Towards friendly force tracking with MQTT over LoRa

Frank T. Johnsen[a], Trude H. Bloebaum[a], and Philip Ø. Puente[b]

[a]Norwegian Defence Research Establishment (FFI), Kjeller, Norway
[b]Norwegian University of Science and Technology (NTNU), Trondheim, Norway

ABSTRACT

With the steady growth of the Internet of Things (IoT) in the civilian commercial sector, the question arises as to whether such IoT concepts can also be used for defense purposes. Military needs, especially those that depend on tactical communications, can be regarded as more challenging to solve than the needs that are focused on in the civilian sector. Therefore, one can expect that much of the work done in the civilian sector cannot necessarily be used directly for military purposes and that targeted research and development in this field will be required to enable effective deployment. This paper explores using Raspberry Pi 3 and the Long-range Radio (LoRa) protocol as an IoT platform for friendly force tracking with the light-weight, industry standard Message Queuing Telemetry Transport (MQTT) publish/subscribe protocol.

Keywords: LoRa, friendly force tracking, MQTT, publish/subscribe

Topic 3: Battlefields of the Future and the Internet of Intelligent Things
Paper ID 14

Point of contact

Frank T. Johnsen
Norwegian Defence Research Establishment (FFI)
P.O. Box 25, 2027 Kjeller, Norway
E-mail: Frank-Trethan.Johnsen@ffi.no

1. INTRODUCTION

For military forces operating at the tactical level, shared situational awareness is a requirement for efficient decision making. Simply put, *situational awareness* is *knowing what is going on around you*.[1] With new, emerging and ever evolving technology, new and faster ways for information sharing are made possible. Having reliable and up-to-date information about the operating environment, including the position and status of friendly forces and other assets available when and where needed, is a key enabler for achieving such shared situational awareness. This type of information can be gathered in a number of different ways, and one civilian trend which is gaining traction also for military use is the deployment of cheap sensor systems as a means to augment the information already available to military decision makers today. This civilian trend is known as the Internet of Things (IoT), can be defined as follows:[2]

> IoT describes the revolution already under way that is seeing a growing number of internet enabled devices that can network and communicate with each other and with other web-enabled gadgets. IoT refers to a state where Things (e.g., objects, environments, vehicles and clothing) will have more and more information associated with them and may have the ability to sense, communicate, network and produce new information, becoming an integral part of the Internet. A widespread Internet of Things has the potential to transform how we live in our cities, how we move, how we develop sustainably, how we age, and more.

In,[3] the authors present a number of possible applications of IoT technology for military use, such as collaborative sensing in the battlefield, logistics and supply chain management, and augmenting situational awareness with sensor information when operating in smart city environments. This latter use case has been the focus of the recently concluded IST-147, a NATO Science and Technology Organization (STO) Research Task Group (RTG) aptly named "Military application of IoT".[4] A follow-on group called IST-176 "Federated Interoperability of Military C2 and IoT Systems" was recently started (May 2019). The ongoing work in this paper is relevant to IST-176, and future work will be aligned with collaborative efforts there.

Work by Suri et al.[3] further highlights six technical barriers that may hinder military adoption of IoT, including issues related to network utilization and interoperability. Network resources are often scarce in tactical military scenarios, and there is a need to prioritize which types of information are allowed to consume network resources in the radio based network systems that the forces use for mission critical tasks such as coordination and position reporting. IoT can possibly have a large footprint by providing huge amounts of data (so-called big data). Because of this, one promising solution is to use decentralized analysis of data, for instance through so-called *fog computing*[5] nodes, to limit the amount of data that needs to be transmitted. Another option is to use an alternative long-range communications carrier for sensor data, so that this information does not add further load to the tactical radio systems tactical forces use as their main communications carrier.

In this paper we investigate using Raspberry Pi 3, which is often employed as a fog computing platform, since it is a readily available, low-cost commercial off-the-shelf (COTS) product. Further, we explore using the Long-range Radio (LoRa) protocol, which is a relatively new, promising development for civilian IoT communications. It has also been proposed for military applications, as a carrier for data in the Internet of Battlefield Things (IOBT).[6] For interoperability, we use the lightweight industry standard Message Queuing Telemetry Transport (MQTT) publish/subscribe protocol.

