



AN1408: OpenThread Mesh Network Performance

This application note details methods for testing OpenThread mesh network performance. With an increasing number of mesh networks available in today's wireless market, it is important for designers to understand the different use cases among these networks and their expected performances. When selecting a network or device, designers need to know the network's performance and behavior characteristics such as battery life, network throughput and latency, and the impact of network size on scalability and reliability.

This application note demonstrates how the OpenThread mesh network differs in performance and behavior from other mesh networks. Silicon Labs conducted tests using Silicon Labs' OpenThread Mesh software stacks and the Wireless Gecko SoC platform capable of running OpenThread Mesh and Proprietary protocols. The test environment was a commercial office building with active Wi-Fi and Zigbee networks in range and with wireless test clusters deployed in hallways, meeting rooms, offices, and open areas. The methodology for performing these benchmarking tests is defined in this application note so that others may run the same tests. 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) is described.
- Wireless conditions and environments are evaluated.
- Mesh network performance including throughput, latency, and large network scalability is presented.

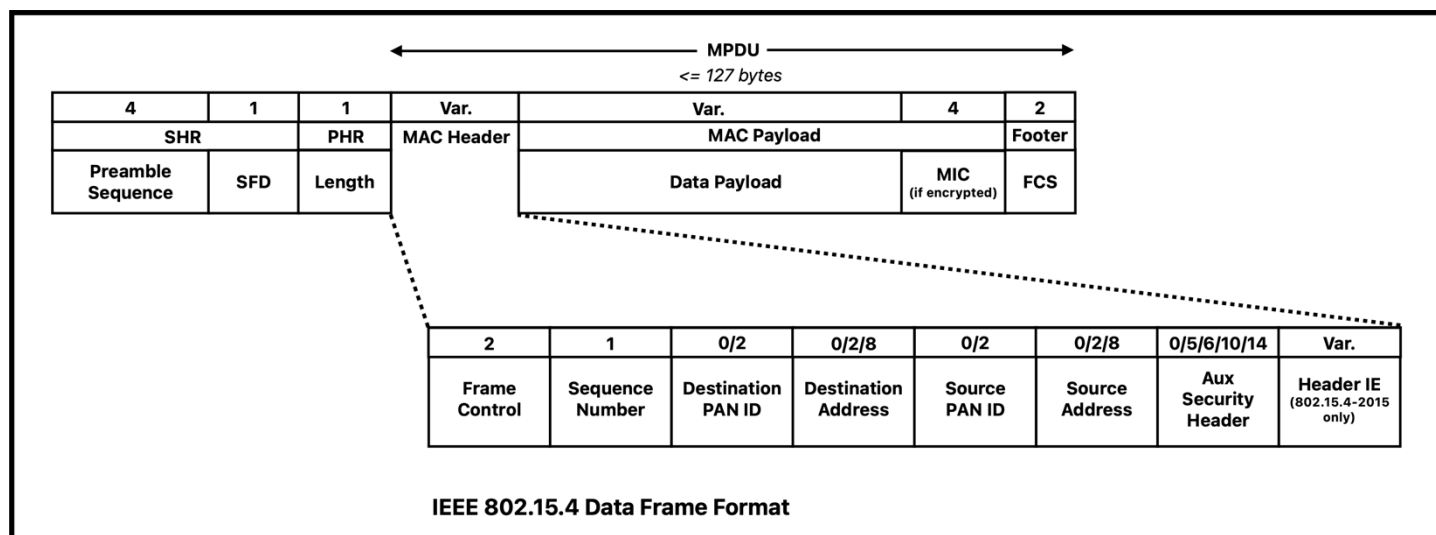
1 Introduction and Background

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. Testing is done across our different mesh network technologies – Zigbee, OpenThread, and Bluetooth and each are presented separately. This application note presents the OpenThread network performance using Silicon Labs products.

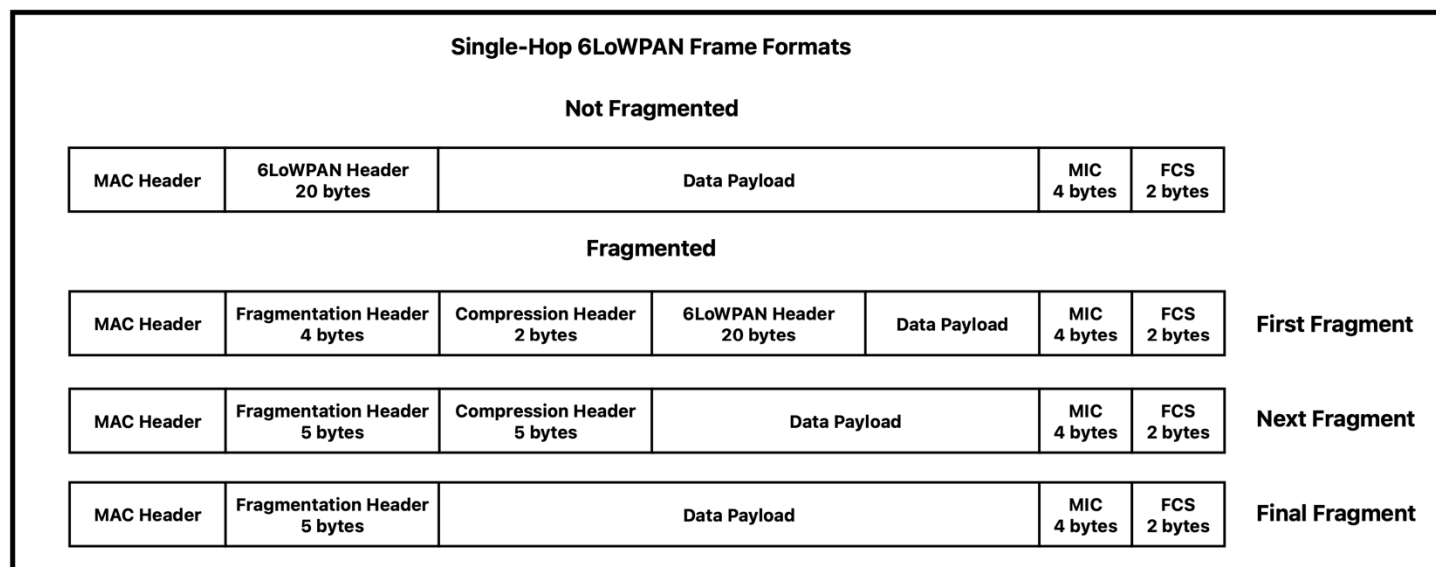
1.1 Underlying Physical Layer and Packet Structure

Because application usage does not account for the packet overhead, network performance is based on payload sizing.

OpenThread supports links based on both IEEE 802.15.4-2006 and IEEE 802.15.4-2015 with a 127-byte maximum transmission unit (MTU) for the MAC Protocol Data Unit (MPDU) and an underlying data rate of 250 kbps.



For payloads above the maximum MTU, the OpenThread stack fragments packets using 6LoWPAN, as shown below. Since the maximum MTU only applies to the MAC Protocol Data Unit (MPDU), the PHY header is not illustrated below.



