Evaluation of Message Broker approaches for Information Exchange in Disadvantaged Tactical Networks in a Federated Environment

Topic 6: Experimentation, Analysis, Assessment and Metrics
Paper 59

Authors

Frank T. Johnsen
Norwegian Defence Research Establishment (FFI)
Norway

Marco Manso
PARTICLE, Lda.
Portugal

Norman Jansen
Fraunhofer FKIE
Germany

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

Abstract

The research task group NATO STO/IST-150 «NATO Core Services profiling for Hybrid Tactical Networks» is working towards profiling Core Services for the tactical domain, which can be described as DIL (disconnected, intermittent and limited) environments. In this context, our work is intended to feed into future spirals of the Federated Mission Networking (FMN) concept, which, until recently, has been mostly focused on fixed/stable infrastructure networks. In IST-150, we specifically address information exchange needs of deployed coalition forces. Hence, our work is an enabler for the future of command and control systems, which will need to use efficient and appropriate solutions that ensure timely delivery of data and information superiority up to the last mile.

In IST-150 we have analyzed and experimented with several message broker mechanisms for information exchange, including the NATO recommended WS-Notification standard for publish/subscribe. We have found that the Message Queuing Telemetry Transport (MQTT) industry standard exhibits low overhead and is worth further studies.

In this paper, we continue our analysis in the application of MQTT to tactical networks. We pursue a comparative experiment between three different approaches to sharing data in a federation. Two of these approaches involve multiple MQTT brokers, using the MQTT bridge and a bespoke MQTT mesh approach, respectively. The third approach is a serverless solution based on UDP. The paper investigates these approaches from a performance viewpoint, varying characteristics such as data exchange frequency, network delay and packet loss in an emulated tactical network. The data exchange consists of Blue Force Tracking information transmitted by all units. We present the obtained performance results between brokers considering the different network characteristics. Importantly, we identify the brokers' advantages and bottlenecks allowing a better understanding of the conditions in which they can be deployed.

1 Introduction and motivation

In NATO's Allied Joint Doctrine, Command and Control (C2) is considered a joint function [1]: "Joint functions provide a sound framework of related capabilities and activities grouped together to assist JFCs[1] to integrate, synchronize, and direct various capabilities and activities in joint operations." In other words, C2 is central to integrating, synchronizing and controlling military operations both horizontally and vertically. C2 is a key element of the planning phases as well as the execution of military operations.

Simply put, C2 can be centralized or decentralized. In centralized C2, there is a need for a very efficient and well-functioning information infrastructure to convey messages, as well as a superior understanding of the situation, to allow appropriate efforts and focus, the right decisions to be made and orders to be disseminated. A challenge here is the need to process possibly large amounts of information in near real time. This requires robust, precise and efficient communications and accompanying information infrastructure.

Conversely, decentralized C2 has less direct control and more emphasis on delegating responsibilities and operating according to the commander's intent. Here, coordination is on a lower level, which enables a more rapid pace in the operation.

[1] JFCs is an abbreviation for "Joint Force Commanders".

Modern technology enables detailed control of an operation over great distances [2]: "New technologies are creating an environment where the strategic, operational, and tactical levels of war can at times be so compressed as to appear virtually as a single function." Hence, the opportunity to do both centralized and decentralized C2 can be beneficial. With NATO Network Enabled Capability (NNEC) [3] came the shift away from platform based thinking, doing away with silo systems, and a goal to organize all available resources for maximum effect. With NNEC, the idea is that one can shift between centralized and decentralized C2, depending on what is considered most appropriate for a certain case. This vision of flexibility has been taken even further with the idea of edge organizations and "Power to the Edge" [4]. This approach to C2 involves a complete distribution of information, unlimited collaboration and a widely delegated mandate for making decisions. Here, flexibility, focus on the network and a large extent of decentralization gives an efficient use of available resources and emerging possibilities.

In NATO, the Consultation, Command and Control (C3) Taxonomy provide a common framework and vocabulary for classifying and naming the ICT functionality needed to support the alliance's C3 capabilities, and the operational contexts these capabilities support [5]. The C3 Taxonomy is divided into two main layers (which again can be subdivided into several other layers):

- Operational context (includes missions, operations, capabilities) - notably, C2 processes are part of this layer
- Communication and Information Systems (CIS) Capabilities (includes user equipment, information systems equipment, and communications equipment)

The research task group (RTG) NATO STO/IST-150 «NATO Core Services profiling for Hybrid Tactical Networks» works towards profiling NATO Core Services (which are part of the technical services CIS capabilities in the C3 Taxonomy) specifically for use in tactical networks. This work is important to support the future of C2 systems, which with the ever increasing need for timely exchange of data, must be able to function in tactical environments that have communications limitations like disconnections from other levels, intermittent connectivity between peers in the tactical domain itself, and last but not least, limitations on throughput and available bandwidth. The RTG is especially focusing on the Message-Oriented Middleware Service, which includes the publish/subscribe (i.e., push-pattern) communication paradigm. In this paper we present our latest findings on one possible realization of that service, namely the industry standard Message Queuing Telemetry Transport (MQTT) [6], and our attempts at employing it for (emulated) tactical communications for a Blue Force Tracking (BFT) application.

The remainder of this paper is organized as follows: Section 2 covers related and previous work. Section 3 introduces the MQTT protocol, and gives a brief introduction to the publish/subscribe communication paradigm. The experiments we performed with MQTT (i.e., setup, execution and results) are covered in Section 4. Finally, Section 5 concludes the paper and outlines some further work.

