the Update Phase set to Idle. The number of request repetitions is implementation specific. The Apply Firmware On Target Nodes procedure completes when one of the following conditions is met:

   - The timer expires.
   - All the nodes in the Nodes List input have responded with a Firmware Update Status message at least once and the client determines that sending subsequent Firmware Update Apply messages will not cause the node to change the field values of the Firmware Update Status message.

     The client may send several Firmware Update Apply messages to get confirmation about the state of the server, because this message behavior is idempotent.

     If the Firmware Update Additional Information state for a node was set to 0x03 and a Firmware Update Status message is not received, then the client should stop sending additional Firmware Update Apply messages to this node.

The client shall process the responses and update the Active Update Receivers state by removing the Target nodes as follows:

- If the Firmware Update Additional Information state for a node was set to 0x03 and the node responded with the Update Phase field set to Idle.
- If the Firmware Update Additional Information state was set to values other than 0x03, and the node did not respond or responded with the Firmware Update Status message with the Update Phase field value not set to Idle and not set to Applying Update.

When the Firmware Update Additional Information state for a node is set to 0x03 and the response to the Firmware Update Apply message is not received, the node shall not be removed from the Active Update Receivers state.

If the Active Update Receivers state is not empty, then the Apply Firmware On Target Nodes procedure completes successfully.

If the Active Update Receivers state is empty, then the Apply Firmware On Target Nodes procedure fails.

7.1.2.6.1 Sending a Firmware Update Apply message

To apply a firmware image on a Target node, a Firmware Update Client shall send a Firmware Update Apply message. The response is a Firmware Update Status message (see Section 5.1.9).

7.1.2.7 Cancel Firmware Update procedure

The Cancel Firmware Update procedure is executed on a Firmware Update Client to cancel a firmware update to a set of Target nodes.

The Cancel Firmware Update procedure consists of the following steps:

1. The client cancels any of the following procedures if they are being executed:
   - The Retrieve Firmware Information procedure.
   - The Firmware Compatibility Check procedure.
   - The Start Firmware Update On Target Nodes procedure.
   - The Apply Firmware On Target Nodes procedure.
   - The Refresh Target Nodes Status procedure.
2. The client shall send the Firmware Update Cancel message (see Section 7.1.2.1) to all Target nodes on the Update Receivers state for which the Retrieved Update Phase field in the Update Receivers state is Transfer Error, Transfer Active, Verifying Update, Verification Succeeded, or Verification Failed. The client shall attempt to gather Firmware Update Status messages for each unicast address in the Update Receivers state.
3. The client shall initialize a timer with a timeout set to the output value of the Calculate Client Timeout procedure (see Section 7.1.2.2).

   While the timer is running, the client may repeat the request to the Target nodes that have not responded. The number of request repetitions is implementation specific. When the timer expires, the client shall update the Active Update Receivers state by removing the Target nodes that have not responded or that have responded with the Status field not equal to Success. The client shall set its Update Phase field in the Update Receivers state to Transfer Canceled.
4. The client shall remove all Target nodes from the Active Update Receivers state.

7.1.2.7.1 Sending a Firmware Update Cancel message

To cancel a firmware update, a Firmware Update Client shall send a Firmware Update Cancel message. The response is a Firmware Update Status message (see Section 5.1.9).

7.1.2.8 Refresh Target Nodes Status procedure

The Refresh Target Nodes Status procedure is executed on a Firmware Update Client to retrieve the status of a firmware update for a set of Target nodes.

As inputs, the procedure takes an Update Multicast Address, a list of the Target nodes for a given firmware image (i.e., the Nodes List), an Update AppKey Index, an Update Client Timeout Base, and an Update TTL.

The Refresh Target Nodes Status procedure consists of the following steps:

1. The client shall remove the Target nodes from the Active Update Receivers state that are not on the Nodes List input.
2. The client shall set the Update Client Timeout Base state to the value of the Update Client Timeout Base input.
3. The client shall send the Firmware Update Get message (see Section 7.1.2.1) and shall wait for a Firmware Update Status message. The client shall attempt to gather Firmware Update Status messages for each entry in the Nodes List input.
4. The client shall initialize a timer with a timeout set to the output value of the Calculate Client Timeout procedure (see Section 7.1.2.2).

   While the timer is running, the client may repeat the request to the Target nodes that have not responded. The number of request repetitions is implementation specific. When the timer expires, the client shall update the Active Update Receivers state by removing the Target nodes that have not responded or that have responded with the Status field not equal to Success.

The Refresh Target Nodes Status procedure completes when either of the following conditions is met:

- If the Active Update Receivers state is not empty, then the Refresh Target Nodes Status procedure completes successfully.
- If the Active Update Receivers state is empty, then the Refresh Target Nodes Status procedure fails.

7.1.2.8.1 Sending a Firmware Update Get message

To get the status and phase of a Firmware Update Server, a Firmware Update Client shall send a Firmware Update Get message. The response is a Firmware Update Status message (see Section 5.1.9).

7.1.2.9 Confirm Update On Target Nodes procedure

The Confirm Update On Target Nodes procedure is executed on a Firmware Update Client to confirm that a firmware update to a set of Target nodes has completed successfully.

As inputs, the Confirm Update On Target Nodes procedure takes a Firmware ID and a Retrieved Information List.

From each entry in the Active Update Receivers state the procedure takes the Address field (i.e., the Checked Address), the Firmware Image Index field (i.e., the Checked Image Index), the Update Additional Information field (i.e., the Checked Additional Information), and the Retrieved Update Phase field (i.e., the Checked Update Phase).

If Checked Additional Information is equal to Device Unprovisioned (see Section 4.1.3), then the conditions in Table 7.2 shall be checked; otherwise, the conditions in Table 7.3 shall be checked. If all

of the conditions are met, then the procedure shall set Checked Update Phase to Apply Success; otherwise, the procedure shall set Checked Update Phase to Apply Failed.

Condition
There is no entry in the Retrieved Information List input with an Address field matching the value of Checked Address (i.e., the node did not respond to the Firmware Update Information Get message)

Table 7.2: Successful firmware update conditions for a device that is unprovisioned after the update

Condition
There is an entry in the Retrieved Information List input whose Address field matches the value of the Checked Address (i.e., the Firmware Information List field of this entry is the Images List)
The value of the Firmware ID input matches the value of the Current Firmware ID field of the entry in the Images List that is indicated by the Checked Image Index

Table 7.3: Successful firmware update conditions for a device that remains provisioned after an update

Then, if at least one receiver in the Active Update Receivers state has a Retrieved Update Phase field value equal to Apply Success, the Confirm Update On Target Nodes procedure completes successfully. Otherwise, the Confirm Update On Target Nodes procedure fails.

7.1.2.10 Get Update Receiver procedure

The Get Update Receiver procedure is executed on a Firmware Update Client to retrieve the status of the transfer to a selected Firmware Update Server.

As input, the procedure takes the Unicast Address of a server.

As output, the procedure gives an entry from the Update Receivers state identified by the Unicast Address input.

7.1.2.11 Get Active Update Receivers procedure

The Get Active Update Receivers procedure is executed on a Firmware Update Client to retrieve the list of Target nodes that are actively participating in the firmware update.

As output, the procedure gives the list of the Target nodes on the Active Update Receivers state.

7.1.2.12 Receiving a Firmware Update Status message

When a Firmware Update Client receives a Firmware Update Status message, it can determine the Update Phase state of a Firmware Update Server.

When a client receives a Firmware Update Status message from a server, the message shall be processed in one of the following ways:

- If the Status field of the message is Success, then the client shall set the server's entry in the Update Receivers state as follows:
  - The client shall set the Status field in the entry to the value of the Status field in the message.
  - The client shall set the Retrieved Update Phase field in the entry to the value of the Update Phase field in the message.
  - If the Additional Information field is present in the message, then the client shall set the Update Additional Information field in the entry to the value of the Additional Information field in the message.

- If the Status field of the message is not Success, then the client shall remove the receiver from the Active Update Receivers state and set the server's entry in the Update Receivers state as follows:
  - The client shall set the Status field in the entry to the value of the Status field in the message.
  - The client shall set the Retrieved Update Phase field in the entry to the value of the Update Phase field in the message.

7.2 Firmware Distribution Client model

7.2.1 Description

The Firmware Distribution Client model is used by the Initiator to send the firmware image and the firmware distribution parameters to the Distributor, and to start the firmware image transfer. The inputs to the procedures on the Firmware Distribution Client model are chosen by the higher-layer application.

The Firmware Distribution Client model is a main model that extends the BLOB Transfer Client model (see Section 6 in [10]) and corresponds with the Firmware Update Client model (see Section 7.1).

The Firmware Distribution Client model requires one element: the Firmware Distribution Client Main element. The Firmware Distribution Client Main element contains the Firmware Distribution Client main model, the BLOB Transfer Client model, and the Firmware Update Client model.

The access layer security on the Firmware Distribution Client model uses application keys.

Table 7.4 illustrates the structure of the element, procedures, and messages used by the Firmware Distribution Client model.

Element	Model Name	Procedure	Message	Rx	Tx
Firmware Distribution Client Main	Firmware Distribution Client	Populate Distribution Receivers List, Get Distribution Progress	Firmware Distribution Receivers Add	–	M
			Firmware Distribution Receivers Delete All	–	M
			Firmware Distribution Receivers Status	M	–
			Firmware Distribution Receivers Get	–	M
			Firmware Distribution Receivers List	M	–
		Distributor Capabilities	Firmware Distribution Capabilities Get	–	M
			Firmware Distribution Capabilities Status	M	–
		Distribution Parameters	Firmware Distribution Get	–	M
			Firmware Distribution Start	–	M
			Firmware Distribution Suspend	–	M
			Firmware Distribution Cancel	–	M
			Firmware Distribution Status	M	–
		Upload Firmware, Upload Firmware OOB	Firmware Distribution Upload Get	–	M
			Firmware Distribution Upload Start	–	M
			Firmware Distribution Upload OOB Start	–	M
			Firmware Distribution Upload Cancel	–	M
			Firmware Distribution Upload Status	M	–
		Initiate Distribution, Firmware List	Firmware Distribution Firmware Get	–	M
			Firmware Distribution Firmware Get By Index	–	M
			Firmware Distribution Firmware Delete	–	M

Element	Model Name	Procedure	Message	Rx	Tx
			Firmware Distribution Firmware Delete All	–	M
			Firmware Distribution Firmware Status	M	–

Table 7.4: Firmware Distribution Client elements, procedures, and messages

7.2.2 Behavior

7.2.2.1 Check For Current Firmware procedure

The Check For Current Firmware procedure is executed on a Firmware Distribution Client to retrieve the information about the firmware subsystems installed on a set of Target nodes.