Multi-Hop 6LoWPAN Frame Formats

Not Fragmented

MAC Header	Mesh Header 5 bytes	6LoWPAN Header 20 bytes	Data Payload	MIC 4 bytes	FCS 2 bytes
------------	------------------------	----------------------------	--------------	----------------	----------------

MAC Header	Mesh Header 5 bytes	Fragmentation Header 4 bytes	Compression Header 5 bytes	6LoWPAN Header 20 bytes	Data Payload	MIC 4 bytes	FCS 2 bytes
------------	------------------------	---------------------------------	-------------------------------	----------------------------	--------------	----------------	----------------

First Fragment

MAC Header	Mesh Header 5 bytes	Fragmentation Header 5 bytes	Compression Header 5 bytes	Data Payload	MIC 4 bytes	FCS 2 bytes
------------	------------------------	---------------------------------	-------------------------------	--------------	----------------	----------------

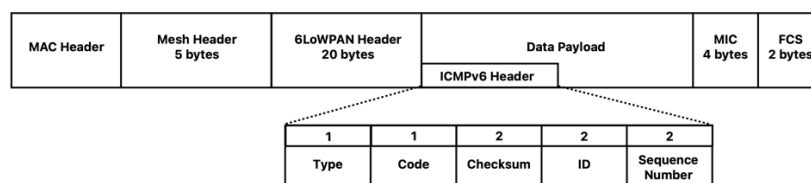
Next Fragment

MAC Header	Mesh Header 5 bytes	Fragmentation Header 5 bytes	Data Payload	MIC 4 bytes	FCS 2 bytes
------------	------------------------	---------------------------------	--------------	----------------	----------------

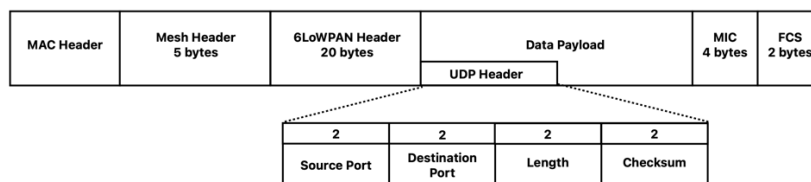
Final Fragment

The message types used in this document use either ICMP or CoAP. The header formats for these protocols, and optionally the DTLS header format (for secured CoAP), are shown below.

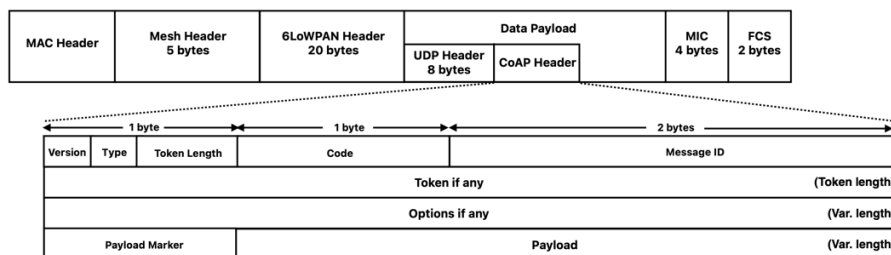
ICMPv6 Header Format



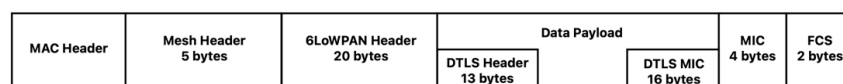
UDP Header Format



CoAP Header Format



DTLS Header Format



Data on performance is based on payload size, as this is the design parameter of concern when building an application. The following is a calculation of the maximum supported effective data payload size. Assuming ML64 (mesh-local) addressing:

```

MAC_Header size for ML64      = (Frame_Control + Seq_No + Dest_PAN + Dest_Addr + Source_Addr + Aux_Sec_Hdr)
                              = (2 + 1 + 2 + 2 + 2 + 6)
                              = 15

Maximum payload for ICMP (ping) = (127 - MAC_Header - Mesh_Header - 6LowPAN_Header - ICMP_Header - MAC_MIC - FCS)
                              = (127 - 15 - 5 - 20 - 8 - 4 - 2)
                              = 73

Maximum payload for secured CoAP = (127 - MAC_Header - Mesh_Header - 6LowPAN_Header - DTLS_Header - DTLS_MIC - MAC_MIC - FCS)
                              = (127 - 15 - 5 - 20 - 13 - 16 - 4 - 2)
                              = 52

```

1.2 Network Routing Differences

OpenThread supports both next hop routing as well as flooding. OpenThread networks maintain next hop routes to all routers as part of normal network maintenance instead of a device performing a flood-based route discovery. OpenThread also minimizes the number of active routers to address scalability to large networks. Multicast flooding previously has been seen as a limitation for embedded 802.15.4 networks because flooding in the presence of a large number of routers limited the frequency and reliability of multicast traffic. Note that the OpenThread network manages the number and spacing of active routers, so user intervention or management is not required.

Networks fragment larger messages into smaller ones to fit within their particular PHY limitations. For OpenThread, the fragmentation is done at the 6LoWPAN layer and is done source to destination (rather than at the individual hops).

For unicast forwarding within these networks, the message is forwarded as soon as the device is ready to send. For a Full Thread Device (FTD), if the Thread interface sends a frame to another Thread router for forwarding (i.e., next-hop router is not the final destination), the sending Thread interface delays any subsequent frame transmission for 8 ms. to give the forwarding router enough time to forward the frame. This is in order to avoid self-interference of packet flows. This does not apply to Minimal Thread Devices (MTDs). MTDs forward all IPv6 packets to their parent without a 6LoWPAN mesh header, and the parent performs IPv6 forwarding for packets from an MTD child. Refer to RFC 4944 for more information about unicast forwarding.

For multicast forwarding, there are generally networking requirements for how messages are forwarded. For OpenThread devices, RFC 7731 MPL forwarding is used. For this method, the trickle timer is set to 64 milliseconds, so that the devices back off a random amount up to this time before re-transmitting.

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.

2 Goals and Methods

This application note defines a set of tests performed to evaluate mesh network performance, scalability, and reliability. The test conditions and infrastructure are described, in addition to the message latency and reliability. This testing is conducted over actual wireless devices in test networks and not in simulation.

This testing is done to provide a relative comparison between the different mesh technologies to better understand and recommend their usage. Different network and system designs have different requirements for devices and networks. As such, no one network fulfills all possible network requirements. However, the three mesh networking technologies we are comparing are all aimed at low power and battery-operated mesh networks used for control and monitoring in the home and commercial buildings.

Normally, when analyzing data on network performance, we then consider what improvements can be made in the network to improve performance. Because of the limited data publicly available on mesh network performance in large networks today, it is difficult to have industry discussions on possible improvements or changes. For example, in commercial buildings there is concern over:

- Other network traffic, since there may be many subnets that interfere with each other.
- Wi-Fi interference from the normal building Wi-Fi infrastructure as these technologies are generally operating in the 2.4 GHz ISM band.
- Network throughput and latency as well as large network multicast latency and reliability due to multicasts being commonly used for lighting controls in dense office environments and the responsiveness in lighting controls that users expect.

Note: The test results here are limited to comparisons of system performance under normal operating conditions, or under stress as noted in particular tests. This application note does not specifically address system interference or other such effects that have been addressed in other published results. However, testing is done in our Silicon Labs R&D facility where there are more than 100 Wi-Fi access points within RF range. The facility also has a 300-node Zigbee lighting network that is not part of this testing but is used for normal lighting control.

2.1 Review of Other Performance Testing

There are no specific, defined methods for evaluating and reporting large network reliability, scalability, or latency. Silicon Labs has published papers comparing network performances based on network testing focused on device behavior and impact on battery life, network throughput, and latency. Large scale multicast testing also requires capturing accurate timing and reliability information from large, distributed networks. All testing was conducted using Silicon Labs' Wireless Gecko SoC platform capable of running Zigbee, OpenThread, Bluetooth Mesh, and Proprietary RF protocols to eliminate the device itself as a difference in the testing. Previously published results showed differences between transceivers, network co-processors, and System-on-Chip designs. These tests all use System-on-Chip designs.

3 Test Network and Conditions

To minimize variability, device testing can also be performed in fixed topologies where the RF paths are wired together through splitters and attenuators to ensure the topology does not change over time. MAC filtering can also be used to achieve the network topology.

A typical wired test configuration is shown in Figure 3-1 Wired RF devices in drawer with splitter and coax cable connectivity.