2 Related and previous work

The paper "Experimental Evaluation of Group Communications Protocols for Data Dissemination at the Tactical Edge" [7] originated from work in the NATO STO/IST-161 RTG, and presents an experimental evaluation of different approaches to group communication. The work includes tests of a mix of different approaches (centralized standardized solutions, decentralized standardized solutions, as well as proprietary decentralized solutions developed for use in tactical networks). The tests were conducted using the Anglova scenario [24] but without any form of routing enabled in the network (so

no multi-hop communication between nodes in the network unless the protocols themselves implemented this). The experiment evaluated traffic in a 24-node company in the Anglova scenario and used one radio model in the frequency range 300 MHz (data rate approx. 640 kb/s). In the selected 20 minutes of the scenario the experiment used, there was good radio coverage so that a high proportion of vehicles were within radio range of each other. Yet, a lack of routing is still part of the reason why solutions with a central point (typically standardized, publish/subscribe protocols) performed very poorly in the tests, while the proprietary, specialized solutions performed well. This contrasts with the work we have been doing in our group, NATO IST-150, where we have tested several standardized subscription-based protocols (also both in EMANE using the Anglova scenario, and also using other approaches – as further discussed below). The main difference is that in our experiments, we have been testing with a functional routing layer (typically OLSRv2 routing in the case of EMANE). In our findings, with an active routing layer, such standardized protocols, notably MQTT, work better than what the IST-161 experiments indicate.

In our previous work, we have evaluated MQTT supporting C2, using it as the information dissemination protocol in experiments and exercises enabling push-based tactical-level data to experimental C2 systems [8,9].

Also, we have performed comparisons of MQTT with other standardized protocols, notably including WS-Notification, which has been identified by NATO [25] previously for the publish/subscribe part of the Message-Oriented Middleware Service. While usable in many cases, we found that WS-Notification does not function well in tactical networks, at least not without performing adaptations to the protocols [10,11]. Hence, our focus was on doing comparative analysis of other industry standards, possibly usable without any adaptations. Through a series of such experiments, e.g., [12,13,14] we have found that MQTT emerges as an interesting alternative to WS-Notification due to it also being an industry standard, but seems to be a better match to the limitations of tactical networks due to it having low overhead [15,16].

In addition to performance, a key property of Core Services is the interoperability aspect. A core service needs to be usable in a federation of systems, typically supporting operational concepts like FMN. WS-Notification, being based on the Web Services family of standards, easily supports interoperability across different systems. Thus far we have established that MQTT is preferable to WS-Notification from a performance/communications overhead perspective, but for us to be able to fully recommend MQTT for use in NATO, we also need to investigate the federated systems perspective further. How to use MQTT in a federated system, and how the protocol's mechanisms perform from such a perspective, i.e., investigating information flow between multiple brokers, is the focus in this paper, which is a continuation of the initial work we did in [16].

3 A brief introduction to MQTT and publish/subscribe

MQTT is a publish/subscribe messaging protocol. The protocol is designed with lightweight low bandwidth operation in focus. We have found that it is able to function even in high latency and low reliability environments [15]. The protocol is very well suited to disseminating sensor information, so it is a much-used protocol for this purpose in applications of the Internet of Things [13]. Also, we have previously identified MQTT as a protocol of potential interest for use in military tactical networks [14]. MQTT is standardized as ISO/IEC PRF 20922 [6].

The publish/subscribe paradigm [17] implies that there are several different roles within the system. Here, we cover publish/subscribe from an MQTT specific perspective. Other protocols may emphasize other solutions. For MQTT, these roles 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 communication. 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 topic 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. The structure of topics can help identify relevant data, by allowing us to uniquely index services (and related messages), specifically. For example, a label can identify the location data coming from the corresponding unit, as represented by a script in our tests (more on that in the following sections in the paper). 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 wildcards. With this powerful feature it is possible to create more complex behavior in the system. For example, wildcards allow for grouping and receiving messages based on both location and nodes.

To deal with different functional requirements, and ultimately 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 received. 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, MQTT supports three different levels of QoS, named 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 communication. 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.

In this paper, we specifically investigate how multiple MQTT brokers can interact. The idea is both to do away with the single point of failure that a solitary broker constitutes, as well as investigating the federation aspect further. In a multinational operation, each nation in the coalition force may bring their own broker to the operation. This translates to a topology where each nation has its own edge broker, see [16] for a discussion on different topologies. With this setting in mind, we next describe our experiment design and setup.

4 Experiments design, setup and results

A main aim of this work is to study efficient mechanisms for information exchange in a multi-nation federated setting, which is common in NATO operations.

In this regard and in line with our past approaches (as described previously in the related work section), we explore the use of MQTT message brokers, following the publish/subscribe paradigm, as the mechanism to exchange information between different nations. Each nation manages and controls its own broker instance. A multi-broker setup is created allowing connecting the various brokers and consequently exchange tactical data pertaining to a mission.

Benefiting from the publish/subscribe paradigm, publishers (those that produce data) and subscribers (those that consume data) are loosely-coupled. The information exchange mechanism works by having a well-defined **topic structure** (to where messages will be published) accessible to subscribers, and to **message format** (allowing subscribers to "decode" the data contained in the message). Note that, while MQTT has a powerful mechanism for formatting and matching topics, the actual topic structure and message format is application specific. This means that, in a federation, the involved partners must agree on this information in advance, for instance through profiling of specific applications. This is not only the case for MQTT, but applies equally to other publish/subscribe mechanisms.

In our experiments, we used the same structure for topics and message format as in our previous work in [14]. The topic structure to publish and subscribe is the following:

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

For example, country-Id "PRT", squad-Id "PRT-UNIT001" and node-Id "PRT-S003" produces a BFT message to service "location", indicated by the topic string exemplified next:

```
/PRT/PRT-UNIT001/PRT-S003/location
```

The message format used for BFT complies with GeoJSON format [18]. An example of a BTF from nation "PRT", unit "PRT-UNIT001" and node identifier "PRT-S003" is given below. Note the message contains a unique message identifier and timestamp used for experiment analysis. Each message is JSON formatted, containing about 255 characters.


```
{"type": "Feature", "geometry": {"type": "Point", "coordinates": [38.74691, -9.156418, 0]},
"properties": {"country": "PRT", "unit": "PRT-UNIT001", "node_id": "PRT-S003", "msg_id":
"PRT-S003_28", "timestamp": 1588759108433}}
```


The experiments herein described instantiated two different multinational settings involved in fictional NATO operations, one involving two nations (NOR and PRT) and another one involving four nations (DEU, NOR, PRT and USA). These two settings allow assessing the multi-broker performance as a function of the number of brokers. The following multi-broker configurations were deployed:

- **Multi-broker configuration using Mosquitto [20]**. In this setup, each nation deploys an instance of the Mosquitto broker. However, since one of the brokers must act as main broker, this configuration has a single point-of-failure.