The remainder of this paper is organized as follows: In Section 2 we cover the technologies we used in our experiment. Sections 3 and 4 present our setup and experiment results, respectively. Section 5 summarizes the paper. Finally, Section 6 outlines plans for future work.

2. TECHNOLOGY OVERVIEW

With the rising popularity of IoT comes a wider and more varied selection of technologies. It is therefore necessary to research what technology can be employed to solve different tasks, what their advantages are and what limitations they might have.

2.1 Raspberry Pi 3 and Raspbian Linux

The Raspberry Pi is a low-cost single-board computer developed by the Raspberry Pi Foundation.[7] It was originally intended to be used for educational purposes, but quickly became popular in several areas, such as for prototyping technical solutions, robotics, and, more recently, IoT. The Raspberry Pi is a simple device, but its power and popularity comes from the flexibility it provides. Unlike typical micro-controllers, the Raspberry allows the user to install an operating system (OS) of their choice on a microSD card, giving access to all the tools that exist for Linux platforms and high level languages. We chose to use Raspbian, a Debian-based distribution of Linux, designed for use with the Raspberry Pi. In most kits Raspbian comes pre-installed on the microSD card, therefore making it the most common OS for the Raspberry Pi. Being the default typically means that it is the most supported system for the platform, both in terms of stability and availability of third party software.

The Raspberry Pi comes with multiple programmable serial pins, where one can attach sensors or extension boards. These boards are called *hats*,[8] and further expand the capability of the device. For more information about Raspberry Pi, see.[7]

2.2 Long-range Radio (LoRa)

LoRa is a wireless communications technology owned by Semtech.[9] It was developed with focus on long range, low power and secure transmission. This makes it naturally appealing for IoT and sensor technologies, as it allows for operating on battery power and can even offer communications over longer distances that mobile networks. Also, it is of interest for defense applications due to these properties. For example,[10] has evaluated diverse LoRa equipment available in the USA (Pycom nodes, mDot/Leonardo nodes, and Multitech gateways) for use in the tactical military environment in a proof-of-concept architecture integrating IoT with military applications. LoRa has also been investigated from a security perspective, with respect to packet interception and injection.[11] The investigation (also performed in the USA) showed that it could be beneficial to consider devising additional security mechanisms on top of LoRa to mitigate the potential for data leaks.

In wardriving[*] attempts with LoRa, it has been proven that in optimal conditions the technology is able to reach up to 200 km.[12] The LoRa technology transmits over a license-free sub gigahertz frequency. The exact frequency varies depending on the region, in Europe the most commonly used frequency is 868 MHz. For the experiments in this paper, we used equipment communicating in this frequency band, thus extending the knowledge of previous works[6, 10, 11] that used the 915 MHz version available in USA. While communicating over license free bands is convenient for IoT and civilian use as one does not have to pay for a fee, it does have a few tradeoffs. Namely there are other actors working in the same bands that might cause interference. Furthermore, there are regulations that restrict how long and how often one can transmit.

LoRa is most often used in conjunction with LoRaWAN. As such the two terms often get mixed or understood as one and the same thing. This is a misconception, however, as they are both components of the low power wireless access network (LPWAN). To clarify:

- *LoRa* is the physical layer and refers to the modulation scheme (sometimes referred to as LoRa-phys), designed for long range low power transmission.
- *LoRaWAN* is the MAC layer, a communications protocol that defines how the system should be structured and what policies devices should follow when communicating.

It is possible to use LoRa without using LoRaWAN. In fact, this is how we used LoRa in the work described in this paper, showing the feasibility of such an approach.

Achieving long transmission range on high power, like AM radio, is fairly trivial, as transmission range is directly correlated with power usage. So, to get similar results on low power devices a tradeoff is required. LoRa realizes its long range by using a narrow bandwidth, which comes at the cost of lower capacity on the channel. LoRa also makes use of techniques that help to further promote battery life, range and stability of the signals, such as Spreading Factor (SF), Cyclic Redundancy Check (CRC) and variable bandwidth. For readers who are

[*] Wardriving is the act of driving around with wireless equipment to test wireless networks.

not familiar with radio and LoRa transmission, the explanation from Richard Wenner[13] is recommended for further explanation of these details.

