NORTH ATLANTIC TREATY ORGANIZATION

SCIENCE AND TECHNOLOGY ORGANIZATION

AC/323(IST-150)TP/1008

www.sto.nato.int

STO TECHNICAL REPORT **TR-IST-150**

NATO Core Services Profiling for Hybrid Tactical Networks

(Profilage de services de base de l'OTAN pour les réseaux tactiques hybrids)

Final report of STO Research Task IST-150/RTG-072.

Published March 2021

NORTH ATLANTIC TREATY ORGANIZATION

SCIENCE AND TECHNOLOGY ORGANIZATION

AC/323(IST-150)TP/1008

www.sto.nato.int

STO TECHNICAL REPORT **TR-IST-150**

NATO Core Services Profiling for Hybrid Tactical Networks

(Profilage de services de base de l'OTAN pour les réseaux tactiques hybrids)

Final report of STO Research Task IST-150/RTG-072.

Editors: Trude H. Bloebaum, Kevin Chan, Norman Jansen, Frank T. Johnsen, Marco Manso, Andrew Toth

The NATO Science and Technology Organization

Science & Technology (S&T) in the NATO context is defined as the selective and rigorous generation and application of state-of-the-art, validated knowledge for defence and security purposes. S&T activities embrace scientific research, technology development, transition, application and field-testing, experimentation and a range of related scientific activities that include systems engineering, operational research and analysis, synthesis, integration and validation of knowledge derived through the scientific method.

In NATO, S&T is addressed using different business models, namely a collaborative business model where NATO provides a forum where NATO Nations and partner Nations elect to use their national resources to define, conduct and promote cooperative research and information exchange, and secondly an in-house delivery business model where S&T activities are conducted in a NATO dedicated executive body, having its own personnel, capabilities and infrastructure.

The mission of the NATO Science & Technology Organization (STO) is to help position the Nations' and NATO's S&T investments as a strategic enabler of the knowledge and technology advantage for the defence and security posture of NATO Nations and partner Nations, by conducting and promoting S&T activities that augment and leverage the capabilities and programmes of the Alliance, of the NATO Nations and the partner Nations, in support of NATO's objectives, and contributing to NATO's ability to enable and influence security and defence related capability development and threat mitigation in NATO Nations and partner Nations, in accordance with NATO policies.

The total spectrum of this collaborative effort is addressed by six Technical Panels who manage a wide range of scientific research activities, a Group specialising in modelling and simulation, plus a Committee dedicated to supporting the information management needs of the organization.

- AVT Applied Vehicle Technology Panel
- HFM Human Factors and Medicine Panel
- IST Information Systems Technology Panel
- NMSG NATO Modelling and Simulation Group
- SAS System Analysis and Studies Panel
- SCI Systems Concepts and Integration Panel
- SET Sensors and Electronics Technology Panel

These Panels and Group are the power-house of the collaborative model and are made up of national representatives as well as recognised world-class scientists, engineers and information specialists. In addition to providing critical technical oversight, they also provide a communication link to military users and other NATO bodies.

The scientific and technological work is carried out by Technical Teams, created under one or more of these eight bodies, for specific research activities which have a defined duration. These research activities can take a variety of forms, including Task Groups, Workshops, Symposia, Specialists' Meetings, Lecture Series and Technical Courses.

The content of this publication has been reproduced directly from material supplied by STO or the authors.

Published March 2021

ISBN 978-92-837-2328-8

Table of Contents

List of Figures

List of Tables

List of Meetings

Kick-off, The Hague, NLD, November 2016.

TIDE, Saint-Malo, FRA, April 2017.

Meeting in Oslo, NOR, June 2017.

Meeting in Oslo, NOR, October 2017.

Coordination with TIDE, Genova, ITA, April 2018.

3rd International Workshop on Service-Oriented Computing in Disconnected, Intermittent and Limited (DIL) Networks (SOC-DIL), IEEE ICC 2018, Kansas City, MO, USA, May 2018.

Meeting in Thun, CHE, December 2018.

Meeting in Lillehammer, NOR, May 2019.

IST-150 Membership List

CHAIR

Mr. Norman JANSEN
Fraunhofer-FKIE
GERMANY
Email: norman.jansen@fkie.fraunhofer.de

MEMBERS

Mr. Erkut BEYDAGLI
TÜBİTAK BİLGEM
TURKEY
Email: erkut.beydagli@tubitak.gov.tr

Dr. Gerome BOVET
Swiss Department of Defense Armasuisse
Science+Technology
SWITZERLAND
Email: gerome.bovet@armasuisse.ch

Dr. Kevin CHAN
US Army Research Laboratory (ARL)
UNITED STATES
Email: kevin.s.chan.civ@mail.mil

Mr. Mark EDWARDS
RINICOM
UNITED KINGDOM
Email: mark.edwards@rinicom.com

Ms. Trude HAFSOE BLOEBAUM
Norwegian Defence Research Establishment (FFI)
NORWAY
Email: trude-hafsoe.bloebaum@ffi.no

Dr. Frank T. JOHNSEN
Norwegian Defence Research Establishment (FFI)
NORWAY
Email: frank-trethan.johnsen@ffi.no

Mr. Marco MANSO
PARTICLE-Summary Lda.
PORTUGAL
Email: marco@particle-summary.pt

Mr. Daniel MARCO-MOMPEL
NCIA
NCIA – NATO Communication Information Agency
Email: daniel.marco-mompel@ncia.nato.int

Dr. Gregorio MARTINEZ PEREZ
University of Murcia (UMU)
SPAIN
Email: gregorio@um.es

Mr. Ali REZAKI
TUBITAK
TURKEY
Email: ali.rezaki@tubitak.gov.tr

Mr. Andrew TOTH
US Army Research Laboratory
UNITED STATES
Email: andrew.j.toth.civ@mail.mil

1st Lieutenant Tuomas TURTO
Army Academy / Army Research Centre
FINLAND
Email: tuomas.turto@mil.fi

PANEL/GROUP MENTOR

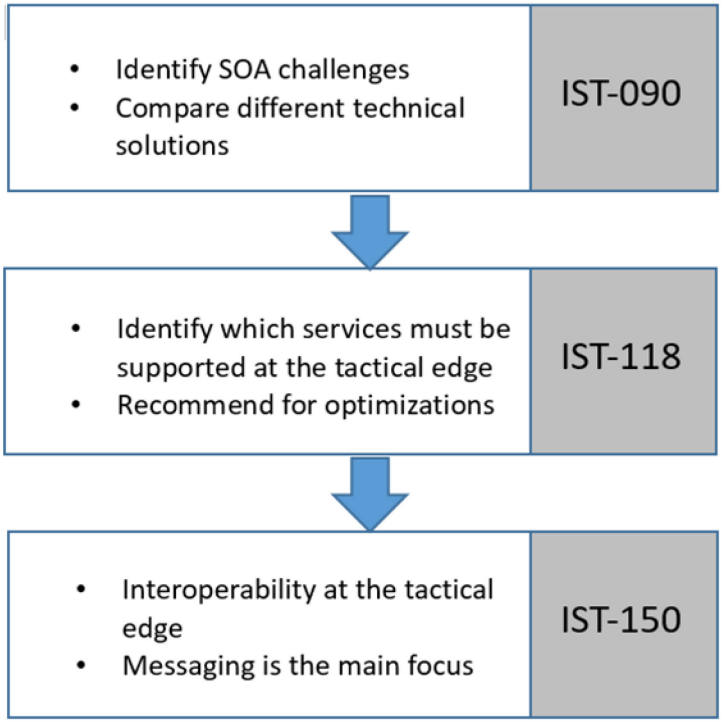
Dr. Markus ANTWEILER
Fraunhofer
GERMANY
Email: markus.antweiler@fkie.fraunhofer.de

NATO Core Services Profiling for Hybrid Tactical Networks

(STO-TR-IST-150)

Executive Summary

NATO IST-150 "NATO Core Services profiling for Hybrid Tactical Networks" is the third in a series of research task groups targeting Service-Oriented Architecture (SOA) in the tactical domain. The first group, IST-090, identified challenges, followed by IST-118, where we identified which services and functions must be supported at the tactical level. Finally, in IST-150, we have concentrated on one enabling function, that of the Message-Oriented Middleware (MOM) Core Service.



Federated Mission Networking (FMN) is the main context and motivation for our work, in that current FMN spirals thus far have not been focusing on the tactical level. The work performed by IST-150 is intended to provide knowledge about services at the tactical level, and possibly feed into future spirals of FMN targeting the tactical level specifically.

MOM can be subdivided in two main communication paradigms: Publish/subscribe communication and request/response communication. In IST-150, we have performed experiments on actual and emulated tactical networks. We summarize this work in the report and based on the obtained results, present the following specific recommendations:

- For publish/subscribe, we have done extensive comparisons between prolific industry standard protocols. Our findings indicate that Message Queueing Telemetry Transport (MQTT) yields

the lowest overhead and thus overall best performance in tactical networks. The report covers our performance experiments, as well as experiments investigating MQTT's capability as a federation protocol between different nations' systems.

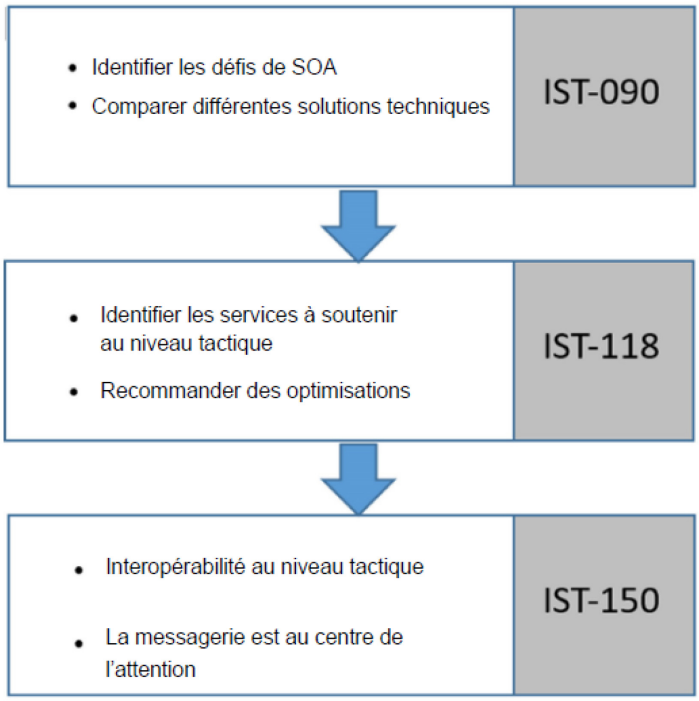
- Considering request/response, we have been looking into efficient approaches for consuming services across tactical networks. Specifically, we have looked into proxies for overcoming disruption problems, as well as replacing the commonly used HTTP/TCP transport (foundational for most SOAP and all REST services) with other approaches. Our findings indicate that replacing HTTP/TCP with CoAP is beneficial in tactical networks, as this protocol exhibits lower overhead and better overall performance under very limited bandwidth conditions where TCP-based solutions suffer. Typically, the TCP retransmission mechanism contributes to congest the link on a narrow channel, since high delay can erroneously be identified as packet loss, hence triggering retransmissions. In such cases, UDP based communications usually leads to a higher amount of delivered packets. This, for example, is why CoAP (being UDP based) fares better than HTTP/TCP for low throughput links. The report covers these findings in detail.

Profilage de services de base de l'OTAN pour les réseaux tactiques hybrides

(STO-TR-IST-150)

Synthèse

L'IST-150 de l'OTAN intitulé « Profilage de services de base de l'OTAN pour les réseaux tactiques hybrides » est le troisième groupe de recherche d'une série ciblant une architecture orientée service (SOA) dans le domaine tactique. Le premier groupe, l'IST-090, a identifié les défis à relever, puis l'IST-118 a identifié les services et les fonctions à soutenir au niveau tactique. Enfin, l'IST-150 s'est concentré sur une fonction facilitatrice, celle du service de base du logiciel médiateur orienté message (MOM).



Le réseau de mission fédéré (FMN) est le principal contexte et la principale motivation de nos travaux, au sens où les spirales FMN ne se sont pas focalisées jusqu'à présent sur le niveau tactique. Les travaux effectués par l'IST-150 sont destinés à fournir des connaissances sur les services au niveau tactique et éventuellement alimenter de futures spirales FMN visant spécifiquement le niveau tactique.

Le MOM peut être divisé en deux paradigmes de communication principaux : publier/s'abonner et requête/réponse. Au sein de l'IST-150, nous avons réalisé des expériences sur des réseaux tactiques réels et émulés. Nous résumons ce travail dans le rapport et, à partir des résultats obtenus, présentons les recommandations particulières suivantes :

- Pour le paradigme publier/s'abonner, nous avons réalisé des comparaisons complètes entre les nombreux protocoles standard du secteur. Nos conclusions indiquent que le protocole Message Queueing Telemetry Transport (MQTT) a le plus faible coût et donc les meilleures performances

globales dans les réseaux tactiques. Le rapport traite de nos expériences mesurant les performances, ainsi que d'expériences étudiant la capacité du MQTT en tant que protocole de fédération entre différents systèmes nationaux.

- Au sujet du paradigme requête/réponse, nous avons étudié les approches efficaces de consommation de services dans les réseaux tactiques. Nous avons en particulier examiné les proxys pour surmonter les problèmes de perturbation et remplacer le protocole de transmission couramment utilisé HTTP/TCP (fondamental pour la plupart des services SOAP et tous les services REST) par d'autres approches. Nos conclusions indiquent que le remplacement du HTTP/TCP par le CoAP est intéressant dans les réseaux tactiques, car le CoAP est moins coûteux et présente de meilleures performances générales, et ce, dans des conditions de bande passante très limitée dans lesquelles les solutions TCP peinent. Généralement, le mécanisme de retransmission TCP contribue à engorger la liaison sur une voie étroite, car un long délai peut être identifié comme une perte de paquet et déclencher des retransmissions. En pareil cas, les communications UDP augmentent habituellement la quantité de paquets livrés. C'est pourquoi, par exemple, le CoAP (basé sur l'UDP) est moins cher que le HTTP/TCP pour les liaisons à bas débit. Le rapport traite de ces découvertes en détail.

NATO CORE SERVICES PROFILING FOR HYBRID TACTICAL NETWORKS

1.0 INTRODUCTION

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 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 to the "central coordination level", as well as develop a superior and detailed understanding of the situation, to allow appropriate efforts and focus, the right decisions to be made and orders to be disseminated across the various levels. A challenge here is the need to make available and process possibly large amounts of information centrally and in near real-time. This requires robust, precise and efficient communications and accompanying information infrastructure, able to disseminate orders across the different levels. Conversely, decentralized C2 has less direct control and more emphasis on delegating responsibilities and operating according to the commander's intent. Here, coordination occurs at "lower levels", which enables a more rapid pace in the operation, however also requiring disseminating more information across different levels.

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 stove-piped platform-based thinking, doing away with silo systems, and a goal to organize all available resources for maximum effect. In addition to technological enhancements on communications and networking, NNEC brought innovative thinking in the way C2 can be conducted, suggesting that one can shift between centralized and decentralized C2 approaches, in order to select one that is appropriate for a specific mission and circumstances. The ability to transition between C2 approaches has been defined as C2 Agility by the NATO SAS-065 [4]. The vision of a fully decentralized C2 was described in the form of edge organizations in "Power to the Edge" [5], involving a broad distribution of information, unconstrained patterns of interaction, unlimited collaboration (including sharing of resources) and a widely delegated mandate for making decisions. For this to happen, communications and information exchange needs to be provisioned at the various C2 levels, including at the technically challenging tactical networks. The Research Task Group (RTG) NATO STO/IST-150 "NATO Core Services profiling for Hybrid Tactical Networks" (IST-150) targets specifically tactical networks. This work is important to support the future of C2 systems, which with the need to 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 both the request/response and the publish/subscribe communication paradigm. This report documents the work the group has performed within this area.

1.1 Importance of SOA and FMN for IST-150

Service-Oriented Architecture (SOA) is a paradigm for how to build highly interoperable distributed systems and is within NATO recognized as a key enabler for building federated systems. Both the NATO Network Enabled Capability (NNEC) and Federated Mission Networking (FMN) visions rely on the SOA paradigm for the technical integration of software components (services and applications) and federation of systems.

Core Services can be seen as a common enabling layer of services and are included in the so-called SOA platform services in the C3 Taxonomy. These services provide basic building blocks to support execution, monitoring, and control of other functional services, information sharing, and security in a SOA environment.

1.1.1 Group Focus

The current focus in FMN is on achieving interoperability between static and deployed systems, and the technical solutions used have not been evaluated for use in tactical networks. It is important to investigate if and how the current Core Services can be deployed and made to work in an FMN context. By enabling this, we can increase the level of future mission interoperability. The IST-150 research task group addressed Core Services in hybrid tactical networks. The group's focus was on NATO Core Services, being deployed on hybrid tactical networks including wireless communication links (e.g., based on military radios, satellite, LTE, and other wireless carriers). Scientific achievements and outreach activities were contributions in form of peer-reviewed conference papers and organized workshops on tactical SOA at international, IEEE-affiliated conferences.

1.1.2 Outcome and Target Community

The results of this task group are intended to be used as a guideline for nations that want to implement support for FMN-related services in the tactical domain. Furthermore, the work should be seen as input to FMN in its later spirals when they extend their scope to include mobility. As such, this report should be of interest and support to ongoing discussions and activities in the Core Enterprise Services Syndicate and the Tactical Edge Syndicate that both are working towards interoperable solutions for future FMN spirals. Any such exploitation of this group's work will be at the discretion of the FMN community.

1.1.3 Experiments Targeted the Messaging Core Service

In predecessor groups, IST-090 and IST-118, we identified SOA challenges and which Core Services we considered essential to support on the tactical level. There, we experimented with a wide array of different services and technologies. IST-150 continues this effort, with a more targeted approach in addressing aspects related to the Messaging Core Service for tactical networks. We have dedicated a high focus on the publish/subscribe paradigm, while also conducting additional work on request/response services.

Next, we will briefly discuss these types of services, while the remainder of the report details the conducted experiments and derived findings.

1.2 Publish/Subscribe

In order to support the publish/subscribe messaging pattern, NATO has pointed to the WS-Notification family of standards. This standard supports both the direct and the brokered publish/subscribe patterns, as illustrated in Figure 1. Other industry standards, notably Message Queuing Telemetry Transport (MQTT), which we have been evaluating alongside WS-Notification, support only the brokered (or multi-brokered) approach.

The direct message exchange, in which the information producers communicate directly with the information consumers require both producer and consumer systems to support the publish/subscribe pattern and protocol. In addition, this direct exchange of information typically means that multiple copies of the same information are sent all the way from the producer to the consumer. Brokerless publish/subscribe may involve exchanging information between producers and consumers using mechanisms like Peer-To-Peer (P2P) technologies, which do not rely on a broker.

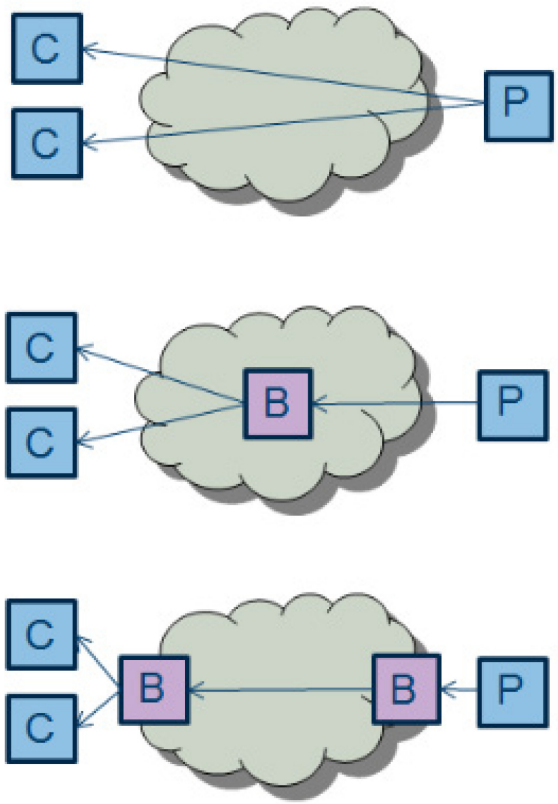


Figure 1: Publish/Subscribe Approaches, from Top to Bottom: Direct, Brokered, Multi-Brokered.

Brokered publish/subscribe involves introducing one or more intermediary nodes, which offload the information producers from such tasks as managing subscriptions and disseminating notifications. These brokers can be deployed in a number of different ways, ranging from a single broker deployment to a mesh of interconnected brokers.

The current NATO profiles from publish/subscribe services, such as the Service Interface Profile (SIP) included in the NATO Interoperability Standards and Profiles (NISP), do not mandate a given deployment strategy. Due to this, we have been pursuing multiple different approaches to publish/subscribe in IST-150.

In a publish/subscribe message exchange, there is also a need for sharing information about interests. When a subscription is created, the broker needs to know what type of information the consumer is interested in receiving. This can either be done by providing a set of keywords, called topics, which is checked against the message metadata every time a new message arrives at the broker. The other option is to use a content filter, which is a filter expression that is applied to the content of the message. In this latter case, the broker needs to understand the filter, read the entire message, and apply the filter to that message. All experiments we have performed in IST-150 typically use solutions based on topics. While technically both topic and content filtering are possible with WS-Notification, other, more common, industry standards like MQTT support only topics.

Experiments with publish/subscribe are further described in Section 3.0.

1.3 Request/Response

Request/response requires the clients to actively query a service for new data. The main difference between this and the publish/subscribe communication paradigm is illustrated in Figure 2.

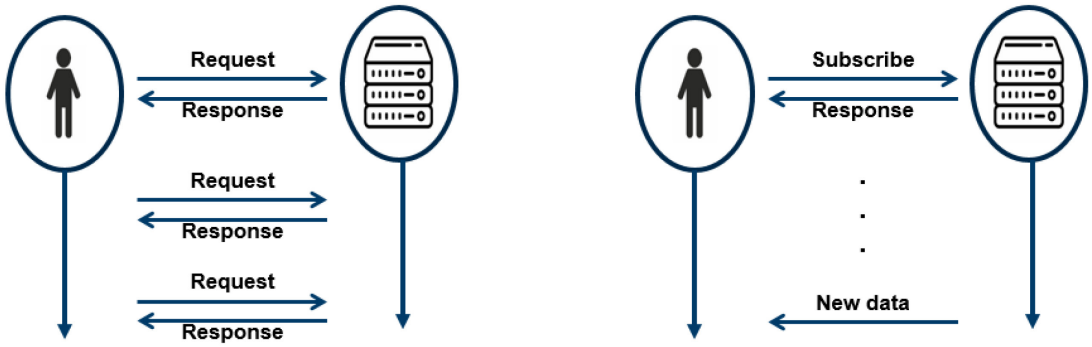


Figure 2: Request/Response vs. Publish/Subscribe (Left to Right).

Concerning request/response, NATO originally identified SOAP Web services as a technology enabler for interoperability [3]. However, as we have seen in later years, the civilian information technology systems are increasingly using the REpresentational State Transfer (REST) [6] flavour of Web services. Due to this, we have investigated proxy solutions that can support both SOAP and REST services across tactical networks. Further, we have addressed the possibilities of REST specifically, since these services have inherent lower overhead than SOAP services, as they need no meta-layer to encode SOAP messages. Instead, for REST, the connector is standardized on the HTTP protocol primitives, but it may also be used with other, possibly more efficient, drop-in replacements for HTTP, like CoAP. In our work, we have evaluated a REST based military messaging service extensively, including leveraging the CoAP protocol.

Experiments with request/response are further described in Section 4.0.

2.0 TESTBED

In this section, we describe the scenario, radio emulator and testbed frameworks and used for conducting the experiments described in this report.

2.1 Scenario

We use a subset of the Anglova scenario [7], Vignette 2 for our experiments. “The second vignette covers the deployment of the coalition forces, a battalion consisting of six companies, into the operational zone.” [8].

2.1.1 Adaptations of Anglova Scenario

During the development of tactical radio models, Swiss research establishments have adapted the Anglova scenario to provide a more realistic emulation of the scenario [9]. The adapted scenario is publicly available [10].

They observed that the 24 nodes used from the Anglova scenario Vignette 2 do not produce a challenging network topology. This is due to the rather short distances between the nodes throughout the scenario. The emulated vehicles move in the form of clusters, which leads to the situation where full connectivity is achieved with only one-hop during most of the emulation. Such conditions are not challenging in terms of multi-hop topologies where performance is relative to the number of hops. They therefore adapted the Anglova scenario in order to generate more hops between the nodes. This was achieved by decreasing the emulated output power to 5 W (37 dBm), which is often a tactical choice allowing lowering the possibility getting spotted by an enemy. Additionally, the locations of selected nodes were changed, so that during certain phases of the scenario, the topology also contains some chains. The average number of hops increased from 1.5 to around 2.5, whereas the maximum number of hops increased from 4 to 7 scenario [9].

The movement pattern of the vehicles in the original Anglova scenario and the adapted version is shown in Figure 8 [9].

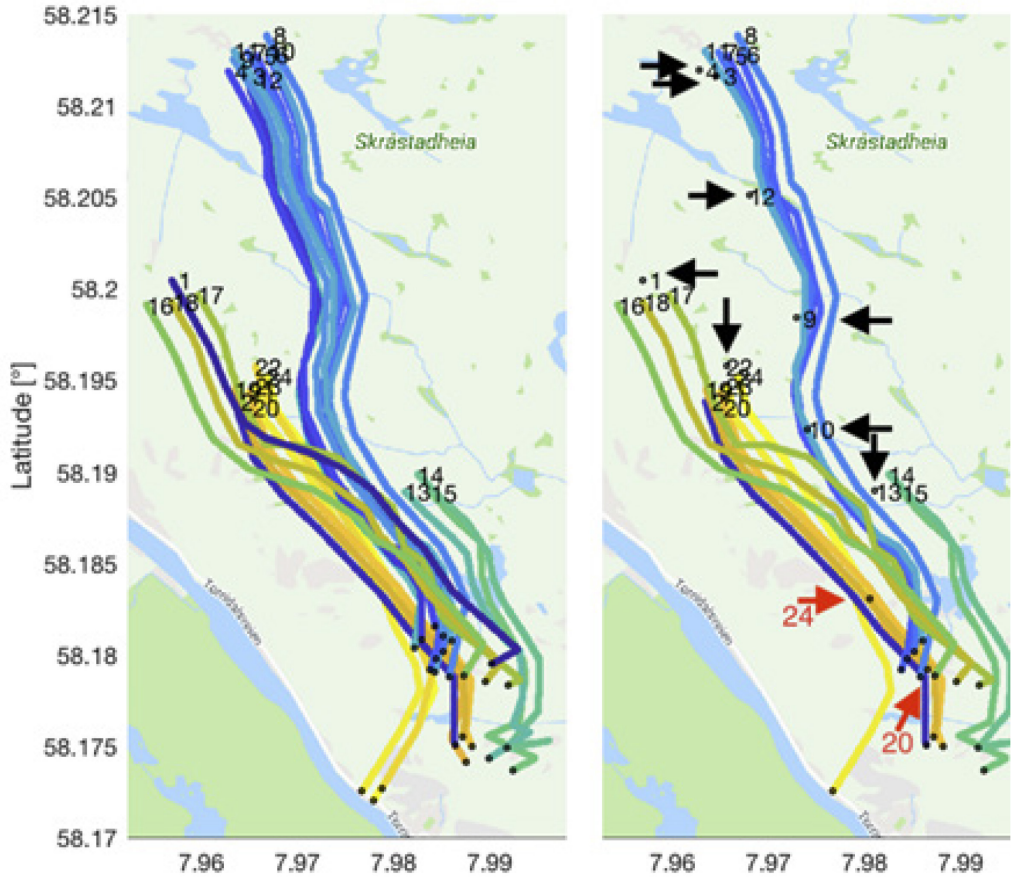


Figure 3: Movement of Vehicles in the Anglova Scenario (Left Hand Side) and Adapted Anglova Scenario (Right Hand Side).

2.2 Radio Emulation

For our experiments, we used the radio models from [10]) which emulate two waveforms (a narrowband and a wideband waveform) of a modern tactical radio [11]).

While the radio models emulate the network layers (according to ISO OSI model) *Link Layer* (especially the sub layer *Medium Access Control, MAC)* and *Physical Layer* of the radio, additionally a realistic physical propagation model is needed, which describes the propagation of electromagnetic waves in a terrain.

We describe the used radio models first and the radio propagation model afterwards. The work to integrate these models into the AuT testbed are described separately in Section 4.3.3.

2.2.1 Radio Model for Two Tactical Waveforms

The authors of Ref. [11] noticed during the first experiments with EMANE [12] leveraging the standard Wi-Fi models used by the community, that the obtained results were not matching the performance of real tactical radios [11]. The Optimized Link State Routing (OLSR) routing tables as well as some performance metrics, such as throughput and latency between emulated nodes led to the two following conclusions:

1) The Wi-Fi models, although tunable, do not allow reproducing the latencies and throughput of real tactical radios. The obtained performance during the emulations is far too optimistic compared to the expected performance in a real deployment.
2) The Anglova Vignette 2 with Company 1 scenario (24 nodes) is not challenging enough, as most of the time the topology tends to be a full-mesh, whereas multi-hop topologies would rather be more realistic.

The combination of these two drawbacks leads to the situation where experiments do not reflect reality, as even heavy protocols, which were not working under lab conditions with real radios, show high performance in the emulated environment. In order to obtain more realistic emulations, they started by reproducing narrowband and wideband tactical radios in EMANE. Their performance (throughput and latency) was measured under lab conditions with various *Received Signal Strength Indicators (RSSIs)*. In a second step, and with the information regarding the *Time-Division Multiple Access (TDMA)* schedules of the real radios, they elaborated TDMA scheduling models in EMANE. As shown in Ref. [11], they were able to reproduce in quite high fidelity the performance of the real radios, including the adaptive rate changing the performance according to the channel quality.

The work regarding adaptations of the Anglova scenario was already described in Section 2.1.

2.2.2 Propagation Model

For the calculation of the path loss between the nodes in Vignette 2 of the Anglova scenario, a radio propagation model based on the *Uniform geometrical Theory of Diffraction (UTD)* from Holm [13] is used. The model uses a digital terrain model to incorporate large scale fading effects (i.e., variations of the signal strength caused e.g., by obstacles between sender and receiver).

For this purpose, in the Anglova scenario the path loss between each pair of nodes was pre-calculated. These path losses are replayed during a scenario run (cf. Refs. [8], [14]). This is necessary because the model from Holm is too time-consuming to be executed in real-time. Since the unit movements are predefined in the scenario, this approach is feasible. The benefit compared to simpler models which can be executed in real-time is a more realistic emulation of the radio propagation.

2.3 Testbed Frameworks

2.3.1 ARL Testbed

The U.S. Army Research Laboratory (ARL) Network Science Research Laboratory (NSRL) is composed of a suite of hardware and software that models the operation of mobile networked device RF links through emulation (not merely simulation). The NSRL enables experimental validation or falsification of theoretical models, and characterization of protocols and algorithms for mobile wireless networks. It is used for a range of experiments, from assessing in-network aggregation of network information for detecting cyber threats, to characterizing the impact of communications disruption on perceived trust and quality of information metrics delivered to Soldiers in tactical mobile environments. Unlike other experimentation facilities for research in wireless networks, the NSRL is focused on Army-unique requirements such as hybrid networks and extensive modelling of environmental and urban effects on RF communications. The NSRL supports both investigation of traditional wireless networking challenges as well as more general network science research issues. The NSRL's emulation environment is result of collaborative efforts between ARL and the U.S. Naval Research Laboratory (NRL).

2.3.1.1 Overview of ARL Testbed

The NSRL provides a controlled, repeatable emulation environment for the research, development, and evaluation of network and information assurance algorithms for tactical wireless mobile ad hoc networks. Using NSRL's capabilities researchers can:

- Model link and physical layer connectivity in real-time.
- Implement actual network protocols and application software (mimics real-world mobile, wireless network systems).
- Provide event-driven control and logging facilities.
- Utilize distributed architectures for experimentation and analysis.
- Leverage larger ARL network science enterprise services including High Performance Computing (HPC) facilities and Research Development and Engineering Network (RDENet) Enclaves.

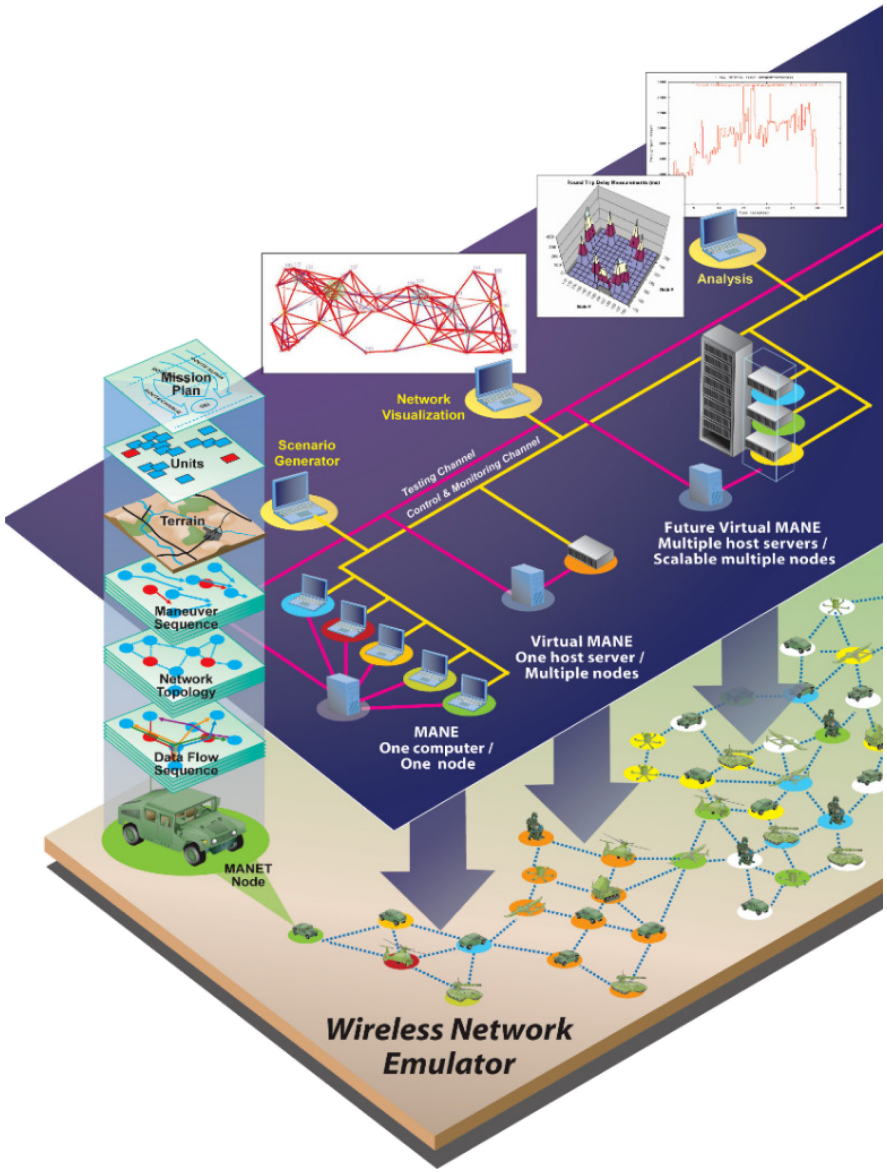


Figure 4: Network Science Research Laboratory Framework.

The NSRL RDENet capability is a vital link to our collaboration partners, providing information sharing and research integration opportunities not previously available to researchers. RDENet enables external research collaborators to remotely access the NSRL and facilitates connection of the NSRL to other ARL experimental labs and assets. RDENet currently connects the NSRL to sensors located throughout the ARL Adelphi Laboratory Center campus and integrates those sensors with current research programs supporting the soldier. The NSRL is also connected to HPC resources located at Aberdeen Proving Ground to extend wireless emulation, supporting research in the area of hybrid network models.

2.3.1.2 Dynamically Allocated Virtual Clustering Management System (DAVC)

DAVC [15] is one of the primary experimentation infrastructure components within ARL's Network Science Research Laboratory. DAVC allows researchers to dynamically create, deploy, and manage virtual clusters of heterogeneous nodes within a cloud-computing environment, abstracting away test bed configuration complexities by automatically assigning and configuring virtual cluster networks and network services such as DNSMASQ, DNS, DHCP, TFTP. Virtual clusters deployed within DAVC can be utilized for a wide variety of tasks such as software development, experimentation, and existing hardware/software integration. DAVC enables researchers to configure robust networking scenarios and complex subnet hierarchies within each cluster, where each cluster is assigned private Virtual Local Area Networks (VLANs) which restrict network traffic within the boundaries of a specific cluster. This also eliminates undesirable crosstalk between clusters and researcher's experiments allowing for multiple experiments to be conducted simultaneously. DAVC ensures efficient utilization of hardware resources by interfacing with Oracle Grid Engine to dynamically assign each virtual node to virtual host server hardware based on CPU, memory, hard disk and network utilization.

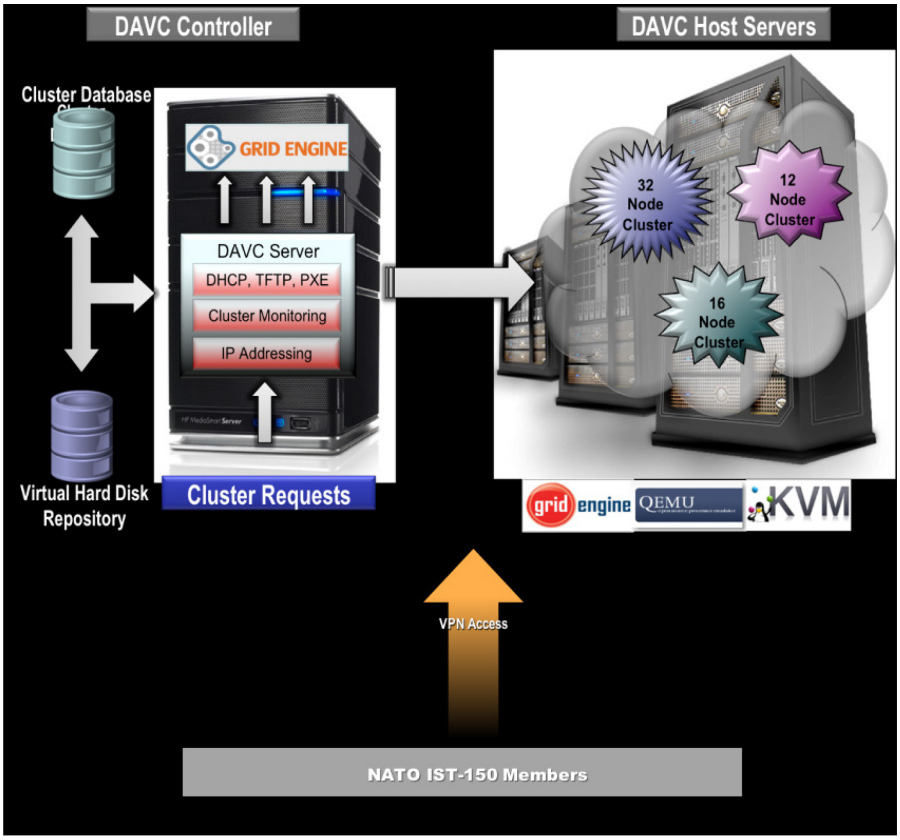


Figure 5: DAVC Architecture.

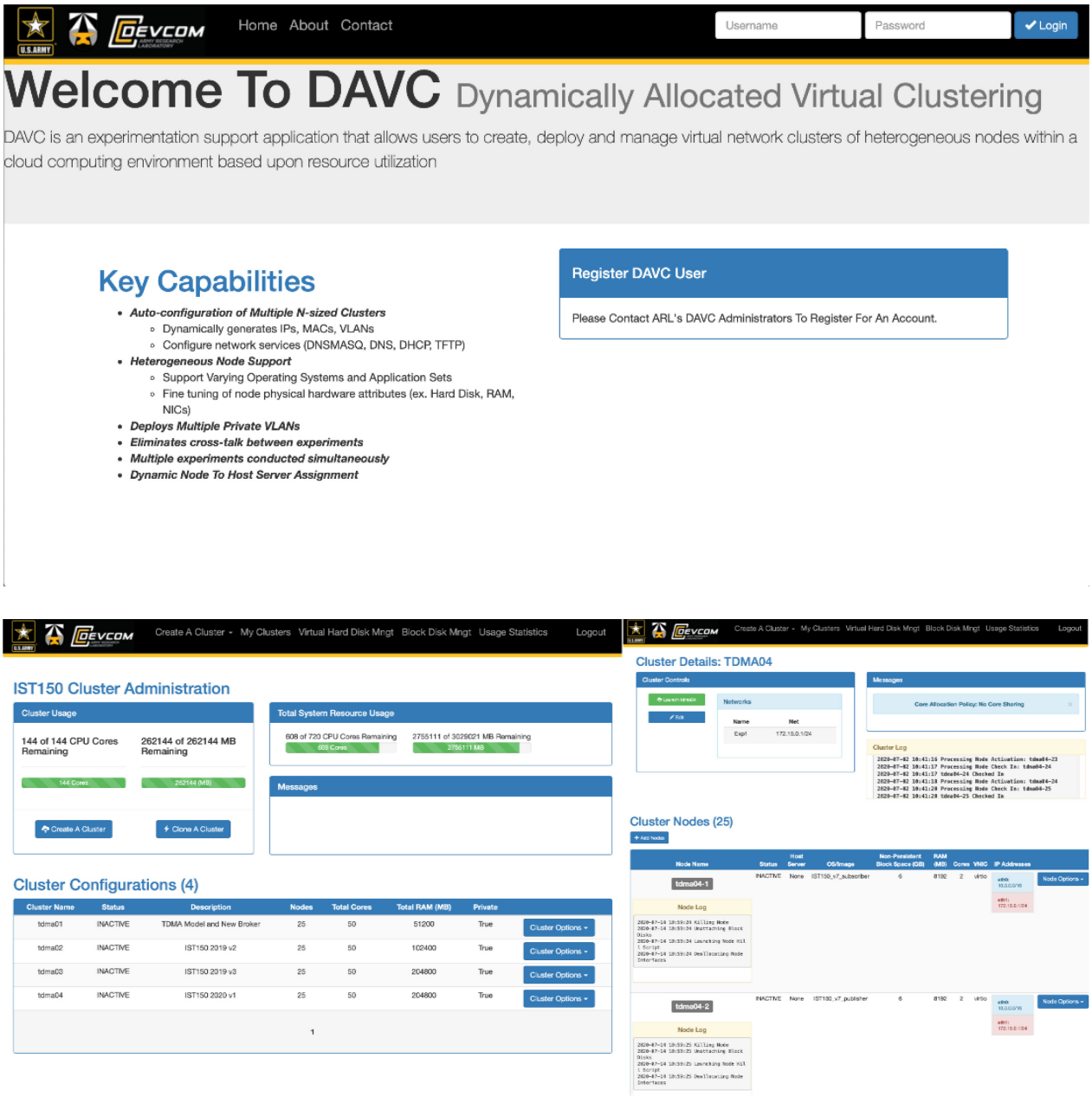


Figure 6: DAVC Web Interface.

