# ANT BLAZE

Libraries Specification

## Copyright Information and Usage Notice

The information contained in this document is subject to change without notice and should not be construed as a commitment by Dynastream Innovations Inc. unless such commitment is expressly given in a covering document.

The Dynastream Innovations Inc. ANT Products described by the information in this document are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Dynastream product could create a situation where personal injury or death may occur. If you use the Products for such unintended and unauthorized applications, you do so at your own risk and you shall indemnify and hold Dynastream and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Dynastream was negligent regarding the design or manufacture of the Product.

## Revision History

| Revision | Effective Date | Description |
|---|---|---|
| 1.0 | June 2017 | Initial specification creation |
| 1.1 | June 2017 | Correct formula for success rate |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

## Table of Contents

## List of Figures

## List of Tables

# 1 Overview

ANT BLAZE is a generic wireless mesh solution for advanced Internet of Things (IoT) applications including lighting control, asset management, environmental monitoring, location determination and many others. It provides for a connectionless self healing, self forming topology built upon ANT's superior channel management capabilities. Operating exclusively on Dynastream's D52 Premium modules, it provides an ideal solution for distributed sensor monitoring or control applications where fast, secure and reliable communication to a gateway is required. The gateway is a special node that acts as a primary endpoint of traffic through the network and may provide connectivity to other networking technologies providing internet connectivity (e.g. Wi-Fi, LTE), as shown in Figure 1.

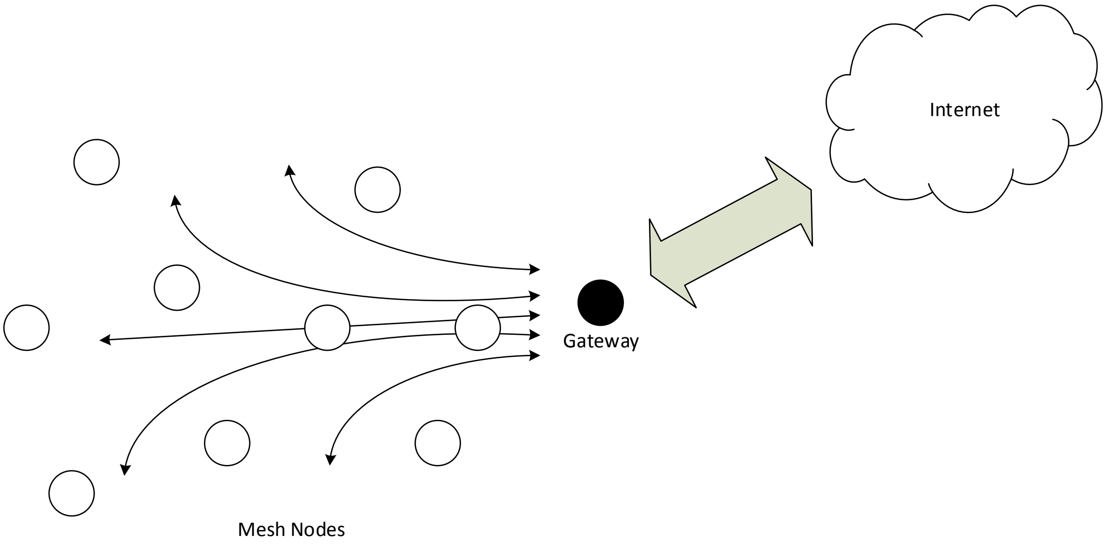


**Figure 1. ANT BLAZE Network**

An ANT BLAZE network supports over 500 mesh nodes interconnecting to a gateway and over 65k uniquely addressable networks.  Network deployments may be sparse or dense as ANT BLAZE can automatically adjust its messaging to provide consistent throughput and reliability. Each node in a network may be addressed individually or as part of a group. Messages may be sent from individual nodes, through the mesh, to a gateway. Or from the gateway, through the mesh, to individual nodes. The throughput of the network, though dependent on node count in the network, is sufficient for sensor or control applications (see Section 6). Barring any application level enhancements an ANT BLAZE network will not interfere with or react to mobile phones or any IT infrastructure.

Any node within an ANT BLAZE network may be extended by the application to connect to a Bluetooth Low Energy (BLE) device (such as a phone) or to another ANT device (such as an ultra-low power sensor). For example, an ANT BLAZE mesh could be used to backhaul information from a wearable coin cell operated sensor to a gateway. A phone application with BLE could be used for configuring the nodes or accessing the gateway through the mesh.

ANT BLAZE consists of two static libraries – one for the node and one for the gateway (Figure 2). The libraries support common compilers for ARM processors (see Section 1.5) and provide a simple API which abstracts the complexities of the mesh from application developers. These libraries operate exclusively on Dynastream D52 Premium modules. The ANT BLAZE SDK, which includes sets of libraries for mesh and gateway nodes, provides interface documentation and sample applications allowing developers to get started quickly. The application may use the radio in parallel to ANT BLAZE, further extending the wireless capability of the application with BLE, ANT or other proprietary protocols.

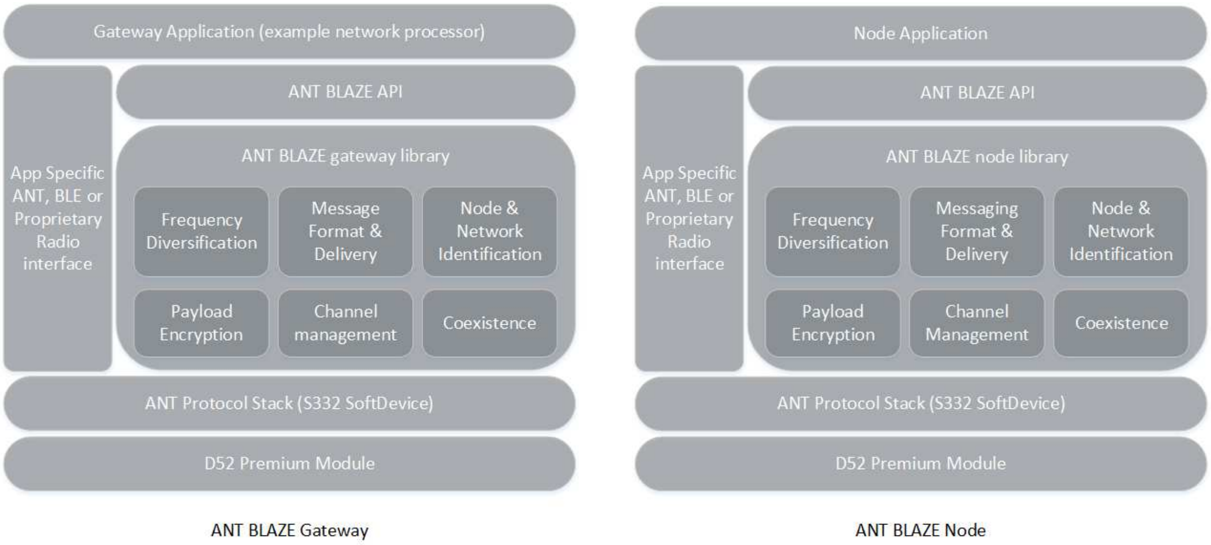

Gateway Application (example network processor)
ANT BLAZE API
App Specific ANT, BLE or Proprietary Radio interface
ANT BLAZE gateway library
Frequency Diversification
Message Format & Delivery
Node & Network Identification
Payload Encryption
Channel management
Coexistence
ANT Protocol Stack (S332 SoftDevice)
D52 Premium Module
ANT BLAZE Gateway
Node Application
ANT BLAZE API
App Specific ANT, BLE or Proprietary Radio interface
ANT BLAZE node library
Frequency Diversification
Messaging Format & Delivery
Node & Network Identification
Payload Encryption
Channel Management
Coexistence
ANT Protocol Stack (S332 SoftDevice)
D52 Premium Module
ANT BLAZE Node

**Figure 2. ANT BLAZE Node and Gateway Stacks**

### 1.1 Feature Summary

The following lists key features of ANT BLAZE technology:

- Self healing, self forming, connectionless mesh network supporting over 500 nodes that can be addressed individually or as a group
- Gateway centric communication, bridging all nodes in the network to single cloud access point
- Operates independent of any IT infrastructure with no impact on Wi-Fi or Bluetooth enabled devices
- Frequency diverse avoiding interference from Wi-Fi or other 2.4GHz radio signals
- Robust and reliable communication with >99% message success rate in dense and sparse installations
- Data throughput suitable for sensor or control applications (dependent on node count and installation). For a typical installation:
  - ~67 bytes/sec (100 nodes)
  - ~47 bytes/sec (300 nodes)
- Encrypted application payload with up to 40 bytes per message
- Embedded libraries, supporting multiple compilers, for end nodes and gateway
- Compatible with RTOS (e.g. FreeRTOS)
- Mesh can run concurrent to other ANT channels or BLE connections on the same device, allowing access to Smartphones and low power sensors outside of the mesh network
- Supported exclusively on Dynastream D52 Premium Modules.

## 1.2 SoftDevice Compatibility

The ANT BLAZE Libraries can be used with the s212 and s332 ANT SoftDevices. For specific SoftDevice version compatibility, refer to the release notes inside the ANT BLAZE SDK.

## 1.3 Hardware Compatibility

The ANT BLAZE Libraries can only be used on Dynastream D52 Premium ANT Modules, including the following part numbers:

- D52QPMM4IA
- D52QPMM4IA-A
- D52MPMM8IA
- D52SKM6IA-A

The D52 Premium ANT Modules are based on Nordic Semiconductor's nRF52832 SoC., supporting ANT, *Bluetooth*® Low Energy and extended features such as NFC. For more information on the D52 ANT SoC Module Series, refer to the "D52 ANT SoC Module Series Datasheet".

## 1.4 Library License Key

The ANT BLAZE Libraries require a license key to operate. An evaluation key is available which will enable full functionality and is to be used for NON-COMMERCIAL USE ONLY.

