ARL-TR-10199 ● SEP 2025

DEVCOM ARMY RESEARCH LABORATORY

Efficient Group- and Information-Centric Communications in Mobile Military Heterogeneous Networks

by Christoph Barz, Hans-Georg Courté, Eelco Cramer, Sepideh Fouladgar, Anders Hansson, Mariann Hauge, Anne Marie Hegland, Arjen Holtzer, Kelvin M. Marcus, Andreas Martens, Levent Misirlioglu, Jan Nilsson, Markus Peuhkuri, Mathias Rautenberg, Eric Saliba, Niranjan Suri, Mauro Tortonesi, Bastiaan Wissingh, Luc Bonnafoux, Maggie Breedy, Lorenzo Campioni, Roberto Fronteddu, Alessandro Morelli, Cedric Nortier, Robert Seepers, and Ronald in ‘t Velt

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-10199 ● SEP 2025

Efficient Group- and Information-Centric Communications in Mobile Military Heterogeneous Networks

Kelvin M. Marcus and Niranjan Suri
DEVCOM Army Research Laboratory

Christoph Barz
Fraunhofer Institute for Communication, Information Processing, and Ergonomics (FKIE), Germany

Hans-Georg Courté
Airbus, Germany

Eelco Cramer, Arjen Holtzer, Bastiaan Wissingh, Robert Seepers, and Ronald in 't Velt
Netherlands Organisation for Applied Scientific Research (TNO), The Netherlands

Sepideh Fouladgar, Eric Saliba, Luc Bonnafoux, and Cedric Nortier
Thales, France

Anders Hansson and Jan Nilsson
Swedish Defence Research Agency (FOI), Sweden

Mariann Hauge
Norwegian Defence Research Establishment (FFI), Norway

Anne Marie Hegland
Kongsberg, Norway

Andreas Martens
IBM, United Kingdom

Levent Misirlioglu
MilSoft, Turkey

Markus Peuhkuri
Aalto University, Finland

Mathias Rautenberg
Rohde and Schwarz, Germany

Mauro Tortonesi
University of Ferrara, Italy

Maggie Breedy, Lorenzo Campioni, Roberto Fronteddu, and Alessandro Morelli
Florida Institute for Human & Machine Cognition, USA

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
September 2025	Technical Report	2/28/2018	2/28/2022

4. TITLE AND SUBTITLE
Efficient Group- and Information-Centric Communications in Mobile Military Heterogeneous Networks

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER	5e. TASK NUMBER	5f. WORK UNIT NUMBER
NATO IST-161 RTG		

6. AUTHORS
Christoph Barz, Hans-Georg Courté, Eelco Cramer, Sepideh Fouladgar, Anders Hansson, Mariann Hauge, Anne Marie Hegland, Arjen Holtzer, Kelvin M. Marcus, Andreas Martens, Levent Misirlioglu, Jan Nilsson, Markus Peuhkuri, Mathias Rautenberg, Eric Saliba, Niranjan Suri, Mauro Tortonesi, Bastiaan Wissingh, Luc Bonnafoux, Maggie Breedy, Lorenzo Campioni, Roberto Fronteddu, Alessandro Morelli, Cedric Nortier, Robert Seepers, and Ronald in ‘t Velt

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-I Adelphi, MD 20783	ARL-TR-10199

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NATO Science and Technology Organization		

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report documents the findings of NATO Science and Technology Organization Research Task Group IST-161. The aim of this group was to investigate potential solutions for group communications in mobile tactical networks to learn more about their behavior. Experiments were conducted using a realistic military scenario called Anglova that contains a battalion-sized mechanized unit comprising four companies. The group considered architectural aspects, introduced a group communications protocol taxonomy, and discussed security aspects. Experiments showed that the Generic Data Exchange Mechanism has the best overall performance in the tested scenarios, followed by DisService and ZeroMQ with NORM. So-called broker-based solutions, such as the Neural Autonomic Transport System (NATS), were shown to be unsuitable for the mobile tactical edge. An alternative approach called named data networking (NDN) was investigated. NDN proved to be insufficiently mature for the group’s use case, missing several essential elements in the protocol stack, including an efficient content synchronization mechanism. The IST-161 group maintained and extended the Anglova experimentation environment with new scripts and emulation models and produced eight scientific papers.

15. SUBJECT TERMS
group communications; tactical edge; mobile military networks; Anglova; emulation; Generic Data Exchange Mechanism; GDEM; DisService; ZeroMQ; NORM; named data networking; NDN; information-centric networking; ICN; synchronized cooperative broadcast; SCB; jamming; Blue Force tracking; BFT; reports; sensor data; taxonomy; security; Military Information Sciences; Network, Cyber, and Computational Sciences

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	143

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Niranjan Suri	(301) 394-5626

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

This is the final report of the NATO Science and Technology Organization (STO) IST-161 Research Task Group (RTG).

1.1 Goal and Scope of the Work

The goal of the IST-161 RTG is to investigate potential solutions for group communications in mobile tactical networks. It focuses on the Information and Communications Technology infrastructure that should be in place to support user applications. In technical terms, IST-161 focuses on transmission, network, and middleware layers. The RTG compared existing solutions, identified merits of those solutions, and identified and implemented improvements. The main goal was to improve the understanding of the design of group communication solutions and their behavior. A secondary goal was to examine novel techniques for group communications such as information-centric networking (ICN) and named data networking (NDN).

1.2 Problem Description

The most important services at the mobile tactical edge are described in MC 0640 (NATO 2019b). The IST-161 RTG has been focusing on the battlegroup and company levels, noting that the minimum services to be supported include audio-based collaboration (tactical voice), text-based collaboration (tactical chat), and message exchange to establish an operational picture for Command and Control (C2) purposes. This includes Friendly Force Tracking information, battle space object, and possibly other messages relevant for creating a Recognized Ground Picture. Operational plans and orders are stated to be communicated via the tactical voice service as a minimal requirement. All these services are typically "point-to-multi-point" services, meaning that a voice or data message is always shared with a group of users within a unit (e.g., a company) or between multiple units (e.g., in a battlegroup).

While voice communications have been around for ages and will remain, the need for sharing valuable data on the battlefield is steadily increasing. While Blue Force tracking (BFT) and C2 messages remain crucial, more and more sensor data (SD) becomes available at the tactical edge and the value of all this information needs to be leveraged to maintain information superiority. In addition, there is an increasing need to operate in a coalition setting, also at the mobile tactical edge. Sharing all this data in a mobile environment that is typically characterized as denied, degraded, intermittent, and limited (DDIL) requires an improved understanding of

data dissemination methods and communication network technology and their efficiency and reliability in mobile military networks.

Sections 1.2.1 and 1.2.2 describe the group communications challenges that have been addressed by the RTG and list the research questions that were handled by the group when managing the challenges.

1.2.1 Overall Challenges

Tactical edge networks are characterized by their heterogeneity in networking and radio technologies, high network dynamics, and limited bandwidth, posing challenges to efficient communications—especially in larger coalition settings. Enabling group communications in the coalition heterogeneous tactical environment still poses a major challenge. For example, existing routing strategies show limitations in terms of scalability and efficiency, which is especially costly when running group services between soldiers in the field. In addition, many data dissemination methods rely on a centralized architecture, making them unsuitable for the mobile tactical environment where the system should be able to operate autonomously.

Because of the nature of tactical networks, information exchange mechanisms must be efficient when it comes to the consumption of bandwidth resources. This means that functionality in the communications stack must be efficiently designed as well. Duplication of similar functionality at different layers of the communications stack (e.g., routing, reliability mechanisms) should be avoided to minimize the signaling overhead on the network. Therefore, the design and architecture of a group communication solution require careful consideration.

In addition, communication protocols and middleware for group communications come in different shapes and sizes and may cover different functionalities. This means that it is not straightforward to make fair performance comparisons between solutions. An approach to classify and structure the different protocols can make this easier.

Different solutions for group communications may perform differently in different scenarios. Scenario aspects (e.g., mobility patterns, the types of radios that are used and their topology, and the availability of electromagnetic spectrum) may impact the performance of one solution differently than another. Learning about the impact of different scenarios on different solutions requires an experimentation environment where these aspects can be modeled. Preferably, such an experimentation environment is made available to other researchers in the community, since it will improve the comparability of scientific results.

Finally, the few solutions that have been developed specifically for group communications in mobile tactical networks are usually proprietary, which makes them unavailable for research. More importantly, it makes them unavailable for coalition use within NATO and Federated Mission Networking (FMN). Therefore, this RTG focuses on researching and developing solutions that are open to either the general public or NATO and its partners.

1.2.2 Research Questions

In addition to addressing the challenges presented in Section 1.2.1, the RTG formulated the following research questions to use as guidelines throughout the lifespan of IST-161:

- How do different approaches for efficient data dissemination compare (both from a functional as well as a performance perspective) to each other, including disruptive technologies (e.g., ICN approaches) that are outside of traditional IP architecture? Which ones are the most promising?
- How can the performance (i.e., efficiency and reliability) of different approaches be improved?
- What are the roles of multicast routing and/or forwarding when using data dissemination middleware?
- What are the most important trade-offs when establishing mobile military heterogeneous networks at the mobile tactical edge to support data and communication group services?
- How can interoperability be improved by applying the lessons learned from IST-161?
- How can group services and unicast services coexist on the same tactical infrastructure?

The RTG's work has provided insight to the answers to many of these questions. However, some are left for future work, such as the question about coexistence of unicast and multicast traffic.

1.3 Approach

The RTG mainly approached the research questions by experimental assessment of candidate solutions running the Anglova scenario[*] in the Extendable Mobile Ad-hoc Networking Emulator (EMANE) environment. When selecting protocols, the RTG looked for candidate solutions that eventually could be used within FMN or NATO, meaning that it must be possible for the RTG to understand the inner

[*] https://www.anglova.net

workings of the protocols to enable any potential standardization in the future. In addition, the RTG focused on solutions for which software implementations were available or could be developed within the timeframe of IST-161. This resulted in a selection of data dissemination protocols that can be found in the public domain or within the NATO community. Also, the national research priorities of the IST-161 participants resulted in the selection of tested protocols and experiments. Many results and findings in the report are the product of discussing the outcomes of experiments, which requires proper understanding of the behavior of the experimentation environment, the protocols themselves, and the targeted operational usage.