  Federation between the Mosquitto brokers is achieved by using the MQTT bridge mechanism. The bridge uses standard MQTT interfaces to configure information flow

between the brokers. This mechanism is not a part of the MQTT standard itself, but has become the de facto way of configuring multi-broker setups using MQTT, since it is based on the standard's primitives. It is supported by a number of different broker implementations, and we tested interoperability between such implementations in [16]. In this current experiment we aim to compare this mechanism with the other approaches outlined next.

- **Multi-broker configuration using VerneMQ [21] mesh configuration**. In this setup, each nation deploys an instance of VerneMQ operating at the same network hierarchical level (i.e., mesh). The mesh configuration is however a non-standard MQTT feature. This feature lets you build a robust cluster of VerneMQ brokers, so that the cluster functions as one single broker to the outside world. Here, both messages and active subscriptions are replicated across the mesh, so that any broker can serve any request. We expect this mechanism to be very resilient towards failure, but want to explore its overhead compared to the bridge as explained above, since the bridge likely has less overhead.
- **Broker-less configuration [22]** that uses MQTT-based messages that are UDP broadcast across networks. This configuration does not require a message broker. This feature is not part of the MQTT standard, however, given the characteristics of a tactical network (e.g., limited bandwidth and intermittent), the use of UDP over TCP is worth investigating further. We expect this approach, since it is based on UDP, to have a small footprint and possibly be very efficient in actual use.

The realization of the experiments involved the instantiation of emulated nodes representing coalition forces from different countries. Within a nation, the units are interconnected by a broadband network (unlimited throughput, always connected). Between nations, a data-link was emulated, using netem [23], allowing network parameters to be set close to representative Combat Network Radio (CNR) tactical network conditions. We chose CNR here because we think that it is a representative communication channel in the field. Also, it is a much narrower link than the tactical broadband that we have investigated earlier. The following network configurations were used to emulate datalinks between nations:

- **Baseline data-link setup**: in this setting, no limitations were set to the network's characteristics. Given the emulated nodes were executed in virtual machines, the network yields high throughput (>100Mbps) and minimal latency (order of a few ms).
- **Tactical data-link setup 1**: in this setting, the network throughput is set to 9.8kbps, with 100 ms latency and 1% packet loss.
- **Tactical data-link setup 2**: in this setting, the network throughput is set to 9.8kbps, with 100 ms latency and 10% packet loss.

When assessing network performance resulting from data exchange in our experiments, the baseline data-link is close to near-optimal conditions, while the two tactical data-link setups are close to tactical network conditions.

Finally, two different update rates (period) are used for the units' locations:

- **Update the units' location every 2 seconds**. A total of 600 location points are published per node over 1200 seconds.
- **Update the units' location every 10 seconds**. A total of 120 location points are published per node over 1200 seconds.

Changing the location update rate results in different network throughput load, which allows assessing which configurations perform best.

Deployment Approach

For the execution of the experiments, the following approach was used:

- A virtual machine was created for each nation.
- The virtual machines were connected to each other by means of a virtual network, accessible via IP addresses.
- The units were emulated by means of software scripts running inside the respective nation's virtual machine. The scripts published and subscribed messages, also generating required logs for analysis.

The experiment settings are described next.

Multinational Setting 1: Multi-Broker Exchange between NOR and PRT

This setup involves two nations - named NOR and PRT - each deploying a convoy comprising 8 mobile units, shown in Figure 1. In order to develop a complete shared situational awareness, nations agree on exchanging BFT between all their units. As stated before, the underlying assumption is that each nation manages its own message broker and they agree on an appropriate multi-broker setup allowing exchanging topics and messages.

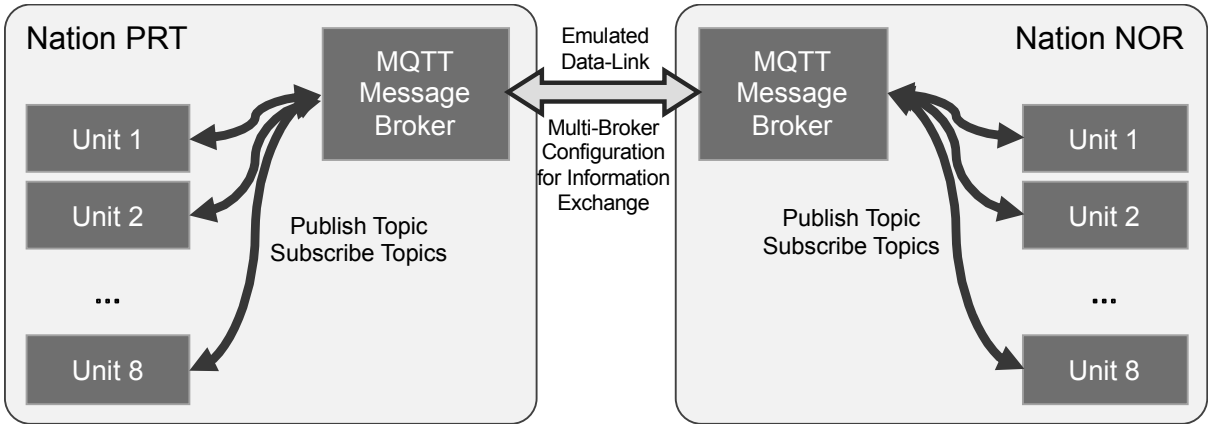

Figure 1 - Multinational Setting 1: Two Nations

The variations used in the experiments are presented in the next table.

Message Broker	Network Configuration	Location Update Period
Mosquitto	Baseline setup	2 seconds
Mosquitto	Tactical setup 1	2 seconds
Mosquitto	Tactical setup 2	2 seconds
Mosquitto	Baseline setup	10 seconds
Mosquitto	Tactical setup 1	10 seconds

Mosquitto	Tactical setup 2	10 seconds
VerneMQ mesh	Baseline setup	2 seconds
VerneMQ mesh	Tactical setup 1	2 seconds
VerneMQ mesh	Tactical setup 2	2 seconds
VerneMQ mesh	Baseline setup	10 seconds
VerneMQ mesh	Tactical setup 1	10 seconds
VerneMQ mesh	Tactical setup 2	10 seconds

Table 1: Experiment variations for Multinational Setting 1

As shown in Table 1, a total of 12 experiment runs were conducted.