The library license key is unrelated to the ANT network key or the ANT BLAZE encryption key (see Section 3), and is not used in over the air mesh transmissions.

The license key required for the ANT BLAZE Libraries is different and independent from the license key required to enable an ANT SoftDevice. Please note that an ANT SoftDevice Commercial license is still required. Since ANT BLAZE operates exclusively on Dynastream D52 Premium ANT modules, once the Distribution Agreement for the D52 Module is completed, SoftDevice royalties will be waived for each instance of ANT SoftDevice used within a D52 module for commercial end product.

Further information about licensing can be found at: www.thisisant.com/developer/ant/licensing.

License validation can extend the library initialization time to up to 100 ms.

## 1.5 Compiler Compatibility

The static binary libraries for ANT BLAZE are provided in variations for different compilers, to provide flexibility to developers using their toolchain of choice. The variation of the library must match the compiler used in application development. For example, if the application that is using the ANT BLAZE node library is being compiled with GCC, the "ANT_BLAZE_Node_Library_GCC.a" must be used.

The example applications included in the ANT BLAZE SDK should serve as a starting point for developing custom applications with the ANT BLAZE libraries. For further details on specific compile options, look at the compiler specific example project files and Makefiles. All possible combinations of compiler and linker options have not been tested.

### *1.5.1 GCC*

The GCC variations of the libraries are built with GNU ARM Embedded Toolchain version 6-2017-q1-update, and uses the settings in Table 1. See example application Makefiles for further details.

**Table 1. GCC Specific Compiler Options**

| Option | Description |
|---|---|
| -mcpu=cortex-m4 | Cortex-M4 |
| -mfloat-abi=hard<br>-mfpu=fpv4-sp-d16 | The libraries are built using hardware floating point numbers.<br>Application settings must match. |
| -std=gnu99 | C99 standards are used in the libraries |
| -mthumb | Thumb mode |

### 1.5.2 Keil

The Keil variation of the libraries are built using with Keil MDK V5.06 update 2, and uses the settings in Table 2. See example application Keil project files (uvprojx files) for further details.

**Table 2. Keil Specific Compiler Options**

| Option | Description |
|---|---|
| --cpu Cortex-M4.fp | Cortex M4 processor. The libraries are built using hardware floating point numbers. Application settings must match. |
| --c99 | C99 standards are used in the libraries |
| --apcs=interwork | Thumb or ARM |

### 1.5.3 IAR

The IAR variation of the libraries are built using version 7.30 of IAR Embedded Workbench, and uses the settings in Table 3. See example application IAR Embedded Workbench project files (ewp files) for further details

**Table 3. IAR Specific Compiler Options**

| Option | Description |
|---|---|
| --cpu=Cortex-M4 | Cortex M4 processor |
| --fpu=VFPv4_sp | The libraries are built using hardware floating point numbers.<br>Application settings must match. |
| --endian=little | Little endian mode |

## 1.6 Electrical Characteristics

| Conditions | Average current draw (mA) | Max current draw (mA) |
|---|---|---|
| DC-DC on @ 3V, 0dBm | 5.4 | 8.1 |
| DC-DC off @ 3V, 0dBm | 8.8 | 16.6 |

For additional electrical specifications, refer to the “D52 ANT SoC Module Series Datasheet”.

# 2 Features

## 2.1 Architecture

ANT BLAZE is a connectionless mesh technology where each node scans for the presence of other nodes in the area and forwards any received information to other nodes. The connectionless nature of ANT BLAZE means that the network can form and repair very quickly, without requiring any handshaking or negotiation.

ANT BLAZE technology relies on two elements:

- ANT Background Scanning Channel: A background scanning channel, configured with high duty search, to scan for transmissions from other nodes.
- Beacon: An ANT BLAZE beacon is an ANT master channel, used to introduce messages to the network as well as retransmit messages for other nodes. Beacon channels are bidirectional. The forward direction (what is transmitted over the beacon) is controlled by the ANT BLAZE libraries; messages received on the reverse direction can be processed by the application in the backchannel callback provided when initializing the library. Up to three beacons can be enabled when configuring the ANT BLAZE libraries, refer to Section 2.4 for more details.

An ANT BLAZE network consists of one gateway and a set of up to 510 mesh nodes. Messages in an ANT BLAZE network can only be sent from the gateway to a node or group of nodes, and from a node to the gateway. This means that, without going through the gateway, a message cannot be sent from one node to another node.

The gateway and mesh nodes require different software stacks; nodes should be created using the ANT BLAZE Node Library, while the gateway should use the ANT BLAZE Gateway Library. From a functionality perspective, there are a few differences between the gateway and a mesh node:

- Messages sent from a mesh node use the address field as the source address; the destination of the messages is always the gateway. Messages sent from the gateway use the address field as the destination address (i.e.; either an individual node or a group address) and the source of the message is always the gateway.
- As the gateway is either the source or destination of messages in the network, the gateway does not forward messages for other nodes. The contents of its beacon are only related to messages originating from the gateway.
- A gateway cannot be assigned to a group (refer to Section 2.8 for more information on groups). The gateway will not process messages intended for the broadcast address for addressing "all nodes" either. However, a gateway can send messages to groups of nodes or to all nodes using the standard ant_blaze_gateway_send_message() API call.
- The gateway does not support dynamic channel period selection (refer to Section 2.5 for more information).
- The gateway is optimized for handling a larger amount of incoming traffic than mesh nodes.

## 2.2 Device Identification

ANT BLAZE supports the establishment of multiple collocated mesh networks, each of them comprising a set of mesh nodes and a gateway. Devices that form part of the same network will be able to communicate with each other, and nodes in the same network will relay messages for one another. Nodes can not receive or relay messages from nodes assigned to a different network. To ensure that only devices that are intended to communicate with each other do so, ANT BLAZE provides three levels of device identification: manufacturer level, network level, and device level.

At the manufacturer level, the 8-byte ANT network key serves to distinguish between products made by different manufacturers. Valid ANT network keys can only be generated by Dynastream Innovations Inc. It is highly recommended that manufacturers of products utilizing the ANT BLAZE library purchase a unique network key from Dynastream Innovations Inc. for use in all their products (i.e., one network key per manufacturer), as the use of a unique network key will prevent crosstalk between devices from different manufacturers.

A manufacturer may have many different networks using the same ANT network key – perhaps even in the same area. To distinguish between networks, a network ID must be set. The 16-bit network ID must be the same for each node within a network, but unique from other networks in the area created by this manufacturer. If a manufacturer runs out of unique network IDs and cannot distinguish them geographically, it may be necessary to purchase an additional ANT network key.

Nodes in the network are addressed using a 9-bit node ID which must be unique to each node in the network. The node ID is used to identify the destination of a message, for messages sent from the gateway to a node, and the source of a message, for messages sent from the node to the gateway.

Table 4 summarizes the device identification parameters used in ANT BLAZE, and how they are mapped to ANT channel parameters. For more details on the ANT network key and ANT channel ID, refer to the "ANT Message Protocol and Usage" document.

**Table 4. Device Identification Parameters**

| Device Identification Parameter | Identification Level | Length (bits) | ANT Channel Parameter |
|---|---|---|---|
| ANT Network Key | Manufacturer | 64 | ANT Network Key |
| Network ID | Network | 16 | ANT Channel ID: Device Number |
| Node ID | Device | 9 | ANT Channel ID:<br>Transmission Type (Bits 0-6): Bits 0-6 of Node ID<br>Device Type (Bits 0-1): Bits 8-9 of Node ID |

Each node must be assigned a network ID, node ID and ANT network key by the application before starting the ANT BLAZE library. See Section 3 for information about how this is set. The gateway must also have its own unique node ID.

The mechanisms for assigning these parameters to a node (node commissioning) are application specific, refer to Section 5.3 for best practices.

## 2.3 Message Propagation

Messages propagate through the network using a flooding approach, where nodes rebroadcast messages received from other nodes to all other nodes in range. Flooding provides inherent redundancy and results in minimum latency, since messages will propagate through the shortest path along with other possible paths. All messages will generally reach all nodes in the network, not just the intended destination, however, the library will ensure that only messages intended for a particular node or group address are passed on to the application. Duplicate message detection is also implemented by the library; only unique messages are passed on to the application.

To optimize throughput, ANT BLAZE manages the messages that are retransmitted by each node to allow multiple messages to be handled by the system in parallel. This multi message coordination results in higher throughput when polling data from a set of nodes using a single request compared to polling each node individually. This feature also allows the application to request to send multiple messages one after the other, without needing to wait until a message has reached its destination to send the next. When managing messages, the library gives priority to newer messages that have not yet propagated through the network over older messages, however, older messages continue to be retransmitted to increase the reliability of message delivery.

## 2.4 Frequency Diversity

Frequency diversity is an optional feature which can improve the robustness of the network against wireless interference by transmitting the messages on several ANT frequencies simultaneously, allowing an ANT BLAZE network to function even if one of the frequencies is jammed by interference.

When frequency diversity is disabled, each node operates on only 1 frequency. When frequency diversity is enabled, each node will transmit on up to 3 frequencies configured by the application. Each node will automatically select the best single frequency to receive on from the configured set. If the performance drops on the selected frequency, the library will automatically select one of the other frequencies to start receiving on.

The application must configure 1 to 3 frequencies when configuring the ANT BLAZE network. The chosen frequencies must be the same for all nodes in the network. See Section 3 for information on how to configure the frequencies. Frequencies should be chosen so that they are spaced out from each other and minimize overlap with other known wireless traffic such as BLE, Wi-Fi and other ANT devices operating in the area.

There is a trade off between the throughput of the network and the increase in reliability gained by enabling frequency diversity on high interference environments. When frequency diversity is enabled, typically there will be a decrease in throughput as compared to using a single frequency; the reason for this is that each node is now using additional radio time to transmit redundant data. For an example of this trade-off, refer to the example test setup and results in Section 6.

