AN1424: Bluetooth® Mesh 1.1 Network Performance

This application note highlights performance metrics of new Bluetooth Mesh 1.1 features and provides a performance overview of Silicon Labs's Bluetooth Mesh software and hardware.

This application note demonstrates how the Bluetooth Mesh network performs overall, with detailed results on the performance of Remote Provisioning and Over-the-Air Device Firmware Update (OTA DFU), both introduced in the 1.1 version of the Bluetooth Mesh specification. The test environment was a commercial office building with active Wi-Fi and Bluetooth Low Energy (LE) networks in range. Wireless test clusters were deployed in hallways, meeting rooms, offices, and open areas. The methodology for performing the benchmarking tests is defined so that others may run the same tests in their environment. These results are intended to provide guidance on design practices and principles as well as expected field performance results.

KEY POINTS

- Wireless test network in Silicon Labs Research and Development (R&D) office is described.
- Wireless conditions and environments are evaluated.
- Mesh 1.1 network performance including throughput, latency, and large network scalability is presented.

1 Introduction and Background

Silicon Labs has provided performance testing results from embedded mesh networks as part of developer conferences and industry white papers. The basic performance data of throughput, latency, and impact of security can be used by system designers to define expected behavior. This testing has previously been presented for Zigbee and Thread networks as basic 15.4 mesh networking technologies. These were presented because performance varies even though both systems use the same underlying physical layer defined by IEEE 802.15.4. With the advent of Bluetooth mesh networks, it is common for questions to arise concerning the expected performance differences between a Bluetooth mesh and these 15.4 mesh networks. Prior to discussing the testing and performance differences, we need to review the underlying technologies of these networks as it helps to understand their performance differences.

1.1 Underlying Physical Layer and Packet Structure

Network performance is based on payload sizing since the application usage does not account for the packet overhead.

Bluetooth low energy is based on the BT 4.x specification and has a 33-byte packet with an underlying data rate of 1 Mbps. The Bluetooth mesh packet size is shown in the figure below and results in a 12 or 16-byte payload. For payloads above 12 bytes, there is a process of segmentation and reassembly.

Bluetooth mesh has a higher data rate but the packet payload is smaller; therefore, it takes more packets to send the same amount of data. Our data on performance is based on payload size as this is the design parameter of concern when building an application. The Bluetooth mesh has designed mesh profiles (application layer) to specifically minimize the packet payload and fit in a single packet where possible.

	1		1	3	2	2	12 or 16	4 or 8
IVI	Network ID	CTL	TTL	Sequence Number	Source Address	Dest Address	Packet Payload	NWK MIC

Bluetooth Mesh Packet Format

1.2 Network Routing Differences

Bluetooth mesh uses a managed flooding technique to relay messages instead of routing. This means that instead of building, maintaining, and using defined routes to send messages, Bluetooth mesh relays relay the messages they hear using the following two simple rules:

1. Every message has a unique sequence number.
2. Relays keep track of the recently seen sequence numbers and do not relay the messages they have already seen or relayed before.

The messages also have a time-to-live counter (TTL) and every time a message is relayed, the counter is reduced by one until it reaches a value indicating that it should no longer be relayed.

As there are no acknowledgements used at the network level, Bluetooth mesh relays can be configured to repeat the same message multiple times so that more reliability is achieved due to the air interface packet loss. Typically, this value is set to 3, so each relay repeats the same message three times. Also, a configurable delay between the repetitions is used to optimize between latency and network performance. The minimum delay between repetitions, called the retransmission interval = (Relay Retransmit Interval Steps + 1) * 10ms + 0-10ms random delay, is typically 15 ms per hop.

Note: This performance data is for the Silicon Labs implementation of these mesh networking stacks. As is shown in the test network and infrastructure provided for this testing, no tests were performed with other stacks or systems.

This application note also focuses on the performance of new features introduced in the Bluetooth Mesh 1.1 specification, such as Remote Provisioning and OTA DFU.

2 Goals and Methods

This application note defines a set of tests performed to evaluate mesh network performance, scalability, and reliability and to evaluate the performance of the new BT Mesh 1.1 features. The test conditions and infrastructure are described for these tests, which are conducted over actual wireless devices in test networks and not in simulation.

This testing was done to provide an overview of the Remote Provisioning and OTA DFU performance and to characterize the network used for these tests with the help of the throughput and latency tests.

Since the tests were carried out in a real-life environment, background noise was present at the time of the tests. These noises come from the Bluetooth/Wi-Fi devices in the office used by the employees, as well as from the other test benches in the office, which are in the order of hundreds. While it might be useful to execute the tests in such a noisy environment, some noise reduction was implemented.

For the Remote Provisioning performance test, a multi-hop setup was used, where the performance based on the number of hops in the network was measured.

For the OTA DFU performance test, since the substantial number of nodes prevented having a shielded environment, the tests were carried out in timeslots with no human presence in the office, and with the testbenches turned off on the whole floor. This enabled testing in a slightly more regulated environment.

3 Test Network and Conditions

For Remote Provisioning performance testing, Silicon Labs built an RF-shielded environment to filter out the ongoing interference in the office area.

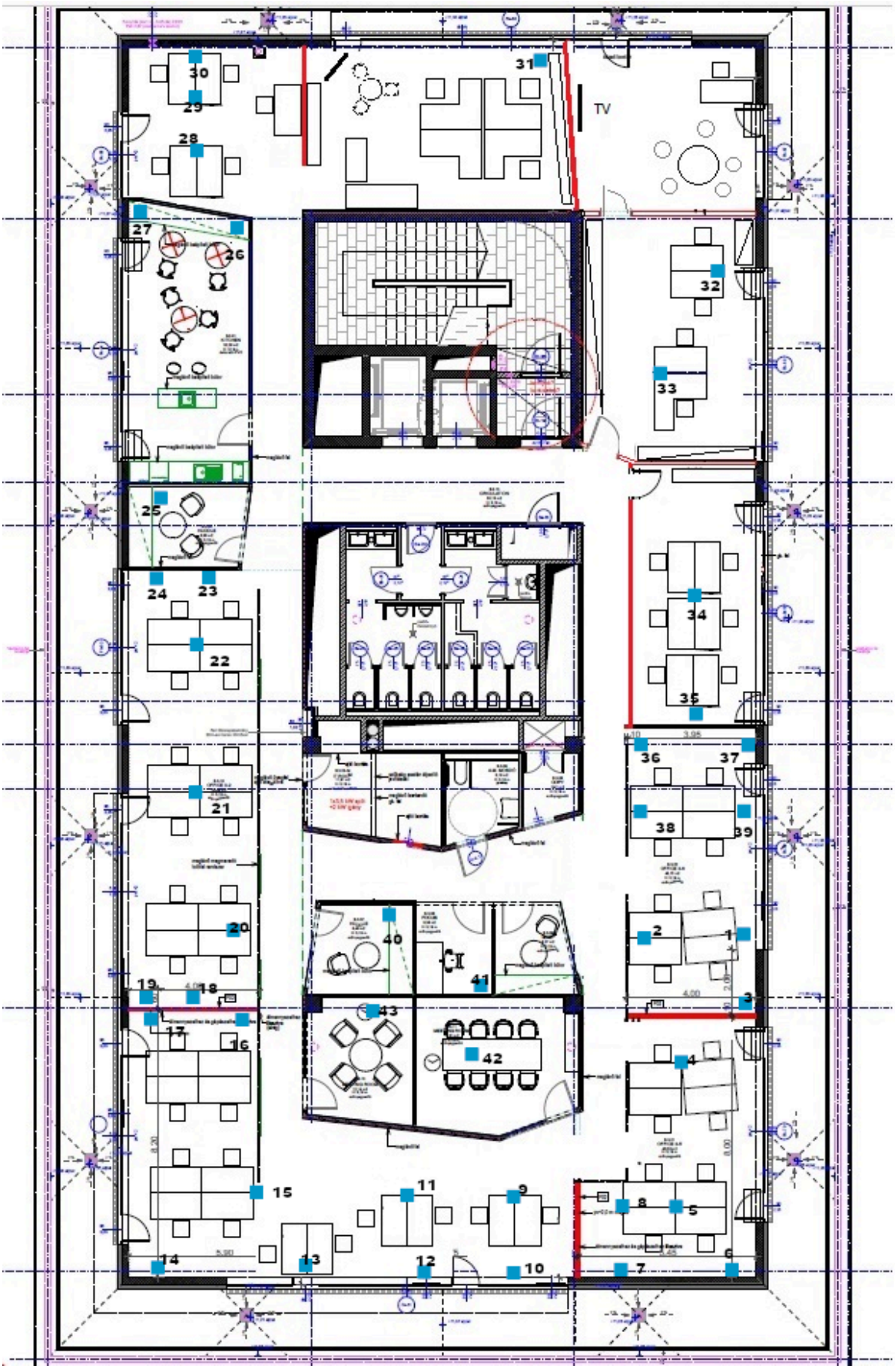
Large network testing is best conducted in an open-air environment for multiple reasons. The existing and varying RF conditions make this environment closer to a real-life scenario. However, the tests were conducted in the evening of workdays, and on the weekend, to eliminate some of the noise created by the human work in the office. The Silicon Labs Budapest office was used for this open-air testing.