2.3.1.3 ARL Experimentation Infrastructure

The ARL experimentation infrastructure embodies the concept of Experimentation-as-a-Service (EaaS) for provisioning reconfigurable, ad hoc and on demand experimentation environments. The EaaS infrastructure makes use of open standards and open source technologies, as well as assets developed by ARL and our research partners, and has been continually updated to serve as the core infrastructure for experimentation in NSRL.

The EaaS infrastructure comprises a layered architecture and a software stack that enables the provisioning of reconfigurable ad hoc and on demand experimentation environments. It is based on standard hardware and open standards software, and also allows for the integration of externally connected Commercial

Off-The-Shelf (COTS) resources and assets such as sensors, mobile devices, radios, Internet of Things (IoT) devices, Artificial Intelligence/Machine Learning (AI/ML) resources, databases and applications, all interconnected via real and emulated networks.

The primary tools used for experimentation are the NRL-developed EMANE [12] for network emulation and the previously mentioned ARL-developed DAVC for experiment configuration and appropriation of compute and connectivity resources. ARL has developed tools to aid the experimentation process, such as TrafficGen and the ARL Visualization Framework (ARLVF). TrafficGen is a visual timeline interface used to create and edit MGEN-format files to create realistic network traffic scenarios, while ARLVF provides an open, publish/subscribe mechanism based on ZeroMQ for developing visualizations and connecting those visualizations to data feeds.

ARLs use of standard hardware and open standards as well as open source software has helped ARL's researchers and their collaborators to evolve the experimentation infrastructure to support ever changing research needs, while also controlling costs. An added benefit to this approach is the high rate of technology transition between government, industry, and academic partners.

2.3.2 AuT Testbed

Measuring the performance of a single messaging service in a lab environment will not indicate how multiple instances of the service deployed together with tactical radio systems in military vehicles will perform in a realistic military scenario. This is the case, because typical lab experiments do not take the dynamic environment into account and are poorly scalable.

Instead, a whole combination of different systems (IT and communications systems) has to be taken into account. For the systems under test – i.e., messaging services in this case – the original software (or virtualized versions) should be run in order to represent the real systems in as much detail as possible. Systems which cannot be virtualized, because the software is not publicly available (e.g., military radios), can be emulated by means of real-time radio simulators (emulators) with realistic radio models.

In this section the testbed which was used for the evaluation of Military Messaging service is described. The testbed is based on Analyse and Test environment (AuT) (cf. Ref. [16]).

2.3.2.1 Overview of AuT Testbed

AuT is a framework for tactical testbeds which can be used for realistic experiments with a combination of IT and communications systems. The main components of AuT are shown in Figure 7. The IT components are deployed as Virtual Machines (VM). These include virtualized command and control systems and tactical routers. VM templates of these systems are available in AuT as a kit for testbed instances. Furthermore, tactical radio networks are emulated including the dynamics of the terrain and the movement of units (cf. "Virtualized Testbed" in Figure 7). An administrator can define operational scenarios with the help of a scenario editor (cf. "Scenario Editor" in Figure 7). Scenarios are stored in a "Scenario Data Base" and thus are available for repeatable tests which are executed by the "Management" component. The movement of units is simulated by a tactical simulator (cf. "TacSim" in Figure 7). For each test run, an analysis of application and network data is conducted and can be visualized by an "Analyser" component with a graphical user interface.

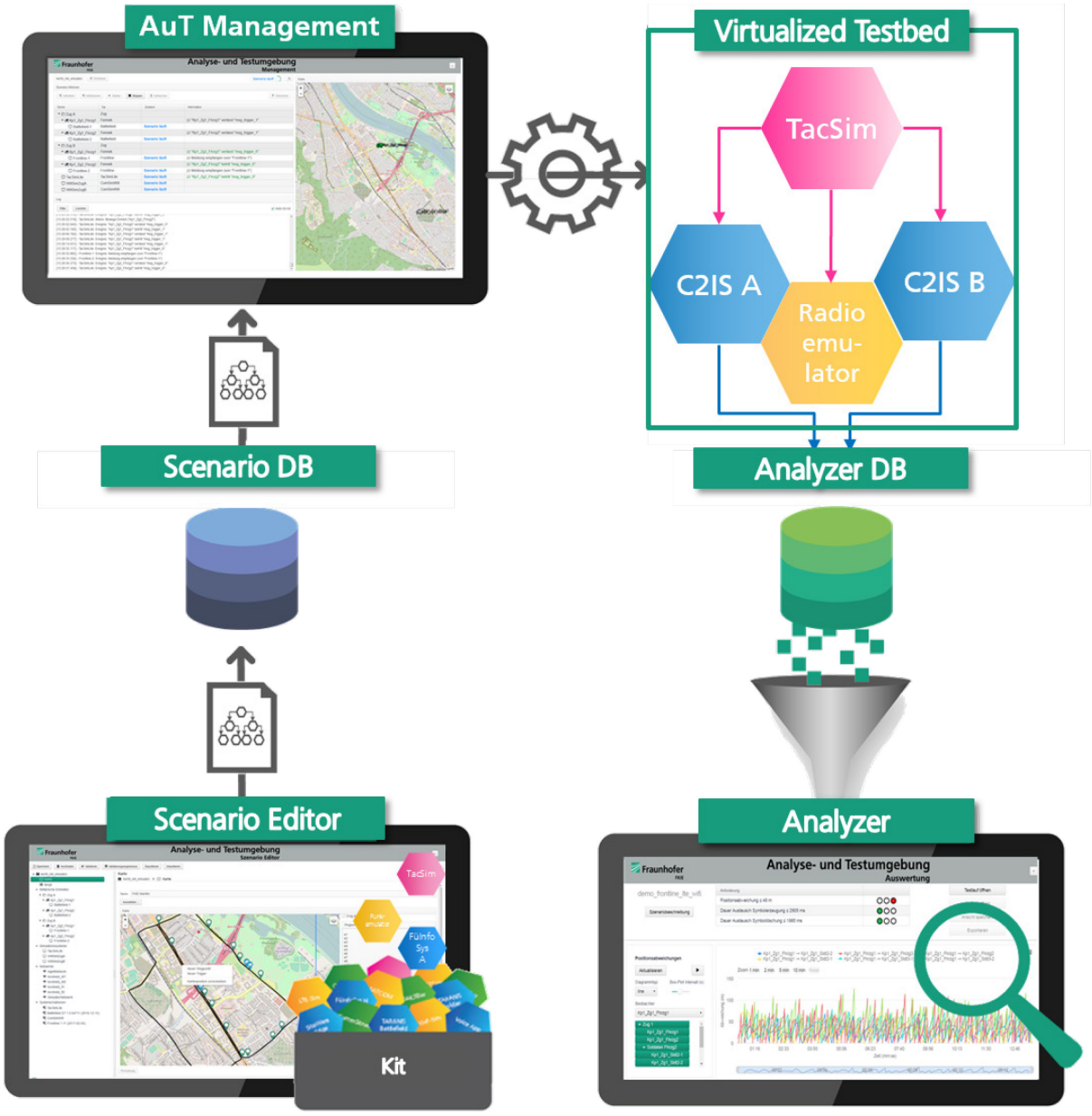


Figure 7: Overview of AuT Components.

2.3.2.2 Analysing Tools

For the analysis of the experiments, we used analysing tools from the Analyse and Test environment (AuT). AuT allows generating suitable metrics for military applications, which are relevant for an assessment in tactical networks. In Hirsch et al. [17] the concepts of AuT for the analysis of experiments are described. These include a mechanism for capturing and analysing runtime data of C2 systems in tactical networks, the specification of suitable metrics for military applications and the definition of different visualizations based on these metrics.

The evaluation approach uses captured information from the network and the application layer.

Logging on Network Layer

For logging of the network traffic, the freely available tool *Wireshark* [18] is used. This allows an administrator to capture all IP packets and analyse them w.r.t. the transmission times and the count of lost messages. Furthermore, for supported protocols, Wireshark provides additional information such as the number of retransmissions or the used bandwidth of the different protocols. Thus, e.g., the impact of the routing protocol on the total resource utilization can be assessed. Additionally, the exact size of the transmitted messages can be determined. We are also working on an extension of AuT to analyse these network captures in an automated way based on Kafka streams.

Logging on Application Layer

On the application layer, to receive accurate data on the whole lifecycle of a message, we define four measuring points, two each for sender and receiver. On the sender side, the times *triggered* and *sent* are logged. *Triggered* means the moment when a user triggers the sending of a message, e.g., by clicking on a button, whereas *sent* means the time when the message was actually sent. The difference between these times is typically small, but there might be a delay due to the behaviour of the system. On the receiver side, received depicts the moment when the message is received, while *processed* describes the moment when the message is shown in the user interface of the system.

The following data is logged for each message:

- The **timestamp** indicating when the message was triggered, sent, received or processed;
- The **system instance** which did receive or sent the message;
- The **status** (triggered, sent, received or processed); and
- An **identifier** of the messages.

The identifier of a message is needed to map sent with received messages.

Analysis of Test Runs

After the execution of an experiment, AuT performs a post-processing of the logged application data. In this process, the correlation of sent and received messages is determined and the results (e.g., transmission times) are stored in a data base.

After this post-processing, tools for the analysis of the logged data are used. On the network layer an analysis is conducted with help of Wireshark. On the application layer the data can be visualized by the Analysis GUI. Alternatively, a tool for generating boxplot diagrams to visualize the transmission times can be used. This tool uses the *R library for statistical computing and graphics* [19].

In the experiments, we mainly assess the performance of the overall combined system by the transmission times and the loss rates of the messages instead of their message size since these are the only relevant performance properties from a user perspective. The message size may however have an important impact on the transmission times and reliability of the transmission if narrowband networks are deployed. This effect will also be visible by evaluating the transmission times and lost messages.

3.0 PUBLISH SUBSCRIBE

3.1 Introduction

This section focusses on the analysis of Message-Oriented Middleware (MOM) Service, which includes the publish/subscribe communication paradigm, for timely exchange of data in tactical environments that

have communications limitations like disconnections from other levels, intermittent connectivity between peers in the tactical domain itself, and limitations on throughput and available bandwidth (i.e., Disconnected, Intermittent and Limited (DIL) networks). It presents the IST-150 group's findings in the possible realization of MOM service using WS-Notification and the industry standard Message Queuing Telemetry Transport (MQTT) [20], and our attempts at employing them for (emulated) tactical communications for a Blue Force Tracking (BFT) application. Our findings result from a series of experiments, presented in scientific conferences and symposia, that are summarised in this section.

It should be noted that we have shown that, irrespective of which topic-based publish/subscribe protocol is used, it is possible to achieve interoperability between different solutions through a multi-protocol broker. This means that even if one is to adopt MQTT (or another protocol) at the tactical level, it is still possible to federate this with other protocols in other networks, like WS-Notification [21]. This means full flexibility in coalition networks with different capacities and allows using different solutions across heterogeneous networks.

This section starts by introducing related and previous work conducted in the field by IST-150 members. It continues by presenting experimentation work conducted to analyse the application of WS-Notification and the MQTT technologies in tactical networks in a coalition environment. It concludes with the presentation of main findings, as well as recommendations for follow-on activities.

3.1.1 Related and Previous Work

As part of NATO IST-150 activities, we have analysed several standardized publish/subscribe technologies candidate for MOM service in a coalition tactical scenario.

There are many prolific publish/subscribe standards, which have been applied to a broad range of applications. For example, the Advanced Message Queuing Protocol (AMQP) [22] is much used in the finance sector as a reliable message queue for exchanging high volumes of transactions. The Extensible Messaging and Presence Protocol (XMPP) [23] is much used as a foundation for chat, but also offers generic publish/subscribe functionality. As such, it has been promoted as a potential carrier for sensor data on the Internet of Things (IoT). Another standard of importance is MQTT, which is the underlying protocol of choice for popular messaging apps since they require an efficient one-to-many dissemination mechanism for their users. WS-Notification [24], a SOAP-based standard from OASIS related to Web services as defined by the World Wide Web Consortium, is NATO's choice for interoperable publish/subscribe [1].

Work conducted by Bloebaum and Johnsen [25] tested AMQP, MQTT, and WS-Notification in a small-scale deployment (3 nodes) using real tactical radios, where MQTT was found to show promise, while AMQP offered reliable communication, but was less efficient than MQTT. Karagiannis et al. [26] have performed a survey of relevant IoT data protocols with respect to IoT specifically, where they considered such publish/subscribe protocols as XMPP, MQTT, and AMQP. Also, they considered non-publish/subscribe approaches like CoAP, REST, and Web sockets.

Through a series of experiments [25], [27], [28], [29], we have found that MQTT emerges as an interesting alternative to WS-Notification, since it is also an industry standard, but its low network overheads makes it a better match to cope with the limitations of tactical networks [30], [31].

Manso et al. [29] provided an overview of similarities and differences between these two standards, herein presented in Table 1.

Table 1: Feature Comparison Between the WS-Notification and MQTT Standards. Source (Manso et al., 2018 [28]).

Property	WS-Notification	MQTT
Protocol stack	SOAP/HTTP/TCP	TCP
Payload format	XML	Payload agnostic
Quality of service	None built in, but can use additional WS-* standards, e.g., WS-ReliableMessaging	Three delivery semantics: Best effort, At-least-once, or At-most-once delivery
Usage	NATO	IoT, sensor networks, etc.
Topologies supported	Direct and brokered	Brokered
Standardization	(OASIS, 2006)	(OASIS, 2015)

Since WS-Notification is based on XML and SOAP, it makes it a more resource demanding protocol than MQTT, which is built directly on TCP. As such, WS-Notification consumes more networking resources than MQTT.

Next, the results of our experiments in applying WS-Notification and MQTT in a simulated setting and subsequent comparison are presented. Given MQTT lightweight approach to publish/subscribe, a dedicated section follows that analyses the performance of different MQTT configurations.

3.2 WS-Notification and MQTT: Experiments and Results

In Manso et al. [29] we have applied WS-Notification and MQTT as the information dissemination mechanism in experiments and exercises enabling publish/subscribe-based tactical level data to experimental C2 systems. This subsection describes the scenario, setup and results of the experiments.

3.2.1 Scenario

We used the Anglova military scenario that *includes detailed mobility patterns for a battalion-sized operation over the course of two hours, which has been developed by military experts in planning and performing real exercises* [14]. Specifically, we employed Vignette 2 of the Anglova's scenario limited to one mechanized battalion constituted by 24 mobile nodes (military vehicles) that is part of a Military Contingent coordinated by the Coalition HQ. We selected the sharing of friendly force information a.k.a. Blue Force Tracking (BFT) as a service to simulate. The NATO Friendly Force Information (NFFI) data format, described in draft STANAG 5527, is used in the BFT service, thus representing a standard payload in the publish/subscribe evaluation.

3.2.2 Experimental Testbed Setup

The experimental testbed used to conduct experiments is the Network Science Research Laboratory (NSRL) [32] (see Section 2.3.1) established by the U.S. Army Research Laboratory (ARL). The NSRL was used for network emulation and scenario reproduction. The Anglova scenario, incorporating WS-N or MQTT broker messaging services, was setup in the NSRL environment. For that, WS-N and MQTT services were installed onto the Virtual Machine (VM) template of the Anglova scenario to enable the publish/subscribe position location information services. The experiments use the **single broker topology** described in Section 3.2.2.1.

For network emulation, we used the Extendable Mobile Ad hoc Network Emulator (EMANE) that provides – besides the emulation of the radio links – signal propagation and mobility representation to the experiment to create a more realistic environment. The mobility information was drawn from Anglova recorded data. The emulation allows for various types of routing and radio models to be used; in this scenario we used Optimized Link State Routing (OLSR) V1 [33] via the OLSR Daemon (OLSRD) on each virtual machine representing a node in the scenario with wireless links based on the EMANE RFPipe model. The RFPipe model was configured to emulate wideband tactical radios operating at 300 MHz with a 250 KHz bandwidth and 175 kbit/s data rate. OLSR was configured with a Hello Interval of 2 seconds, Hello Validity Time of 20 seconds, Topology Control Interval of 8 seconds, and Topology Control Validity time of 80 seconds.

In the initial set of experiments, we ran the first 30 minutes of the Anglova scenario vignette excerpt consisting of 24 nodes. We set up a DAVC cluster of 24 "Anglova" nodes and one controller node. Node 1 for this experiment is arbitrarily established as the broker node (i.e., runs the WS-N or MQTT server).

For our scenario, we set the publishing of the node locations (i.e., NFFI messages) every 10 seconds. In this experiment, we have Nodes 2 through 24 as publishers. See Figure 8 for a depiction of the network experiment architecture.

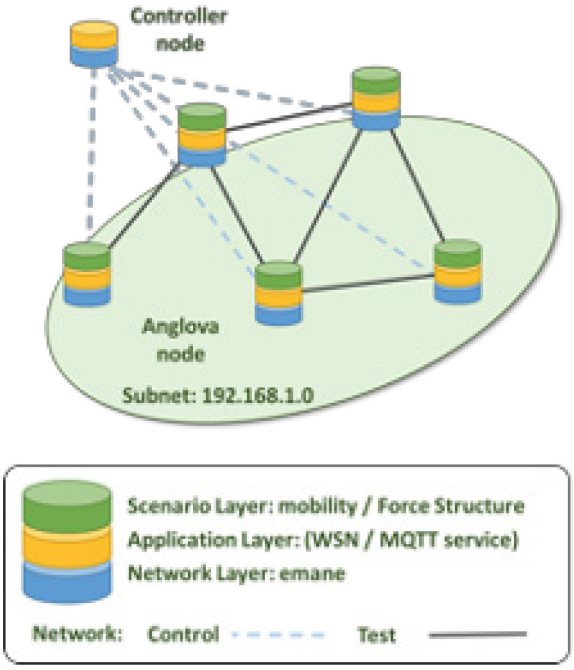


Figure 8: Architecture of Network Experiment Including Network Emulation, Application and Scenario Layers.

3.2.2.1 Publish/Subscribe Software

The publish/subscribe message broker middleware selected for this work is the following:

- For **WS-Notification broker**, we used *microWS-N*, which is a closed source Norwegian Defence Research Establishment (FFI) implementation of a subset of the WS-Notification family of standards. This implementation has been tested for interoperability at the NATO Coalition Warrior Interoperability eXploration, eXperimentation, eXamination, eXercise (WIX) in 2014. There, we found that the standard functions microWS-N offers were compliant with WS-Notification version 1.3, which is the most recent specification [34].

- For **MQTT messaging broker**, we used the open source Mosquitto from the Eclipse foundation, which is freely available online [35]. It should be noted, though, that Mosquitto has some stability issues, notably when used together with Transport Layer Security (TLS) and Web sockets. At the time of writing this section, these issues are known but still unresolved[1]. So, to ensure the stability of our experiments, we used Mosquitto without TLS enabled to avoid crashes and we made the assumption that security (as in confidentiality and integrity) would be ensured at the radio and network levels (e.g., through IP-Sec or link layer encryption). We also excluded the use of Websockets in the experiments.

In addition to the brokers, we also needed to implement producers and consumers to use in the evaluation:

- For WS-Notification, we used the closed source client libraries of microWS-N as the basis for setting up subscriptions and publishing data.
- For MQTT, the producer and consumer software were implemented using the Fuse source library [36]. Since messages relate to location information periodically produced, the MQTT clients were configured to request at-most-once delivery from the broker (i.e., MQTT QoS = 0 that is the most efficient but least reliable setting).

As explained in Section 3.2.2, Node 1 functions as broker node (i.e., runs the WS-N or MQTT server) and a consumer node (i.e., runs the consumer software subscribing to all messages). Nodes 2 to 24 run the producer software that publishes a NFFI message each 10 seconds.

3.2.3 Experiments Results and Evaluation

The results and evaluation of the experiments are presented in detail in Ref. [29]. Here, we present their highlights.

3.2.3.1 WS-N with OLSR and Broadband Radio Links

This setup involved the deployment of the WS-N broker *microWS-N* together with one WS-N subscriber on node 1 (the HQ node). Nodes 2 to 24 (23 nodes in total) each run a WS-N producer software publishing a NFFI message every 10 seconds. The average transmission times (in seconds) of NFFI messages is presented in Figure 9, showing the first quartiles, medians and third quartiles.

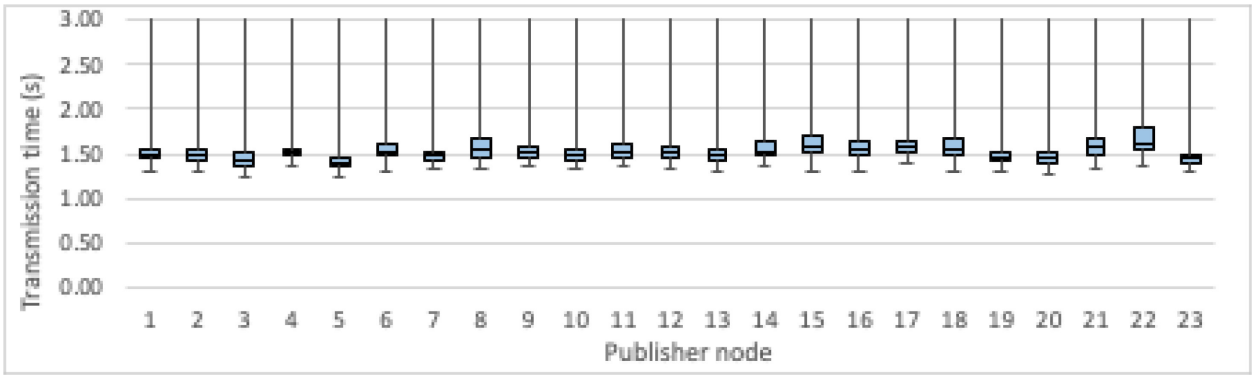


Figure 9: Transmission Times of WS-N-Based NFFI Messages (Enlarged View).

In all, 2955 messages were published. None of these were lost. The overall median was 1.5 s. For each publisher, the median transmission time was between 1.4 s and 1.7 s as shown in Figure 9.

[1] Mosquitto segmentation fault during client connection: https://github.com/eclipse/mosquitto/issues/406.

3.2.3.2 MQTT with OLSR and Broadband Radio Links

This setup involved the deployment of the MQTT broker *Mosquitto* together with one MQTT subscriber on Node 1 (the HQ node). Nodes 2 to 24 (23 nodes in total) each run an instance of the MQTT producer software publishing a NFFI message every 10 seconds. The average transmission times (in seconds) of NFFI messages is presented in Figure 10, showing the first quartiles, medians and third quartiles.

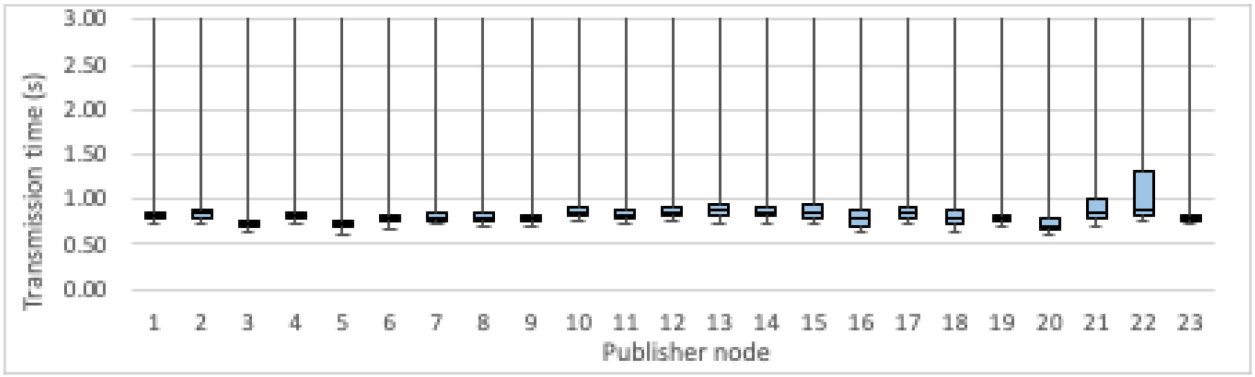


Figure 10: Transmission Times of MQTT-Based NFFI Messages (Enlarged View).

In all, 3073 messages were published. None of these were lost. For each publisher, the median transmission time was between 0.7 s and 0.9 s as shown in Figure 10. The overall median was 0.8 s.

3.2.3.3 Comparison Analysis and Results

A comparison between results obtained with WS-N and MQTT is presented next. The measurements used to support our analysis are presented in Table 2.

Table 2: Results from Experiments for WS-N and MQTT.

	WS-N	MQTT
Network Layer		
Data rate (kbit/s)	42	23
Message size (bytes)	1939	909
TCP retransmissions	3594	1954
Application Layer		
Messages lost	0	0
Delay (median) (sec)	1.5	0.8
Maximum Tx Time (sec)	86	92

From the evaluation of the experiments with WS-N and MQTT as message brokers, it can be seen that MQTT outperforms WS-N:

- In overall (including the whole communication stack) MQTT consumes about half the data rate than WS-N (23 kbit/s vs. 42 kbit/s) of the available data rate of 175 kbit/s which is provided by the radios.

- MQTT generated message size is less than half the WS-N's message size (909 bytes vs. 1939 bytes).
- MQTT caused about half TCP retransmissions than WS-N (1954 vs. 3594).
- Consistently, the median message transmission times measured on the application level were half as large with MQTT (0.8 s) compared to WS-N (1.5 s).
- A few large delays were observed, being the maximum observed pertaining to MQTT (92 seconds) closely followed by WS-N (86 seconds). These, however, seem more related to TCP protocol or networking aspects, and not associated to the message broker.

As expected, MQTT exhibits a "lighter" and more efficient network performance than WS-N, which makes it suitable for mobile tactical environments, where network resources are scarce. Additional optimisations and configurations can be pursued aiming to further improve network performance.

Concerning network-related measurements, we also noted that the OLSR routing protocol generated a large amount of data rate volume (70% and 80% of the data rate for WN-S and MQTT, respectively), thus OLSR improvements (e.g., different protocol update rates) should be investigated or, otherwise, alternative routing protocols better suited for tactical mobile environments using wideband (or narrowband) radios should be deployed. Moreover, albeit TCP assures delivery of all messages, it was noted that both WS-N and MQTT setups produced many "spurious" TCP retransmits[2]. This indicates that TCP is not well suited for the kind of wireless networks used in this scenario. Thus, alternative transport protocols should be sought, such as UDP.

3.2.4 Conclusion

The analysis taken from these experiments allowed us to conclude that WS-N requires more network resources than MQTT to achieve the same functionality. This leads to increased network resource use (about twice compared to MQTT), as well as an increased transmission time (also about twice) of end-to-end messaging. We can conclude, that for the part of the scenario we evaluated, MQTT was the superior protocol based on the considered metrics.

Our analysis also showed that the used network protocols, specifically OLSR and TCP, also play a significant role regarding the use of network resources: OLSR generated 70% or 80% of the overall traffic for WS-N and MQTT respectively, while TCP produced many "spurious" packet retransmissions. There is a need to investigate optimisations or even alternative protocols that are better suited for tactical mobile networks (e.g., UDP replacing TCP).

While our analysis is based on two specific implementations (i.e., FFI *microWS-N* and Eclipse Mosquitto) and that different implementations may yield different results, the overall differences between WS-Notification and MQTT should still be evident due to the differences between these standards.

3.3 MQTT-Based Multi-Broker Experiments and Results

Given the promising results obtained for MQTT technology for MOM Services, we chose to further analyse its performance in the context of a federated deployment (approach consistent with a coalition mission) based on multi-broker (or brokerless) deployments.

[2] "Spurious" means that a packet was unnecessarily retransmitted, because the respective acknowledgement arrived too late at the sender. Since the congestion control mechanism of TCP interprets "lost" (actually belated in this case) acknowledgements as buffer overflows, the congestion window is unnecessarily decreased, which leads to a reduced throughput.

3.3.1 Scenario

Two different scenarios were created for these experiments:

- MULTINATIONAL SETTING 1: This setup involves two nations – named PRT and NOR – each deploying a convoy comprising 8 mobile units. In order to develop a complete shared situational awareness, nations agree on exchanging BFT messages between all their units. In this setting, each nation manages its own message broker and all nations agree on an appropriate multi-broker setup allowing exchanging topics and messages.
- MULTINATIONAL SETTING 2: This setup involves four nations – named DEU, PRT, NOR and USA – each deploying a convoy comprising 8 mobile units (cf. Figure 11). The intent to share position information, as well as having each nation managing its own message broker, is the same as for the multinational setting 1.

In this section, we only include results related with MULTINATIONAL SETTING 2. Please see Ref. [37] for the complete analysis.

The publish/subscribe paradigm operates based on the definition of *topics*, which typically are string based keywords (i.e., UTF-8 strings) that are attached to the messages as metadata. MQTT does not have a formal way to describe its topic structure. It uses a simple, but highly expressive topics structure, where more advanced topics can be formed using a (hierarchical) multi-level structure, where each level is separated by a forward slash.

Manso, Brannsten, and Johnsen [38] proposed a topic structure in the context of a deployed force by a single-nation that is herein adapted considering a coalition environment:

```
coalition-Id/country-Id/unit-Id/entity-Id/service-Type
```

Where:

- *coalition-Id* uniquely identifies the coalition.
- *country-Id* uniquely identifies the country that is part of "coalition-Id". For example, according to the NATO STANAG 1059, "NOR" is used for Norway.
- *unit-Id* is an arbitrary string that uniquely identifies the unit (or group of entities) that belongs to country-Id.
- *entity-Id* is an arbitrary string that uniquely identifies an entity (e.g., a soldier or a vehicle) that belongs to "unit-Id".
- *service-Type* is a string that uniquely identifies the type of service provided by or associated with entity-Id. For example, in this paper we use the "location" topic to publish information pertaining to the unit's location. Other topic names representing services associated with a unit could be "health_status", "ISR_report" and "chat".