1.4 Report Structure

The Anglova scenario that was used for the RTG's experiments is described in more detail in Section 2, as well as considerations for traffic models for future experiments are presented. Section 3 describes the architectural work of IST-161, including the group communication protocol taxonomy. Security considerations are the subject of Section 4. In Section 5, the experimentation environment is explained further, focusing on EMANE, the developed radio models, and the test harness that was used for data generation and measurements. The virtual private network (VPN) setup that has been used for experiments at Coalition Warrior Interoperability eXercise (CWIX) is also presented in Section 5. Section 6 focuses on the protocol developments that have been done by participants of the RTG, including the Generic Data Exchange Mechanism (GDEM), improvements to DisService, and additions to NDN, as well as several other developments. In Section 7, the RTG's experiments and results are presented, including references to the related scientific publications. Finally, Section 8 formulates the conclusions and recommendations from IST-161.

2. Scenario

For its activities, the IST-161 RTG used the Anglova scenario that was developed in IST-124 (NATO Science and Technology Organization 2019) with a specific focus on a mechanized infantry battlegroup in Vignette 2. During the timeframe of the IST-161 group, the following topics were considered from the viewpoint of group communications:

- Roles and platform types of the nodes in the Anglova scenario
- Radio deployment
- Traffic modelling
- Jamming scenario

The ground vehicles trajectories in Anglova Vignette 2 are originally one of several Swedish scenario concepts for a mechanized battalion derived in 2000 with a time horizon for 2010. The focus was on high mobility and capability of focusing force both in time and space. The trajectories were developed by tactical experts modelling realistic detailed movements in the terrain. Upon being included in Anglova for IST-124, the Swedish scenario and the original path losses were moved to Norway as the basis for an international scenario.

2.1 Roles and Platforms in the Anglova Scenario

A detailed description of the nodes in Vignette 2 of the Anglova scenario has been provided by the Swedish Defence Research Agency (FOI). The Anglova battlegroup is visualized in Figure 1. The battalion consists of two tank companies (Companies 1 and 2), two mechanized infantry companies (Companies 3 and 4), one staff and mortar company (Company 5), and one logistics company (Company 6). The battalion vehicles are listed in Table 1. The details on roles and vehicle types of the individual nodes can be found on the Anglova website (NATO 2019b).

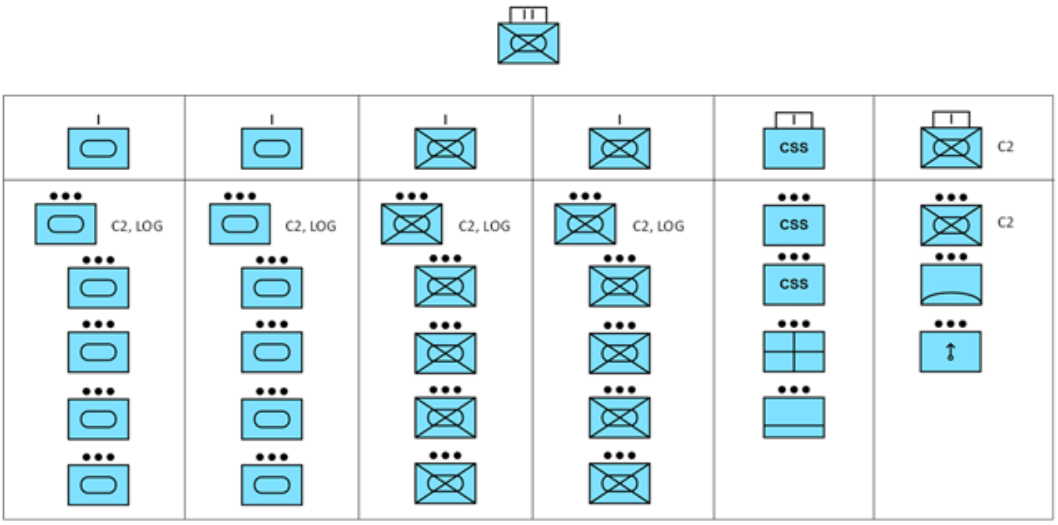


Figure 1. Anglova Vignette 2 task organization.

Table 1. Company vehicles.

Companies 1–4		
	Platoons	**Vehicles**
	1	Company Command Deputy Company Command Repair and Recovery (two vehicles) Bridge Layer (two vehicles) Logistics (six vehicles)
	2–5	Tank/Mechanized infantry (three vehicles)
Company 5		
	Platoon	**Vehicles**
	1	Battalion Command Deputy Battalion Command Battalion Headquarters (two vehicles) Radio Gateway (three vehicles) UAV Reconnaissance (three vehicles)
	2	Air Defense (three vehicles)
	3	Armored Mortar (four vehicles) Mortar Ammunition (four vehicles)
Company 6		
	Platoons	**Vehicles**
	1 and 2	Logistics (10 vehicles)
	3	Medical Transport (four vehicles)
	4	Food Service and Logistics (15 vehicles)

The ground vehicles trajectories of the companies for the Anglova Vignette 2 are shown in Figure 2. The vehicles move on roads in the beginning but are spread out on smaller paths and terrain in the end of the vignette.

Figure 2. Trajectories of the ground vehicles in Anglova Vignette 2.

2.2 Anglova Radio Networks

The Anglova network architecture comprises both very high frequency (VHF) narrowband waveform (NBWF) and ultra high frequency (UHF) wideband waveform (WBWF) networks. The UHF WBWF networks are used mainly for battlefield management systems (BMS) data, BFT, and SD; whereas the VHF NBFW networks are used mainly for voice, group calls, and push-to-talk (PTT). Two examples of network plans are provided and described in this section. These plans comprise of all nodes; nevertheless, a smaller unit consisting of just a subset of the nodes can be cut out. In our experiments we particularly considered the four armored/mechanized companies and their UHF WBWF networks.

Two types of architecture have been considered. In the first flat architecture, all nodes belong to the same WB UHF network (network plan 1; Figure 3). In our experiments, we have considered flat networks of 24, 48, 72 and 96 nodes (i.e., in the last case, all four companies of 96 nodes form a flat WB UHF network [radio network A]). The second considered architecture is heterogeneous (network plan 2; Figure 4). It consists of 4 WB UHF radio networks of 24 nodes within each company (radio networks A, B, C and D) and a WB UHF battalion radio network of 38 nodes that contain 4 vehicles from each company and all platforms from the command company (Company 5; radio network E). This latter WB UHF network E can be used to exchange information between companies. The command company (Company 5) has not been used in the experiments of IST-161. A detailed overview of voice and data radio networks can be found in Appendix A.

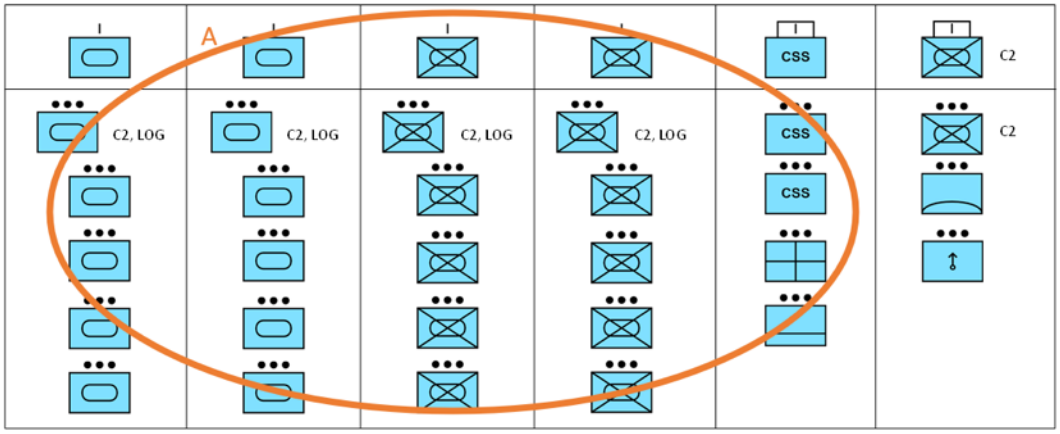


Figure 3. Anglova network plan 1, schematic view.

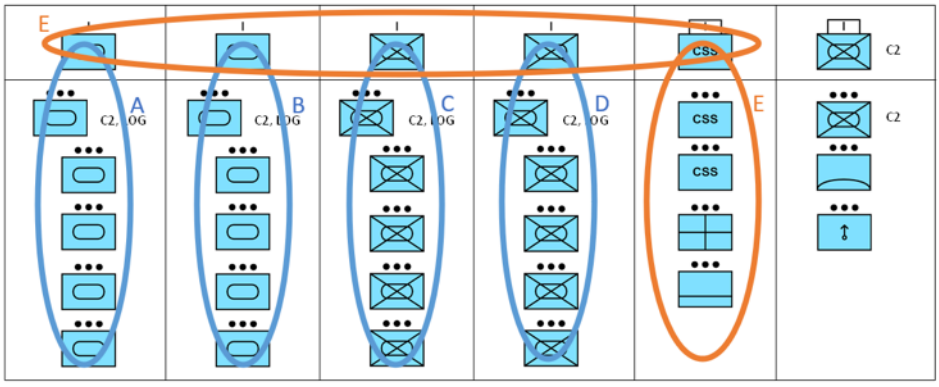


Figure 4. Anglova network plan 2, schematic view.

The four nodes per fighting company that are part of UHF network E and their own company UHF network can act as routing platforms or gateways to exchange data between the different radio networks. Only two nodes per company were chosen to be gateways in the IST-161 experiments. As such, these nodes have two

interfaces—one to the company network and one to the overlay network—and can send packets between the radio networks.

2.3 Connectivity

The connectivity in the Anglova scenario depends on the radio network technology that is used. This report only considers the heterogeneous architecture and focuses on the last part of the scenario that contains the most mobility between nodes (i.e., from 5000 to 7000 s). Precomputed synchronized cooperative broadcast (SCB) topologies can be generated and used to calculate the connectivity for SCB waveforms (see Sections 5.1.5 and 7 for further SCB descriptions and experiment results). The theoretical connectivity of a radio network can then be calculated as the fraction of node pairs that are connected via one or more hops in the SCB waveform. In other words, "connected" means they are reachable inside the multi-hop radio network. Therefore, the connectivity depends on the distance between nodes, the terrain, and the capabilities of the waveform to relay between nodes.

The movement patterns of the four companies in the selected time-slice of the scenario are shown in Figure 5. The connectivity of the four-company overlay network for Anglova Vignette 2 is shown in Figure 6. The connectivity is lower in the beginning and end, and better within the companies. Without any interferences or jamming, the connectivity of the company networks for the four companies at the same scenario time is one (not shown).