LoRa uses a technique called Chirp Spread Spectrum (CSS) to encode information. While frequency modulated keying (used in FM radio) encodes information by alternating between two frequencies, CSS uses a rising and falling signal called an up-chirp and a down-chirp. Making use of the entirety of the broadband that is allocated, in combination with some of the other mechanisms LoRa deploys, makes it is more resistant to several common types of interference.[14] Amongst these are single channel noise, the Doppler effect and multi-path fading.

Every LoRa transmission starts with a preamble, this is usually a set of eight up-chirps followed by a synchronization signal consisting of two down-chirps. Then, the payload consisting of modulated up-chirps and finally the optional CRC.

Another mechanism LoRa makes use of is the SF, which is defined from SF7-SF12, where each level takes twice as long as the last to transmit. The reason to have an adaptive SF is to make it easier to communicate over noisy channels or extremely long distances. Somewhat analogous to talking slower when speaking in a noisy environment, a higher SF (thus longer over the air time) makes the message easier to decode.

2.3 Dragino LoRa GPS hat

Dragino is a manufacturer of chipsets and microcontroller boards with a focus on IoT devices. At the time of writing, their product portfolio includes a LoRa extension board for the Raspberry Pi and a stand-alone LoRa Gateway, among other things.[15]

The LoRa GPS HAT for the Raspberry Pi we used is based on the SX1276 and RF96 chips for LoRa communications on the 868MHz band and an L80 GPS chip for GPS data. It supports a variety of modulation schemes such as: Frequency-shift keying (FSK),[16] Gaussian frequency-shift keying (GFSK) as used in Bluetooth,[17] minimum-shift keying (MSK),[18] Gaussian minimum-shift keying (GMSK)[19] used in GSM mobile networks, LoRa and on-off keying (OOK).[20] If the standard GPS antenna is not powerful enough, external ones can be used and the hardware will switch automatically without losing signal.

This hat was intended for LoRaWAN solutions, so there are other, leaner and more low power-friendly options out there. That said, using a Raspberry Pi is convenient, especially for prototyping, as it is popular, widespread and therefore supports multiple other IoT technologies and platforms. Also, for further pursuing fog computing aspects, a somewhat capable platform like the Raspberry Pi is needed. Hence, we chose to use three of these hats from Dragino for our initial experiments with LoRa.

2.4 MQTT

Message Queuing Telemetry Transport (MQTT) is a publish/subscribe messaging protocol. The protocol is designed with lightweight low bandwidth operation in focus and is able to function even in high latency and low reliability environments. Due to these properties, MQTT is also of interest in military tactical networks,[21] where it has been shown, through work performed in context of the NATO IST-RTG 150 “NATO Core Services profiling for hybrid tactical networks”, to perform more efficiently than other comparable publish/subscribe technologies.[22,23] Our previous work targeting IoT specifically, also shows that MQTT performs very well compared to other industry standards in terms of being a light-weight protocol with little overhead.[24] Today, MQTT is managed by the OASIS Foundation,[25] and its specification can be found on their website.[26]

MQTT is a publish/subscribe protocol, meaning that there are several different roles within the system. These are:

- A *publisher*, which produces messages,
- a *subscriber*, which consumes messages, and
- the *broker*, which gathers and distributes the messages to the correct devices.

One device can be both a publisher and subscriber at the same time, allowing for two-way communications. In addition to these roles, there is a topic system that provides an effective and powerful method of distributing and receiving only the relevant data for that device. Decoupling the information producers from the information consumers is done through the introduction of a broker, which functions as an intermediary in all message exchanges. The broker takes on the role of handling subscription management, message to topics matching and message forwarding. A topic is defined by a list of strings separated by topic-level separator "/". It is natural to think of the topics and sub topics as a hierarchy, much like a file structure.