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
```

In addition to defining the topic structure, the exchanged messages' structure also needs to be defined and agreed by coalition partners, ensuring that publishers know what should be published and that subscribers are

able to "decode" and process them. Herein, we opt to continue with our approach in adopting Web-friendly technologies and formats to continue with the use of the general-purpose standard for location information GeoJSON [39]. As we already demonstrated in Ref. [28], GeoJSON can be used to share location information related with each unit.

3.3.2 Experimental Testbed Setup

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 setting for MULTINATIONAL SETTING 2 is described next.

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 network setup is depicted in Figure 11.

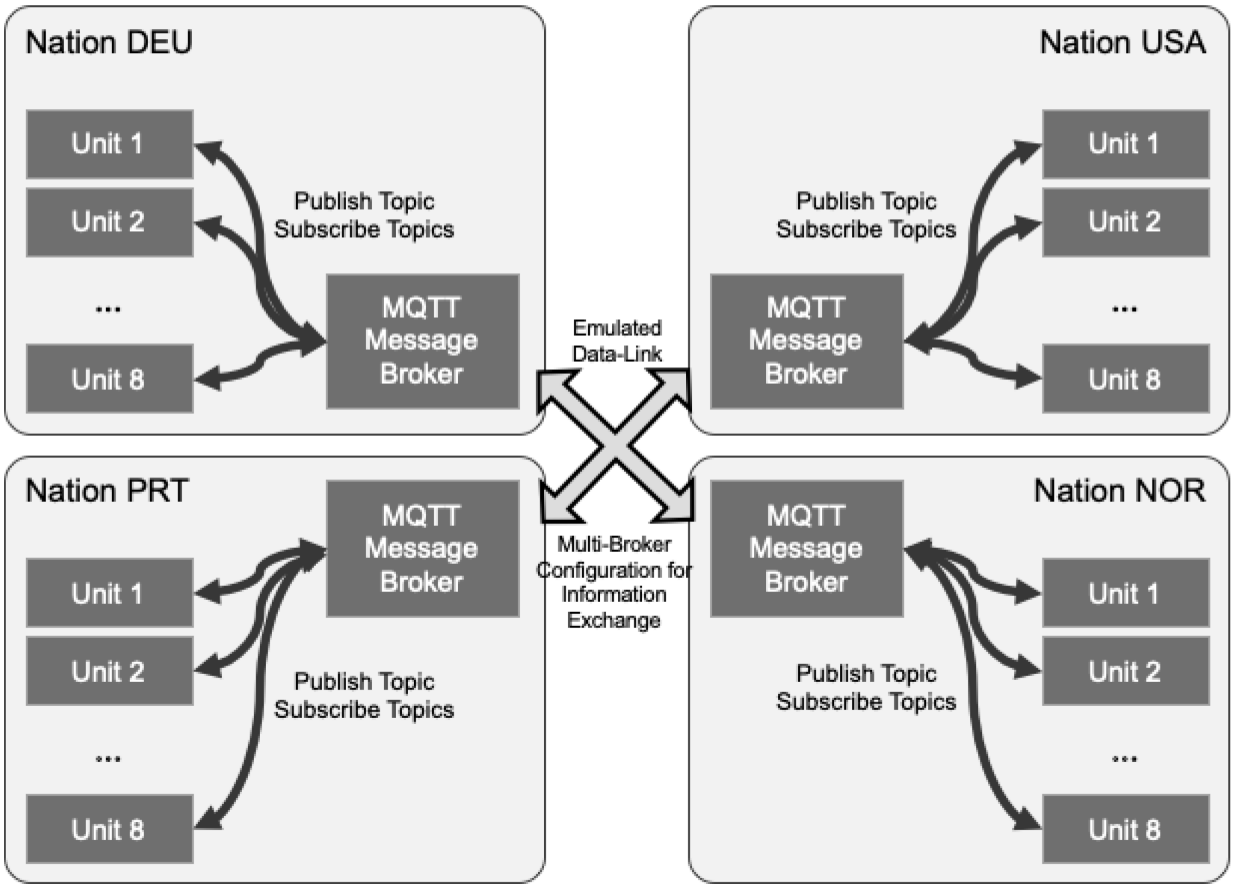


Figure 11: Multinational Setting 2: Four Nations.

The variations used in the experiments are presented in Table 3.

Table 3: Experiment Variations for Multinational Setting 2.

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

As shown in Table 3, a total of 18 experiment runs were conducted.

3.3.2.1 Publish/Subscribe Software

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** [35]. In this setup, each nation deploys an instance of the Mosquitto broker. However, it is noted that 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 [31].

- **Multi-broker configuration using VerneMQ** [40] **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 builds 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.
- **Brokerless configuration** [41] 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. Since it is based on UDP, we expect this approach 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 [42], 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 common and representative communication channel used in the field. Also, it is a much narrower link than the tactical broadband that we have investigated earlier in Ref. [43]. The following network configurations were used to emulate datalinks between nations:

- **Baseline 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 (>100 Mbps) and minimal latency (order of a few ms).
- **Tactical setup 1**: in this setting, the network throughput is limited to 9.8 kbps, with 100 ms latency and 1% packet loss.
- **Tactical setup 2**: in this setting, the network throughput is limited to 9.8 kbps, with 100 ms latency and 10% packet loss.

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

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

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

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

For the analysis of the experiments, we used analysis tools from the Analyse and Test environment (AuT) as described in Section 2.3.2.
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 analysing these logs was extended to automatically generate the following statistical information: number of sent messages per group, number of received messages per group, and delay of message transfers with median, minimum, maximum and quartiles. Based on these statistics, boxplot diagrams can be generated allowing to visualize and compare the performance of the MQTT implementations in the different test setups.

To evaluate the transmission reliability of the different MQTT implementations, we measured how many messages got lost in different setups. It should be noted that we used QoS 0 in these experiments. Our previous works have compared different QoS settings and their reliability and overhead [44].

3.3.3 Experiments Results and Evaluation

The results and evaluation of the experiments are presented in detail in Ref. [37]. Here, we present their highlights.

3.3.3.1 Transmission Reliability Results

The transmission reliability is determined based on the percentage of messages lost: the lower the percentage, the higher the reliability.

The results for the multinational setting 2 (four brokers and 32 nodes) are presented in Table 4 and Figure 12.

Table 4: Multinational Setting 2: Transmission Reliability.

Multinational Setting 2 (4 Brokers)	% Messages Lost	
	10 Seconds	2 Seconds
Mosquitto (Baseline)	0.00%	0.00%
VerneMQ (Baseline)	0.00%	0.00%
MQTT UDP (Baseline)	0.00%	0.00%
Mosquitto (tactical setup 1)	27.14%	64.28%
VerneMQ (tactical setup 1)	31.42%	67.36%
MQTT UDP (tactical setup 1)	0.66%	0.70%
Mosquitto (tactical setup 2)	26.40%	62.87%
VerneMQ (tactical setup 2)	33.49%	61.39%
MQTT UDP (tactical setup 2)	7.89%	7.53%

Note: The following colour code is used: values above 20% are orange, values above 40% are red.

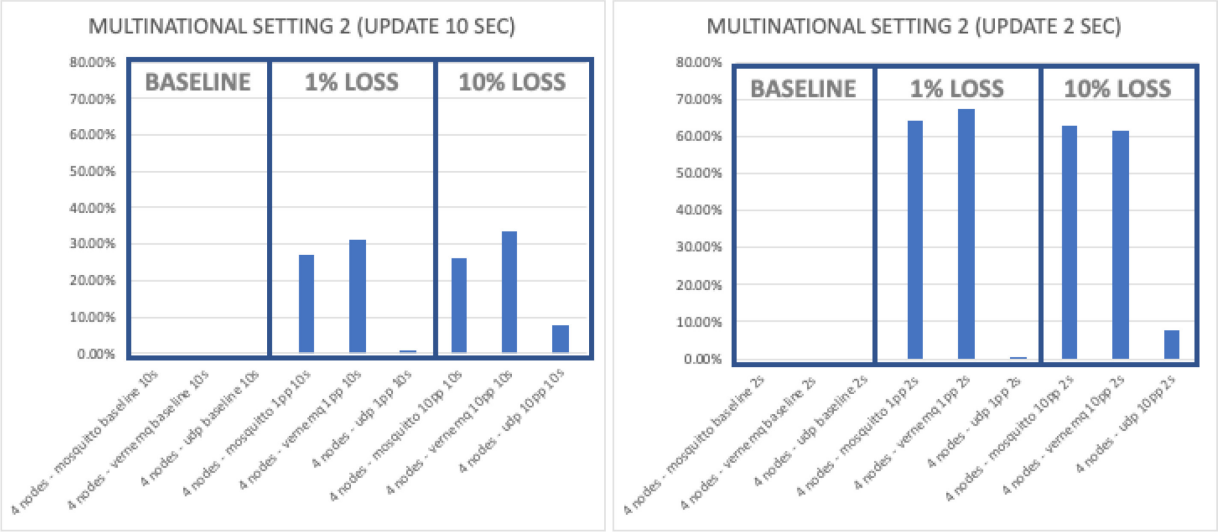


Figure 12: Multinational Setting 2: Transmission Reliability Based on % Lost Messages for BFT Updated Each 10 Seconds (Left) and 2 Seconds (Right).

As expected, there were no lost messages for the baseline setup.

For the case of DIL networks, we observe a reduction in reliability, especially for Mosquitto and VerneMQ. Interestingly, in this setting, Mosquitto and VerneMQ yield similar results.

Setting a period of 10 seconds for BFT messages in tactical setup 1, Mosquitto and VerneMQ exhibit a message loss percentages of 27% and 31% respectively.

When setting a period of 2 seconds for BFT messages, we observe a considerable reduction in reliability, with % messages lost higher than 60% for Mosquitto and VerneMQ.

MQTT UDP delivered surprisingly good results in this setting, which were independent of the BFT period: for tactical setup 1 there were almost no lost messages (below 0.7%) while for tactical setup 2 we recorded almost 8% lost messages. We emphasize the following:

- The UDP low traffic overheads seems offer to an advantage in effectively exchanging data in tactical networks, when compared with TCP. The used BFT periods did not cause network congestion in UDP.
- Lacking delivery assurance, MQTT UDP was especially affected by the network loss parameter (in tactical setting 2).

In the multinational setting 2, MQTT UDP greatly outperformed Mosquitto and VerneMQ, in our view due to the advantage of UDP over TCP in DIL networks. Mosquitto and VerneMQ reliability levels were similar.

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 low. Hence, it could be beneficial to favour UDP under such conditions.

3.3.3.2 MQTT Performance Results

To compare the message transmission delay of the different MQTT implementations in different setups, boxplots were generated for each test run, one for all exchanged messages and one for all messages exchanged between brokers (Broker-Broker, *i.e.*, crossing an emulated (possibly DIL) network link). We use as metric "message delay", measured as the time difference between when a message is published until a message is received (in seconds), with median, minimum, maximum and quartiles values.

The results for the multinational setting 2 (four brokers and 32 nodes) are presented in Table 5 and Figure 13.

Table 5: Multinational Setting 2: Message Delay.

Multinational Setting 2 (4 Brokers)	**Message Delay (in seconds) Broker-Broker Exchange**				
BFT period 10 seconds	**min**	**lower quartile**	**median**	**upper quartile**	**max**
Mosquitto (Baseline)	0.00	0.00	0.01	0.98	3.07
VerneMQ (Baseline)	0.00	0.00	0.01	0.02	0.23
MQTT UDP (Baseline)	0.00	0.00	0.00	0.01	0.11
Mosquitto (tactical setup 1)	0.00	10.88	55.15	99.35	407.88
VerneMQ (tactical setup 1)	0.48	4.07	9.11	158.95	1159.51
MQTT UDP (tactical setup 1)	0.36	0.36	0.36	0.36	0.41
Mosquitto (tactical setup 2)	0.00	16.50	48.75	102.35	278.52
VerneMQ (tactical setup 2)	0.42	14.03	389.21	1037.26	1525.38
MQTT UDP (tactical setup 2)	0.36	0.36	0.36	0.36	0.48
BFT period 2 seconds	**min**	**lower quartile**	**median**	**upper quartile**	**max**
Mosquitto (Baseline)	0.00	0.00	0.01	0.15	1.05
VerneMQ (Baseline)	0.00	0.01	0.02	0.04	1.27
MQTT UDP (Baseline)	0.00	0.00	0.00	0.00	0.15
Mosquitto (tactical setup 1)	0.38	26.23	106.73	190.33	436.56
VerneMQ (tactical setup 1)	0.87	43.43	98.00	222.10	1258.94
MQTT UDP (tactical setup 1)	0.36	10.07	20.27	30.50	48.14
Mosquitto (tactical setup 2)	0.38	67.26	136.95	206.06	649.63
VerneMQ (tactical setup 2)	0.65	70.96	137.10	327.74	1446.22
MQTT UDP (tactical setup 2)	0.00	0.00	0.01	0.15	1.05

Note: The following colour code is used: values above 5 are orange, values above 20 are red.

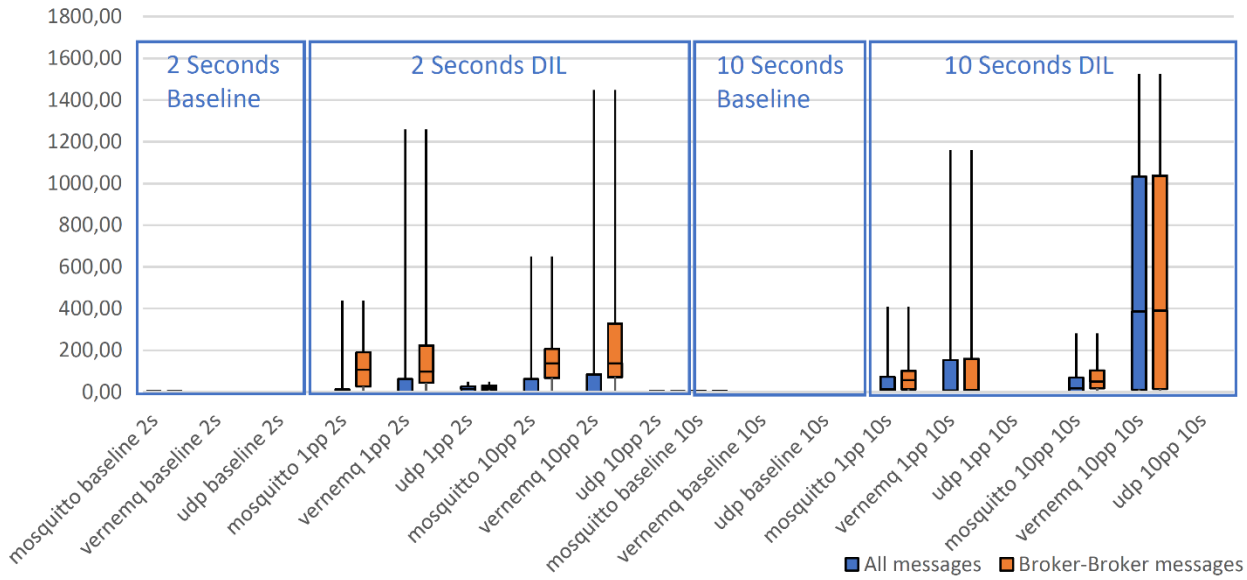


Figure 13: Transmission Delay, Four Servers.

The baseline setup performed well as expected (barely any delays were observed).

For the case of DIL networks, setting the BFT update period to 10 seconds in tactical setup 1 caused an increase in message delay in Mosquitto (median value of 55s.15 seconds) and VerneMQ (median value of 9.11 seconds). Herein, VerneMQ greatly outperforms Mosquitto in terms of median value, however registering higher delays in the upper quartile (158.95 vs. 99.35) and maximum recorded value (1159.51 vs. 407.88). For tactical setup 2, we see, unexpectedly, a slight performance increase in Mosquitto (median value of 48.75 seconds), but a dramatic decrease in VerneMQ (median value of 389.21 seconds).

When setting the BFT update period to 2 seconds, the performance is further degraded: in tactical setup 1, Mosquitto and VerneMQ message delay median value was 106.73 and 98.00 seconds, respectively. Performance was further impacted when changing from tactical setup 1 to tactical setup 2: Mosquitto and VerneMQ message delay median value increased to 136.95 (28% increase) and 137.10 (40% increase) seconds, respectively.

Concerning the results observed with MQTT UDP, we confirm the advantage of using UDP over TCP in tactical networks. MQTT UDP outperformed all other used implementations, exhibiting very small delays in message delivery, except in tactical setup 2 when using a BFT update period of 2 seconds. First, UDP functions as a “fire and forget” mechanism without delivery assurance (absence of acknowledgement and retransmissions) which makes is very efficient at the cost of reliability. For the case of tactical setup 2 at BTF period of 2 seconds, where we recoded median delays of 20.27 seconds, we speculate it is a result of data packet queuing in the IP stack resulting from high traffic and low network throughput.

To further analyse why the transmission significantly delays the rise when the sending rate and packet loss are increased, we used another visualization graph provided by AuT Analyser Component. As an example, Figure 14 shows the transmission times of all messages separately (cf. red dots) for the Mosquitto multinational setting (4 brokers), tactical setup 1 (DIL with 10% loss rate) and BFT period of 2 seconds. 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.

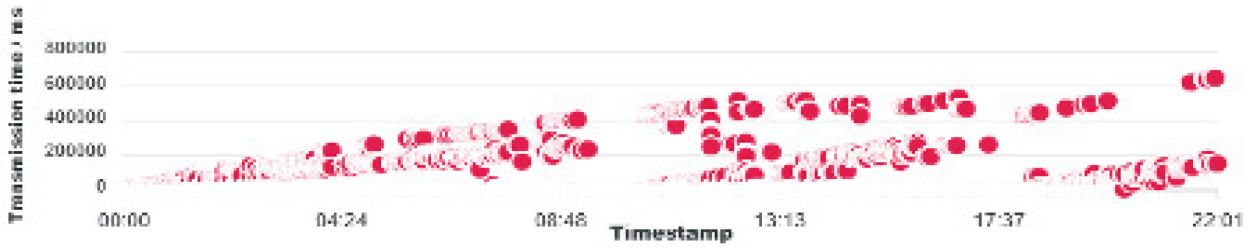


Figure 14: Transmission Times of All Messages.

3.3.4 Conclusion from Experiments

The investigated multi-coalition scenario bridging four nations' networks for information exchange using a message broker mechanism, compatible with the principles of a federated architecture, yielded interesting findings concerning the performance of the tested different MQTT implementations, namely:

- Mosquitto (with the MQTT bridge mechanism) [35];
- VerneMQ (with its bespoke mesh approach to build an MQTT cluster) [40]; and
- MQTT UDP, a non-standard UDP based brokerless implementation of MQTT [41].

From our experiments, we found that the use of MQTT with UDP seems to be superior in DIL networks (i.e., low bandwidth networks with high packet losses) when compared to the standard TCP based flavours of MQTT. Only one run (out of 6, see Table 5) generated noticeable performance degradation in message delivery, which we presume was caused as a result of traffic congestion (attempt to send more data than the available network capacity) and IP queuing.

Furthermore, we tested VerneMQ with its clustering mechanism in order to achieve a fully decentralized deployment (compliant with the principles of a federation of systems) and overcome the single-point-of-failure issue present in most MQTT platforms (like Mosquitto). We observed that, for most runs, 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. While the bridge in Mosquitto implements a selected forwarding of configured topics, the clustering mechanism in VerneMQ provides full redundancy on both data and subscriptions in the cluster. The trade-off for this additional functionality is, naturally, more resource use than the simpler mechanism in Mosquitto.

When considering its application in tactical (DIL) networks, a drawback in today's MQTT standard is that it is indeed TCP based. Our experiments show that UDP is a better match for this kind of message distribution mechanism in tactical networks. So, ideally the MQTT specifications should evolve to support UDP. 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, as described by Suri et al. [7].

3.4 Conclusion

In this section, we presented our analysis and findings of technologies for Message-Oriented Middleware (MOM) Service, supporting the publish/subscribe communication paradigm for timely exchange of data in tactical environments (i.e., DIL networks). Supported by previous studies and conducted research, we presented candidate technologies for information exchange based on publish/subscribe, including the NATO recommended WS-Notification and the emerging MQTT protocol. We described experiments

conducted as part of NATO IST-150 group in order to better understand the application of the technologies in tactical networks in a coalition environment. Our main findings include:

- WS-Notification is based on XML and SOAP, it makes it a more resource demanding protocol than MQTT, which is built directly on TCP. As such, WS-Notification consumes more networking resources than MQTT.
- MQTT exhibits a "lighter" and more efficient network performance than WS-N, which makes it suitable for mobile tactical environments, where network resources are scarce.
- The OLSR routing protocol generates a large amount of data volume, thus protocol improvements (e.g., different update rates) should be investigated or, otherwise, alternative routing protocols better suited for tactical mobile environments using wideband (or narrowband) radios should be deployed.
- TCP has been designed to assure delivery of all messages. However, in tactical networks (low bandwidth and intermittent), it produces many "spurious" TCP retransmits which leads paradoxically to significantly more lost messages than with unreliable UDP. This indicates that TCP is not well suited for tactical network environments. Our experiments show that UDP is a better match for this kind of message distribution mechanism in tactical networks.
- The use of MQTT with UDP was superior in DIL networks when compared to the standard TCP-based flavours of MQTT. Evolving the MQTT specifications to support UDP is therefore highly recommended. 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, as described by Suri et al. [7].
- A clustering broker mechanism (as supported by VerneMQ) complies with the principles of a federation of systems and overcome the single-point-of-failure issue present in most MQTT platforms. The trade-off for this additional functionality is, as expected, more resource use than, for example, the simpler mechanism used in Mosquitto (always relying on a "main" broker).

4.0 REQUEST RESPONSE

Besides the publish/subscribe type of Message-Oriented Middleware (MOM) Services discussed in Section 3.0, IST-150 also investigated middleware services for request/response. Request/response is a messaging pattern in which one entity seeking information, the client, sends a request message to the information source, and gets a response back. It is also possible to use this pattern to push information from one entity to another and get a delivery receipt back. Thus, this messaging pattern fits naturally to the direct distribution of military messages or commands from one sender to a receiver. If the used transport protocol supports multicast, this pattern can also be used to push information to a group of receivers.

We investigated using proxies, to add delay and disruption tolerance to the otherwise disruption prone request/response operation. Following the initial experiments with proxies, we investigated whether the request/response pattern can be implemented by a RESTful Web service in a way that it distributes information very efficiently and thus can be used in tactical networks. One main advantage of REST (e.g., implemented with HTTP and JSON) compared to other middleware approaches is that it is simple and widely used. Because of its simple structure compared to SOAP, it can be easily standardized, and the risk of incompatible implementations caused by ambiguous specifications is lower. Furthermore, REST has a much lower overhead compared to SOAP.

To evaluate our approach based on RESTful Web services to implement request/response in tactical networks, we developed a Military Messaging service with support of different transport protocols. First, in Section 4.1 a proxy experiment is described. The Military Messaging service is explained in Section 4.2. The testbed used for the evaluation is shown in Sections 2.3.2 and 4.3. The experiments are described in Section 4.4. This section concludes and gives recommendations for request/response in Section 4.5.3.

4.1 Proxy Experiment

Initial work performed around the time IST-118 ended and IST-150 was started, encompassed several industry standard transport protocols and leveraging a bespoke proxy implementation supporting these standards to overcome certain aspects of the DIL challenges of tactical networks. The work was motivated by challenges identified in IST-090 and 118 related to using standard Web services across DIL networks, where we found that military communication may have DIL limitations, so that Web services built for civilian usage may not handle these limitations. To investigate this problem space further, we developed a proxy with the following requirements:

1) Support HTTP RESTful and W3C SOAP Web services.
2) Work in DIL networks.
3) Be interoperable with standards-based COTS solutions.
4) Work with security mechanisms.

Here, the idea was that the proxy should ideally be payload agnostic, up to the point of supporting any request/response standard service mechanism (point 1, both SOAP and REST support, as well as point 4, not modifying the payload in transit as to not break any security mechanism, e.g., digital signature). The proxy needed to add delay and disruption tolerance to such services. The common denominator of COTS SOAP and REST services is that they both use HTTP over TCP as the underlying protocol, so we implemented the proxy to support any HTTP-type service (point 2, cope with DIL, and point 3, interoperable with standards-based COTS).

The implementation was made as a proxy pair, so that the pair was intended to be deployed on the client and service side, respectively, with the DIL network as transit network in between, as illustrated in Figure 15:

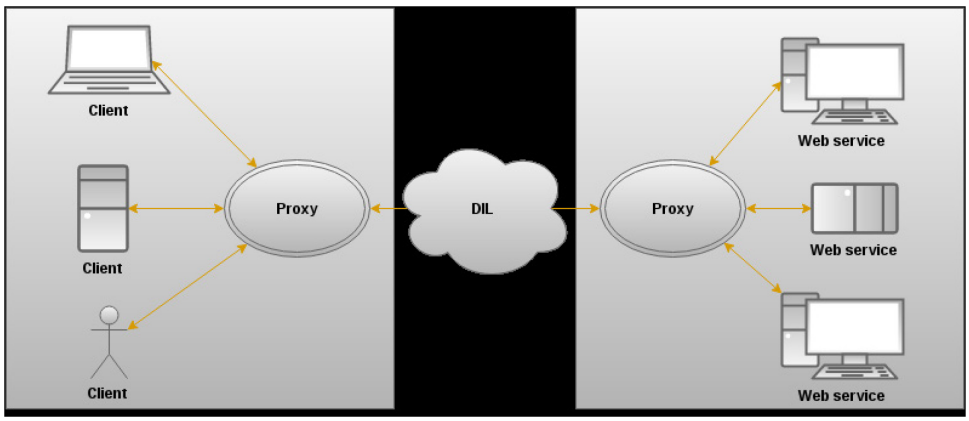


Figure 15: DIL Proxy Pair Approach.

Here, different clients may have a proxy on their side (located on the same local network, or perhaps even on the same physical client). The clients interact with the proxy using COTS Web services (be it either REST or SOAP), in that the client is explicitly configured to use the Proxy as its HTTP proxy. The proxy then handles delay and disruption tolerance across the DIL network, where the traffic reaches the proxy on the service side. There, the proxy connects to the service on behalf of the original client and mediates any response back to the clients subsequently. So, client/service side part of the proxy pair uses the standard Web service interfaces (HTTP requests and responses), whereas between the proxy pairs (inter-proxy communication) it is possible to configure using different protocols, illustrated in Figure 16.

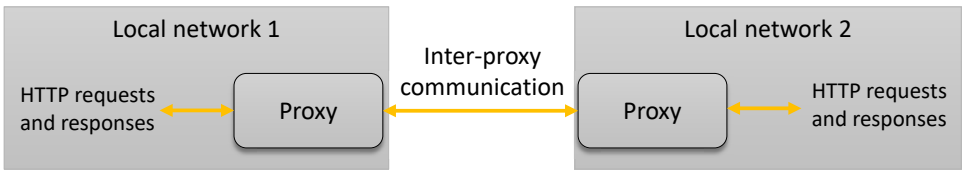


Figure 16: Proxy Pair Communications.

The inter-proxy communication could be configured to use different communication protocols, e.g., HTTP, CoAP and others. It was also possible to enable or disable non-lossy compression of the payload in the proxy, to further reduce overhead while in transit across the network.

We performed experiments with this approach using NetEm to emulate various types of networks; the same networks identified by IST-118 for experiments, shown in Table 6.

Table 6: IST-118 Representative Experiment Networks for NetEm Configurated Experiments.

DIL Networks			
Network	**Data Rate**	**Delay**	**Error Rate**
Satellite Communication	250 kbps	550 ms	0 %
Line of Sight	2 mbps	5 ms	0 %
WiFi 1	2 mbps	100 ms	1 %
WiFi 2	2 mbps	100 ms	20 %
Combat Net Radio	9.6 kbps	100 ms	1 %
EDGE	50 kbps up / 200 kbps down	200 ms	0 %

In addition to experimenting across emulated networks, we also performed an experiment using KDA WM600 tactical broadband radios (in a lab setting, back-to-back but with a dampening device in between to yield conditions representative of field usage). Our findings indicated, that for the most part, the delay and disruption tolerance in the inter-proxy communication worked fine with HTTP. Only for the most limited type networks with respect to throughput, e.g., Combat Net Radio (CNR) and the actual tactical broadband radios, did we find that switching to CoAP was preferable. The experiments are further described and detailed in Ref. [44]. The summary of our findings is presented in Table 7 (recommendations), as well as graphs of results from the WM600 tests (Figure 17).

Table 7: Resulting Recommendations (Networks Correspond to above IST-118 Networks Figure Tests).

Network	NFFI Test case	REST Test Case
SATCOM	HTTP proxy	HTTP proxy
LOS	HTTP proxy	HTTP proxy
WiFi 1	HTTP proxy	HTTP proxy
WiFi 2	HTTP proxy	HTTP proxy
CNR	CoAP proxy	CoAP proxy
EDGE	HTTP proxy	HTTP proxy

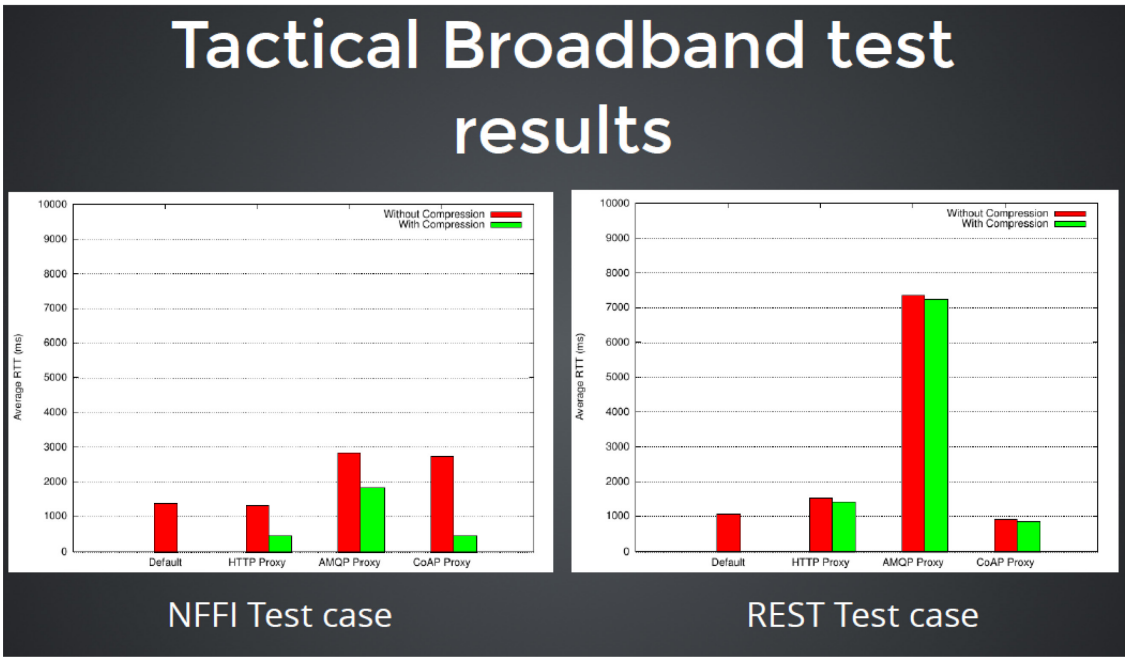


Figure 17: Tests with WM600 Using SOAP (Lefthand Side, NFFI Service) and REST (Righthand Side).

Here, we can see that enabling compression (green bars) has a positive effect. Also, we can see that using CoAP overall yields the better performance. It should be noted that the NFFI tests have a large payload compared to the REST service. So, while the NFFI test has periodic large payload transfer (blue force tracking), the REST service has multiple small interactions, so more frequent information exchange. An interesting thing to note here, is that AMQP suffers in tactical networks with a lot of small packets, whereas CoAP overall copes well (with compression on, notably) for both the REST and the SOAP service. Note that for NFFI, we have the SOAP going end to end, and for REST we have JSON going end to end.

Due to these results pointing to CoAP potentially being a very capable protocol in tactical networks in the face of DIL characteristics, IST-150 has performed further in-depth analysis of this protocol, as well as additional data formats, as described in the remainder of this section.

4.2 Restful Military Messaging Service

To compare the performance of different transport protocols and data formats/compression methods, we developed a RESTful Military Messaging service which is used to distribute military messages (e.g., commands) from one sender to a receiver or a group of receivers.

While REST usually is deployed in a client-server setting, we aimed for a decentralized solution which is better suited for tactical networks. For this purpose, we designed the Military Messaging service to be deployed as a server instance on each network node. Thus, each node can push messages via REST to each other node and optionally can get a receipt acknowledgment back. This also enables the sending entity to send a message via multicast to a group of receivers if the transport protocols support multicast.

The Military Messaging service supports sending of messages according to the data/compression formats JSON, CBOR [45] and EXI [46] such as the protocols HTTP and CoAP [47]. CoAP can be used with the transport protocols TCP or UDP. In case of UDP, multicast can be used optionally.

Remark (transport protocols): HTTP has to be used in conjunction with TCP and thus is connection-oriented. CoAP can be used with different transport protocols, amongst others with connectionless UDP and TCP. CoAP assures that messages are delivered reliably even when a connectionless transport protocol is used. This can be configured with help of the corresponding QoS setting. “Best Effort” delivery is also supported, but is not used for Military Messaging, since messages shall be delivered reliably in this case.

4.2.1 Data Model for Military Messages

Since no international standardized data formats for military messages were available, we based our specification on the data format “Operational Message” which was specified in CoNSIS project [48]. Based on the part defining a free-text message, we defined a data model which is shown in Figure 18. The data model is designed in a way that it can be extended to include other types than free-text messages as needed.

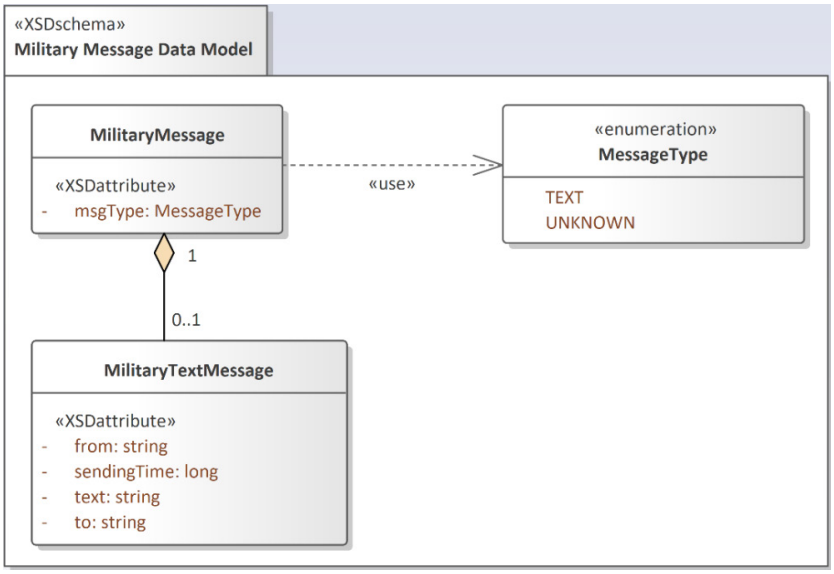


Figure 18: Data Model for Military Message.

This data model can be instantiated with different data formats. As an example, Figure 19 shows a representation based on JSON.