As inputs, the procedure takes a Multicast Address, a list of Target nodes (i.e., the Nodes List), an AppKey Index, an Update Client Timeout Base, and a Procedure TTL.

The Firmware Distribution Client shall start the Retrieve Firmware Information procedure from the Firmware Update Client with the following inputs: Multicast Address input, Nodes List input, AppKey Index input, Update Client Timeout Base input, and a Procedure TTL input. The Firmware Distribution Client gets the Result List output from the Retrieve Firmware Information procedure.

The Firmware Distribution Client stores the Result List output in the Firmware Images List field of the Update Information From Target Nodes state.

7.2.2.2 Check For Updated Firmware procedure

The Check For Updated Firmware procedure is executed on a Firmware Distribution Client to check for the availability of a new firmware image.

As input, the procedure takes an Update URI and the Current Firmware ID.

As output, the procedure gives the number of firmware updates needed to install the latest firmware image available for the Target node (i.e., the Update Count).

The Check For Updated Firmware procedure consists of the following steps:

1. The Update URI is used to check for availability of a new firmware image. The check is performed based on the URI scheme that is used in the Update URI input:
    - If the Update URI input uses the https scheme, then the Firmware Distribution Client shall execute the Firmware Check Over HTTPS procedure described in Section 3.2 by using the Update URI and Current Firmware ID as inputs.
    - If the Update URI input is not using the https scheme, then the check is implementation specific, and the retrieved new firmware description file should use the file format defined in Section 3.1.
2. The client shall set the Current Firmware ID field in the Update Information From Target Nodes state to the value of the Current Firmware ID input and shall set the other fields in this entry to default values.
3. The client shall set the entry indicated by the Current Firmware ID input in the Update Information From Target Nodes state as follows:
    - The New Firmware ID field shall be set to the value extracted from the manifest/firmware/firmware_id member of the new firmware description file.

If the manifest/firmware/dfu_chain_size member is present in the new firmware description file, then the Update Count shall be set to the value extracted from the manifest/firmware/dfu_chain_size member. If this member is not present, then the Update Count shall be set to a value of 1.

The procedure completes if any of the following conditions are met:

- If the Firmware Check Over HTTPS procedure returns code 200 OK, then the Check For Updated Firmware procedure completes successfully.
- If the Firmware Check Over HTTPS procedure returns code 404 Not Found, then the Check For Updated Firmware procedure completes successfully, indicating that no new firmware image is available.
- If the Firmware Check Over HTTPS procedure fails, then the procedure fails.
- If the vendor-specific check fails, then the procedure fails.

7.2.2.3 Retrieve Updated Firmware procedure

The Retrieve Updated Firmware procedure may be executed on a Firmware Distribution Client to retrieve a new firmware image.

As input, the procedure takes an Update URI and the Current Firmware ID.

The Retrieve Updated Firmware procedure consists of the following steps:

1. The new firmware image is retrieved based on the URI scheme that is used in the Update URI input:
    - If the Update URI input uses the https scheme, then the Firmware Distribution Client shall execute the Firmware Retrieval Over HTTPS procedure described in Section 3.3 by using the Update URI and Current Firmware ID as input.
    - If the Update URI input does not use the https scheme, then the download process is implementation specific, and the retrieved firmware image should use the firmware archive file format defined in Section 3.1.
2. The client shall set the Current Firmware ID field in the Update Information From Target Nodes state to the value of the Current Firmware ID input and shall set the other fields in this entry to default values.
3. The client shall set the entry indicated by the Current Firmware ID input in the Update Information From Target Nodes as follows:
    - The New Firmware ID field shall be set to the value extracted from the manifest/firmware/firmware_id member in the Manifest file of the Firmware Archive file.
    - The Firmware Image field shall be set to the binary data file identified by the value extracted from the manifest/firmware/firmware_image_file member in the Manifest file of the Firmware Archive file.
    - If the manifest/firmware/metadata_file member is present in the Manifest file of the Firmware Archive file, then the Firmware Metadata field shall be set to the binary data file identified by the value extracted from the manifest/firmware/metadata_file member in the Manifest file of the Firmware Archive file.

The Retrieve Updated Firmware procedure completes if any of the following conditions are met:

- If the response code from the Firmware Retrieval Over HTTPS procedure is 404 Not Found, then the Retrieve Updated Firmware procedure completes successfully, indicating that there is no firmware update available.
- If the response code from the Firmware Retrieval Over HTTPS procedure is 200 OK, and the firmware archive file follows the format described in Section 3.1, then the Retrieve Updated Firmware procedure completes successfully.

- If the response code from the Firmware Retrieval Over HTTPS procedure is not 200 OK or is not 404 Not Found, then the Retrieve Updated Firmware procedure fails.
- If the implementation-specific firmware download procedure fails, then the Retrieve Updated Firmware procedure fails.
- If the retrieved firmware image does not follow the firmware archive file format described in Section 3.1, then the Retrieve Updated Firmware procedure fails.

7.2.2.4 Populate Distribution Receivers List procedure

The Populate Distribution Receivers List procedure is executed on a Firmware Distribution Client to add Target nodes to the Distribution Receivers List state of a Firmware Distribution Server.

As input, the procedure takes a set of Target nodes and the index of the firmware subsystem on the node to update (i.e., the Receivers List).

The Populate Distribution Receivers List procedure consists of the following steps:

1. The Firmware Distribution Client may start the Distributor Capabilities procedure to check whether the Firmware Distribution Server can store all entries from the Receivers List input.
2. The client may check the content of the Distribution Receivers List state on a Firmware Distribution Server by sending a Firmware Distribution Receivers Get message.
3. The client may delete the content of the Distribution Receivers List state on a Firmware Distribution Server and reset the server to the Idle phase by sending a Firmware Distribution Receivers Delete All message.
4. The client adds one or more entries from the Receivers List input to the Distribution Receivers List state of a Firmware Distribution Server by sending a Firmware Distribution Receivers Add message. The client shall add one Receiver Entry to the message for each entry in the Receivers List input. The fields of the Receiver Entry are set as follows:
   - The Address field of the Receiver Entry shall be set to the unicast address of the entry from the Receivers List input.
   - The Update Firmware Image Index field of the Receiver Entry shall be set to the index of the firmware subsystem of the entry from the Receivers List input.

   The number of Receiver Entries shall not cause the message payload to exceed the maximum Access PDU size.

When a Firmware Distribution Client sends a message to a Firmware Distribution Server, the client shall wait an implementation-specific amount of time for the server to respond with a Firmware Distribution Receivers Status message.

As outputs, the procedure gives a list of Target nodes from the Receivers List input that were successfully added to the Distribution Receivers List state of the server, and a list of Target nodes from the Receivers List input that were not added to the Distribution Receivers List state of the server.

The higher-layer application may use this information to restart the Populate Distribution Receivers List procedure with a different value of the Receivers List input.

7.2.2.4.1 Sending a Firmware Distribution Receivers Add message

To add Target nodes to the Distribution Receivers List state of a Firmware Distribution Server, a Firmware Distribution Client shall send a Firmware Distribution Receivers Add message. The response to a Firmware Distribution Receivers Add message is a Firmware Distribution Receivers Status message.

7.2.2.4.2 Sending a Firmware Distribution Receivers Delete All message

To clear the Distribution Receivers List state and set the Distribution Phase state to Idle for a Firmware Distribution Server, a Firmware Distribution Client shall send a Firmware Distribution Receivers Delete All message. The response to a Firmware Distribution Receivers Delete All message is a Firmware Distribution Receivers Status message.

7.2.2.4.3 Receiving a Firmware Distribution Receivers Status message

When a Firmware Distribution Client receives a Firmware Distribution Receivers Status message, it can determine the number of entries on the Distribution Receivers List state of the server.

Upon receiving a Firmware Distribution Receivers Status message, if the Status field is Success, then the client can determine the size of the Distribution Receivers List state on the server, and can determine whether the receivers were successfully added to the list. If the Status field is not Success, the Status field contains the error condition.

7.2.2.5 Distributor Capabilities procedure

The Distributor Capabilities procedure is executed on a Firmware Distribution Client to determine the value of the Distributor Capabilities state of a Firmware Distribution Server.

As output, the procedure gives the value of the Distributor Capabilities state.

The Distributor Capabilities procedure consists of the following steps:

1. The Firmware Distribution Client shall send a Firmware Distribution Capabilities Get message.
2. The client shall wait an implementation-specific amount of time for the Firmware Distribution Server to respond. The client may repeat the request to the server an implementation-specific number of times.

The Distributor Capabilities procedure completes when either of the following conditions is met:

- If the server responds, then the Distributor Capabilities procedure completes successfully.
- If the server does not respond, then the Distributor Capabilities procedure fails.

7.2.2.5.1 Sending a Firmware Distribution Capabilities Get message

A Firmware Distribution Client shall send a Firmware Distribution Capabilities Get message to determine the following:

- The maximum number of Target nodes supported by a Firmware Distribution Server.
- The maximum number of entries in the Firmware List state of a Firmware Distribution Server.
- The maximum size of one firmware image stored on a Firmware Distribution Server.
- The total storage size dedicated for firmware images on a Firmware Distribution Server.
- The available storage size dedicated for firmware images on a Firmware Distribution Server.
- The value of the Out-of-Band Retrieval Supported state of a Firmware Distribution Server.
- The value of the Supported URI Scheme Names state of a Firmware Distribution Server.