Multinational Setting 2: Multi-Broker Exchange between DEU, NOR, PRT and USA

This setup involves four nations - named DEU, NOR, PRT and USA - each deploying a convoy comprising 8 mobile units. The intent to share up to date position information, as well as having each nation managing its own message broker, is the same as for the multinational setting 1.

The network setup is depicted in Figure 2.

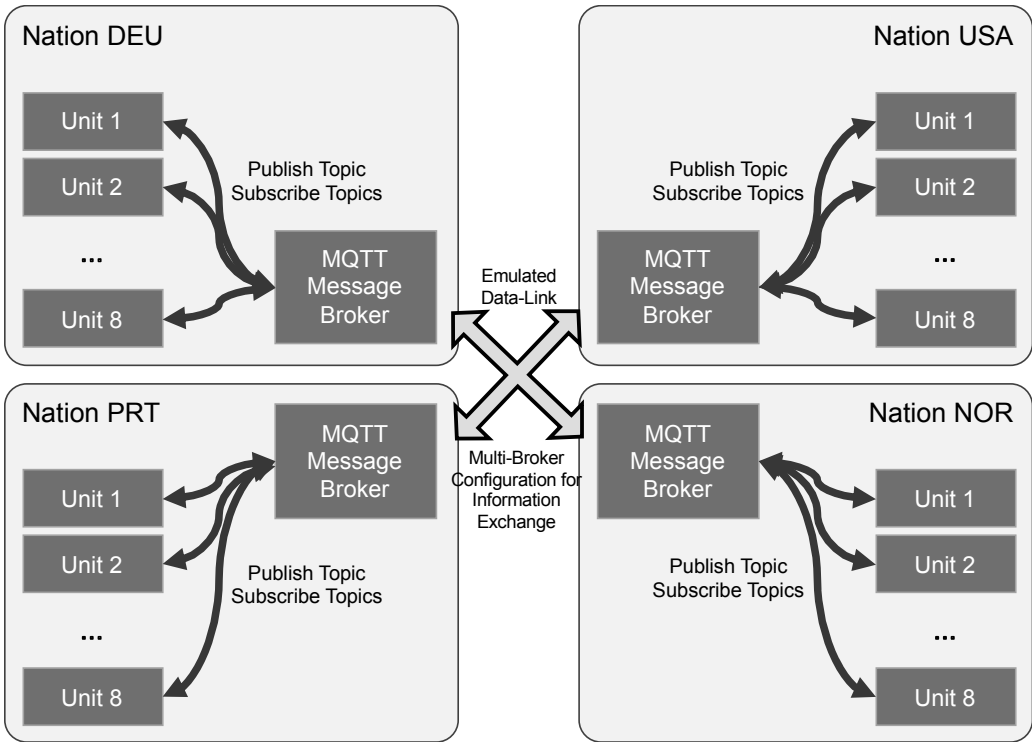

Figure 2 - Multinational Setting 2: Four Nations

The variations used in the experiments are presented in the next table. Note that in this setting, the MQTT UDP brokerless setting is introduced.

Message Broker	Network Configuration	Location Update Period
Mosquitto	Baseline setup	2 seconds
Mosquitto	Tactical setup 1	2 seconds
Mosquitto	Tactical setup 2	2 seconds
Mosquitto	Baseline setup	10 seconds
Mosquitto	Tactical setup 1	10 seconds
Mosquitto	Tactical setup 2	10 seconds
VerneMQ mesh	Baseline setup	2 seconds
VerneMQ mesh	Tactical setup 1	2 seconds
VerneMQ mesh	Tactical setup 2	2 seconds
VerneMQ mesh	Baseline setup	10 seconds
VerneMQ mesh	Tactical setup 1	10 seconds
VerneMQ mesh	Tactical setup 2	10 seconds
UDP MQTT (Brokerless)	Baseline setup	2 seconds
UDP MQTT (Brokerless)	Tactical setup 1	2 seconds
UDP MQTT (Brokerless)	Tactical setup 2	2 seconds
UDP MQTT (Brokerless)	Baseline setup	10 seconds
UDP MQTT (Brokerless)	Tactical setup 1	10 seconds
UDP MQTT (Brokerless)	Tactical setup 2	10 seconds

Table 2: Experiment variations for Multinational Setting 2

As shown in Table 2, a total of 18 experiment runs were conducted.

Experiment results

For the analysis of the experiments, we used analyzing tools from the Analyze and Test environment (AuT) project of Fraunhofer FKIE [19]. AuT allows generating suitable metrics for military applications, relevant to conduct an assessment of tactical networks.

For this purpose, we included in the MQTT client implementation a logging component, which logs relevant information in a JSON format defined in AuT. For each message which was sent or received a corresponding logging entry is generated:

- At the producer side, details concerning each produced message, including producer id (i.e., node id), timestamp and destinations (i.e., all node ids).
- At the subscriber side, details concerning each received message, including receiver id (i.e., node id), producer id and timestamp (by receiver).

The framework for analyzing these logs was extended to automatically generate statistics information (number of sent messages per group, number of received messages per group, delay of message transfers with median, minimum, maximum and quartiles). Based on these statistics, boxplot diagrams can be generated visualizing the performance of the MQTT implementations in the different test setups.

To evaluate the transmission reliability of different MQTT implementations, we measured how many messages got lost in different setups. For this purpose, we compared Mosquitto with a bridge approach and VerneMQ with a mesh-approach in scenarios with two MQTT servers and scenarios with four MQTT servers respectively. In the four server scenarios additionally a UDP based MQTT implementation [22] was tested. All experiments were conducted in the two server settings with a sending period rate of one message every two seconds or one message every 10 seconds, respectively. Additionally, all experiments were conducted with a perfect (LAN) network and two disadvantaged DIL link setups (9.8 kbit/s, delay 100 ms and 1% packet loss or 10% packet loss, respectively).