## 2.5 Enhanced Coexistence in Varying Node Density Scenarios

The scan and forward mechanism requires that all nodes share the same RF space, i.e., all nodes in a network must be transmitting in the same set of frequencies for the background scanning channel to be able to pick up the transmissions. ANT transmitters sharing the same radio frequency manage coexistence in the time domain, occupying the radio frequency for small periods of time (timeslots). ANT's unique adaptive isochronous mechanism allows nodes to actively detect and avoid interference with each other by being "time-aware" of each other. When opening a master channel (beacon), ANT will only transmit in clear timeslots. During operation, after every transmission, ANT will also open a receive window to listen for drifting transmissions (e.g. due to natural component variance), and will alter its transmission slot by retiming retransmissions if necessary to avoid collisions.

In environments with a high density of mesh nodes, the master channels will detect each other and move out of the way, allowing a better use of the available timeslots. This mechanism is very effective, however, there is a physical limit on how many transmissions can fit in a given time span.

To further improve performance on high density scenarios, each node in the ANT BLAZE network can dynamically change the length of its channel period based on the density of nodes in its immediate vicinity. This increases the total throughput and reliability of the network.

To enable dynamic management of the channel period on an ANT BLAZE node, the channel_period parameter in the node configuration must be set to 0. To manually specify the channel period instead, set the channel_period parameter to any other valid ANT channel period.

Note that this functionality exists only on ANT BLAZE Nodes and not on the gateway. If the channel_period is configured to 0 on the gateway, it will default to 4 Hz.

## 2.6 Messaging

Over the air mesh packets support a maximum 5 bytes of application payload per packet. To support larger application payloads (up to 40 bytes per message), fragmentation and reassembly are implemented by ANT BLAZE. The library will automatically split larger messages into packets at the source node, and will reassemble the fragments at the destination, ensuring that only fully reassembled messages are passed to the application.

Message transmission latency will generally increase as payload size increases, due to the increase in the number of packets required to send the entire message, as well as potential retransmissions to ensure that all the pieces are received successfully. Actual transmission latency depends on the specific network setup and environment, but an example illustrating the effect of payload size in latency is provided in Section 6.

## 2.7 Message Indexing

To uniquely identify received messages, the library uses a 32-bit monotonically increasing index number. This index is provided to the application together with the reassembled received message. Index numbers can be used by the application to identify whether a message is older than another one: higher index numbers are associated with newer messages. This is useful when multiple messages are generated close in time. These messages may arrive out of order at the destination, as their multiple pieces are transmitted in parallel across the network; the index number can be used to determine their original intended order.

## 2.8 Groups

### *2.8.1 Broadcast Addressing*

Address 0, the broadcast address, is a special address that identifies all nodes in a network. A message sent with destination address 0 will be received by all nodes.

### *2.8.2 Group (Multicast) Addressing*

A group address is an identifier for a group of nodes in the network. Group addresses allow the gateway to send a single message to be received by multiple nodes simply by setting the destination address to the address corresponding to a group. Nodes belonging to a group must still have a unique node ID, i.e., group addresses do not replace individual node IDs.

The number of addresses reserved for groups must be configured by the application at every node; the configuration must match across all nodes in the network. This configuration option allows the application to balance the number of individual nodes and groups in a network to better suit its requirements. The number of group addresses can be configured to any number between 0 (no groups) and 511 (all addresses are group addresses, except for the broadcast address). The range of group addresses encompasses from (512 – num_group_addresses) to 511.

For example, consider an application that requires 10 addressable groups in the network. To achieve this, the num_group_addresses parameter in the node configuration must be set to 10.  This would reserve addresses 502 to 511 as group addresses. Addresses in the range 1 to 501 can be used to identify individual nodes.

In order to receive messages intended for a group address, the application must add the node to a group. Each node can be assigned to up to 8 groups. For the example above, if a node with node ID = 5 is added to group 508, it will receive messages with destination address 5 (its assigned node ID), 508 (the group address) and 0 (broadcast address). If the node is later added to group 510, it will additionally receive messages with destination address 510. Nodes can be added to and removed from groups at any time.

## 2.9 Security

The payload of the mesh packets can be optionally encrypted using AES-128 in CTR mode, to prevent eavesdropping of transmitted data.

Payload encryption needs to be enabled when configuring the library, along with a 128-bit key. The same key must be configured for all nodes in the network. Key distribution and generation mechanisms are defined by the application.

# 3 Application Programming Interface

The API reference for the ANT BLAZE Node Library and the ANT BLAZE Gateway library can be found in the headers in the inc folder included in the ANT BLAZE Libraries package. These files need to be included by the application to access the ANT BLAZE static libraries. Function definitions and API documentation for the ANT BLAZE Node Library and ANT BLAZE Gateway library are present in the *ant_sf_interface.h* and *ant_sf_gateway_interface.h* headers, respectively. Constants and data structures that are used by both libraries are defined in *ant_sf_defines.h*.

To use either of the ANT BLAZE libraries, the application must follow the sequence depicted in Figure 3. For the specific name of the library calls outlined in the diagram, refer to the API headers for the corresponding library. It is highly recommended that library calls are not performed in interrupt context.

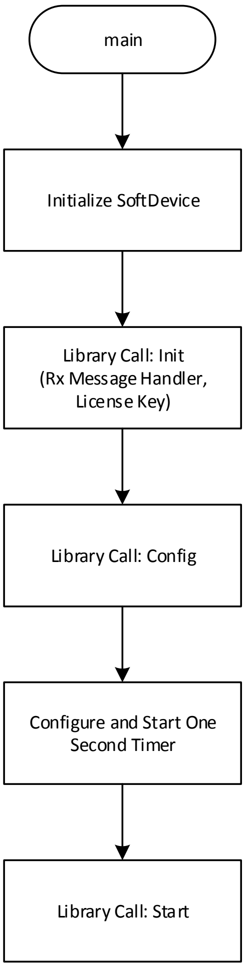


**Figure 3. ANT BLAZE Usage Sequence**

The application must configure and enable the SoftDevice, with at least 5 ANT channels enabled. ANT channels 0-4 are reserved for use by the library. All ANT channels used by the libraries are assigned to ANT network 0. Additional ANT channels, as well as BLE connections, can be enabled/configured as required by the application.

After configuring the SoftDevice, the application must initialize the library using a valid library license key. During initialization, the application configures a handler for messages received over the mesh network, a handler for messages received over the beacon's back channel, and an error handler for errors generated by the library itself.

Once the library is initialized, the application must configure the library and specify:

- Node ID: 9-bit identifier for each node in a network. Must be unique for all nodes in the network, including the gateway (i.e., the gateway must have a node ID within the valid range that is different from the ID of all other nodes in the network).
- Network ID: 16-bit identifier for a network (deployment).
- Channel Period: ANT channel period for the beacons. Setting this value to zero enables the dynamic channel period feature.
- Radio Frequencies: List of up to three radio frequencies that beacons use to transmit on.
- Number of Channels: Number of beacons to enable (maximum 3).
- Transmit Power: ANT transmit power level for the beacons.
- Number of Group Addresses (node library only): Number of addresses reserved as group addresses.
- Encryption Enabled: Specifies whether payload encryption is enabled or not.
- ANT Network Key: 8-byte ANT network key used by all ANT channels in use by the library. The network key is used to distinguish between networks created by different manufacturers. This network key will be applied to network 0. See section 2.2 for details about obtaining a private network key.
- Encryption Key: 16-byte encryption key for a network. The encryption key is used to encrypt and decrypt the payload of the ANT BLAZE messages for security. All possible 16-byte values of the encryption key are valid. See section 2.9 for more details.

All configuration elements must be explicitly configured by the application. Configuration parameters (except by node ID) must match on all nodes in the network, including the gateway.

The application must configure and start a one second timer. On every one second interval, the application must call the process timeout function of the library, as shown in Figure 4. Once the timer has started, the application can start the library as well.

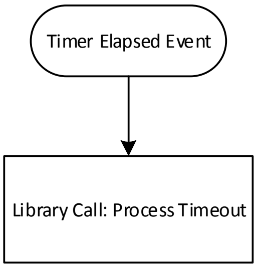


**Figure 4. Timer Event Handling**

Figure 5 shows the application flow for handling ANT events. The application must process directly ANT events generated by the SoftDevice, and pass the ANT events for the channels in use by the library (ANT channels 0-4) to the library. ANT events for custom channels configured by the application can be handled directly by the application.

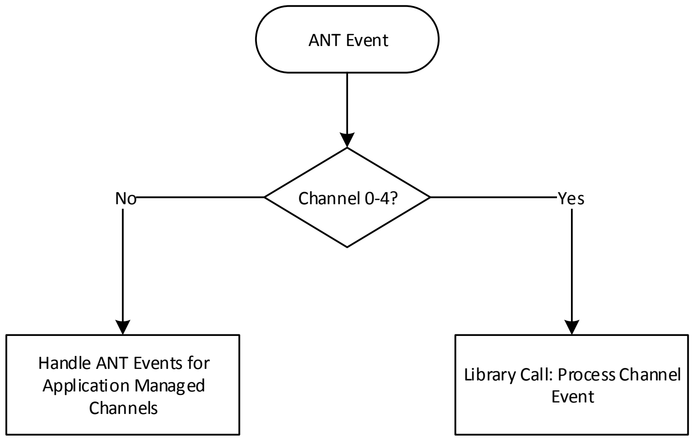

ANT Event
Channel 0-4?
No
Yes
Handle ANT Events for Application Managed Channels
Library Call: Process Channel Event

**Figure 5. ANT Event Handling**

The application can request to send messages through the mesh network at any time. For example, Figure 6 illustrates how sending messages can be initiated by application specific events (e.g. a button press), as well as responding to application defined messages received over the mesh network.