3.1 Office and Test Network Conditions

The multicast latency and the OTA DFU tests were carried out on the large network test setup located in the office. A total of 43 boxes were scattered around the floor, each containing 4-6 devices to create a 256-node network on a large area with naturally occurring hops in the network.

Each box contained six Silicon Labs Wireless Starter Kits (WSTKs), except one, which contained four WSTKs. The first 27 boxes had four BRD4186C (EFR32xG24) and two BRD4181B (EFR32xG21) radio boards each. Boxes 28-42 had three BRD4186C and three BRD4181B radio boards. Box 43 had four BRD4186C radio boards and one BRD4181B radio board.

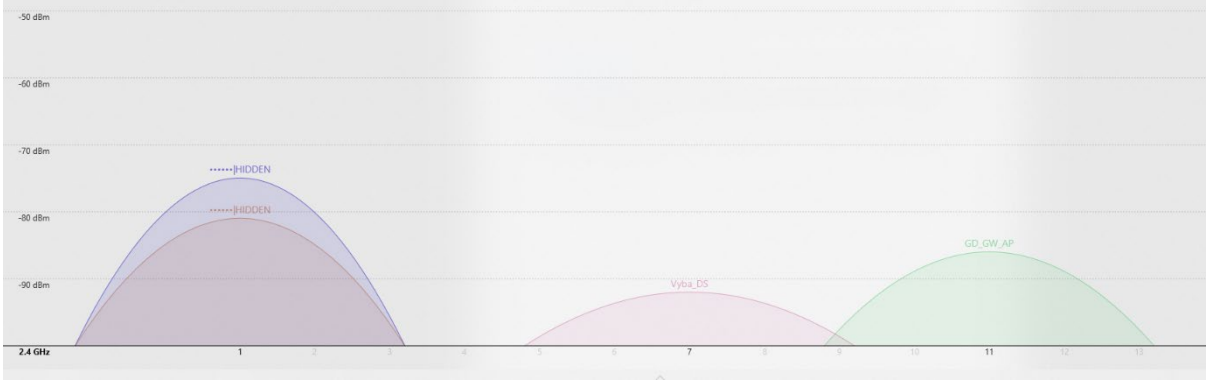
The office is rectangular in shape, with sides of 38m and 19m. There is an 18.5m by 7.5m area where devices were not placed because of stairs, elevators, bathrooms, and different maintenance rooms.



Large Network Topology

3.2 Wireless Conditions in the Office

There were multiple test setups located on the floor, with hundreds of devices using BLE protocol. In the office, there were also four different Wi-Fi networks accessible on 2.4 GHz. The following chart was taken as a snapshot of a normal workday Wi-Fi scan. This is considered the normal Wi-Fi background traffic.



Running the tests on the evenings and weekends eliminated a large part of this noise, while maintaining real-life scenarios.

3.3 Multi-Hop Test Network

For the Remote Provisioning performance and multi-hop latency, tests were carried out in an RF-shielded multi-hop test network. Eight RF isolation boxes were chained together via SMA and attenuation barrels, and each of these boxes contained at least one BRD4187C (EFR32xG24) radio board, which were used for the BT Mesh test cases. This setup ensured that no RF interference affected the testing, minimizing variability, while also creating six hops in the network.

4 Testing and Results

4.1 Latency

The latency was tested both in the open-air and RF-shielded environment.

4.2 Bluetooth Mesh Unicast Latency

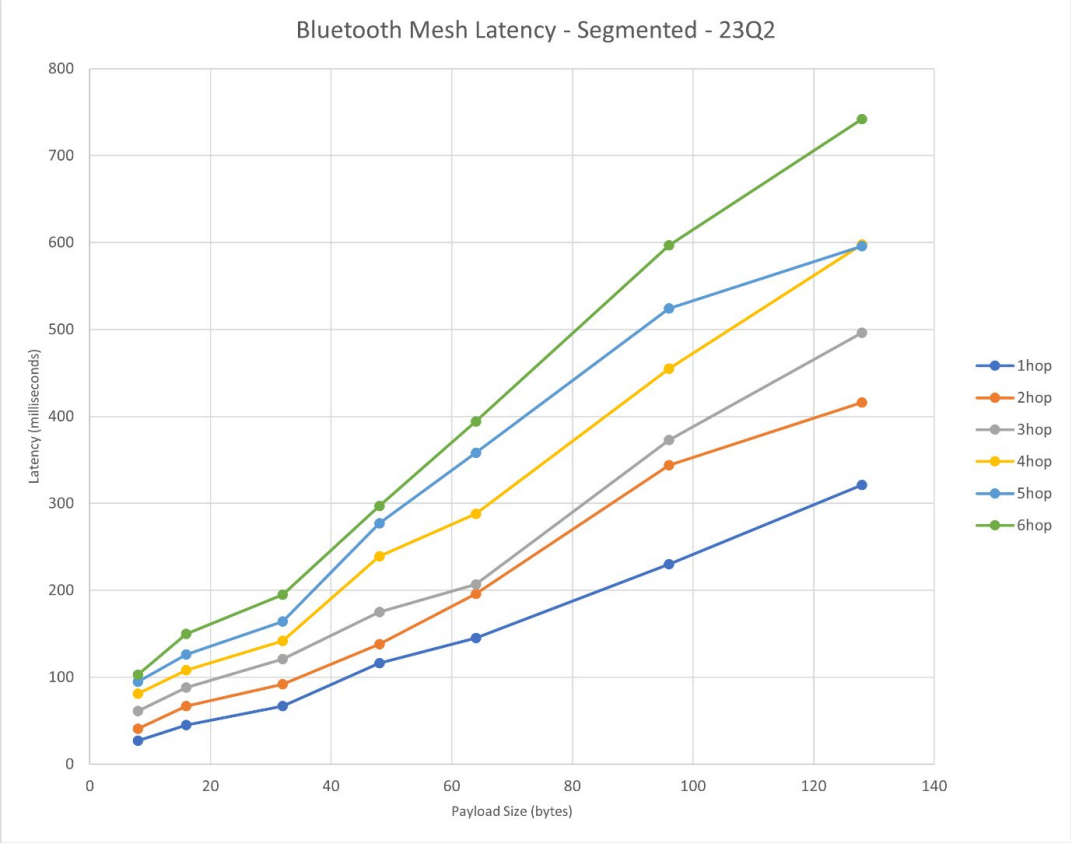
Silicon Labs tested sending unicast one-to-one payload messages at the defined rate and measuring the packet round trip time, by having the client model acknowledge the packets sent by the server mode in one of two ways:

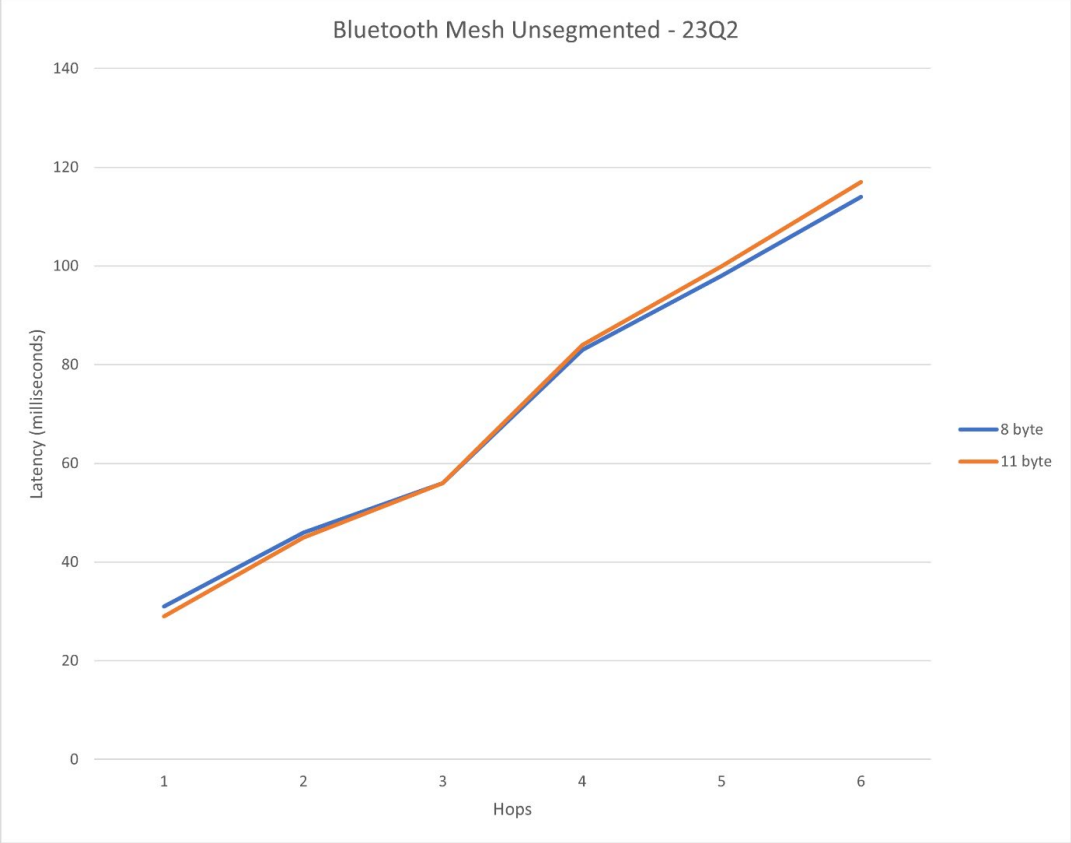
- Lower transport layer acknowledgements of segmented PDUs
- Client model layer acknowledgements of unsegmented PDUs

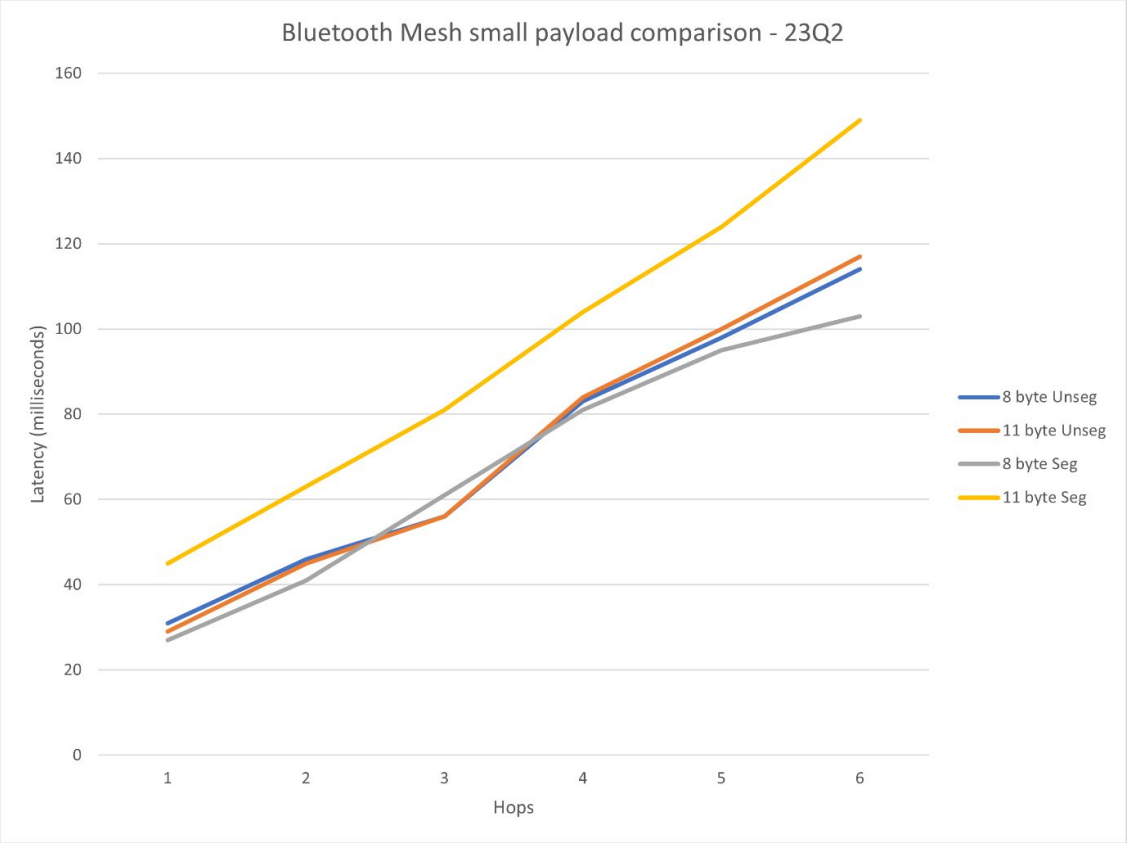
During the tests, only one relay was configured for each hop.

The following parameters were used for each test:

Parameter	Description	Used Values
Size of network	Number of nodes used for the test	2-8
Run size	Number of payload sends to attempt	1,000
PDU size	Size of the PDU in bytes	8, 11, 16, 32, 48, 64, 96, 128
Net TX	Network transmission repeat count	3
Relay TX	Relay level repetition count	3
Rate	Rate at which the messages are dispatched from the server in ms	1,500
Number of hops	The number of daisy-chained relay nodes in the network	1-7
Segmentation	8 and 11 bytes PDUs can be send out without segmentation, but segmentation forcing can be set up	8, 11 bytes: with and without segmentation 16, 32, 48, 64, 96, 128: with segmentation


Bluetooth Mesh Latency - Segmented - 23Q2
Latency (milliseconds)
Payload Size (bytes)
0
100
200
300
400
500
600
700
800
0
20
40
60
80
100
120
140
1hop
2hop
3hop
4hop
5hop
6hop


Bluetooth Mesh Unsegmented - 23Q2
Latency (milliseconds)
Hops
0
20
40
60
80
100
120
140
1
2
3
4
5
6
8 byte
11 byte



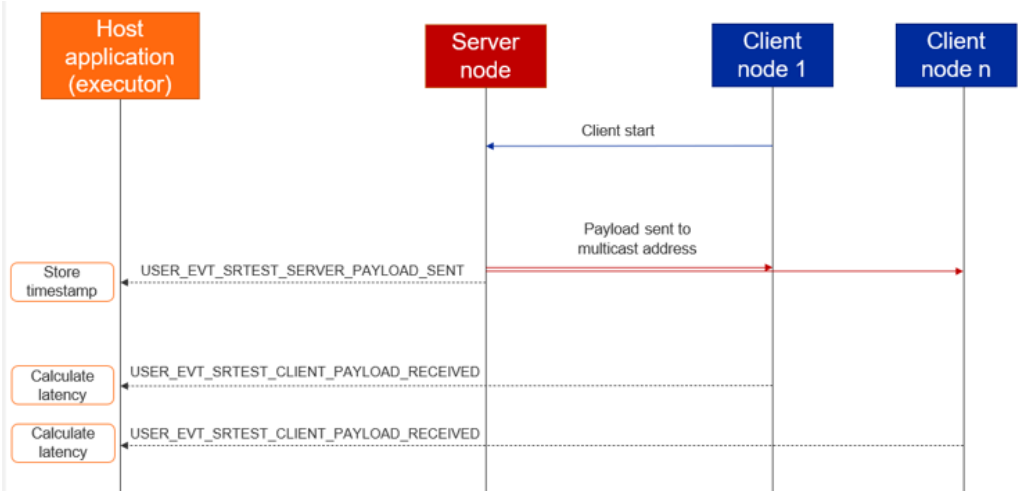
4.2.1 Bluetooth Mesh Multicast Latency

For this test, there was one server node and multiple client nodes subscribed to a network address. The server sent the packets to this address, and the time between sending and receiving on a given client was measured for each client separately.