As shown in Figure 3, in the multinational setting 1: two server scenarios VerneMQ did not work well with respect to transmission reliability in case of DIL networks when the sending rate was high (2 seconds). If the sending rate was lower (10 seconds) Mosquitto and VerneMQ performed reliably. The graph shows for each setup the loss rates for all messages and for all messages between groups (i.e. crossing an emulated (possibly DIL) network link). It should be noted that we used QoS0 in these experiments. Our previous works have compared different QoS settings and their reliability and overhead [15].

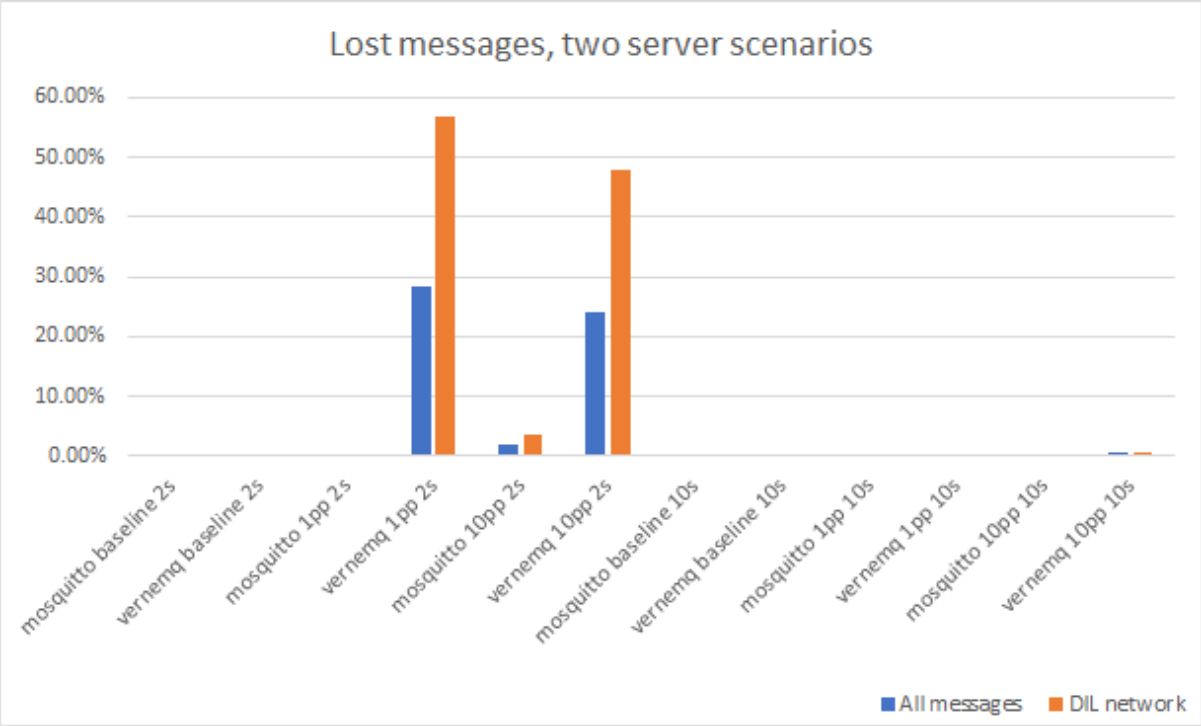

Figure 3 - Lost messages, two servers

In the multinational setting 2: four server scenarios (see Figure 4) both Mosquitto and VerneMQ did not perform well with loss rates of 60-85 % with the 2 seconds sending period and about 27-36 % with the 10 seconds period. It's notable to see that it did not make a difference with respect to the reliability if the emulated DIL link had a loss of 1 % or 10 % in this case. The UDP based MQTT server had about 1 % loss in the 1 % loss scenarios and 10 % loss in the 10 % loss scenarios. We conclude from this, that the use of TCP in DIL networks led to message losses, because TCP is not well suited for links with low data rates and high loss rates. The network logs show duplicated acknowledgements and spurious retransmissions in this case. Furthermore, we see that reliability strongly depends on the number of messages sent in a time period. Lowering the sending rate can help to improve the reliability in this case. Overall the experiments confirm that in a DIL network setup, a UDP based protocol has less inherent overhead, and thus can move more payload compared to the TCP based solutions when the network capacity is really low. Hence, it could be beneficial to favor UDP under such conditions.