```
{
"msgType" : "TEXT",
"from" : "Carla",
"to" : "Eugen",
"text" : "Hi Eugen!",
"sendingTime" : 1553618941218
}
```

Figure 19: Military Message in JSON Representation (Example).

4.2.2 Configuration of Military Messaging Service

We specified a configuration data model which can be used to configure which transport protocol and data formats/compression method is used by the Military Messaging service. This configuration data model is shown in Figure 20. As data formats JSON, CBOR or EXI are supported. Supported transport protocols are HTTP and CoAP. CoAP can be used with TCP, UDP or Multicast/UDP. The configuration data model was used to integrate Military Messaging service in the AuT testbed. This allows the operator of the testbed to specify the configuration of the service in an AuT scenario.

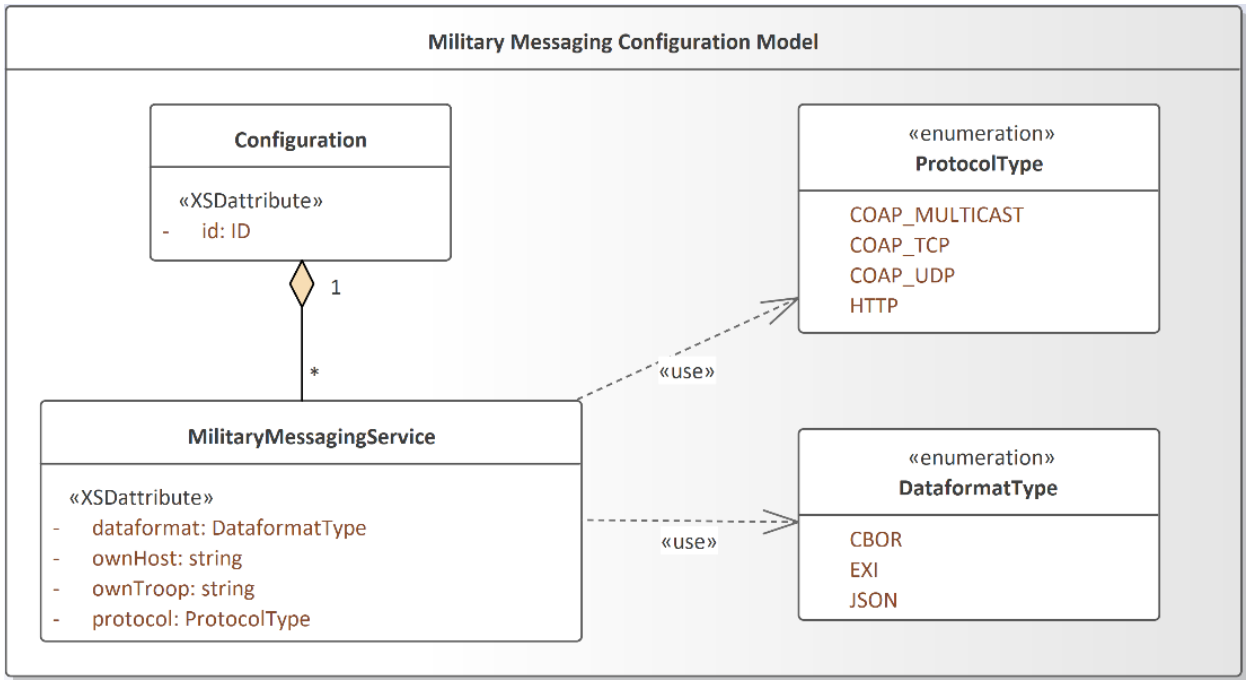


Figure 20: Configuration Data Model of Military Messaging Service.

4.2.3 Implementation

We implemented the Military Messaging service in Java. The HTTP part is based on “Eclipse Jersey” [49] – a framework for RESTful Web Services. The CoAP part is based on “Eclipse Californium” [50] – a framework for CoAP based services. Based on Californium three transmission variants were implemented: Unicast/TCP, Unicast/UDP und Multicast/UDP. As a basis for the serialization/deserialization of messages with JSON, CBOR and EXI as data formats, the library “Jackson” [51] was used. So far, we did not run experiments with multicast, but the implementation already supports this.

To increase the portability of the service, we deployed it in a docker container. This alleviates the integration into the AuT testbed. We implemented a plugin for the AuT scenario editor, which allows the administrator to configure which protocols and data formats are used in a scenario.

4.3 Use of AuT Testbed for Experiments

As a basis for the testbed for the experiments with Military Messaging service, we used the Analyse and Test environment (AuT) (cf. Refs. [16], [17]), which is outlined in Section 2.3.2. For the experiments, the AuT framework had to be extended. An AuT scenario including the IT system instances, networks and the application traffic had to be specified (see next section). Furthermore, Military Messaging service, a tactical router and the network emulator (see Sections 4.3.2 and 4.3.3) have been integrated into the testbed.

4.3.1 Scenario

We used a subset of Vignette 2 of the Anglova scenario [8] with two platoons of the first company. We assigned each platoon five simulated vehicles and the corresponding systems (Military Messaging service, tactical router and radio). Thus, the scenario contains ten units which is in our opinion sufficient for the planned analysis of the performance of transport protocols and data formats. The systems and network connections are shown below (see network plan).

We integrated the specification of the unit movements which is contained in an EMANE file of the Anglova scenario into the tactical simulator (TacSim) of AuT. TacSim replays the tracks and provides corresponding position events to all systems connected via AuT.

4.3.1.1 Network plan for Military Messaging Experiments

The network plan of the AuT scenario is shown in Figure 21. Each of both groups contains five units, each equipped with a tactical router (see “MOTOR” in Figure 21) and an application host for Military Messaging service. Units of each group are connected via a tactical broadband network (see “EMANE-1” and “EMANE-2” in Figure 21) with a bandwidth of 1 MHz and a variable data rate of 1 Mbit/s, 500 kbit/s or 380 kBit/s. Both group leaders are additionally connected with each other by a narrowband network (see “EMANE-0” in Figure 21) with 15 kbit/s data rate and 25 kHz bandwidth. Each of these three networks is realized by a separate instance of the network emulator (see Section 2.2 and 4.3.3).

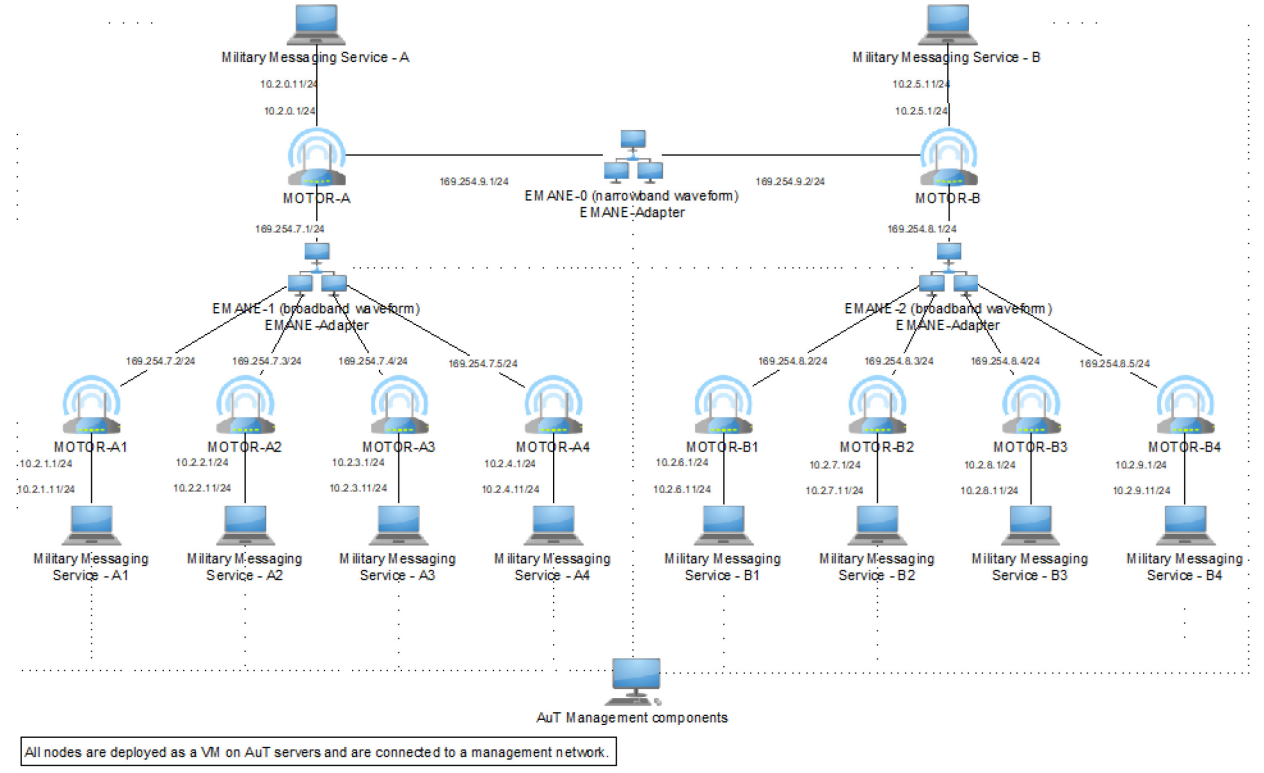


Figure 21: Network Plan.

4.3.1.2 Application Traffic

Since we aimed for a realistic military process in the scenario, we defined a script which specifies exactly when a message is sent by a unit and what is the receiver and content of this message. The script contains 97 messages according to a military scenario. The AuT management component reads the script and sends corresponding "SEND" action events to the systems. Each system sends a message when it receives a "SEND" action event from AuT.

4.3.2 Tactical Router

To provide a realistic network environment for the system-under-test (Military Messaging service), we needed a tactical router which could be connected to the hybrid network emulated by the network emulator(s).

MOdular Tactical rOuteR (MOTOR) is a tactical router developed at Fraunhofer FKIE which is based on OLSRv2 routing protocol [52].

MOTOR provides:

- Proactive routing with MPR support (for an efficient distribution of topology information);
- Multi-Topology routing;
- Support of different network technologies in the same routing domain:
  - Different update rates for different technologies; and
- EMCON support.

We have integrated ten MOTOR instances into the testbed and defined the AuT scenario in a way that ten instances of the Military Messaging service are each connected to a different tactical router. The tactical routers are connected to the network emulator instances as shown in Figure 21 (network plan).

4.3.3 Integration of Network Emulator

We have integrated the network emulator (see Section 2.2) into the AuT testbed by the implementation of an adapter, which supports the configuration interfaces and the control interfaces of AuT. In this way, the configuration of the network emulator instances can be defined in an AuT scenario.

The configuration contains the following properties:

- The selected **waveform**: broadband or narrowband.
- The **bandwidth** of the waveform.
- The **frequency** of the waveform.
- The **sending power** of the waveform.
- The **initial data rates** for both broadband and narrowband waveform.
- For the broadband waveform:
  - **Three supported data rates** which can be switched by the radio dynamically by changing the modulation scheme.
  - **Three SINR threshold values** (signal to interference plus noise ratio) used to indicate when the radio switches between the different modulation schemes / data rates.
- The **ID of the military unit**, used to map units in the scenario to network nodes in the network emulation.
- **Network Emulation Modules ID (NEM-ID)**: the ID of the EMANE node.

During the initialization phase, the adapter gets this configuration for each radio, configures the radios and starts all necessary components (EMANE and additional scripts/processes).

Furthermore, the adapter dynamically creates TDMA schedules according to the number of nodes in the scenario and provides these schedules to the TDMA model of EMANE with help of a TDMA Schedule Event.

For the connection of the network emulator with the tactical router instances, VLAN interfaces are created on one of the interfaces of each emulator VM. The AuT testbed connects the tactical routers and the network emulators with virtual networks (with the same VLAN id as defined in the emulator(s)). This ensures that packets sent by a tactical router to the emulator(s) are forwarded to the correct virtual interface based on the VLAN id. Furthermore, the adapter configures EMANE in a way that each NEM (Network Emulation Module) is assigned to the corresponding VLAN interface. Thus, packets of a tactical router arrive at the corresponding NEM in EMANE.

4.4 Experiments

In this section the experiments with Military Messaging service in the AuT testbed are described.

4.4.1 Goal of Experiments

The experiments are used to evaluate whether RESTful services can be used in tactical networks to efficiently exchange messages according to the request/response communication pattern. With help of the experiments the efficiency of different data formats (JSON, XML, CBOR), compression methods (EXI) and transport protocols (HTTP/TCP, CoAP/TCP, CoAP/UDP) in a realistic network environment according

to a military scenario is to be assessed. For the analysis we measure the delivery times and loss rates of messages in the selected scenario.

4.4.2 Results

In the following experiments the RESTful Military Messaging service was used. The service supports different variants of transport protocol and data format. These variants were tested (Table 8):

Table 8: Tested Variants of Transport Protocol and Data Format.

Variant	Protocol	Data format
A	HTTP/TCP	JSON
B	HTTP/TCP	CBOR
C	HTTP/TCP	EXI
D	CoAP/UDP	JSON
E	CoAP/UDP	CBOR
F	CoAP/TCP	CBOR

The results are shown below.

4.4.2.1 HTTP/JSON

The overall view of the transmission times when using HTTP/JSON is shown as BloxBot diagram in Figure 22. Figure 23 shows a more detailed view of this diagram.

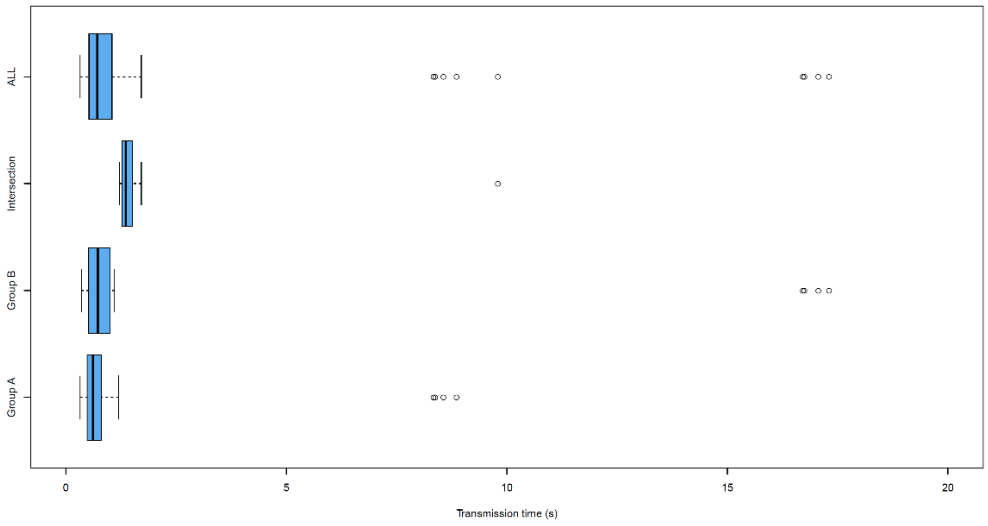


Figure 22: Results of the Experiments with Military Messaging Service (HTTP/JSON), Overall View.

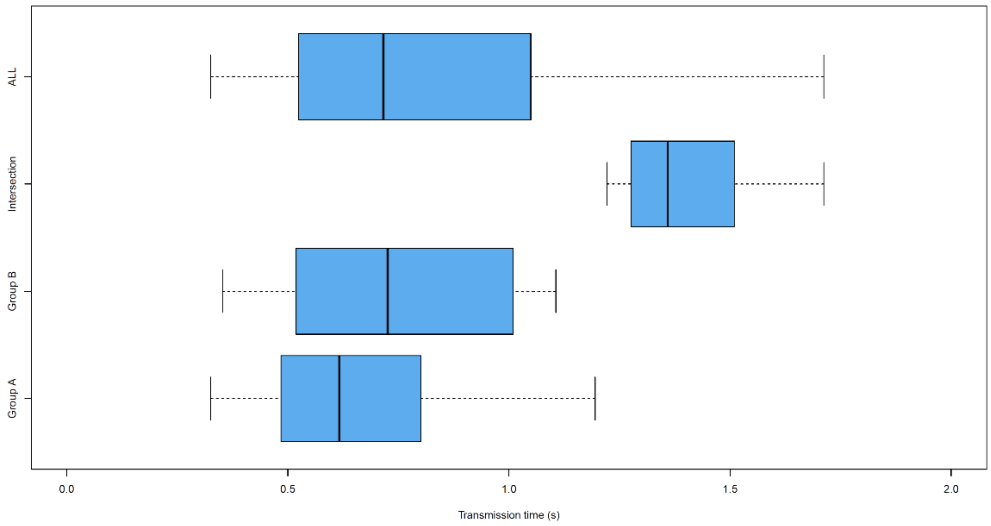


Figure 23: Results of the Experiments with Military Messaging Service (HTTP/JSON), Detail View.

The transmission times are split into three groups, because in the scenario two military groups were used. The members of one group (group A or group B) are connected via a broadband radio network inside of the group. The group leaders are connected via a narrowband radio link. Thus, the communication is divided into communication inside of group A, inside of group B and communication between group A and group B (called "intersection" in Figure 22). Furthermore, a boxplot showing all transmissions is shown in the graphs. The same scheme of visualization is used for the experiments with other protocols or data formats.

As one could expect, messages sent from one group to another (and thus crossing the narrowband link), have significantly higher transmission times.

The main results for HTTP/JSON are shown in Table 9; 6 (6.19%) of 97 sent messages were lost. The size of a message with small content was 418 Bytes.

Table 9: Main Results of Experiments with Military Messaging Service (HTTP/JSON).

Sent Messages	Lost Messages	Transmission Time (Min)	Transmission Time (Median)	Transmission Time (Max)	Transmission Time (Narrowband, Median)
97	6 (6.19%)	0.33 s	0.72 s	17.31 s	1.36 s

4.4.2.2 HTTP/CBOR

The results for HTTP/CBOR are shown in Figure 24 and Figure 25.

The main results for HTTP/CBOR are shown in Table 10. 7 messages (7.22%) of 97 sent messages were lost. The size of a message with small content was 369 Bytes.

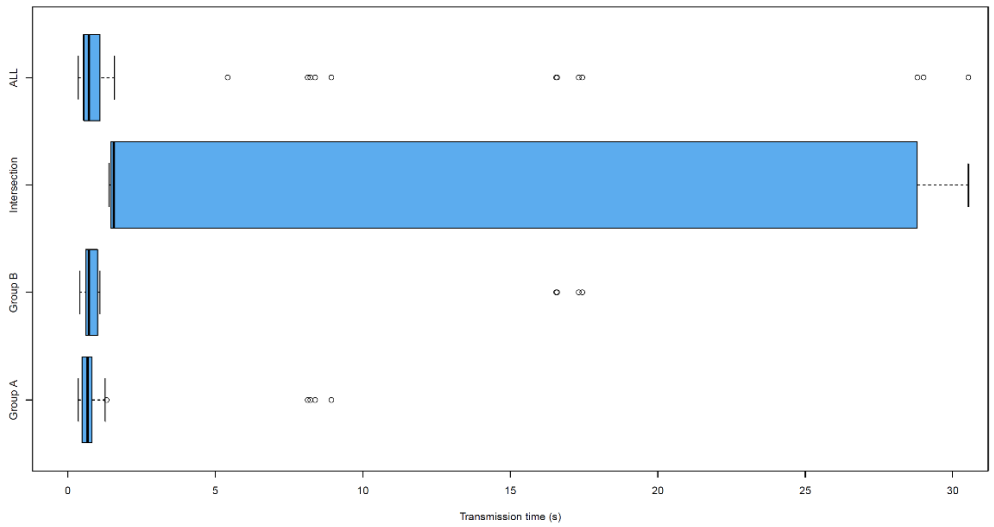


Figure 24: Results of the Experiments with Military Messaging Service (HTTP/CBOR), Overall View.

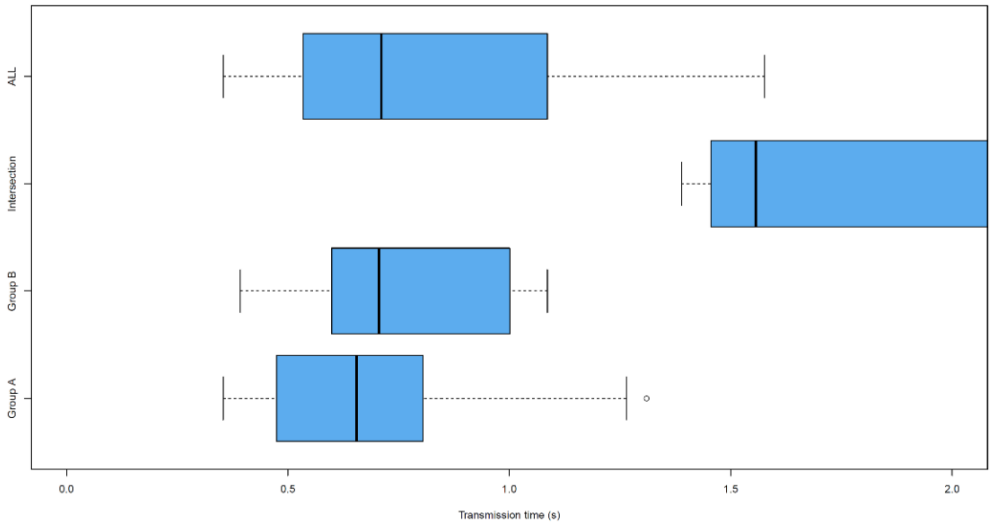


Figure 25: Results of the Experiments with Military Messaging Service (HTTP/CBOR), Detail View.

Table 10: Main Results of Experiments with Military Messaging Service (HTTP/CBOR).

Sent Messages	Lost Messages	Transmission Time (Min)	Transmission Time (Median)	Transmission Time (Max)	Transmission Time (Narrowband, Median)
97	7 (7.22%)	0.35 s	0.71 s	30.53 s	1.56 s

4.4.2.3 HTTP/EXI

The results for HTTP/EXI are shown in Figure 26 and Figure 27.

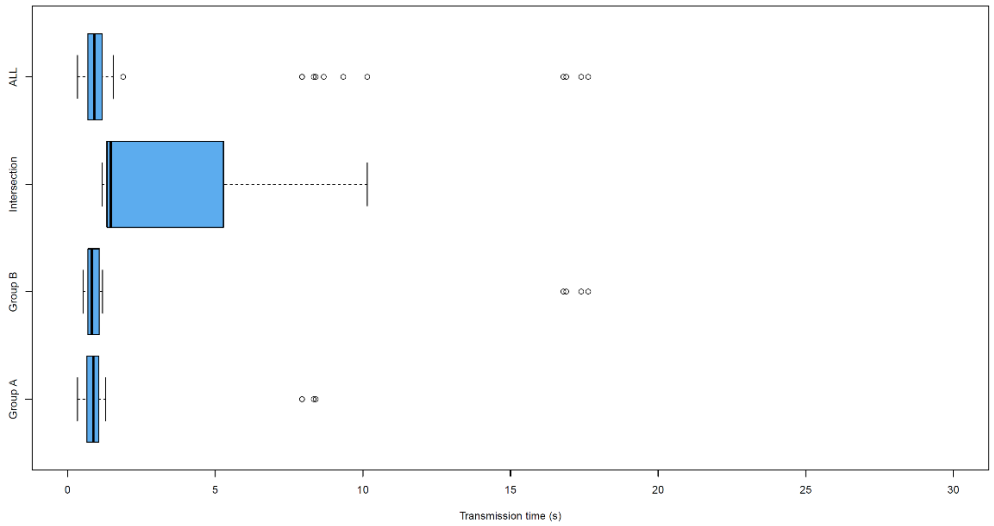


Figure 26: Results of the Experiments with Military Messaging Service (HTTP/EXI), Overall View.

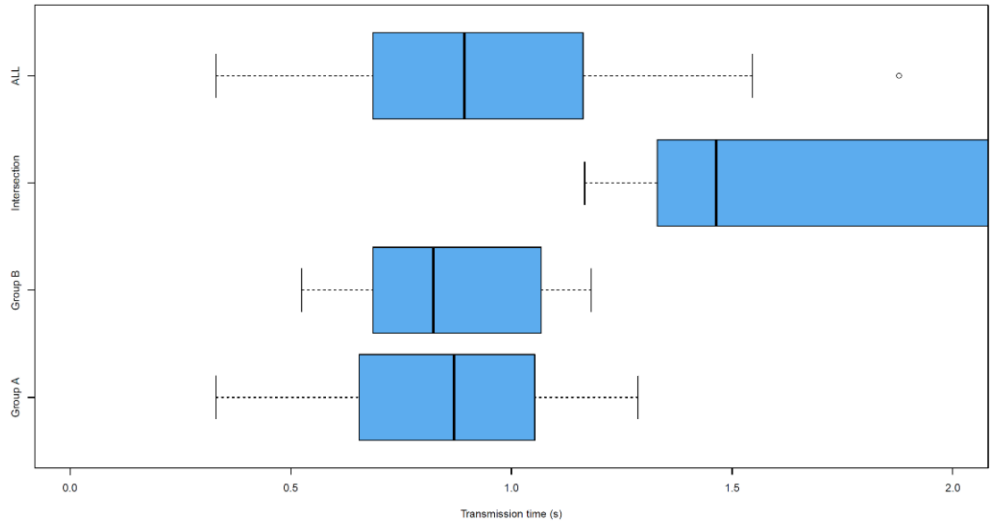


Figure 27: Results of the Experiments with Military Messaging Service (HTTP/EXI), Detail View.

The main results for HTTP/EXI are shown in Table 11. 6 (6.19%) of 97 sent messages were lost. The size of a message with small content was 377 Bytes.

Table 11: Main Results of Experiments with Military Messaging Service (HTTP/EXI).

Sent Messages	Lost Messages	Transmission Time (Min)	Transmission Time (Median)	Transmission Time (Max)	Transmission Time (Narrowband, Median)
97	6 (6.19%)	0.33 s	0.89 s	17.39 s	1.46 s

4.4.2.4 CoAP/UDP/JSON

The results for CoAP/UDP/JSON are shown in Figure 28 and Figure 29.

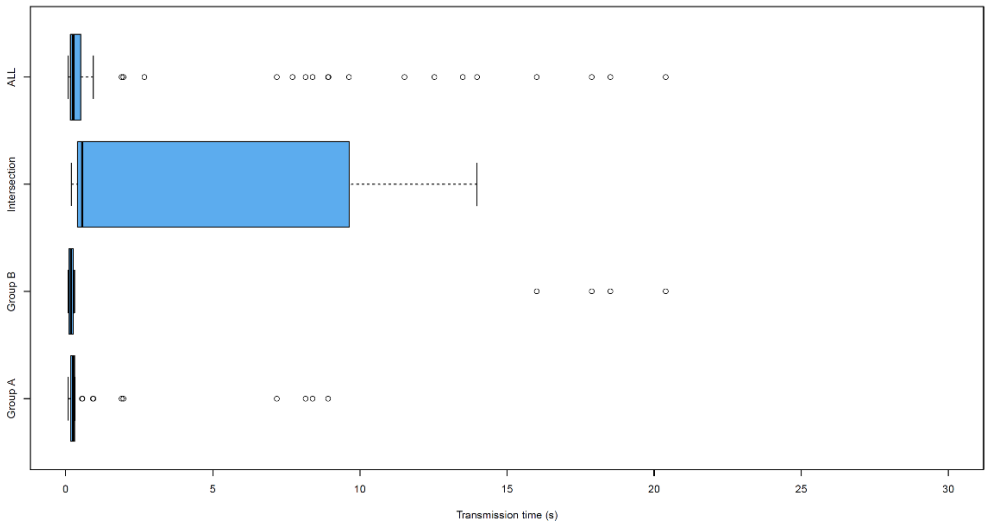


Figure 28: Results of the Experiments with Military Messaging Service (CoAP/UDP/JSON), Overall View.

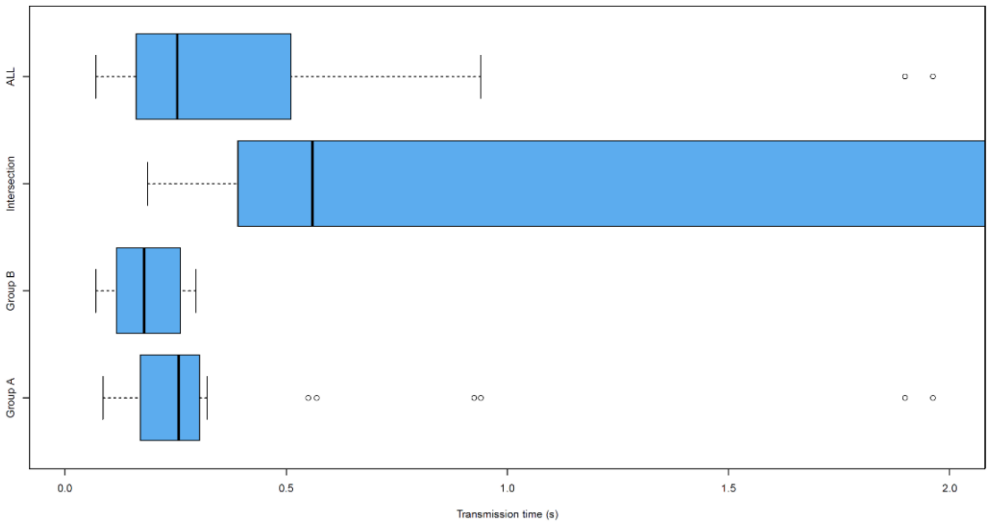


Figure 29: Results of the Experiments with Military Messaging Service (CoAP/UDP/JSON), Detail View.

The main results for CoAP/UDP/JSON are shown in Table 12. All messages arrived. The size of a message with small content was 153 Bytes.

Table 12: Main Results of Experiments with Military Messaging Service (CoAP/UDP/JSON).

Sent Messages	Lost Messages	Transmission Time (min)	Transmission Time (Median)	Transmission Time (max)	Transmission Time (Narrowband, Median)
97	0 (0%)	0.07 s	0.25 s	20.39 s	0.56 s

4.4.2.5 CoAP/UDP/CBOR

The results for HTTP/CBOR are shown in Figure 30 and Figure 31.

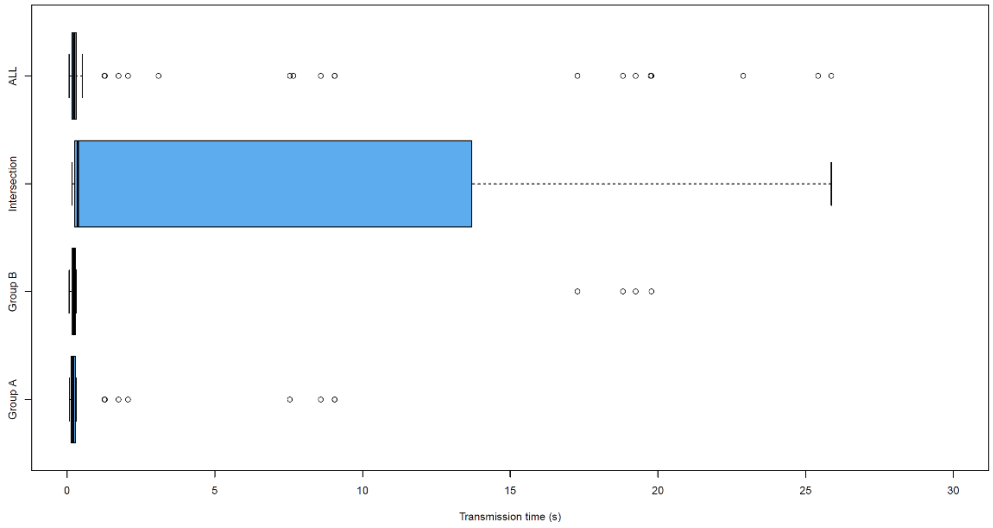


Figure 30: Results of the Experiments with Military Messaging Service (CoAP/UDP/CBOR), Overall View.

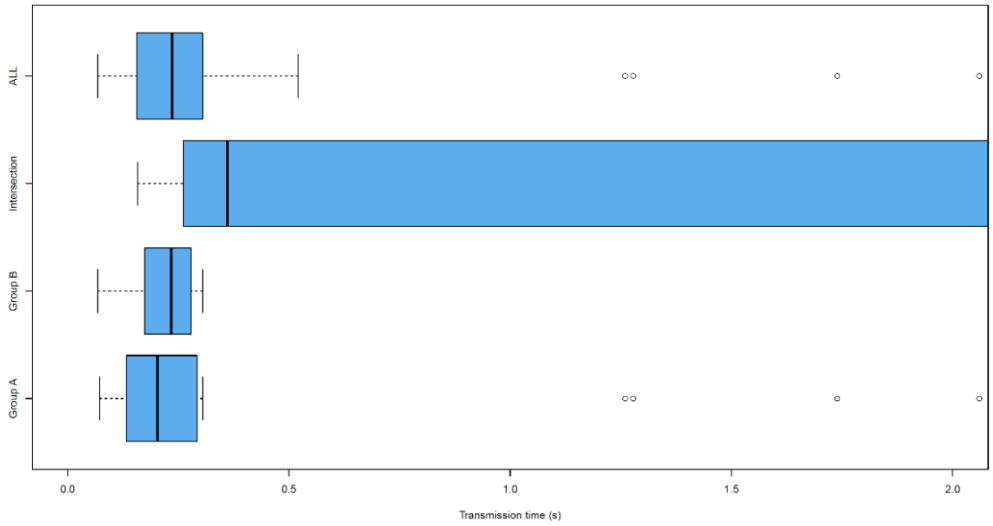


Figure 31: Results of the Experiments with Military Messaging Service (CoAP/UDP/CBOR), Detail View.

The main results for CoAP/UDP/CBOR are shown in Table 13. One message (1.04%) of 97 sent messages were lost. The size of a message with small content was 105 Bytes.

Table 13: Main Results of Experiments with Military Messaging Service (CoAP/UDP/CBOR).

Sent Messages	Lost Messages	Transmission Time (Min)	Transmission Time (Median)	Transmission Time (Max)	Transmission Time (Narrowband, Median)
97	1 (1.04%)	0.07 s	0.24 s	25.87 s	0.36 s

4.4.2.6 CoAP/TCP/CBOR

The results for HTTP/CBOR are shown in Figure 32 and Figure 33.

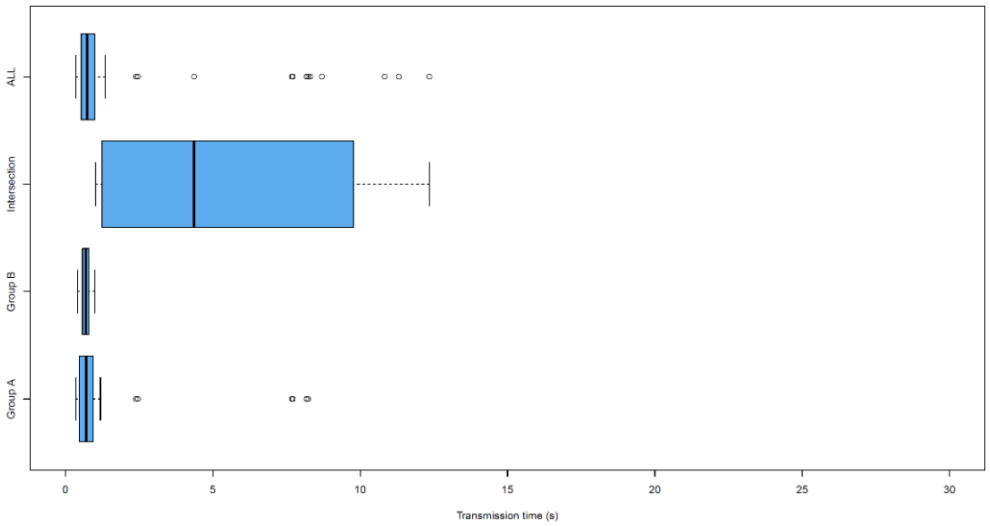


Figure 32: Results of the Experiments with Military Messaging Service (CoAP/TCP/CBOR), Overall View.

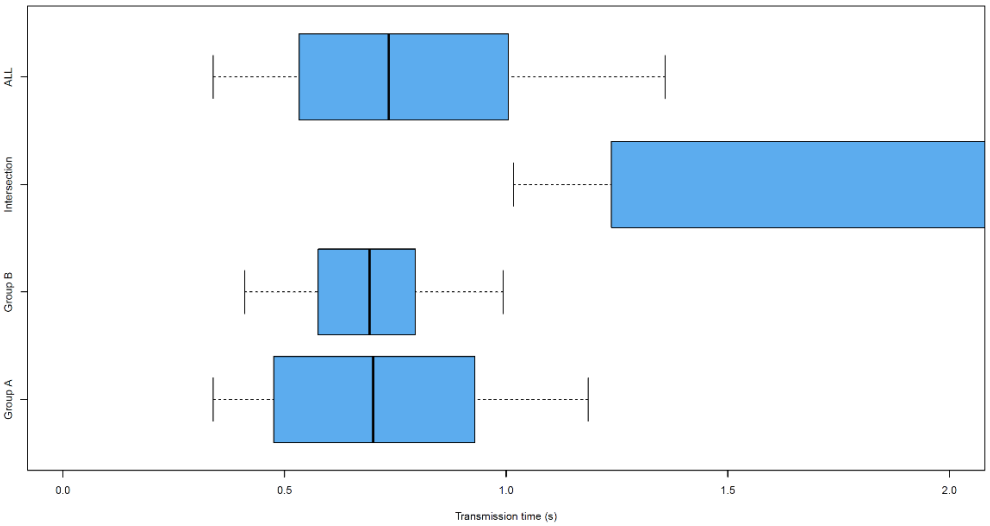


Figure 33: Results of the Experiments with Military Messaging Service (CoAP/TCP/CBOR), Detail View.

The main results for CoAP/TCP/CBOR are shown in Table 14. 10 messages (10.31%) of 97 sent messages were lost. The size of a message with small content was 104 Bytes.

Table 14: Main Results of Experiments with Military Messaging Service (CoAP/TCP/CBOR).

Sent Messages	Lost Messages	Transmission Time (Min)	Transmission Time (Median)	Transmission Time (Max)	Transmission Time (Narrowband, Median)
97	10 (10.31%)	0.34 s	0.74 s	12.35 s	4.36 s

4.5 Conclusions and Recommendations

4.5.1 Conclusions

In this section the results of Section 4.4 are summarized and compared to each other. Recommendations for request/response communication in tactical networks are described in Section 4.5.3.

The experiments with Military Messaging service were conducted with different transport protocols and data formats or compression methods, respectively.

Table 15 shows the measured message sizes of different protocols and data formats (including the headers of HTTP or CoAP). These sizes were obtained with messages with a very small textual content. As we can see in the table, when using HTTP, the benefit of a binary data format (CBOR) or compression (EXI) is small, because the overhead of header and TCP protocol is larger than the content of the messages. This benefit may be higher if larger text messages are sent. When using the CoAP protocol, the message size is reduced by 31% when CBOR or EXI is used. Whether this reduced message size results in a significantly improvement of the communication will be shown by the experiments.

Table 15: Message Sizes of Military Messages.

Test Case	Message Size
REST HTTP/JSON	418 Bytes
REST HTTP/CBOR	369 Bytes
REST HTTP/EXI	377 Bytes
REST CoAP/UDP/JSON	153 Bytes
REST CoAP/UDP/CBOR	105 Bytes
REST CoAP/TCP/CBOR	104 Bytes

An overview of the test results is shown in Table 16. First, results for all messages (without differentiation of the groups) are shown.

Of particular interest for a tactical middleware are messages which are transmitted between the groups A and B, because they have to cross the narrowband link connecting the group leaders. These are depicted below.

Remark: Since just 17 messages were sent over the narrowband link in the selected scenario, some variations can occur in these results. This means that e.g., a value of 41% loss rate does not have to be better than 35% in a repeated execution of the experiments. But since some of the results are by a magnitude larger than others and we observed always similar results when repeating the test runs, we can conclude from these results which variants are more successful than others.

Table 16: Comparison of Different Protocols and Data Formats/Compression (Overall Network).

Test Case	Sent Messages	Lost Messages	Transmission Time (Min)	Transmission Time (Median)	Transmission Time (Max)
REST HTTP/JSON	97	6.19%	0.33 s	0.72 s	17.31 s
REST HTTP/CBOR	97	7.22%	0.35 s	0.71 s	30.53 s
REST HTTP/EXI	97	6.19%	0.33 s	0.89 s	17.39 s
REST CoAP/UDP/JSON	97	0.0%	0.07 s	0.25 s	20.39 s
REST CoAP/UDP/CBOR	97	1.04%	0.07 s	0.24 s	25.87 s
REST CoAP/TCP/CBOR	97	10.31%	0.34 s	0.74 s	12.35 s

4.5.1.1 Impact of Protocols

As we can see in Table 16, the transmission times of all UDP based variants were significantly lower than the times of the TCP based variants if the communication was restricted to the broadband networks.

If the narrowband link was used (see Table 17), all TCP based protocols (HTTP and CoAP/TCP) perform badly w.r.t. loss rates (loss rate between 35.29 % and 58.82 %). In contrast, CoAP/UDP transmits almost all messages reliably and in a timely fashion, even if the narrowband network link is used.

CoAP with UDP is considerably more reliable than CoAP with TCP, because according to our experience from other experiments TCP in general is very unreliable if used in narrowband tactical networks.

The use of CoAP compared to HTTP led to considerably lower transmission times (e.g., the median for JSON was 0.25 s compared vs. 0.72 s for all messages and 0.56 s vs. 1.36 s when the narrowband link was used) and remarkable better reliability (0 – 6 % loss rate with CoAP vs. 35 – 41 % loss rate with HTTP when using the narrowband link).

Overall, CoAP/UDP provides a very reliable communication with low transmission times independent of the data format used.

Table 17: Comparison of Different Protocols and Data Formats/Compression (Narrowband Network).

Test Case	Send Messages	Lost Messages	Transmission Time Narrowband (Min)	Transmission Time Narrowband (Median)	Transmission Time Narrowband (Max)
REST HTTP/JSON	17	35.29%	1.22 s	1.36 s	17.31 s
REST HTTP/CBOR	17	41.18%	1.39 s	1.56 s	30.53 s
REST HTTP/EXI	17	35.29%	1.17 s	1.46 s	10.15 s
REST CoAP/UDP/JSON	17	0.0%	0.19 s	0.56 s	13.98 s
REST CoAP/UDP/CBOR	17	5.88%	0.16 s	0.36 s	25.87 s
REST CoAP/TCP/CBOR	17	58.82%	1.02 s	4.36 s	12.35 s

4.5.1.2 Impact of Binary Format and Compression

For HTTP there was no significant improvement of the transmission times by use of the binary CBOR format or the EXI compression compared to JSON. We conclude that the overhead of the protocol to ensure reliability in a tactical network weights more than the size of the messages content, since messages have to be sent repeatedly and acknowledged. Furthermore, REST based messages are already quite compact when they are JSON encoded (see Table 15). If the content of the messages is larger, the benefit from compression will be more relevant.

For CoAP there was a benefit by use of binary CBOR format or compression with EXI (e.g., 36% lower transmission time when using the tactical link). This is the case, because CoAP has a lower overhead than HTTP. Thus, a reduction of the content of the message has a higher impact on the overall (including headers) packet size.

4.5.2 Considerations About SOAP

The experiments were conducted with a RESTful Military Messaging service to evaluate the impact of different transport protocols and data formats or compression methods on the performance in tactical networks. We did not run the same experiments with SOAP/HTTP, the standard recommended so far by NATO, but did some experiments with SOAP/UDP.

In overall, we assume that the results for REST with HTTP can be transferred to SOAP with HTTP to a certain extent as well, since the experiments suggest that the transport protocol is more important for

the overall performance in tactical networks than the data formats. But as we have measured, the message size for military messages based on SOAP (Operational Message format) is much higher than for REST as can be seen in Table 15, Table 18, and Table 19. Compression does not have a noteworthy benefit (about 18% with BFT and EXI) if only the body of a SOAP message is compressed for interoperability reasons. In case of a small text message, the size of the GZIP compressed message even gets larger.

Table 18: Message Sizes for Military Messages with SOAP/UDP.

Test Case	Message Size
SOAP UDP/XML	1597 Bytes
SOAP UDP/XML/GZIP	1724 Bytes

Table 19: Message Sizes for Blue Force Tracking with SOAP/UDP.

Test Case	Message Size
SOAP UDP/XML	1838 Bytes
SOAP UDP/XML/GZIP	1792 Bytes
SOAP UDP/XML/EXI	1503 Bytes

First experiments with SOAP/UDP with the same testbed setup showed that the reliability is unsatisfying (loss rate about 40 – 60 %) if the narrowband link is used and that the transmission times are much higher than with REST/CoAP/UDP (about 1.8 s vs. 0.56 s with CoAP/UDP) in this case. The transmission times are low (similar to CoAP/UDP) inside of the groups (broadband network). This means that UDP helps to reduce transmission times compared to TCP but leads to similar high loss rates as TCP in narrowband networks. In narrowband networks, the large size of the messages caused by SOAP leads to high transmission times even though UDP is used.

We expect SOAP/HTTP to perform worse that SOAP/UDP w.r.t. to transmission times and reliability.

4.5.3 Recommendations

The experiments with a RESTful Military Messaging service showed that REST can be used in an efficient way in resource constrained hybrid tactical network. We have evaluated Military Messaging service in a military scenario with different transport protocols (HTTP and CoAP) and data formats/compression (JSON, CBOR, EXI). The results showed that the service performed best with CoAP/UDP w.r.t. to transmission times and reliability. CoAP/UDP had significant lower transmission times than HTTP/TCP. Furthermore, CoAP/UDP had much higher reliability in the narrowband network than TCP based protocols (loss rate about 0 – 6 % vs. 35 – 58 %).

First experiments with SOAP/UDP showed that UDP (without further mechanisms to ensure reliability) is not suitable for narrowband networks when services which require reliable transmissions (like Military Messaging) are used. The same holds for all TCP based protocols.

The use of compression did provide just a small benefit when the text messages were small. This benefit will be higher if larger messages are used.

Overall, implementing request/response with REST, CoAP/UDP and JSON (or formats like CBOR or EXI) seems to be a good choice in hybrid tactical networks and should be preferred to SOAP/HTTP or SOAP/UDP.

For future work, we plan to investigate the use of REST for Blue Force Tracking based on Multicast with CoAP/UDP.

5.0 SUMMARY AND RECOMMENDATIONS

In this report, we have presented the experiments and findings of IST-150. The group has concentrated its efforts on analysis of protocols and standards that may be applicable to realizing the Message-Oriented Middleware (MOM) Core Service at the tactical level. Keep in mind that current FMN work has, for the most part, targeted higher level and deployed networks, leaving the tactical domain for future spirals. This means that the current recommendations for standards, which include SOAP-based communication like W3C Web services and WS-Notification, for request/response and publish/subscribe communication, are not directly applicable in tactical networks. Our work in predecessor groups (i.e., IST-090 and IST-118) has shown that there is considerable overhead associated with SOAP-based solutions. SOAP-based services need adapting to work in the tactical domain, in which case one may also start considering other approaches that may be better suited to such Disconnected, Intermittent and Limited (DIL) environments. In IST-150, we have particularly been investigating alternate industry standards, in an attempt to identify better suited approaches to realizing MOM in the tactical domain than offered by the SOAP Web services family of standards. For request/response, we have performed comparative evaluations of SOAP and REST services, including alternate transport mechanisms like CoAP to replace the HTTP/TCP connector. For publish/subscribe, we have previously looked at several industry standards besides WS-Notification and found that MQTT was the most promising one of the prolific standards. Alternative and proprietary solutions that perform well in tactical networks, were presented by Suri et al. [7]. But, since we are targeting future FMN spirals with our work, we have focused mostly on solutions based on industry standards in our work. Basing on standards is, in our opinion, preferable in the long run, since standardization promotes interoperability, and hence potential usability in a coalition network.

With respect to publish/subscribe, our main findings include:

- WS-Notification is based on XML and SOAP, it makes it a more resource demanding protocol than MQTT, which is built directly on TCP. As such, WS-Notification consumes more networking resources than MQTT.
- MQTT exhibits a “lighter” and more efficient network performance than WS-Notification, which makes it suitable for mobile tactical environments, where network resources are scarce.
- The OLSR routing protocol generates a large amount of data volume.
- TCP has been designed to assure delivery of all messages. However, in tactical networks (low bandwidth and intermittent), it produces many “spurious” TCP retransmits which leads paradoxically to significantly more lost messages than with unreliable UDP. This indicates that TCP is not well suited for tactical network environments. Our experiments show that UDP is a better match for this kind of message distribution mechanism in tactical networks.

- The use of MQTT with UDP was superior in DIL networks when compared to the standard TCP-based flavours of MQTT. Evolving the MQTT specifications to support UDP is therefore highly recommended. A clustering broker mechanism (as supported by VerneMQ) complies with the principles of a federation of systems and overcome the single-point-of-failure issue present in most MQTT platforms. The trade-off for this additional functionality is, as expected, more resource use than, for example, the simpler mechanism used in Mosquitto (always relying on a "main" broker).
- We have shown that, irrespective of which topic-based publish/subscribe protocol is used, it is possible to achieve interoperability with other solutions though a multi-protocol broker. This means that even if one is to adopt MQTT (or another protocol) at the tactical level, it is still possible to federate this with other protocols in other networks, like WS-Notification [21].

With respect to request/response, our findings include:

- The experiments with a RESTful Military Messaging service showed that REST can be used in an efficient way in resource constrained hybrid tactical network. We have evaluated the service in a military scenario with different transport protocols (HTTP and CoAP) and data formats/compression (JSON, CBOR, EXI). The results showed that the service performed best with CoAP/UDP w.r.t. to transmission times and reliability. CoAP/UDP had significant lower transmission times than HTTP/TCP. Furthermore, CoAP/UDP had much higher reliability in the narrowband network than TCP based protocols (loss rate about 0 – 6 % vs. 35 – 58 %).
- First experiments with SOAP/UDP showed that UDP (without further mechanisms to ensure reliability) is not suitable for narrowband networks when services which require reliable transmissions (like Military Messaging) are used. The same holds for all TCP based protocols.
- The use of compression did provide just a small benefit when the text messages were small. This benefit will be higher if larger messages are used.
- In overall, implementing request/response with REST, CoAP/UDP and JSON (or formats like CBOR or EXI) seems to be a good choice in hybrid tactical networks and should be preferred to SOAP/HTTP or SOAP/UDP.

5.1 Future Work

Though this report presents the state of the art at the culmination of IST-150, we have identified some areas that could benefit from further work. Either as a follow-on to IST-150, or as part of other, related activities:

- Investigate the use of REST for Blue Force Tracking based on Multicast with CoAP/UDP.
- Investigate the use of MQTT variations (multi-broker, UDP based) for efficient message exchange in tactical networks in a federated environment.
- Investigate the implementation of cross-layer mechanisms allowing to fine-tune and adapt message delivery according to actual network capacity, thus avoiding network congestion and message loss.
- OLSR has a high overhead. Thus, protocol improvements (e.g., different update rates) should be investigated or, otherwise, alternative routing protocols better suited for tactical mobile environments using wideband (or narrowband) radios should be deployed and evaluated.

6.0 REFERENCES

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

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

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

[4] Alberts, D.S., (2010). NATO NEC C2 Maturity Model. DoD Command and Control Research Program.

[5] Alberts, D.S., and Hayes, R.E., (2003). Power to the Edge. DoD Command and Control Research Program.

[6] Fielding, R.T., (2000). REST: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis. Irvine, CA, USA.

[7] Suri, N., Breedy, M.R., Marcus, K.M., Fronteddu, R., Cramer, E., Morelli, A. et al. (2019). Experimental Evaluation of Group Communications Protocols for Data Dissemination at the Tactical Edge. 2019 International Conference on Military Communications and Information Systems (ICMCIS). Budva, Montenegro: IEEE.

[8] Suri, N., Nilsson, J., Hansson, A., Sterner, U., Marcus, K., Misirlioğlu, L. et al. (2018). The Angloval Tactical Military Scenario and Experimentation Environment. International Conference on Military Communications and Information Systems (ICMCIS). Warschau, Polen: IEEE.

[9] Wagen, J.-F., Adalid, V., Waeber, G., Buntschu, F., and Bovet, G., (2019). Performance Profiling of Radio Models and Anglova Based Scenarios. International Conference on Military Communications and Information Systems (ICMCIS 2019). Budva, Montenegro.

[10] PredicTAKE, (2019). Retrieved from: https://gitlab.forge.hefr.ch/predictake/.

[11] Nikodemski, A., Wagen, J.-F., Buntschu, F., Gisler, C., and Bovet, G., (2018). Reproducing Measured Manet Radio Performances Using the Emane Framework. IEEE Communications Magazine, vol. 56, p. 155.

[12] EMANE, (2020). Retrieved from: http://www.nrl.navy.mil/itd/ncs/products/emane and https://github.com/adjacentlink/emane/wiki.

[13] Holm, P., (1996). UTD-Diffraction Coefficients for Higher Order Wedge Diffracted Fields. IEEE Transactions on Antennas and Propagation, vol. AP-44, pp. 879-888.

[14] Suri, N., Hansson, A., Nilsson, J., Lubkowski, P., Marcus, K., Hauge, M., Peuhkuri, M., (2016). A Realistic Military Scenario and Emulation Environment for Experimenting with Tactical Communications and Heterogeneous Networks. International Conference on Military Communications and Information Systems (ICMCIS). Brüssel, Belgien: IEEE.

[15] Marcus, K., (2014). Application of the Dynamically Allocated Virtual Clustering Management System to Emulated Tactical Network Experimentation. Proc. SPIE 9079, Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR V. doi:10.1117/12.2054771.

[16] Angelstorf, F., Becker, A., Jansen, N., and Noth, F., (2017). Analysis and Test Framework for the Integration of ICT Systems in the Tactical Domain. ICMCIS 2017.

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

[18] Wireshark, (2019). Retrieved from: https://www.wireshark.org.\

[19] R-Projekt, (2019). Retrieved from: https://www.r-project.org.

[20] ISO/IEC JTC 1 Information Technology, (2016). Message Queuing Telemetry Transport (MQTT) v3.1.1. Retrieved from: https://www.iso.org/standard/69466.html.

[21] Bertelsen, E., Berthling-Hansen, G., Bloebaum, T.H., Duvholt, C., Hov, E., Johnsen, F.T. et al. (2018). 2018 9th IFIP International Conference on New Technologies Mobility and Security (NTMS). Paris, France.

[22] AMQP, (2020). Retrieved from https://www.amqp.org/.

[23] XMPP, (2020). Retrieved from https://xmpp.org/.

[24] OASIS, (2017). Web Services Notification Technical Committee. Retrieved on August 22., 2017 from: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn.

[25] Bloebaum, T.H., and Johnsen, F.T., (2015). Evaluating Publish/Subscribe Approaches for Use in Tactical Broadband Networks. 2015 Military Communications Conference (MILCOM). Tampa, Florida, USA.

[26] Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F., and Alonso-Zarate, J., (2015). A Survey on Application Layer Protocols for the Internet of Things. Transaction on IoT and Cloud Computing, 3(1), pp. 11-17, ISSN: 2331-4761.

[27] Johnsen, F.T., (2018). Using Publish/Subscribe for Short-lived IoT Data. 2nd Workshop on Internet of Things – Enablers, Challenges and Applications (IoT-ECAW'18). Poznań, Poland.

[28] Manso, M., Johnsen, F., Lund, K., and Chan, K., (2018). Using MQTT to Support Mobile Tactical Force Situational Awareness. International Conference on Military Communications and Information Systems (ICMCIS). Warsaw, Poland. Retrieved from: https://doi.org/10.1109/ICMCIS.2018.8398732.

[29] Manso, M., Jansen, N., Chan, K., Toth, A., Bloebaum, T.H., and Johnsen, F.T., (2018). Mobile Tactical Force Situational Awareness: Evaluation of Message Broker Middleware for Information Exchange. ICCRTS 2018. Pensacola, Florida, USA.

[30] Johnsen, F.T., Bloebaum, T.H., Jansen, N., Bovet, G., Manso, M., Toth, A., and Chan, K.S., (2019). Evaluating Publish/Subscribe Standards for Situational Awareness using Realistic Radio Models and Emulated Testbed. 24th International Command and Control Research and Technology Symposium (ICCRTS). Laurel, Maryland, USA.

[31] Manso, M., Guerra, B., Freire, F., Jansen, N., Chan, K., Toth, A., Johnsen, F.T., (2019). Mobile Tactical Forces: Experiments on Multi-broker Messaging Middleware in a Coalition Setting. 24th International Command and Control Research and Technology Symposium (ICCRTS). Laurel, Maryland, USA.

[32] ARL, (2020). Retrieved from: https://www.arl.army.mil/www/default.cfm?page=2485.

[33] OLSR, (2020). Retrieved from http://www.olsr.org.

[34] Bloebaum, T.H., and Johnsen, F.T., (2014). CWIX 2014 Core Enterprise Services Experimentation. FFI-report. Retrieved from: https://www.ffi.no/no/Rapporter/14-01510.pdf.

[35] Mosquitto, (2020). Retrieved from: https://mosquitto.org/.

[36] FuseSource, (2020). Retrieved from: https://github.com/fusesource/mqtt-client.

[37] Johnsen, F.T., Manso, M., and Jansen, N., (2020). Evaluation of Message Broker Approaches for Information Exchange in Disadvantaged Tactical Networks in a Federated Environment. International Command and Control Research and Technology Symposium (ICCRTS).

[38] Manso, M., Brannsten, M.R., and Johnsen, F.T., (2017). A Smart Devices Concept for Future Soldier Systems. 22nd International Command and Control Research & Technology Symposium (ICCRTS). Los Angeles, CA, USA.

[39] IETF, (2020). The GeoJSON format. Retrieved from: https://tools.ietf.org/html/rfc7946.

[40] VerneMQ, (2020). Retrieved from: https://vernemq.com/.

[41] MQTT-udp, (2020). Retrieved from: https://mqtt-udp.readthedocs.io/en/latest/.

[42] Linux Manual Page. (2020). tc-NetEm. Retrieved from: https://man7.org/linux/man-pages/man8/tc-netem.8.html.

[43] NATO IST-118, (2013). IST-118 SOA Recommendations for Disadvantaged Grids: Tactical SOA Profile, Metrics and the Demonstrator Development Spiral. SCI-254 Symposium on Architecture Assessment for NEC. Estonia.

[44] Lindquister, J.J., Johnsen, F.T., and Bloebaum, T.H., (2017). Proxy Pair Optimizations for Increased Service Reliability in DIL Networks. IEEE MILCOM 2017. Baltimore, MD, USA.

[45] CBOR, (2019). Retrieved from: http://cbor.io/.

[46] Efficient XML Interchange (EXI) Format 1.0 (Second Edition). (2019). Retrieved from: https://www.w3.org/TR/exi/.

[47] CoAP, (2019). Retrieved from: https://coap.technology/.

[48] Coalition Networks for Secure Information Sharing. (2013). Final report version 1.0. Retrieved from: http://www.consis.info/content/dam/fkie/consis/en/documents/CONSIS%20Final%20report%20v%201%200.pdf.

[49] Eclipse Jersey, (2019). Retrieved from: https://projects.eclipse.org/proposals/eclipse-jersey.

[50] Eclipse Californium, (2019). Retrieved from: https://www.eclipse.org/californium/.

[51] Jackson, (2019). Retrieved from: https://github.com/FasterXML/jackson.

[52] Clausen, T., Dearlove, C., Jacquet, P., and Herberg, U., (2014). RFC 7181: The Optimized Link State Routing Protocol Version 2. Retrieved from: https://www.rfc-editor.org/rfc/rfc7181.txt.

Annex A – IST-150 PEER-REVIEW PUBLICATIONS AND SCIENTIFIC OUTREACH ACTIVITIES

A.1 INTERNATIONAL CONFERENCE PROCEEDINGS

[1] Frank T. Johnsen, Marco Manso and Norman Jansen. Evaluation of Message Broker Approaches for Information Exchange in Disadvantaged Tactical Networks in a Federated Environment. *ICCRTS 2020*. 2020.

[2] 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.

[3] 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.

[4] Frank T. Johnsen, Lars Landmark, Mariann Hauge, Erlend Larsen and Øyvind Kure. Publish/Subscribe Versus a Content-Based Approach for Information Dissemination. *MILCOM*. 2018. Los Angeles, CA, USA.

[5] Marco Manso, Norman Jansen, Kevin Chan, Andrew Toth, Trude H. Bloebaum and Frank T. Johnsen. Mobile Tactical Force Situational Awareness: Evaluation of Message Broker Middleware for Information Exchange. *ICCRTS*. 2018. Pensacola, Florida, USA.

[6] Trude H. Bloebaum and Frank T. Johnsen. A Hybrid Push/pull C4IS Information Exchange Architecture Concept. *ICCRTS 2018* Concept Paper. 2018. Pensacola, Florida, USA.

[7] Marco Manso, Frank T. Johnsen, Ketil Lund, Kevin S. Chan. Using MQTT to Support Mobile Tactical Force Situational Awareness. *ICMCIS*. 2018. Warsaw, Poland.

[8] Eirik Bertelsen, Gabriel Berthling-Hansen, Trude H. Bloebaum, Christian Duvholt, Einar Hov, Frank T. Johnsen, Eivind Morch, Andreas H. Weisethaunet. Federated Publish/subscribe Services, *2018 9th IFIP International Conference on New Technologies Mobility and Security (NTMS)*. 26-28 February 2018. Paris, France.

[9] Joakim J. Lindquister, Frank T. Johnsen, Trude H. Bloebaum. Proxy Pair Optimizations for Increased Service Reliability in DIL Networks. *IEEE MILCOM*. 2017. Baltimore, MD, USA.

[10] Peter-Paul Meiler, Frank T. Johnsen, Trude H. Bloebaum. Improving Integration between Tactical and HQ Levels by making SOA applicable on the Battlefield. *ICCRTS*. 2017. Los Angeles, CA, USA.

[11] Marco Manso, Frank T. Johnsen, Marianne R. Brannsten. A Smart Devices Concept for Future Soldier Systems. *ICCRTS*. 2017. Los Angeles, CA, USA.

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

A.2 SCIENTIFIC OUTREACH ACTIVITIES

- Organized the third iteration of the SOC-DIL workshop at *IEEE ICC 2018*. May 2018. Kansas City, MO, USA.

Annex B – MOBILE TACTICAL FORCE SITUATIONAL AWARENESS: EVALUATION OF MESSAGE BROKER MIDDLEWARE FOR INFORMATION EXCHANGE

23RD ICCRTS: "MULTI-DOMAIN C2"

Paper ID: 37

Topic 5: Highly Connected, Automated, and Autonomous Forces
Topic 3: Implications of the Internet of Intelligent Things
Topic 6: Interoperability, Integration and Security

Authors

Marco Manso
PARTICLE, Lda.
Portugal

Norman Jansen
Fraunhofer FKIE,
Germany

Kevin Chan and Andrew Toth
Army Research Lab (ARL)
USA

Trude H. Bloebaum and Frank T. Johnsen
Norwegian Defence Research Establishment (FFI)
Norway

Point of Contact
Marco Manso, Rua da Venezuela, n 29, 14 E, 1500-618 Lisbon, PORTUGAL

ABSTRACT

Situational Awareness (SA) is an important aspect of Command and Control and a critical element for the development of mission command capabilities. Developing SA requires information exchange between a tactical force's elements, including in a mobile scenario. We propose the use of the publish/subscribe paradigm to achieve this.

In this paper, we present our experiments in the evaluation of two publish/subscribe mechanisms – based on Web Services Notification (WS-N) and Message Queue Telemetry Transport (MQTT) – applied to a realistic military scenario. Our analysis concluded that WS-N requires more network resources than MQTT to achieve the same functionality. According to our assessment, MQTT was the superior protocol. Furthermore, our evaluation showed that used network protocols, specifically OLSR and TCP, also play a significant role regarding the high use of network resources. In a mobile tactical environment, where network resources are scarce, it is recommended, as future work, to investigate optimisations or even alternative protocols that are better suited for this type of environments.

Keywords: Mobile tactical forces; Tactical networks; Publish/subscribe; Message broker; MQTT; WS-Notification.

B.1 INTRODUCTION

Recent U.S. Army doctrine has identified several requirements categories relevant to the development of mission command capabilities, including a commander's Situational Awareness (SA) and Common Operational Picture (COP) (U.S. Army, 2013). During an operation, as events unfold, decision makers need to receive the information they require in a timely fashion: on the one hand, soldiers, deployed (or opportunistic) sensors and vehicles generate valuable field information that needs to be disseminated

to a potentially large number of recipients, including the Headquarters (HQ). On the other hand, aggregated information – such as a subset of a Common Operational Picture (COP) generated in a HQ – should be distributed to commanders in the field.

The person or system's role in the operation determines the information that it is able to receive, as defined by policies. From a technological viewpoint, a logical approach to support message exchanges from many information producers to many information consumers is through a *publish/subscribe* message exchange pattern. Using publish/subscribe, it is possible to send information from any number of information *producers* to a set of information *consumers* based on their information requirements, effectively decoupling producers from consumers. In the publish/subscribe messaging pattern, this operational need is expressed to the system as an *interest* in a certain type of information, which then drives the information flow.

In our previous work, we have presented a simple proof-of-concept showcasing how publish/subscribe can be applied to a C2 system using smart devices and COTS consumer electronics (smartphones) to generate shared situational awareness between a squad of 9 mobile nodes and 1 fixed node (Manso, Johnsen, and Brannsten, 2017). More recently, we further investigated the use of the MQTT publish/subscribe broker to share information in a tactical environment using a reference scenario developed by the NATO IST-124 group. The work confirmed that the MQTT lightweight approach is appropriate for a mobile environment, however, its server-based nature creates a single point of failure that is problematic in disruptive environments (Manso et al., 2018).

In this paper, we continue our efforts by evaluating the publish/subscribe paradigm in a formalized testbed provided by the U.S. Army Research Laboratory (ARL). We perform a comparative evaluation of two publish/subscribe mechanisms – Web Services Notification (aka WS-Notification or WS-N) (OASIS, 2006) and Message Queue Telemetry Transport (MQTT) (OASIS, 2015, ISO/IEC, 2016) – applied to a realistic military scenario. While NATO is currently recommending WS-Notification for publish/subscribe (NATO C3 Board, 2011), it has been shown that MQTT is a more lightweight approach to publish/subscribe that may be better suited to the tactical domain (Bloebaum and Johnsen, 2015).

The remainder of this paper is organized as follows: In Section B.2, we discuss different publish/subscribe topologies as well as relevant standards. Section B.3 describes the experiments conducted, including presentation of the scenario (and its military relevance), the setting of the testbed (including configuration, broker topology and message publishing frequency) and the choice of broker software. Section B.4 presents the results of the experiments for both WS-N and MQTT (with measurements addressing the network and application layers), together with a comparison analysis. We conclude the paper in Section B.5 where we present our main findings as well as our plans and suggestions for future work.

B.2 PUBLISH/SUBSCRIBE APPROACHES

Event-driven message exchange, or publish/subscribe as it is often called, is a message exchange pattern in which entities that have information they want to share (i.e., *producers* or *publishers*) can publish this information. Information consumers (i.e., *consumers* or *subscribers*) can subscribe to specific types of information they want to receive. When information is published that matches the subscribed interests, it is sent to the subscriber(s). The distribution of information is performed by a *message-broker*.

There are a number of ways in which *subscribers* can indicate which types of information they are interested in when making their subscriptions (Eugster et al., 2003), but the most common approach is basing the subscriptions on so-called *topics*. Topics are keywords that are used to create logical different channels for transmitting information. Information publishers will label their messages with one or more topics, which will be matched to the interests that the consumers have subscribed to using the same topic structure.

Next, several publish/subscribe topologies are presented followed by relevant publish/subscribe standards.

B.2.1 Publish/Subscribe Topologies

The general publish/subscribe message exchange pattern can be implemented in different ways and message related tasks (e.g., producing, forwarding and subscription management) can be distributed differently between the system entities based on the chosen *publish/subscribe topology*, which include the direct messaging topology, the single broker topology, multi-broker topologies and brokerless topologies. The choice of topology will impact the level of complexity in implementation of the different roles, on the amount of network traffic generated in the underlying networks, and on the kind of optimizations one can do to the traffic flow to limit the overhead of the solutions. In this section we present the several different potential topologies, including their main benefits and drawbacks.

B.2.1.1 Direct Messaging Topology

In the **direct messaging topology**, which is shown in Figure B-1, the information producer is responsible for most of the management. An information consumer connects directly to the producer that has the information in which it has an interest and sends its subscription request to that producer. This topology means that the main workload is on the producer, as it is responsible for managing those subscriptions, matching the produced information to subscriptions and forwarding the correct messages to the correct consumers.

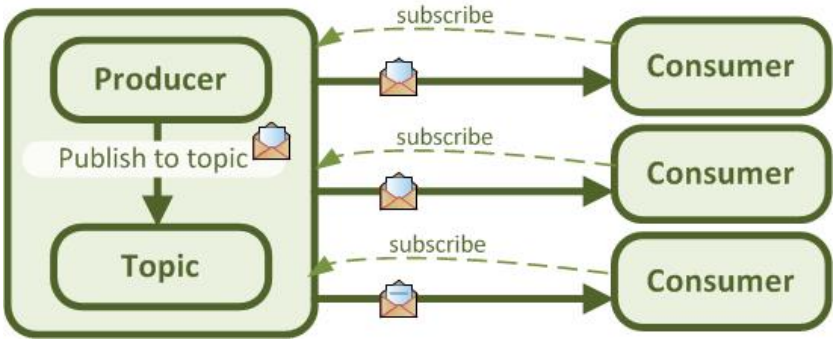


Figure B-1: Direct Publish/Subscribe.

The simple structure of this topology results in a tighter coupling between consumers and producers than for the other topologies described next. The consumers need to know, before any communication can take place, which producers exist, which of those producers offer the information the consumer is interested in, which topics the information is offered under, and how to connect and subscribe to each individual producer. Also, if multiple different producers provide different information on the same topic, the consumers will have to create a subscription to each individual producer. An inherent feature of this topology is that the information producer retains control over how its data is distributed.

B.2.1.2 Single Broker Topology

Decoupling the information producers from the information consumers can be done through the introduction of a *broker*, which functions as an intermediary in all message exchanges. The simplest brokered topology is the one in which a single broker is used, as shown in Figure B-2. The broker takes on the role of handling subscription management, message to topics matching and message forwarding. In this topology all producers send their topic-labelled messages to the broker, and the consumers send their subscription requests to the same broker.

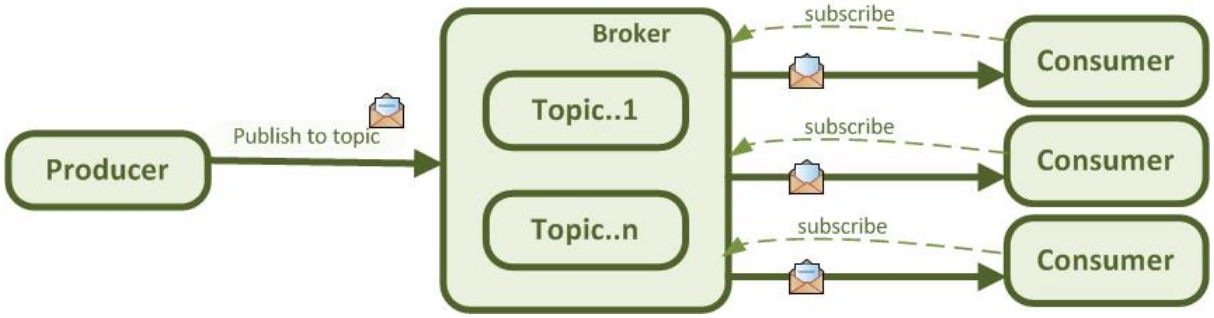


Figure B-2: Brokered Publish/Subscribe with a Single Broker Instance.

In this topology, the reliance on pre-distributed knowledge is lessened, as the consumers do not need to know which producers exist or where the wanted information is generated. They just need to know how to find and connect to the broker. The reliance on having a shared knowledge of a common topic structure is retained, however.

An obvious drawback of the single broker topology is that the broker becomes a single-point-of-failure. In addition, as all message exchange happens through the same broker node, scalability is likely to be an issue (e.g., a single server only supports a limited number of subscribers and/or messages/interests).