The response to a Firmware Distribution Capabilities Get message is a Firmware Distribution Capabilities Status message.

7.2.2.5.2 Receiving a Firmware Distribution Capabilities Status message

When a Firmware Distribution Client receives a Firmware Distribution Capabilities Status message from a Firmware Distribution Server, the client can check the server's firmware storage parameters, support for OOB mechanisms, and supported URI scheme names.

The client uses the values of the Out-of-Band Retrieval Supported field and the Supported URI Scheme Names field to determine whether the Upload Firmware OOB procedure can be started and a new firmware image can be transferred to the server using an OOB mechanism.

The client uses the values of the Max Distribution Receivers List Size field, Max Firmware Images List Size field, Max Firmware Image Size field, Max Upload Space field, and Remaining Upload Space field to determine the acceptable number of Target nodes, the acceptable number of firmware images, and the acceptable size of a firmware image that the server can receive.

7.2.2.6 Initiate Distribution procedure

The Initiate Distribution procedure is executed on a Firmware Distribution Client to trigger a Firmware Distribution Server to start a firmware update to the Target nodes in the Distribution Receivers List state of the server.

As inputs, the procedure takes a Distribution Multicast Address, a Distribution AppKey Index, a Distribution TTL, an Update Policy, a Distribution Timeout Base, a Distribution Transfer Mode, and a Distribution Firmware Image Index.

The Initiate Distribution procedure consists of the following steps:

1. The Firmware Distributor Client shall send a Firmware Distribution Start message with the Distribution Multicast Address field, the Distribution AppKey Index field, the Distribution TTL field, the Distribution Timeout Base field, the Distribution Transfer Mode field, the Update Policy field, and the Distribution Firmware Image Index field set to the values of the corresponding inputs.
2. The client shall wait an implementation-specific amount of time for the Firmware Distribution Server to respond. The client may repeat the request to the server an implementation-specific number of times.

The Initiate Distribution procedure completes when any of the following conditions is met:

- If the server responds with the Status field set to Success, then the Firmware Distribution Client shall store all inputs of the Initiate Distribution procedure as Initial Distribution values and the Initiate Distribution procedure completes successfully.
- If the server responds with the Status field set to any value other than Success, then the Initiate Distribution procedure fails.
- If the server does not respond, then the Initiate Distribution procedure fails.

7.2.2.6.1 Sending a Firmware Distribution Get message

To determine the status of a firmware image distribution, a Firmware Distribution Client shall send a Firmware Distribution Get message. The response to a Firmware Distribution Get message is a Firmware Distribution Status message.

7.2.2.6.2 Sending a Firmware Distribution Start message

To start or resume a firmware image distribution, a Firmware Distribution Client shall send a Firmware Distribution Start message. The response to a Firmware Distribution Start message is a Firmware Distribution Status message.

7.2.2.6.3 Sending a Firmware Distribution Cancel message

To cancel a firmware image distribution and set the Distribution Phase state to Idle for a Firmware Distribution Server, a Firmware Distribution Client shall send a Firmware Distribution Cancel

message. The response to a Firmware Distribution Cancel message is a Firmware Distribution Status message.

7.2.2.6.4 Sending a Firmware Distribution Apply message

To apply a distributed firmware image, a Firmware Distribution Client shall send a Firmware Distribution Apply message. The response to a Firmware Distribution Apply message is a Firmware Distribution Status message.

7.2.2.6.5 Receiving a Firmware Distribution Status message

When a Firmware Distribution Client model receives a Firmware Distribution Status message, the Firmware Distribution Client model can determine the Distribution Phase state of a Firmware Distribution Server.

When a Firmware Distribution Status message contains the Distribution Multicast Address field, the client can also determine the Distribution Multicast Address state, the Distribution AppKey Index state, the Distribution TTL state, the Distribution Timeout Base state, the Distribution Transfer Mode state, the Update Policy state, and the Distribution Firmware Image Index state of the server.

7.2.2.6.6 Suspend Distribution procedure

The Suspend Distribution procedure is executed on a Firmware Distribution Client to suspend a Distribute Firmware procedure on a Firmware Distribution Server.

The Firmware Distribution Client shall send the Firmware Distribution Suspend message to the Firmware Distribution Server. The response to a Firmware Distribution Suspend message is a Firmware Distribution Status message.

The Suspend Distribution procedure completes when any of the following conditions are met:

- If the server responds with the Status field set to Success, then the Suspend Distribution procedure completes successfully.
- If the server responds with the Status field set to any value other than Success, then the Suspend Distribution procedure fails.
- If the server does not respond, then the Suspend Distribution procedure fails.

7.2.2.6.7 Resume Distribution procedure

The Resume Distribution procedure is executed on a Firmware Distribution Client to resume a suspended Distribute Firmware procedure on a Firmware Distribution Server.

The Firmware Distribution Client shall send the Firmware Distribution Start message with parameters set to the Initial Distribution values of the corresponding Initiate Distribution procedure.

The Resume Distribution procedure completes when any of the following conditions are met:

- If there are no Initial Distribution values available, then the Resume Distribution procedure fails.
- If the server responds with the Status field set to Success, then the Resume Distribution procedure completes successfully.
- If the server responds with the Status field set to any value other than Success, then the Resume Distribution procedure fails.
- If the server does not respond, then the Resume Distribution procedure fails.

7.2.2.7 Get Distribution Progress procedure

The Get Distribution Progress procedure is executed on a Firmware Distribution Client to retrieve the progress of a firmware update for each Target node on the Distribution Receivers List state of a Firmware Distribution Server.

As outputs, the procedure gives a list of Target nodes with the Update Phase state, firmware progress information, the status of the BLOB transfer, and the status of firmware image distribution (i.e., the Result List).

The Get Distribution Progress procedure consists of the following steps:

1. The client shall send a Firmware Distribution Receivers Get message with the First Index field set to 0 and the Entries Limit field set to an implementation-specific value greater than 0.
2. The client shall wait an implementation-specific amount of time for the Firmware Distribution Server model to respond. The client may try to get the response for the request from the server an implementation-specific number of times.
3. The client shall add the entries from the Receivers List field of the message to the Result List.

If the Receivers List field does not contain information about all Target nodes in the Distribution Receivers List state of the server (i.e., the sum of the First Index field and the number of entries in the Receivers List field is less than the value of the Receivers List Count field), then the client shall send additional Firmware Distribution Receivers Get messages to retrieve the information for the remaining Target nodes.

The Get Distribution Progress procedure completes when any of the following conditions is met:

- If the information has been retrieved for all Target nodes in the Distribution Receivers List state of the server, then the Get Distribution Progress procedure completes successfully.
- If the server responds with the Status field set to any value other than Success, then the Get Distribution Progress procedure fails.
- If the server does not respond, then the Get Distribution Progress procedure fails.

7.2.2.7.1 Sending a Firmware Distribution Receivers Get message

To determine the status of a firmware update, a Firmware Distribution Client shall send a Firmware Distribution Receivers Get message. The response to a Firmware Distribution Receivers Get message is a Firmware Distribution Receivers List message.

7.2.2.7.2 Receiving a Firmware Distribution Receivers List message

When a Firmware Distribution Client receives a Firmware Distribution Receivers List message, the Firmware Distribution Client can determine the status of a firmware update on a set of Target nodes and the size of the Distribution Receivers List state of a Firmware Distribution Server.

7.2.2.8 Upload Firmware OOB procedure

The Upload Firmware OOB procedure is executed on a Firmware Distribution Client to trigger a firmware image transfer to a Firmware Distribution Server by using an OOB mechanism.

As inputs, the procedure takes an Upload OOB URI and an Upload OOB Firmware ID.

The Upload Firmware OOB procedure consists of the following steps:

1. The Firmware Distribution Client shall send a Firmware Distribution Upload OOB Start message with the Upload OOB URI Length field, the Upload OOB URI field, and the Upload OOB Firmware ID field set to the value of the corresponding input. The client shall wait an implementation-specific amount of time for the Firmware Distribution Server to respond with a Firmware Distribution Upload Status message (see Section 7.2.2.12).
2. The client may try to get the response for the request from the server an implementation-specific number of times to get the status of the firmware upload progress. The client may repeat either a Firmware Distribution Upload OOB Start message or a Firmware Distribution Upload Get message to solicit a response. If the response message has the Status field set to Success and

the Upload Phase field set to Transfer Active, the client can repeat step 2 after an implementation-specific amount of time.

The Upload Firmware OOB procedure completes when any of the following conditions is met:

- If the server responds with the Status field set to Success and Upload Phase field set to Transfer Success, then the procedure completes successfully.
- If the server responds with the Status field set to any value other than Success or Upload Phase field set to Transfer Error, then the procedure fails.
- If the server does not respond, then the procedure fails.

7.2.2.8.1 Sending a Firmware Distribution Upload OOB Start message

To start OOB delivery of a firmware image to a Firmware Distribution Server, a Firmware Distribution Client sends a Firmware Distribution Upload OOB Start message, setting the Upload OOB URI field and the Upload OOB Firmware ID to the intended new values. The Upload OOB URI Length field shall be set to the size of the Upload OOB URI field. The response to a Firmware Distribution Upload OOB Start message is a Firmware Distribution Upload Status message.

7.2.2.9 Upload Firmware procedure

The Upload Firmware procedure is executed on a Firmware Distribution Client to transfer a firmware image to a Firmware Distribution Server.

As inputs, the procedure takes a Firmware Image, Upload Firmware Metadata, an Upload Firmware ID, an Upload TTL, an Upload AppKey Index, and an Upload Timeout Base.

