2020

Evaluation of the Robot Operating System 2 in Lossy Unmanned Networks

Thulasiraman, P.; Chen, Z.; Allen, B.; Bingham, B.

IEEE

P. Thulasiraman, Z. Chen, B. Allen and B. Bingham, Evaluation of the Robot Operating System 2 in Lossy Unmanned Networks, in Proc. Of IEEE International Systems Communications Conference (SYSCON) 2020, April 2020.
http://hdl.handle.net/10945/69142

Evaluation of the Robot Operating System 2 in Lossy Unmanned Networks

P. Thulasiraman, Z. Chen, B. Allen and B. Bingham
Naval Postgraduate School, Monterey, CA, USA
pthulas1@nps.edu, zhaolin.chen.sn@nps.edu, bdallen@nps.edu, bbingham@nps.edu

***Abstract*— The Robot Operating System 2 (ROS 2) is an open source middleware used for robotic applications. ROS 2 provides extensive security enhancements and quality of service (QoS) profiles not available in ROS 1. This paper studies the performance of ROS 2 in a small network of nodes, similar to how a group of unmanned assets would operate. Specifically, we analyze ROS 2 under varying QoS and security constraints in a wireless, lossy environment. This is the first work to comprehensively study ROS 2 network performance using QoS and security classification as a function of network scale in an environment that uses Wi-Fi communications. We custom build a simulation architecture that integrates ROS 2 with NS-3, an open source network simulator. Network performance metrics include latency and message drop rate. We show that enabling security results in a higher message drop rate across all QoS profiles. We also show that scaling the network to more nodes results in various consequences with the use of different QoS settings, including an increase in the average latency of messages. We also highlight some of the limitations there were observed with NS-3 and ROS 2.**

***Keywords*-unmanned systems, security, robotic systems, ROS**

I. Introduction

Advances in the technology of unmanned systems (UxS) have enabled their use in a growing variety of military applications. In [1], the authors explain how UxS are seen as game changers for the military in terms of intelligence, surveillance, and reconnaisance. Yet, the introduction of UxS technology for both military and civilian applications has brought with it its own set of challenges. The Unmanned System Integrated Roadmap (2017-2042) released by the U.S. Department of Defense (DoD), listed interoperability and network security as critical UxS needs [2]. Interoperability is described as allowing for interactions between systems and allowing for information to be transmitted in a timely fashion between different users. A common or open architecture is seen as a key enabler for interoperability. Network security is described as being vital to protect the integrity, availability, and confidentiality of information flow between UxS assets.

A. Robot Operating System as a Common Framework

One of the difficulties in the development of any new robotics or UxS program lies in the amount of resources required to establish the software infrastructure. Code has to be written to interface and drive the hardware within the system being developed. This means that across multiple programs, the software infrastructure has to be re-developed rather than be reused. The use of a common software infrastructure would mitigate this wastage of resources, as well as any interoperability issues.

The Robot Operating System (ROS), managed and developed by Open Robotics, is a framework that provides the software infrastructure on which others can build their UxS. As a framework, ROS provides a set of tools and libraries that simplify the task of creating a new robot. The second version of ROS, ROS 2, was developed to address the shortcomings of ROS 1 that were identified by industry and academia. Deficiencies of ROS 1 include its dependence on a central node, which is seen as single point of failure. In addition, ROS 1 lacks fundamental communications security protocols. ROS 2 was first released in 2015, while the first version with long term support (LTS) was released in June 2019.

B. Research Motivations and Contributions

ROS 2 is a new technology that has not been tested in Naval use cases. In order for the Navy to transition to ROS 2, the technology must be assessed in terms of its network performance, particularly in a lossy, wireless environment. We use simulation as a testing tool in order to rapidly evaluate performance. NS-3 is used as the simulation platform. Our aim is to demonstrate ROS 2 performance when the following parameters are adjusted: 1) quality of service (QoS) settings, 2) security settings, and 3) node scalability. The contributions of this paper are:

- Development of a simulation framework and infrastructure that integrates NS-3 and ROS 2, allowing for the network performance of multiple ROS 2 nodes to be evaluated without the need for multiple hardware to host the ROS 2 nodes.
- Simulation and evaluation of ROS 2 network performance under varying QoS profiles and security settings. Latency and message loss rate are examined.
- Simulation and evaluation of the network performance when the network is scaled to more ROS 2 nodes.

To the best of our knowledge, this is the first work that comprehensively studies ROS 2 network performance using QoS and security classifications as a function of scale in a lossy, wireless environment.