The server node was set to be a device on box #6 in the lower right corner of the office, in order to ensure that at least one naturally occurring hop was in the network.

The test steps:

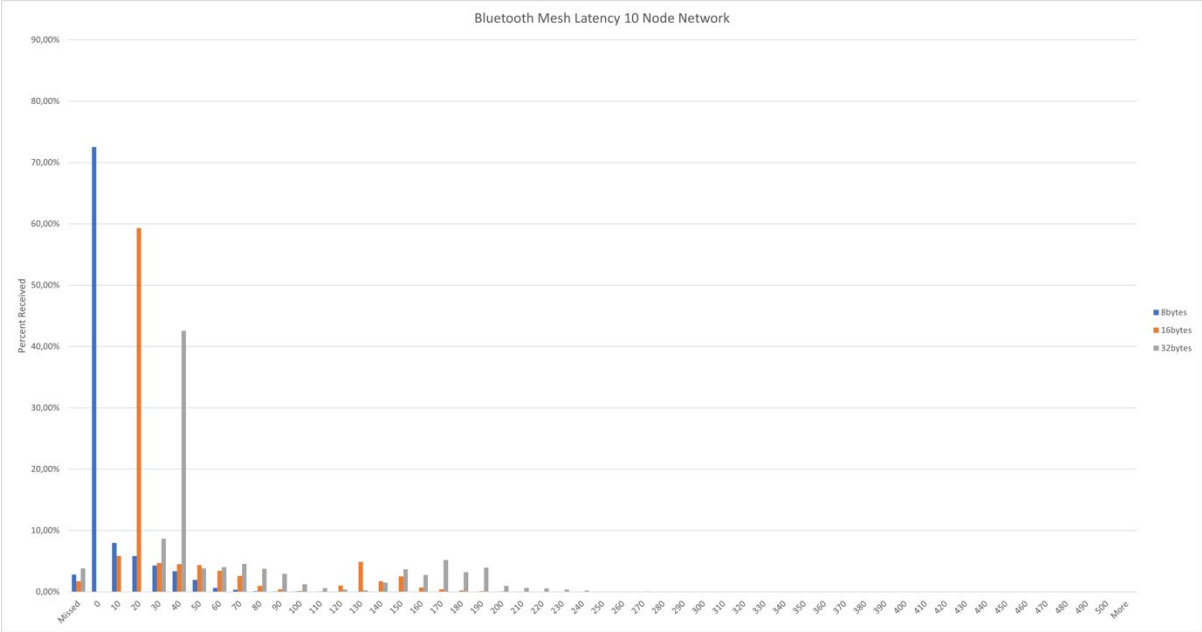
3. Configure the server and the clients.
4. One client triggers the payload sending on the server with the desired configuration using a Test Config message.
5. Server sends payloads and generate BGAPI events, which are recorded.
6. Clients receive payloads and generate BGAPI events, which are recorded.
7. Process the recorded event to calculate latency.

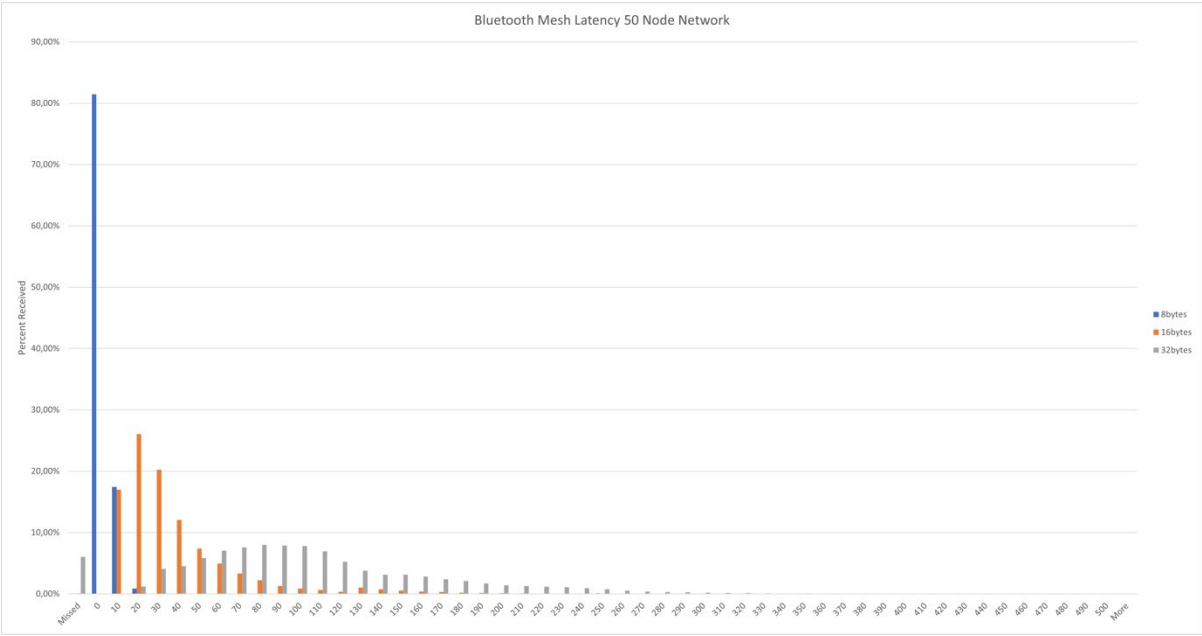


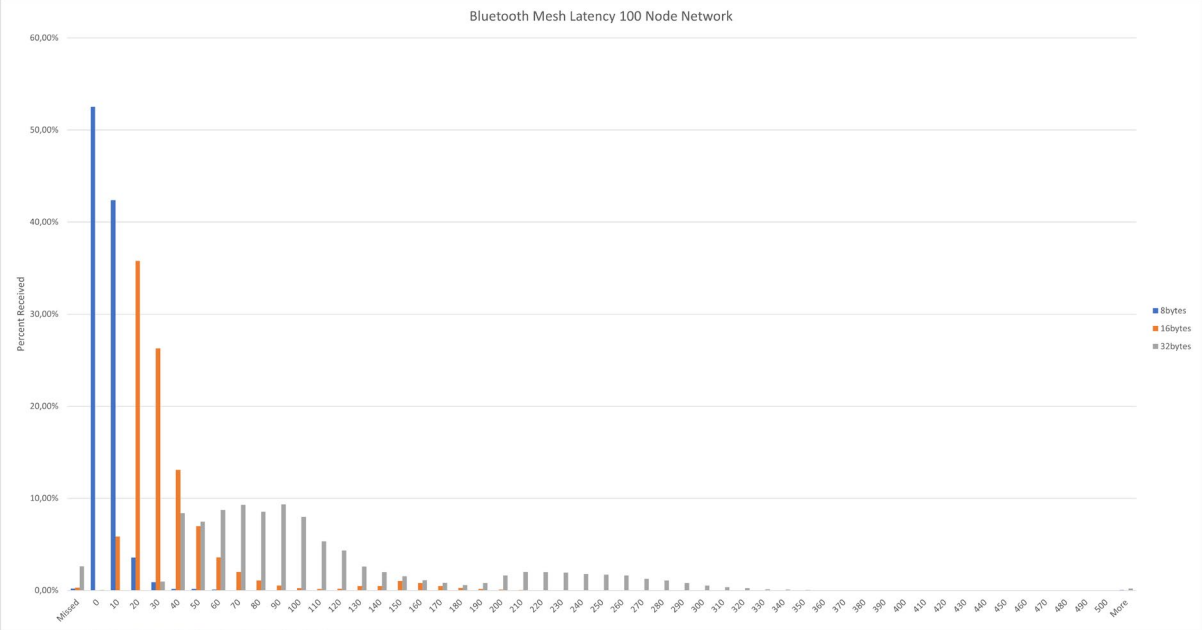
The parameters used for the test:

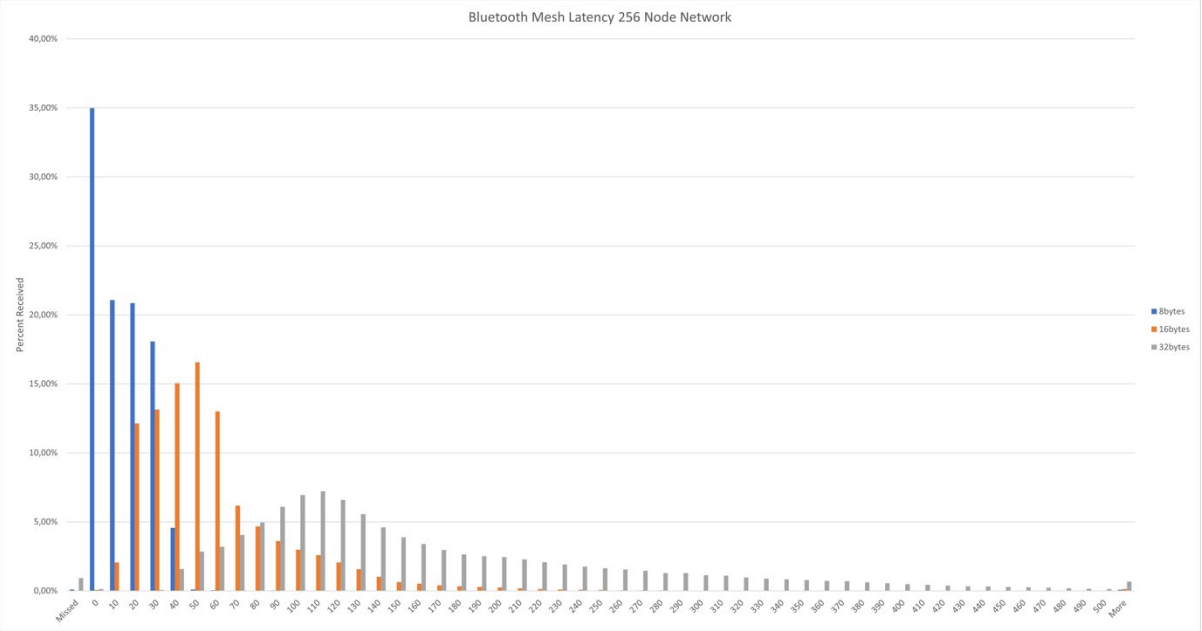
Parameter	Description	Used Values
Size of network	Number of nodes used for the test	10, 50, 100, 256
Run size	Number of payload sends to attempt	1,000
PDU size	Size of the PDU in bytes	8, 16, 32
Net TX	Network transmission repeat count	3
Relay TX	Relay level repetition count	3
Rate	Rate at which the messages are dispatched from the server in ms	3,000
Relay frequency	The frequency at which the nodes are set to behave as a relay (5 meaning that every 3rd node in the network is set to be a relay)	1, 6