The Upload Firmware procedure consists of the following steps:

1. The client shall generate an 8-octet BLOB ID value. The BLOB ID should be unique, and the method of generating the BLOB ID value is implementation specific.
2. The client shall send a Firmware Distribution Upload Start message with the Upload BLOB ID field set to the generated BLOB ID value, and the Upload Firmware Size field, the Upload TTL field, the Upload Timeout Base field, the Upload Firmware Metadata field, the Upload Firmware Metadata Length field, and the Upload Firmware ID field set to the values of the corresponding inputs.
3. The client shall wait an implementation-specific amount of time for the Firmware Distribution Server to respond with a Firmware Distribution Upload Status message (see Section 7.2.2.12). The client may try to get the response to the request from the server an implementation-specific number of times.
4. The client shall start the Transfer BLOB procedure on the BLOB Transfer Client model with the following inputs:
    - The Address List input is set to the unicast address of the Distributor's element that includes the BLOB Transfer Server used by the Firmware Distribution Server.
    - The Multicast Address input is set to the Unassigned Address.
    - The AppKey Index input is set to the Upload AppKey Index input of the Upload Firmware procedure.
    - The Transfer TTL input is set to the Upload TTL input of the Upload Firmware procedure.
    - The BLOB ID input is set to the generated BLOB ID.
    - The BLOB input is set to the Firmware Image input of the Upload Firmware procedure.
    - The Transfer Mode input is set to Push BLOB Transfer Mode.

- The Client Timeout Base input is set to the Upload Timeout input of the Upload Firmware procedure.

The Upload Firmware procedure may be suspended by the upper layer by suspending the Transfer BLOB procedure from the BLOB Transfer Client. If the Transfer BLOB procedure from the BLOB Transfer Client cannot be suspended, then the Upload Firmware procedure cannot be suspended. The Upload Firmware procedure may be resumed by the upper layer by resuming the Transfer BLOB procedure. When the Transfer BLOB procedure from the BLOB Transfer Client is suspended, the Upload Firmware procedure shall be suspended.

The Upload Firmware procedure completes when any of the following conditions is met:

- If the Transfer BLOB procedure completes successfully, then the Upload Firmware procedure completes successfully.
- If the Transfer BLOB procedure fails, then the Upload Firmware procedure fails.
- If the Firmware Distribution Server responds with a Firmware Distribution Upload Status message with the Status field not equal to Success, then the Upload Firmware procedure fails.
- If the Firmware Distribution Server does not respond, then the Upload Firmware procedure fails.

7.2.2.9.1 Sending a Firmware Distribution Upload Get message

To determine the Upload Phase state, the Upload Progress state, the Upload Type state, and either the Upload Firmware ID state or the Upload OOB Firmware ID state of a Firmware Distribution Server, a Firmware Distribution Client shall send a Firmware Distribution Upload Get message. The response to a Firmware Distribution Upload Get message is a Firmware Distribution Upload Status message.

7.2.2.9.2 Sending a Firmware Distribution Upload Start message

To start a firmware image upload to a Firmware Distribution Server, a Firmware Distribution Client shall send a Firmware Distribution Upload Start message, setting the Upload TTL field, the Upload Timeout Base field, the Upload BLOB ID field, the Upload Firmware Size field, and the Upload Firmware ID field to the intended new values. When the firmware image's associated metadata is present, the Upload Firmware Metadata Length field shall be set to the size of the metadata and the Upload Firmware Metadata field shall contain image metadata. When the firmware image's associated metadata is not present, the Upload Firmware Metadata Length field shall be set to 0 and the Upload Firmware Metadata field shall be omitted. The response to a Firmware Distribution Upload Start message is a Firmware Distribution Upload Status message.

7.2.2.9.3 Sending a Firmware Distribution Upload Cancel message

To cancel an ongoing firmware image upload to a Firmware Distribution Server, a Firmware Distribution Client shall send a Firmware Distribution Upload Cancel message. The response to a Firmware Distribution Upload Cancel message is a Firmware Distribution Upload Status message.

7.2.2.10 Firmware Images List Management procedure

The Firmware Images List Management procedure is executed on a Firmware Distribution Client to manage the firmware images stored in the Firmware Images List state of a Firmware Distribution Server.

To manage the Firmware Images List state, a Firmware Distribution Client sends Firmware Distribution Firmware Get, Firmware Distribution Firmware Get By Index, Firmware Distribution Firmware Delete, or Firmware Distribution Firmware Delete All messages.

Before transferring a firmware image to a Firmware Distribution Server, a Firmware Distribution Client should check if the image is already stored on the server by sending a Firmware Distribution Firmware Get message or a Firmware Distribution Firmware Get By Index message.

A Firmware Distribution Client may also delete one or more firmware images stored on a Firmware Distribution Server by sending a Firmware Distribution Firmware Delete message or a Firmware Distribution Firmware Delete All message.

7.2.2.10.1 Sending a Firmware Distribution Firmware Get message

To determine whether a Firmware Distribution Server includes a specific firmware image in the Firmware Images List state, a Firmware Distribution Client shall send a Firmware Distribution Firmware Get message with the Firmware ID field identifying the firmware image to check. The response to a Firmware Distribution Firmware Get message is a Firmware Distribution Firmware Status message.

7.2.2.10.2 Sending a Firmware Distribution Firmware Get By Index message

To determine the Firmware ID of a firmware image stored with a specific index in the Firmware Images List state of a Firmware Distribution Server, a Firmware Distribution Client shall send a Firmware Distribution Firmware Get By Index. The response to a Firmware Distribution Firmware Get By Index message is a Firmware Distribution Firmware Status message.

7.2.2.10.3 Sending a Firmware Distribution Firmware Delete message

To delete a firmware image stored on a Firmware Distributor Server, a Firmware Distribution Client shall send a Firmware Distribution Firmware Delete message with the Firmware ID field identifying the firmware image to delete. The response to a Firmware Distribution Firmware Delete message is a Firmware Distribution Firmware Status message.

7.2.2.10.4 Sending a Firmware Distribution Firmware Delete All message

To delete all firmware images and set the Distribution Phase state to Idle for a Firmware Distribution Server, a Firmware Distribution Client shall send a Firmware Distribution Firmware Delete All message. The response to a Firmware Distribution Firmware Delete All message is a Firmware Distribution Firmware Status message.

7.2.2.10.5 Receiving a Firmware Distribution Firmware Status message

When a Firmware Distribution Client receives a Firmware Distribution Firmware Status message, it can determine the Status Code of the last operation on the Firmware Images List state and the size of the Firmware Images List state.

When the Status field of the message is Success, and the Distribution Firmware Image Index field of the message is not 0xFFFF, the client can determine the index of the firmware image with the Firmware ID identified by the Firmware ID field of the message.

When the Status field of the message is Success, and the Distribution Firmware Image Index field is 0xFFFF, the firmware image identified by the Firmware ID field is not present on the Firmware Images List state.

When the Status field of the message is not Success, the client can determine the error that occurred.

7.2.2.11 Additional procedures supported

A Firmware Distribution Client may support the Firmware Retrieval Over HTTPS procedure and the Firmware Check Over HTTPS procedures.

7.2.2.12 Receiving a Firmware Distribution Upload Status message

When a Firmware Distribution Client receives a Firmware Distribution Upload Status message, the client can determine the Upload Phase state of the server and the status of a firmware upload.

When the Upload Progress field is present, it indicates the upload progress of the firmware image.

When the Upload Type field is present, it indicates the Upload Type state of the server, and indicates that either the Upload Firmware ID field or the Upload OOB Firmware ID is present in the message.

When the Upload Firmware ID field is present, it indicates the Upload Firmware ID state of the server.

When the Upload OOB Firmware ID field is present, it indicates the Upload OOB Firmware ID state of the server.

8 Message flows and example sequence charts

This section illustrates common scenarios of a firmware update and does not cover all possible alternatives.

8.1 Retrieve information about firmware installed on a Target node

Figure 8.1 illustrates an example of how an Initiator can retrieve the information about the firmware subsystems installed on a set of Target nodes and the URI used to check for a new firmware image to perform a firmware update.

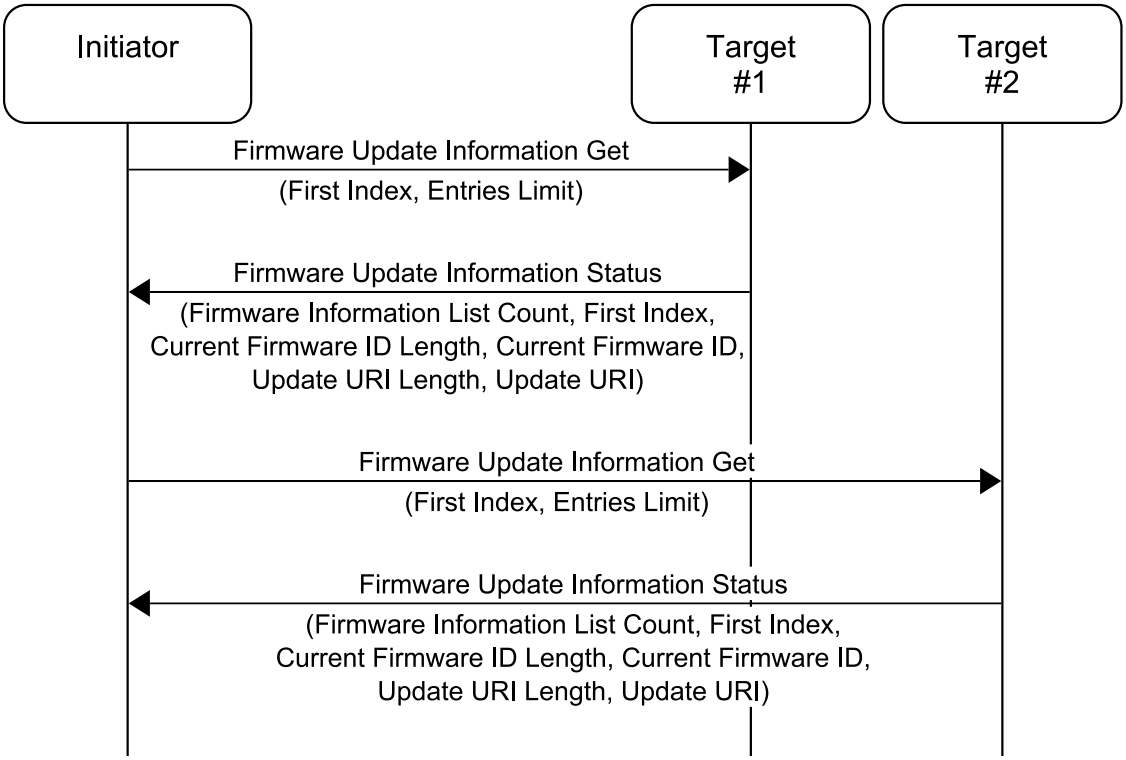

Figure 8.1: Example of Initiator retrieving information about the firmware subsystems installed on Target nodes

8.2 Firmware image retrieval over HTTP protocol using secure transport

Figure 8.2 illustrates the flow of messages when an Initiator retrieves the information about the firmware subsystems installed on a Target node, checks an HTTP server for availability of a new firmware image, and triggers a Distributor to download the new firmware image from the HTTP server.