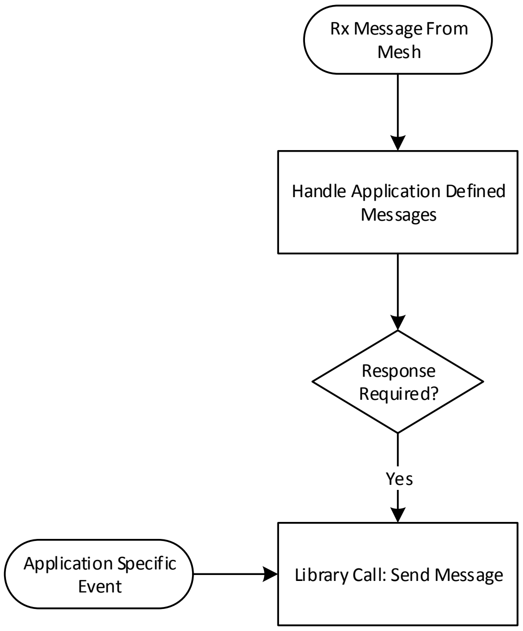

Rx Message From Mesh
Handle Application Defined Messages
Response Required?
Yes
Application Specific Event
Library Call: Send Message

**Figure 6. Receive Message Handler and Sending Messages**

When sending a message from a node to the gateway, it is not necessary to configure the address field in the message to send, as the address is automatically set in the library to the node ID of the current node (i.e. current node ID is the source address for the message).

When sending a message from the gateway to a node, the address field in the message to send should be set to the intended destination of the message (e.g., if sending a message to node with ID 5, the address should be set to 5). Broadcast and multicast messaging is supported as described in Section 2.8.

# 4 Application Integration Considerations

The Premium D52 modules contain a single radio which must be shared by ANT BLAZE, BLE and any application specific ANT channels. It is important to be aware of how much radio time BLE and custom ANT channels are using so that the ANT BLAZE network remains reliable. As more radio time is used by the application,less radio time is given to ANT BLAZE to scan for messages, increasing the latency for retransmitting messages.

## 4.1 BLE

ANT BLAZE is designed to work while the application is using BLE; however, some general guidelines should be observed. As ANT BLAZE relies on the ability to use a background scanning channel for as long as possible, it is recommended to implement a BLE peripheral, as the scanning operation on BLE central devices cannot be performed while ANT BLAZE is active. Very short connection intervals or advertising intervals take up a lot of radio time which could degrade the performance of ANT BLAZE. Advertising intervals at or above 200ms will limit the performance degradation to the ANT BLAZE mesh. Testing has been performed at 200ms advertising interval with minimal effect on mesh throughput. If a BLE connection is required, the connection interval should be kept at or above 100ms. Testing has been performed with 100ms connection interval where the BLE connection was maintained for one hour without a significant effect on the performance of the network. Shorter connection intervals may be used if the BLE connection is only required to be maintained for a short amount of time, but some decrease in ANT BLAZE throughput must be expected while the connection is active, and the node involved in the connection may not be capable of responding to messages while engaged in a connection. If the BLE connection interval is shortened enough such that a reliable connection cannot be maintained, it is recommended that ANT BLAZE be temporarily stopped until the BLE connection is terminated.

## 4.2 Custom ANT Channels

The application may also configure custom ANT channels to communicate with devices outside of the ANT BLAZE network. Note that ANT channel numbers 0-4 are reserved for the ANT BLAZE library and must not be modified by the application. Any ANT channels configured by the application must be channel number 5 or higher. ANT network number 0 is used by ANT BLAZE and its network key should not be modified by the application. The network key is configured when calling ant_blaze_node_config() or ant_blaze_gateway_config(), and custom channels can operate using this same network key after these calls. Performance of the ANT BLAZE network will decrease as more application specific ANT channels are added and as the channel periods of the application specific ANT channels decrease.

ANT BLAZE is always background scanning in high duty search mode. Since only one search can be run at a time, the application may not run its own search unless it is also using the ANT BLAZE network key, RF frequency, and CRC mode (3-byte). Therefore, if the application needs to perform a search (e.g., when opening an ANT slave channel), it is recommended to stop the ANT BLAZE network temporarily and then restart it after completing the search. To simplify integration of custom ANT connectivity with ANT BLAZE, it is recommended to use ANT master channels if possible.

Upon initialization, ANT BLAZE sets up Lib Config so that the Channel ID is output in the extended data of each received message. The ANT BLAZE initialization function will set the required flag in Lib Config while not modifying other flags that may have been set by the application. The application must not disable Channel ID output after the ANT BLAZE library has been initialized or the library will not function. For more information on Lib Config, see the ANT Message Protocol and Usage document.

## 4.3 Flash Operations

While the ANT BLAZE library is running, flash erases are not possible and flash writes may fail. Therefore, it is recommended to temporarily stop the ANT BLAZE library while doing flash operations. See the s332 SoftDevice Specification Document (SDS) for more information on high duty search and using it with various SoftDevice features.

# 5 Best Practices

## 5.1 Physical Installation

The upper end of range for ANT on the D52 Premium module is 30 metres with line of sight. Actual range for a specific product incorporating the module will vary with the enclosure and environment. For best results, limit the spacing between ANT BLAZE nodes to 10 meters. If it is necessary to make a hop greater than 10 meters, or there is no line of sight between nodes, consider placing additional nodes as relays to help span the gap.

Avoid "daisy chain" situations where each node is only in range of 2 other nodes. For the best reliability, messages should have multiple paths available to reach their destination. The ANT BLAZE API provides the ability to retrieve the number of nodes in range for a given node; developers can make use of this information to aid with deployment, for example, providing a warning to users when a low number of nodes near the current node is detected.

Very dense layouts can also cause issues due to over utilization of available time slots in each RF frequency as well as increase in network traffic. Densities of up to 1 node / 3.5m$^2$ were tested with only slight degradation in performance. In higher density situations, it may be necessary to scale back the rate of data being requested from the network. See Section 6.2 for test results for a high density layout.

Other considerations also apply which are consistent across all 2.4GHz technologies such as limiting the amount of metal in the enclosures, avoiding placing nodes at ground level, and not placing nodes in the immediate vicinity of other wireless transmitters (e.g. placing nodes at least 2 meters away from Wi-Fi access points).

## 5.2 Gateway Installation

The gateway should be installed in a central location in the network in range of several ANT BLAZE mesh nodes. Standard ANT BLAZE placement recommendations covered earlier in section 5.1 apply for the gateway as well.

Only one gateway should be used per ANT BLAZE network, but there are situations where more than one gateway may be required for a single installation. An installation here means a single location where an ANT BLAZE mesh network is to be deployed but may consist of several individual networks.

For instance, if there are more nodes required in an area than there are available addresses (510 maximum, depending on number of addresses assigned to groups), it will be necessary to create more than one ANT BLAZE network and thus have one gateway for each network. If there are large gaps in installation location, such as several separate buildings or metal fire walls separating parts of the building, it may also be necessary to create separate networks with individual gateways.

If several networks are to occupy the same area, ensure that they do not interfere with each other by assigning them different network ID's. See section 2.2 for information on node and network identification. While overlapping networks using the same RF frequencies can coexist together, the dynamic channel period selection does not take into account nodes from a different network when determining network density, so depending on the density of the multiple networks, it may be desirable to configure them with different radio frequencies.

## 5.3 Node Commissioning

The commissioning scheme (how each node is assigned its addressing information) is left up to the application and is out of scope of the ANT BLAZE library.

Some examples of possible commissioning schemes include:

- Writing node IDs and network IDs into flash or non-volatile registers such as UICR during manufacturing
- Setting the node ID and network ID based on hardware configuration
- Using a PC or mobile app to assign the node ID and network ID during the setup of the network over a custom BLE connection or a custom ANT channel

In any case, it is critical that no duplicate node IDs exist within a single network.

## 5.4 Throughput Optimization

For collecting data from a large number of nodes while maximising throughput, a "group polling" mechanism is recommended. Due to the way ANT BLAZE routes the messages, it is the same amount of network traffic to send a message from the gateway to a specific node ID as it is to send a message to all the nodes or a group of nodes in the network. Therefore, the gateway can send a data request to an entire group of nodes using the grouping feature in a single message. Each node then can reply with the requested data. ANT BLAZE will manage the responses from the group of nodes, ensuring they are transmitted in alternating time slots so that they can reach the gateway without further management from the application. The gateway collects the data and sends out the request to the next group and so on until all nodes have been polled. For example, if there were 100 nodes in a network with groups of 20 nodes each, the gateway would only need to send 5 messages to request data from all of nodes. This is the approach used in the example in Section 6.

The ANT BLAZE network can route several messages with different origins and destinations through the network concurrently. Another strategy for optimizing throughput is making use of multi packet messages (larger than 5 bytes). For example, if groups of 20 nodes are polled, requesting each node to send a 5-packet response, the gateway would only need to send 5 messages to request 40 bytes of data from every node. When requesting data from multiple nodes at a time, it is recommended that the total traffic does not exceed 100 packets at a time.

Both parameters – size of groups (number of nodes to request data from simultaneously) and response size (number of packets in the response transmitted by each node) can be varied to meet specific application needs. The best combination of these parameters may vary with node density, distribution, number of nodes. In general, higher success rate can be achieved if collecting smaller amounts of data from many nodes, than collecting large amounts of data from a few nodes. This is because no partial messages are passed on to the application, so if a single packet from a multipacket message is missed at the gateway within the timeout, the entire message is missed.

## 5.5 Retries

Automatic retries are already implemented in the ANT BLAZE libraries, however, additional application level retries can be implemented at the gateway for added reliability. When polling nodes for data, if a response has not been received by the gateway within a defined timeout, the application at the gateway can simply re-request the data from that particular node.

## 5.6 Asynchronous Messaging

ANT BLAZE is suitable for applications requiring asynchronous communication, where devices send messages triggered by events, instead of when polled by the gateway. In this type of applications, it is recommended that the total traffic in the network does not exceed 100 packets (500 bytes) within an interval of 20 seconds.

# 6 Appendix A - Example Setup and Performance Testing

This section provides some examples of the expected performance for an office-building installation of the ANT BLAZE mesh. Note that performance will vary based on any number of factors such as network interference, building construction, etc.