II. Background

ROS 2 addresses many of the shortcomings of ROS 1. One significant change is the use of the Data Distribution

978-1-7281-5365-0/20/$31.00 ©2020 IEEE

Service (DDS) for communication between nodes in ROS 2. DDS is a middleware framework to address the need for real-time data exchange by various applications. As part of the framework, messages are exchanged between nodes using the Real-Time Publish-Subscribe (RTPS) protocol. A publish-subscribe model is when a node (publisher) sends out a message on a given topic. A topic is a name that is used to identify the content of the message (i.e., in UxS applications, topics can include video content, heartbeat messages etc.). A node that is interested in a certain kind of data (subscriber) subscribes to that particular topic.

The implementation of QoS and security settings for ROS 2 is handled within the DDS application. All ROS code is agnostic to the DDS implementation, while all DDS code is agnostic to the ROS code, with the intra-process application programming interface (API) handling the interface between the two. Figure 1 illustrates how the ROS application layer works with the DDS middleware.

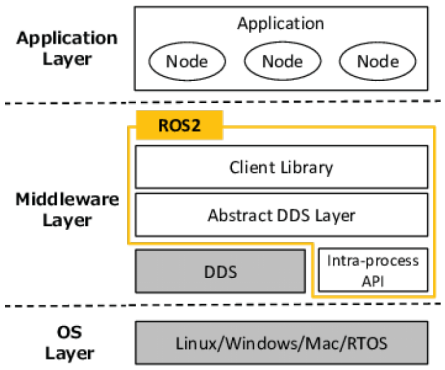

Fig. 1. ROS 2 Architecture and DDS. Adapted from [3]

ROS 2 currently supports three different DDS vendors: RTI, eProsima, and ADLINK. The different DDS vendors are expected to be compatible, as they are implementations of the same DDS framework. As such, each node in a network could be using a different DDS vendor and still be able to communicate with the others.

A. ROS 2 Quality of Service Settings

DDS allows for different QoS policies, providing the user with control over the behavior of the network. These policies address four main aspects of network performance: Real-time Delivery, Bandwidth, Redundancy, and Persistence. Although DDS supports a multitude of QoS policies, as of the ROS 2 Dashing Diademata release, ROS 2 only supports four different policies. These policies are: History, Depth, Reliability and Durability. This paper focuses on these four policies. Table I provides a description of each QoS policy supported by ROS 2.

TABLE I
DESCRIPTION OF ROS 2 QOS POLICIES

QoS Policy	Description
History	There are two settings, "Keep last" and "Keep all." History serves to configure the number of messages that the Publisher or Subscriber will keep in its cache.
Depth	Size of the queue if "Keep last" is configured as the history setting.
Reliability	There are two settings, "Reliable" and "Best effort." The Reliable setting helps ensure that all messages are delivered. Best effort attempts to send each message only once.
Durability	This policy determines whether the Publisher sends past messages to a newly joined Subscriber.

B. Secure ROS 2

DDS security features are made available for use with ROS 2 through a set of tools named Secure ROS 2 (SROS 2). DDS-Security is a set of specifications that expands on the original DDS and includes a set of Service Plugin Interfaces (SPI). SPIs implement the security model as defined or required by the user [4]. As of the Dashing Diademata version, ROS 2 makes use of only three SPIs. The three SPIs are:

- Authentication: Verification of the identity of the Publisher/Subscriber nodes.
- Access Control: Enforces which topics the authenticated nodes can publish or subscribe to.
- Cryptography: Implementation of cryptographic operations. DDS has separate SPIs that perform encryption, signing as well as hashing.

SROS 2 currently does not allow the user to define the SPIs used by the DDS. Therefore, there is only one global setting that either applies security to all nodes in the network or applies no security to any nodes (either security is turned off or turned on).

III. EXPERIMENTAL DESIGN AND SETUP

The intent of the simulations is to evaluate the network performance of ROS 2 with different QoS and security settings. Different simulations are performed with a varying number of ROS 2 nodes in order to evaluate the impact of network performance when the number of ROS 2 nodes is scaled up.

Our simulations utilize NS-3 to simulate a lossy, wireless network between the ROS 2 nodes. NS-3 is a discrete-event network simulator that supports the use of different simulation models, allowing it to be used as a real-time network emulator [5].

Our simulation architecture makes use of network namespaces to virtualize the network stack. Each network namespace creates its own network stack for processes within each unique network namespace, including its own network interfaces. For our simulations, a Wi-Fi network interface is created for each individual network namespace. Each ROS 2 node is then executed within its own network namespace, with NS-3 simulating a Wi-Fi network connecting each ROS 2 node. Figure 2 depicts how five ROS 2 nodes within their own network namespace communicate with each other via the simulated NS-3 Wi-Fi.