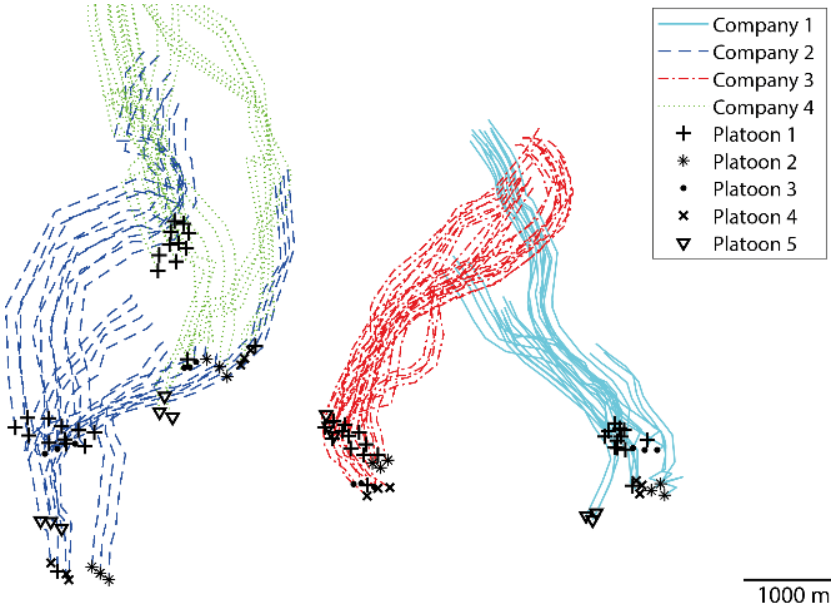


Figure 5. Movements of the four companies.

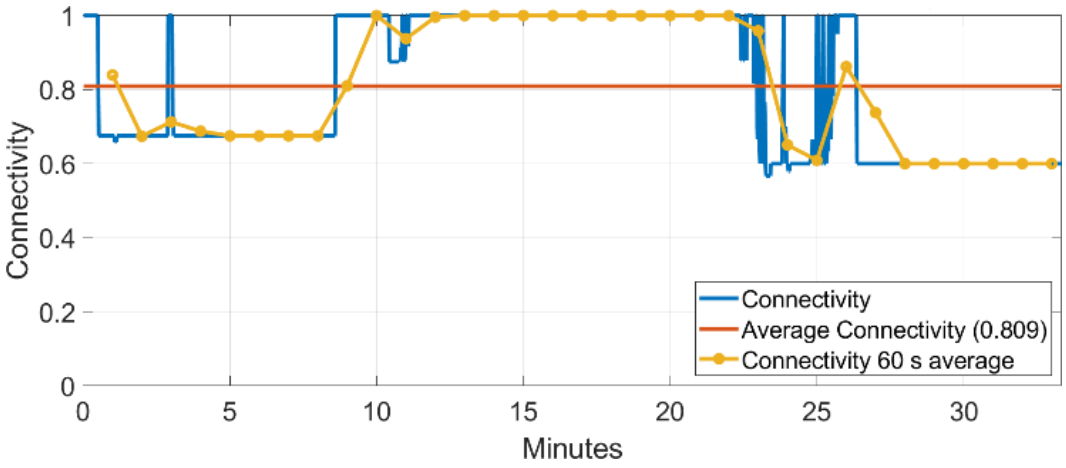


Figure 6. Connectivity of the overlay data network network (Company 5; radio network E) of network plan 2 for Anglova Vignette 2.

2.4 Traffic Modeling

When running an experiment using Anglova, it is important to model user traffic in the network that serves the purpose of the experiment. One can take two viewpoints when designing the traffic model:

1) Use "scientific" traffic models that are artificially created for the sole purpose of analyzing certain behavior of the protocols under test (e.g., stress-testing).
2) Use a realistic future traffic scenario based on real data traffic logs from the exercise or based on operational Information Exchange Requirements (IERs).

The IST-161 RTG used a combination of these viewpoints. A baseline scenario was developed based on the experience of group members with military tests/exercises. This scenario contained BFT messages, reports, and SD. This traffic model has been the basis for the IST-161 RTG's experimental work.

To understand how protocols and middleware work, a simple traffic model is best suited. For the baseline study, BFT messages were used as test traffic. The message size was a fixed 128 bytes with a sending interval between 5 and 10 s. Every node generated these messages and every node received these, so their delivery statistics are a good indicator of a protocol's performance in the entire network. The BFT traffic was included in all experiments.

The second traffic type used was an emulated sensor, like a camera. It generated a message every 3 min with a size of 128,000 bytes (i.e., 1000 times larger than the BFT message) and this message was sent only to a small subset of other nodes.

A larger message type used in some experiments was a document (e.g., report) type. For example, this document would correspond to an extensive recognized ground picture or other information sent from headquarters to lower echelon leads. The chosen size was four times that of the SD (512,000 bytes) and it was sent to an even smaller number of nodes every 6 min. These three traffic patterns are summarized in Table 2.

Table 2. Overview of traffic patterns used in the IST-161 experiments.

Traffic type	Size (bytes)	Frequency	Transmission pattern
BFT	128	Every 5–10 s	Generated by all nodes to all nodes
SD	128,000	Every 3 min	Generated by one node to a subset of nodes
Document/report	512,000	Every 6 min	Generated by one node to a small subset of nodes

The aforementioned test traffic enables straightforward analysis by observing message delivery rates and their delays. However, to understand further implications such as applying message priorities, more complex and closer-to-realistic traffic models are required. Additionally, the group found that the traffic model should be extended for future testing. Several inputs have been collected for this, but more are needed in collaboration with the operational community.

There is not a single "right" tactical group communication traffic model; the traffic patterns are widely different depending on the operation. A ground reconnaissance will have different communication patterns than medical evacuation via a helicopter towards navy ships that are far from the coastline. As stated previously, this work focuses on mechanized ground troops according to Anglova Vignette 2. The Anglova scenario has already been widely used in scientific publications and other work and has been referenced in 68 publications as of 2022 (NATO Science and Technology Organization 2019).

The quality of service (QoS) requirements development work done in IST-124 was selected as a basis for traffic model development. The following basic information is needed to evaluate system performance, traffic demand, traffic patterns, and delivery:

1) Traffic demand indicated with some combination of total traffic volume, packet count, packet size, packet intervals, or traffic event duration. Traffic demand also changes over time. For example, in Anglova Vignette 2, there could be some traffic for coordination at the start of the vignette, less traffic when battalion is moving along road, and another increase of traffic when the end of the deployment is reached.

2) Traffic patterns include one-to-one, one-to-many, many-to-one, and many-to-many. The "many" can include only some nodes within a company, all members of a company, or even larger group of nodes.
3) Traffic delivery characteristics can be indicated with maximum allowed packet (or message) loss, typical and maximum delay, and minimum bandwidth needed.
4) Value of information in received packets is more complex to assess, but the value of different messages can be different, even within the same traffic type. There is more value for a BFT information message from a fast-moving vehicle from which the latest update is 15 min old than from a stationary unit whose location was updated only seconds ago. IST-161 contributed the concept of Value of Information as a Tech Watch Card (Suri et al. 2019a).

The packet loss and maximum delay are interconnected with the observation that if information does not arrive before it is to be consumed, it can be considered lost. In the IST-161 experiments, BFT latency thresholds of 5 and 10 s in addition to ever-delivered message counts were used to collect message delivery statistics.

Table 3, adapted from Table I-1 of the IST-124 report (NATO Science and Technology Organization 2019) (see Table 3) describes different services with consideration for bandwidth usage, packet loss, and delay tolerance.

Table 3. Different services and requirements for bandwidth, packet loss, and delay.

Service	Bandwidth	Loss	Delay
Network control	-	-	-
Voice	+	0	+
Video	++	0	0
Situational awareness	-	-	- (+)
Formal messages, real time	-	0	0 (++)
Formal messages, non-real time	+	-	--
Informal messaging	++	-	--
Mass data, non-real time	++	-	--
Internet and remote access	++	-	-
Database queries and synchronized messages	-	+	-

Key	[bits/s]		
--	< 100	20,00 %	30 s
-	1000	5,00 %	5 s
0	. . .	1,00 %	1 s
+	16k	0,10 %	<250 ms
++	> 200k	0,01 %	<50 ms

Table 3 has been further developed in IST-161 using two different approaches. The first one was based on communication structure used in exercises between vehicles. Traffic was classified into the following five different types:

- BFT to be sent via multicast from all vehicles to every other vehicle.
- Text-based messaging using either chat functionality (unicast) or group messaging (multicast).
- C2 or command/status messages sent from the company lead vehicle to other vehicles.
- Group voice messaging (multicast) sent from a vehicle to a subset of vehicles. There would be different voice groups to which each vehicle belongs.
- Streaming video sent via multicast from a vehicle.

These traffic types were collected to create the IERs in Table 4. The table is grouped into three sections: The first involves operational elements, the second specifies the information exchanged, and the third provides the typical information exchange attributes. The Information Exchange ID in the first row corresponds to the five traffic types listed above. Table 4's values were obtained by an IST-161 participant; they have not been validated with a wider operational community.

Table 4. Possible company Information Exchange Requirements.

Information	Exchange ID	1 (BFT)	2.1 (single-user chat)	2.2 (group chat)	3 (C2)	4 (Voice)	5 (Video stream)
Operational elements involved	**Information producing node**	Vehicle	Vehicle	Vehicle	Lead vehicle	Vehicle	Vehicle
	Information consuming node(s)	All vehicles	Vehicle	Vehicles	Vehicles	Vehicles	Vehicles
	Communication form (uni-, multi- or broadcast)	Multicast	Unicast	Multicast	Multicast	Multicast	Multicast
Description of information	**Type of information** (e.g., video, voice, chat)	BFT (GPS data)	Chat and messaging (1:1)	Chat and messaging (1:n)	C2 command / status messages	Voice (1:n)	Video (1:n)
	Transport characteristics	Messages/ reliable	Message / reliable	Message / reliable	Messages / reliable	stream / best effort	stream / best effort
Typical required information exchange attributes	**Latency in seconds**	1 s	1 s	1 s	1 s	100 ms	100 ms
	Quantity message: bytes **Stream**: avg payload size in bytes	1 kbit	several kbit	several kbit	several kbit	1-8 kbit (codec dep)	Hundreds of kbits/s (codec dep)
	Update rate message: repetition rate in packets per second (pps) **Stream**: packet rate in pps	0,1-1 pps	single event	single event	10 pps	TBD	TBD
	Data rate in kbit/s	2.5			11.4		

Table 4 gives a basis for creating a traffic mix with traffic generators within one company. The data rate is based on a company size of 24 vehicles, as in Anglova Vignette 2. The latencies are shorter than used in experiments as limits; e.g., in the IST-161 experiments, 5 and 10 s are used as cut-off time metrics for BFT. The 1-s requirement mentioned in Table 4 was not met, even in the absence of other traffic.