For these examples, the network was set up in an office building which contained cubicles and enclosed offices. Testing was conducted using a 100-node layout, with frequency diversity enabled and disabled; as well as a 300 node, high density layout with frequency diversity enabled. The tests were performed using the Mesh Tester tool which uses the "group polling" approach as described in section 5.4.

For example, a ping request message could be sent to node IDs 1-20 requesting a 25-byte response from each node ID, then node IDs 21-40 would be sent the same request, then nodes 41-60, etc. until all nodes had been pinged, then the next cycle would begin. Hundreds of cycles were run per test to get averaged data, and tests were repeated to verify repeatability of the results.

The total time it takes from when a request message is issued by the gateway until the all the response messages are received is recorded. This recorded time and the amount of data received is combined to calculate the throughput of the network for the entire test. The formula used to calculate the data throughput is as follows:

$$Throughput = \frac{Number\ of\ Bytes\ Received\ from\ All\ Ping\ Responses}{Total\ Time\ Waiting\ for\ Ping\ Responses}$$

Success rate was calculated by the following formula:

$$Success\ Rate = \frac{Number\ of\ Ping\ Responses\ Received}{Number\ of\ Ping\ Responses\ Requested} * 100\%$$

The ANT BLAZE frequencies used in this test were chosen to avoid BLE advertising frequencies and Wi-Fi bands present in the test building (Channels 1, 6, and 11). The test office contained 21 Wi-Fi access points distributed throughout, as well as additional Wi-Fi traffic from neighboring buildings, as confirmed by a site survey. All ANT BLAZE nodes and gateway were placed at least 2 meters away from wireless access points, both vertically and horizontally. Figure 7 shows the ANT BLAZE frequencies in relation to nearby Wi-Fi bands, BLE advertising frequencies, and the heavily utilized (in this environment) ANT+ network frequency.

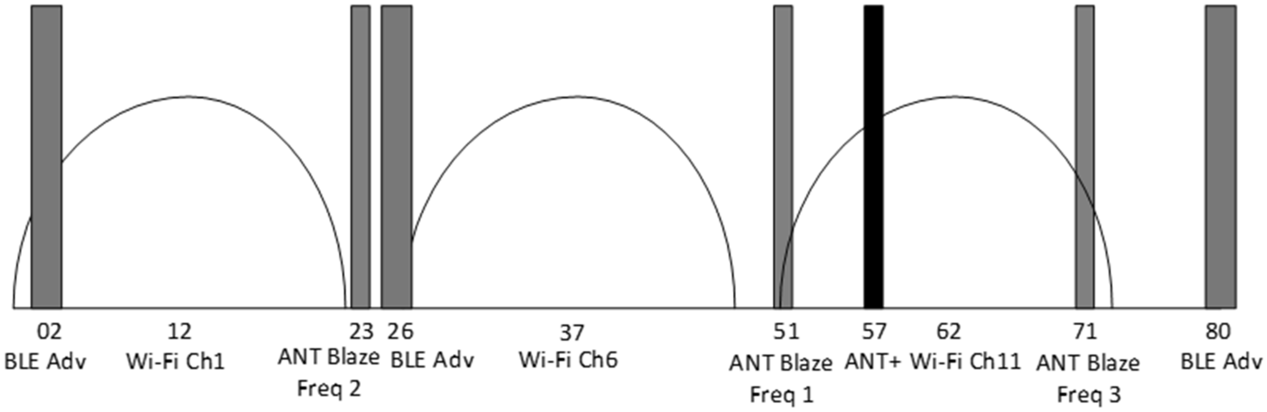


**Figure 7. Selected ANT BLAZE Frequencies and Interferers**

## 6.1 Low Node Density Environment

To evaluate the performance of ANT BLAZE in a low node density environment, 100 nodes were distributed as shown in Figure 8.

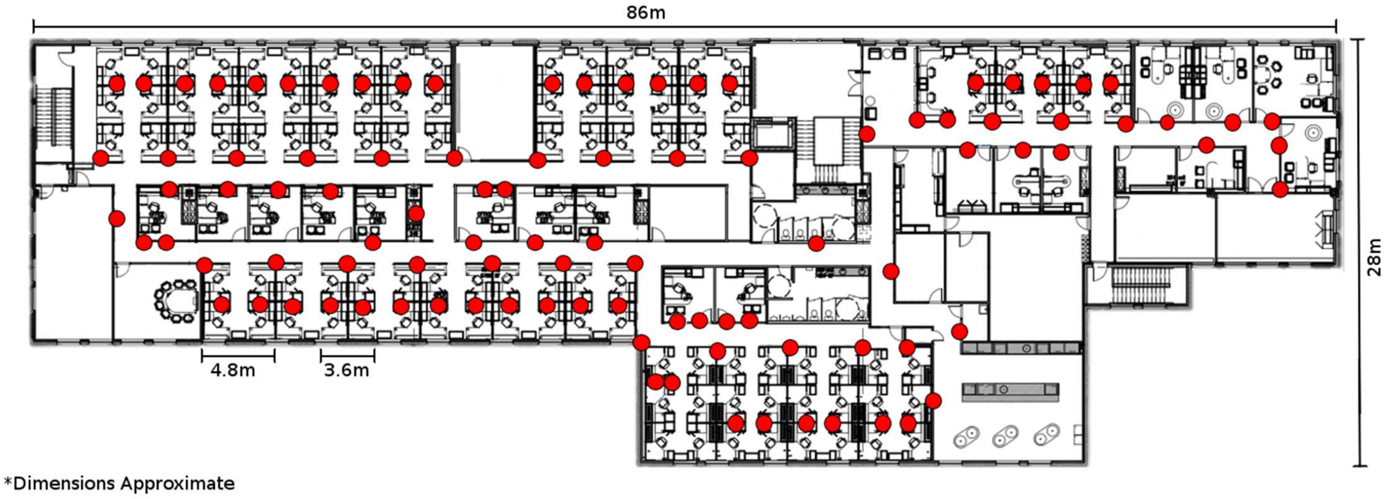


**Figure 8. 100 Node Placement Map**

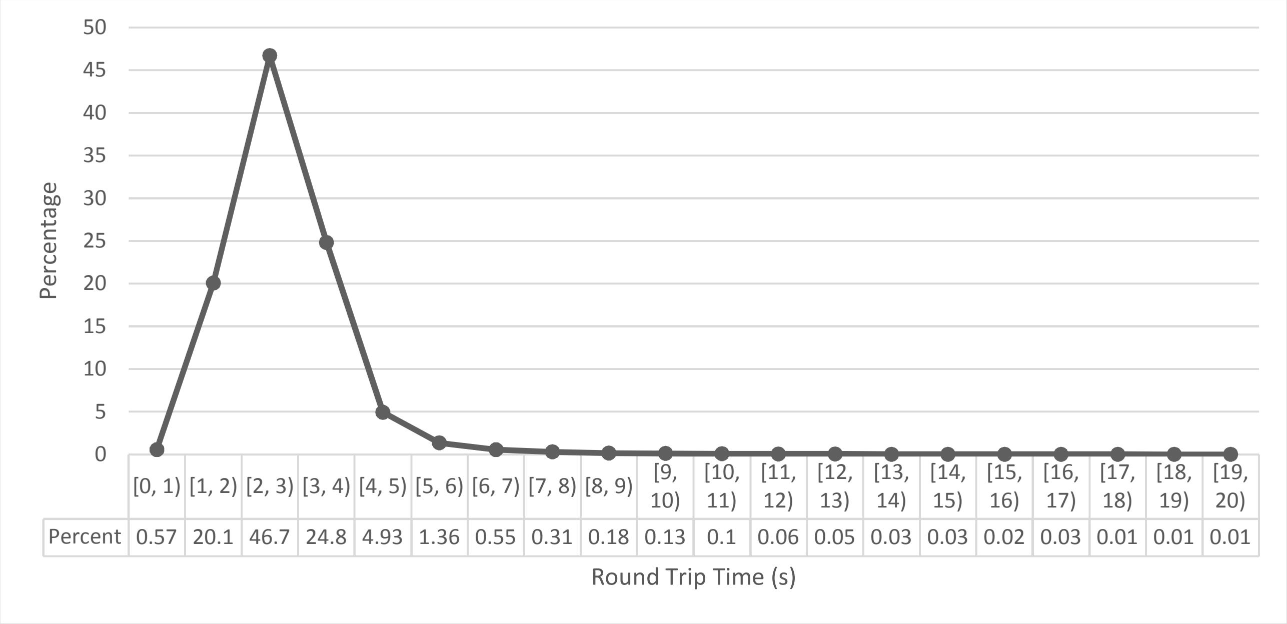
To perform the test, 20 nodes were pinged at a time and the amount of data requested from each node was 25 bytes. For the single frequency scenario, frequency diversity was disabled, and the radio frequency used was 23. For the frequency diversity scenario, the 3 radio frequencies used were 23, 51, and 71. A summary of the results of the tests is provided in Table 5. In the test environment, operation in a single frequency provided a high success rate due to the judicious selection of the frequency. Enabling frequency diversity resulted in a slightly higher success rate, with a significant decrease in throughput.