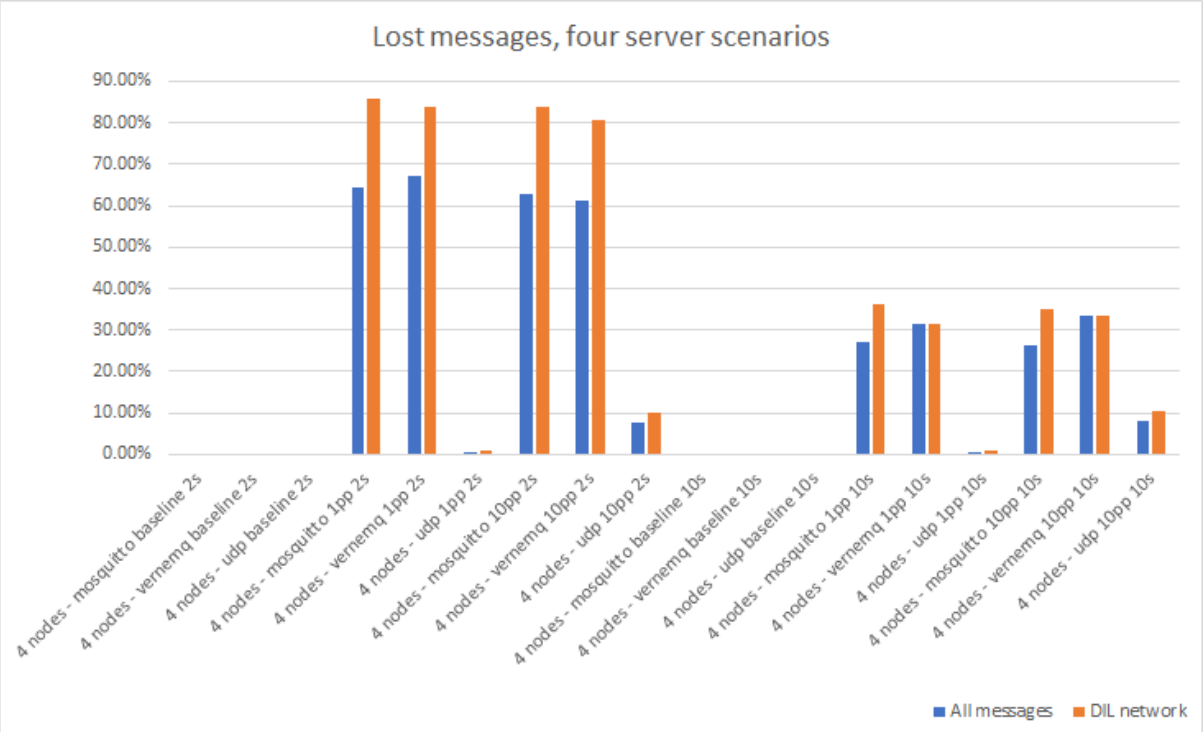

Figure 4 - Four server tests

To compare the transmission delay of the different MQTT implementations in different setups, boxplots were generated for each test run, one for all messages and one for all messages between two groups (i.e. crossing an emulated (possibly DIL) network link).

As shown in Figure 5, in the multinational setting 1: two server setup the performance of VerneMQ and Mosquitto degrades when many messages are sent (2 seconds period) and when the DIL link gets worse with respect to packet losses. Mosquitto performs slightly better relating to the transmission delay. If the sending rate is reduced, both solutions perform better. Mosquitto still has lower transmission delays. This is, naturally, due to the differences in the broker-to-broker mechanism. Mosquitto uses an MQTT bridge approach, which in essence means forwarding messages that match the configured bridge criteria. VerneMQ, on the other hand, uses a bespoke clustering mechanism, which replicates not only messages but also information on active subscriptions between all brokers in the cluster. So, while Mosquitto's approach is more efficient, the approach of VerneMQ offers the most versatile approach.

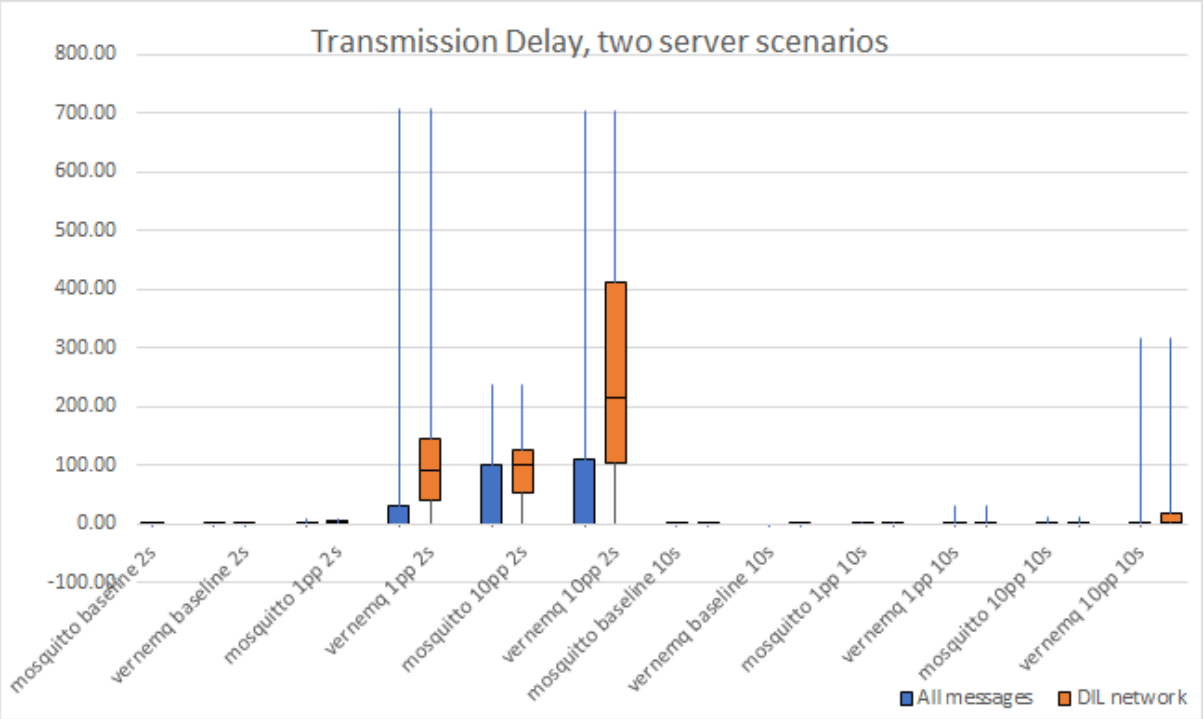

Figure 5 - Transmission delay, two servers

In the multinational setting 2: four server scenarios (see Figure 6), which also includes the UDP based MQTT implementation, you can see that Mosquitto performs better than VerneMQ in most cases (despite of the run with 10 % loss, 2 seconds period), but both do not perform well in response to packet loss. In contrast, the UDP based implementation has significantly lower transmission delays and much lower packet losses (see above). In overall, the use of MQTT with UDP seems to be superior in low-bandwidth networks with high packet losses compared to TCP. This is likely due to the effect of packet loss on TCP retransmission, which leads to more retransmissions, further bogging down the network with traffic. The use of a mesh approach in VerneMQ does not seem to be beneficial compared to the bridge approach used with Mosquitto in a setup with up to four servers.

To further analyze why the transmission delays rise significantly when the sending rate and packet loss are increased, we used another visualization graph provided by AuT Analyzer Component. As an example Figure 7 shows the transmission times of all messages separately (cf. red dots in Figure 7) for the Mosquitto four server scenario with 10% loss rate and two seconds sending period. The transmission times increase during the test run. We assume that this is the case, because more messages are sent than the DIL network links can cope with. This leads to increasing queue levels in the IP stack of the sending servers.