Results based on node network size

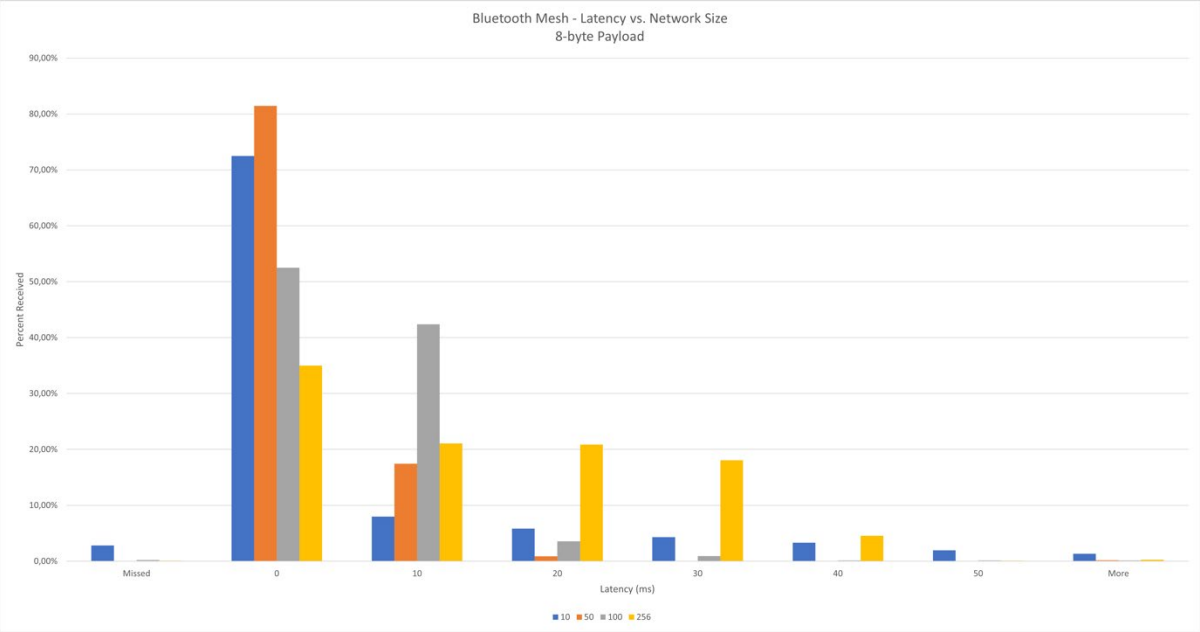


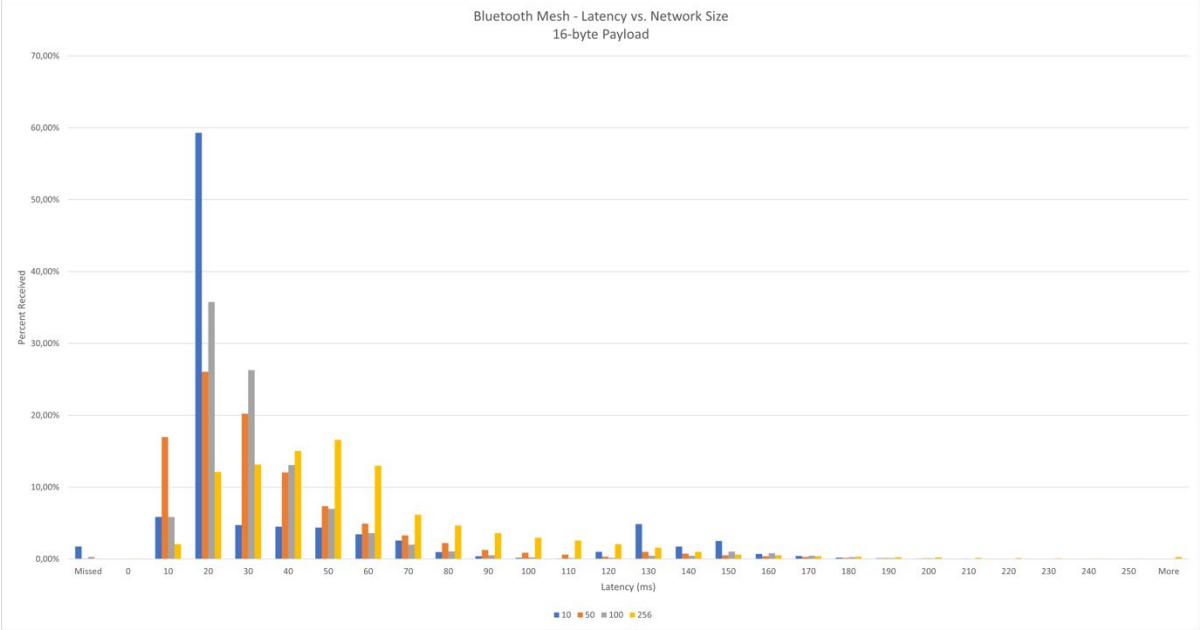


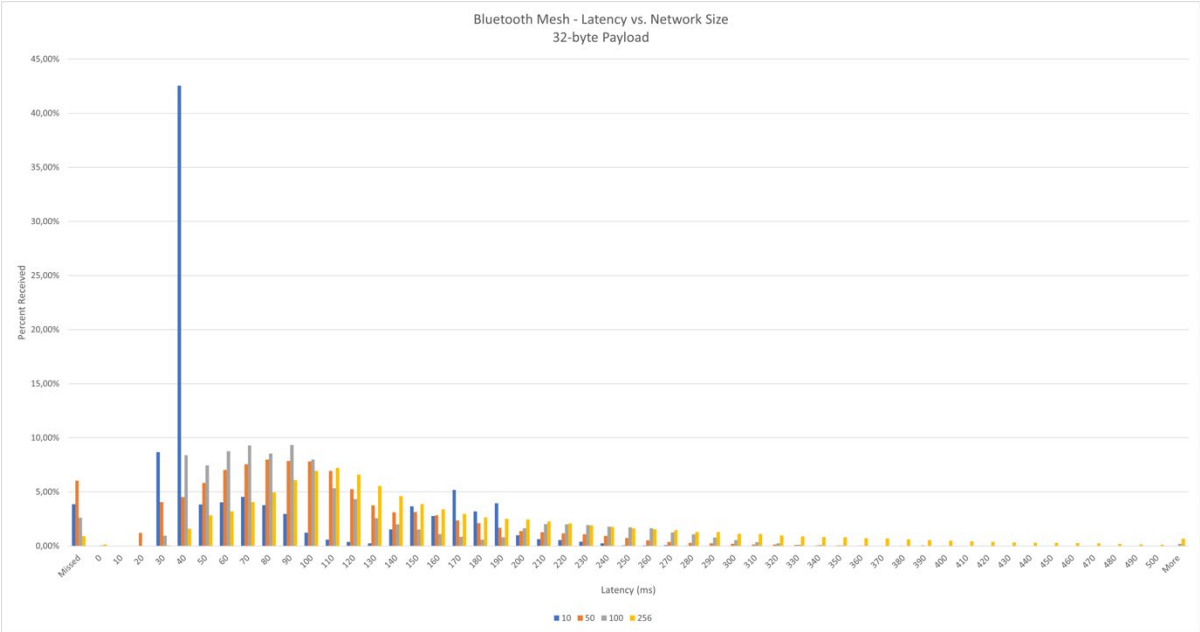

Bluetooth Mesh Latency 100 Node Network
Percent Received
8bytes
16bytes
32bytes


Bluetooth Mesh Latency 256 Node Network
Percent Received
8bytes
16bytes
32bytes

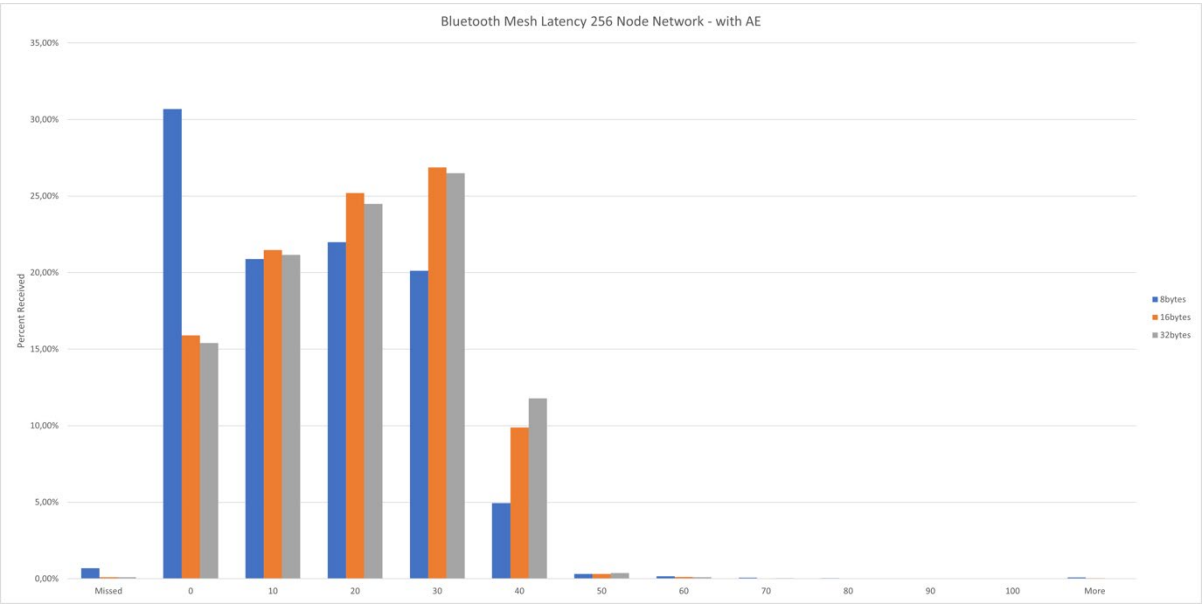
Results based on PDU size







Results with Advertising Extension (AE)



4.3 Remote Provisioning Performance

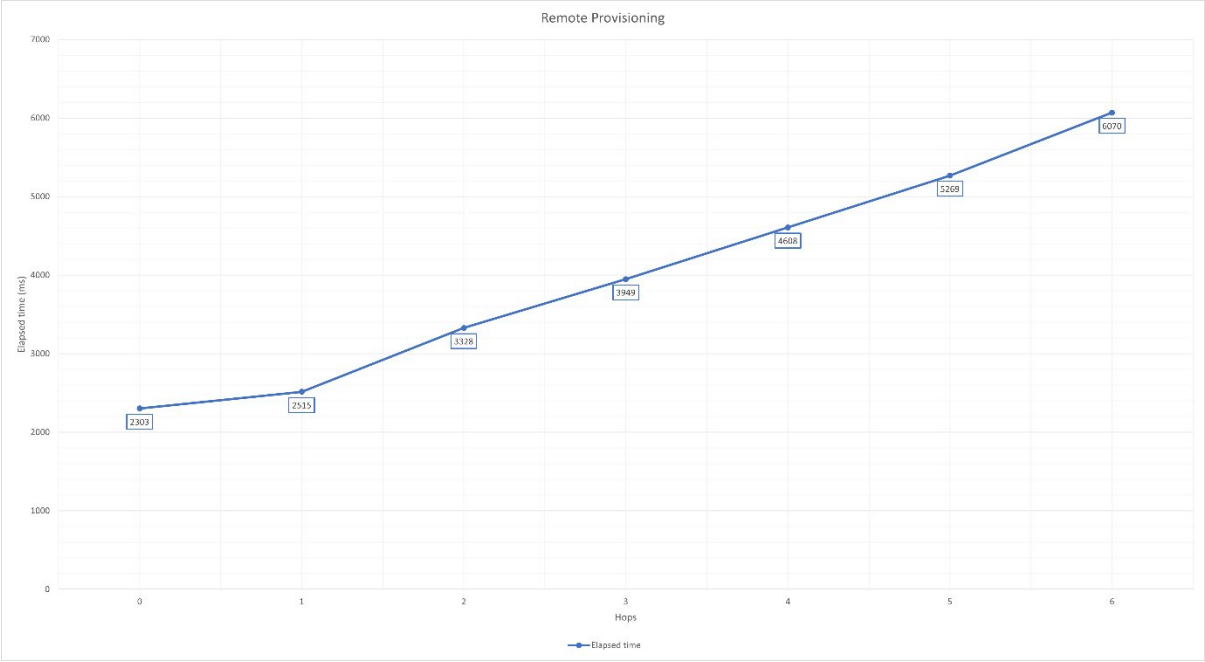
To test the time of the remote provisioning based on the number of hops between the Remote Provisioning Server and Remote Provisioning Client, the test is carried out in the multi-hop RF-shielded test setup. The layout of the devices in the RF-shielded boxes can be seen on the next table. Note that each box can receive messages only from the neighboring boxes.