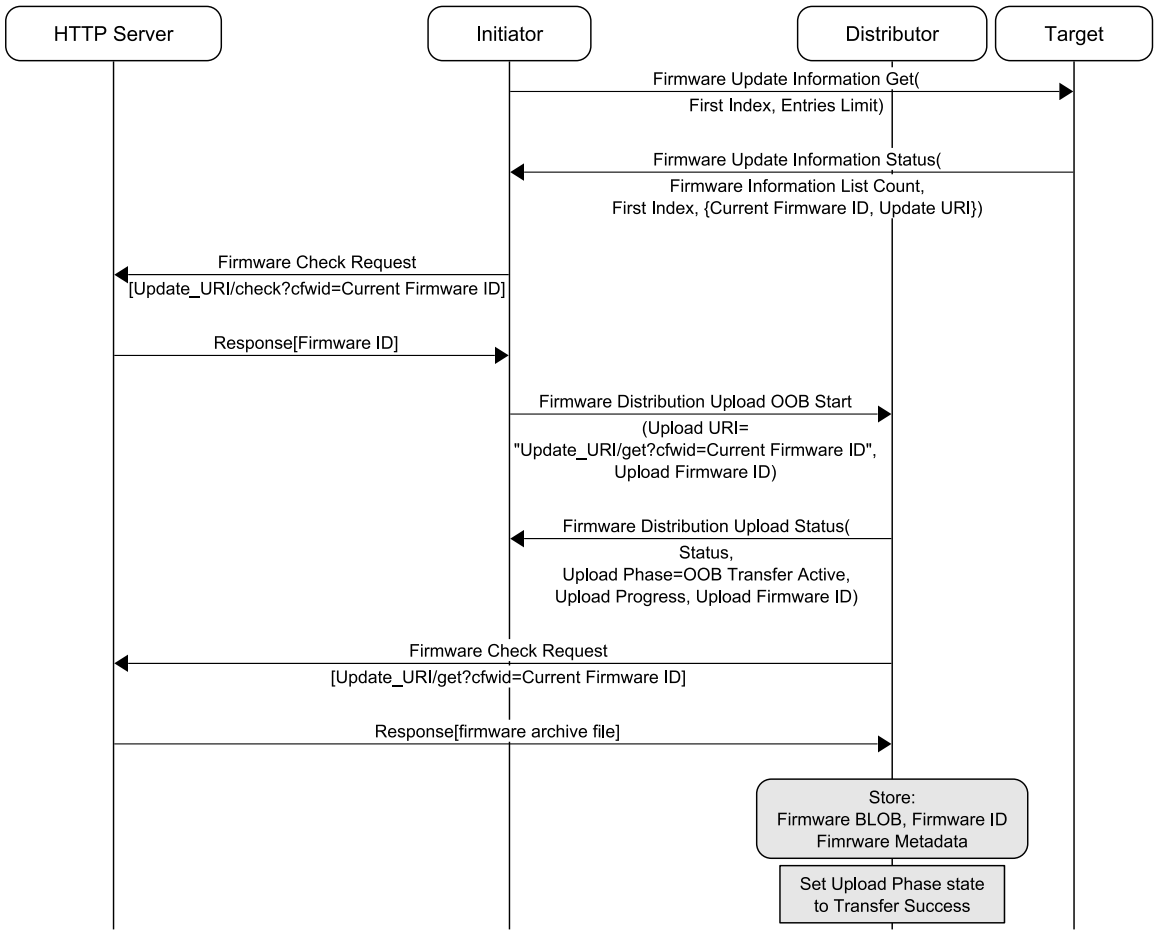

Figure 8.2: Example of using HTTP to check availability of a new firmware image and download the new firmware image

8.3 Initiator updating the receivers list for a firmware update

Figure 8.3 illustrates an example of an Initiator checking the capabilities of a Distributor to determine the maximum number of Target nodes that can be added to the Distribution Receivers List state, then querying Target nodes in the network to check which nodes will accept a new firmware image so that these Target nodes can be added to the Distribution Receivers List state.

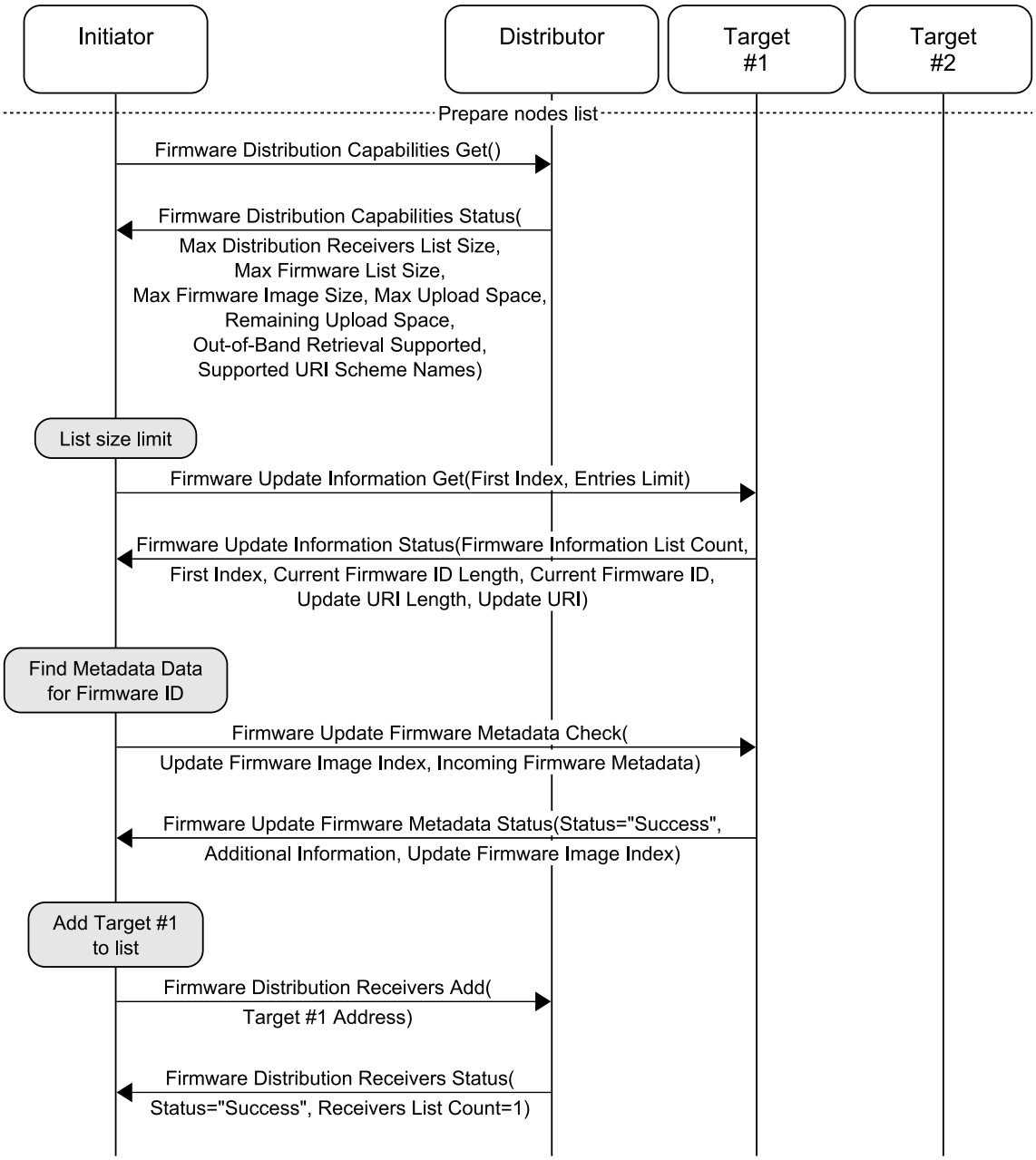

Figure 8.3: Example of Initiator checking Distributor capabilities, and checking firmware image compatibility on the Target nodes to construct the nodes list

8.4 Managing firmware images on the Distributor

Figure 8.4 illustrates an example of how an Initiator manages the firmware images stored on a Distributor.