Using the example from,[21] we see how the structure of topics can help identify relevant data, by allowing us to uniquely index services (and related messages), specifically:

```
/country-Id/squad-Id/node-Id/service-type
```

Here, *country-Id* uniquely identifies the country. For example, according to NATO STANAG 1059, "NOR" is used for Norway. The *squad-Id* is an arbitrary string that uniquely identifies the squad. The *node-Id* is an arbitrary string that uniquely identifies the node (i.e., a single unit within the squad). The *service-type* is a string that uniquely identifies the type of service. For example, "location" is used as the *service-type* to access information pertaining to soldier location (e.g., latitude and longitude). In our case, we instantiate *country-Id*, *squad-Id* and *service-type* to "NOR", "BATTALION001" and location. For example, the node with id "1" will publish messages to the following topic:

```
/NOR/BATTALION001/1/location
```

This label will then identify the actual GPS location data coming from the corresponding unit, namely one of the Raspberry Pi 3 devices.

We may subscribe to each topic individually or as a group by making use of a feature called wildcards. Wildcards act as a replacement for any topic and come in two forms, "+" for a single-level wildcard and "#" for multi-level wildcard. With this powerful feature it is possible to create more complex behavior in the system.

Wildcards allow for grouping and receiving messages based on both location and nodes, this is called topic based filtering. As such, in order to receive location updates from all battalion units, clients issue the following subscription:

```
/NOR/BATTALION001/+/location
```

To deal with unreliable connections, MQTT supports multiple levels of Quality of Service (QoS). Simply put, QoS is a set of predefined policies that describe to what degree one should ensure that a message is sent. In some situations it might not be a problem if a message or two gets lost. Conversely, in other situations it may be critical that messages not only arrive, but are also not duplicated. To support different needs, QoS has three levels ranging from 0-2:

0. **At most once delivery:** No guarantees
1. **At least once delivery:** A message is always received, and duplicates are allowed
2. **Exactly once delivery:** A message is always received, no duplicates are allowed

QoS behavior is realized through standard acknowledgment exchange as is also observed in other protocols.

Since messages are not transmitted directly from publisher to subscriber, we do not have to demand that these entities are connected to the network at the same time. As such we can support asynchronous communications. This is very convenient because it allows for operation over unstable networks and to make use of sleeping devices and sensors, further reducing energy demand.

3. INFRASTRUCTURE AND SETUP

This experiment used three Raspberry Pis, each with a Dragino hat. These can act as both receivers and transmitters, thus allowing them to function as a single channel gateway. This experiment only made use of simple sending and receiving over LoRa and relaying to MQTT over TCP.

We used one Raspberry Pi as a gateway, leaving us with two that could act as transmitters (see Figure 1). The transmitters where set up to gather GPS data from the Dragino hat and then transmit it over the 868 MHz LoRa band. These transmissions were then picked up by the Raspberry Pi functioning as gateway, and published as messages through MQTT over a TCP/WiFi connection to a laptop running an instance of OKSE[27] as the MQTT broker. OKSE is a multi-protocol publish/subscribe broker that can be used to mediate between different industry standards. It can be used to integrate different publish/subscribe systems in a federation of systems, by translating between different protocols. For the sake of this paper, only the MQTT support was used.