Box	1	2	3	4	5	6	7	8
Devices in box	Remote Provisioning Server (**S**) Remote Provisioning Client (**C**)	Device0 (**D_0**)	Device0 (**D_1**)	Device0 (**D_2**)	Device0 (**D_3**)	Device0 (**D_4**)	Device0 (**D_5**)	Device0 (**D_6**)

Test Steps

1. The Remote Provisioning Server node (server = **S**) was provisioned traditionally by the Provisioner (Remote Provisioning Client = client = **C**). Relay mode was enabled on **S**.
2. The first device (device0 = **D_0**) was provisioned remotely, this was the 0 hops case. Elapsed time was measured.
3. Relay mode was enabled on **D_0**.
4. Remote Provisioning Server was initialized on **D_0**.
5. Iterated through the remaining devices, where i started at 1:
6. **D_i** started unprovisioned beaconing.
7. **C** opened link between **D_i** and **D_{i-1}** using the Remote Provisioning Bearer.
8. **D_i** was provisioned remotely. Elapsed time was measured.
9. Relay mode was enabled on **D_i**.
10. Remote Provisioning Server was initialized on **D_i**.

Only the time between starting the provisioning (issuing `btmesh prov_provision_remote_device`) and receiving the provisioned event (`btmesh_evt_node_provisioned`) was measured.



4.4 Over-The-Air Device Firmware Update (OTA DFU) Performance

4.4.1 Introduction

The Bluetooth Mesh Device Firmware Update feature is described in detail in *AN1319: Bluetooth® Mesh Device Firmware Update*. This document focuses more on the performance aspects of this feature, when running on larger, open-air networks.

Note that, while the target nodes are receiving the new image, the Mesh network remains operational, although the packet latency and loss rate might increase due to the increased amount of network traffic. It is also important to note that, if some target nodes fail to receive the new image, or the whole distribution fails for some reason, then the target nodes remain operational (running their old software), and the device firmware update procedure can be repeated.

It is possible to do multiple rounds of image distribution with different sets of target nodes, or with different distributor nodes, so it is not mandatory to update the whole network at once.

To test the Mesh OTA DFU feature, a software setup is used that would be easy to reproduce. That means such examples are used that are present in the release GSDK package, so no custom embedded application is needed to reproduce the results.

The image distribution is done using push transfer mode.

Distribution time is also measured with the Advertisement Extension (AE) feature enabled, which achieves better performance by increasing the network packet size. This is described in detail in *AN1405: Bluetooth Mesh on Advertising Extensions*.

4.4.2 Initiator Configuration

As an initiator, the BT-Mesh Host DFU example application was used, running on a PC (tested on Windows and Linux as well), with a Bluetooth Mesh - NCP Empty v1.1 application on the embedded side, running on a BRD4186C board. The example application had the following changes in it:

New component added:

- btmesh_stack_advertiser_extended

Parameters changed:

- NVM3_DEFAULT_CACHE_SIZE:2200
- SL_BT_CONFIG_BUFFER_SIZE:40000
- NVM3_DEFAULT_NVM_SIZE:0x18000
- SL_HEAP_SIZE:30000
- SL_STACK_SIZE:11264
- SL_BTMESH_CONFIG_MAX_PROVISIONED_DEVICES:155
- SL_BTMESH_CONFIG_RPL_SIZE:155

4.4.3 Distributor Configuration

As a distributor, the Bluetooth Mesh - SoC DFU Distributor example application was also running on a BRD4186C. It had the following changes:

New components added:

- btmesh_event_log
- btmesh_ae_server
- btmesh_firmware_update_server

Parameters changed:

- SL_BTMESH_CONFIG_MAX_RECV_SEGS:40
- SL_BTMESH_CONFIG_MAX_SEND_SEGS:20
- SL_BTMESH_FW_DIST_SERVER_MULTICAST_THRESHOLD_DEFAULT_CFG_VAL:1
- SL_BTMESH_BLOB_TRANSFER_CLIENT_LOGGING_CFG_VAL:0
- SL_BTMESH_BLOB_TRANSFER_CLIENT_LOG_BLOB_STATUS_MSG_CFG_VAL:0

- SL_BTMESH_BLOB_TRANSFER_SERVER_DIST_TARGET_MAX_CHUNKS_PER_BLOCK_CFG_VAL:0x28
- SL_BT_CONFIG_BUFFER_SIZE:60000
- SL_HEAP_SIZE:50000
- SL_STACK_SIZE:16384
- SL_BTMESH_CONFIG_MAX_PROV_BEARERS:3
- SL_BTMESH_FW_DIST_SERVER_MAX_NODE_LIST_SIZE_CFG_VAL:155
- SL_BTMESH_BLOB_TRANSFER_CLIENT_MAX_SERVERS_CFG_VAL:155
- SL_BTMESH_BLOB_TRANSFER_CLIENT_RETRY_TIME_MS_DEFAULT_CFG_VAL:1000
- SL_BTMESH_CONFIG_RPL_SIZE:155
- SL_BTMESH_BLOB_TRANSFER_CLIENT_MAX_BLOCKS_CFG_VAL:1024

The SAR Transmitter state (SAR Configuration Server model) was also modified:

- SAR Multicast Retransmissions Count state = 2
- SAR Multicast Retransmissions Interval Step = 25 ms

4.4.4 Target Nodes Configuration

As target nodes, the Bluetooth Mesh - SoC Sensor Server was used both on BRD4186C and BRD4181B boards.

New components added:

- btmesh_event_log
- btmesh_ae_server

Parameters changed:

- SL_BTMESH_BLOB_TRANSFER_SERVER_DFU_TARGET_MAX_BLOCK_SIZE_LOG_CFG_VAL:0xb
- SL_BTMESH_BLOB_TRANSFER_SERVER_DFU_TARGET_MAX_CHUNKS_PER_BLOCK_CFG_VAL:0x28

4.4.5 Test Setup

The distributor and the initiator node were chosen to be at the center of the network as much as possible, and close to each other. They were chosen to be in cluster number 8 (see section 3.1 Office and Test Network Conditions).

The tests were run with multiple network sizes and with or without the increased network PDU size (advertisement extension).

Two GBL files were used as update targets. One was a so called, dummy image that was smaller, and the nodes failed to verify it so would not apply it, and the other one was a real SOC Sensor Server GBL file that was much larger, and the nodes successfully applied it after the distribution. In the case of the dummy image, the test expected outcome was that the nodes would report a Verification Failed state at the end.

4.5 Summary of Important DFU Parameters

4.5.1 Block Size and Block Count

The Block size is negotiated during the distribution, and in this case is affected by the Max Block Size Log configuration value on the target nodes. Note that increasing the maximum block size will result in increased heap usage, which could be especially critical on the target nodes, because those are usually more limited in resources such as RAM, and the whole block needs to be stored there during distribution. The higher block size will increase the chunk size as well when the maximum chunk count cannot be satisfied with the preferred chunk size.

Larger chunks have a higher probability of being lost because they contain more segments. The penalty of lost chunks is higher because more segments need to be retransmitted.

The configuration parameter (on the target nodes):

- SL_BTMESH_BLOB_TRANSFER_SERVER_DFU_TARGET_MAX_BLOCK_SIZE_LOG_CFG_VAL

Found in `sl_btmesh_blob_transfer_server_config.h`.

The Block Count was calculated from the BLOB Size and Block Size by ceil(BLOBSize / BlockSize) formula.

The Block Count affected the size of BLOB Transfer Status message and the number of Block Start & Query phases during BLOB Transfer.

N block results to ceil((23 + 8 x ceil(N / 8)) / 12) BLOB Transfer Status segments.

4.5.2 Firmware Distribution Server BLOB Multicast Threshold

If the number of servers for any step exceeded or was equal to this number, then the group address was used; otherwise, servers were looped through one by one. Value of 0 disabled the feature.

Configuration parameter (on distributor):

- SL_BTMESH_FW_DIST_SERVER_MULTICAST_THRESHOLD_DEFAULT_CFG_VAL

Found in `sl_btmesh_fw_distribution_server_config.h`.

4.5.3 Distributor Segmentation and Reassembly Multicast Retransmission Count / Interval

These parameters determined how many times and what interval each segment of a segmented multicast message was repeated on the Distributor.

The fields that were modified were:

- SAR Multicast Retransmissions Count State
- SAR Multicast Retransmissions Interval Step

These parameters were configured by setting the SAR Transmitter state at the SAR Configuration Server model on the distributor.