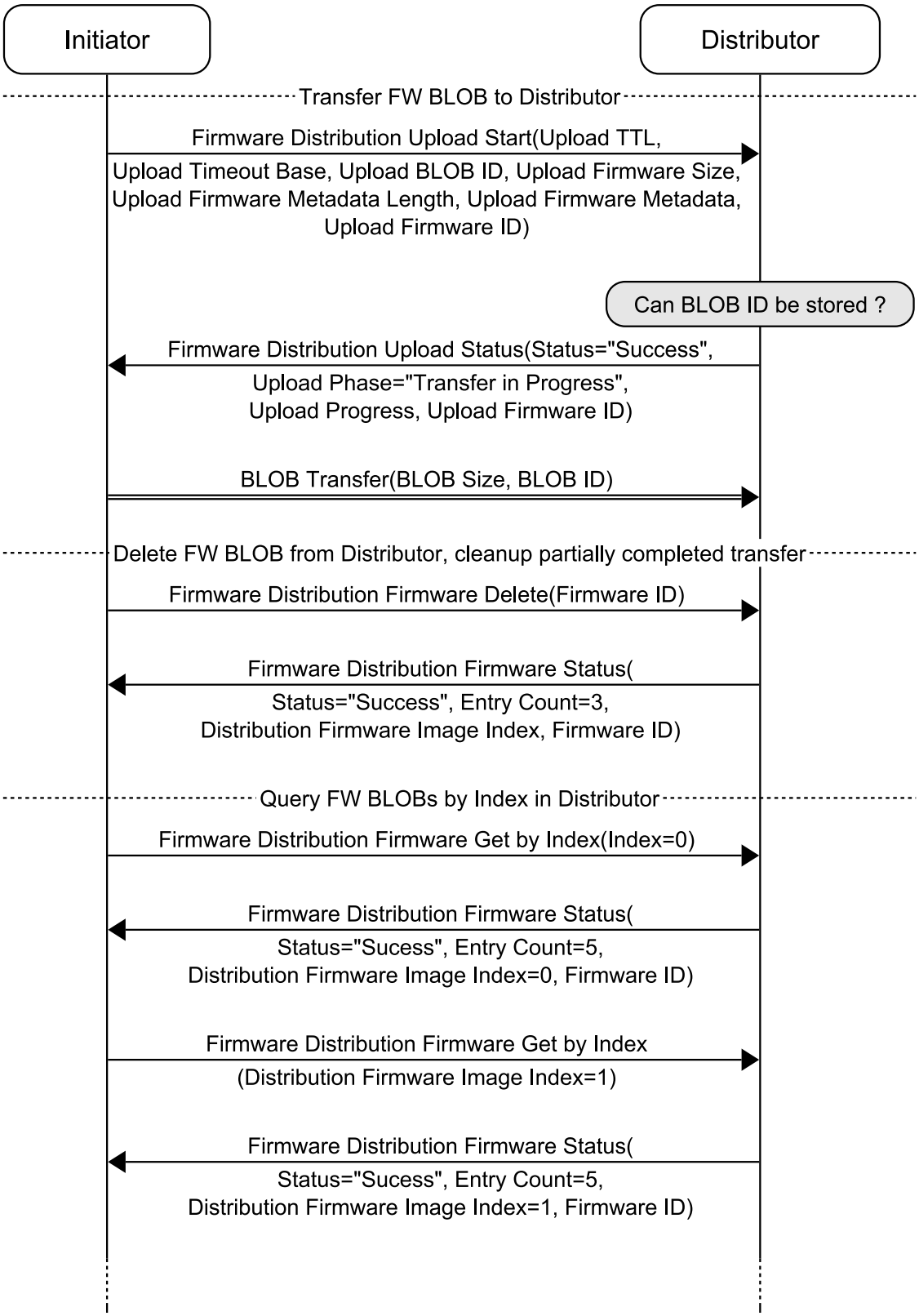

Figure 8.4: Example of managing firmware images on a Distributor

8.5 Starting firmware image distribution

Figure 8.5 illustrates an example of an Initiator adding a set of Target nodes to the Distribution Receivers List state of a Distributor and starting firmware image distribution to the Target nodes.

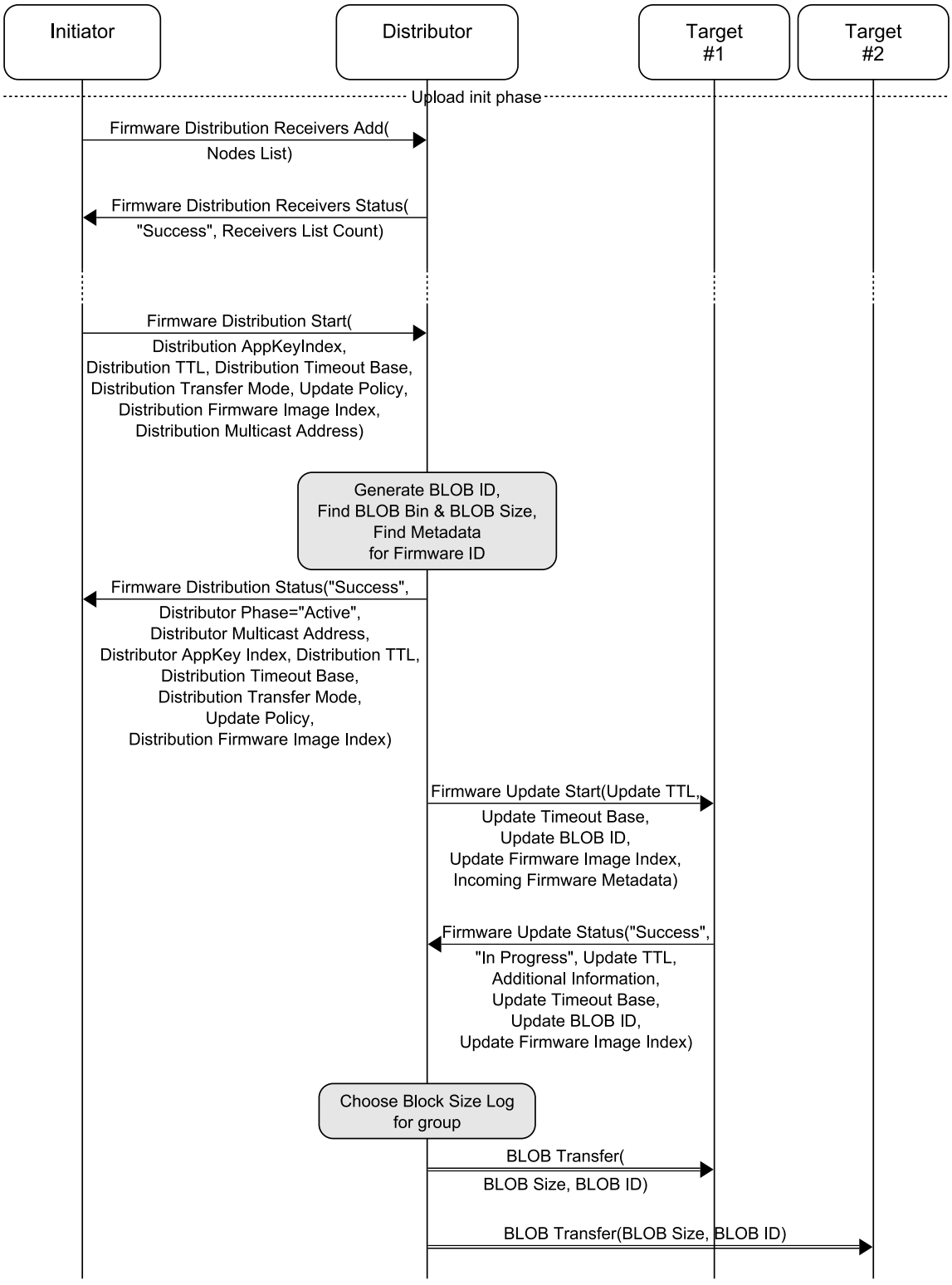

Figure 8.5: Communication between the Initiator, the Distributor, and Target nodes when firmware image distribution is started

Figure 8.6 illustrates an example of error conditions that a Distributor can return when an Initiator attempts to start a firmware image distribution.