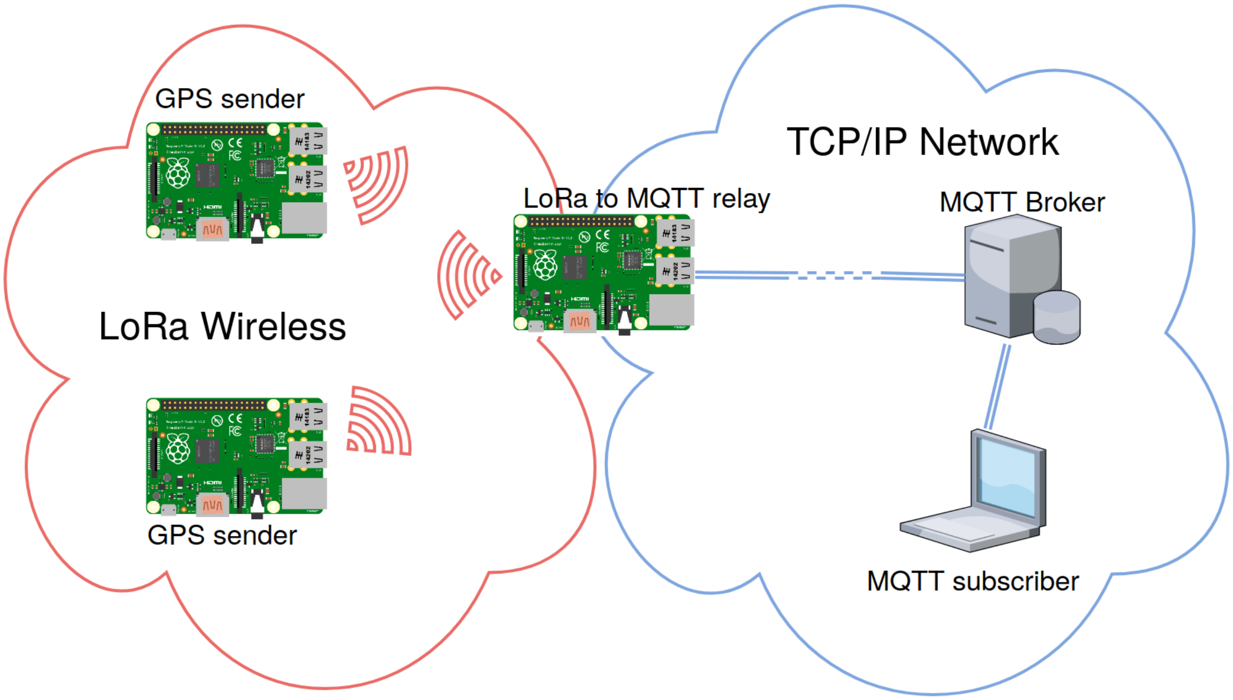

Figure 1. Testing setup of Raspberry Pi units

For the GPS we used *gpsd*,[28] a standard GPS daemon for Linux devices, and the *python-gps*[29] library, which is a Python interface for gpsd. For LoRa we used the *PyLora*[30] library and wrote a Python script to be able to send GPS data over LoRa and receive and print incoming transmissions. This enables the LoRa gateway when receiving a GPS message to either print it out to the terminal or to publish it using MQTT. For further details on this script and how we installed necessary drivers, see.[31]

4. TESTS AND RESULTS

During this experiment two types of tests were conducted: A small-scale range test, and a round trip time (RTT) test. The range tests did not give any interesting results as they worked fine within the area that was tested