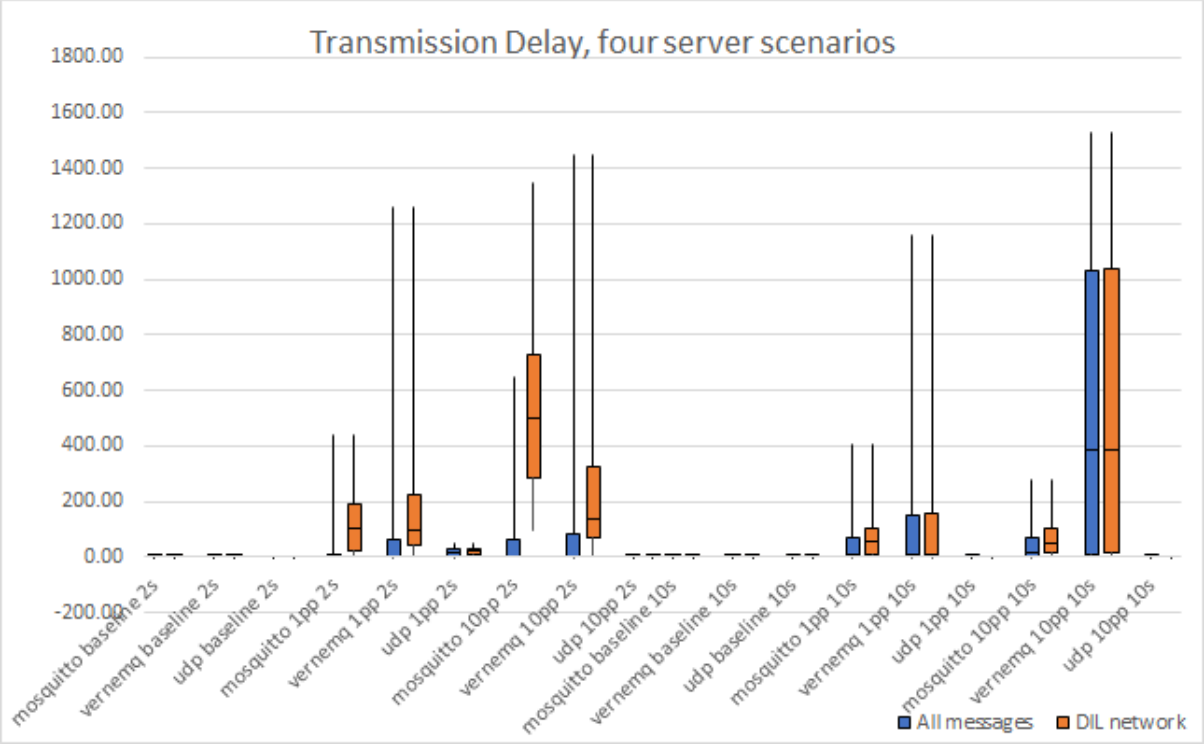

Figure 6 - Transmission delay, four servers

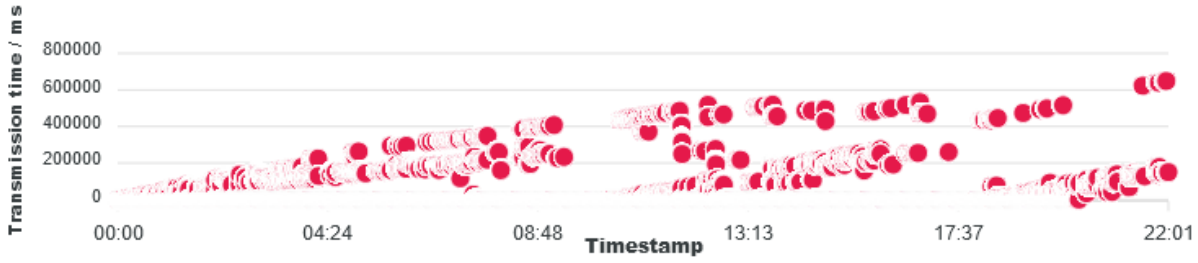

Figure 7 - Transmission times of all messages

5 Conclusion and future work

In this paper, we have investigated scenarios bridging two and four nations' networks, respectively. We tested three different MQTT implementations, notably

1. Mosquitto (with the MQTT bridge mechanism) [20]
2. VerneMQ (with its bespoke mesh approach to build an MQTT cluster) [21]
3. A proprietary UDP-based version of MQTT [22]

From our experiments, we found that the use of MQTT with UDP seems to be superior in low-bandwidth networks with high packet losses compared to the standard TCP-based flavors of MQTT. Further, the clustering mechanism in VerneMQ does not seem to be beneficial compared to the bridge approach used with Mosquitto in a setup with up to four servers. Naturally, this boils down to the more elaborate mechanism implemented by VerneMQ, which shares not only data between brokers, but also subscription information. So, the bridge in Mosquitto implements a selected forwarding of configured topics, whereas the clustering mechanism in VerneMQ provides full redundancy on both data and subscriptions in the cluster. The tradeoff for this additional functionality is, naturally, more

resource use than the simpler mechanism in Mosquitto. But, based on pure performance, we would have to recommend pursuing the UDP version further as of now, since it is the most efficient communications wise.

A drawback in today's MQTT standard is that it is indeed TCP based. However, our experiments show that UDP is a better match for this kind of messaging in tactical networks. So, ideally the MQTT specs should evolve to support this. Alternatively, one could also consider using something other than MQTT entirely. If one is considering moving away from industry standards, then there are other, proprietary options that perform well in tactical networks, see e.g., [7].

For future work, we would like to investigate cross-layer optimizations in conjunction with MQTT, and perhaps also other protocols. In our opinion, the experiments indicate that cross-layer optimizations are needed to dynamically adapt the sending rate of applications to the capacity of the network links. If the applications are not coordinated with the current network status, the transmission times will grow due to increasing queue levels in the applications or IP stack (see Figure 7). This leads to highly delayed position messages and will hinder an up-to-date common operational picture.