Using this topology, or any other topology that uses intermediaries to forward messages, means that the producer surrenders control of how its information is distributed, which in turn means that there has to be a high level of trust in the intermediaries.

B.2.1.3 Multi-Broker Topologies

The single broker topology can be extended to increase scalability and to avoid the single-point-of-failure by increasing the number of intermediary broker nodes. This results in a **multi-broker topology**, as illustrated in Figure B-3 There are a number of different ways in which such a multi-broker topology can be structured, ranging from a fully connected mesh of brokers to hierarchies and mixed deployments to segmented topologies where each broker handles a distinct subset of topics. Common to all the multi-broker topologies is the requirement for controlling the responsibilities of each broker (for instance which producers and consumers a broker serves), and to manage the information flow between the brokers (as most multi-broker topologies require messages to flow between the brokers).

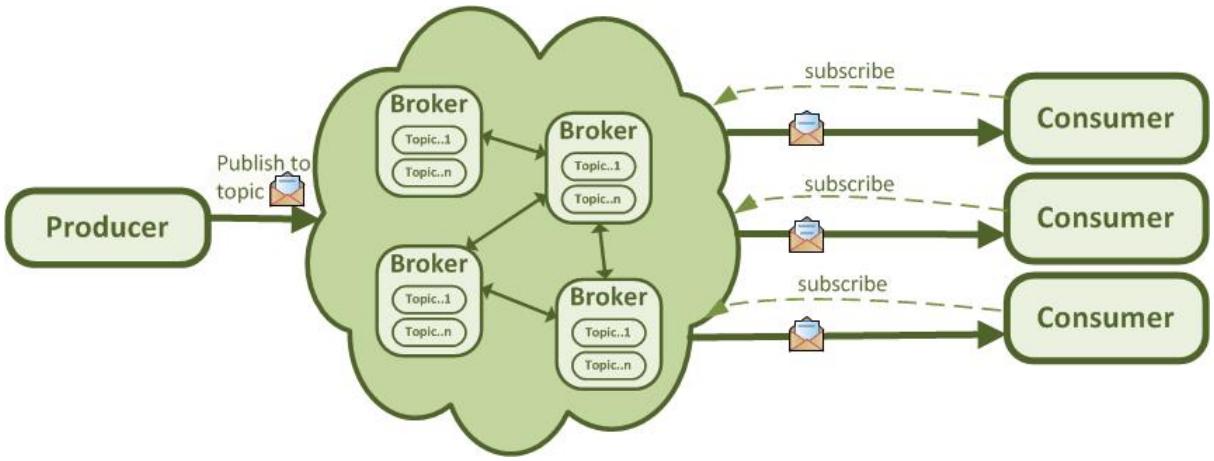


Figure B-3: A Multi-Brokered Publish/Subscribe Topology (Represented by a Partially Connected Mesh of Brokers).

In a **mesh or hierarchical topology**, brokers are connected to each other and are able to exchange information on a peer-to-peer basis. Similar to the single broker topology, these topologies enable a simple and straightforward configuration of producers and consumers, as each of these nodes only interacts with a single broker instance. Therefore, the information flow between a producer connected to broker A and a consumer connected to broker B relies on proper configuration of information sharing between brokers A and B. This inter-broker information flow can be realised either dynamically or manually, however, dynamic control of the information flow requires inter-broker communication beyond what is covered by the publish/subscribe standards.

In a **segmented multi-broker topology**, each broker is responsible for a subset of the information available in the system. As an example, one broker could be responsible for weather forecasts, while another broker could handle position updates for German forces. In such a topology, producers and consumers which produce or consume more than one information type need to connect to multiple brokers, where each broker represents a single-point-of-failure for its information type (note that it is possible to combine segmented and for instance meshed topologies to avoid this issue). The benefit of using a segmented approach is that the amount of coordination that is needed between brokers is limited to deciding which broker is responsible for which topic subset.

B.2.1.4 Brokerless Topologies

In all the above topologies, message exchange relies on end-to-end connections between the consumer and the entity providing information to the client (either the producer in a direct topology, or a broker in a brokered topology). In a brokered topology, the same is true for the connection between the producer and broker. In a tactical networking scenario, this reliance on end-to-end connections might be too strict.

An alternative approach is to use a brokerless distribution mechanism to realise the message exchange. Brokerless distribution mechanisms include Peer-To-Peer (P2P) technologies (Skjegstad, 2009) and Information-Centric Networking (ICN). ICN, in particular, are better suited than more traditional end-to-end communications to cope with the non-trivial communication challenges that military operations present (Morelli et al., 2017) as they implement a distributed information discovery mechanism at the network layer without a single point of failure as it exhibits a decentralized broker behaviour. Another benefit is that by handling this on the network level, it is more tightly coupled with routing.

B.2.2 Publish/Subscribe Standards

There are many prolific publish/subscribe standards, which have been applied to a broad range of applications. For example, the Advanced Message Queuing Protocol (AMQP)[1] is much used in the finance sector as a reliable message queue for exchanging high volumes of transactions. The Extensible Messaging and Presence Protocol (XMPP)[2] is much used as a foundation for chat, but also offers generic publish/subscribe functionality. As such, it has been promoted as a potential carrier for sensor data on the Internet of Things (IoT). Another standard of importance in IoT is MQTT, which currently is more widespread in use than XMPP. Also, MQTT[3] is the underlying protocol of choice for popular messaging apps since they require an efficient one-to-many dissemination mechanism for their users. WS-Notification (OASIS, 2006), a SOAP-based standard from OASIS related to Web services as defined by the World Wide Web Consortium, is NATO's choice for interoperable publish/subscribe.

Work by Bloebaum and Johnsen (2015) has tested AMQP, MQTT, and WS-Notification in a small scale deployment (3 nodes) using real tactical radios, where MQTT was found to show promise, while AMQP

[1] https://www.amqp.org/.

[2] https://xmpp.org/.

[3] http://mqtt.org.

offered reliable communication, but was less efficient than MQTT. (Karagiannis et al., 2015) have performed a survey of relevant IoT data protocols with respect to IoT specifically, where they considered such publish/subscribe protocols as XMPP, MQTT, and AMQP. Also, they considered non-publish/subscribe approaches like CoAP, REST, and Web sockets. Based on recommendations from these studies, we chose to pursue MQTT further as the seemingly best alternative among those protocols tested.

Hence, in this paper we compare WS-Notification to MQTT with regards to performance. For an overview of similarities and differences between these two standards, see Table B-1.

Table B-1: Feature Comparison Between the WS-Notification and MQTT Standards.

Property	WS-Notification	MQTT
Protocol stack	SOAP/HTTP/TCP	TCP
Payload format	XML	Payload agnostic
Quality of service	None built in, but can use additional WS-* standards, e.g., WS-ReliableMessaging	Three delivery semantics: Best effort, At-least-once, or At-most-once delivery
Usage	NATO	IoT, sensor networks, etc.
Topologies supported	Direct and brokered	Brokered
Standardization	(OASIS, 2006)	(OASIS, 2015)

A notable difference between the two standards is that WS-Notification is based on XML and SOAP, inherently making it a more resource demanding protocol than MQTT which is built directly on TCP. As such, WS-Notification is expected to consume more networking resources than MQTT.

B.3 EXPERIMENTS

This section describes the scenario used for the purpose of the experiments, followed by the experimental testbed and the publish/subscribe software used.

B.3.1 Scenario Subject

In terms of military relevant scenario development, the recently completed NATO STO/IST-124 "Heterogeneous Tactical Networks: Improving Connectivity and Network Efficiency" working group has developed a scenario called Anglova that *includes detailed mobility patterns for a battalion-sized operation over the course of two hours, which has been developed by military experts in planning and performing real exercises* (Suri et al., 2016).

The scenario and vignettes include a narrative, mobility scripts and Order-Of-Battle (ORBAT) for several companies operating in a fictitious location of Anglova to conduct several missions. The networks and assets are represented by 243 nodes. Anglova has enabled researchers to study routing in various settings to understand scalability and performance issues of the routing protocol involving highly mobile nodes.

The scenario for our experiments, built based on Anglova, consists in a (fictional) attack involving a mechanized battalion against insurgent's forces. Specifically, we employ Vignette 2 of the Anglova's scenario limited to one mechanized battalion constituted by 24 mobile nodes (military vehicles) that is part

of a Military Contingent coordinated by the Coalition HQ. The battalion nodes are equipped with tactical radios that are used to exchange information. We use nodes' location information and radio signal pathloss between nodes recoded in the Anglova study.

One can anticipate a set of services that need to be supported to give decision makers adequate information to achieve SA. One such service is that of friendly force information (aka Blue Force Tracking (BFT)). Accurate and timely BFT is important to avoid friendly fire, so-called "blue on blue" situations. In the experiments in this paper, we use the NATO Friendly Force Information (NFFI) data format for the BFT service, described in draft STANAG 5527. NFFI has originally emerged to support interoperable BFT in the Afghan Mission Network (IST-118, 2013) and since then it has been successfully used in many contexts. Hence, we consider it a representative standard payload in the publish/subscribe evaluation.

From (Manso et al., 2018), Figure B-4 shows the nodes' location and evolution over time. Note that the actual nodes' location information presented in this paper has been modified in order to be unclassified.

Figure B-4: Visualisation of the Vehicles' Location and History (Blue Line). Left image: Vehicles are starting to move. Right image: Vehicles' location at the end of the exercise.

B.3.2 Experimental Testbed Setup

The experimental testbed used to conduct experiments is the Network Science Research Laboratory (NSRL)[4] established by the U.S. Army Research Laboratory (ARL). The NSRL provides network emulation capabilities and military relevant data and scenarios for the testing and evaluation of various networking oriented technologies and approaches. The facility has enabled collaboration between ARL researchers and those from other organizations. Additionally, infrastructure in the way of dynamic virtualization has been developed to assist in the execution of experiments in the NSRL. To enable repeatability and scalability of experimentation, ARL has also developed a platform called Dynamically Allocated Virtual Clustering (DAVC) Management System. DAVC provides the capability to dynamically create and deploy virtual clusters of heterogeneous nodes as specified by virtual machines.

[4] https://www.arl.army.mil/www/default.cfm?page=2485.

Experiments are completely reconfigurable through the DAVC interface, with minor modifications to parameters defined in custom scripts (e.g., nodes' location and radio signal pathloss between nodes, as provided by Anglova).

Both the Anglova scenario and DAVC are releasable through NATO collaboration.

The Anglova scenario, incorporating WS-N or MQTT broker messaging services, was setup in the NSRL environment. For that, WS-N and MQTT services were installed onto the Virtual Machine (VM) template of the Anglova scenario to enable the publish/subscribe position location information services. The experiments use the **single broker topology** described in Section B.2. The VM template is deployed to nodes during runtime of the scenario. This is illustrated in Figure B-5.

For network emulation, we use the Extendable Mobile Ad hoc Network Emulator (EMANE) that provides – besides the emulation of the radio links – signal propagation and mobility representation to the experiment to create a more realistic environment. The mobility information was drawn from Anglova recorded data.

The emulation allows for various types of routing and radio models to be used; in this scenario we use Optimized Link State Routing (OLSR)[5] (OLSR, 2016) V1 via the OLSR Daemon (OLSRD) on each virtual machine representing a node in the scenario with wireless links based on the EMANE RFPipe model. The RFPipe model was configured to emulate wideband tactical radios operating at 300 MHz with a 250 KHz bandwidth and 175 kbit/s data rate. OLSR was configured with a Hello Interval of 2 seconds, Hello Validity Time of 20 seconds, Topology Control Interval of 8 seconds, and Topology Control Validity time of 80 seconds.

In the initial set of experiments, we ran the first 30 minutes of the Anglova scenario vignette excerpt consisting of 24 nodes. We set up a DAVC cluster of 24 "Anglova" nodes and one controller node. The controller node is used as the orchestration node and is not represented in the experiment nor does it take part in the scenario. Node 1 for this experiment is arbitrarily established as the broker node (i.e., runs the WS-N or MQTT server). It also has a subscriber service running on it (i.e., subscribes to and receives messages from all publishers). We note that the platform allows for any configuration of broker and subscriber services.

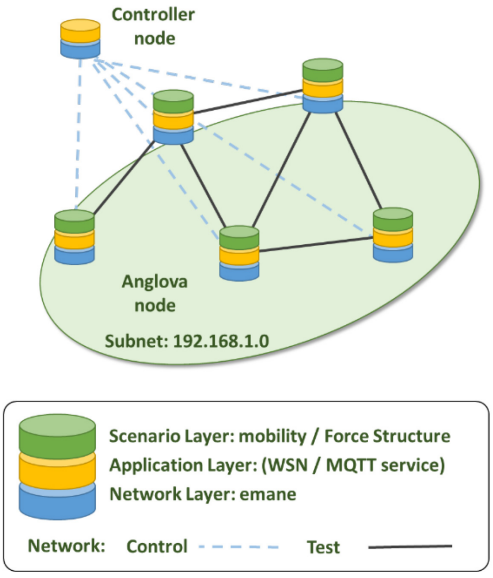


Figure B-5: Architecture of Network Experiment Including Network Emulation, Application and Scenario Layers.

[5] http://www.olsr.org.

Additionally, to facilitate the execution of these experiments, we have created services that launch EMANE and the Anglova configuration. We also have Linux shell scripts that can start and stop the publisher services for both WS-N and MQTT as well as gathering generated pcap and log files.

For our scenario, we set the publishing of the node locations (i.e., NFFI messages) every 10 seconds. In this experiment, we have Nodes 2 through 24 as publishers.

B.3.3 Publish/Subscribe Software

The publish/subscribe message broker middleware selected for this work includes open source implementations and closed-source software developed in-house at the Norwegian Defence Research Establishment (FFI) as follows:

- For **WS-Notification broker**, we use *microWS-N*, which is a closed-source FFI implementation of a subset of the WS-Notification family of standards. This implementation has been tested for interoperability at the NATO Coalition Warrior Interoperability eXploration, eXperimentation, eXamination, eXercise (CWIX) in 2014. There, we found that the standard functions microWS-N offers were compliant with WS-Notification version 1.3, which is the most recent specification (Bloebaum and Johnsen, 2014).
- For **MQTT messaging broker**, we initially used the Moquette MQTT broker[6]. However, the broker had issues with deadlocks and race conditions, leading to message loss and the broker freezing up[7]. We then switched to the open source Mosquitto from the Eclipse foundation, which is freely available online[8]. It should be noted, though, that Mosquitto also has some stability issues, notably when used together with transport layer security (TLS) and Web sockets. At the time of writing this paper, these issues are known but still unresolved[9]. So, to ensure the stability of our experiments, we used Mosquitto without TLS enabled to avoid crashes and we made the assumption that security (as in confidentiality and integrity) would be ensured at the radio and network levels (e.g., through IP-Sec or link-layer encryption). Also, we also excluded the use of Websockets in the experiments.

In addition to the brokers, we also needed to implement producers and consumers to use in the evaluation:

- For WS-Notification, we used the closed source client libraries of microWS-N as the basis for setting up subscriptions and publishing data.
- For MQTT, the producer and consumer software were implemented using the Fuse source library[10]. Since messages relate to location information periodically produced, the MQTT clients were configured to request at-most-once delivery from the broker (i.e., MQTT QoS = 0 that is the most efficient but less reliable setting).

As first described in Section B.3.2, Node 1 functions as broker node (i.e., runs the WS-N or MQTT server) and a consumer node (i.e., runs the consumer software subscribing to all messages). Nodes 2 to 24 run the producer software that publishes a NFFI message each 10 seconds.

As a final remark, it is worth noting that the results in this paper stem from using the above mentioned software and using brokered publish/subscribe with a single broker (see Section B.2.1.2). Using different implementations may yield somewhat different results regarding performance, robustness and stability.

[6] http://andsel.github.io/moquette/.

[7] This known issue is still unresolved: https://github.com/andsel/moquette/issues/208.

[8] https://mosquitto.org/.

[9] Mosquitto segmentation fault during client connection: https://github.com/eclipse/mosquitto/issues/406.

[10] https://github.com/fusesource/mqtt-client.

B.4 EXPERIMENTS RESULTS AND EVALUATION

In this section, the results and evaluation of the experiments are presented, with the objective to provide a comparison between the two different publish/subscribe standards (WS-N and MQTT) using a realistic tactical emulation environment (provided by ARL NSRL) based on a relevant military scenario (Anglova).

Our evaluation approach makes use of logging information from both the **network layer** as well as the **application layer**. For the **network layer**, we logged the network traffic via "tcpdump" resulting in "pcap" files. For the **application layer**, we logged the application traffic via a logging interface which we defined by a JSON schema. The logging interface was implemented into the publisher and subscriber services.

For the analysis of the application log files, we used analysing tools from the *Analyse and Test environment* (*AuT*) project of Fraunhofer FKIE (Angelstorf, Becker, Jansen, and Noth, 2017).

B.4.1 WS-N with OLSR and Broadband Radio Links

In this setup, we deploy the WS-N broker *microWS-N* (see Section B.3.3) together with one WS-N subscriber on Node 1 (the HQ node). Nodes 2 to 24 (23 nodes in total) each run a WS-N producer software publishing a NFFI message every 10 seconds. The measurements pertaining to network and application layers are presented next.

B.4.1.1 Network Layer

By analysing the network level log files (packet captures) the consumed data rates shown in Figure B-6 could be obtained. The figure shows the data rates related to WS-N-based messages and OLSR-based messages each divided into the different protocol layers. The WS-N-based communication consumes 42 kbit/s of the available data rate of the radios (175 kbit/s), which consists of 23 kbit/s of WS-N- (i.e., HTTP-)based packets, 12 kbit/s related to TCP, 4.1 kbit/s related to IP headers and 2.9 kbit/s related to Ethernet headers. The routing protocol (OLSR) generates an overall traffic of 98 kbit/s.

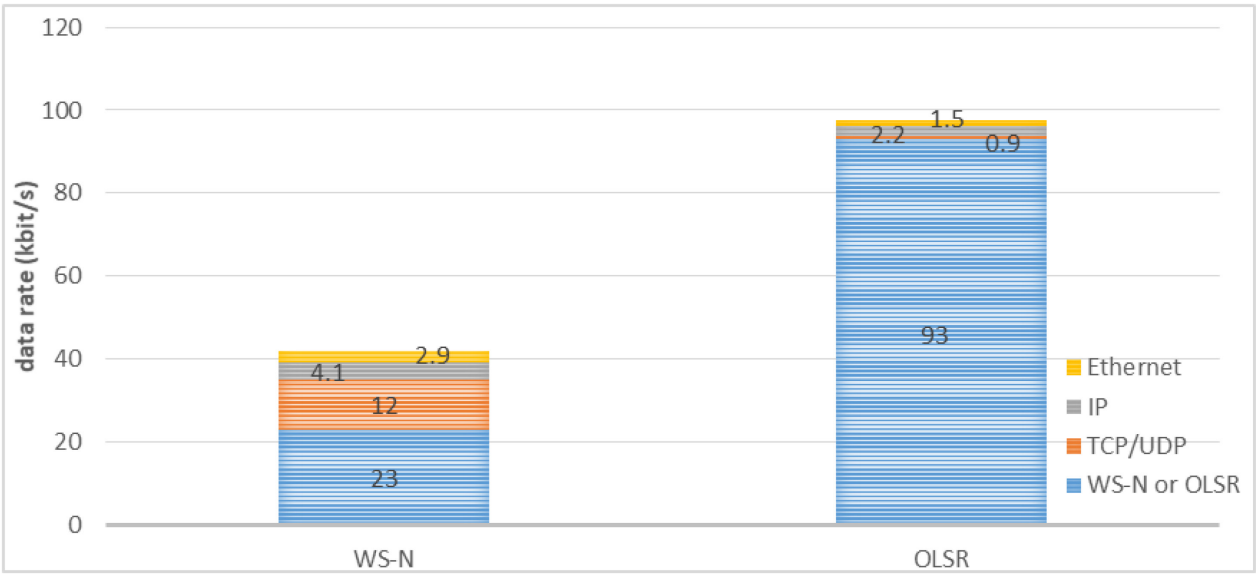


Figure B-6: Consumed Data Rates of WS-N and OLSR Divided into Protocol Layers.

Overall, 2955 WS-N-based NFFI messages were sent. The size (application content) of each NFFI message was 1939 Bytes. All messages were delivered. The network logs show that 3594 TCP retransmissions were produced. Most of them were of type "spurious", which means that a packet was retransmitted, because an acknowledgement arrived too late at the sender and messages are thus unnecessarily retransmitted, which leads to a larger communication overhead.

B.4.1.2 Application Layer

The application logs consist of logging entries of the senders (publishers) of NFFI messages and logging entries of the receiver (broker and subscriber) of these messages. This approach allows us to calculate the overall transmission times of NFFI messages, which represent the age of the positions as observed by the user at the receiver node. The results were analysed with help of analysing tools of AuT project and are shown in Figure B-7 and Figure B-8. Figure B-7 shows as a boxplot diagram the transmission times of all publishers including the maximum values, whereas Figure B-8 shows an enlarged view of the diagram showing the first quartiles, medians and third quartiles in more detail.

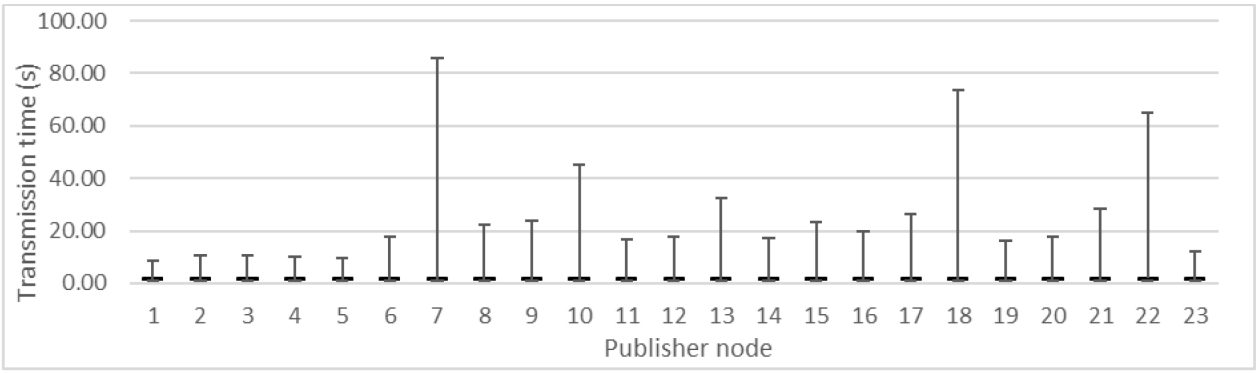


Figure B-7: Transmission Times of WS-N-Based NFFI Messages (Whole Diagram).

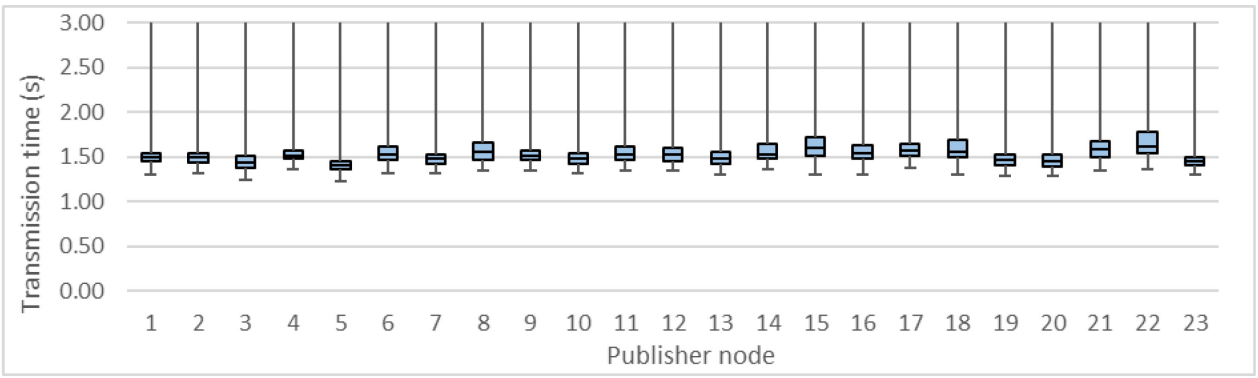


Figure B-8: Transmission Times of WS-N-Based NFFI Messages (Enlarged View).

In all 2955 messages were published. None of these were lost. The overall median was 1.5 s. For each publisher, the median transmission time was between 1.4 s and 1.7 s as shown in Figure B-8. As Figure B-8 shows, most of the messages were in this time interval. But there are some extreme values which took much longer. For each publisher, the maximum transmission time was between 8 s and 86 s as shown in Figure B-7.

B.4.2 MQTT with OLSR and Broadband Radio Links

In this setup, we deploy the MQTT broker *Mosquitto* (see Section B.3.3) together with one MQTT subscriber on Node 1 (the HQ node). Nodes 2 to 24 (23 nodes in total) each run an instance of the MQTT producer software publishing a NFFI message every 10 seconds. The measurements pertaining to network and application layers are presented next.

B.4.2.1 Network Layer

The analysis of the network level log files (packet captures) results in the consumed data rates shown in Figure B-9, which were measured at the broker/subscriber node. The figure shows the data rates which are related to MQTT-based messages and OLSR-based messages, each divided into the different protocol layers. The MQTT-based traffic consumes 23 kbit/s of the available data rate of the radios (175 kbit/s), which consists of 11 kbit/s of MQTT-based packets, 6 kbit/s related to TCP, 3.2 kbit/s related to IP headers and 2.2 kbit/s related to Ethernet headers. The routing protocol (OLSR) generates an overall traffic of 93 kbit/s.

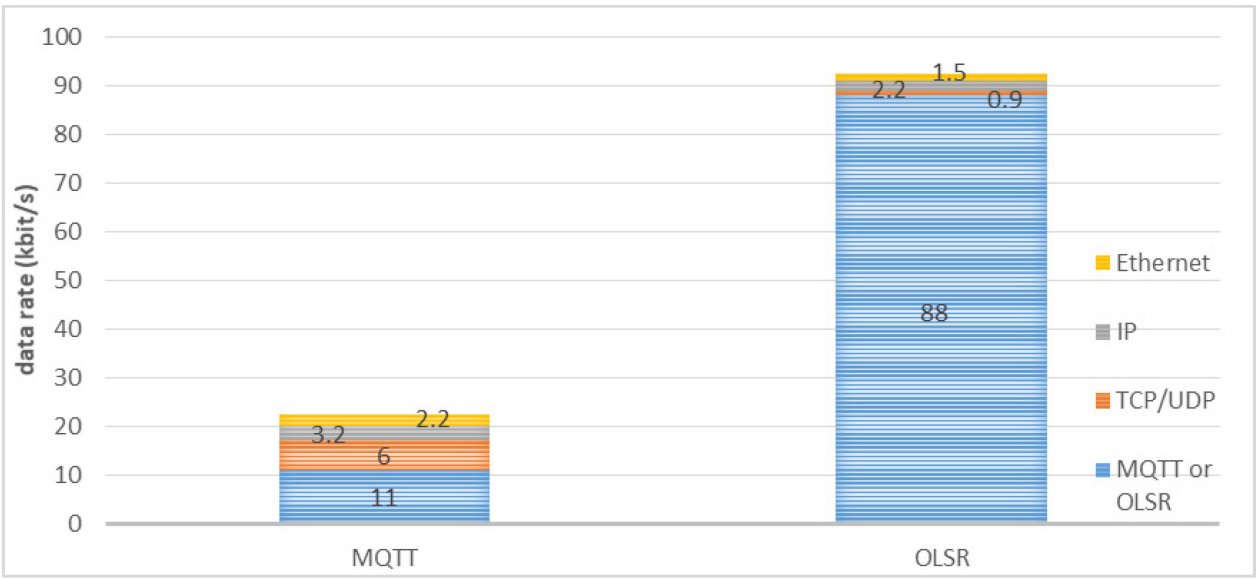


Figure B-9: Data Rates of MQTT and OLSR Divided into Protocol Layers.

Overall, 3073 MQTT messages were sent. The size (content) of each message was 909 Bytes (WS-N's size increase was due to extra overheads from using SOAP and XML).

All messages were delivered. The network logs show that 1954 TCP retransmissions were produced. Most of them were of type "spurious" similar to the setup with WS-N.

B.4.2.2 Application Layer

In Figure B-10 and Figure B-11 the average transmission times of the messages are shown for each publisher in a boxplot diagram. Figure B-10 shows the whole diagram including the maximum values, whereas Figure B-11 shows an enlarged view of the diagram showing the first quartiles, medians and third quartiles in more detail.

As mentioned above, none of the overall 3073 messages were lost. For each publisher, the median transmission time was between 0.7 s and 0.9 s as shown in Figure B-11. The overall median was 0.8 s. As Figure B-11 shows, most of the messages were in this time interval. But there are some extreme values which took much longer. For each publisher, the maximum transmission time was between 5 s and 92 s as shown in Figure B-10.

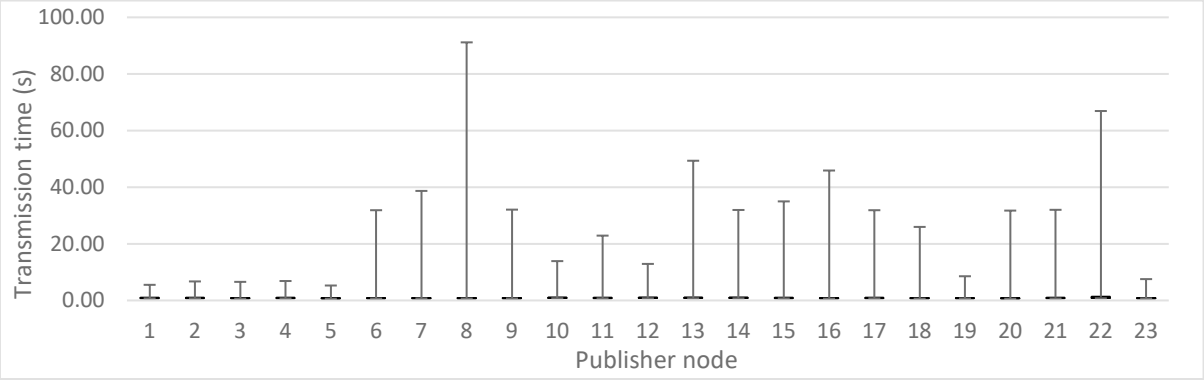


Figure B-10: Transmission Times of MQTT-Based NFFI Messages (Whole Diagram).

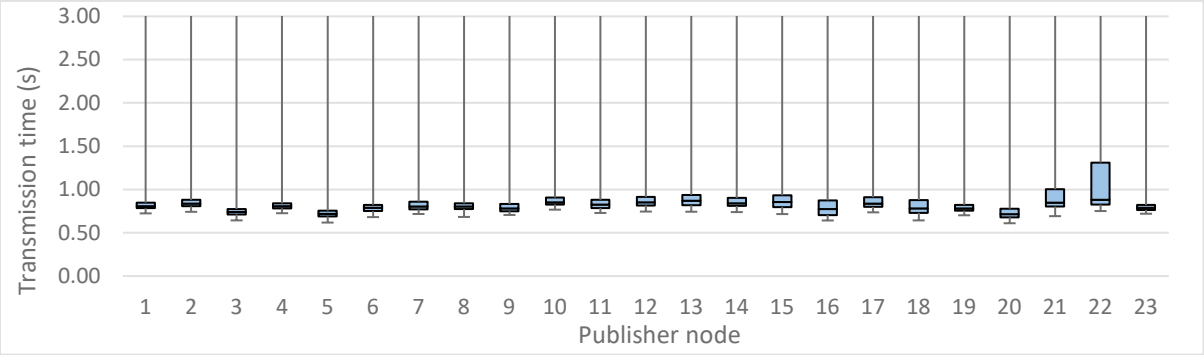


Figure B-11: Transmission Times of MQTT-Based NFFI Messages (Enlarged View).

B.4.3 Comparison Analysis and Results

A comparison between results obtained with WS-N and MQTT is presented next. The measurements used to support our analysis are presented in Table B-2.

Table B-2: Results from Experiments for WS-N and MQTT.

	WS-N	MQTT
Network Layer		
Data rate (kbit/s)	42	23
Message size (bytes)	1939	909
TCP retransmissions	3594	1954
Application Layer		
Messages lost	0	0
Delay (median) (sec)	1.5	0.8
Maximum Tx Time (sec)	86	92

From the evaluation of the experiments with WS-N and MQTT as message brokers, it can be seen that MQTT outperforms WS-N:

- Overall (including the whole communication stack) MQTT consumes about half the data rate than WS-N (23 kbit/s vs. 42 kbit/s) of the available data rate of 175 kbit/s which is provided by the radios.
- MQTT generated message size is less than half the WS-N's message size (909 bytes vs. 1939 bytes).
- MQTT caused about half TCP retransmissions than WS-N (1954 vs. 3594).
- Consistently, the median message transmission times measured on the application level were half as large with MQTT (0.8 s) compared to WS-N (1.5 s).
- A few large delays were observed, being the maximum observed pertaining to MQTT (92 seconds) closely followed by WS-N (86 seconds). These, however, seem more related to TCP protocol or networking aspects, and not associated to the message broker.

As expected, MQTT exhibits a "lighter" and more efficient network performance than WS-N, which makes it suitable for mobile tactical environments, where network resources are scarce. Additional optimisations and configurations can be pursued aiming to further improve network performance.

Concerning network-related measurements, it is worth mentioning the following aspects:

- The network captures showed that most of the data rate volume is caused by the OLSR routing protocol (70% and 80% of the data rate for WN-S and MQTT respectively). This suggests investigating OLSR improvements (e.g., different protocol update rates) or deploying alternative routing protocols better suited for tactical mobile environments using wideband (or narrowband) radios.
- TCP assured delivery of all messages. However, the evaluation of the network logs showed that both WS-N and MQTT setups produced many "spurious" TCP retransmits[11]. This indicates that TCP is not well suited for the kind of wireless networks used in this scenario. Thus, alternative transport protocols should be sought. For example, the use of an UDP-based protocol could reduce the message overhead and increase the utilization of the network bandwidth by eliminating the coordination problems between network links and the congestion control mechanism of TCP.

B.5 CONCLUSION

In this paper, we have evaluated two publish/subscribe standards (WS-N and MQTT) using the Anglova scenario. The scenario was driven by the ARL NSRL testbed, which offers large scale emulation capabilities. We used a combination of open source software and software developed in-house at FFI to realize the publish/subscribe infrastructure.

The experiments comprised a cluster of 25 nodes, where 24 represented operational nodes (part of the mechanised battalion) and the 25th node functioned as the experiment controller. Nodes were connected by broadband wireless links using OLSR.

Following the experiments' execution, application logs and packet captures were collected and analysed using Fraunhofer FKIE tools and expertise.

[11] Here, "spurious" means that a packet was unnecessarily retransmitted because the respective acknowledgement arrived too late at the sender. Since the congestion control mechanism of TCP interprets "lost" (actually belated in this case) acknowledgements as buffer overflows, the congestion window is unnecessarily decreased, which leads to a reduced throughput.

Overall, our analysis concluded that WS-N requires more network resources than MQTT to achieve the same functionality. This leads to increased network resource use (about twice compared to MQTT), as well as an increased transmission time (also about twice) of end-to-end messaging. We can conclude, that for the part of the scenario we evaluated, MQTT was the superior protocol based on the considered metrics.

Our evaluation also showed that the used network protocols, specifically OLSR and TCP, also play a significant role regarding the use of network resources: OLSR generated 70% or 80% of the overall traffic for WS-N and MQTT respectively, while TCP produced many "spurious" packet retransmissions. There is a need to investigate optimisations or even alternative protocols that are better suited for tactical mobile networks (e.g., UDP replacing TCP).

It is worth noting that the results observed stem from using specific software implementations. Using different implementations may yield different results regarding performance and stability, though the overall differences between WS-Notification and MQTT should still be evident due to the differences between these standards.

B.6 FUTURE WORK

From the experiments we have seen that MQTT outperforms WS-N in the scenario used. Therefore, we plan to continue investigating MQTT in more detail, evaluating different options, configurations and software implementations (supporting e.g., TLS and what the overhead of security will be in these cases). In addition, since the experiments indicate that TCP is not well suited for tactical wireless networks (wideband or narrowband), we aim to investigate brokers based on MQTT-SN (MQTT for Sensor Networks), since MQTT-SN is based on UDP.

The central nature of the single broker also makes it a single point-of-failure, which must be avoided in a real deployment. As such, future work will cover multi-broker publish/subscribe with different topologies, including investigating where the brokers should be placed in the network. Matching the broker placement with the structure of the tactical networks and the information needs of the users at the lower tactical level can be used to help limit the amount of network traffic. In this context, is will be important to investigate the use of dynamic service discovery (Skjegstad, 2009) to allow publishers and consumers to discover and leverage brokers in a plug-and-play manner, using zero-configuration networking during runtime.

Exploratory work will also be conducted for brokerless topologies (e.g., ICN mechanism) that mainly operates at the network level and does not exhibit a single point-of-failure.

Next experiments will also consider a more realistic emulation of tactical radio networks by using a model which is tailored to specific tactical radios (broadband and narrowband tactical waveforms).

B.7 ACKNOWLEDGEMENTS

This work has been performed in the context of the NATO STO/IST-150 "NATO Core Services profiling for Hybrid Tactical Networks" group. Thanks to Marianne R. Brannsten for providing the publish/subscribe figures. Thanks to Andreas Becker for supporting the evaluation of the experiments.

B.8 REFERENCES

Angelstorf, F., A. Becker, N. Jansen, and F. Noth. Analysis and Test Framework for the Integration of ICT Systems in the Tactical Domain. ICMCIS 2017, Oulu, Finland, May 15 – 16, 2017.

Bloebaum, T., and F. Johnsen. CWIX 2014 Core Enterprise Services Experimentation. FFI-report. November 2014. ISBN 978-82-464-2460-6. Available at: Https://Www.Ffi.No/No/Rapporter/14-01510.Pdf.

Bloebaum, T., and F. Johnsen. Evaluating Publish/Subscribe Approaches for Use in Tactical Broadband Networks. IEEE MILCOM 2015, October 26 – 28, Tampa, Florida, 2015. DOI: 10.1109/MILCOM .2015.7357510.

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

ISO/IEC 20922:2016. Information technology – Message Queuing Telemetry Transport (MQTT) v3.1.1. ISO/IEC JTC 1. Information Technology, June 2016. https://www.iso.org/standard/69466.html.

Karagiannis, V., P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate. A Survey on Application Layer Protocols for the Internet of Things. Transaction on IoT and Cloud Computing 3(1), 11-17, 2015. ISSN: 2331-4761.

Manso M., F. Johnsen, and M. Brannsten. A Smart Devices Concept for Future Soldier Systems. ICCRTS 2017, Los Angeles, USA, Nov 6 – 8, 2017.

Manso M., F. Johnsen, K. Lund, and K. Chan. Using MQTT to Support Mobile Tactical Force Situational Awareness. ICMCIS 2018, Warsaw, Poland, May 22 – 23, 2018.

Morelli, A., M. Tortonesi, C. Stefanelli, and N. Suri. Information-Centric Networking in Next-generation Communications Scenarios. Journal of Network and Computer Applications. Volume 80 Issue C, February 2017 (Pages 232-250) DOI: 10.1016/j.jnca.2016.12.026

NATO C3 Board. Core Enterprise Services Standards Recommendations: The SOA baseline profile v.1.7. Enclosure 1 to AC/322-N(2011)0205. NATO Unclassified releasable to EAPC/PFP, 11 Nov 2011.

NATO IST-118. IST-118 SOA Recommendations for Disadvantaged Grids: Tactical SOA Profile, Metrics and the Demonstrator Development Spiral. Paper presented at the SCI-254 Symposium on “Architecture Assessment for NEC”. Estonia, May 14 – 15, 2013.

OASIS. OASIS Web Services Notification (WS-N) TC. 2006. Available at: https://www.oasis-open.org/committees/WS-N/.

OASIS. MQTT Version 3.1.1 Plus Errata 01. 10 December 2015. Available at: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.

Skjegstad, M. Search+: An efficient P2P Service Discovery Mechanism. Master Thesis. University of Oslo, 2009. Available at: http://urn.nb.no/URN:NBN:no-23707.

Suri, N., A. Hansson, J. Nilsson, P. Lubkowski, K. Marcus, M. Hauge, K. Lee, B. Buchin, L. Misirlioglu, and M. Peuhkuri. A Realistic Military Scenario and Emulation Environment for Experimenting with Tactical Communications and Heterogeneous Networks. IEEE ICMCIS 2016. May23 – 24 2016, Belgium, Brussels.

U.S. Army. U.S. Army Mission Command Strategy. June 2013. Available at: http://usacac.army.mil /cac2/MCCOE/.

Annex C – MOBILE TACTICAL FORCES: EXPERIMENTS ON MULTI-BROKER MESSAGING MIDDLEWARE IN A COALITION SETTING

24th ICCRTS: "MANAGING CYBER RISK TO MISSION"

Paper ID: 35

Topic 5: Highly Connected, Automated, and Autonomous Forces
Topic 6: Interoperability, Integration and Security

Authors

Marco Manso and Barbara Guerra
PARTICLE, Lda.
PORTUGAL

Ret.Col. Fernando Freire
Portuguese Army
PORTUGAL

Norman Jansen
Fraunhofer FKIE,
GERMANY

Kevin Chan and Andrew Toth
Army Research Lab (ARL)
USA

Trude H. Bloebaum and Frank T. Johnsen
Norwegian Defence Research Establishment (FFI)
NORWAY

Point of Contact

Marco Manso, PARTICLE, Lda, PORTUGAL, marco@particle-summary.pt

ABSTRACT

The environment in which tactical forces operate is characterized by disconnected intermittent connectivity and limited bandwidth (DIL). This environment significantly constrains the application of widely used technologies. These characteristics require that technology and standards need to be carefully selected and that appropriate profiles are set. The NATO IST-150 research group is tackling this challenge by analysing standards and technologies appropriate for tactical networks, obtaining promising results with the Message Queue Telemetry Transport (MQTT).

As a result of the NATO IST-150 activities, this paper presents the application and evaluation of MQTT technologies in the context of a three nation coalition setting (i.e., federated-based setup) – specifically NOR (Norway), Portugal (PRT) and the United States of America (USA) – supporting information exchange between brokers, while preserving the Nations' ownership (and control) over its resources.

Using a simplified version of the Blue Force Tracking (BTF) service, the experiment demonstrates the MQTT ability to propagate messages across the whole coalition. Moreover, the experiment results show a high-reliability and low latency in delivering messages (including between coalition brokers).

The North Atlantic Treaty Organization (NATO) places a high priority in achieving technical interoperability between Allied forces, including at the tactical edge, in which IST-150 findings and recommendations will provide valuable inputs.

Keywords: Mobile forces; MQTT; Multi-Brokers; NATO; Situational awareness; Publish-Subscribe.

C.1 INTRODUCTION

The North Atlantic Treaty Organization (NATO) places a high priority in achieving interoperability between Allied forces. Defined as "the ability for Allies to act together coherently, effectively and efficiently to achieve tactical, operational and strategic objectives" (NATO, 2017). Understanding that interoperability encompasses various dimensions – such as doctrine, procedures, human, language and technology – in this work we are addressing aspects dealing with Information Technology (IT) technical interoperability, specifically on information exchange between IT systems in a coalition environment. In this regard, NATO promotes the Federated Mission Networking (FMN) initiative that was created with the purpose to improve information sharing during common missions, aiming FMN affiliates to contribute *Federated Mission Networking-ready forces to a mission on short notice and with minimal preparation* (NATO, 2015).

Up to now, most of FMN's standardization and profiling work has focused on static and deployed networks, where networking resources are stable and plentiful. However, tactical forces operate on significantly different conditions, that is, deployed tactical (mobile) networks – the so called tactical edge –, an environment that is characterized by disconnected intermittent connectivity and limited bandwidth (DIL). This means that different and alternative profiles and standards are required.

NATO has supported several Research Task Groups (RTG) on Information Systems Technology (IST) addressing standardization and profiling for tactical DIL networks, including the NATO RTG IST-090 (SOA Challenges for Real-Time and Disadvantaged Grids), IST-118 (SOA Recommendations for Disadvantaged Grids in the Tactical Domain) and the on-going IST-150 (NATO Core Services profiling for Hybrid Tactical Networks)[1]. Building on the findings of these groups, this paper explores novel approaches and open technologies for efficient information exchange in constrained settings. Specifically, it experiments with the publish-subscribe paradigm using multi-broker topologies – where each message broker is managed by a different nation – demonstrating bi-directional message exchange between brokers and, ultimately, between coalition members.

This paper is structured as follows: Section C.2 presents the background work used in this paper, consisting in past NATO activities, previous research work and the chosen publish-subscribe paradigm for these experiments (i.e., MQTT), together with its relevant features and implications; Section C.3 describes the conducted experiment, starting by explaining its purpose, scenario and setup, to then presents its results. Section C.4 concludes the paper, presenting its main findings and recommendations for future work.

This paper results from activities conducted within the NATO RTG IST-150 "NATO Core Services Profiling for Hybrid Tactical Networks".

C.2 BACKGROUND WORK

The environment in which tactical forces operate is characterized by disconnected intermittent connectivity and limited bandwidth (DIL). This environment significantly constrains the application of widely used Internet-based technologies (designed for stable and well performing (broadband) networks). These characteristics require that technology and standards need to be carefully selected and that appropriate *profiles* are set.

NATO has supported several RTGs addressing standardization and profiling for tactical DIL networks that form the foundations of this work, as introduced next.

The IST-090 and IST-118 studied the application of Web and Internet-based approaches in "disadvantaged" tactical networks, including applying Services Oriented Architecture (SOA) principles, Internet-Protocol

[1] See list of activities in: https://www.sto.nato.int/Lists/test1/webview.aspx.

(IP)[2] and Web services. IST-090 (NATO IST-090, 2014) demonstrated that SOA could work at lower levels than previously thought, providing guidance and best practices on the application of SOA in tactical networks (including suggestions for extensions to the NATO SOA Baseline (NATO C3 Board, 2011)). IST-118 demonstrated the application of SOA services in a mobile environment constituted by a force connected by broadband mesh radios (Manso et al., 2015) using the OASIS standard WS-Notification (WS-N)[3] as publish-subscribe service and the functional service NATO Friendly Force Information (see NATO STANAG 5527)[4]. The experiments also showed that the WS-N is a resource heavy protocol and its application at the tactical level requires applying proprietary optimizations (hence, causing interoperability issues) (P. Meiler et al., 2013).

IST-150 continued the activities of IST-090 and IST-118 by analysing new standards and defining a set of profiles appropriate for the deployment of services in the context of tactical networks. The group evaluated the use of lightweight and resource constrained protocols, choosing the standard Message Queue Telemetry Transport (MQTT) (see Section C.1) for experimentation, given its open-source availability, low footprint, wide use and extensive set of features. Despite NATO recommendation on the use of Web-based services, including WS-Notification for publish/subscribe (NATO C3 Board, 2011), it has been shown that MQTT is a more lightweight approach to publish/subscribe better suited to the tactical domain: MQTT outperformed WS-N by consuming less bandwidth and producing lower delays in message delivery (Bloebaum and Johnsen, 2015) (Manso et al., 2018). Furthermore, it was demonstrated in Manso, Johnsen, Lund and Chan (2018) MQTT's flexibility to cope with various message payloads, message's size and number of subscribers.

Given the promising results obtained with MQTT, the group continued to analyse the application of MQTT technologies in tactical environments, including the feasibility to deploy a coalition setting (i.e., federated-based setup), supporting information exchange between brokers, while preserving the Nations' ownership (and control) over its resources. This paper describes the selected approach and obtains experimentation results. Next, the MQTT is introduced, together with the approach used to build a federated multi-broker deployment.

C.2.1 MQTT: Publish-Subscribe Event-Driven Message Exchange

Event-driven message exchange, or publish/subscribe as it is often called, is a message exchange pattern in which entities that have information they want to share (i.e., *producers* or *publishers*) can publish this information. Information consumers (i.e., *consumers* or *subscribers*) can subscribe to specific types of information they want to receive. When information is published that matches the subscribed interests, it is sent to the subscriber(s). The distribution of information is often performed by a *message-broker*.

MQTT is an ISO standard (ISO/IEC PRF 20922) which is built on the TCP/IP protocol. It is designed for connections with remote locations where a small footprint is required (both related to code and network). Although several independent implementations of standard-compliant brokers and clients exist[5], being a standard, clients and brokers from different vendors interoperate seamlessly and promote interoperability.

Within an operational environment, there may be one or more MQTT brokers available and it is possible to define rules for message exchange between them. Such is called a multi-broker deployment[6]. This capability

[2] The NNEC Feasibility Study recommends that all heterogeneous networks forming the Networking Infrastructure (NI) should be able to transfer IP based traffic (NATO NC3A, 2005).

[3] OASIS. OASIS Web Services Notification (WSN) TC. Available at: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn.

[4] NATO STANAG 5527: NATO Friendly Force Information Standard for Interoperability of Force Tracking Systems.

[5] See https://github.com/mqtt/mqtt.github.io/wiki/libraries for an overview.

[6] For further details on the need for such multi-broker deployments, see our discussion in (Manso et al., 2018).

becomes crucial when considering a coalition network, where each participating nation may manage one (or more) brokers but wishes to build a shared information environment using their broker infrastructure and without impacting producers and subscribers.

In this paper, we consider the MQTT v3.1.1 standard[7], which is mature and well supported these days. The MQTT standard defines the API that clients should use to interact with the MQTT broker (e.g., set up a subscription, publish messages). The standard does not describe how to build multi-broker setups or robust MQTT broker clusters. Some broker implementations support a proprietary approach to broker clustering (e.g., the VerneMQ[8] broker used by NOR for these experiments – see Section C.6), whereas others do not (e.g., Eclipse Mosquitto[9], used by PRT and USA for these experiments – see Section C.6). However, there is an approach to achieving multi-broker setups that uses a standard API to build a so-called *MQTT-bridge*, as explained next.

C.2.2 A Federated MQTT Multi-Broker Approach Supporting a Coalition Environment

The MQTT-bridge principle is to interconnect the MQTT broker it is associated to with another MQTT-broker. Therefore, by defining a main MQTT broker in a coalition environment (eventually having redundant brokers to avoid a single point of failure) and configuring the MQTT-bridge belonging to each remaining MQTT-broker a coalition MQTT environment can be obtained. In addition to interconnecting brokers, the MQTT bridges also avoid message loops by adhering to specific topic exchange configurations.

This principle is illustrated in Figure C-1. Three MQTT brokers are deployed, each serving a given Nation having its own publishers, subscribers and topics. When entering a coalition environment, Nations agree to use Nation A as "main broker". Nations B and C configure a MQTT-bridge that connects their brokers to Nation A's broker. The MQTT-bridge defines which topics should be replicated and in which direction (i.e., in, out or both). In other words, Nations explicitly choose which topics (and information) is to be shared.

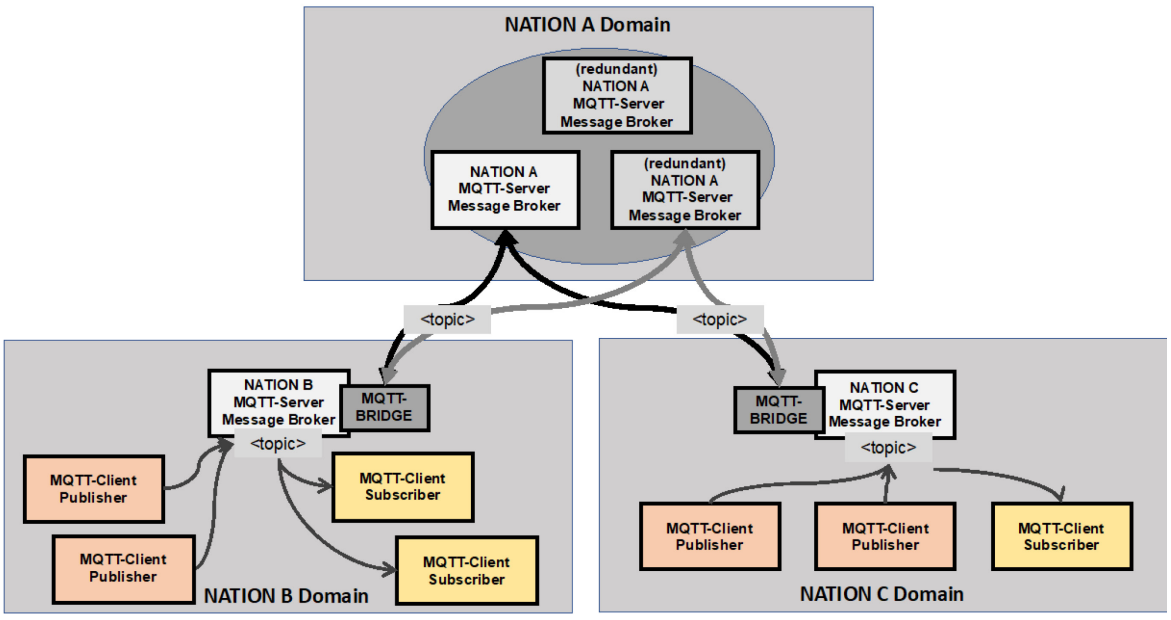


Figure C-1: MQTT Multi-Broker Deployment in a Coalition Environment.

[7] OASIS MQTT Version 3.1.1 Plus Errata 01. 10 December 2015. Available at: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.

[8] See https://vernemq.com.

[9] See https://mosquitto.org.

The MQTT-bridge principle will be used in the experiments described in this paper, to test the federation aspects of MQTT while still remaining standard compliant. It is acceptable in a coalition environment to use certain proprietary elements (e.g., internally, a Nation may use vendor specific features to achieve broker redundancy and automatic failover), but the interfaces used within the coalition should be standardized to promote interoperability and remain true to the FMN mindset.

C.2.3 Topic Definition in a Coalition Context

The publish/subscribe paradigm operates based on the definition of *topics*, which typically are string based keywords (i.e., UTF-8 strings) that are attached to the messages as metadata. Different publish/subscribe standards have different rules for how complex a topic can be.

MQTT does not have a formal way to describe its topic structure. It uses a simple, but highly expressive topics structure, where more advanced topics can be formed using a (hierarchical) multi-level structure, where each level is separated by a forward slash.

An important feature in MQTT is that topics can be formed dynamically and on-demand facilitating the process of their creation and operation. Furthermore, MQTT accepts the use of wildcards, allowing subscribing to multiple topics of interest, instead of requiring individual topic subscriptions.

As with most publish/subscribe systems, there are certain limitations to the MQTT topic handling that should be taken into account when deciding which topic structure to utilize:

1) There is no concept of topic namespaces in MQTT, which means that if multiple communities want to be able to use the same MQTT broker, the communities need to de-conflict their use of at least the root topics so that the same community will not use the same topic strings. In a multi-broker topology, the same is true across the entire federation of brokers (unless one or more of the brokers do topic remapping).
2) MQTT messages consist of *one* topic string and the message payload. This means that the topic structure must be carefully considered. Either the single topic string needs to contain all the elements an information receiver might need to filter on, or the same message might have to be published more than once in order to fully capture all possible filter expressions. An example would be a location report for a Norwegian military aircraft flying over a given city. This information object could be of interest both to subscribers interested in air traffic in general, and to subscribers interested in the movements of all Norwegian military units within the boundaries of that city. MQTTs support for wild cards (i.e., '#' and '+') matching of topics makes it possible to support both of these interests, as along as both the fact that the track is for an aircraft and that which city the track is located in is a part of the topic structure.
3) MQTT does not support discovery of topics. This means that consumers and managers responsible for configuring MQTT bridges need to be made aware of which topics are available either known in advance or shared out of band. One possible solution in a multi-broker topology is to have all information be shared across the MQTT bridge. This solution can lead to a high number of messages being passed between brokers, and it is thus only viable either when network resources are plentiful, or when all information handled by the brokers is of common interest.

Taking the above limitations in MQTTs topic handling mechanism into account means that defining a consistent and known structure for topics, becomes a central element in enabling coalition partners to subscribe to topics of interest across national system boundaries.

In M. Manso, F. Johnsen, M. Brannsten (2017), we proposed a topic structure in the context of a deployed force by a single-nation that is herein adapted considering a coalition environment:

```
coalition-Id/country-Id/unit-Id/entity-Id/service-Type
```

Where:

- "coalition-Id" uniquely identifies the coalition.
- "country-Id" uniquely identifies the country that is part of "coalition-Id". For example, according to the NATO STANAG 1059, "NOR" is used for Norway.
- "unit-Id" is an arbitrary string that uniquely identifies the unit (or group of entities) that belongs to "country-Id".
- "entity-Id" is an arbitrary string that uniquely identifies an entity (e.g., a soldier or a vehicle) that belongs to "unit-Id".
- "service-Type" is a string that uniquely identifies the type of service provided by or associated with "entity-Id". For example, in this paper we use the "location" topic to publish information pertaining to the unit's location. Other topic names representing services associated with a unit could be "health_status", "ISR_report" and "chat".

In this paper, we use the "location" service, representing a simple version of the "Blue Force Tracking" (BFT) service, to evaluate the multi-broker MQTT performance. This service is appropriate for this purpose because it generates the necessary amount of network traffic by periodically sending messages from each unit.

In addition to defining the topic structure, the exchanged messages' structure also needs to be defined and agreed by coalition partners, ensuring that publishers know what should be published and that subscribers are able to "decode" and process them. Herein, we opt to continue with our approach in adopting web-friendly technologies and formats to continue with the use of the general-purpose standard for location information GeoJSON[10]. As we already demonstrated in Manso, Johnsen, Lund and Chan (2018), GeoJSON can be used to share location information related with each unit. We extended GeoJSON to support domain-specific information, such as "country-Id" and "entity-Id", a presented in Figure C-4.

It is outside the scope of this paper to propose a complete topic structure and taxonomy in the context of military and coalition operations. However, this is a necessary step to undertake in future work for the successful adoption of publish/subscribe event-driven message exchange approaches in a coalition environment.

C.3 EXPERIMENTS

C.3.1 Purpose

This section describes the experiment conducted to evaluate the MQTT multi-broker deployment in the context of a coalition environment. A simulation environment was created that generates location messages over time pertaining to the coalition.

The main purpose of this experiment is to demonstrate the message exchange capability between brokers, enabling cross-nation exchange, where each nation effectively maintains ownership over its broker. More specifically, the experiment aims at demonstrating the ability to exchange BFT information among the force, where each entity produces a GeoJSON message periodically to its MQTT-broker and, via the MQTT-bridge

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

feature, messages are propagated across the whole coalition. The evaluation of the MQTT multi-broker deployment will be based on its reliability (percentage of messages received vs. messages lost) and performance (message latency between producer and subscriber).

Furthermore, with help of the experiments, we will demonstrate the ability to interoperate between different broker implementations via the use of the MQTT bridge standard and analyse the performance of this approach.

C.3.2 Scenario

The scenario chosen is a three nation coalition – specifically NOR (Norway), Portugal (PRT) and the United States of America (USA) – each bringing one unit constituted by eight soldiers. The coalition hierarchy, properly named as IST150, is depicted in Figure C-2.

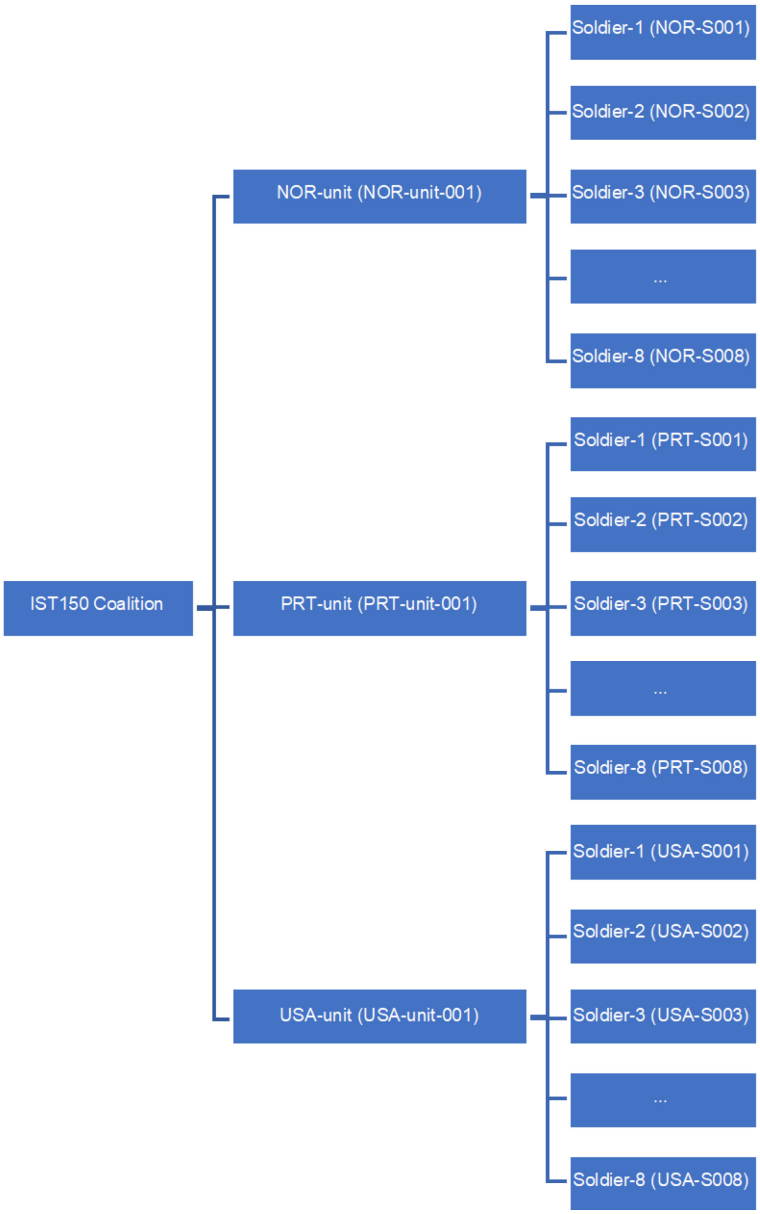


Figure C-2: Three Nation Coalition Used for Experiments.

In this multi-level hierarchy, each node represents a *topic* that has a unique identifier per level, allowing to clearly differentiate and discriminate which topics to publish or subscribe.

For example, location messages related with Soldier 2 belonging to PRT Unit 1 will be published to the following topic:

```
IST150/PRT/PRT-unit-001/PRT-S002/location
```

From a subscriber perspective, individual locations can be obtained by subscribing to the above topic.

Alternatively, if a subscriber intends to receive location pertaining to the whole PRT unit 1, wildcards can be used as follows:

```
IST150/PRT/PRT-unit-001/+/location
```

Or, for the whole coalition, can subscribe to the below:

```
IST150/+/+/+/location
```

For purposes of generating location information, GPS Exchange format (GPX) files were created, one for each soldier. The GPX files contain a sequence of location points (with latitude and longitude information) close to the city of Lisbon, Portugal. The location points do not represent any military exercise and their sole purpose is to generate messages for analysis and allow its presentation on a map by means of a visualisation application (see Figure C-3).

Figure C-3: IST150 Coalition in Action.

Figure C-3 illustrates two moments of the experiment: left (initial stage) and right (intermediate stage) .Each simulated entity (i.e., soldier) periodically generates a GeoJSON message, as introduced in Section C.2.3. Figure C-4 shows an example of a GeoJSON message is presented for soldier PRT-S001. The topic is also presented on top of the table. Note the extensions to the GeoJSON standard under “properties”, which allows adding explicit information concerning the entity and the message.

Topic	IST150/PRT/PRT-UNIT001/PRT-S001/location

GeoJSON Message	<pre>{
	"type": "Feature",
	"geometry":
	{
	"type": "Point",
	"coordinates": [38.747092, -9.156584, 0]
	},
	"properties":
	{
	"country-Id": "PRT",
	"unit-Id": "PRT-UNIT001",
	"entity-Id": "PRT-S001",
	"msg_id": "PRT-S001_676",
	"timestamp": 1560011584283
	}
	}</pre>

Figure C-4: Example of a MQTT Topic and Published GeoJSON Message.

C.3.3 Setup

The experiment is performed using a simulation environment created for this purpose. We define our coalition to be constituted by three nations (NOR, PRT and USA), each managing their own MQTT broker. In order to demonstrate the MQTT-bridge interoperability capabilities, different vendors are selected.

The simulation environment consists of the following:

- **One MQTT-broker managed by NOR**. The VerneMQ broker is be used.
- **One MQTT-broker managed by PRT**. This broker has a MQTT-bridge that is used to connect to the NOR MQTT-Broker. The Mosquitto MQTT broker is used.
- **One MQTT-broker managed by USA**. This broker has a MQTT-bridge that is used to connect to the NOR MQTT-Broker. The Mosquitto MQTT broker is used.
- **MQTT-bridges** are configured to publish/subscribe topics of interest to/from the NOR MQTT-broker (effectively replicating topics and messages across MQTT-brokers).
- A publisher node that can be instantiated to simulate a specific entity (i.e., soldier). The publisher node reads location information from a specific GPX file and publishes location messages at a pre-defined frequency. For this experiment, **24 publisher nodes are instantiated** (3 nations x 1 unit x 8 soldiers).
- **One subscriber node** that receives location information related to the whole coalition (i.e., all 24 entities).

The experiment setup is depicted in Figure C-5. The figure shows the three MQTT-brokers, where NOR operates as a main node to where the PRT and USA MQTT-brokers connect to via their bridges. Furthermore, it can be seen that the MQTT-client entities are connected to their nation respective MQTT-broker. Besides having each nation managing its own MQTT-broker, an entity (subscriber) from PRT connects to the MQTT-broker from PRT. This preserves each nation full control over its domain, while allowing information exchange among them.

The MQTT-brokers were deployed in cloud-hosted computers accessible via the Internet. Therefore, at this stage of the experiment, a stable network environment could be expected with enough bandwidth to cope with the generated network traffic.

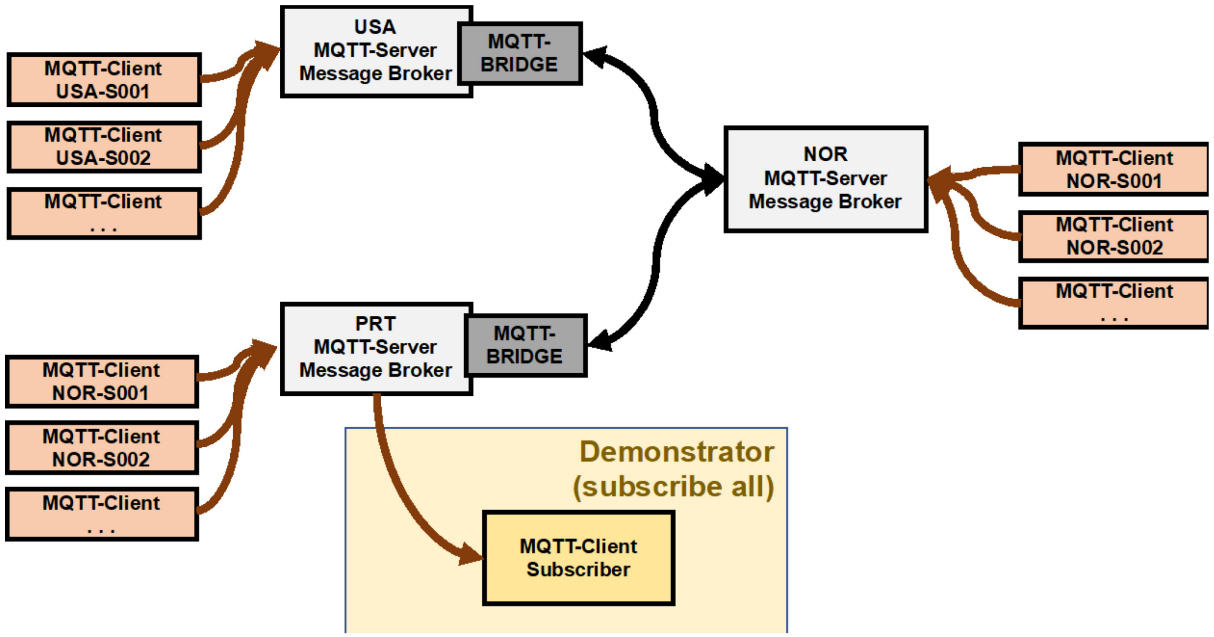


Figure C-5: Multi-Broker Deployment in Experiment.

To prevent time synchronisation issues between nodes, all MQTT-clients (24 publishers and 1 subscriber) were deployed in the same machine.

Concerning the publishers, they only publish messages to the MQTT-broker managed by their respective nation. For example, PRT entities only publish messages to the PRT MQTT-broker.

Concerning the subscriber, it belongs to PRT and should receive location information pertaining to the whole coalition. Furthermore, it only makes subscriptions to the PRT MQTT-broker. The MQTT-bridge functionality allows topics of interest (and messages) from other brokers (i.e., NOR MQTT-broker and USA MQTT-broker) to be "replicated" in the PRT MQTT-broker. Subsequently, PRT subscribers can subscribe to topics of interest using the PRT MQTT-broker.

In this experiment, location messages were generated (pertaining to the 24 entities). Each entity generated a location message each two seconds. Since location messages are produced periodically, the associated MQTT parameter *QoS* was set to 0 (i.e., fire-and-forget, the less reliable but more efficient and thus the most suitable in situations where one can afford to lose some messages).

For purposes of analysis, the following was logged:

- **Message ID**, allowing to track published and received messages (used for purposes of determining the reliability of the system).
- **Timestamp** (in ms) associated with the time when a message is published and when a message is received (used to determine message latency between publisher and subscriber).

The results of the experiment are presented next.

C.3.4 Results and Evaluation

In this section, the results of the performed experiment are presented. The following metrics are used for its assessment:

- **System reliability**: measured based on the percentage of messages lost (i.e., messages published but not received by the subscriber)
- **System performance**: measured based on the delay in delivering messages (i.e., difference between the time when a message is received and the time when a message is published).

C.3.4.1 System Reliability

The 24 publishers produced a total of 21704 location messages. The subscriber received all 21704 location messages. As presented in Table C-1, the percentage of messages lost was 0% and the system reliability was therefore 100%.

Table C-1: System Reliability.

System Reliability (based on messages send and received)	TOTAL
Messages Sent	21 704
Messages Received	21 704
Messages Lost	0 (0.0%)

The MQTT-broker and the use of MQTT-bridges deliver reliable outcomes, especially when considering we used the QoS parameter set to 0 ("fire-and-forget"). We also benefitted from having a stable network connection (i.e., Internet) to conduct the experiments.

C.3.4.2 System Performance

We measured the delay between the instant in time a message is published and the instant in time a message was received by the subscriber.

The overall results are presented in Figure C-6 and Table C-2.

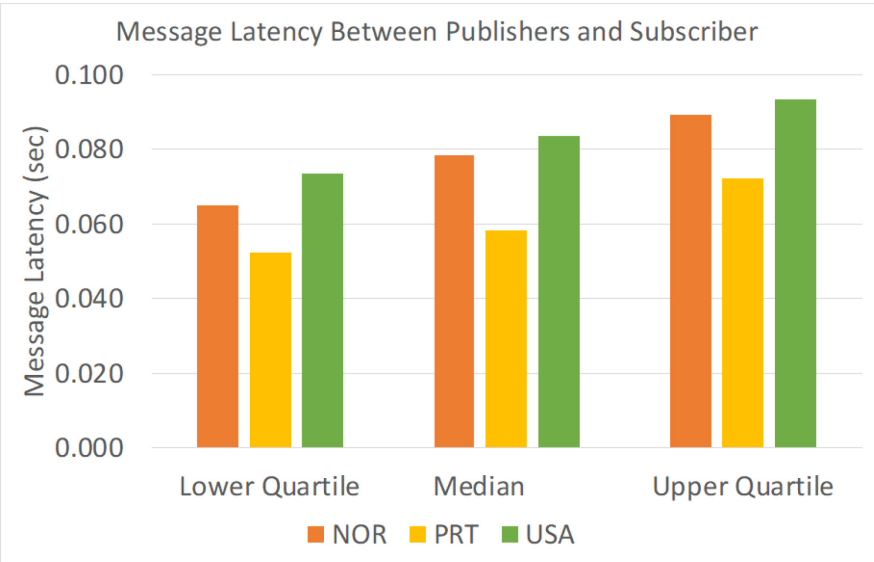


Figure C-6: System Performance: Message Latency.

Table C-2: System Performance: Message Latency Detailed Measurements (in seconds).

Results (Average)	NOR	PRT	USA
Lower Quartile	0.065	0.053	0.074
Median	0.079	0.059	0.084
Upper Quartile	0.089	0.072	0.093

On average, messages take a few milliseconds (about 70 ms) between being published and being received. The lower and upper quartile also have a small deviation from the median value (about 10 ms), which indicates performance is, in overall, good and with small deviations.

As expected, the messages related to PRT entities exhibit the lowest latency (59 ms median) since they are locally distributed by the PRT MQTT-broker without undergoing through the MQTT-bridge. On the other hand, NOR and USA messages are conveyed via the MQTT-bridge to the PRT MQTT broker thus exhibiting an additional delay (about 25 ms). There is also a small overhead (about 5ms) in the USA messages, that might be a result of the delay from the USA MQTT-bridge to the NOR MQTT-broker.

Figure C-7 provides a detailed view of the measured message latency for all entities. This allows to visualize a few deviations from the statistical results presented before. It is worth to mention that albeit the MQTT delivers a good overall performance, it is observed that a few messages take more than 0.5 seconds to be received. This might be a result of temporary loss of connectivity or network congestion related to the Internet connection (in other words, non-deterministic conditions). Despite representing a small number of messages and effectively being outliers, it is observed that MQTT still was capable to deliver all (100%) messages, thus, overcoming these disturbances.

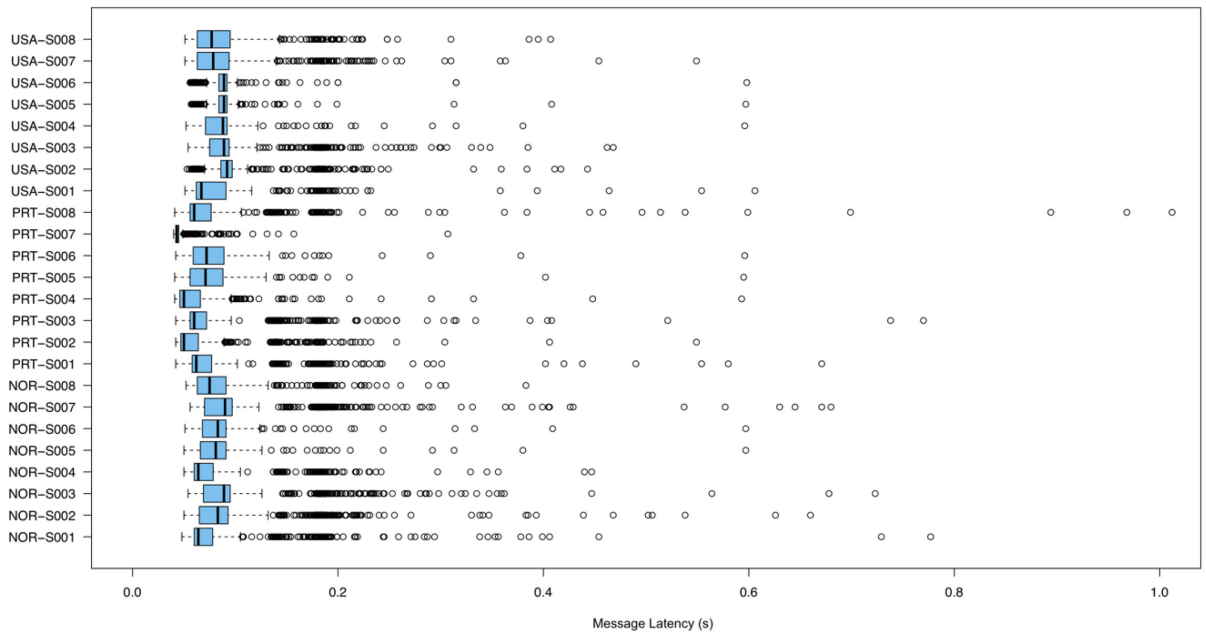


Figure C-7: Overall Message Latency Measured in the Subscriber.

C.4 CONCLUSION

Pursuing the objective to achieve technical interoperability in a coalition environment, covering as well tactical forces operating in DIL network environments, NATO has supported a number of initiatives and research groups, including the IST-150 that is analysing promising standards and emerging technologies. In this regard, the group identified the MQTT protocol – an open standard, lightweight, loosely-coupled and widely used – as a good candidate for tactical environments.

In this paper and as part of the IST-150 activities, we continued our analysis of MQTT as an enabling platform for information exchange based on the publish-subscribe paradigm (thus event-driven). Specifically, we considered a coalition deployment and described a possible federated-based setup using multiple MQTT brokers and MQTT-bridges as a way to exchange information between brokers, while preserving the Nations' ownership over its resources.

The results of our experiments show that MQTT delivered good results, with 100% success message delivery and most messages being delivered in less than 100ms. Moreover, the used of MQTT-bridges yield small overheads (order of a few ms).

The obtained results show a promising use of the MQTT multi-broker functions (based on the MQTT-bridge functionality) using a stable and fast (broadband) network environment. However, a tactical environment is characterized by disconnected intermittent connectivity and limited bandwidth (DIL), which challenge most Internet-based technologies, including MQTT.

In future work, we plan to specifically address DIL network environments and the introduction of realistic tactical radio models in a simulated environment. This will allow to further evaluate and fine-tune the application of MQTT technologies (including MQTT-SN) in an environment that is closer to a real deployment.

Moreover, this work introduced an approach to define topics in the context of a coalition environment, also allowing to take advantage of MQTT wildcard features. For topic-based approaches to work, rules and structures should be defined and agreed. Future work should also address this area, ideally involving a large number of coalition partners, eventually resulting in a future standard to be adopted.

C.5 REFERENCES

Bloebaum, T., and F. Johnsen. Evaluating Publish/Subscribe Approaches for Use in Tactical Broadband Networks. IEEE MILCOM 2015, Oct 26 – 28, Tampa, Florida, 2015. DOI: 10.1109/MILCOM.2015. 7357510.

Manso M., J. Alcaraz Calero, C. Barz, T. Bloebaum, K. Chan, N. Jansen, F. Johnsen, G. Markarian, P. Meiler, I. Owens, J. Sliwa, Q. Wang. SOA and Wireless Mobile Networks in the Tactical Domain: Results from Experiments. MILCOM Oct 26 – 28, Tampa, Florida, 2015.

Manso M., F. Johnsen, M. Brannsten, A Smart Devices Concept for Future Soldier Systems. ICCRTS 2017, Los Angeles, USA, Nov 6 – 8, 2017.

Manso, M., N. Jansen, T. Bloebaum, F. Johnsen, K. Chan and A. Toth. Mobile Tactical Force Situational Awareness: Evaluation of Message Broker Middleware for Information Exchange. 23rd ICCRTS: Multi-Domain C2, Pensacola, Florida, Nov 6 – 9, 2018.

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

Meiler, P., T. Bloebaum, F. Johnsen, N. Jansen, I. Owens. IST-118 SOA Recommendations for Disadvantaged Grids: Tactical SOA Profile, Metrics and the Demonstrator Development Spiral. Paper presented at the SCI-254 Symposium on Architecture Assessment for NEC. STO-MP-SCI-254, 2013.

NATO. Interoperability: Connecting NATO Forces. Updated: Jun 06, 2017. Available at: https://www.nato.int/cps/en/natohq/topics_84112.htm.

NATO. Federated Mission Networking. Feb 26 2015. Available at: https://www.act.nato.int/fmn.

NATO IST-090. SOA Challenges for Real-Time and Disadvantaged Grids. AC/323(IST-090)TP/520. NATO, 2014.

NATO IST-150. NATO Core Services Profiling for Hybrid Tactical Networks. Available at: https://www.sto.nato.int/Lists/test1/activitydetails.aspx?ID=16530.

NATO C3 Board. Core Enterprise Services Standards Recommendations: The SOA baseline profile v.1.7. Enclosure 1 to AC/322-N(2011)0205. NATO Unclassified releasable to EAPC/PFP, 11 Nov 2011.

NATO NC3A. NATO Network Enabled Capability Feasibility Study Executive Summary. Version 2.0. Oct 2005.

Annex D – EVALUATING PUBLISH/SUBSCRIBE STANDARDS FOR SITUATIONAL AWARENESS USING REALISTIC RADIO MODELS AND EMULATED TESTBED

24th ICCRTS: "MANAGING CYBER RISK TO MISSION"

Paper ID: 15

Topic 9: Experimentation, Analysis, Assessment and Metrics

Authors:

Frank T. Johnsen and Trude H. Bloebaum
Norwegian Defence Research Establishment (FFI)
Norway

Norman Jansen
Fraunhofer FKIE
Germany

Gerome Bovet
Armasuisse
Switzerland

Marco Manso
PARTICLE, Lda.
Portugal

Andrew Toth and Kevin S. Chan
CCDC Army Research Lab (ARL)
USA

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

There is currently an ongoing initiative to improve the interoperability between nations and other partners during common missions through Federated Mission Networking (FMN). So far, the focus of the standardization and profiling work done in FMN has mostly been on static and deployed networks, where networking resources are stable and plentiful. There is however also a need for interoperability at the tactical edge, between mobile units that have limited and often disrupted communications. In a previous study, we compared different protocols for subscription based distribution of information. We concluded that the WS-Notification standard, which is currently used in NATO, has a too large overhead in lower capacity tactical networks, and that for instance the Message Queuing Telemetry Transport (MQTT) protocol could be used instead.

In this paper, we expand upon those findings by investigating the applicability of MQTT in tactical networks further. Here, we address one of the main shortcomings in the testbed used in our previous experiments by adding in new and more realistic radio models, which allow us to better assess the performance of MQTT in the tactical domain. Furthermore, we also expand our experiments evaluating MQTT for Sensor Networks (MQTT-SN) as well. The reason for adding MQTT-SN to the experiments is that this protocol is based on UDP rather than TCP.

This work has been performed in the context of the NATO STO/IST-150 "NATO Core Services profiling for Hybrid Tactical Networks" working group.

D.1 INTRODUCTION

There is currently an ongoing initiative to improve the interoperability between nations and other partners during common missions through Federated Mission Networking (FMN). The goal of this initiative is to enable so-called zero-day interoperability by establishing an increasingly mature framework for mission interoperability ahead of time. This framework includes all aspects of establishing a mission network, such as governance, procedures and also standardized technical services.

So far, the focus of the standardization and profiling work done in FMN has mostly been on static and deployed networks, where networking resources are stable and plentiful. Current directions of military operations are trending towards pushing decision making and collaboration at the tactical edge. Operations at the tactical edge are significantly different from the enterprise networked environment. The networks that support these tactical edge operations are often characterized as a disconnected intermittent connectivity and limited bandwidth (DIL) environment, or more recently a congested, contested operational environment. Current approaches within FMN are relevant for, and validated on, enterprise (perhaps wired) networks, but may not be applicable in environments with the aforementioned challenges present in the tactical domain.

Despite these challenges in the networking environment, operations must occur at much faster timescales and deal with increased uncertainty of information and operations, and as a collaborative effort between partners. There is thus a need for interoperability at the tactical edge, between mobile units that have limited and often disrupted communications. When there is a need of many-to-many information exchange based on the relevance of, or interest for, a given type of information, the subscription-based information exchange is a pattern that is well known also in these types of environments. In our previous study [1], we compared different protocols for subscription based distribution of information between a number of nodes. We concluded that the WS-Notification (WS-N) standard [2], which is currently used in NATO, has a too large overhead in lower capacity tactical networks, and that for instance the Message Queuing Telemetry Transport (MQTT) [3] protocol could be used instead.

In this paper, we expand upon those findings by investigating the applicability of MQTT in tactical networks further. Here, we address one of the main shortcomings in the testbed used in our previous experiments by adding in new and more realistic radio models, which allow us to better assess the performance of MQTT in the tactical domain. Furthermore, we also expand our experiments evaluating MQTT for Sensor Networks (MQTT-SN) as well. The reason for adding MQTT-SN to the experiments is that this protocol is based on UDP rather than TCP.

One can expect a variety of tactical services relevant to operations in this environment. For example, position location information is usually invoked as the primary shared situation awareness requirement in most operations. In this paper, we have considered Blue Force Tracking (BFT) as a representative service. We do note that other services such as sharing of video or imagery may demand more resources than typically available for these networks. One standing challenge is the optimization of multiple networked services for resource-constrained networks in these operational environments.

This work has been performed in the context of the NATO STO/IST-150 “NATO Core Services profiling for Hybrid Tactical Networks” research task group.

D.2 TESTBED

Measuring the performance of a single BFT service in a lab environment will not indicate how multiple instances of the BFT service deployed together with tactical radio systems in military vehicles will perform in a realistic military scenario. This is the case, because typical lab experiments do not take the dynamic environment into account and are poorly scalable.

Instead, a whole combination of different systems (IT and communications systems) has to be taken into account. For the systems under test – i.e., BFT services – the original software (or virtualized versions) should be run in order to represent the real systems in as much detail as possible. Systems which cannot be virtualized, because the software is not publicly available (e.g., military radios), will be emulated by means of real-time radio simulators with realistic radio models (see Section D.3).

We use a subset of the *Anglova scenario* [4] for our experiments. Specifically, we model a mechanized battalion with 24 military vehicles coordinated by the Coalition HQ. The battalion nodes are equipped with tactical radios that are used to exchange information. To drive the network emulation, we employ the Extendable Mobile Ad hoc Network Emulator (EMANE) [5], which provides radio link emulation, signal propagation and mobility representation to the experiment. Advantages of this testbed approach are scalability and (to some degree) repeatability. Consider that the behaviour of the applications may not be completely deterministic since real software is running in real-time.

D.3 NEW RADIO MODELS

During the first experiments with EMANE leveraging the standard Wi-Fi models used by the community, we noticed that the obtained results were not matching the performance of real tactical radios [6]. The Optimized Link State Routing (OLSR) routing tables as well as some performance metrics, such as throughput and latency between emulated nodes led us to the two following conclusions:

1) The Wi-Fi models, although tuneable, do not allow reproducing the latencies and throughput of real tactical radios. The obtained performance during the emulations is far too optimistic compared to the expected performance in a real deployment.
2) The Anglova Vignette 2 with Company 1 scenario (24 nodes) is not challenging enough, as most of the time the topology tends to be a full-mesh, whereas multi-hop topologies would rather be more realistic.

The combination of these two drawbacks leads to the situation where experiments do not reflect reality, as even heavy protocols, which were not working under lab conditions with real radios, show high performance in the emulated environment. In order to obtain more realistic emulations, we started by reproducing Narrowband and Wideband tactical radios in EMANE. Their performance (throughput and latency) was measured under lab conditions with various Received Signal Strength Indicators (RSSIs). In a second step, and with the information in our possession regarding the Time-Division Multiple Access (TDMA) schedules, we elaborated TDMA scheduling models in EMANE. As shown in Ref. [6], we were able to reproduce in quite high fidelity the performance of the real radios, including the adaptive rate changing the performance according to the channel quality.

As previously mentioned, the 24 nodes we used from the Anglova scenario do not produce a challenging network topology. This is due to the rather short distances between the nodes throughout the scenario. The emulated vehicles move in the form of clusters, which leads to the situation where full connectivity is achieved with only one-hop during most of the emulation. Such conditions are not challenging in terms of multi-hop topologies where performance is relative to the number of hops. We therefore adapted the Anglova scenario in order to generate more hops between the nodes [7]. This was achieved by decreasing the emulated output power to 5W (37dBm), which is often a tactical choice allowing lowering the possibility getting spotted by an enemy. Additionally, the locations of selected nodes were changed, so that during certain phases of the scenario, the topology also contains some chains. The average number of hops increased from 1.5 to around 2.5, whereas the maximum number of hops increased from 4 to 7. In this paper, we refer to this version as “Modified Anglova” and the original as “Anglova scenario”. We perform experiments with both versions of the scenario using the Wideband TDMA scheme developed by Switzerland.

D.4 TEST APPLICATIONS AND SOFTWARE

In our experiments we use the NATO Friendly Force Information (NFFI) data format in our BFT services. The reason for choosing the NFFI data format (described in draft STANAG 5527) is that it has been used with great success in many contexts, after it originally emerged to support interoperable friendly force tracking in the Afghan Mission Network. We consider it a good example of a representative standard payload for our experiment. The dissemination mechanisms we use are WS-N, MQTT, and MQTT-SN, respectively. Each of these three standards provide the functionality necessary to distribute information from a provider to the interested consumers. It should be noted that WS-N consists of three standards: WS-BaseNotification, WS-BrokeredNotification and WS-Topics. For the work in this report, we use WS-Notification including the broker functionality described by WS-BrokeredNotification [2]. The BFT services were implemented by the Norwegian Defence Research Establishment (FFI).

WS-N is a part of the family of SOAP Web services standards. SOAP services promote interoperability, but the cost is increased overhead. Hence, it is not necessarily well suited for use in tactical networks where network capacity typically is low. As a consequence, we investigate two other publish/subscribe industry standards that can possibly provide the same functionality as WS-N, but with less overhead. Previously, we have compared WS-N with MQTT, and found MQTT to be more efficient [1]. In this paper, we continue our experiments using the above mentioned radio models, as well as adding on the UDP-based counterpart to MQTT, namely MQTT-SN.

We used a closed-source implementation of WS-N developed in-house at FFI. However, this implementation has been tested for interoperability at the NATO Coalition Warrior Interoperability eXercise (CWIX) in 2014, where it was shown that the functions used (subscribing to a topic, publishing to a topic, and notifying the subscribers of new data) in our experiments were indeed compliant with the standard [8].

For MQTT we used the open source VerneMQ broker which is freely available [9]. MQTT-SN is usually not supported natively by existing brokers, so we added MQTT-SN support to VerneMQ by installing the free, open source gateway solution from the Eclipse Paho project [10]. It should be noted that since MQTT-SN has to be offered via a gateway, this may negatively impact the performance of the protocol as opposed to if it were offered as a complete stand-alone solution.

D.5 EXPERIMENT EXECUTION

Overview of experiments:

Experiment Series/Protocol	WS-Notification	MQTT	MQTT-SN
Anglova scenario, Swiss TDMA	Wideband radio	Wideband radio	Wideband radio
Modified Anglova, Swiss TDMA	Wideband radio	Wideband radio	Wideband radio

The experimental testbed used to conduct experiments is the Network Science Research Laboratory (NSRL) [11] established by the CCDC Army Research Laboratory (ARL). The NSRL provides network emulation capabilities and military relevant data and scenarios for the testing and evaluation of various networking oriented technologies and approaches. The facility has enabled collaboration between ARL researchers and those from other organizations. Additionally, infrastructure in the way of dynamic virtualization has been developed to assist in the execution of experiments in the NSRL. To enable repeatability and scalability of experimentation, ARL has also developed a platform called Dynamically Allocated Virtual Clustering Management System (DAVC). DAVC provides the capability to dynamically create and deploy virtual clusters of heterogeneous nodes as specified by Virtual Machines (VMs).

Experiments are completely reconfigurable through the DAVC interface, with minor modifications to parameters defined in custom scripts (e.g., nodes' location and radio signal path loss between nodes, as provided by Anglova).

Both the Anglova scenario and DAVC are releasable through NATO collaboration.

The Anglova scenario, incorporating WS-N, MQTT, and MQTT-SN broker messaging services, was setup in the NSRL environment. For that, WS-N and MQTT services were installed onto the VM template of the Anglova scenario to enable the publish/subscribe position location information services. The experiments use a **single broker topology**. The VM template is deployed to nodes during runtime of the scenario. This is illustrated in Figure D-1.

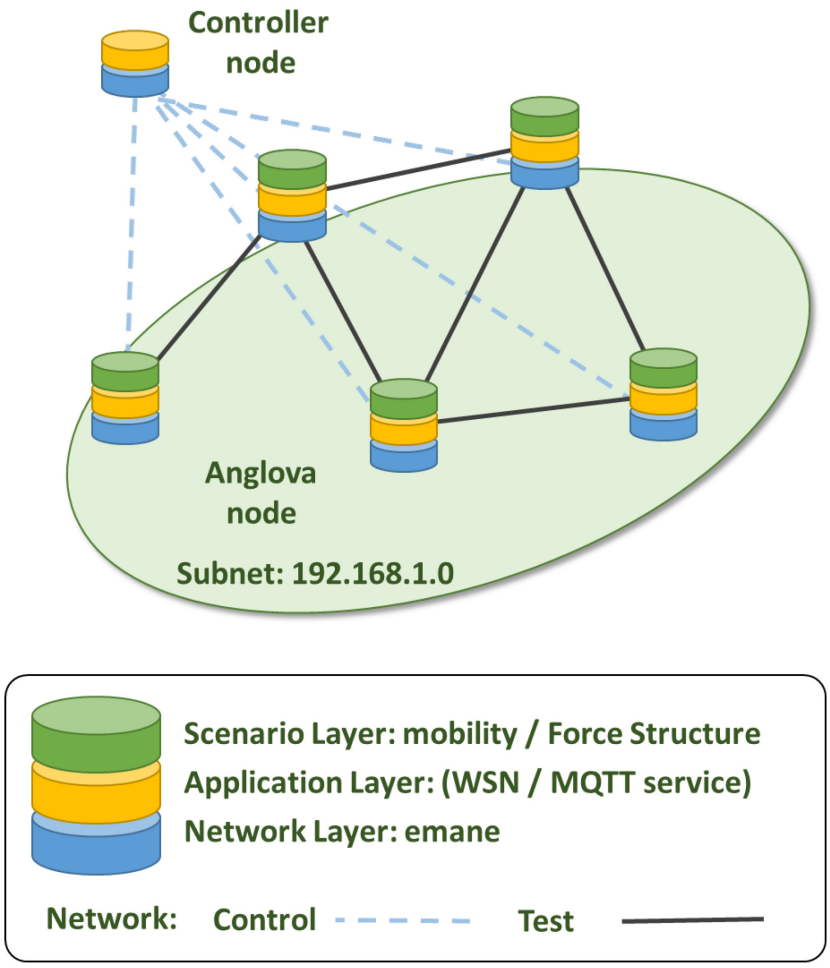


Figure D-1: Architecture of Network Experiment Including Network Emulation, Application and Scenario Layers.

For network emulation, we use the EMANE that provides – besides the emulation of the radio links – signal propagation and mobility representation to the experiment to create a more realistic environment. The mobility information was drawn from Anglova recorded data.

The emulation allows for various types of routing and radio models to be used; in this scenario we use Optimized Link State Routing (OLSR) [12] (OLSR 2016) V2 via the OLSR Daemon (OLSRD) on each virtual machine representing a node in the scenario with wireless links based on the Swiss TDMA Wideband

model. The TDMA model was configured to emulate wideband tactical radios operating at 300 MHz with a 250 KHz bandwidth and 1 Mbit/s data rate. The TDMA model was configured with 8 frames with 24 slots, a slot overhead of 3 us, and slot duration of 5000 us. OLSR V2 was configured with a Hello Interval of 2 seconds, Hello Validity Time of 20 seconds, Topology Control Interval of 8 seconds, and Topology Control Validity time of 80 seconds.

In the initial set of experiments, we ran 20 minutes of the Anglova scenario vignette excerpt consisting of 24 nodes. The publishers on nodes 2 through 24, which sent node locations (i.e., NFFI messages) every 10 seconds, were started at the 1-minute mark, and stopped after 20 minutes at minute 21 in the scenario. We set up a DAVC cluster of 24 "Anglova" nodes and one controller node. The controller node is used as the orchestration node and is not represented in the experiment nor does it take part in the scenario. Node 1 for this experiment is arbitrarily established as the broker node (i.e., runs the WS-N, VerneMQ Broker, or VerneMQ broker with the MQTT-SN Gateway). It also has a subscriber service running on it (i.e., subscribes to and receives messages from all publishers). We note that the platform allows for any configuration of broker and subscriber services.

Additionally, to facilitate the execution of these experiments, we have created services that launch EMANE and the Anglova configuration. We also have Linux shell scripts that can start and stop the publisher services for both WS-N and MQTT as well as gathering generated pcap and log files used for analysis.

D.6 ANALYSIS

The experiments described in this paper aim to test the performance of several different ways to distribute the information from BFT services (the system under test) in a realistic setup with emulated radio communications systems according to a realistic military scenario (Anglova). Two different scenario setups were used for the experiments. The first one uses the original Anglova scenario, but with the TDMA model described in Section D.3 "New radio models" above. The second one also uses this TDMA model and additionally all other adaptations described in Section D.3. These include decreasing the emulated output power to 5W and changing the positions of some of the units to generate more transmission hops. Thus, this second version of the scenario is even more challenging than the first one.

For the BFT service different protocol standards (WS-N, MQTT and MQTT-SN) have been evaluated.

For the analysis of the experiments, we used analysing tools from the *Analyze and Test environment* (*AuT*) project of Fraunhofer FKIE, Germany. In Ref. [13], concepts and tools for analysing complex military experiments in a virtualized testbed are described. These include a concept for capturing and processing monitoring data from C2IS applications used in distributed tactical networks, the specification of suitable metrics for military applications and the definition of different visualizations based on these metrics.

Our evaluation approach makes use of monitoring data from both the network layer as well as the application layer. For the network layer, the network traffic was logged via publicly available network logging tools (tcpdump). For the application layer, the application traffic was logged by the application service itself at different measuring points (e.g., after a message was received, after a message was processed by the application, etc.). This has been done via a logging interface which we defined by a JSON schema. For this purpose, we implemented the logging interface into the publisher and subscriber services. The JSON logs and tcpdumps are used to calculate packet and message losses.

D.6.1 WS-N with Anglova Scenario

In this setup, we deploy a closed-source WS-N broker together with one WS-N subscriber on Node 1. Nodes 2 to 24 (23 nodes in total) each run a WS-N producer software publishing a NFFI message every 10 seconds. The measurements pertaining to network and application layers are presented next.

D.6.1.1 Network Layer

By analysing the network level log files (packet captures) the data volume produced by the WS-N could be obtained (see Table D-1). This data volume contains all data from the different transmission layers (Ethernet, IP, TCP, HTTP). The WS-N-based communication produces 40 kbit of data per second. The message size of a WS-N message was 1863. The network logs show that there were 2468 TCP Duplicate Acknowledgments and that 2761 TCP retransmissions produced. 1761 of them were of the type "spurious"[1]. This problem often arises when using TCP in networks with a high bandwidth-delay product.

Table D-1: Results from Experiments for WS-N Anglova Scenario (Network Layer).

Data Volume (per second)	Message Size	TCP Duplicate ACK	TCP Spurious Retransmissions	TCP Retransmissions
40 kbit/s	1863 bytes	2468	1761	2761

D.6.1.2 Application Layer

The application logs consist of logging entries of the senders (publishers) of NFFI messages and logging entries of the receiver (subscriber) of these messages. This approach allows us to calculate the overall transmission times of NFFI messages, which represent the age of the positions as observed by the user at the receiver node. The results were analysed with help of analysing tools of the AuT project and are shown in Figure D-2. Figure D-2 shows as a boxplot diagram the transmission times of all publishers. Note that the nodes in the Anglova scenario are named by numbers from 100 to 1450, whereas the nodes in the modified Anglova scenario (cf. Section D.6.4 below) are renamed to 1, 2, ..., 24.

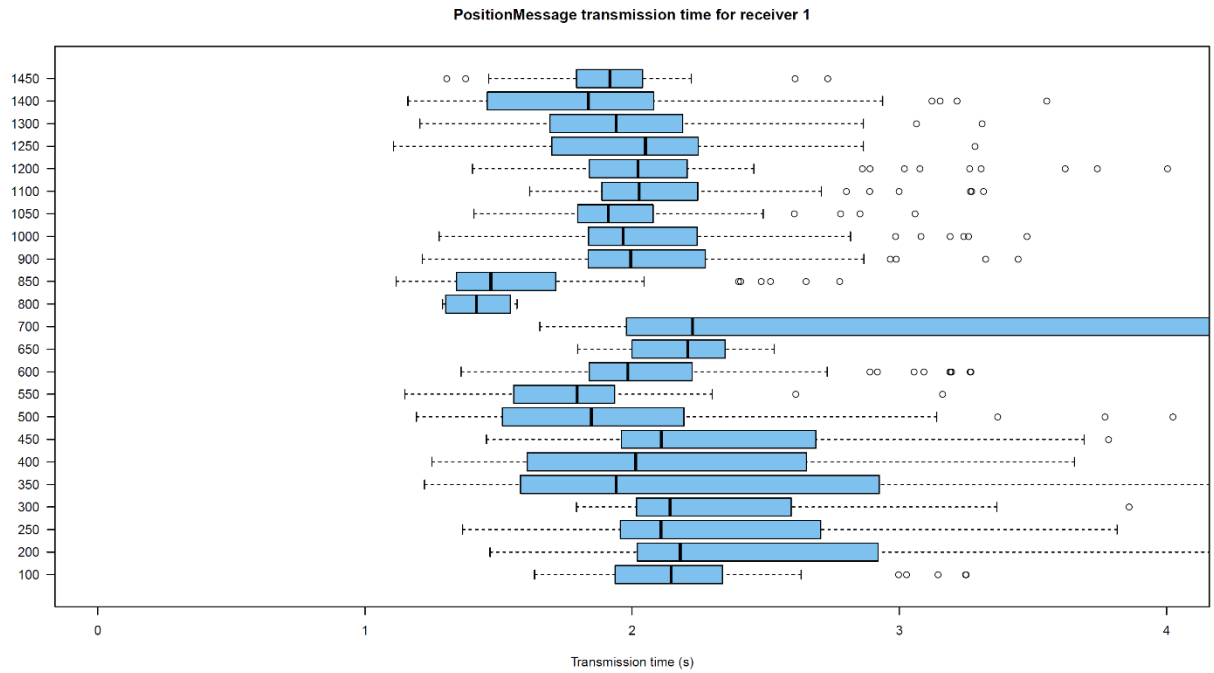


Figure D-2: Transmission Times of WS-N-Based NFFI Messages.