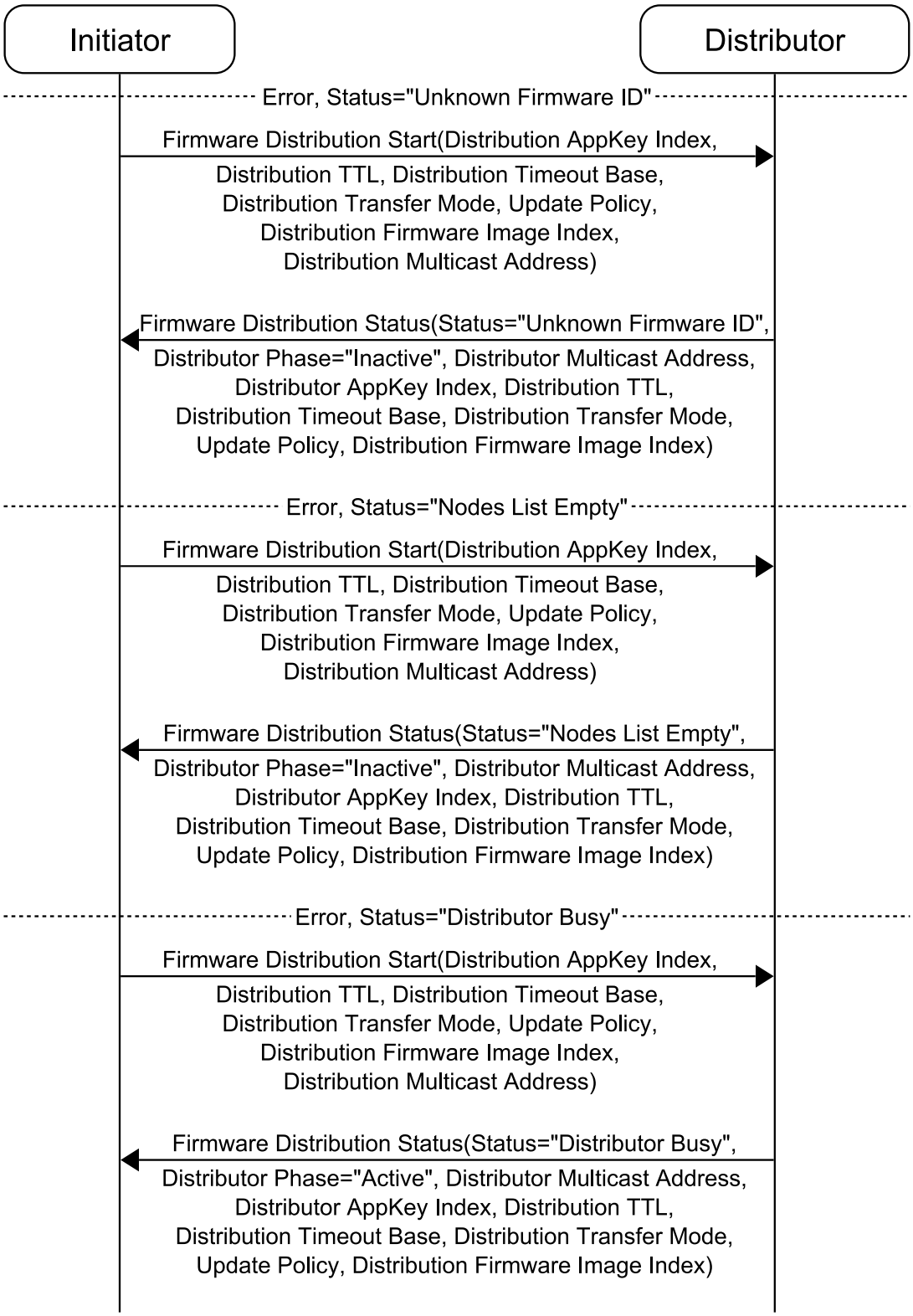

Figure 8.6: Error responses from a Distributor when an Initiator attempts to start firmware image distribution

Figure 8.7 illustrates an example of the status codes from a Target node when a Distributor attempts to start a firmware image transfer.

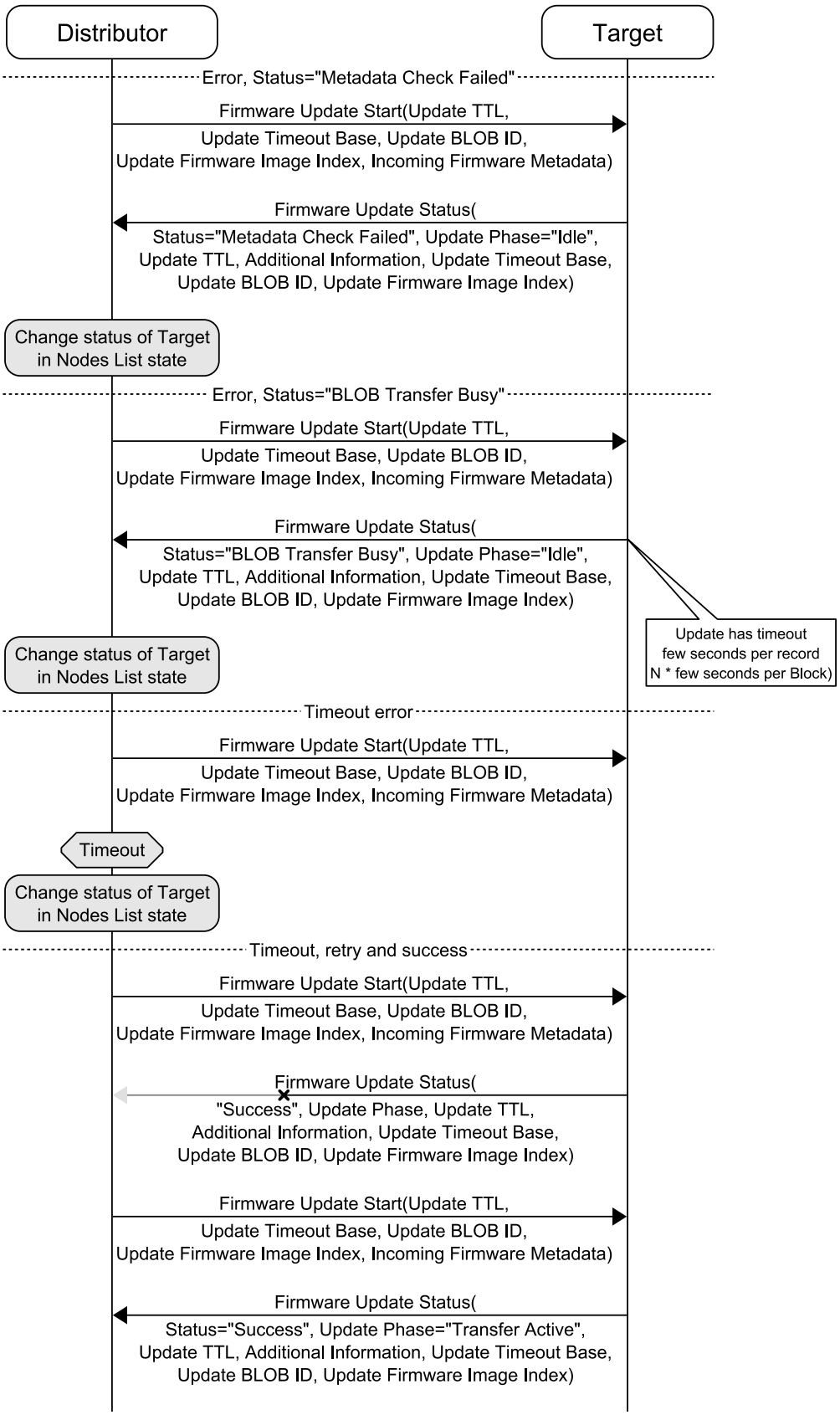

Figure 8.7: Error conditions from a Target node when a Distributor attempts to start a firmware image transfer

8.6 Checking distribution progress

Figure 8.8 illustrates an example of an Initiator querying a Distributor for BLOB transfer progress.

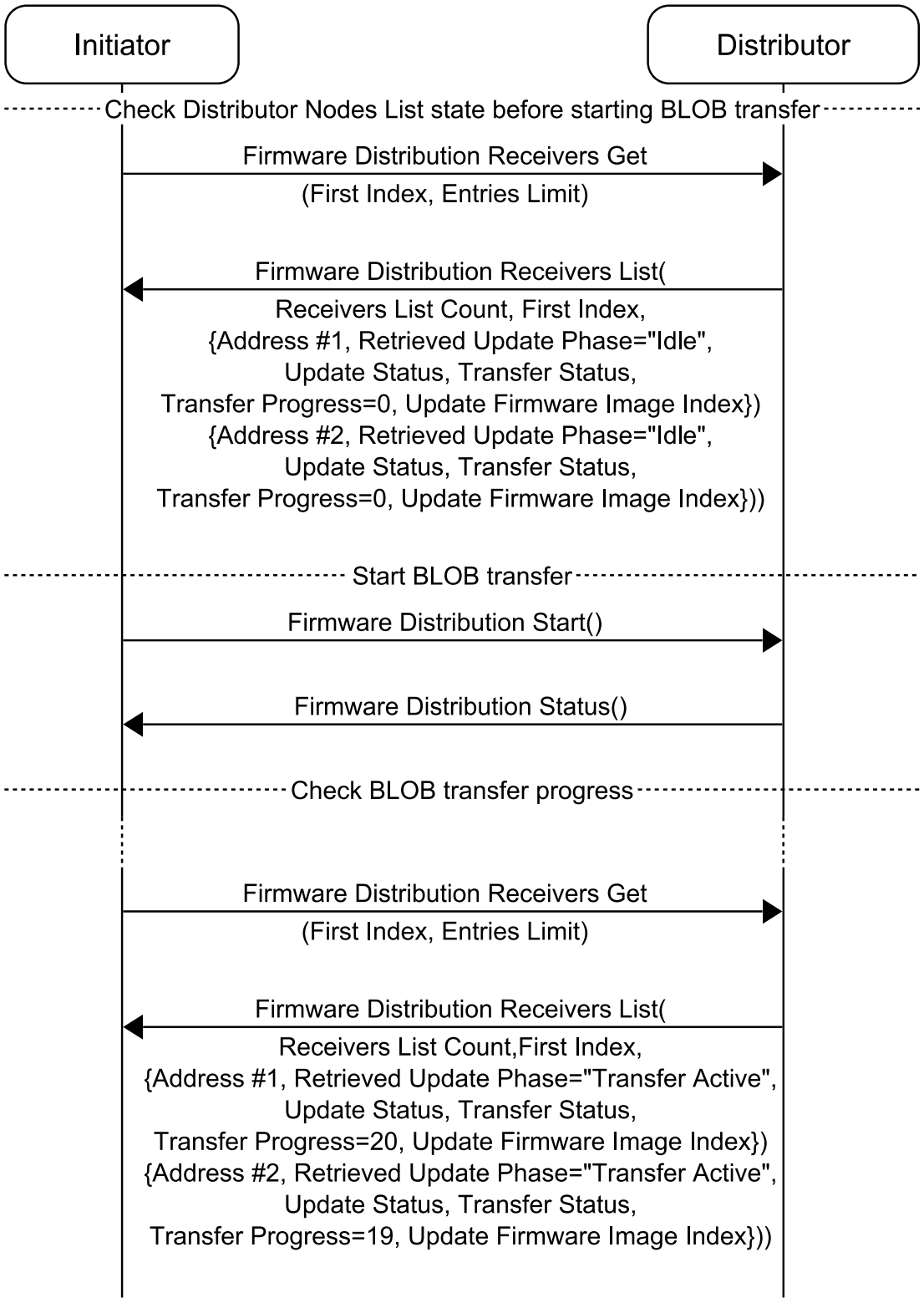

Figure 8.8: Example of checking the firmware update status of the Target nodes

8.7 Applying a firmware update

Figure 8.9 illustrates an example of the application of a firmware update on a set of Target nodes when the Update Policy for this firmware update is Verify And Apply. The Initiator checks the result of the firmware update by querying the Distributor.