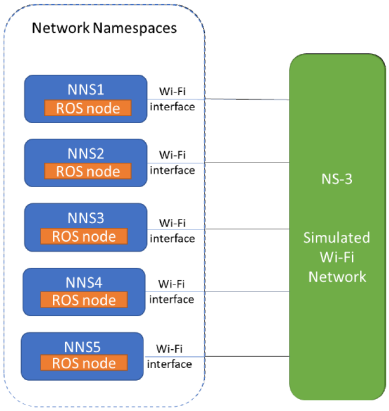

Fig. 2. Simulation architecture showing simulation of five ROS 2 nodes using network namespaces and NS-3

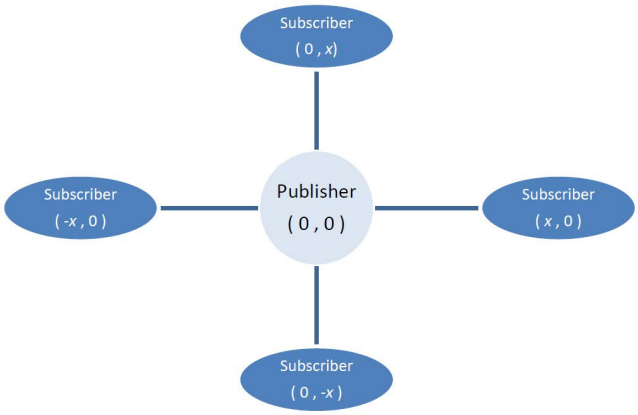

Fig. 3. Top down view of the position of Subscriber nodes relative to the Publisher node

A. NS-3 Settings

NS-3 allows for simulations of different Wi-Fi models. The models used as well as the settings such as antenna strength and throughput can be changed as required. For our simulations, we used the following settings:

- Wi-Fi Standard: 802.11a
- Type of Network: Ad Hoc
- Data mode: Constant rate OFDM 54 Mbps
- Mobility Model: Constant Position

We use the default log distance propagation loss model (default exponent 3) that is provided in NS-3. In NS-3, nodes are positioned using a 2D Cartesian coordinate system. Using the constant position model, the NS-3 simulator starts with nodes at a distance x from a central node. Figure 3 depicts how four ROS 2 Subscriber nodes are positioned around the central node, which contains the ROS 2 Publisher node. Nodes are connected through Wi-Fi channels. The simulation is run for two minutes, during which the data for the network performance is collected. After collection of the required data, the simulation is re-started with a new distance. Through multiple iterations of this process, network performance at different distances is measured.

For our experiments, we made use of eProsima Fast RTPS, an open-source DDS implementation.

B. QoS Policies in ROS 2

ROS 2 defines three profiles: Default, Sensors and Parameters. Each profile is defined with the four policies described in Section II-A. Table II lists the details of the QoS policies and QoS profiles tested in our simulations.

The QoS policies of Reliability, History, Depth and Durability are used together to determine the overall reliability with which messages are sent between nodes. These policies affect the reliability of delivery of messages sent from Publisher to Subscriber nodes, especially in a lossy network.

As was stated in Table I, within the Reliability policy, there are two sub-policies: RELIABLE and BEST_EFFORT.

TABLE II

QOS PROFILES AND POLICIES USED FOR THE SIMULATIONS

QoS profile	History	Depth	Reliability	Durability
Default	KEEP_LAST	10	RELIABLE	VOLATILE
Sensors	KEEP_LAST	5	BEST_EFFORT	VOLATILE
Parameters	KEEP_ALL	1000	RELIABLE	VOLATILE

If the RELIABLE policy is used, the Publisher waits for an acknowledgment from the Subscriber after each message. If an acknowledgment is not received, the original message is re-transmitted by the Publisher until the Subscriber receives the message. If a BEST_EFFORT policy is used, the Publisher does not listen for any acknowledgment message, and transmits new messages as required.

History controls whether messages are stored in the cache. Within the History policy, there are two sub-policies: KEEP_LAST and KEEP_ALL. If a KEEP_ALL policy is used, all messages transmitted by a node are stored in the cache of the data writer, up to the system resource limit. If a KEEP_LAST policy is used, then the DEPTH parameter is used to determine the number of messages that are kept in the cache.

Durability policies control what to do with nodes that join the network late. Durability has two subpolicies: TRANSIENT_LOCAL and VOLATILE. If the TRANSIENT_LOCAL policy is used, all messages stored in the cache are sent over to the Subscriber. If the VOLATILE policy is used, data is not stored in the cache, and is not sent to any nodes that join the network late. In the case of our simulations, the Durability policies are not changed between simulations as all nodes are initialized and join the network together.