A way may exist to reserve fixed allocations for specific traffic depending on the radio technology used (e.g., by reserving some time slots for specific critical traffic). On the other hand, traffic control in IP networks takes place using different queuing disciplines that are not distributed in nature. In deployed networks, traffic treatment is defined by Standardization Agreement (STANAG) 4711 (NATO 2018a) that follows a Differentiated Services style per-hop behavior (RFC 4542

[Baker and Polk 2006], RFC 4594 [Babiarz et al. 2006], and RFC 5127 [Chan et al. 2008]). STANAG 4711 defines four treatment aggregates (TAs) as follows: real-time, near real-time, assured elastic, and elastic (Table 5) (NATO 2018a).

Table 5. TAs and their quality constraints, according to STANAG 4711 (NATO 2018a).

TA	Tolerance to loss	Tolerance to delay	Tolerance to jitter
Real-time	Very low	Very low	Low
Near real-time	Low	Low to medium	Medium
Assured elastic	Low	Medium	High
Elastic	Medium	Medium to high	High

If we consider the Company IERs from Table 4, we can see that all messaging services (BFT, chat, C2 messages) use assured elastic as they have a medium tolerance to delay. Voice uses real-time and video uses near real-time, as it has higher tolerance to delay.

2.5 Jamming Scenarios

To experiment with the impact of jamming on group communication protocols, jammers can be modelled in the Anglova scenario. A first experiment was executed in IST-161 with a simple jamming setup. This is described in Section 7.4. In the future, the Anglova scenario can be extended with more realistic scenarios.

3. Architecture Considerations

3.1 Purpose of Architecture for Group Communications

Different solutions and concepts for group communications provide functionality in different ways. The solutions do not only differ in protocol, they are also different in terms of architecture. Solutions may implement similar functionality in a different layer of the communications protocol stack, or they may be using a different reference model than the IP stack. The variation in approaches can be visualized conceptually by mapping them onto a layered communications model. Figure 7 shows how a traditional IP solution, an IP solution without IP routing, and a non-IP based solution could be mapped onto a layered communications model. DisService is an example of an IP approach without IP routing (Suri et al. 2010). Without running an IP routing protocol, NDN is an example of a non-IP approach (Wikipedia 2024) using the transmission mechanism as "face,"* rather than using an IP interface as face.

* "Face" is a term used by NDN to indicate the type of interface the NDN messages are send towards. For example, an IP interface or an Ethernet interface.

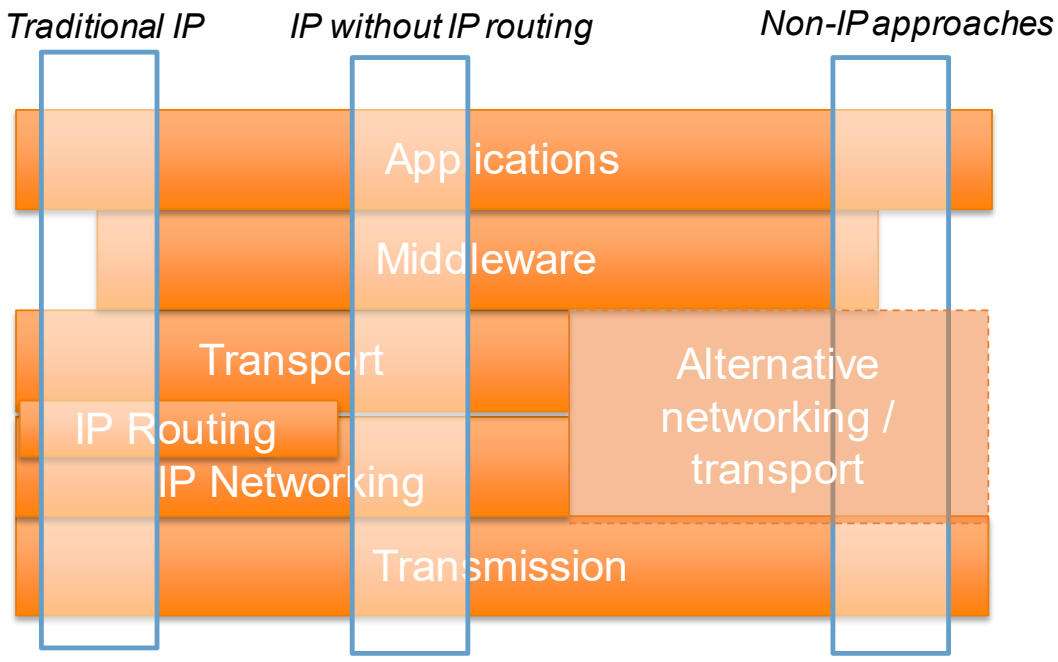


Figure 7. Solutions can conceptually map onto different layered communication models.

This section presents the following two types of frameworks that have been discussed in IST-161: comparing protocol stacks (technical layers) and framework for positioning a protocol in context of other protocols (taxonomy). The terms mentioned in the taxonomy can be used when comparing different group communication solutions.

3.2 Comparing Protocol Stacks

For fair comparisons of solutions, the IST-161 RTG tried to work out protocol stacks that would comprise a solution providing group communications. The initial motivation for this activity was the question of how to make a fair comparison between IP-based solutions and NDN-based solutions. We refer to the Joint Dismounted Soldier System (JDSS) (NATO 2017a) as an example of IP-based solutions. JDSS was selected by FMN in Spiral 4 for message exchange between units that reside in the mobile tactical edge. An IP-solution such as JDSS comprises more layers and functionality than the vanilla NDN solution the IST-161 group used, thus the NDN stack had to be enhanced with higher layer functionality (e.g., data synchronization) for a fair comparison. The examination of protocol stacks enabled the RTG to identify what functionality still needed to be included. This vanilla NDN protocol stack, including the lacking functionality, is shown in Figure 8.

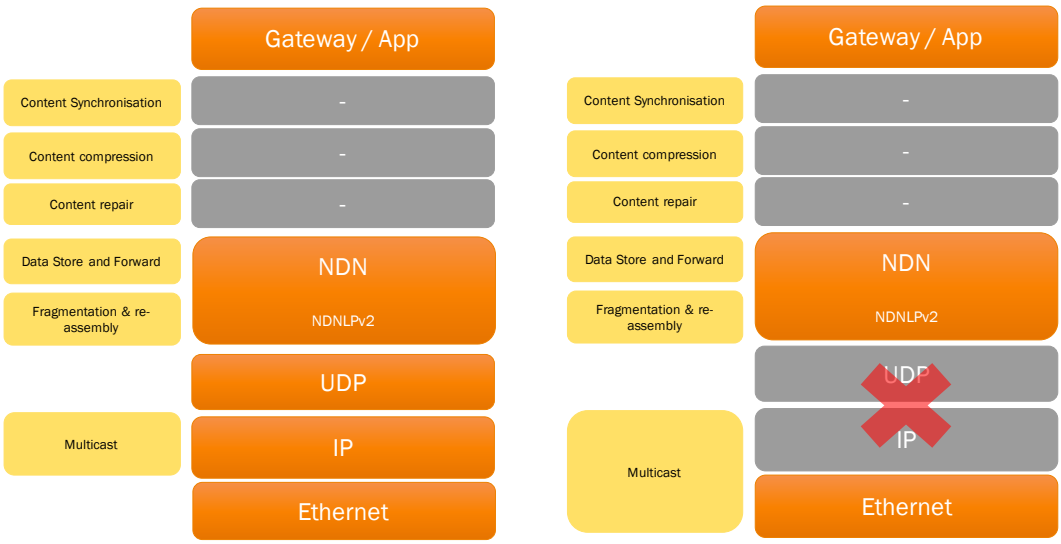


Figure 8. Protocol stacks of vanilla NDN using IP as face (left) or using Ethernet as face (right).

Another example using the protocol stack comparison is in analyzing differences and similarities between a JDSS-based solution and a Generic Data Exchange Mechanism (GDEM)-based solution. GDEM is an enhancement to JDSS that was developed by TNO. Figure 9 shows different transports that were used in experiments related to GDEM and JDSS, as well as the difference between the GDEM version (v)1 and v2 protocol stacks.

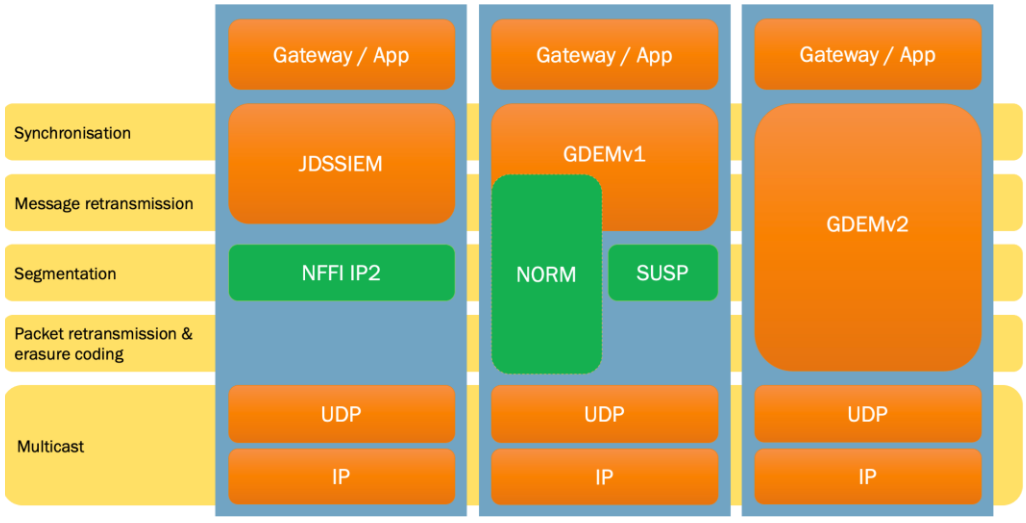


Figure 9. Protocol stacks for JDSS, GDEM v1, and GDEM v2 (Suri et al. 2010).