Figure 3-1 Wired RF devices in drawer with splitter and coax cable connectivity

The method above was previously used for 7hop test scenarios, but the current results presented in this application note now use isolation (Ramsey) boxes that are chained together with SMA cabling and proper attenuation between the boxes validated to ensure that each hop is honored and each device can only hear it's direct neighbor.



Figure 3-2 Isolated multi-hop environment

Large network testing is best conducted in an open-air environment where device behavior is based on the existing and varying RF conditions. The Silicon Labs R&D facility is used for this open-air testing.

3.1 Facility and Test Network Conditions

The Silicon Labs R&D facility consists of a central core with an elevator shaft, other services with an open floor plan on the west end of the building, and offices and conference rooms on the east end. The overall facility measures approximately 120 feet by 200 feet. The image below shows the facility layout. The darker lines represent hard walls, and the lighter lines represent cube partitions.



Figure 3-3 Silicon Labs facility layout used for wireless testing

The testing devices are installed at various locations around the facility. These devices all have Ethernet backchannel connectivity to allow:

- Firmware updates
- Command line interface
- Scripting
- Timing analysis
- Packet capture
- Energy measurements

The testing cluster in Figure 3-4 Typical Testing Cluster includes:

- Six EFR32MGxx Devices (two are MG12 board 4161a, while other four are MG24 board 4187c).
- Multi-band support to allow testing both 2.4 GHz (PCB antenna) and proprietary sub-GHz protocols (external antenna).
- USB power and Ethernet connectivity.

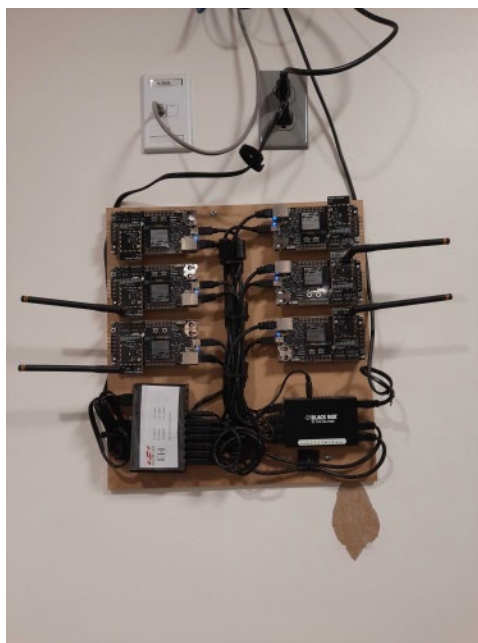


Figure 3-4 Typical Testing Cluster

The testing clusters are spread throughout the facility in both high and low locations, open areas, and enclosed meeting rooms and offices, as shown in Figure 3-5 Testing Clusters in the Silicon Labs R&D Facility.



Figure 3-5 Testing Clusters in the Silicon Labs R&D Facility

This test network has devices added or removed from it on a regular basis, but at the time of this testing it consisted of a mix of MG12 and MG24 devices.

This network represented devices that were used for open-air testing by the networking and software quality assurance teams. All devices are controlled from a central test server and infrastructure, which allows scripted regression testing or manual testing by engineers.

3.2 Wireless Conditions in the Facility

The Silicon Labs R&D facility has a full Zigbee lighting control system including motion and lighting sensors and switches. This is not part of the test network and is used as a normal building control system independent of any testing being run.

In addition to the existing Silicon Labs R&D facility Wi-Fi infrastructure, there are over 100 Wi-Fi access points within RF distance of the facility. The following charts were taken as a snapshot of a normal work day Wi-Fi scan. This is considered the normal Wi-Fi background traffic.

Note: These charts were created years ago during the creation of the original AN1141 for Silicon Labs Thread. Over the years, the interference has grown considerably with many more Wi-Fi access points, as well as more 15.4 and Wifi devices over the office.

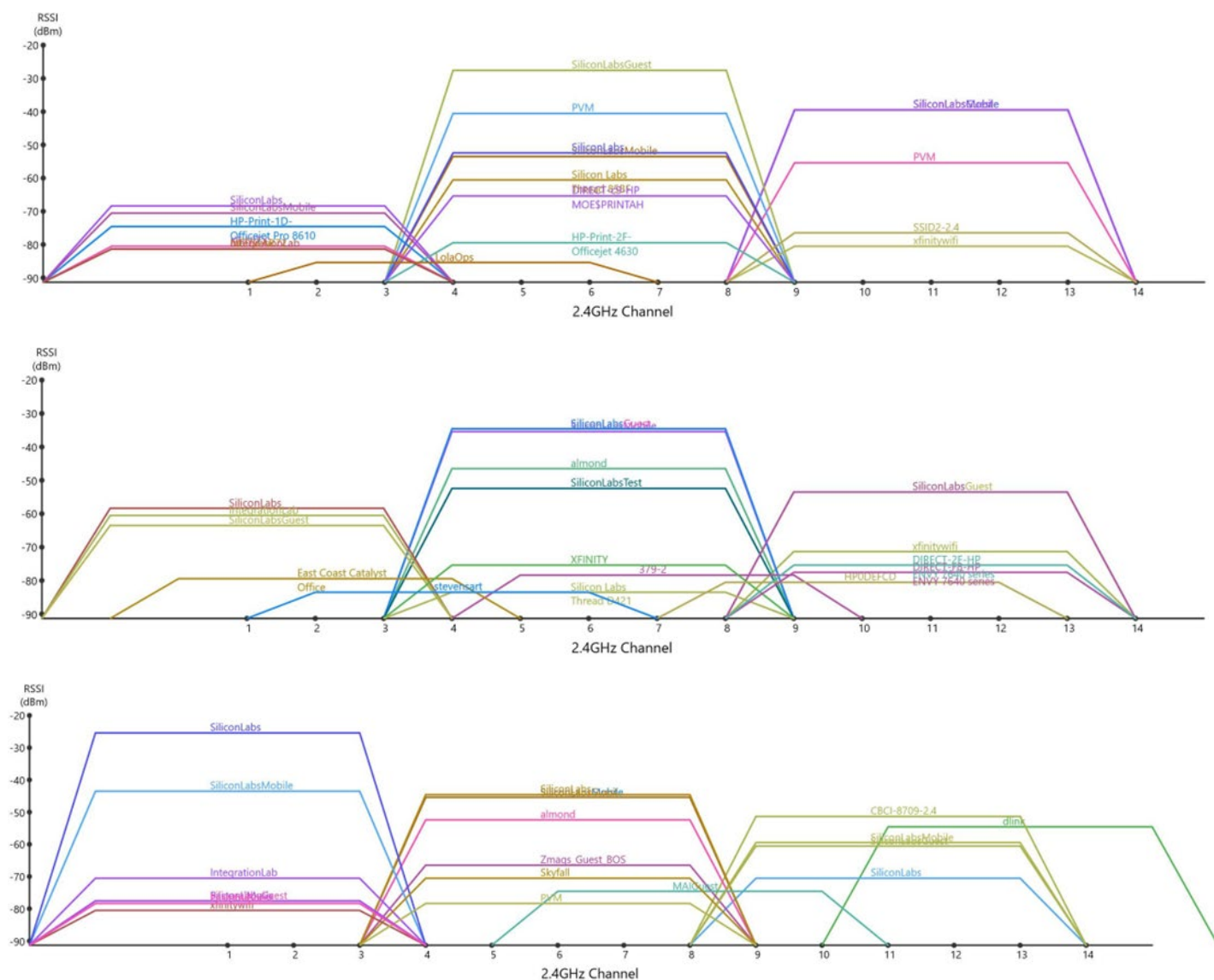


Figure 3-6 Wi-Fi Scans on a Normal Day

The Wi-Fi scans in Figure 3-6 Wi-Fi Scans on a Normal Day were taken at the southeast corner office, on the west side, and on the north side in the main conference room, respectively. These locations showed 62, 104, and 83 Wi-Fi access points within RF range.