The Default profile provides the default QoS settings for publishers and subscribers. By default, publishers and

subscribers use a RELIABLE connection in ROS 2, have VOLATILE durability, and "KEEP_LAST" history (see Table II). For sensor data, in most cases it is more important to receive readings in a timely fashion, rather than ensuring that all of them arrive. UxS operators want the latest data samples as soon as they are captured, at the expense of losing some data packets. For that reason the Sensor data profile uses best effort reliability and a smaller queue depth. The Parameters profile are for non-sensor data services and use a much larger queue depth so that requests do not get lost.

C. Security Settings

SROS 2 uses Authentication, Access Control and Cryptography.

1) Authentication: The authentication plugin allows for mutual authentication between discovered nodes. After initial discovery, authentication must be completed before information can be exchanged between nodes. A trusted Certificate Authority (CA) is used as part of the authentication process. The Elliptic Curve Digital Signature Algorithm is used to generate the public key [6]. The Elliptic Curve Diffie-Hellman Key Agreement Method is then used to derive a shared key between both nodes.

2) Access Control: After a node is authenticated, validation of its permissions is performed. Access control expresses the type of access that is granted to each node for each specific topic.

3) Cryptography: The cryptography plugin used by RTPS provides authenticated encryption using Advanced Encryption Standard in Galois Counter Mode [6]. Message authentication is provided through message authentication codes (MACs) using Galois MAC.

D. System Setup

Simulations were performed on a single computer. The computer used for the simulations had the following hardware:

- Processor: Intel(R) Core (TM) i7-8700K CPU @ 4.60GHz (6 cores)
- Memory: 32 GB DDR4
- OS: Ubuntu 18.04
- ROS 2.0 version: Dashing Diademata Patch Release 1
- NS-3 version: NS-3.29

IV. Simulation and Analysis of Results

In order to observe whether NS-3 correctly simulates packet loss in a wireless network, we performed a simulation using NS-3 without ROS 2. Packet sizes of 60 bytes were sent between two nodes in NS-3. Figure 4 illustrates the average percentage of packet loss at different distances for an 802.11a ad-hoc network. The results indicate that packet loss begins to increase at the 25 meter mark.

Next, sets of simulations were carried out, each with a specific QoS profile and security setting. Each set consisted of a series of simulation runs in which the nodes were placed at different distances from one another. With different distances, the wireless network has a different simulated packet drop rate, which affects the message transmissions in ROS 2. Due to space, we show a subset of results. Additional results and wireshark captures can be found in [7].

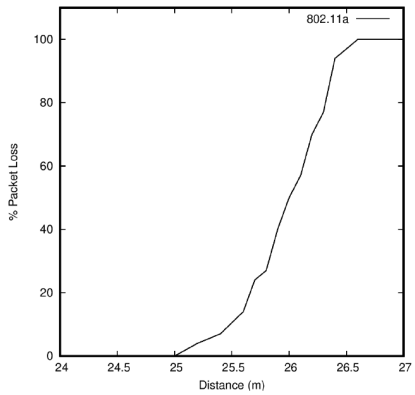

Fig. 4. Rate of packet loss versus distance (m) as simulated in NS-3 without ROS 2. Packet loss begins at the 25m mark.

For each simulation run, ROS 2 messages were published by a single Publisher node at a frequency of 2 Hz. Each simulation ran until either 200 messages were received by the Subscriber nodes or a time-out error was reached. Each message was 45 bytes long and consisted of a generic "Hello World!" string. A counter and time stamp were also appended to the message. For each run, the rate of message loss and the average latency incurred by each message was recorded.

The rate of message loss is defined as the ratio of messages received by the Subscriber to the total messages that the Subscriber was supposed to receive. Each published message is stamped with a counter indicating the index of the message. Based on the counter in the message, the Subscriber node determines whether any message was not properly received. The Subscriber then compares the counter of the message received to that of the last message received. If the counter is not in running sequence, the Subscriber is able to determine the number of messages that were lost.

Latency is defined as the delay between the time that a specific message is published by a Publisher and the time that it is read by the Subscriber. Each message includes the system time of when the message was prepared.When the Subscriber receives a message, it compares the current system time with the system time included in the message. In this way, the measured latency includes the delay incurred by ROS middleware to translate the message, the delay incurred by the DDS middleware to process each message packet, as well as the actual network propagation delay.

A. Network Performance With Different QoS Settings

We used the ROS 2 QoS profiles shown in Table II as a baseline to measure network performance. We ran additional simulations with QoS policies set at custom values in order to measure the specific impact of changing the settings of that

specific QoS policy. Simulations were initially performed with one Publisher and one Subscriber node.