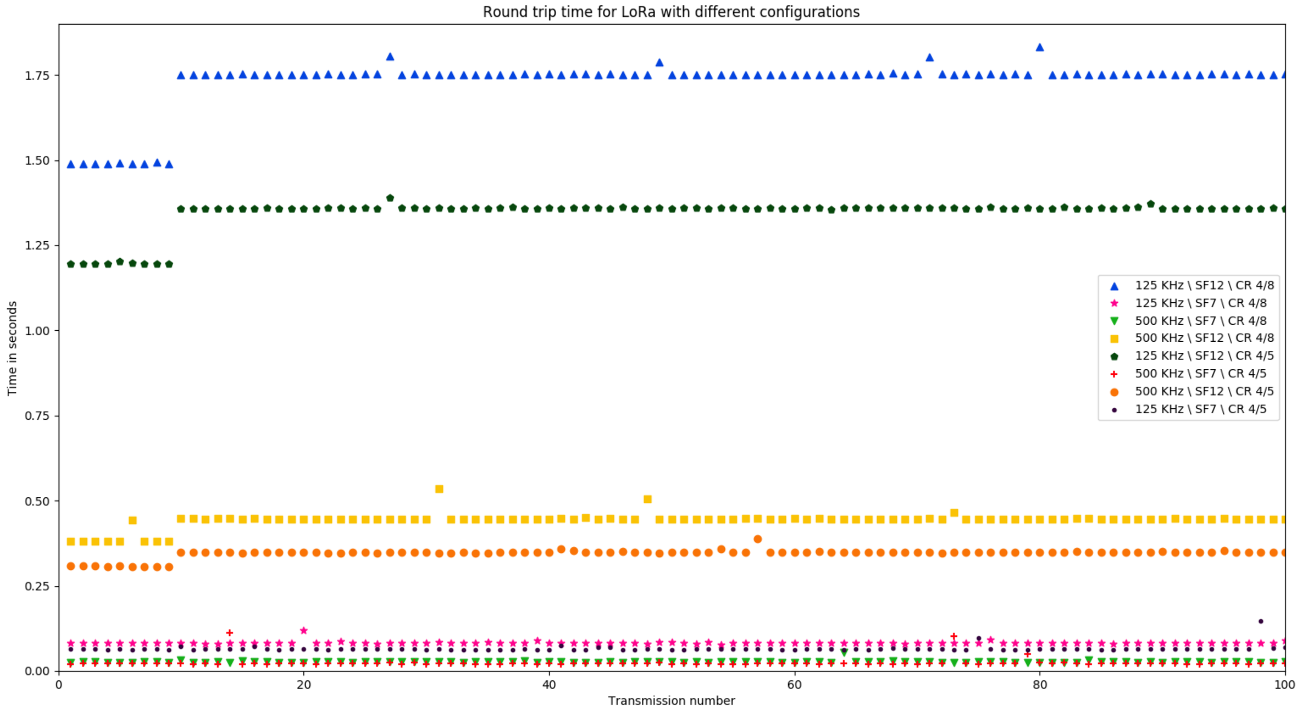

Figure 2. Round trip time Lora transmitter to MQTT subscriber

Bandwidth	**125 Khz**				**500 KHz**			
Spread factor	**SF7**		**SF12**		**SF7**		**SF12**	
Code rate	**4/5**	**4/8**	**4/5**	**4/8**	**4/5**	**4/8**	**4/5**	**4/8**
Average	0.0652	0.0827	1.3441	1.7300	0.0235	0.0266	0.3452	0.4429
Max	0.1462	0.1183	1.3899	1.8328	0.1123	0.0550	0.3896	0.5346
Min	0.0618	0.0778	1.1941	1.4886	0.0196	0.0242	0.3070	0.3802
Loss	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %

Table 1. Min, Max and average of RTT, 100 samples per configuration

(indoors and outside in the immediate vicinity around the Department of Technology Systems (ITS) at Kjeller, Norway). The conclusion is, therefore, that tests need to be performed over even larger ranges (more than one km) and in areas with more possible noise and interference. This would be interesting to get an idea of the limits of LoRa as realized with our setup.

The timer mode we implemented was intended to perform RTT tests, in other words measuring the time from sending a message over LoRa to receiving a message over a MQTT subscription. Three parameters were tested: Bandwidth, spread factor and code rate. Bandwidth is how much of the frequency spectrum the chirp transmits over, spread factor is how fast it chirps over that distance and code rate is how many correction bits we include in each transmission. These factors were chosen since they can be expected to impact the transmission time.

We see from Figure 2 and Table 1 that the largest factor in transmission time is the spread factor. SF7 is faster than SF12 with quite a large margin. This is natural, as spread factor dictates how fast the chirp is and thus how fast we can send information. The parameter with the second most impact seems to be the bandwidth, where a larger bandwidth yields a shorter transmission time.

Another finding was that a higher code rate results in a slightly higher transmission time. This is of course because with a higher code rate you send more error correction symbols. Keep in mind that a higher code rate might get a faster average transmission time if communications is performed in noisy conditions, as packet loss or corruption would become a larger factor.

For the SF12 samples we can see that the first ten transmission have a visible drop in RTT, we are uncertain about what causes this, however.

4.1 Takeaway points

In general the technologies investigated in this initial experiment look promising, and as they become more widespread more uses for them will emerge. During this experiment we found that the Raspberry Pi solution for LoRa was a lot simpler to use and more flexible than expected. Further, the solution comes with the possibility to extend the device with other sensors designed with the Raspberry Pi in mind. A lot of time was spent using lower level C libraries at first, but in the end we found that the higher level Python library did everything we wanted to accomplish without sacrificing control, and it was a lot easier to use than the C counterpart.

In this experiment we used GPS as the "sensor data", it would of course be possible to use other kinds of data as well, since MQTT is a payload agnostic protocol.