Analyzing these protocol stacks initiated a discussion in the RTG about the exact definition of the terms used for certain functionalities such as *synchronization*, *content repair*, and *packet retransmission* since they are not used consistently throughout literature and discussions among experts. The RTG concluded that

agreement on a protocol taxonomy would support similar discussions and solution comparisons in the future. This led to the creation of the group communications protocol taxonomy presented in Section 3.3.

3.3 Group Communications Protocol Taxonomy

During the experimental comparison of group communications solutions, the RTG encountered many different middleware/solutions that offer different functionality (as explained in Section 3.2). This makes it more difficult to discuss the performance differences and the underlying design choices that led to those performance differences. Thus, the IST-161 RTG developed a group communications protocol taxonomy that was published in Fronteddu (2022). The taxonomy is shown in Figure 10 and a brief description of terminology is provided in Table 6. This taxonomy can be used by future groups to research group communication protocols and properly compare them.

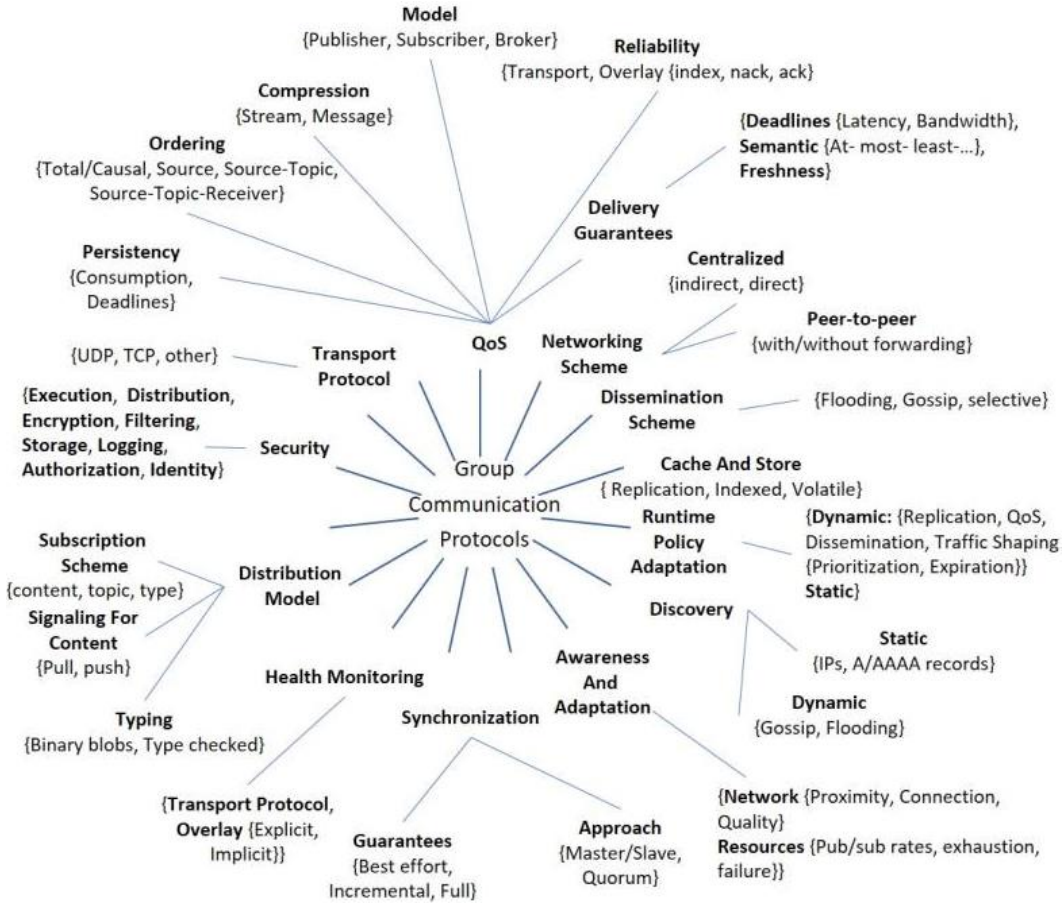


Figure 10. IST-161 group communications protocol taxonomy plot.

Table 6. Group communications protocol taxonomy definitions.

Functionality	Definition	Example mechanism
Network scheme	The network scheme describes how peers organize to exchange information. Typically, implementations can be defined as *centralized* or *decentralized* based on whether they rely on a middleman called a *broker* to communicate.	• Neural Autonomic Transport System (NATS): broker-based • GDEM: peer-to-peer
Reliability	Reliability describes the mechanisms that group communication solutions implement to ensure message delivery. This facet particularly describes whether solutions implement functionalities related to reliability on top of what is provided by transport protocols.	• Negative Acknowledgement (NACK) message retransmission or PAR2 data verification and repair • Kafka: index-based • Based on automatic retransmission of data, ACK, or NACK
Delivery guarantees	Delivery guarantees expand on the reliability, describing the constraints of the reliability mechanisms. For example, some solutions only guarantee the delivery of messages delivered after a node subscription, while others query for content generated before the subscription. Moreover, delivery guarantees also describes the limitations that can be specified by the user on message delivery. Typical examples are to set deadlines on the number of attempts or the bandwidth consumed to deliver a message.	• Data Distribution Service (DDS) bandwidth/latency/ retransmissions deadlines • Message Queuing Telemetry Transport (MQTT): at-most-once/at-least-once
Transport protocol/ dissemination/ store, carry and forward	This facet describes lower-level features of the group communication solutions, such as advantages that can be achieved by using multicast protocols or proactive dissemination of message fragments over multiple paths.	• Transmission Control Protocol (TCP)/User Datagram Protocol (UDP)/ Multicast/Unicast • Opportunistic caching • Fragmentation/reassembly • Single/multi-hop multicast/unicast
Runtime-adaptation/ awareness and adaptation	This facet describes the level of adaptation that solutions provide in response to user inputs or feedback from the network. Examples of these capabilities are the ability to switch subscription channels delivery guarantee at runtime without tearing down channels, or the implementation of algorithms that have networks of brokers reconfigure message routing in case of network failure.	• NATS: routing reconfiguration between brokers
Discovery	Discovery describes how peers find each other. For example, by advertising their presence to a statically configured centralized entity, or by implementing a multicast or broadcast-based discovery protocol.	• Static configuration/ broadcast

Table 6. Group communications protocol taxonomy definitions (continued).

Content synchronization	Content synchronization describes the algorithm that group communication solutions implement to ensure content synchronization across peers. For example, certain solutions implement algorithms that prevent the delivery of messages to the application unless messages that were previously generated have been delivered to all the intended recipients.	• JDSS-Information Exchange Mechanism (IEM): full sync
Monitoring	Monitoring describes the facilities that group communication solutions provide to monitor and log their status. Status detection can be performed actively or passively depending on whether extra messages are used to verify the status of a connection.	• Kafka peer disconnection for peers that drift too far
Distribution model	The distribution model facet describes whether a protocol is push- or pull-based. Push approaches have publishers decide when it is time to share new content, while pull approaches leave this choice to each subscriber.	• Kafka index pull-based mechanism • NATS push-based mechanism
Security	Security describes the level of facilities provided by group communication solutions that enable them to conduct message distribution securely. This facet assumes many forms, ranging from a mechanism to enforce limits to publications and open subscriptions to controlling access to information and routing of messages.	• Encryption of data-metadata • Encryption of monitoring information • Information routing/path traversal control • Publish/subscribe rate control
Other/QoS	This facet describes other functionalities that group communication solutions may provide, such as the ability to internally compress messages, enforce or relax ordering, and provide persistency of messages for a configurable amount of time.	• Disservice reliable ordered or unordered delivery guarantee. • Efficient XML Exchange (EXI)

4. Security Challenges and Solution Space

The primary challenges addressed by the IST-161 security working group were identifying the main security obstacles regarding future battlefield communications and identifying potential solutions for them. For the purposes of this report, *security* refers to the protection in confidentiality, integrity, and availability of information and communication.

The scope and boundaries/interfaces of tactical networks (as well as limitations faced within this scope), assets that need to be protected, and security objectives all need to be understood to provide appropriate security controls. This gives some hints about what the appropriate security controls are. This approach is used in the

IST-161 security working group and is consistent with NIST 800-18 (Swanson et al. 2006). The following sections describe the results of the RTG's work on security.

4.1 System Boundaries

The scope of this section is the communication and information systems (CIS) present in mobile tactical environments. Tactical edge networks normally consist of one or more interconnected radio networks. Radio networks are broadcast by default, and the predominant type of communication is group communication, either by one-hop broadcast or multi-hop multicast. The tactical systems can have connections to strategic networks (e.g., wired or beyond line-of-sight communication links).

Loss of equipment must be expected in this environment. Consequently, highly classified information should be avoided as much as possible. Depending on the radio system(s) in use, the channel capacity may be limited and pose long delays. The bit error rate is generally significantly higher in wireless networks than in fixed networks and the link quality and connectivity vary. Factors such as throughput, connectivity, delay, and number of hops to traverse all have implications for the applicability of different security solutions. Figure 11 shows generic challenges faced when designing communication systems for this environment.

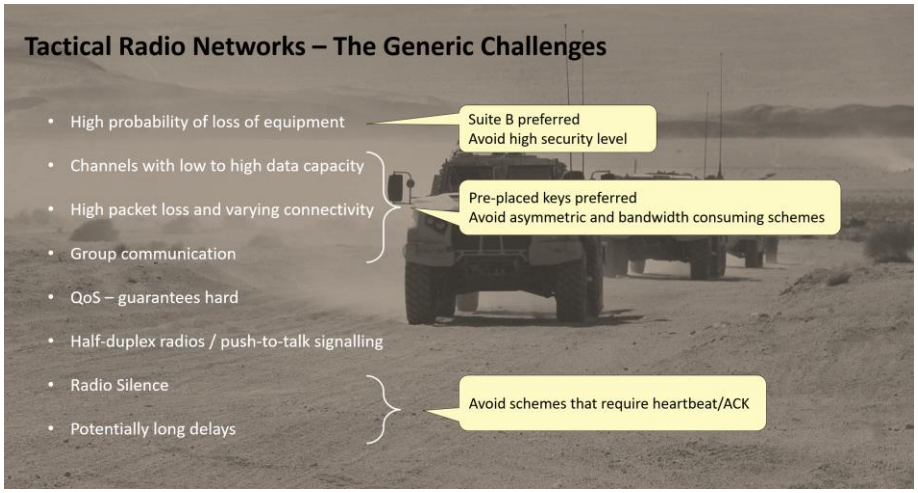


Figure 11. Characteristics of tactical radio networks and their security implications.