4.5.4 Distributor Max Send Segs and Max Recv Segs

Maximum number of simultaneous BT-Mesh segmented message transmissions and receptions.

Configuration parameters (on distributor):

- SL_BTMESH_CONFIG_MAX_SEND_SEGS,
- SL_BTMESH_CONFIG_MAX_RECV_SEGS

These parameters are found in `sl_btmesh_config.h`.

4.6 Test Procedure Steps

The tests used the BT-Mesh Host DFU (`btmesh_host_dfu`) example script. For details about the following commands and their arguments, check the script help.

When the script runs the first time, it creates a default configuration file `btmesh_host_dfu_cfg.ini`. In this file, the following options are changed to accommodate the large network setup:

- [common]
- retry_max_default = 15
- retry_interval_default = 4.0
- retry_cmd_max_default = 15
- retry_cmd_interval_default = 2.0
- retry_multicast_threshold_default = 20
- retry_auto_unicast_default = True
- conf_retry_max_default = 10
- silabs_retry_max_default = 15
- silabs_retry_interval_default = 2.0
- auto_conf_dcd_query = false

- auto_conf_default_ttl = false
- auto_conf_network_tx = false
- auto_conf_sar = false
- [dist_clt]
- dist_retry_max_default = 100
- dist_retry_multicast_threshold_default = 100
- [network]
- random_netkey = false
- random_appkey = false

For further details, check the created INI file next to the script, where you will find these options with their default values and descriptions.

Target nodes are flashed with the modified SOC Sensor Server example application in batches of 5. Then, provisioning is done using advertisement bearer.

Groups are created from the nodes by executing the following commands:

group add -a 0 -g 0xC000 -s *Dist*[1] -m distributor -n DG1

group add -a 0 -g 0xC001 -s *Target*[0] -m target_node -n TNG1

The SAR configuration is done on the distributor node:

conf set --sar-tx-mc-retx-cnt 2 --sar-tx-mc-retx-int 25 --update *Dist*

If the advertisement extension feature is used for testing, then the following group is created, and configuration is done to set the PDU size to 227 bytes:

group add -a 0 -g 0xC002 -s *Node* -S SILABS_CONFIGURATION_SERVER -n AE1

conf ae -t 2m -p 227 --en --mdls *BLOB* -e 0 1 -g 0xFFFF -n *

The tested update target image is uploaded to the distributor:

dist upload -d *Dist*[1] -f 0x02ff:s:test_image -m s:0123 -T 1 example.gbl

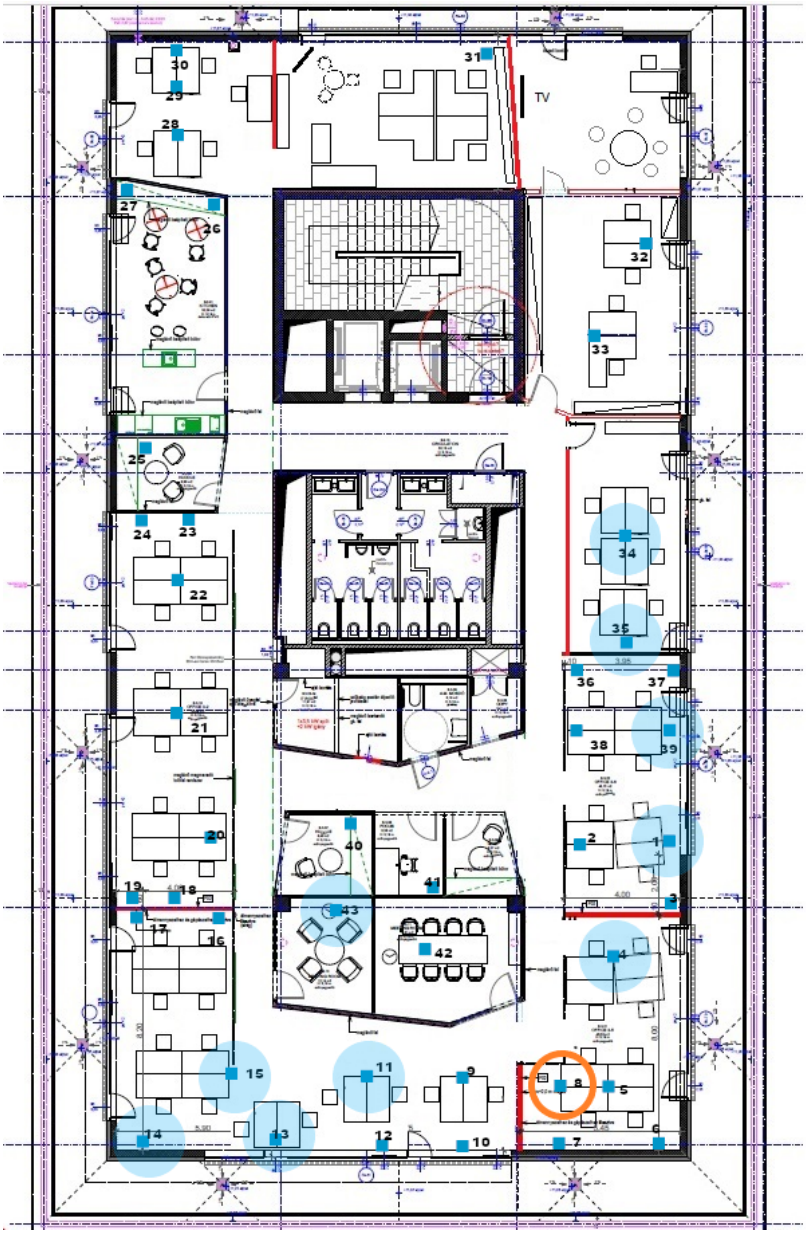
The test starts a timer to measure the distribution time, and then executes the distribution:

dist start -d *Dist*[1] -f 0x02FF:s:test_image -G TNG1 -T 10 --poll-int 20

After the above command finishes, a table is printed out on the console with the target node statuses, and in case of a dummy image, the test expects the state to be “verification failed”, otherwise it expects “apply success”. The test measures the time of distribution when this table is printed out.

4.7 Network Topology

For measuring the transfer speed, a 60-node Mesh network was created. There were no relay nodes in this setup. On the floor plans below, the orange circle signals the location of the initiator and distributor nodes. The pale blue areas are where the clusters (containing six devices each), indicated with solid blue boxes, were used to form the network.



60-Node Network Floor Plan

4.8 60-Node Network Results

Without the Advertising Extension feature enabled:

Image Type	Image Size (b)	Time (min)	Failed Node Count
Dummy	21,160	16.0	1
Real	350,076	203.2	0

With the Advertising Extension feature enabled:

Image Type	Image Size (b)	Time (min)	Failed Node Count
Dummy	21,160	12.5	0
Real	350,076	158.2	0

5 Summary

Performance testing of Bluetooth mesh shows excellent latency when the payload is contained within a single packet. Throughput results show latency can be maintained below 200 milliseconds, even out to 6 hops if the payload is less than 16 bytes.

For larger networks, as the number of nodes in the network increase or the packet payload increases, the latency also increases. The network size has a smaller effect on latency than the payload size, which can result in a large increase.

The reliability of these networks when running these results is greater than 99%.

To obtain low latency and high reliability in Bluetooth mesh applications:

- Application payload should fit into a single packet.
- Applications which require multicast messaging should not use segmented messages.

5.1 Follow-up Testing Considerations

The testing described in this application note requires follow-up tests to further define the device behavior and network operations. The following specific items are noted for follow-up testing:

1. Failure testing can also be added by dropping nodes out of this network during these tests to evaluate recovery time and impact on reliability.
2. Testing should be performed with different device types running in System-on-Chip and Network Co-Processor (NCP) modes. Previous testing has revealed some differences between these modes of operation, so this should be further characterized.

5.2 Related Literature

This application note has provided information on Bluetooth mesh networking. For information on Zigbee and Thread mesh networking, and a comparison of all three technologies, refer to the following application notes:

AN1138: Zigbee Mesh Network Performance

AN1141: Thread Mesh Network Performance

AN1142: Mesh Network Performance Comparison

Simplicity Studio

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

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

SILICON LABS

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

www.silabs.com