5. SUMMARY

Internet of Things (IoT) is becoming more and more widespread. From a military perspective, IoT can potentially provide a new and cost effective approach to gathering information. With this in mind, we examined some of the different technologies used in IoT systems today, specifically the combination of Raspberry Pi 3 (platform), LoRa (communications), and MQTT (publish/subscribe protocol).

As we have seen in this initial experiment, platforms like Raspberry Pi and protocols like LoRa and MQTT look very promising. Base level testing and experimentation has been done, concluding with the need for more research of these and other IoT platforms. More work is especially needed regarding actual usage environments, such as field testing. Specifically, within the IST-176 group there is the potential for an international testing ground involving several NATO partners with an interest in military applications of these and other civilian IoT technologies.

6. FUTURE WORK

For future work, it would be interesting to perform additional tests with more devices, and also include fog computing aspects. Here, since we only had three devices available, the scale and insight from our test is somewhat limited. We considered evaluating throughput, but the nature of LoRa also meant that testing throughput would not be constructive as any result would be limited by the restrictions placed on the frequency band by laws and not any hardware or software configurations.

Further, testing in more challenging environments (in terms of range and noise), typically using soldier and/or vehicle mounted devices, would be interesting to perform, as well as gaining better insight into use, practicality and reliability in terms of battery life, especially in cold climates. Also, the unexpected drops in RTT for SF12 should be investigated further.

REFERENCES

[1] Endsley, M., "Theoretical underpinnings of situation awareness: A critical review." Situation awareness analysis and measurement, Jan. 2000, pp. 332.

[2] IoT Special Interest Group, "Technology Strategy Board." 2013.

[3] Suri, N., Tortonesi, M., Michaelis, J., Budulas, P., Benincasa, G., Russell, S., Stefanelli, C., and Winkler, R., "Analyzing the applicability of Internet of Things to the battlefield environment." 2016 International Conference on Military Communications and Information Systems (ICMCIS), 23-24 May, Brussels, Belgium (2016).

[4] Johnsen, F. T., Zielinski, Z., Wrona, K., Suri, N., Fuchs, C., Pradhan, M., Furtak, J., Vasilache, B., Pellegrini, V., Dyk, M., Marks, M., and Krzyszton, M., "Application of IoT in Military Operations in a Smart City." 2018 International Conference on Military Communications and Information Systems (ICMCIS), 22-23 May, Warsaw, Poland (2018).

[5] Poltronieri, F., Stefanelli, C., Suri, N., and Tortonesi, M., "Analyzing and Evaluating Information-Centric and Value-based Fog Service Architectures in Military Environments." 2018 International Command and Control Research and Technology Symposium (ICCRTS), 6-9 Nov, Pensacola, Florida, USA (2018).

[6] Michaelis, J., Morelli, A., Raglin, A., James, D., and Suri, N., "Leveraging LoRaWAN to Support IoBT in Urban Environments." IEEE World Forum on Internet of Things (WF-IoT) 2019, 16 April 2019, Special session on military applications of IoT, Limerick, Ireland.

[7] Raspberry Pi, "Raspberry Pi Foundation." Retrieved June 3rd, 2019, from https://www.raspberrypi.org/about/.

[8] James Adams, "Introducing Raspberry Pi hats." Retrieved June 3rd, 2019, from https://www.raspberrypi.org/blog/introducing-raspberry-pi-hats/.

[9] Semtech, "What is LoRa?." Retrieved June 3rd, 2019, from https://www.semtech.com/technology/lora/what-is-lora.

[10] Jalaian, B., Gregory, T., Suri, N., Russell, S., Sadler, L., and Lee, M., "Evaluating LoRaWAN-based IoT devices for the tactical military environment." 4th IEEE World Forum on Internet of Things (WF-IoT) 2018, Singapore, February 5-8, 2018 (2018).

[11] Søndrol, T., Jalaian, B., and Suri, N., "Investigating LoRa for the Internet of Battlefield Things: A Cyber Perspective." IEEE Military Communications Conference (MILCOM) 2018, October 29-31, LA, USA (2018).

[12] Spiess, A., "LoraWan Range World Record Attempt." Published 16 February, 2017. Retrieved June 3rd, 2019, from https://www.youtube.com/watch?v=adhWIo-7gr4.