4.2 Assets

It is important to keep in mind what assets need protection when designing an appropriate security solution. We have identified the following categories of assets, inspired by Farsund et al. (2017):

- *User payload* is data that the end-users either generate or consume. This may include data from various types of applications as well as voice. Confidentiality and integrity are essential (i.e., the user payload needs protection against unauthorized access and modification).
- *User metadata* is information about who the user is, where they are, communication peers, and communication patterns. It should be possible to conceal user metadata from unauthorized parties.
- *Network management data* is the information that is exchanged to monitor and configure the network. Integrity and authenticity are essential to prevent disturbance and disruption of the network service. Confidentiality protection may also be required.
- *Network control traffic and network metadata* are the information that needs to be exchanged to keep the network service running. It is sometimes referred to as *network signaling*. Network metadata is information on the network itself (e.g., information on the network elements, addresses, communication patterns, movement patterns, and the importance of specific nodes). Integrity and authenticity are important to prevent disturbance and disruption of the network service. Confidentiality protection also may be required to conceal network control traffic and metadata from unauthorized parties.
- *Network availability* includes the protection of the availability of above assets and the availability of network components to keep the network service running. Network availability must be maintained regardless of the actual user payload on the network.

4.3 Security Objectives and Challenges

The current security ecosystem focuses on secure information sharing between users of CIS. This security primarily relies on the following aspects:

- Classifying the information according to set guidelines (i.e., setting the NATO classification between RESTRICTED to COSMIC TOP SECRET according to NATO policies). Typically, information is classified upon creation but may be declassified afterwards.
- Declassifying short-lived information in tactical systems with dynamic deployment configuration based on specific mission requirement or after well-defined processing (e.g., some subpart of the information can be declassified in some circumstances).
- Handling (i.e., processing and storing) this information in specific, accredited, systems within a security domain that have taken classification-appropriate security measures to minimize the risk of breaching confidentiality of the information.

- Sharing this information with only those with a relevant security clearance and an operational need to access it (need-to-know). The need-to-know can vary for each user in tactical systems depending on the evolution of a mission or a deployment.

A coarse sketch of the interplay between these aspects is illustrated in Figure 12. Naturally, a higher security classification poses stronger security requirements on the infrastructure where information is hosted. It also ensures that fewer people gain access to the information (i.e., there needs to be a stronger need-to-know). This need-to-know is typically enforced in multiple aspects of access to the CIS. The first is gaining access to the systems that are part of the information domain. This includes both getting authorization to access particular systems on a command post and having a radio setup with a specific time-bound cryptographic key to be able to communicate over a radio channel. The second aspect is gaining proper security authorization after having access to the information domain. There are various ways in which this authorization can be enforced, briefly summarized as different security modes* that are defined in the following paragraphs.

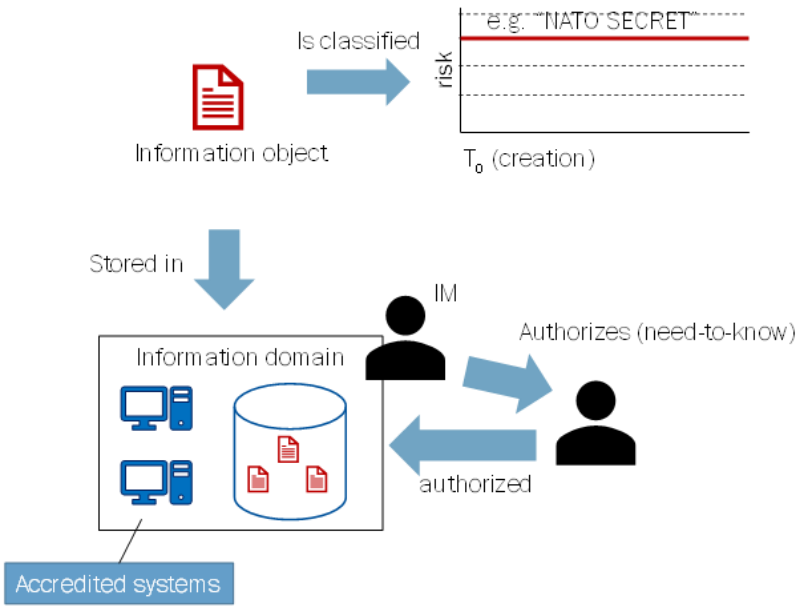


Figure 12. High-level illustration of interplay between information classification, system-high environment, and need-to-know.

Dedicated is a mode of operation in which all individuals with access to the CIS are cleared and authorized for the highest classification level of information handled within the CIS, and with a common need-to-know for all the information

* The NATO directive on CIS security (AC/35-D/2004-REV3) requests CIS designers to define the security mode of operation of the system (NATO Security Committee 2013).

handled within the CIS. Figure 13 shows a possible scenario of the application of access enforcement in a dedicated mode of operation. All users share the same multicast group with the same traffic encryption key (TEK) protecting the information.

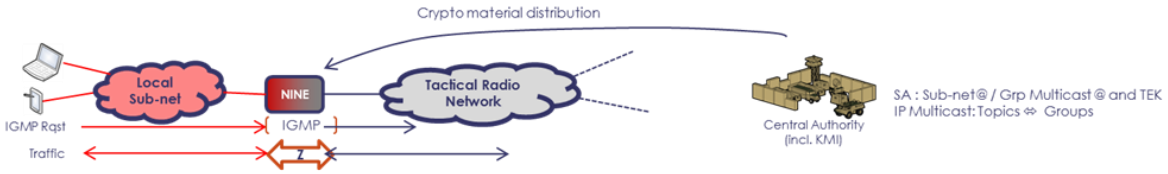


Figure 13. Dedicated mode of operation.

System high is a mode of operation in which all individuals with access to the CIS are cleared and authorized for the highest classification level of information handled within the CIS, but not all share the same need-to-know. Users with the same need-to-know can access the information using a common TEK. The access is locally controlled. A possible application of access enforcement is visualized in Figure 14. In this case, all users can access the multicast distribution group (protected with the same TEK). However, access requests are controlled locally.

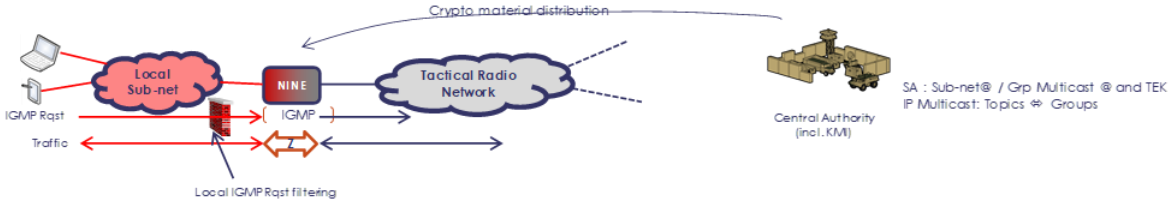


Figure 14. System high mode of operation.

Compartmented is a mode of operation in which all individuals with access to the CIS are cleared and authorized for the highest classification level of information handled within the CIS, but not all share the same need-to-know. Users with the same need-to-know can access the information using individual TEKs, and the access is centrally controlled. Figure 15 depicts a potential method of access enforcement in such a mode of operation. Security associations (SAs) are created for each topic/multicast group and user access is enforced by encryption using different TEKs for each data source.

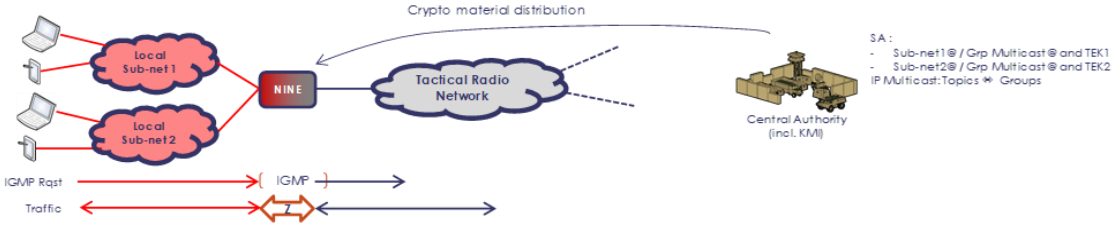


Figure 15. Compartmented mode of operation.

Multilevel is a mode of operation in which not all individuals with access to the CIS are cleared to the highest classification level of information handled within the CIS. Users may access the data for which they have a proper security clearance, authorization, access approval, and a valid need-to-know. The CIS includes information at multiple classification levels and the different classification levels are separated. Figure 16 shows such a mode of operation. Each level of information is protected by dedicated crypto devices with user access controlled locally.

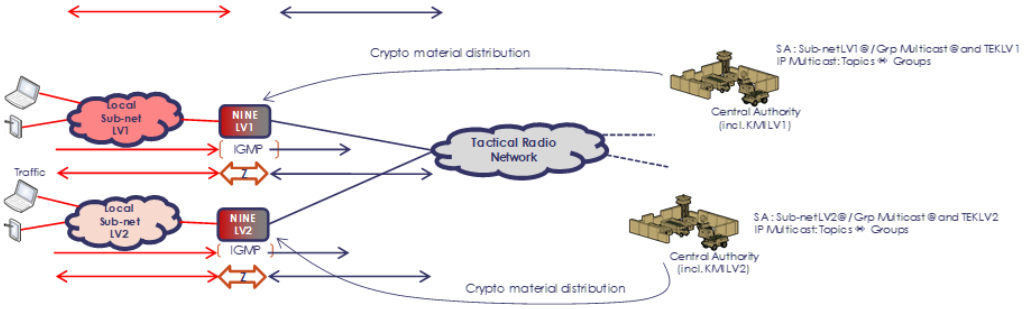


Figure 16. Multilevel mode of operation.

The NATO directive on CIS Security (NATO Security Committee 2013) that requests CIS designers to define the security mode of operation of the system is particularly relevant for group and information-centric communications. The more complex the system security is, the harder it is to get a security accreditation (i.e., the more data sharing is "open" or "flexible," the more complex the security becomes resulting in a more stringent and difficult accreditation process—and, from a commercial point of view, time-to-market). In part due to the accreditation process—but also history, culture, and other reasons—a typical solution relies on systems in system-high modes where a particular set of classified information is stored, exchanged, and processed.