4 Testing and Results

4.1 Throughput and Latency

This testing is conducted on a small, isolated, and controlled testbed in order to test each hop with increasing packet payloads. The normal configuration is to test for up to 7 hops. Physical isolation and SMA attenuation is used. Testing is done with one source node and a series of destination nodes to allow the number of hops to be varied, as in Figure 3-2 Isolated multi-hop environment.

The term *throughput* is used loosely here, as we are measuring time from a round trip exchange over varying size data payloads.

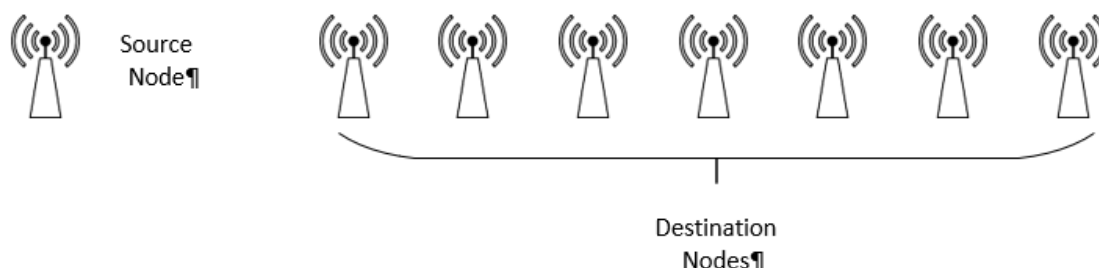


Figure 4-1 Throughput and Latency Test Configuration

This testing is done using the following configuration:

1. Multiple messaging types measured for round trip and either based on ICMP or CoAP.
2. Packet payload from 8 bytes to up to ~300 bytes in 16-byte increments for the latency test.
3. From 1 to 7 hops using the leader as the source.
4. Using 1 packet in flight.
5. Sending as fast as possible given ack timing.
6. Measuring round trip latency (source to destination to source) in milliseconds.

4.1.1 OpenThread Multi-Hop Latency

Both ICMP and CoAP test results are measured in round trip time. A significant difference between the two is that ICMP returns the payload that was sent, whereas CoAP returns a CoAP response of only about 6 bytes.

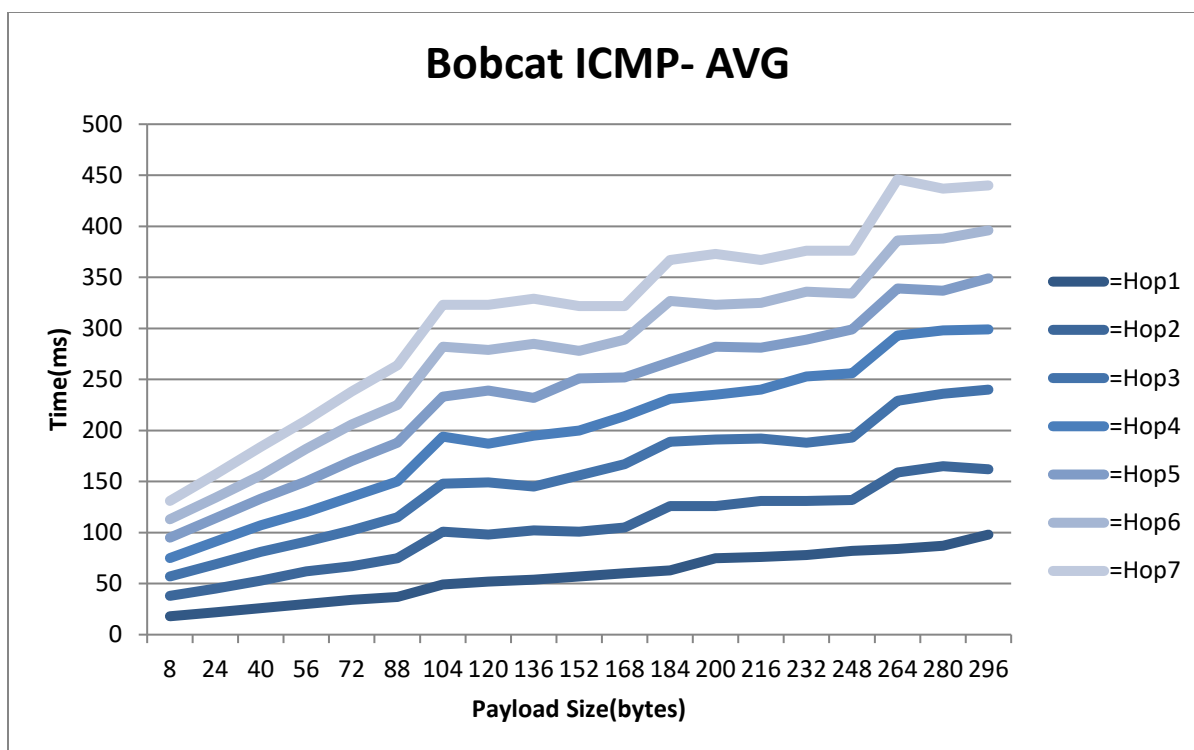


Figure 4-2 ICMP - AVG

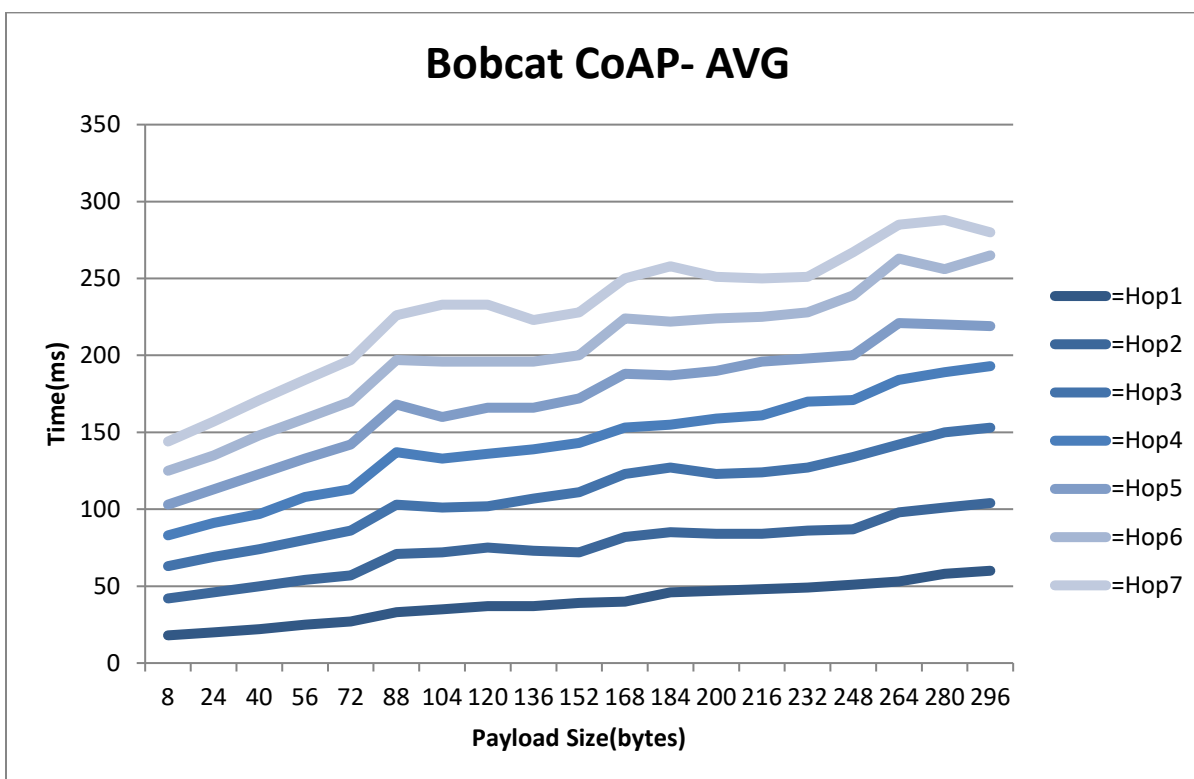


Figure 4-3 CoAP - AVG

4.2 Network Tests versus Network Size

Open air large network testing is required to validate stack performance under less controlled conditions. These networks are configured within normal Silicon Labs office space with normal Wi-Fi interference, other network operations, and building control systems. No attempt is made to isolate these network RF conditions.