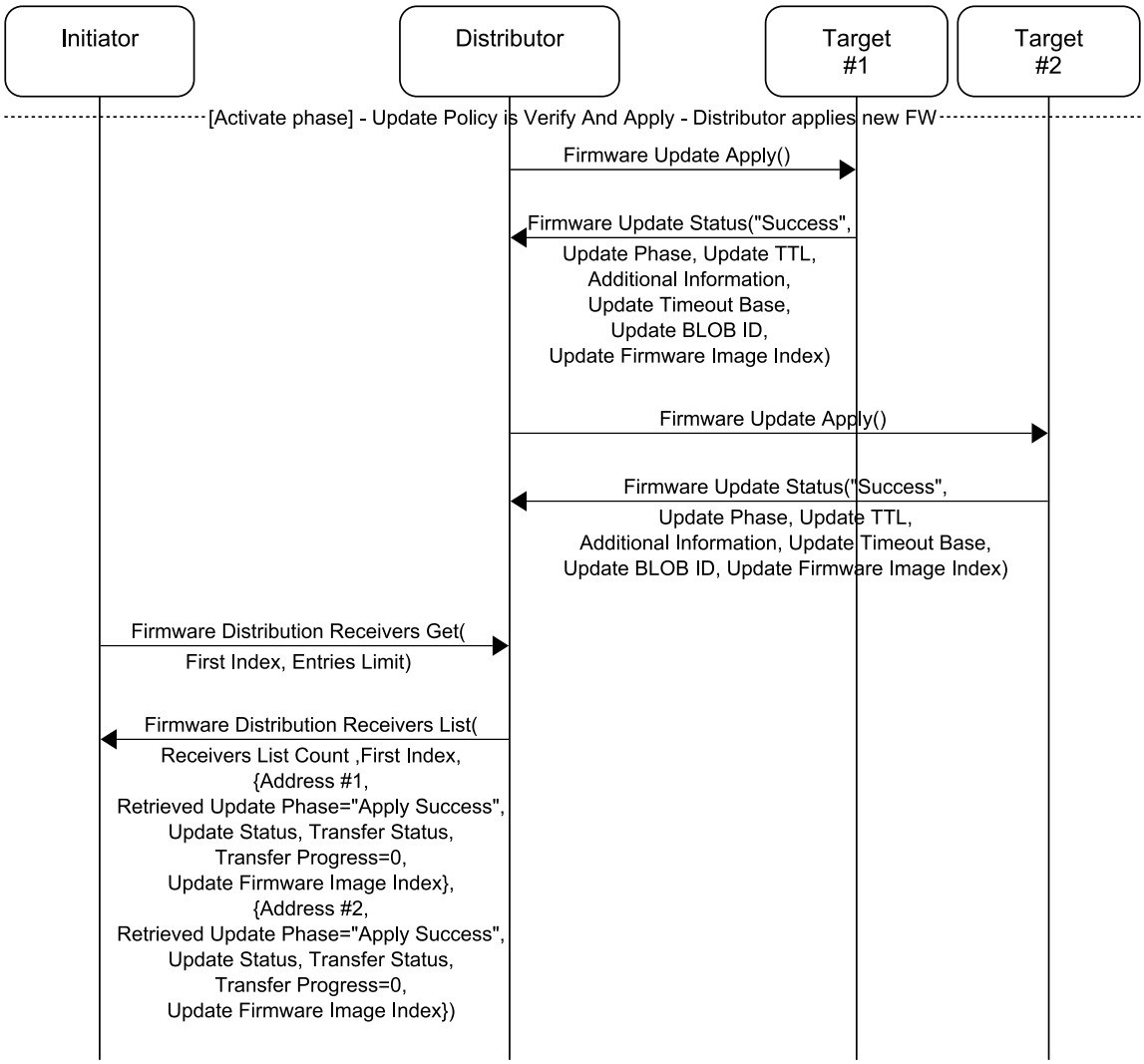

Figure 8.9: Example of a Distributor applying a firmware update and an Initiator checking the progress of the firmware update when the Update Policy is Verify And Apply

Figure 8.10 illustrates an example of the application of a firmware update on a set of Target nodes when the Update Policy of this firmware update is Verify Only.

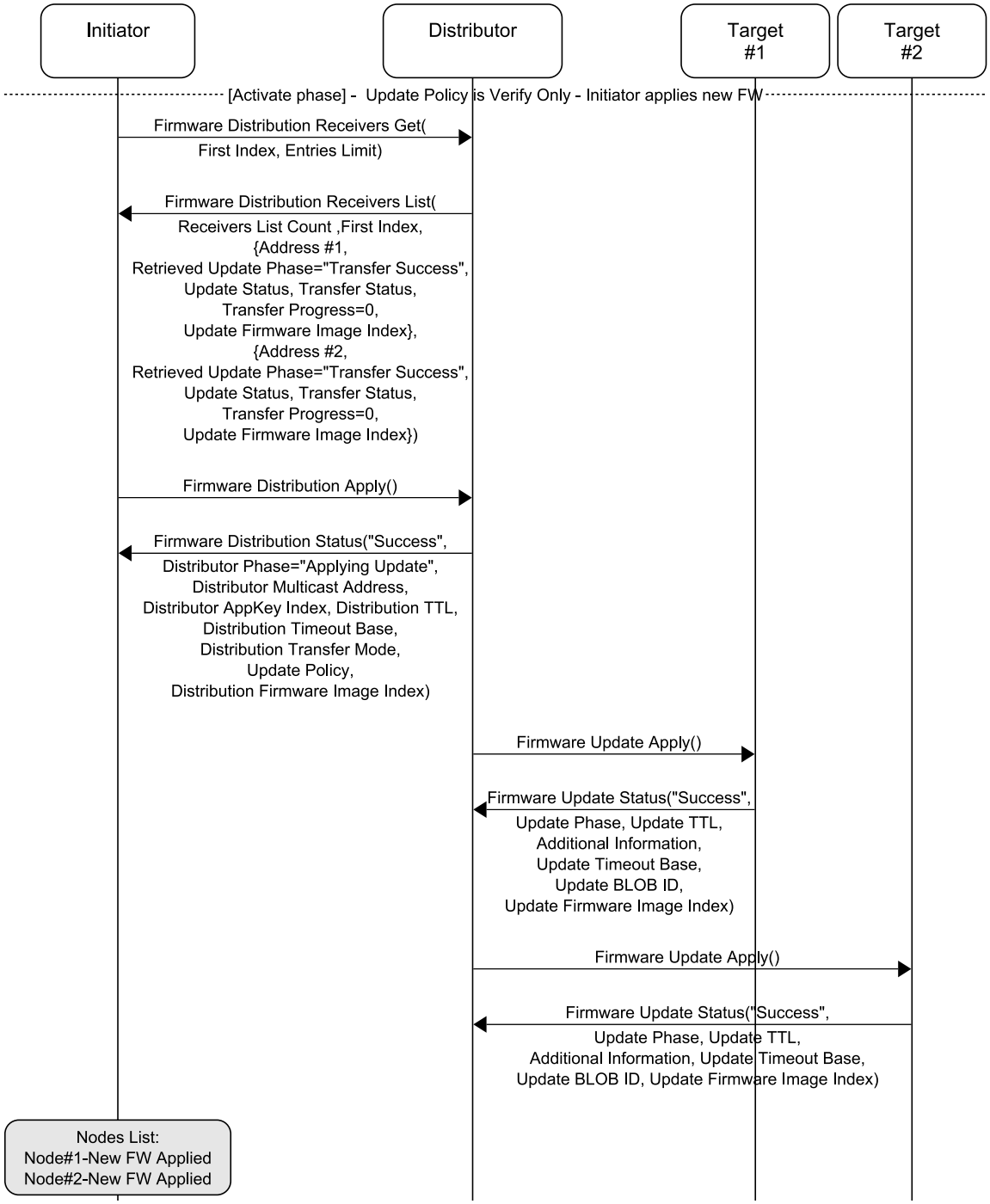

Figure 8.10: Example of an Initiator applying a firmware update when the Update Policy is Verify Only

9 Summary

The Assigned Numbers document [9] provides a summary of device firmware messages and models, as follows:

- **Device firmware messages summary:** The definitions of messages, and their opcodes, that are available for each of the mesh models
- **Device firmware models summary:** All models defined in this specification and their corresponding Model IDs

10 Acronyms and abbreviations

Acronym/Abbreviation	Meaning
BLOB	binary large object
CD	Composition Data
JSON	JavaScript Object Notation
OOB	out-of-band
PDU	Protocol Data Unit
RFU	Reserved for Future Use
TTL	time to live
URI	Uniform Resource Identifier

Table 10.1: Acronyms and abbreviations

11 References

[1] Mesh Protocol Specification, Version 1.1 or later

[2] Internet Engineering Task Force (IETF), “Internet Standard (STD) 66 – Uniform Resource Identifier (URI): Generic Syntax”, January 2005, https://tools.ietf.org/html/std66

[3] National Institute of Standards and Technology (NIST), “Special Publication (SP) 800-193 – Platform Firmware Resiliency Guidelines”, May 2018, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-193.pdf

[4] IETF, “Request for Comment (RFC) 4648 – The Base16, Base32, and Base64 Data Encodings”, October 2006, http://www.ietf.org/rfc/rfc4648.txt

[5] IETF, “RFC 1952 – GZIP file format specification version 4.3”, May 1996, https://tools.ietf.org/html/rfc1952

[6] IETF, “RFC 8259 – The JavaScript Object Notation (JSON) Data Interchange Format”, December 2017, https://tools.ietf.org/html/rfc8259

[7] JSON Schema Draft 7, http://json-schema.org/specification-links.html#draft-7

[8] Core Specification Supplement (CSS), Version 11 or later

[9] Bluetooth SIG Assigned Numbers, https://www.bluetooth.com/specifications/assigned-numbers/

[10] Mesh Binary Large Object Transfer Model Specification, Version 1.0 or later