Finally, the client side dynamics should be further explored. Here, we have tested both MQTT bridging and MQTT clustering, which mitigate the single point of failure of a single MQTT broker. Yet, to fully overcome the problem, the client side of the system needs to be aware of available brokers and dynamically switch to another, available one if the broker it is currently connected to fails.

References

[1] NATO AJP-3. Allied Joint Doctrine for the Conduct of Operations, Brussels, NATO 2011.

[2] Peter W. Singer, "Tactical Generals: Leaders, Technology, and the perils of Battlefield Micromanagement", Air and Space Power Journal XXIII, no. 2, 2009.

[3] NATO, MCM-0032-2006 "NATO Network-Enabled Capabilities Feasibility Study", Brussels, NATO, 2006.

[4] David S. Alberts and Richard E. Hayes, "Power to the Edge", DoD Command and Control Research Program, 2003.

[5] NATO, "CONSULTATION, COMMAND AND CONTROL BOARD (C3B) C3 TAXONOMY BASELINE 3.1", AC/322-D(2019)0034, 2019

[6] ISO/IEC 20922:2016. Information technology – Message Queuing Telemetry Transport (MQTT) v3.1.1. ISO/IEC JTC 1 Information technology. Publication date: June-2016. https://www.iso.org/standard/69466.html

[7] Suri, N., Breedy, M.R., Marcus, K.M., Fronteddu, R., Cramer, E.R., Morelli, A., Campioni, L., Provosty, M., Enders, C., Tortonesi, M., and Nilsson, J., "Experimental Evaluation of Group Communications Protocols for Data Dissemination at the Tactical Edge", 2019 International Conference on Military Communications and Information Systems (ICMCIS 2019), 14 May 2019 through 15 May 2019, Budva, Montenegro.

[8] Marco Manso, Marianne R. Brannsten and Frank T. Johnsen, "A Smart Devices Concept for Future Soldier Systems", 22nd International Command and Control Research & Technology Symposium (ICCRTS), Nov 6, 2017 - Nov 8, 2017, Los Angeles, USA

[9] Trude H. Bloebaum and Frank T. Johnsen, “A Hybrid Push/pull C4IS Information Exchange Architecture Concept”, 23rd ICCRTS, 6 - 9 November 2018, Pensacola, Florida, USA

[10] Trude H. Bloebaum, Frank T. Johnsen, Marianne R. Brannsten, Jose Alcaraz-Calero, Qi Wang, and James Nightingale, “Recommendations for realizing SOAP publish/subscribe in tactical networks”, 2016 International Conference on Military Communications and Information Systems (ICMCIS), Brussels, Belgium, 23rd-24th May 2016

[11] Frank T. Johnsen, Trude H. Bloebaum, Jose M. Alcaraz Calero, Qi Wang, James Nightingale, Marco Manso, Norman Jansen, “WS-Notification Case Study and Experiment”, International Conference on Military Communications and Information Systems (ICMCIS) 2017, Oulu, Finland

[12] Trude H. Bloebaum and Frank T. Johnsen, “Evaluating publish/subscribe approaches for use in tactical broadband networks”, 2015 Military Communications Conference (MILCOM), 26 - 28 October 2015, Tampa, Florida, USA

[13] Frank T. Johnsen, “Using Publish/Subscribe for Short-lived IoT Data”, 2nd Workshop on Internet of Things - Enablers, Challenges and Applications (IoT-ECAW’18) Poznań, Poland, 9 - 12 September, 2018

[14] Marco Manso, Frank T. Johnsen, Ketil Lund, Kevin S. Chan, “Using MQTT to Support Mobile Tactical Force Situational Awareness”, 2018 Military Communications and Information Systems ICMCIS (former MCC), 22nd - 23rd May 2018, Warsaw, Poland

[15] Frank T. Johnsen, Trude H. Bloebaum, Norman Jansen, Gerome Bovet, Marco Manso, Andrew Toth and Kevin S. Chan, “Evaluating Publish/Subscribe Standards for Situational Awareness using Realistic Radio Models and Emulated Testbed”, 24th International Command and Control Research and Technology Symposium (ICCRTS), October 29-31 2019, Laurel, Maryland, USA.

[16] Marco Manso, Barbara Guerra, Fernando Freire, Norman Jansen, Kevin Chan, Andrew Toth, Trude H. Bloebaum and Frank T. Johnsen, “Mobile Tactical Forces: Experiments on Multi-broker Messaging Middleware in a Coalition Setting”, 24th International Command and Control Research and Technology Symposium (ICCRTS), October 29-31 2019, Laurel, Maryland, USA.

[17] P.T. Eugster, P. A. Felber, R. Guerraoui, and A-M. Kermarrec, “The Many Faces of Publish/Subscribe”, ACM Computing Surveys, Vol. 35, No. 2, June 2003

[18] IETF. The GeoJSON format. Available: https://tools.ietf.org/html/rfc7946

[19] M. Hirsch, A. Becker, F. Angelstorf, and F. Noth. “Performance Analysis of C2IS in Distributed Tactical Networks,” 2019 International Conference on Military Communications and Information Systems (ICMCIS 2019), Budva, Montenegro, 2019.

[20] Eclipse Foundation. “Eclipse Mosquitto™ An open source MQTT broker”, https://mosquitto.org/

[21] Octavo Labs. “VerneMQ”. https://vernemq.com/

[22] MQTT/UDP. https://mqtt-udp.readthedocs.io/en/latest/

[23] Linux manual page. “Tc-netem”. https://www.man7.org/linux/man-pages/man8/tc-netem.8.html

[24] N. Suri, J. Nilsson, A. Hansson, U. Sterner, K. Marcus, L. Misirlioglu, M. Hauge, M. Peuhkuri, B. Buchin,R. in't Velt, and M. Breedy, “The angloval tactical military scenario and experimentation environment,” 2018 International Conference on Military Communications and Information Systems (ICMCIS), Warsaw, Poland, 22 - 23 May 2018.

[25] NATO CESWG, ”Core Enterprise Services Standards Recommendations -The SOA Baseline Profile version 1.7”, dated 11.11.2011