For this application note, Silicon Labs has tested for network sizes ~ 25, 50, 100, 150, 200.

Note: For any of these tests, the specific number of devices is acceptable within +/- 5% of these test network targets for a given set of testing. Testing in this large network for this application note is done in SoC mode for devices.

These networks are all configured as powered devices unless there is specific testing for sleeping end.

For each of these networks the testing will validate reliability and latency for a set of traffic conditions. Testing is intended to be done over 100 messages but longer runs with 10000 messages for reliability are also done. The same devices were used across the tests to keep the topology and density of the different test runs similar. Actual over the air conditions will vary and cannot be controlled in these tests.

4.2.1 OpenThread Large Network Testing Results

The OpenThread network limits the number of routers to 32 or less, and those devices that are active routers can change over time as the network grows or conditions change. Those devices that are not active routers are referred to as Router Eligible End Devices (REEDs) and they act as Always On children. The initial device sends the broadcast, which is heard by all routers within RF range, as well as any REED devices having that initial device as a parent. REEDs are required by the OpenThread specification to synchronize with a single primary parent, but also synchronize with at least three additional parents to improve multicast reliability.

OpenThread devices use a multicast backoff of 32-64 milliseconds before a device relays the multicast.

Silicon Labs has not specifically identified the results of each established network in terms of multihop or parent-child relationships.

The graphs below represent the OpenThread network multicast behavior over network and payload size. A single device of the network transmits a multicast of varying size at a rate of 1 multicast from 1 device per 1000ms. For this testing, the transmitter is identified at the start of the test and is the only device that transmits each multicast. The x-axis shows how many results were received within that histogram bucket. $\text{Bin}(x)$ = how many metrics over all the receiver recorded discrete time values was within $\text{time}(\text{ms}) \times \text{thru } x+9$, eg, $\text{Bin}(0)$ = receiver recorded 0 - 9 ms of time to get that multicast. $\text{Bin}(10)$ = receiver recorded 10-19 ms of time. A value of -10 is meant to represent that a single receiver did not receive a packet, a value of -20 is meant to represent that the transmit failed (ie, that not one of the devices received a specific sequence number).

These histograms are presented as frequency percentage based on the y-axis for ease of comparison. A frequency of 60% indicates that of all the multicast packets received, 50% of them had values of that histogram bucket. For example, in a network of 25 devices where 100 multicast packet were transmitted from 1 device, it is expect that 2400 multicast packets are received. If the chart shows 50% received in $\text{Bin}(10)$, then 1200 discrete results have a value from 10-19 milliseconds.