**Table 5. Low Node Density Test Result Summary**

| Scenario | Success Rate<br>(%) | Average Throughput<br>(Bytes/s) | Average Time to Poll Data from all Nodes<br>(s) |
|---|---|---|---|
| Single Frequency | 99.50 | 95.78 | 26.1 |
| Frequency Diversity | 99.679 | 60.746 | 41.15 |

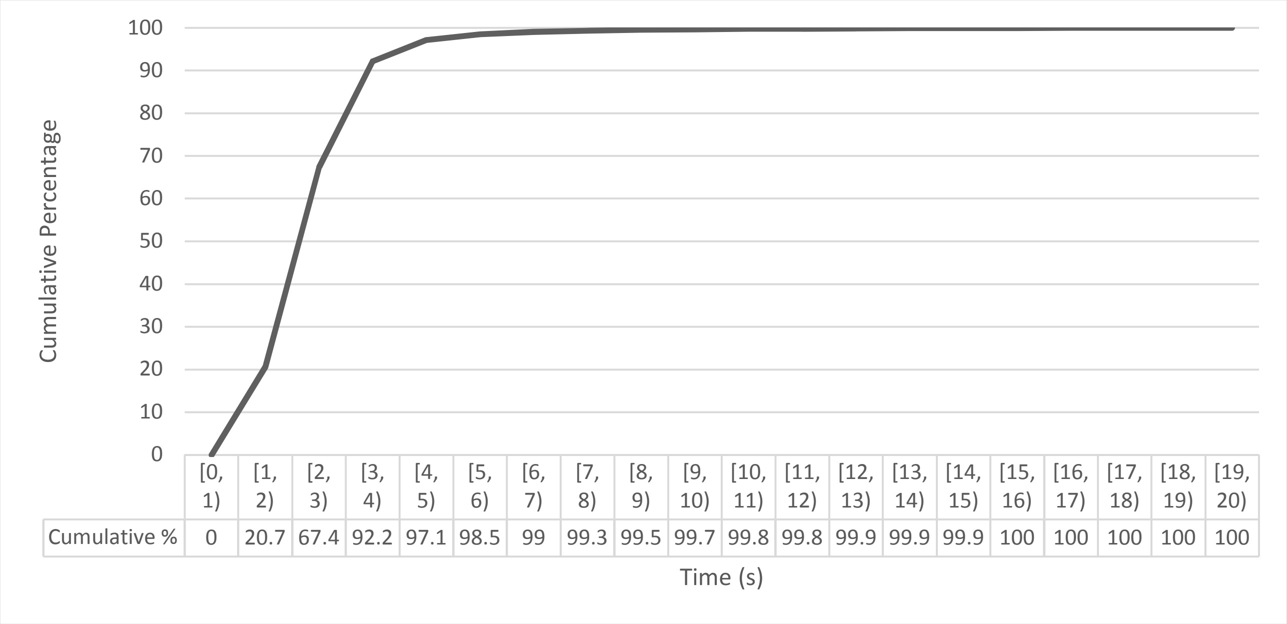
### *6.1.1 Single Frequency Test Results*

Figure 9 is a histogram of the round trip times for individual ping responses, showing the distribution of the latency of the received messages in bins, e.g. between 0-1 seconds, between 1-2 seconds, etc. For example, from this graph, the largest percentage (46.69%) of responses came between 2 and 3 seconds after the request was sent. Figure 10 shows the cumulative distribution of the round trip times. From this graph, over 90% of messages were received under 4 seconds. This data may be useful in deciding how long to wait for responses before issuing a retry.



| | [0, 1) | [1, 2) | [2, 3) | [3, 4) | [4, 5) | [5, 6) | [6, 7) | [7, 8) | [8, 9) | [9, 10) | [10, 11) | [11, 12) | [12, 13) | [13, 14) | [14, 15) | [15, 16) | [16, 17) | [17, 18) | [18, 19) | [19, 20) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Percent | 0.57 | 20.1 | 46.7 | 24.8 | 4.93 | 1.36 | 0.55 | 0.31 | 0.18 | 0.13 | 0.1 | 0.06 | 0.05 | 0.03 | 0.03 | 0.02 | 0.03 | 0.01 | 0.01 | 0.01 |

**Figure 9. Round Trip Time Distribution (100 Nodes, Single Frequency)**



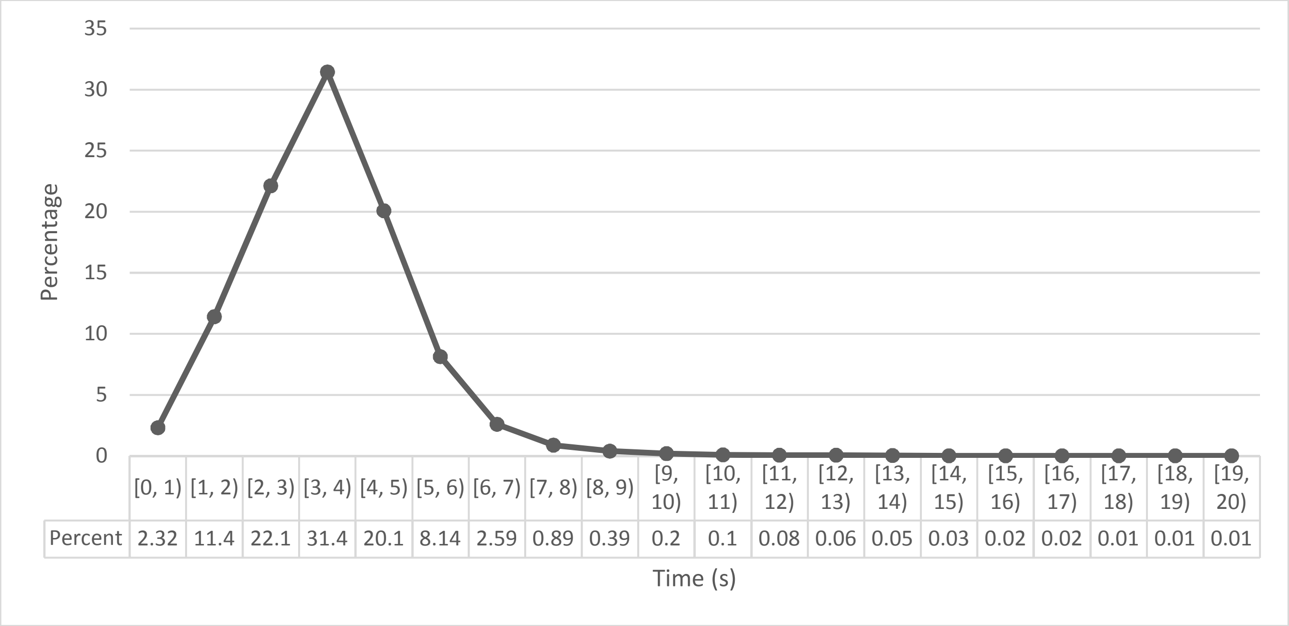
| | [0, 1) | [1, 2) | [2, 3) | [3, 4) | [4, 5) | [5, 6) | [6, 7) | [7, 8) | [8, 9) | [9, 10) | [10, 11) | [11, 12) | [12, 13) | [13, 14) | [14, 15) | [15, 16) | [16, 17) | [17, 18) | [18, 19) | [19, 20) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cumulative % | 0 | 20.7 | 67.4 | 92.2 | 97.1 | 98.5 | 99 | 99.3 | 99.5 | 99.7 | 99.8 | 99.8 | 99.9 | 99.9 | 99.9 | 100 | 100 | 100 | 100 | 100 |

**Figure 10. Round Trip Time Cumulative Distribution (100 Nodes, Single Frequency)**

.

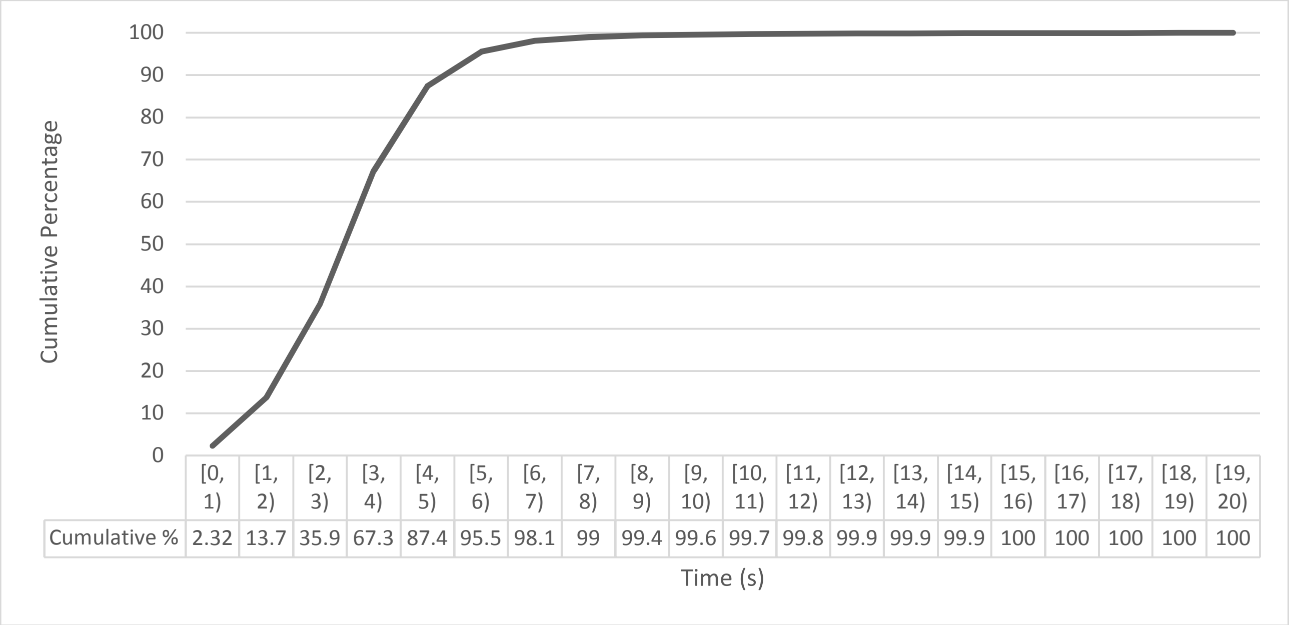
### *6.1.2 Frequency Diversity Test Results*

Figure 11 shows a histogram of the response round trip time for the frequency diversity scenario; the largest percentage (31.44%) of responses came in between 3 and 4 seconds after the request was sent. Notice that the addition of frequency diversity shifted the entire distribution to the right, as compared to the single frequency scenario. Figure 12 shows the cumulative distribution of the round trip times. The cumulative distribution graph shows that over 90% of messages had been received by the 6 second mark.