1) Message Loss Rate: The message loss rates for the three QoS profiles described in Table II are shown in Figure 5. With a Sensor profile, the message loss rate starts to increase gradually as the distance between nodes is increased. This is to be expected as the Sensor profile utilizes a BEST_EFFORT policy. As the distance between nodes increases, the rate of packet loss increases, similar to what is shown in Figure 4. Thus, the chance of a message being dropped increases as well.

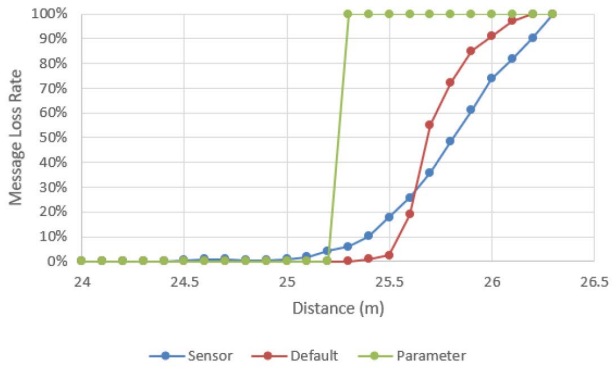

Fig. 5. Message Loss Rate versus Distance for QoS profiles shown in Table II

The Default profile has a Depth = 10 (see Table II). The Default profile produced a lower message drop rate until 25.7m, as compared to using the Sensor profile. The Default profile utilizes a RELIABLE policy, which resends the messages if the Publisher does not receive an acknowledgment message from the Subscriber.

Since the Publisher retransmits all the missing messages in a single packet, the message loss rate becomes larger than that experienced by the Sensor profile when the distance between nodes is larger than 25.7m.

The Parameter profile has a History policy of KEEP_ALL and a Depth policy of 1000. As such, the packets that are sent are very large, as it would include both the current message as well as all past messages. In this situation, either the Subscriber receives all the messages or all the messages are dropped. It performs slightly better than the Sensor profile for a small set of distances (24.8m-25.3m), as the Subscriber can still retrieve any dropped messages from subsequent message deliveries. When the network gets more lossy with increased distance, however, none of the messages manage to be delivered; the packet drop rate for such a large packet size was very high.

In our next experiment, the Depth of the Default profile is changed to 1, and the results of the simulations are compared to that of the original Default profile (Depth = 10). The results are shown in Figure 6. With Depth = 1, the network performs similarly to that of the Sensor profile. Since the packet sizes are smaller due to the smaller depth size, even with greater distance, more packets are delivered as compared to the profile with a large Depth (Depth = 10).

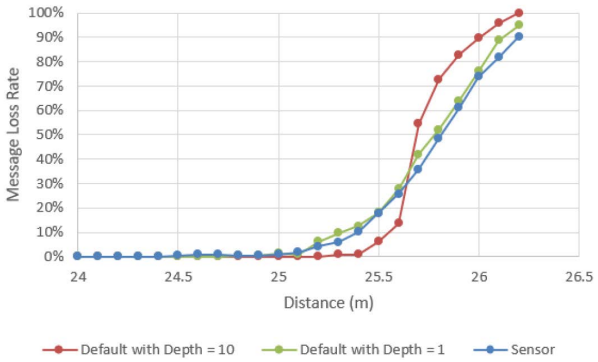

Fig. 6. Comparison of message loss rate when Depth =1 and Depth = 10 for the Default profile and the Sensor profile with Depth = 5

2) Latency: The simulation results that measure the impact of the QoS profiles on latency are plotted for 24m to 26m distances in Figure 7. As seen in Figure 4, given a packet size of 60 bytes, packet loss occurs starting at 25 m. As such, in order to review the impact that the QoS profiles have on a lossless network, we evaluate the latency from 24m to 26m for the different QoS profiles.

As can be seen in Figure 7, as the distance increases, the network experiences increasing packet loss. Accordingly, the latency for messages with the Default profile increases significantly. This is due to the time incurred from the retransmission of messages not received by the Subscriber node. This can take multiple retransmissions in a lossy environment, with the Subscriber node receiving the message much later than when it was originally sent by the Publisher node.

There is no significant trend for using the Parameters profile, as all messages are dropped from distances 25.2 m and beyond (as was shown in Figure 5).

The Sensor profile experiences very little latency in this instance and outperforms the Default and Parameters profile. This indicates that the Sensors profile is best suited for low latency transmission of data over longer distances.