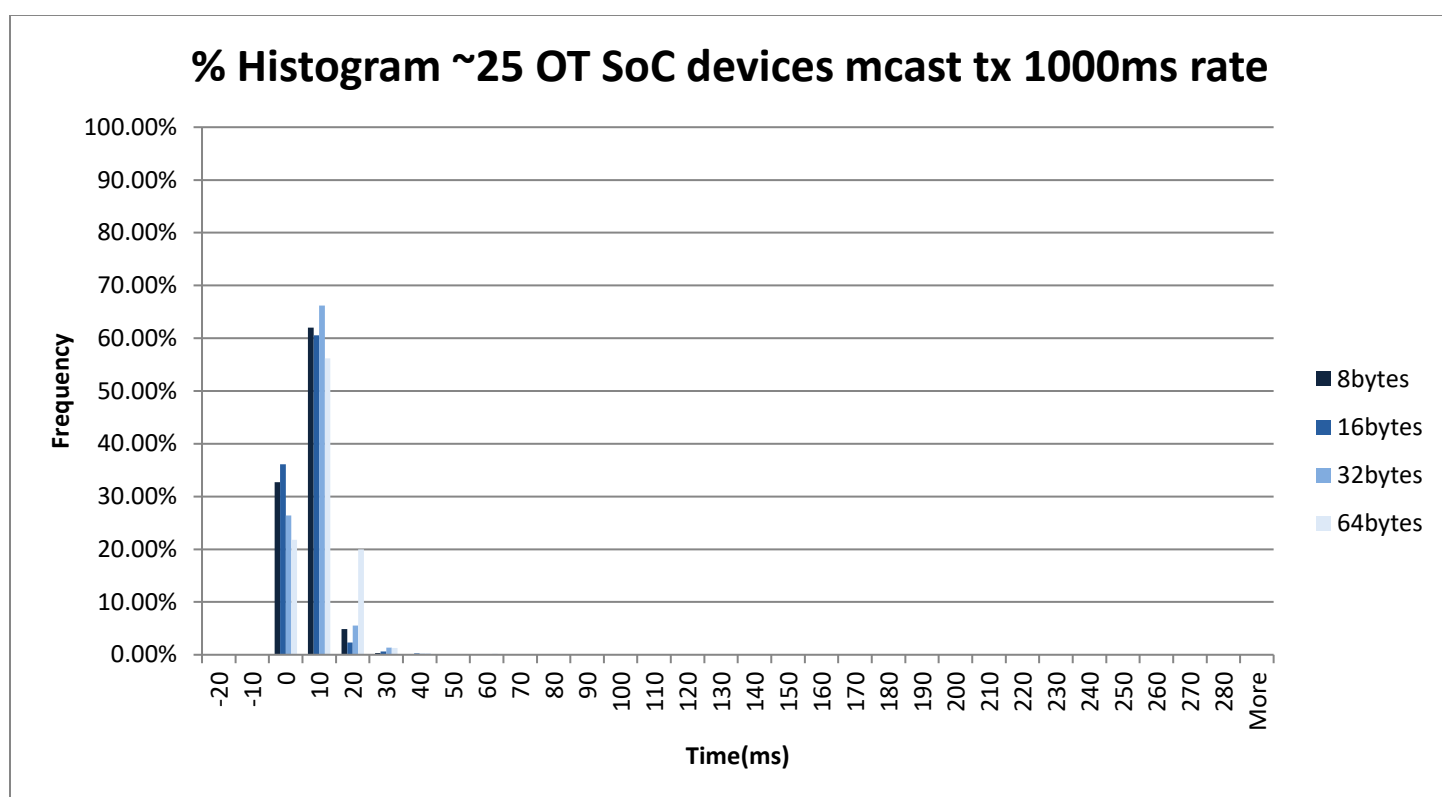


Figure 4-4 % Histogram ~25 OT SoC devices mcast tx 1000ms rate

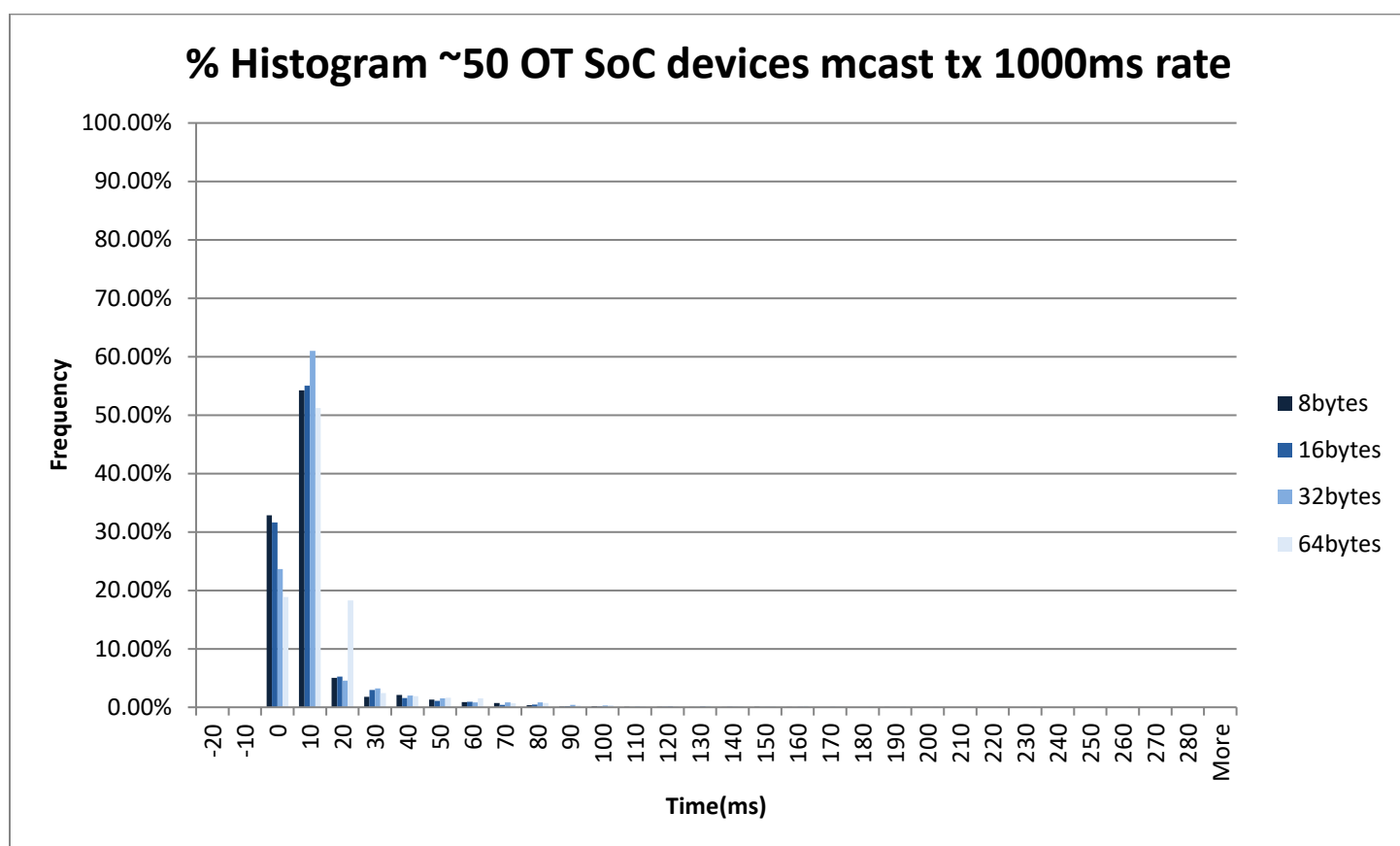


Figure 4-5 % Histogram ~50 OT SoC devices mcast tx 1000ms rate

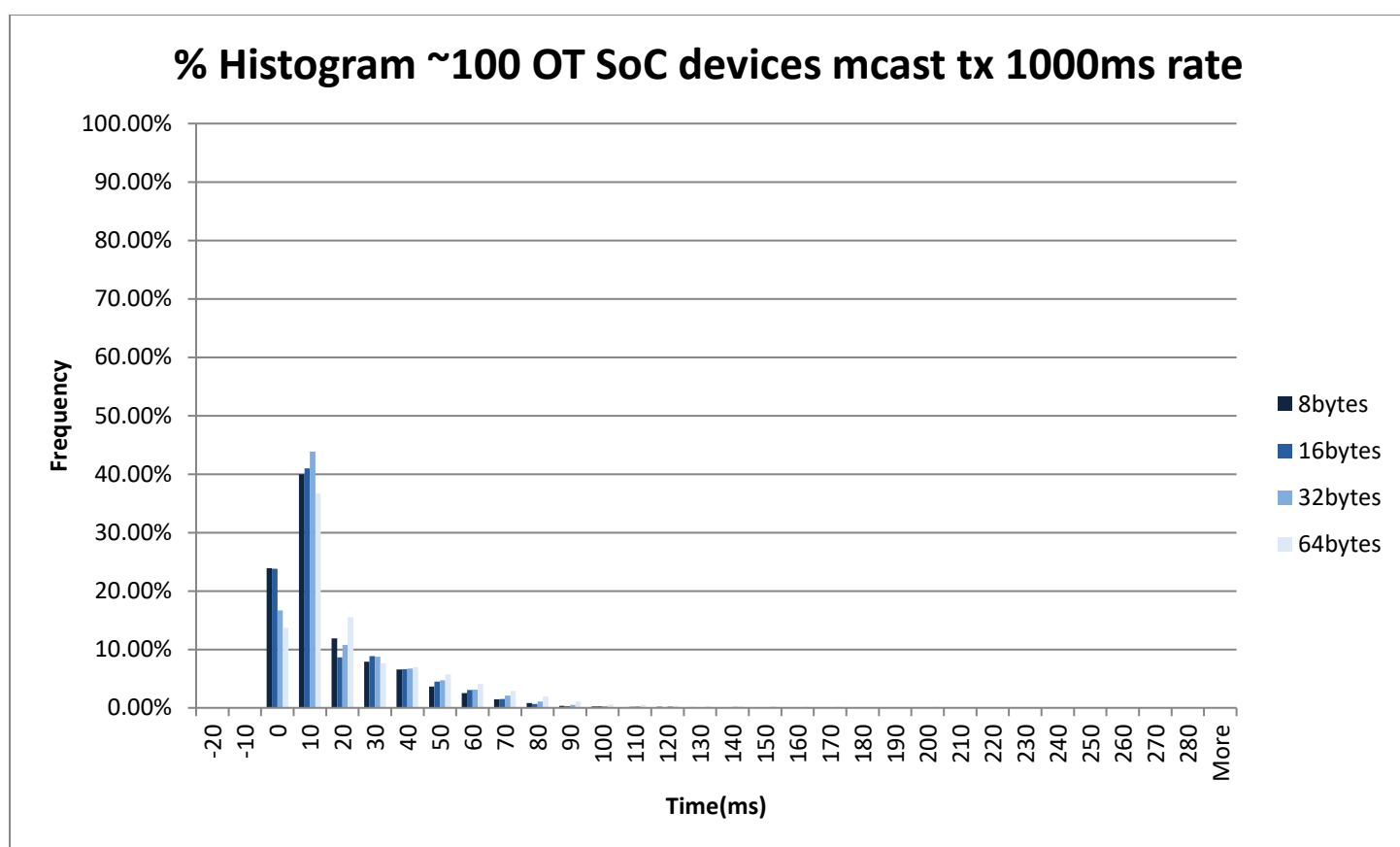


Figure 4-6 % Histogram ~100 OT SoC devices mcast tx 1000ms rate