| | [0, 1) | [1, 2) | [2, 3) | [3, 4) | [4, 5) | [5, 6) | [6, 7) | [7, 8) | [8, 9) | [9, 10) | [10, 11) | [11, 12) | [12, 13) | [13, 14) | [14, 15) | [15, 16) | [16, 17) | [17, 18) | [18, 19) | [19, 20) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Percent | 2.32 | 11.4 | 22.1 | 31.4 | 20.1 | 8.14 | 2.59 | 0.89 | 0.39 | 0.2 | 0.1 | 0.08 | 0.06 | 0.05 | 0.03 | 0.02 | 0.02 | 0.01 | 0.01 | 0.01 |

**Figure 11. Round Trip Time Distribution (100 Nodes, Frequency Diversity)**



| | [0, 1) | [1, 2) | [2, 3) | [3, 4) | [4, 5) | [5, 6) | [6, 7) | [7, 8) | [8, 9) | [9, 10) | [10, 11) | [11, 12) | [12, 13) | [13, 14) | [14, 15) | [15, 16) | [16, 17) | [17, 18) | [18, 19) | [19, 20) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cumulative % | 2.32 | 13.7 | 35.9 | 67.3 | 87.4 | 95.5 | 98.1 | 99 | 99.4 | 99.6 | 99.7 | 99.8 | 99.9 | 99.9 | 99.9 | 100 | 100 | 100 | 100 | 100 |

**Figure 12. Round Trip Time Cumulative Distribution (100 Nodes, Frequency Diversity)**

## 6.2 High Node Density Environment

300 nodes were distributed in a high-density configuration across 2 floors as shown in Figure 13. The nodes depicted in the left side of the lower level were placed in varying heights across a staircase.

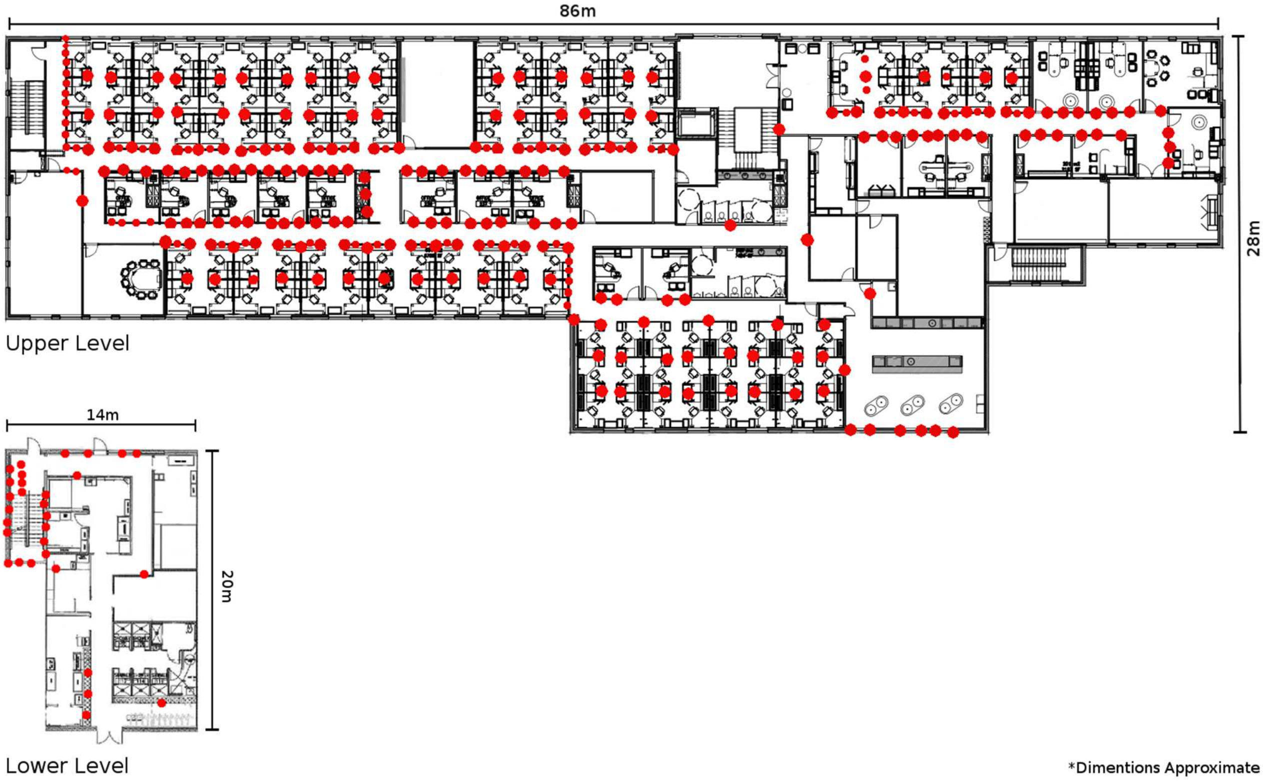


**Figure 13. 300 Node Placement Map**

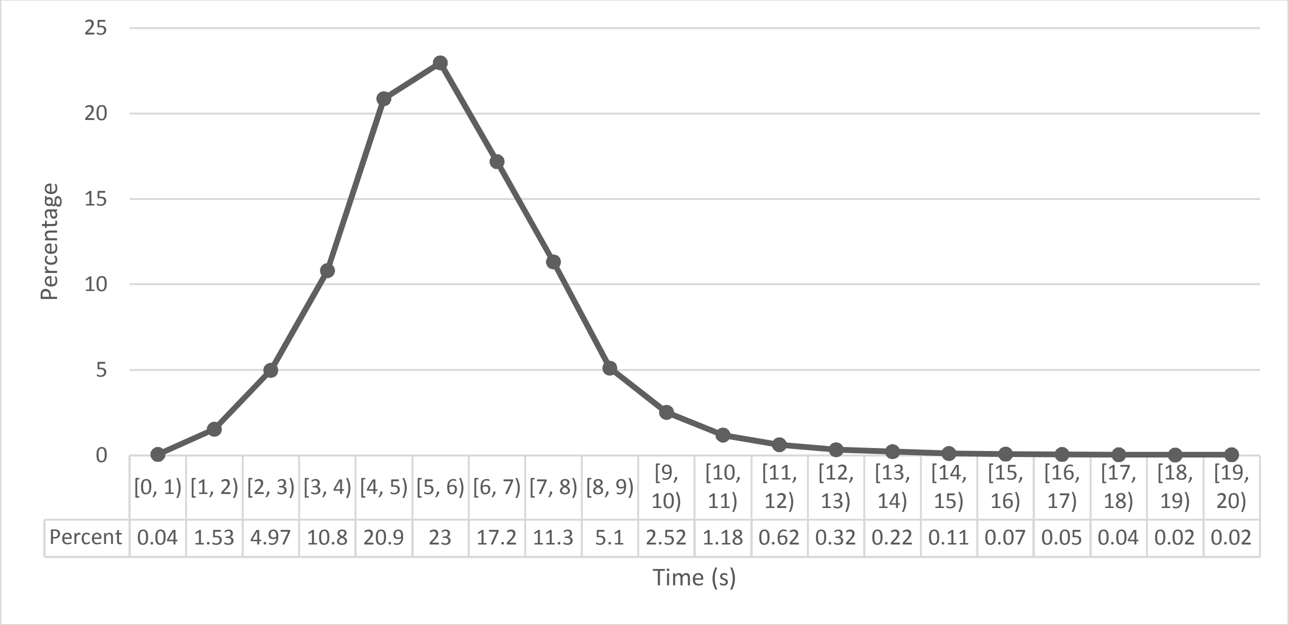
In this test, frequency diversity was enabled, using radio frequencies 23, 51, and 71. Sets of 20 nodes were pinged at a time and the amount of data requested from each node was 20 bytes for the large payload test, and 10 bytes, for the small payload test. A summary of the results of the tests is provided in Table 6. The overall throughput and success rate was lower than the low node density scenario, due to nodes decreasing their transmission rate to accommodate the large number of transmissions. For high density situations like this, it may be advisable to modify the polling scheme to request smaller payloads.

**Table 6. High Node Density Test Result Summary**

| Scenario | Success Rate<br>(%) | Average Throughput<br>(Bytes/s) | Average Time to Poll Data from all Nodes<br>(s) |
|---|---|---|---|
| Large Payload (20 bytes) | 98.519 | 39.426 | 152.18 |
| Small Payload (10 bytes) | 99.667 | 47.559 | 63 |

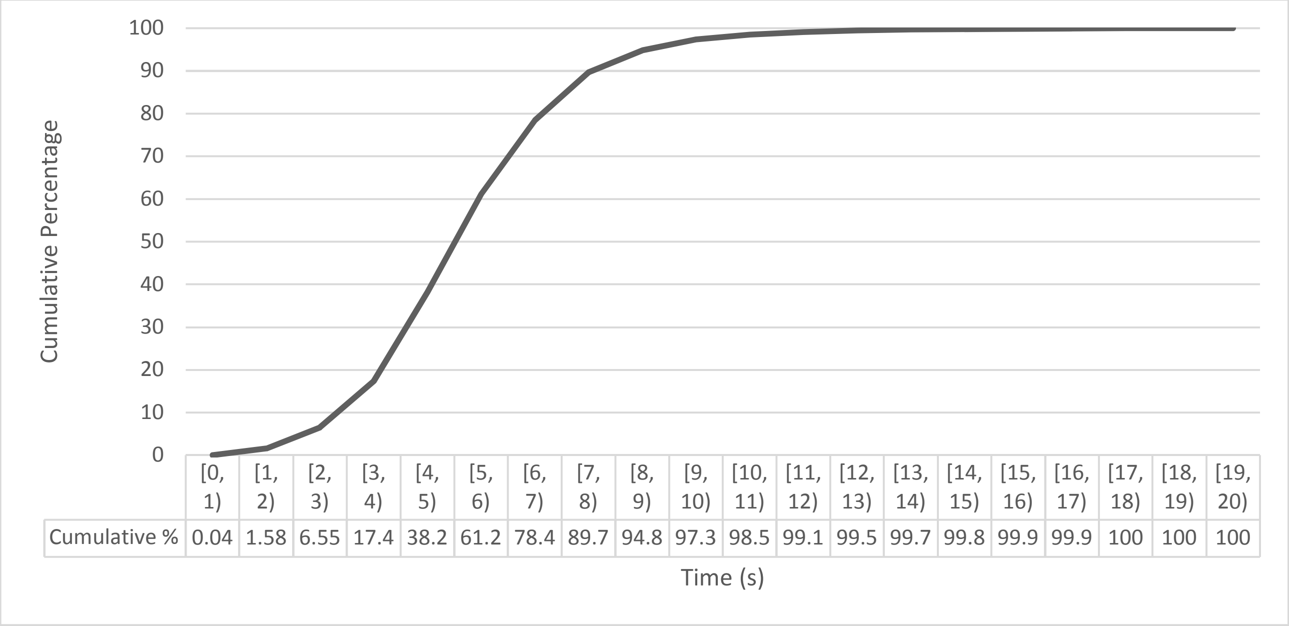
### *6.2.1 Large Payload Test Results*

Figure 14 shows the largest percentage (23%) of responses came in between 5 and 6 seconds after the request was sent. Figure 15 shows that over 90% of the responses came in under 9 seconds.