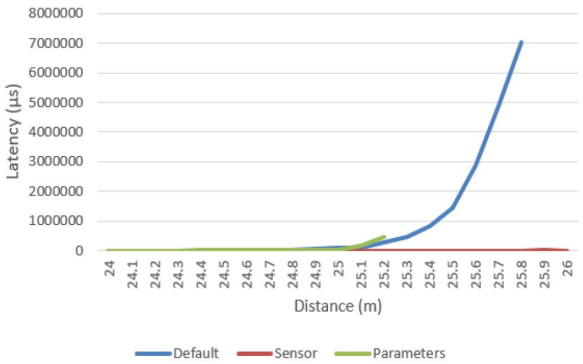

Fig. 7. Latency of messages with different QoS profiles in a lossy network

B. Results with Security On and Off

We next study the performance of ROS 2 when security is turned off or on. The results shown in Figures 8-10 depict the message loss rates for the QoS profiles from Table II with security turned on and off. Simulations were performed with one Publisher and one Subscriber node. It can be seen that for all profiles (Sensor, Default and Parameters), messages are dropped at a shorter distance when security is turned on as compared to when security is turned off.

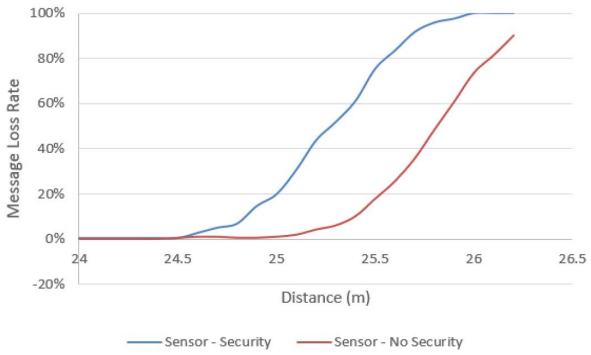

Fig. 8. Message loss rate with security turned on and off using the Sensor profile

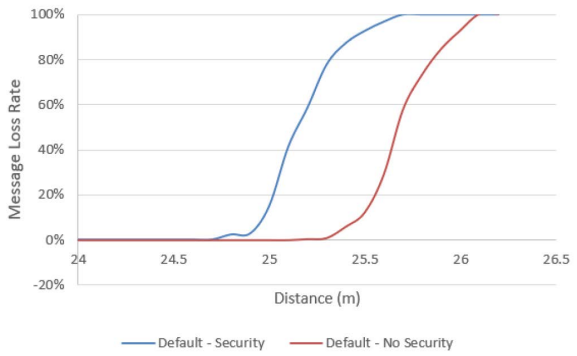

Fig. 9. Message loss rate with security turned on and off using the Default profile

Figure 11 and Figure 12 show the impact on latency with security turned on and off for the Sensor profile and the Default profile in a lossy network, respectively. Latency increases for both profiles when security is turned on. In Figure 11, it can be seen that from 24 m, the latency starts to increase immediately with security turned on, as compared to 24.5 m with security turned off. Using the Default profile, latency starts to increase at 24.7 m with security turned on as compared to 25.2 m with security turned off, as shown in Figure 12. In addition, the latency incurred by the Sensor profile is significantly less. The maximum latency for the Sensor profile at 25.5 m with security turned on is 4 ms. The maximum latency for the Default profile at 25.5 m with security turned on is 10 secs.

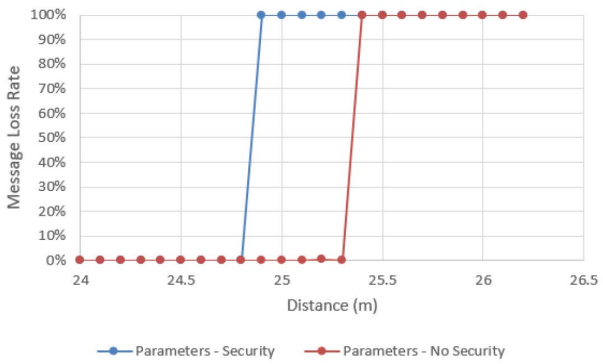

Fig. 10. Message loss rate with security turned on and off using the Parameters profile

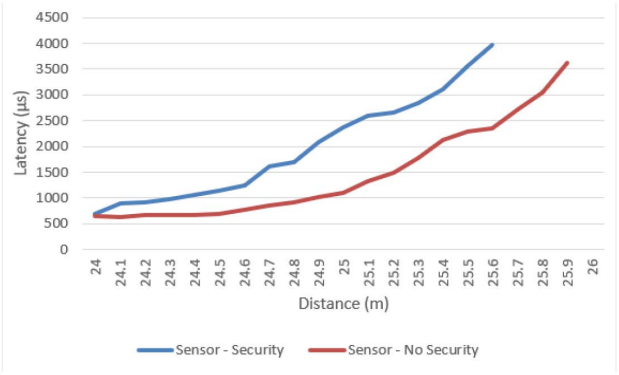

Fig. 11. Latency of messages for the Sensor profile with security turned on and off (beyond 24 m)