Information is (and is expected to increasingly be) a facilitator for military operations. Voice and sharing information (e.g., position data, alarms, and other data between the coalition parties) are key factors. Due to the sensitive nature of these various types of data, they must be secured to ensure that they do not fall into the wrong hands, and their integrity and authenticity are preserved. Securing information is a requirement for safe and successful military operations.

Threats to security in mobile military heterogeneous networks include jamming, tampering, eavesdropping, loss of equipment, improper implementation of cryptographic protection, vulnerable protocols, wrong use of equipment, complex networks, and lack of operator education. With more actors and a growing number of nodes in the heterogeneous network, the traditional threat model with active and passive outsiders needs to be extended also to include insiders. Threats from the inside can be both deliberate and accidental.

Future operations will be more dynamic and require tighter integration between coalition partners at the tactical edge. There will be a stronger operational desire to exchange information in a dynamic and more ad-hoc manner. While it is not impossible to perform future military operations under the existing security paradigm, it is highly static, planning-based, and thus not as well-suited for such dynamic operations. For example, information can only be created and processed on specific systems and networks that are approved for its classification. Securely exchanging information with other coalition or civilian partners that do not employ such systems is not possible. Furthermore, secure radio communication involves shared security keys that are often distributed through the chain-of-command and manually loaded into radios. This is not well-suited for operations when new interactions may be required on the fly.

4.4 Security Solution Space

Efficient and safe future military operations may be facilitated by changing the employed security concepts. Currently used security paradigms, such as system-high security mode and relying on predefined authorized interactions, limit the flexibility required in the modern battlefield. Security concepts that lend themselves to better facilitate operational demand should be researched and considered further. The following sections describe several security concepts that may be part of the security solution space.

4.4.1 Separating Security of Information and Infrastructure

Security in mobile military networks typically shows a coupling between infrastructure, information, and security. Security controls are applied at the infrastructure level (e.g., the communication key that is loaded in a radio) and no or limited additional controls are applied on the information level. This system-high security mode allows for stringent control over the security but does not facilitate flexibility well.

By separating the security on infrastructure and information, as depicted in Figure 17, it becomes possible to introduce flexibility. Security measures taken on the infrastructure level will be related to availability of the network and securing networking (e.g., routing) information. Securing the payload (i.e., information) will be done through its own set of cryptographic means. Such an approach already shows military applications (e.g., protected core networking).

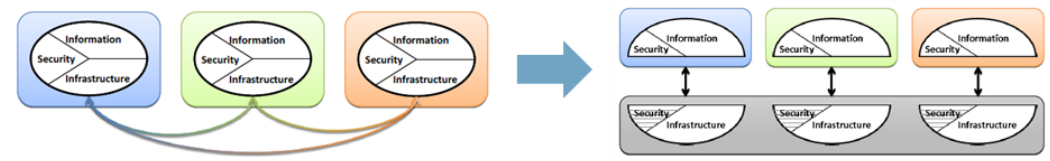


Figure 17. Separating information, infrastructure, and security (adapted from Boonstra et al. [2012]).

Figure 18 illustrates a potential application for tactical mobile networks. The left part of the figure shows the current situation where there is a tight coupling between infrastructure (network), information (voice), and security, depicted as different silos with their own security locks. The right part of the figure shows the separation model. Communities of Interest (CoIs; voice channels depicted) that share the same security requirements in terms of networking (e.g., availability requirements) and share a common network for data transport. Security controls are applied to this network layer, primarily in the context of Transmission Security and Network Security. Additional controls are placed on the information layer for securing information. For example, one cryptographic key may be required for accessing the network, and another additional key is required for accessing the CoIs. In other words, access to the network does not directly give access to the CoIs.

This approach outlines two potential benefits. First, security controls can be better selected to the security goals and threats faced on each layer (i.e., setting appropriate controls for infrastructure vs information). Moreover, a more shared infrastructure has potential for improved connectivity and quality of service. Second, a key needs to be physically loaded into a communication device in the current situation. In the concept, this physical loading is still required for gaining access to the infrastructure, but it may be possible to then use this infrastructure connection for gaining access to different CoIs through the network. Of course, security requirements for exchanging such keys must still be met. This is a topic of research and will be described further in Section 4.4.3.

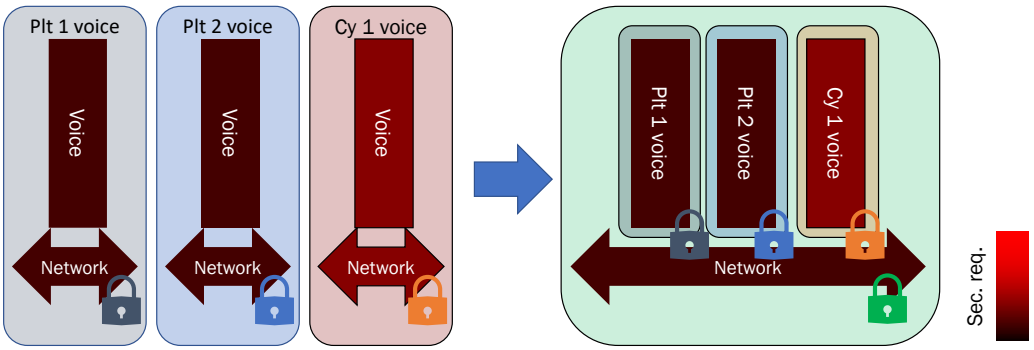


Figure 18. Separating the security of information and infrastructure for tactical networks (right) vs. tightly coupling networks and services in secure silos (left).

4.4.2 Multilevel Security (MLS)

A more flexible infrastructure is created through the separation of infrastructure, information, and security. Different systems may share a common network and preserve the information from compromise by controlling the data path with end-to-end protection. If this paradigm is well-defined and the baseline for many NATO system architectures, further flexibility may be obtained by allowing different systems to process and exchange information of different classifications.

Two security models that can conceptually be employed for this purpose are MLS or multiple independent levels of security (MILS). Efficient and secure exchange of information across different classification levels remains a topic for further studies. While research has been going on for several decades, generally adopted standards are still lacking. The solutions also tend to be expensive or useful only for dedicated problems. Progress made within virtualization technology has enabled solutions based on MILS. The need for secure exchange of information between different security levels persists, and MILS/MLS are still subject to more research.

4.4.3 Key Management

Cryptography is one of the primary security control mechanisms involved in the protection of information. Major security features are the key infrastructure (protection coverage over the sensitive information), distribution (giving access to protected information), and management (forward secrecy, key rollover, deny).

Key management is often mentioned as a management plan of security in parallel to user and signaling plans that effectively manage operation of the security controls and directly impacts the system availability and efficiency from the operational point of view. Key management can be considered the “sinews of war” for all protected systems and its mode of operation (e.g., predefined, dynamic) and usually is correlated with the use of the systems (e.g., planning, flexibility). It is part of a wider management process that includes the management of many other parameters (e.g., frequency planning, networks addressing plans and identifiers) and does not limit configuration data to the only cryptographic parameters.

In tactical networks, most of the communication groups are set up predefined and static. Communications and credentials are planned and injected manually. However, the need for on-demand or short-lived groups increases. This is the case in the current security ecosystem and will only expedite when moving towards a more flexible infrastructure. Moreover, moving towards more fine-grain security paradigms, such as data-centric security (DCS; see Section 4.4.4), may put further requirements on the key management model. A key management solution is required that better enables unplanned or short-lived group communications. It

should be compatible with low bandwidth for constrained networks and simplify the steps involved for the user.

Solutions can be found in both operational policy and new technology that may be employed separately or in tandem. From a policy point of view, it could be considered to employ different key-management organizations (e.g., pushing responsibilities to a lower level in the chain of command to enable faster key distribution). From a technology point of view, over-the-air (OTA) rekeying or other methods for accelerating the distribution of cryptographic keys through existing networks may be considered. Moving towards the type of network described in Section 4.1 may also have an impact on the key management employed. Access to different CoIs may be achieved through online, rather than physical, key distribution. Such concepts are not new, but adopting them as part of military networks has been challenging.

4.4.4 DCS

DCS is an approach to security in which individual information objects are protected. The concept of DCS has been around for some time, but there is still a long way to go to have a commonly adopted and applicable solution implemented. NATO has provided its vision on the adoption of DCS (NATO 2019a) detailing different maturity levels from level 1 (all new data is labelled) to 3 (data is cryptographically protected).

Topics for future research include:

- How will different objects be protected (e.g., with individual cryptographic keys), and how does this translate to the constraints faced by tactical networks?
- How well should data labels be protected, and what mechanism should facilitate this?
- To what extent can DCS facilitate concepts (e.g., end-to-end encryption) be used as a driver to step out of the existing siloed nature of systems?

One potential facilitator of moving towards DCS level 3 is attribute-based encryption (ABE) (Tomida et al. 2021). ABE makes it possible to encrypt an object with one key and decrypt it with different keys by recipients that possess the right attributes. All users with the right security attributes (e.g., their role in an operation) can use their different secret key to decrypt the message. Another important feature is that if more parties collaborate and they join their keys, their combined key does not become more powerful than their original separate keys.

ABE represents a possible way to implement a DCS model; it fits well for a federated network environment where sharing of information is vital, while at the

same time some of the information should not be shared with everyone. The participants may be authorized to gain access to different information objects. Each object can only be deciphered by the group of users with the right attributes. If this was to be solved using standard encryption techniques, the set of potential keys grows exponentially with the number of decryption requirements (called *attributes*) since any combination of attributes requires its own unique key. The results in high memory consumption as well as a slow system since finding the correct key is time-consuming.

An efficient implementation of ABE would solve both these problems. ABE was first introduced by Sahai and Waters (2005) and has since gained a lot of attention from the academic community with new and improved schemes being proposed at steady intervals. The focus mainly has been on reducing the size of the secret keys and improved security proofs (Tsabary 2019; Gong and Wee 2020; Lin and Luo 2020). However, not only the academic community has shown interest in ABE. In 2018, the European Telecommunication Standards Institute published a standard for ABE encryption in highly distributed systems (Antipolis 2018), stating use cases together with an algorithm specification. ETSI also showed that the key size grows linearly with the number of attributes. Although the size is not as bad as the exponential growth required without ABE, the ABE keys still have potential to become impractically large. Zickau et al. (2016) conducted an applied study implementing currently available ABE schemes. It gives an overview of the total expected execution time on an i7-4710HQ at 3.40 GHz as well as a Raspberry Pi 2 at 900 MHz. The former had a median total execution time of 7,651 s for 100 iterations. In other words, the schemes are still quite slow for practical uses. Although this study is four years old at the time of this writing, there is little to suggest that these times have since improved dramatically.