[1] Here, "spurious" means that a packet was unnecessarily retransmitted because the respective acknowledgement arrived too late at the sender. Since the congestion control mechanism of TCP interprets «lost» (actually belated in this case) acknowledgements as buffer overflows, the congestion window is unnecessarily decreased, which leads to a reduced throughput.

In total 1697 messages were published, of which 22 (1.30%) were lost (cf. Table D-2). The overall median of the transmission delay was 1.97 s (averaged over all messages from all publishers). The minimum delay was 1.11 s and the maximum delay was 86.78 s. As you can see in Figure D-2, most messages were near the median delay, but there are higher values for some publishers, who have suffered from a poor connection to the other nodes at some time in the scenario.

Table D-2: Results from Experiments for WS-N, Anglova Scenario (Application Layer).

Messages Sent	Messages Lost	Delay (Min)	Delay (Overall Median)	Delay (Max)
1697	22 (1.30%)	1.11 s	1.97 s	210 s

D.6.2 MQTT with Anglova Scenario

In this setup, we deploy the VerneMQ broker together with one MQTT subscriber on Node 1. Nodes 2 to 24 (23 nodes in total) each run an instance of the MQTT producer software publishing a NFFI message every 10 seconds. For the MQTT publisher the Quality of Service class *QoS0*[2] was used. The measurements pertaining to network and application layers are presented next.

D.6.2.1 Network Layer

The analysis of the network level log files (packet captures) results in the data shown in Table D-3. The table shows that the MQTT-based traffic produced 31 kbit/s of data volume. The size (content) of each message was 880 Bytes (WS-N's size increase was due to extra overhead from using SOAP and XML). The network logs show that there were 1922 TCP Duplicate Acknowledgements. Furthermore, 4281 TCP retransmissions were produced, 1436 of them were of type «spurious» similar to the setup with WS-N.

Table D-3: Results from Experiments for MQTT Anglova Scenario (Network Layer).

Data Volume per second	Message size	TCP Duplicate ACK	TCP Spurious Retransmissions	TCP Retransmissions
31 kbit/s	880 bytes	1922	1436	4281

D.6.2.2 Application Layer

In Figure D-3 the average transmission times of the messages are shown for each publisher in a boxplot diagram.

In total 2553 messages were published, from which 383 (15%) were lost (see Table D-4). The overall median of the transmission delay was 1.46 s (averaged over all messages from all publishers). The minimum delay was 0.60 s and the maximum delay was 225 s.

Table D-4: Results from Experiments for MQTT Anglova Scenario (Application Layer).

Messages Sent	Messages Lost	Delay (min)	Delay (overall median)	Delay (max)
2553	383 (15%)	0.60 s	1.46 s	225 s

[2] QoS0 gives *at most once* delivery semantics, whereas QoS1 gives *at least once* delivery semantics.

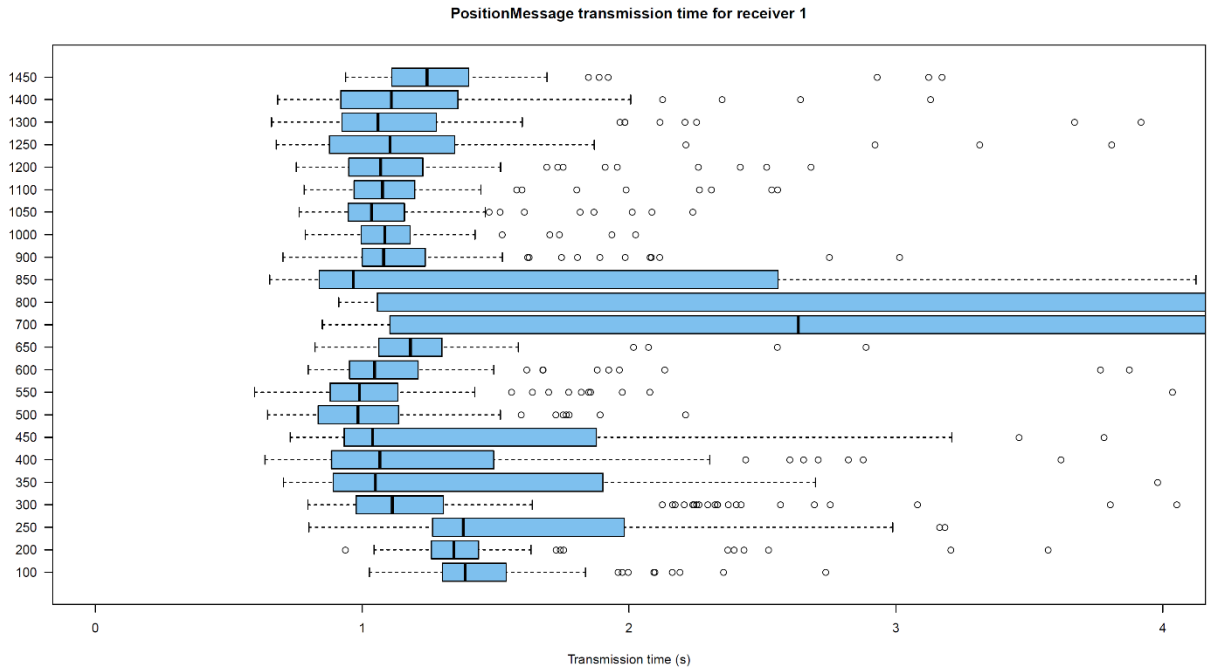


Figure D-3: Transmission Times of MQTT-Based NFFI Messages (Whole Diagram).

D.6.3 MQTT-SN with Anglova Scenario

In this setup, we deploy the VerneMQ broker in conjunction with the open source MQTT-SN gateway solution from the Eclipse Paho project. Furthermore, one MQTT-SN subscriber on Node 1 is deployed. Nodes 2 to 24 (23 nodes in total) each run an instance of the MQTT-SN producer software publishing a NFFI message every 10 seconds. For the MQTT publisher the Quality of Service class *QoS0* was used. The measurements pertaining to network and application layers are presented next.

D.6.3.1 Network Layer

The analysis of the network level log files (packet captures) results in the data shown in Table D-5. The MQTT-SN-based traffic produced 13 kbit/s of data volume. The size (content) of each message was 894 bytes and thus similar as the message size of MQTT. Since MQTT-SN uses UDP, there are no TCP retransmissions.

Table D-5: Results from Experiments for MQTT-SN Anglova Scenario (Network Layer).

Data Volume per second	Message Size	TCP Duplicate ACK	TCP Spurious Retransmissions	TCP Retransmissions
13 kbit/s	894 bytes	NA	NA	NA

D.6.3.2 Application Layer

In Figure D-4 the average transmission times of the messages are shown for each publisher in a boxplot diagram.

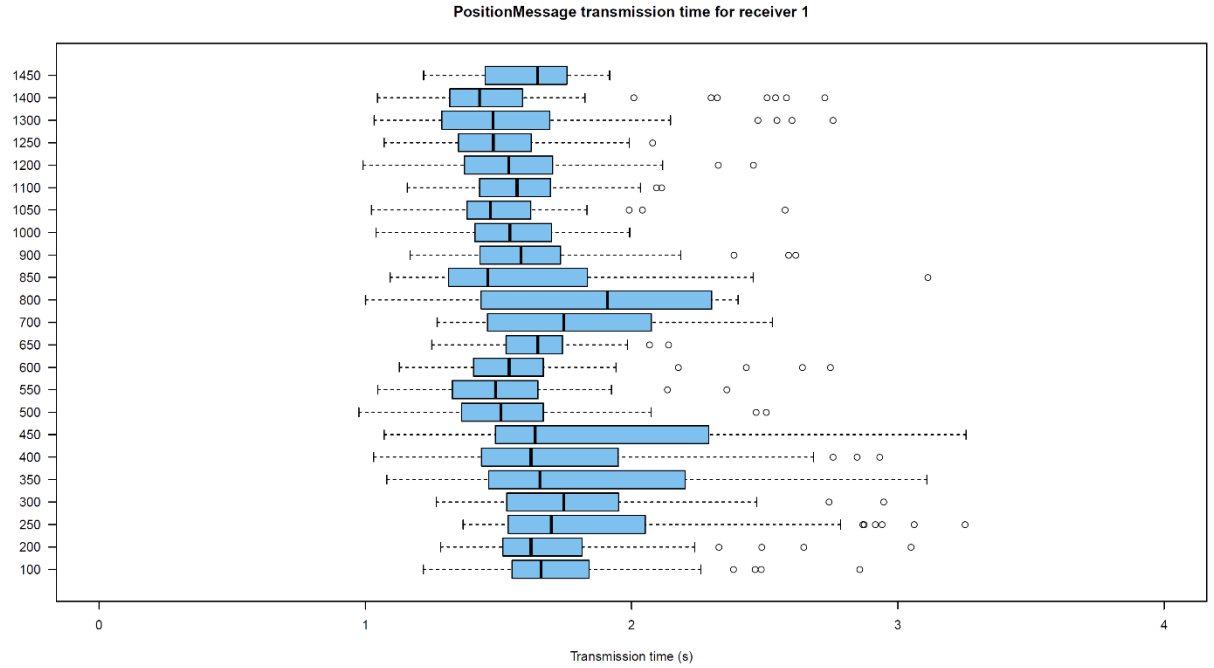


Figure D-4: Transmission Times of MQTT-Based NFFI Messages (Whole Diagram).

In total 2075 messages were published, of which 360 (17.35%) were lost (see Table D-6). The overall median of the transmission delay was 1.60 s (averaged over all messages from all publishers). The minimum delay was 0.98 s and the maximum delay was 3.26 s. As you can see in Figure D-4, most messages were near the median delay. In contrast to WS-N and MQTT, there are no high values for some publishers. This means that MQTT-SN (which is based on UDP) drops these messages at some point, while WS-N and MQTT still try to deliver them after more than 200 seconds.

Table D-6: Results from Experiments for MQTT-SN Anglova Scenario (Application Layer).

Messages Sent	Messages Lost	Delay (Min)	Delay (Overall Median)	Delay (Max)
2075	360 (17.35%)	0.98 s	1.60 s	3.26 s

D.6.4 WS-N with Modified Anglova Scenario

In this setup, we deploy the same services as in Section D.6.1 (WS-N broker, one WS-N subscriber on Node 1, WS-N producer software on Nodes 2 to 24). The measurements pertaining to network and application layers are presented next.

D.6.4.1 Network Layer

Table D-7 shows the results from the network analysis. The WS-N-based communication produced a data volume of 39 kbit/s. The message size was 1863 bytes as in Section D.6.1. The logs show that 2652 duplicate acknowledgments were produced and that 2741 TCP retransmissions were caused, from which 1821 were of type “spurious”.

Table D-7: Results from Experiments for WS-N Modified Anglova Scenario (Network Layer).

Data Volume per second	Message Size	TCP Duplicate ACK	TCP Spurious Retransmissions	TCP Retransmissions
39 kbit/s	1863 bytes	2652	1821	2741

D.6.4.2 Application Layer

In Figure D-5 the average transmission times of the messages are shown for each publisher in a boxplot diagram.

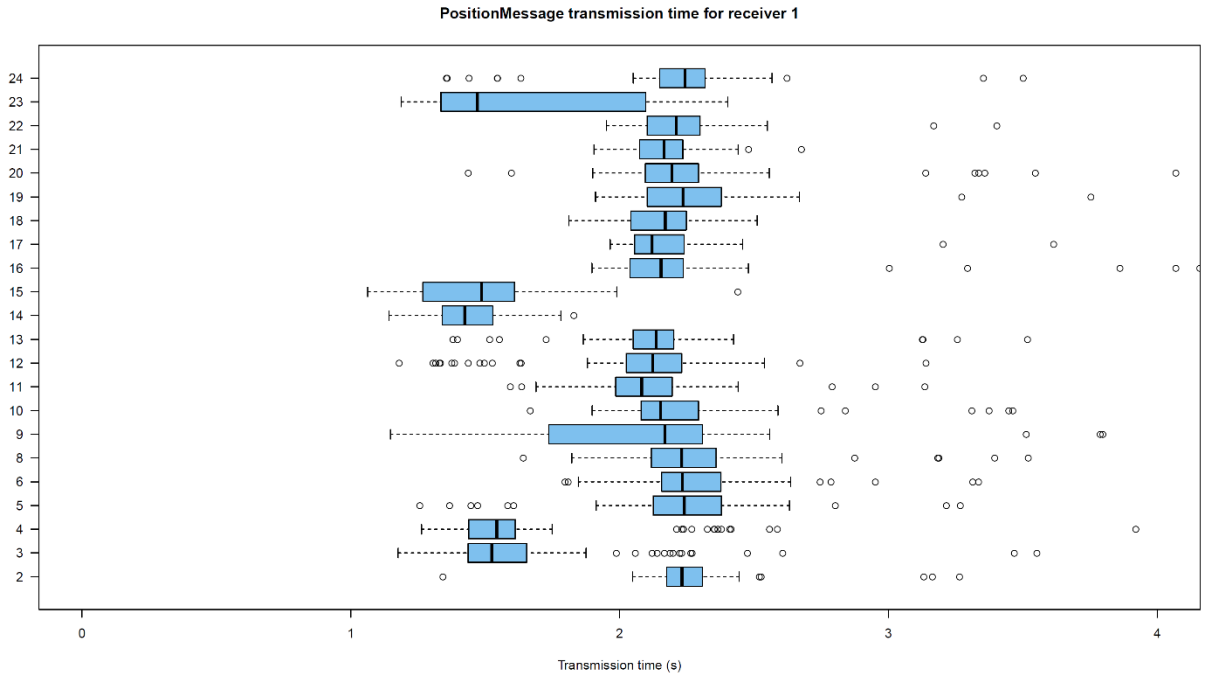


Figure D-5: Transmission Times of WS-N-Based NFFI Messages.

In total 1725 messages were published, of which 22 (1.28%) were lost (see Table D-8). The overall median of the transmission delay was 2.02 s (averaged over all messages from all publishers). The minimum delay was 1.06 s and the maximum delay was 2.67 s. This means all messages were near the median delay.

Table D-8: Results from Experiments for WS-N, Modified Anglova Scenario (Application Layer).

Messages Sent	Messages Lost	Delay (Min)	Delay (Overall Median)	Delay (Max)
1725	22 (1.28%)	1.06 s	2.02 s	79 s

D.6.5 MQTT with Modified Anglova Scenario

In this setup, we deploy the same services as in Section D.6.2 (VerneMQ broker, one MQTT subscriber on Node 1, MQTT producer software on Nodes 2 to 24). For the MQTT publisher the Quality of Service class *QoS0* was used. The measurements pertaining to network and application layers are presented next.

D.6.5.1 Network Layer

The results from the network analysis are shown in Table D-9. The MQTT-based communication produced a data volume of 33 kbit/s. The message size was 880 bytes as in Section D.6.2. The logs show that 2336 duplicate acknowledgements were produced and that 5091 TCP retransmissions were caused, from which 1999 were of type "spurious".

Table D-9: Results from Experiments for MQTT, Modified Anglova Scenario (Network Layer).

Data Volume per second	Message Size	TCP Duplicate ACK	TCP Spurious Retransmissions	TCP Retransmissions
33 kbit/s	880 bytes	2336	1999	5091

D.6.5.1 Application Layer

In Figure D-6 the average transmission times of the messages are shown for each publisher in a boxplot diagram.

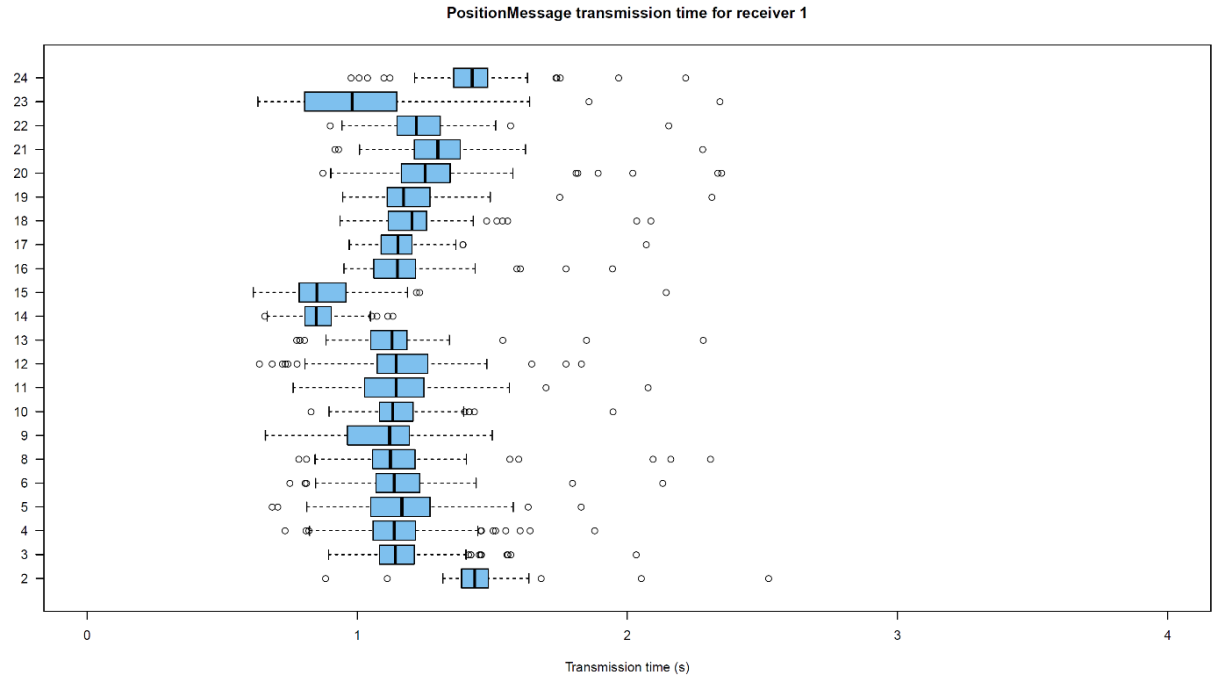


Figure D-6: Transmission Times of MQTT-Based NFFI Messages.

In total 2490 messages were published, from which 367 (14.74%) were lost (see Table D-10). The overall median of the transmission delay was 1.15 s (averaged over all messages from all publishers). The minimum delay was 0.62 s and the maximum delay was 1.64 s. This means all messages were near the median delay.

Table D-10: Results from Experiments for MQTT, Modified Adapted Anglova Scenario (Application Layer).

Messages Sent	Messages Lost	Delay (Min)	Delay (Overall Median)	Delay (Max)
2490	367 (14.74%)	0.62 s	1.15 s	6.8 s

D.6.6 MQTT-SN with Modified Anglova Scenario

In this setup, we deploy the same services as in Section D.6.3 (VerneMQ broker, MQTT-SN gateway, one MQTT subscriber on Node 1, MQTT-SN producer software on Nodes 2 to 24). For the MQTT publisher the Quality of Service class *QoS0* was used. The measurements pertaining to network and application layers are presented next.

D.6.6.1 Network Layer

The results from the network analysis are shown in Table D-11. The MQTT-SN-based communication produced a data volume of 14 kbit/s. The message size was 894 bytes as in Section D.6.3. Since MQTT-SN is UDP-based, there are no TCP retransmissions.

Table D-11: Results from Experiments for MQTT-SN, Modified Anglova Scenario (Network Layer).

Data Volume per Second	Message Size	TCP Duplicate ACK	TCP Spurious Retransmissions	TCP Retransmissions
14 kbit/s	894 bytes	N/A	N/A	N/A

D.6.6.2 Application Layer

In Figure D-7 the average transmission times of the messages are shown for each publisher in a boxplot diagram.

In total 2093 messages were published, of which 354 (16.91%) were lost (see Table D-12). The overall median of the transmission delay was 1.60 s (averaged over all messages from all publishers). The minimum delay was 0.90 s and the maximum delay was 2.17 s. This means all messages were near the median delay.

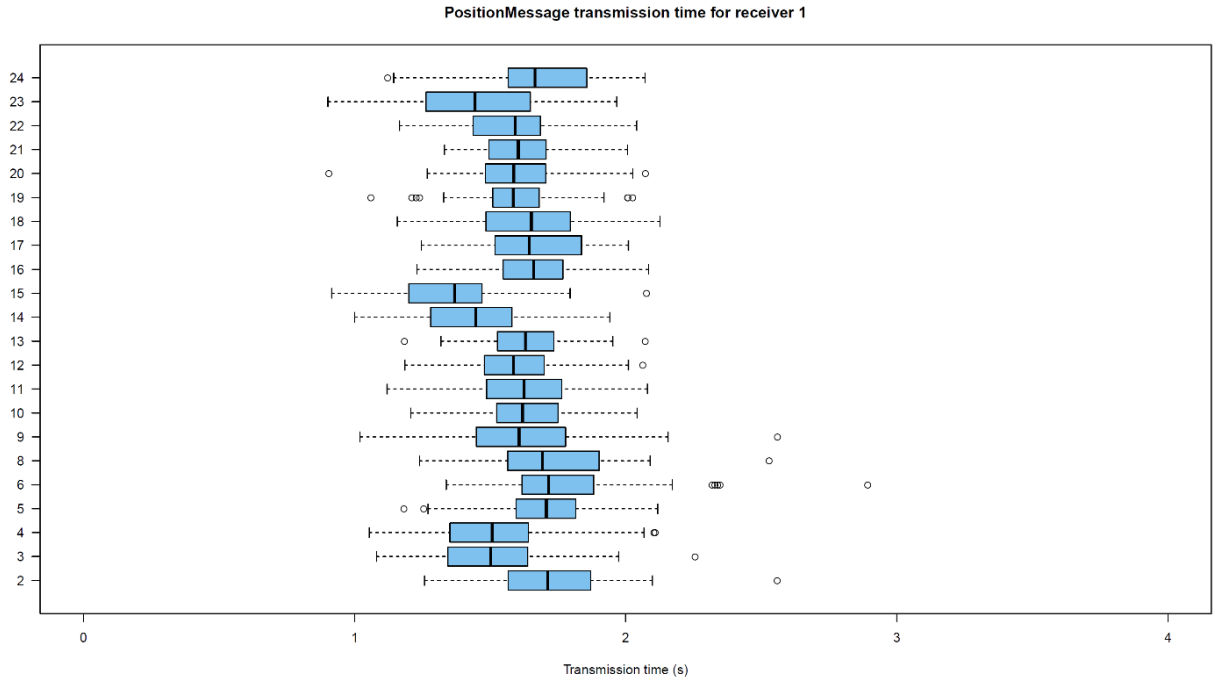


Figure D-7: Transmission Times of MQTT-Based NFFI Messages (Whole Diagram).

Table D-12: Results from Experiments for MQTT-SN Modified Anglova Scenario (Application Layer).

Messages Sent	Messages Lost	Delay (Min)	Delay (Overall Median)	Delay (Max)
2093	354 (16.91%)	0.90 s	1.60 s	12 s

D.6.7 Comparing Quality of Service Settings in MQTT/MQTT-SN

In this setup, we will compare two Quality of Service settings in MQTT and MQTT-SN. For this purpose, the MQTT publisher software was updated to use QoS1. The MQTT-related experiments above were conducted with QoS0. Besides this change of the publisher software, the same software as described above was used (VerneMQ broker, MQTT-SN gateway, one MQTT subscriber on Node 1, MQTT-SN producer software on Nodes 2 to 24). The measurements were conducted in the Modified Anglova scenario and are presented next.

Table D-13 shows the results from network analysis. Results from the experiments with QoS0 (cf. Sections D.6.2, D.6.3, D.6.5, and D.6.6) are also listed in the table for comparison reasons.

Table D-13: Comparison of MQTT, MQTT-SN for QoS0 and QoS1 (Network Layer).

Experiment	Data Volume per Second	Message Size	TCP Duplicate ACK	TCP Spurious Retransmissions	TCP Retransmissions
MQTT, QoS0, Modified Anglova	33 kbit/s	880 bytes	2336	1999	5091
MQTT-SN, QoS0, Modified Anglova	14 kbit/s	894 bytes	N/A	N/A	N/A
MQTT, QoS1, Modified Anglova	38 kbit/s	893 bytes	3255	1981	10565
MQTT-SN, QoS1, Modified Anglova	13 kbit/s	910 bytes	N/A	N/A	N/A

It can be seen from the table that MQTT produces about double the number of retransmission when used in reliable mode (QoS1). The produced data volume increased from 33 kbit/s to 38 kbit/s for MQTT with QoS1. The data for MQTT-SN remains the same when QoS1 was used.

In Figure D-8 and Figure D-9, the average transmission times of the messages are shown for MQTT (QoS1) and MQTT-SN (QoS1) using the Modified Anglova scenario.

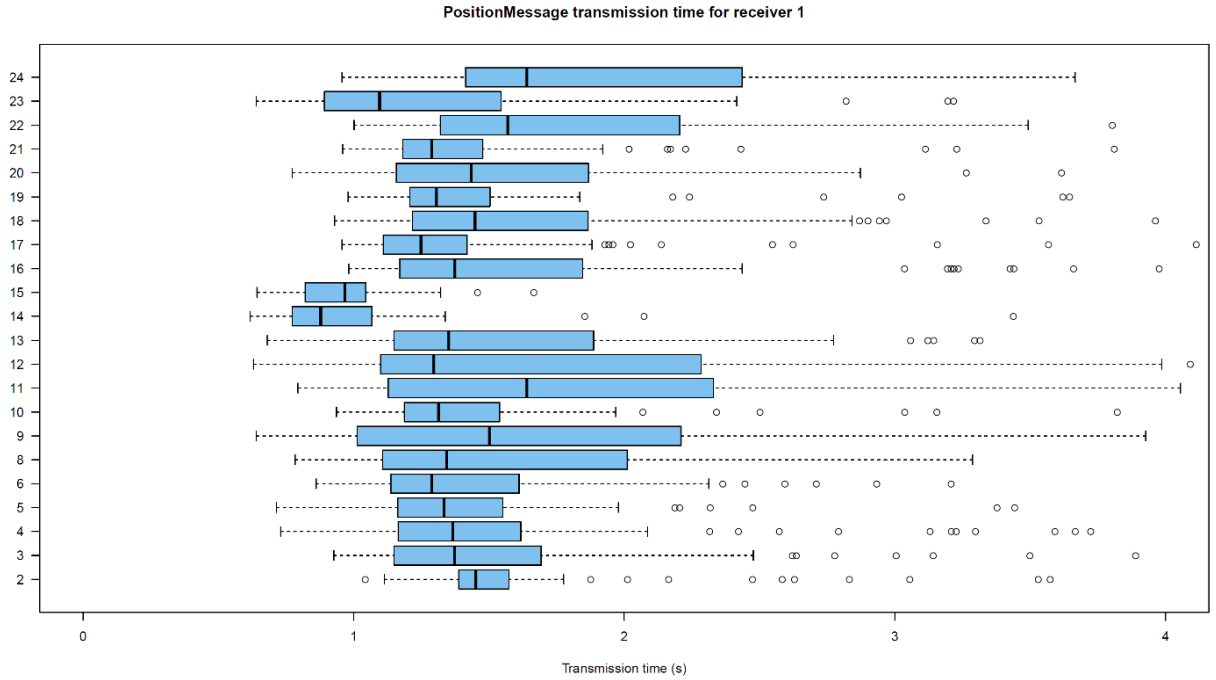


Figure D-8: Transmission Times of MQTT-Based NFFI Messages QoS1 (Modified Anglova).

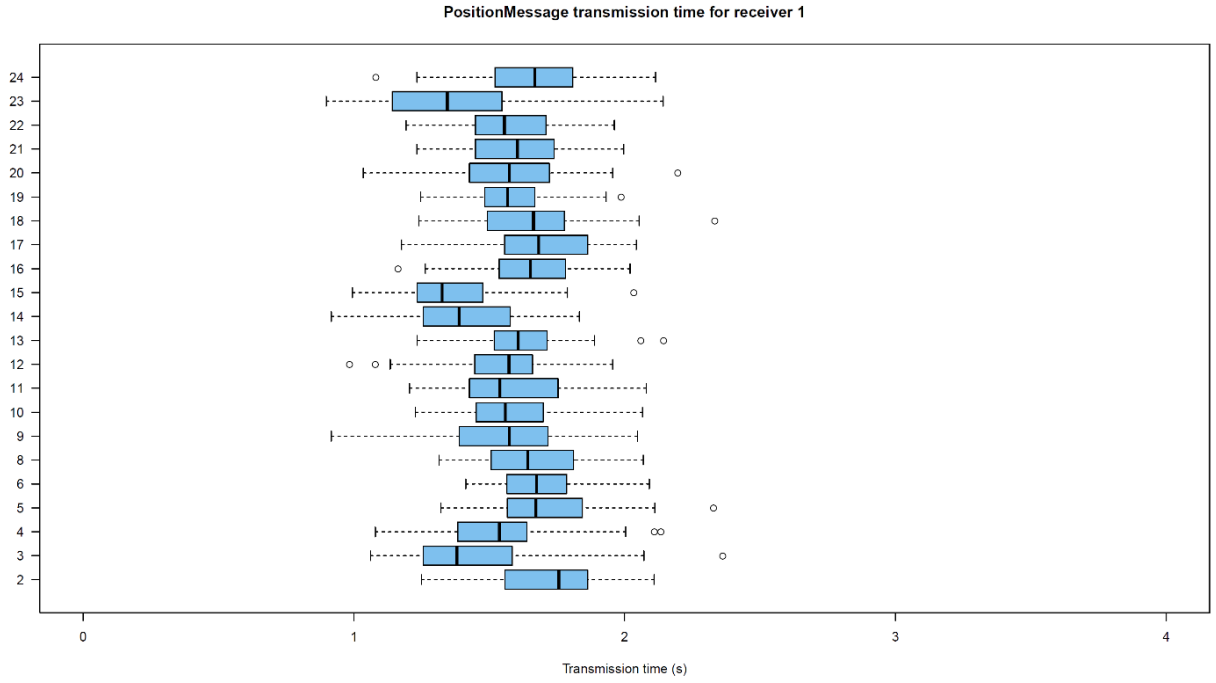


Figure D-9: Transmission Times of MQTT-SN-Based NFFI Messages QoS1 (Modified Anglova).

An overview of the results is shown in Table D-14. The results from the experiments with QoS0 (cf. Sections D.6.2, D.6.3, D.6.5, and D.6.6) are also listed in the table for comparison reasons.

Table D-14: Comparison of MQTT and MQTT-SN for QoS0 and QoS1 (Application Layer).

Experiment	Messages Sent	Messages Lost	Delay (Min)	Delay (Overall Median)	Delay (Max)
MQTT, QoS0, Modified Anglova	2490	367 (14.74%)	0.62 s	1.15 s	6.8 s
MQTT-SN, QoS0, Modified Anglova	2093	354 (16.91%)	0.90 s	1.60 s	12 s
MQTT, QoS1, Modified Anglova	2399	327 (13.63%)	0.62 s	1.34 s	62.43 s
MQTT-SN, QoS1, Modified Anglova	1663	8 (0.48%)	0.90 s	1.57 s	11.67 s

The results show that most values remain the same when MQTT or MQTT-SN are used with QoS1 instead of QoS0 (e.g., the transmission times). But the reliability improves significantly for MQTT-SN when QoS1 is used.

D.6.8 Comparison Analysis and Results

A comparison between results obtained with WS-N, MQTT and MQTT-SN in the two scenarios is presented next. The combined measurement results from Sections D.6.1 – Section D.6.7 used to support our analysis are presented in Table D-15 and Table D-16.

Table D-15: Overview of the Results from Experiments (Network Layer).

Experiment	Data Volume per Second	Message Size	TCP Duplicate ACK	TCP Spurious Retransmissions	TCP Retransmissions
Anglova, WS-N	40 kbit/s	1863 bytes	2468	1761	2761
Anglova, MQTT, QoS0	31 kbit/s	880 bytes	1922	1436	4281
Anglova, MQTT-SN, QoS0	13 kbit/s	894 bytes	N/A	N/A	N/A
Modified Anglova, WS-N	39 kbit/s	1863 bytes	2652	1821	2741
Modified Anglova, MQTT, QoS0	33 kbit/s	880 bytes	2336	1999	5091
Modified Anglova, MQTT-SN, QoS0	14 kbit/s	894 bytes	N/A	N/A	N/A
Modified Anglova, MQTT, QoS1	38 kbit/s	893 bytes	3255	1981	10565
Modified Anglova, MQTT-SN, QoS1	13 kbit/s	910 bytes	N/A	N/A	N/A

Table D-16: Overview of the Results from Experiments (Application Layer).

Experiment	Messages Sent	Messages Lost	Delay (min)	Delay (overall median)	Delay (max)
Anglova, WS-N	1697	22 (1.30%)	1.11 s	1.97 s	210 s
Anglova, MQTT, QoS0	2553+	383 (15%)	0,60 s	1,46 s	225 s
Anglova, MQTT-SN, QoS0	2075	360 (17.35%)	0.98 s	1.60 s	3.26 s
Modified Anglova, WS-N	1725	22 (1.28%)	1.06 s	2.02 s	79 s
Modified Anglova, MQTT, QoS0	2490	367 (14.74%)	0.62 s	1.15 s	6.8 s
Modified Anglova, MQTT-SN, QoS0	2093	354 (16.91%)	0.90 s	1.60 s	12 s
Modified Anglova, MQTT, QoS1	2399	327 (13.63%)	0.62 s	1.34 s	62.43 s
Modified Anglova, MQTT-SN, QoS1	1663	8 (0.48%)	0.90 s	1.57 s	11.67 s

From the evaluation of the experiments, it can be seen that:

- Overall (including the whole communications stack), MQTT-SN produces a data volume of about 13 – 14 kbit/s compared to about 31 – 38 kbit/s (MQTT) and about 39 – 40 kbit/s (WS-N).
- The message sizes of MQTT and MQTT-SN (about 900 bytes) are about half the size of WS-N (about 1850 bytes).
- MQTT caused notably more TCP retransmissions than WS-N, which is in contrast to former experiments with Wi-Fi links, where the opposite could be observed. The causes for the high number of retransmissions have to be further analysed in the future. When QoS1 was used with MQTT there were even much more retransmissions (double the size of QoS0).
- WS-N has less message losses (1.3%) compared to MQTT (15%) and MQTT-SN (17%) if these are run with QoS0. The use of QoS1 with MQTT does not increase the reliability significantly (still 14%). But for MQTT-SN the reliability improves significantly by using QoS1 (packet loss 0.48% vs. 17%).
- The average delay is higher for WS-N than for MQTT or MQTT-SN. MQTT has the lowest delay.
- The transmission results of MQTT-SN do not contain messages with a very high delay (see e.g., Anglova, MQTT with 225 s). But the loss rate is slightly higher than MQTT when used with QoS0. It seems like MQTT-SN discards messages if the transmission takes too long. The use of QoS1 with MQTT-SN improves the reliability significantly (0.48% message loss) and thus leads to an even higher reliability than WS-N.

In summary, it could be seen that WS-N, MQTT and MQTT-SN behaved slightly different in the scenarios we used for the experiments. For QoS0, MQTT and MQTT-SN produced more message losses than WS-N, while WS-N produced higher delays for the transmission of messages. MQTT-SN was very reliable when used with QoS1. The delay remained the same for MQTT-SN when using QoS1 instead of QoS0.

Since we used a realistic radio model in conjunction with challenging tactical scenarios, TCP produced many "spurious" TCP retransmits. This indicates that TCP is not well suited for the kind of wireless networks used in this scenario. It has to be analysed further why MQTT caused more TCP retransmissions than WS-N in this case, while we observed the opposite (MQTT causing half retransmission than WS-N) in former experiments with less challenging Wi-Fi links.

One could expect that MQTT-SN, which is based on UDP, would be better suited for these kinds of scenarios and would have lower transmission delays. But our results show that the MQTT-SN implementation had an about 50% higher transmission delay than MQTT when used with QoS0 and a slightly higher loss rate (17% vs. 15%). One has to keep in mind that MQTT-SN was deployed by using an additional MQTT/MQTT-SN gateway. Possibly some amount of the delay was caused by the processing times of this additional component. For QoS1 the transmission delays remained the same for MQTT-SN and increased for MQTT. Regarding the reliability of message delivery, MQTT did not benefit notably from using QoS1, while MQTT-SN benefits from this QoS setting significantly.

For BFT the transmission delay is most important. Since newer positions are transmitted periodically every 10 seconds, the transmission of outdated position messages does not necessarily increase the user experience. Thus, for this kind of services the higher reliability of WS-N is not essential for the choice of the middleware and MQTT has performed best in the two scenarios we have investigated.

For other services which rely on a reliable delivery of messages, the use of MQTT-SN with QoS1 could be considered, because MQTT-SN with QoS1 was the most reliable middleware in our experiments. Additionally, the delay was lower than for WS-N.

Furthermore, the results from the network analysis showed that MQTT-SN produces less than half the amount of network data per second than MQTT and WS-N. We expect that MQTT-SN is better suited for resource constrained devices and could be superior in networks with very limited data rates. But further experiments are needed to prove these assumptions.

D.7 CONCLUSIONS

In this paper, we have investigated the three industry standards WS-N, MQTT, and MQTT-SN in a comparative study using the Anglova scenario (both the original, and a modified version created by Switzerland) using Swiss Wideband TDMA models. We used EMANE and DAVC as our testbed, hosted and operated by ARL, USA. The service we used was BFT with NFFI data, as implemented by FFI, Norway. Fraunhofer FKIE, Germany provided the analysis tools that were used to evaluate our results.

In our experiments, we considered that for the BFT service transmission delay is the most important metric. Since newer positions are transmitted periodically every 10 seconds, the transmission of outdated position messages does not increase the user experience. Hence, for MQTT it makes sense to use QoS0 to reduce overhead for such messages, as reliability is not needed.

We found that MQTT-SN produces a data volume of about 13 – 14 kbit/s compared to about 31 – 38 kbit/s (MQTT) and about 39 – 40 kbit/s (WS-N). The message sizes of MQTT and MQTT-SN are about half the size of WS-N, which makes sense since WS-N has a SOAP message layer that MQTT does not. MQTT caused notably more TCP retransmissions than WS-N, which is in contrast to former experiments with Wi-Fi links, where the opposite could be observed. The causes for the high number of retransmissions have to be further analysed in the future.

WS-N has less message losses compared to MQTT and MQTT-SN if these are run with QoS0. The use of QoS1 with MQTT does not increase the reliability significantly. But, for MQTT-SN, the reliability improves

significantly by using QoS1. This makes sense since the underlying TCP in MQTT can be expected to provide some reliability, unlike UDP in MQTT-SN, which requires the additional handshaking of QoS1 to increase its reliability. The loss rate is slightly higher than MQTT when used with QoS0. It seems like MQTT-SN discards messages if the transmission takes too long. The use of QoS1 with MQTT-SN improves the reliability significantly and thus leads to an even higher reliability than WS-N.

Of specific importance to our BFT service, was, as mentioned, the delay. The average delay is higher for WS-N than for MQTT or MQTT-SN. MQTT has the lowest delay. Hence, we can conclude that for BFT services, MQTT can be a better choice than WS-N in Wideband tactical networks with similar characteristics to what we evaluated here.

D.8 ACKNOWLEDGMENTS

We thank Mr. Chien Hsieh (ICF) for performance enhancements to DAVC and for installing and configuring the TDMA-based EMANE radio models.

D.9 REFERENCES

[1] Marco Manso, Norman Jansen, Kevin Chan, Andrew Toth, Trude H. Bloebaum and Frank T. Johnsen. Mobile Tactical Force Situational Awareness: Evaluation of Message Broker Middleware for Information Exchange, 23rd International Command and Control Research and Technology Symposium (ICCRTS), Pensacola, FL, USA, 2018.

[2] OASIS. Web Services Brokered Notification 1.3 (WSBrokeredNotification), OASIS Standard, 1 October 2006. http://docs.oasis-open.org/wsn/wsn-ws brokered notification-1.3-spec-os.pdf.

[3] OASIS. MQTT Version 3.1.1, OASIS Standard 29 October 2014. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf.

[4] N. Suri, A. Hansson, J. Nilsson, P. Lubkowski, K. Marcus, M. Hauge, K. Lee, B. Buchin, L. Misirlioglu, and M. Peuhkuri. A Realistic Military Scenario and Emulation Environment for Experimenting with Tactical Communications and Heterogeneous Networks, 2016 International Conference on Military Communications and Information Systems (ICMCIS 2016), Brussels, Belgium, 2016.

[5] U.S. Naval Research Laboratory. Extendable Mobile Ad-hoc Network Emulator (EMANE), accessed 2019-03-18, https://www.nrl.navy.mil/itd/ncs/products/emane.

[6] A. Nikodemski, J.-F. Wagen, F. Buntschu, C. Gisler et G. Bovet, Reproducing Measured Manet Radio Performances Using the EMANE Framework, IEEE Communications Magazine, vol. 56, pp. 155-155, October 2018.

[7] J.-F. Wagen, V. Adalid, G. Waeber, F. Buntschu et G. Bovet, Performance Profiling of Radio Models and Anglova Based Scenarios, 2019 International Conference on Military Communications and Information Systems (ICMCIS 2019), Budva, Montenegro, 2019.

[8] Trude H. Bloebaum and Frank T. Johnsen. CWIX 2014 Core Enterprise Services Experimentation, FFI-Report 2014/01510. https://www.ffi.no/no/Rapporter/14-01510.pdf.

[9] VerneMQ homepage. VerneMQ – A MQTT Broker that is Scalable, Enterprise Ready, and Open Source, Accessed 2019-03-18, https://vernemq.com/.

[10] Eclipse, MQTT-SN Transparent Gateway, Accessed 2019-03-18, https://www.eclipse.org/paho/components/mqtt-sn-transparent-gateway/.

[11] CCDC Army Research Laboratory. Network Science Research Laboratory, accessed 2019-06-13, https://www.arl.army.mil/www/default.cfm?page=2485.

[12] OLSR Optimized Link State Routing Protocol. Accessed 2019-06-13, http://www.olsr.org.

[13] 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.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's References	3. Further Reference	4. Security Classification of Document
	STO-TR-IST-150 AC/323(IST-150)TP/1008	ISBN 978-92-837-2328-8	PUBLIC RELEASE

5. Originator

Science and Technology Organization
North Atlantic Treaty Organization
BP 25, F-92201 Neuilly-sur-Seine Cedex, France

6. Title

NATO Core Services Profiling for Hybrid Tactical Networks

7. Presented at/Sponsored by

Final report of STO Research Task IST-150/RTG-072.

8. Author(s)/Editor(s)	9. Date
Multiple	March 2021
10. Author's/Editor's Address	**11. Pages**
Multiple	128

12. Distribution Statement

There are no restrictions on the distribution of this document. Information about the availability of this and other STO unclassified publications is given on the back cover.

13. Keywords/Descriptors

COAP; DIL; Disadvantaged Grids; Message-Oriented Middleware; MQTT; Publish/subscribe; Request/response; REST; Service-Oriented Architecture; SOAP; Tactical Networks; WS-Notification

14. Abstract

Federated Mission Networking (FMN) is the main context and motivation for our work. IST-150 "NATO Core Services Profiling for Hybrid Tactical Networks" is intended to provide knowledge about services at the tactical level, and possibly feed into future spirals of FMN targeting the tactical level specifically. We target Service-Oriented Architecture (SOA) in the tactical domain, and specifically the Message-Oriented Middleware (MOM) Core Service.

MOM covers two communication patterns: Publish/subscribe communication and request/response communication. In IST-150 we have experimented with both these patterns, evaluating industry standards in typical tactical network settings, using both military radio hardware and network emulators. Based on our findings, we recommend the industry standard Message Queueing Telemetry Transport (MQTT) for publish/subscribe, and we recommend using REST-based services replacing the HTTP/TCP transport with CoAP for request/response type services. The report covers these findings in detail, supporting these recommendations.

Our work should be taken both as input to future FMN spirals as well as continuing IST research task groups where MOM services play a role.

NORTH ATLANTIC TREATY ORGANIZATION

BP 25
F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE
Télécopie 0(1)55.61.22.99 • E-mail mailbox@cso.nato.int

SCIENCE AND TECHNOLOGY ORGANIZATION

DIFFUSION DES PUBLICATIONS
STO NON CLASSIFIEES

Les publications de l'AGARD, de la RTO et de la STO peuvent parfois être obtenues auprès des centres nationaux de distribution indiqués ci-dessous. Si vous souhaitez recevoir toutes les publications de la STO, ou simplement celles qui concernent certains Panels, vous pouvez demander d'être inclus soit à titre personnel, soit au nom de votre organisation, sur la liste d'envoi.

Les publications de la STO, de la RTO et de l'AGARD sont également en vente auprès des agences de vente indiquées ci-dessous.

Les demandes de documents STO, RTO ou AGARD doivent comporter la dénomination « STO », « RTO » ou « AGARD » selon le cas, suivi du numéro de série. Des informations analogues, telles que le titre est la date de publication sont souhaitables.

Si vous souhaitez recevoir une notification électronique de la disponibilité des rapports de la STO au fur et à mesure de leur publication, vous pouvez consulter notre site Web (http://www.sto.nato.int/) et vous abonner à ce service.

CENTRES DE DIFFUSION NATIONAUX

ALLEMAGNE
Streitkräfteamt / Abteilung III
Fachinformationszentrum der Bundeswehr (FIZBw)
Gorch-Fock-Straße 7, D-53229 Bonn

BELGIQUE
Royal High Institute for Defence – KHID/IRSD/RHID
Management of Scientific & Technological Research
for Defence, National STO Coordinator
Royal Military Academy – Campus Renaissance
Renaissancelaan 30, 1000 Bruxelles

BULGARIE
Ministry of Defence
Defence Institute "Prof. Tsvetan Lazarov"
"Tsvetan Lazarov" bul no.2
1592 Sofia

CANADA
DGSIST 2
Recherche et développement pour la défense Canada
60 Moodie Drive (7N-1-F20)
Ottawa, Ontario K1A 0K2

DANEMARK
Danish Acquisition and Logistics Organization
(DALO)
Lautrupbjerg 1-5
2750 Ballerup

ESPAGNE
Área de Cooperación Internacional en I+D
SDGPLATIN (DGAM)
C/ Arturo Soria 289
28033 Madrid

ESTONIE
Estonian National Defence College
Centre for Applied Research
Riia str 12
Tartu 51013

ETATS-UNIS
Defense Technical Information Center
8725 John J. Kingman Road
Fort Belvoir, VA 22060-6218

FRANCE
O.N.E.R.A. (ISP)
29, Avenue de la Division Leclerc
BP 72
92322 Châtillon Cedex

GRECE (Correspondant)
Defence Industry & Research General
Directorate, Research Directorate
Fakinos Base Camp, S.T.G. 1020
Holargos, Athens

HONGRIE
Hungarian Ministry of Defence
Development and Logistics Agency
P.O.B. 25
H-1885 Budapest

ITALIE
Ten Col Renato NARO
Capo servizio Gestione della Conoscenza
F. Baracca Military Airport "Comparto A"
Via di Centocelle, 301
00175, Rome

LUXEMBOURG
Voir Belgique

NORVEGE
Norwegian Defence Research
Establishment
Attn: Biblioteket
P.O. Box 25
NO-2007 Kjeller

PAYS-BAS
Royal Netherlands Military
Academy Library
P.O. Box 90.002
4800 PA Breda

POLOGNE
Centralna Biblioteka Wojskowa
ul. Ostrobramska 109
04-041 Warszawa

PORTUGAL
Estado Maior da Força Aérea
SDFA – Centro de Documentação
Alfragide
P-2720 Amadora

REPUBLIQUE TCHEQUE
Vojenský technický ústav s.p.
CZ Distribution Information Centre
Mladoboleslavská 944
PO Box 18
197 06 Praha 9

ROUMANIE
Romanian National Distribution
Centre
Armaments Department
9-11, Drumul Taberei Street
Sector 6
061353 Bucharest

ROYAUME-UNI
Dstl Records Centre
Rm G02, ISAT F, Building 5
Dstl Porton Down
Salisbury SP4 0JQ

SLOVAQUIE
Akadémia ozbrojených síl gen.
M.R. Štefánika, Distribučné a
informačné stredisko STO
Demänová 393
031 01 Liptovský Mikuláš 1

SLOVENIE
Ministry of Defence
Central Registry for EU & NATO
Vojkova 55
1000 Ljubljana

TURQUIE
Milli Savunma Bakanlığı (MSB)
ARGE ve Teknoloji Dairesi
Başkanlığı
06650 Bakanliklar – Ankara

AGENCES DE VENTE

The British Library Document
Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
ROYAUME-UNI

Canada Institute for Scientific and
Technical Information (CISTI)
National Research Council Acquisitions
Montreal Road, Building M-55
Ottawa, Ontario K1A 0S2
CANADA

Les demandes de documents STO, RTO ou AGARD doivent comporter la dénomination « STO », « RTO » ou « AGARD » selon le cas, suivie du numéro de série (par exemple AGARD-AG-315). Des informations analogues, telles que le titre et la date de publication sont souhaitables. Des références bibliographiques complètes ainsi que des résumés des publications STO, RTO et AGARD figurent dans le « NTIS Publications Database » (http://www.ntis.gov).

NORTH ATLANTIC TREATY ORGANIZATION

BP 25
F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE
Télécopie 0(1)55.61.22.99 • E-mail mailbox@cso.nato.int

SCIENCE AND TECHNOLOGY ORGANIZATION

DISTRIBUTION OF UNCLASSIFIED STO PUBLICATIONS

AGARD, RTO & STO publications are sometimes available from the National Distribution Centres listed below. If you wish to receive all STO reports, or just those relating to one or more specific STO Panels, they may be willing to include you (or your Organisation) in their distribution.

STO, RTO and AGARD reports may also be purchased from the Sales Agencies listed below.

Requests for STO, RTO or AGARD documents should include the word 'STO', 'RTO' or 'AGARD', as appropriate, followed by the serial number. Collateral information such as title and publication date is desirable.

If you wish to receive electronic notification of STO reports as they are published, please visit our website (http://www.sto.nato.int/) from where you can register for this service.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Royal High Institute for Defence – KHID/IRSD/RHID
Management of Scientific & Technological Research for Defence, National STO Coordinator
Royal Military Academy – Campus Renaissance
Renaissancelaan 30
1000 Brussels

BULGARIA
Ministry of Defence
Defence Institute "Prof. Tsvetan Lazarov"
"Tsvetan Lazarov" bul no.2
1592 Sofia

CANADA
DSTKIM 2
Defence Research and Development Canada
60 Moodie Drive (7N-1-F20)
Ottawa, Ontario K1A 0K2

CZECH REPUBLIC
Vojenský technický ústav s.p.
CZ Distribution Information Centre
Mladoboleslavská 944
PO Box 18
197 06 Praha 9

DENMARK
Danish Acquisition and Logistics Organization (DALO)
Lautrupbjerg 1-5
2750 Ballerup

ESTONIA
Estonian National Defence College
Centre for Applied Research
Riia str 12
Tartu 51013

FRANCE
O.N.E.R.A. (ISP)
29, Avenue de la Division Leclerc – BP 72
92322 Châtillon Cedex

GERMANY
Streitkräfteamt / Abteilung III
Fachinformationszentrum der Bundeswehr (FIZBw)
Gorch-Fock-Straße 7
D-53229 Bonn

GREECE (Point of Contact)
Defence Industry & Research General Directorate, Research Directorate
Fakinos Base Camp, S.T.G. 1020
Holargos, Athens

HUNGARY
Hungarian Ministry of Defence
Development and Logistics Agency
P.O.B. 25
H-1885 Budapest

ITALY
Ten Col Renato NARO
Capo servizio Gestione della Conoscenza
F. Baracca Military Airport "Comparto A"
Via di Centocelle, 301
00175, Rome

LUXEMBOURG
See Belgium

NETHERLANDS
Royal Netherlands Military Academy Library
P.O. Box 90.002
4800 PA Breda

NORWAY
Norwegian Defence Research Establishment, Attn: Biblioteket
P.O. Box 25
NO-2007 Kjeller

POLAND
Centralna Biblioteka Wojskowa
ul. Ostrobramska 109
04-041 Warszawa

PORTUGAL
Estado Maior da Força Aérea
SDFA – Centro de Documentação
Alfragide
P-2720 Amadora

ROMANIA
Romanian National Distribution Centre
Armaments Department
9-11, Drumul Taberei Street
Sector 6
061353 Bucharest

SLOVAKIA
Akadémia ozbrojených síl gen M.R. Štefánika, Distribučné a informačné stredisko STO
Demänová 393
031 01 Liptovský Mikuláš 1

SLOVENIA
Ministry of Defence
Central Registry for EU & NATO
Vojkova 55
1000 Ljubljana

SPAIN
Área de Cooperación Internacional en I+D
SDGPLATIN (DGAM)
C/ Arturo Soria 289
28033 Madrid

TURKEY
Milli Savunma Bakanlığı (MSB)
ARGE ve Teknoloji Dairesi Başkanlığı
06650 Bakanliklar – Ankara

UNITED KINGDOM
Dstl Records Centre
Rm G02, ISAT F, Building 5
Dstl Porton Down, Salisbury SP4 0JQ

UNITED STATES
Defense Technical Information Center
8725 John J. Kingman Road
Fort Belvoir, VA 22060-6218

SALES AGENCIES

The British Library Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
UNITED KINGDOM

Canada Institute for Scientific and Technical Information (CISTI)
National Research Council Acquisitions
Montreal Road, Building M-55
Ottawa, Ontario K1A 0S2
CANADA

Requests for STO, RTO or AGARD documents should include the word 'STO', 'RTO' or 'AGARD', as appropriate, followed by the serial number (for example AGARD-AG-315). Collateral information such as title and publication date is desirable. Full bibliographical references and abstracts of STO, RTO and AGARD publications are given in "NTIS Publications Database" (http://www.ntis.gov).

ISBN 978-92-837-2328-8