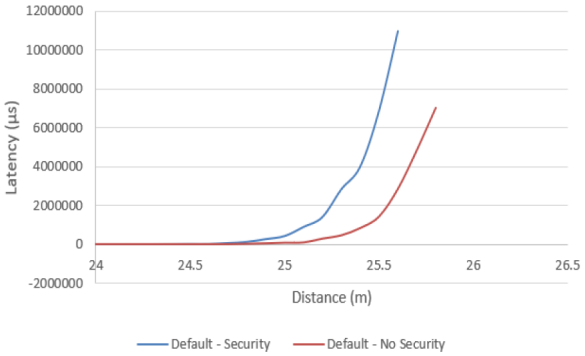

Fig. 12. Latency of messages for the Default profile with security turned on and off (beyond 24 m)

C. Impact of Network Scale on Performance

In this section, we compare the network performance of ROS 2 when the number of nodes in our simulations is increased to five. We use the same profile combinations shown in Table II.

1) Message Drop Rate: Figure 13 compares the message drop rate using the Sensor profile when the network has two nodes (one Publisher and one Subscriber) and five nodes (one Publisher and four Subscribers). The results show that as long as the network has sufficient bandwidth, the message drop rate is not affected when scaling up to more nodes when using the Sensor profile. Nevertheless, there is a difference when the Default profile is used. Figure 14 shows that five nodes have a lower message drop rate with the Default profile than when the network has only two nodes with the Default profile.

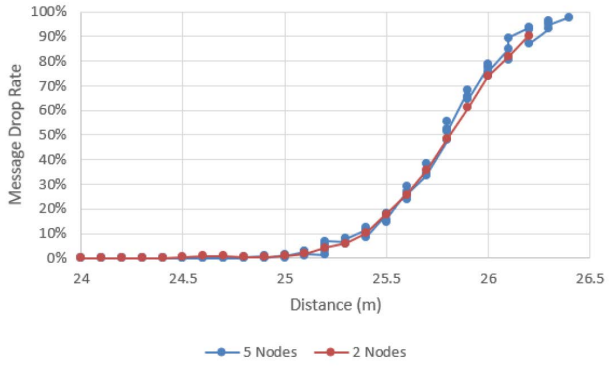

Fig. 13. Message drop rates comparing two nodes and five nodes with the Sensor profile

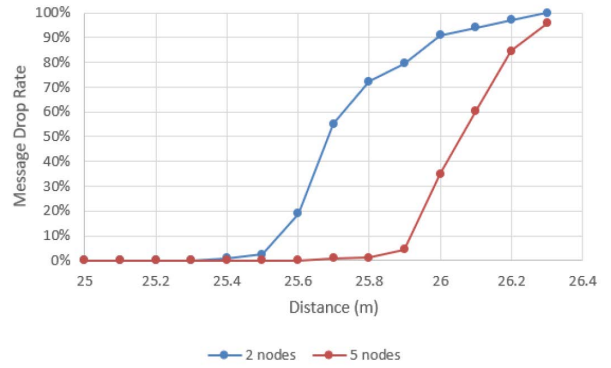

Fig. 14. Message drop rates comparing two nodes and five nodes with the Default profile

2) Latency: ROS 2 transmits each message sequentially to each individual node. This means that for a single message with four Subscribers, the same message is transmitted four times. This is depicted in the Wireshark capture shown in Figure 15. These sequential transmissions result in a significant increase in latency as more nodes are added to the network. Table III shows the latency experienced by each node in a five-node network that has one Publisher and four Subscribers as compared to the latency in a two-node network with only one Subscriber.

TABLE III
LATENCY OF MESSAGES FOR EACH NODE IN A ONE SUBSCRIBER NETWORK AND A FOUR SUBSCRIBER NETWORK (NANOSECONDS)

QoS profile	Two-node network	Five-node network			
	One Subscriber	Node 1	Node 2	Node 3	Node 4
Sensor	614 292	684 854	10 645 701	20 617 056	30 396 675
Default	639 480	712 332	10 706 034	20 725 019	30 793 318
Parameters	647 597	716 007	10 724 211	20 734 675	31 384 246

```
726 3.389198984   10.0.0.1   10.0.0.4   RTPS   146 INFO_TS, DATA
727 3.389206363   10.0.0.1   10.0.0.7   RTPS   146 INFO_TS, DATA
728 3.389210036   10.0.0.1   10.0.0.10  RTPS   146 INFO_TS, DATA
729 3.389213151   10.0.0.1   10.0.0.13  RTPS   146 INFO_TS, DATA
```