[13] Wenner, R., "Lora CHIRP." Published 16 November, 2017. Retrieved June 3rd, 2019, from https://www.youtube.com/watch?v=dxYY097QNs0.

[14] Staniec, K. and Kowal, M., "LoRa Performance under Variable Interference and Heavy-Multipath Conditions." Wireless Communications and Mobile Computing, vol. 2018, Article ID 6931083, 9 pages, 2018, https://doi.org/10.1155/2018/6931083.

[15] Dragino, "Dragino: Open source wifi, linux appliance." http://www.dragino.com/, accessed 2019-06-03.

[16] Kennedy, G. and Davis, B., "Electronic Communication Systems." McGraw-Hill International, 1992, 4th ed., p 509, ISBN 0-07-112672-4.

[17] Sweeney, D., "An introduction to bluetooth a standard for short range wireless networking." 15th Annual IEEE International ASIC/SOC Conference, 25-28 Sept. 2002, https://doi.org/10.1109/ASIC.2002.1158106.

[18] Doelz, M. and Heald, E., "Minimum Shift Data Communication System." US Patent 2977417, 1958, Retrieved June 3rd, 2019, from http://www.freepatentsonline.com/2977417.html.

[19] Poole, I., "What is GMSK Modulation - Gaussian Minimum Shift Keying." Retrived June 3rd, 2019, from https://www.radio-electronics.com/info/rf-technology-design/pm-phase-modulation/what-is-gmsk-gaussian-minimum-shift-keying-tutorial.php.

[20] Lesurf, J., "Simple Binary Modulation - One Bit at a Time!." Retrived June 3rd, 2019, from https://www.st-andrews.ac.uk/~www_pa/Scots_Guide/RadCom/part19/page1.html.

[21] Manso, M., Johnsen, F. T., Lund, K., and Chan, K. S., "Using MQTT to Support Mobile Tactical Force Situational Awareness." 2018 International Conference on Military Communications and Information Systems (ICMCIS), 22-23 May, Warsaw, Poland (2018).

[22] Manso, M., Jansen, N., Chan, K., Toth, A., Bloebaum, T. H., and Johnsen, F. T., "Mobile Tactical Force Situational Awareness: Evaluation of Message Broker Middleware for Information Exchange." 2018 International Command and Control Research and Technology Symposium (ICCRTS), 6-9 Nov, Pensacola, Florida, USA.

[23] Johnsen, F. T., Landmark, L., Hauge, M., Larsen, E., and ivind Kure, "Publish/Subscribe Versus a Content-Based Approach for Information Dissemination." IEEE Military Communications Conference (MILCOM) 2018, October 29-31, LA, USA.

[24] Johnsen, F. T., "Using Publish/Subscribe for Short-lived IoT Data." 2nd Workshop on Internet of Things - Enablers, Challenges and Applications (IoT-ECAW18), 9-12 September, 2018, Poznan, Poland.

[25] OASIS, "OASIS Foundation." Retrieved June 3rd, 2019, from https://www.oasis-open.org/org.

[26] Edited by Andrew Banks and Rahul Gupta, “MQTT Version 3.1.1 Plus Errata 01.” OASIS Standard Incorporating Approved Errata 01, 10 December 2015. Retrieved June 3rd, 2019, from http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.

[27] Bertelsen, E., Berthling-Hansen, G., Bloebaum, T. H., Duvholt, C., Hov, E., Johnsen, F. T., Morch, E., and Weisethaunet, A. H., “Federated Publish/subscribe Services.” 2018 9th IFIP International Conference on New Technologies Mobility and Security (NTMS), 26-28 February, Paris, France (2018).

[28] Treffkorn, R., Brashear, D., Nelson, R., and Raymond, E. S.

[29] Zeimetz, B., “python-gps.” Retrieved June 3rd, 2019, from https://packages.debian.org/jessie/python-gps.

[30] Basseto, B., “PyLora.” Retrieved June 3rd, 2019, from https://github.com/Inteform/PyLora.git.

[31] Puente, P. Ø. and Johnsen, F. T., “Towards IoT in a military context: Using civilian commercial devices and open-source software.” FFI-Eksternnotat 18/00718, 2018.