| | [0, 1) | [1, 2) | [2, 3) | [3, 4) | [4, 5) | [5, 6) | [6, 7) | [7, 8) | [8, 9) | [9, 10) | [10, 11) | [11, 12) | [12, 13) | [13, 14) | [14, 15) | [15, 16) | [16, 17) | [17, 18) | [18, 19) | [19, 20) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Percent | 0.04 | 1.53 | 4.97 | 10.8 | 20.9 | 23 | 17.2 | 11.3 | 5.1 | 2.52 | 1.18 | 0.62 | 0.32 | 0.22 | 0.11 | 0.07 | 0.05 | 0.04 | 0.02 | 0.02 |

**Figure 14. Round Trip Time Distribution (300 Nodes, 20-Byte Payload)**

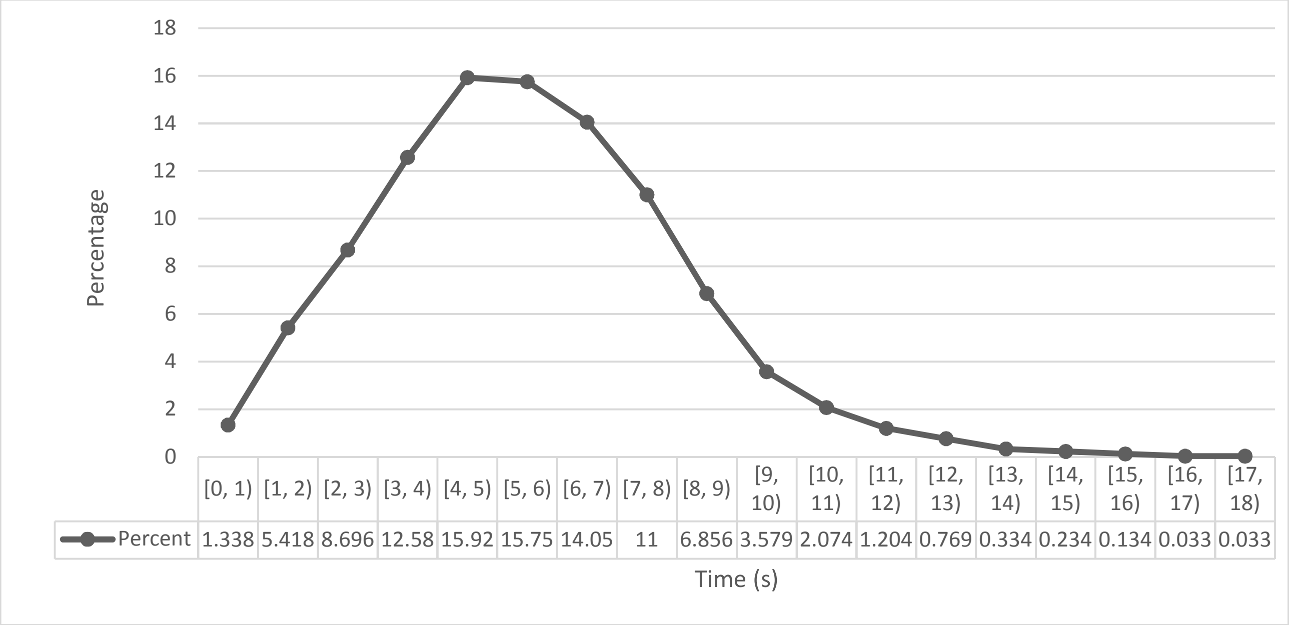


| | [0, 1) | [1, 2) | [2, 3) | [3, 4) | [4, 5) | [5, 6) | [6, 7) | [7, 8) | [8, 9) | [9, 10) | [10, 11) | [11, 12) | [12, 13) | [13, 14) | [14, 15) | [15, 16) | [16, 17) | [17, 18) | [18, 19) | [19, 20) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cumulative % | 0.04 | 1.58 | 6.55 | 17.4 | 38.2 | 61.2 | 78.4 | 89.7 | 94.8 | 97.3 | 98.5 | 99.1 | 99.5 | 99.7 | 99.8 | 99.9 | 99.9 | 100 | 100 | 100 |

**Figure 15. Round Trip Time Cumulative Distribution (300 Nodes, 20-Byte Payload)**

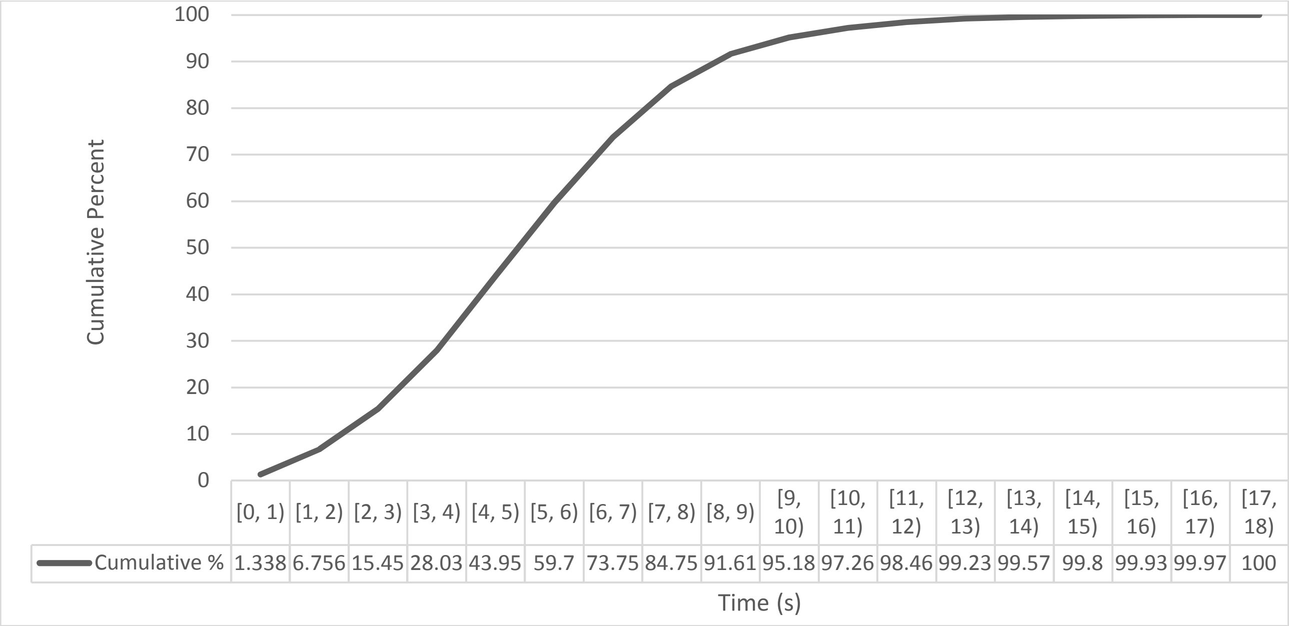
### *6.2.2 Small Payload Test Results*

Figure 16 shows that the largest percentage of responses (15.92%) was received between 4 and 5 seconds. Notice the entire graph has been shifted to the left and flattened, with a larger number of responses received under 4 seconds compared to the large payload scenario. Figure 17 shows that over 90% of the responses came in under 9 seconds.



| | [0, 1) | [1, 2) | [2, 3) | [3, 4) | [4, 5) | [5, 6) | [6, 7) | [7, 8) | [8, 9) | [9, 10) | [10, 11) | [11, 12) | [12, 13) | [13, 14) | [14, 15) | [15, 16) | [16, 17) | [17, 18) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Percent | 1.338 | 5.418 | 8.696 | 12.58 | 15.92 | 15.75 | 14.05 | 11 | 6.856 | 3.579 | 2.074 | 1.204 | 0.769 | 0.334 | 0.234 | 0.134 | 0.033 | 0.033 |

**Figure 16. Round Trip Time Distribution (300 nodes, 10-Byte Payload)**



| | [0, 1) | [1, 2) | [2, 3) | [3, 4) | [4, 5) | [5, 6) | [6, 7) | [7, 8) | [8, 9) | [9, 10) | [10, 11) | [11, 12) | [12, 13) | [13, 14) | [14, 15) | [15, 16) | [16, 17) | [17, 18) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cumulative % | 1.338 | 6.756 | 15.45 | 28.03 | 43.95 | 59.7 | 73.75 | 84.75 | 91.61 | 95.18 | 97.26 | 98.46 | 99.23 | 99.57 | 99.8 | 99.93 | 99.97 | 100 |

**Figure 17. Cumulative Round Trip Time Distribution (300 nodes, 10-Byte Payload)**