Fig. 15. Wireshark capture of messages sent by Publisher that show the sequential transmission of four messages

D. Summary of Findings

The simulation results demonstrated that the integration of NS-3 as a simulation platform for ROS 2 is useful and an effective way to rapidly test network performance. The different QoS profiles affect the network performance in distinctive ways. The results from the various simulations demonstrated the trade-offs in network performance when using different QoS profiles. The Sensor profile delivers messages as quickly as possible, with a minimal impact on latency. It also outperforms the Default profile in terms of message drop rate in a network of high wireless loss. The Parameter profile has a large depth to cater to situations where the Subscriber node is repeatedly unable to reach the Publisher node. This results in a larger latency compared to the other profiles. In addition, the percentage of messages delivered is either 100% or 0% and would be not be suitable for all occasions.

There was also significant overhead when security settings were turned on. Using the Default profile, it incurred a 60% increase in latency, with the overhead likely to be much higher if performed using a slower processor. The overhead from having security turned on also meant a higher message drop rate across all QoS profiles. In comparison, the Sensor profile incurred a 35% increase in latency. We also observed that the increase in latency when security is enabled for the Default profile is 2500 times higher than for the Sensor profile with security enabled for the same range of distances. This is a significant liability for the Default profile.

Scaling up the number of nodes in the network to five nodes from two nodes resulted in varying consequences with the use of different QoS settings. Of significance is the increase in latency and message drop rate when the Default profile is used. It is likely that in a swarm network with 30–50 unmanned assets, a Reliable QoS policy cannot be used as the latency incurred would be too high.

Message drop rate was minimal with the Sensor profile when the network was scaled to 5 nodes. These results show that the Sensor profile performs better than the other two profiles, even in the case where security is enabled. Intuitively, this makes sense since the Sensor profile uses the least restrictive policies for data transmission. The Parameters profile performed quite poorly over all simulations, which limits its use in practical scenarios.

Through our simulations, we conclude that the security settings and the network size has a significant impact on network performance, more so than specific QoS settings. For system designers this suggests that where and how security is implemented in the system will have a large influence on performance, especially compared to the impact of QoS profiles. We found that scaling beyond five nodes (1 publisher and 4 subscriber) induces significant delay that prohibited messages from being delivered to any destination. We believe these delays are due to 1) limitations on NS-3 co-operating with the underlying DDS communications protocol of ROS 2 and 2) the sequential nature of message transmission in DDS to each node. We also believe that the ad hoc Wi-Fi mode of NS-3 and the DDS protocol both try to manage dynamic network configuration, resulting in network traffic that burdens any existing load. This greatly impacts the performance of ROS 2 beyond five nodes.

V. Conclusion

In this paper, we proposed and validated a simulation architecture that used NS-3 to study the network performance of ROS 2 using varying QoS profiles and security settings. We found that security had a significant impact on all QoS profiles in terms of message latency and message drop rate. The number of nodes in the network also produced similar increases in latency and message drop rate. We found that the Sensor profile outperformed the Default and Parameter profiles over all simulations and instances. However, our scaling efforts were limited by the underlying affects of using NS-3 and DDS together. We believe further experimentation with different emulators (i.e., Mininet Wi-Fi) as well as other DDS developers (i.e., RTI) will provide a clearer picture of ROS 2 performance in larger networks.

Acknowledgements

This work was funded and sponsored by the Office of Naval Research via the Consortium for Robotics and Unmanned Systems Education and Research (CRUSER) at NPS.

References

[1] S. Brimley, B. Fitzgerald, K. Sayler, and P.W. Singer, "Game changers: Disruptive technology and u.s. defense strategy," 2013, http://www.cnas.org/files/documents/publications/CNAS_Gamechangers_BrimleyFitzGeraldSayler_0.pdf.

[2] Office of the Secretary of Defense, "Unmanned systems integrated roadmap fy 2017-2042," 2017, https://www.defensedaily.com/wp-content/uploads/post_attachment/206477.pdf.

[3] Y. Maruyama, S. Kato, and T. Azumi, "Exploring the performance of ros2," in *Proc. of ACM International Conference on Embedded Software*, 2016.

[4] Object Management Group, "Dds security version 1.1," 2018, https://www.omg.org/spec/DDS-SECURITY/1.1.

[5] "Ns-3," https://www.nsnam.org/doxygen/tap-wifi-virtual-machine_8cc_source.html.

[6] "Security-fast rtps 1.9.0 documentation," https://fast-rtps.docs.eprosima.com/en/latest/security.html.

[7] Zhaolin Chen, "Performance analysis of ros 2 networks using variable quality of service and security constraints for autonomous systems," M.S. thesis, Naval Postgraduate School, September 2019.