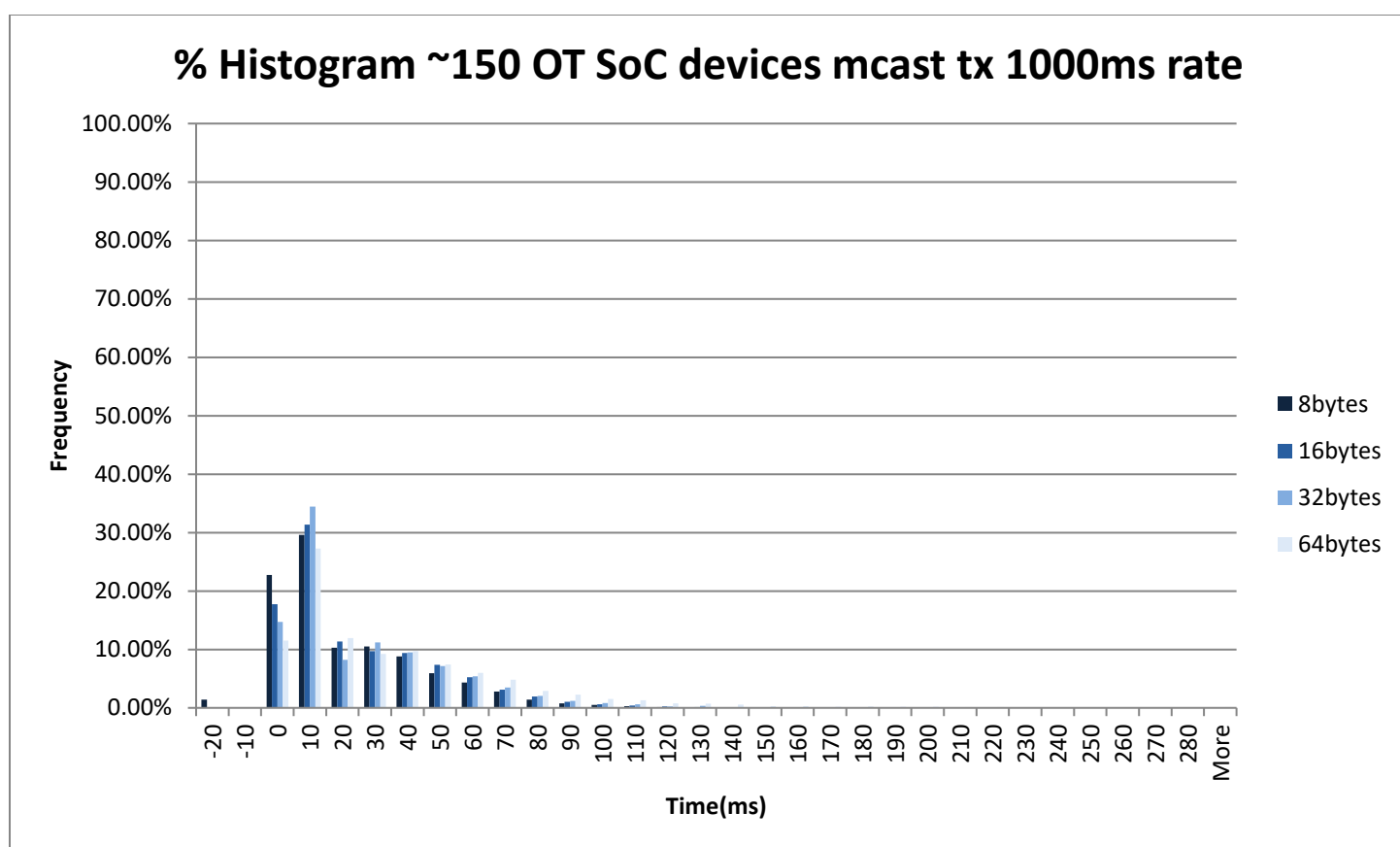


Figure 4-7 % Histogram ~150 OT SoC devices mcast tx 1000ms rate

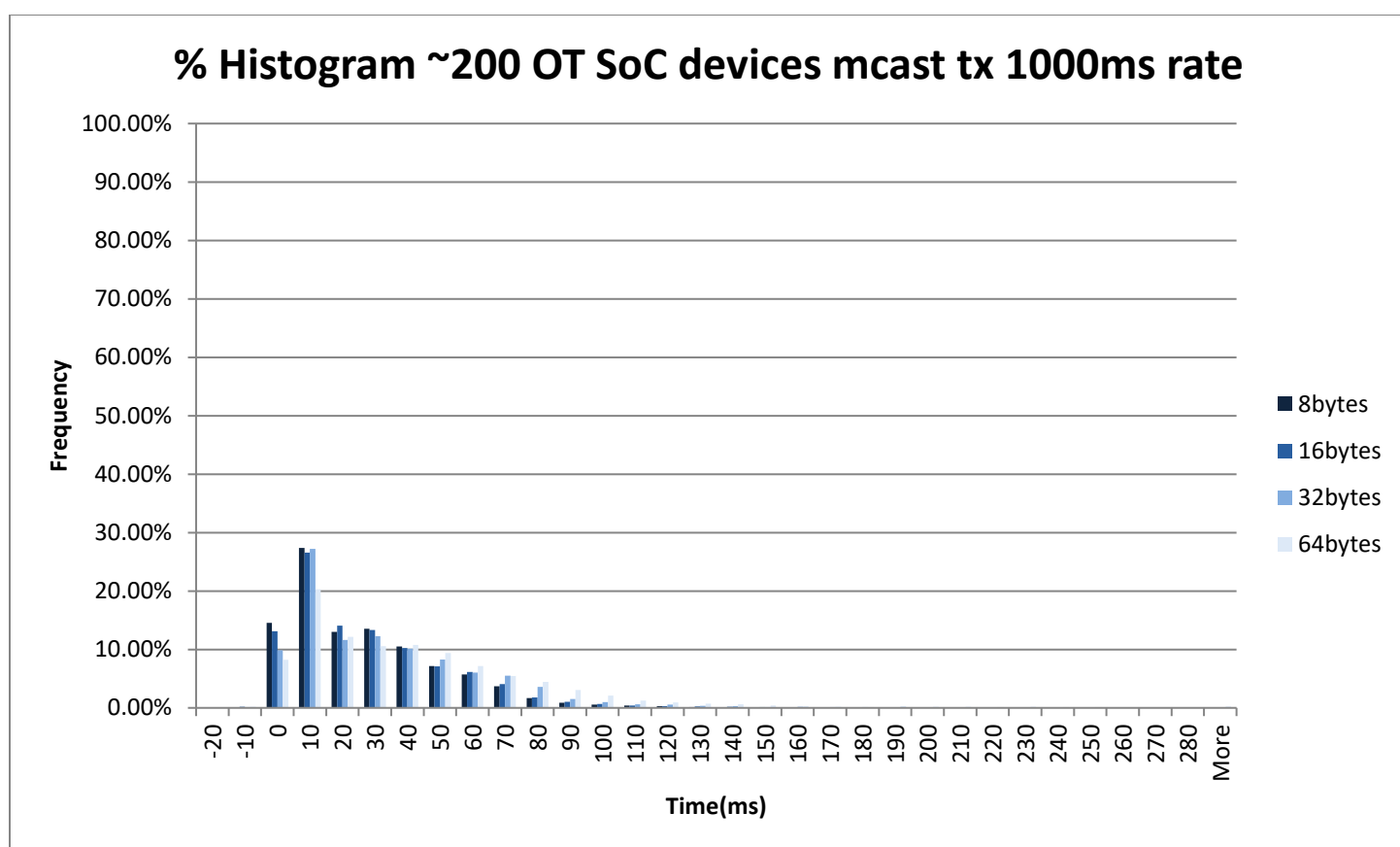


Figure 4-8 % Histogram ~200 OT SoC devices mcast tx 1000ms rate

Performance comparison over network size:

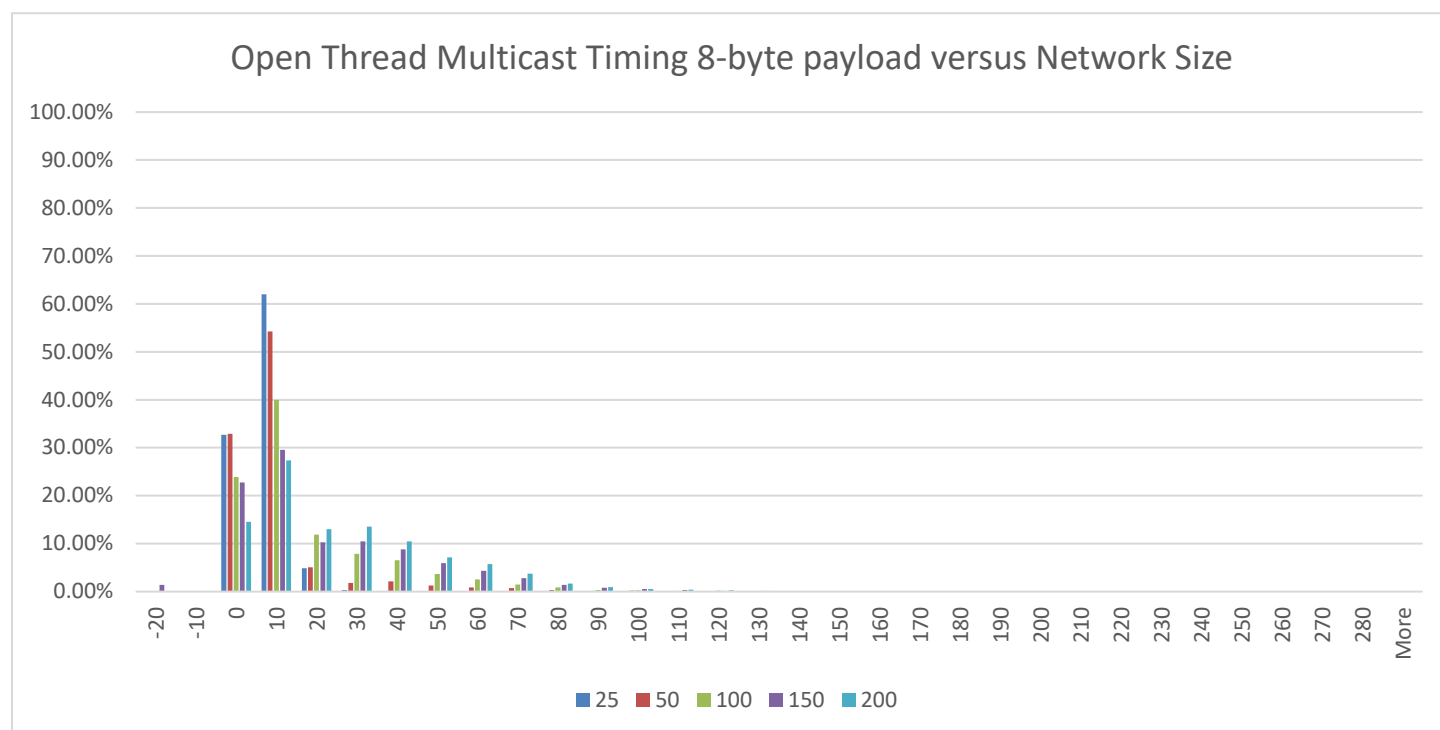


Figure 4-9 Open Thread Multicast Timing 8-byte payload versus Network Size

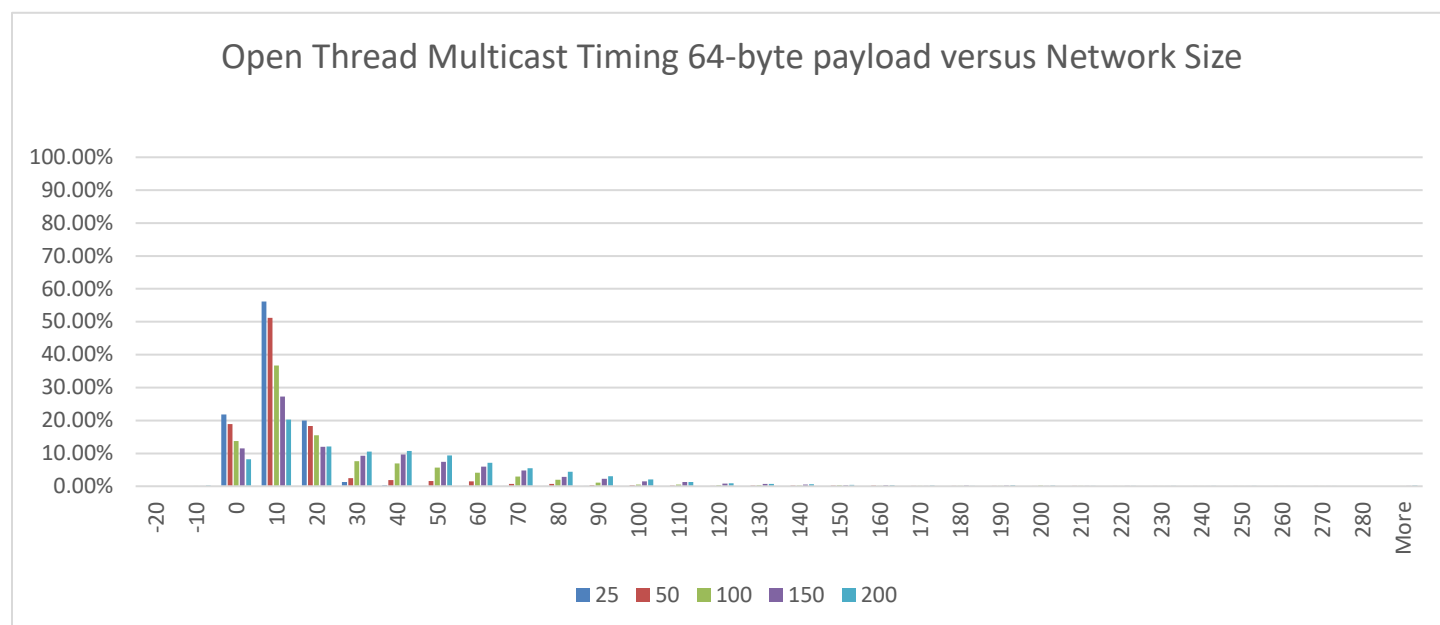


Figure 4-10 Open Thread Multicast Timing 64-byte payload versus Network Size

Of note in these results:

- OpenThread network behavior is very consistent across network sizes, and the latency increases as the packet size increases.
- OpenThread latency does not spread out as much as network size increases. This is expected as there are fewer routers, resulting in less congestion on the network.
- These tests all showed very close to 100% reliability with very few losses even for the larger networks.

5 Summary

OpenThread shows excellent reliability and latency below the 200-millisecond timing typically required for human interaction with devices. Even in multicast, large network conditions, every second the OpenThread network handles the traffic and maintains latency and reliability. OpenThread shows little change in network behavior as the network scales up in size due to the flexible router configuration done at the network layer. This is somewhat expected since the device is a newer architecture, runs at a higher clock speed, and has more RAM for packet handling. As packet payload increases, the latency across the network also increases, but this is a smaller impact as shown in Figure 4-9 Open Thread Multicast Timing 8-byte payload versus Network Size and Figure 4-10 Open Thread Multicast Timing 64-byte payload versus Network Size.

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 OpenThread mesh networking. For information on Bluetooth and Zigbee mesh networking, refer to the following application notes:

- AN1137: Bluetooth Network Performance
- AN1138: Zigbee Mesh Network Performance

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

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

Note: This content may contain offensive terminology that is now obsolete. Silicon Labs is replacing these terms with inclusive language wherever possible. For more information, visit www.silabs.com/about-us/inclusive-lexicon-project

Trademark Information

Silicon Laboratories Inc.[®], Silicon Laboratories[®], Silicon Labs[®], SiLabs[®] and the Silicon Labs logo[®], Bluegiga[®], Bluegiga Logo[®], EFM[®], EFM32[®], EFR, Ember[®], Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Redpine Signals[®], WiSeConnect, n-Link, ThreadArch[®], EZLink[®], EZRadio[®], EZRadioPRO[®], Gecko[®], Gecko OS, Gecko OS Studio, Precision32[®], Simplicity Studio[®], Telegesis, the Telegesis Logo[®], USBXpress[®], Zentri, the Zentri logo and Zentri DMS, Z-Wave[®], and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.



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

www.silabs.com