The conclusion is that ABE may solve many problems in current schemes. It is a candidate solution for the implementation of a DCS model, reducing the number of keys needed and increasing control of the information distribution. Research is still being conducted into efficient implementations of ABE, keeping key sizes small and the key derivation schemes less time consuming. It is important to gain understanding and improve current proposed schemes in respect to usability.

4.4.5 Examples of Applying NATO Standards to Achieve Secure Group Communications

In this section, examples are given of applying the NATO crypto standards "Network and Information Infrastructure Network Encryptor" (NINE) (NATO 2018b) and "Secure Tactical Communications Interoperability Specification" (STaC-IS) (NATO 2020) for secure group communications.

4.4.5.1 Group and Information-centric Communications Using NINE

NINE provides inline encryption based on IP and SAs as defined in an SA Database (SAD) and a Security Policy Database. Additional network services encompass Internet Group Management Protocol, Robust Header Compression, and others. Key management is usually supported by a key management infrastructure (KMI) and additional key management services. Figure 19 shows the architecture considered. NINE provides protection of classified enclaves to support system security accreditations. Users are cleared to access the CIS within the protected enclaves. There is a need-to-know separation between users inside the same CIS/enclaves. Other features include packet integrity service (GCM-96/128) with anti-replay windows that take care of desequencing and Delay/Disruption-Tolerant Networking functions.

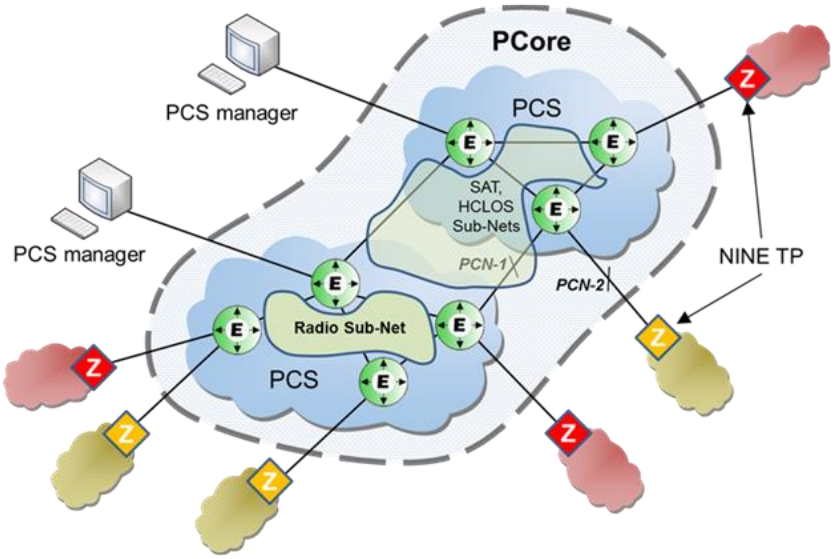


Figure 19. Assumed architecture of NINE for group communications.

The underlying topologies of the group communications can take multiple forms depending on the following two main criteria:

- The use of a data broker or not
  - Some systems are organized around a data broker (e.g., MQTT) that acts as an intermediate between the information publishers and the information consumers with a publish/subscribe mechanism. Brokers store and forward messages. The main drawback of this approach in conjunction with security placed at the IP level is that the broker will access all plaintext (PT) messages, thus breaking the need-to-know restriction.
  - Other publish/subscribe systems work without data brokers (e.g., DDS), in which case they can rely on IP multicast distribution. This

approach can be made compatible more easily with IP-level security.

- Whether the distribution is topic- or content-oriented
    - If the publish/subscribe mechanism is topic-oriented, there is no need to process PT content or attributes to filter packets. Data consumers subscribe to topic-based distribution groups that can be materialized by ciphertext IP multicast distribution.
    - In case of content-based subscription, the broker needs to process PT content or attributes to distribute messages. This process would break the need-to-know restriction.

NINE could be used for group distribution in systems without data broker and with topic-based groups. In this system, NINE devices would be configured with predefined multicast SAs and security policies per topic. Discrimination could be made between publishing and subscribing with the use unidirectional SA.

4.4.5.2 Group and Information-centric Communications Using Secure Communications Interoperability Protocol (SCIP) 214.6 (STaC-IS)

SCIP 214.6 (STaC-IS) (NATO 2020) is a Communication Security solution used to protect broadcast communications in a radio network. It is optimized for legacy radio waveforms but can also be adapted to packet-based transmissions. Each transmission is unidirectional and realized in a PTT fashion beginning with the transmission of the unique cryptographic parameters used for the encryption of the data payload. The Crypto Sync field shown in Figure 20 contains these parameters. The Sync Verification field verifies the correctness of the cryptographic synchronization.

Narrowband Data Mode

	1 160	1 252	1 8	1 112		1 336
Bit Sync (96 bits minimum)	**PN Framing Sequence**	**Crypto Sync**	**Dummy Bits**	**Sync Verification** (1 frame)	**Data** (variable)	**EOM** (3 frames)

Figure 20. Black-side sequence of transmission for STaC-IS narrowband data mode.

The STaC-IS specification does not specify the key management policies in the network, and the cryptographic preamble of each transmission lacks fields to indicate either the group for which the transmission is intended or the key used to encrypt it. Thus, this protocol is not well-suited for packet-based data transmissions. However, it is possible to investigate adaptations to achieve packet-based IP communications without changing the current specification. An update of the standard to accommodate this is under development.

Each data topic uses different encryption keys. The distribution of these keys to the participants of each topic is outside the scope of STaC-IS. Nodes without the right key for a topic cannot access the encrypted data, but they can relay the information and, in some cases, adapt the transmission format (narrow band voice <=> wide band voice, or change data transmission rate).

The difficulty lies on the reception side to determine to which topic the transmission belongs. With the current version of the specification, it is possible to study the possibility to use the Process ID (PID) field in the Application IV field contained in the Crypto Sync field, as shown in Figure 21. This PID field is 4 bits long and each transmitter can assign a value at its convenience. Hence, it could be possible to assign a unique PID value to each data topic and to its key. However, this solution is limited to a maximum of 16 data topics in one network. In a dedicated trusted network, multiple topics can use the same key; the segregation between the topics uses only the PID value.

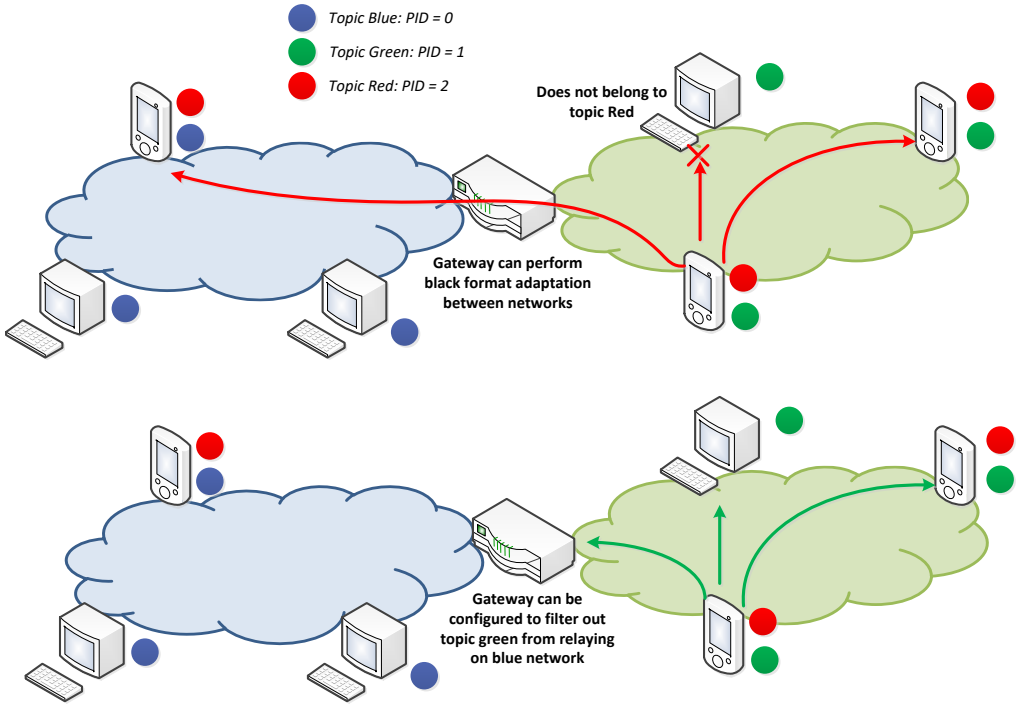


Figure 21. Data-centric communication with STaC-IS using PID value.

5. Emulation Environment

Section 5 provides an overview of the emulation environment and tools that were created or co-opted for the activities of the NATO IST-161 RTG. The scenarios and use cases deployed with the environment are described in Section 2 and details about the mobility patterns and experimentation methodologies for the Anglova scenario are described in Section 5.3. The IST-161 RTG has continued to use and

build upon the Anglova scenario that was initially developed by the IST-124 RTG using the EMANE (Navy Research Laboratory 2024). This section discusses alternate deployment models for the experimentation environment and provides some details on EMANE and the SCB radio model developed in the EMANE framework by the RTG to conduct its research efforts. Three deployment models for the emulation environment are discussed as follows: the U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory's (ARL's) dynamically allocated virtual clustering environment (DAVC), VMware ESXi, and the Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE) VPN Environment for Distributed Experimentation. Additional details about using these environments to conduct experiments are also covered, including the test platform used to orchestrate the multitude of group communications experiments conducted throughout the RTG.

5.1 Experimentation Environment and Tools

The Anglova scenario is accompanied by a set of tools that simplifies modification of the scenario and its use in an experimentation environment for collaborative research. The IST-161 RTG believes that the emulation-based approach to experimentation adopted by the RTG provides a good compromise and stepping-stone between simulations and using actual hardware (Suri et al. 2018). In a typical emulation-based experiment, the software components that are deployed in the experiment are ideally the same components that would be utilized on actual hardware and in the field. Therefore, there is no need to maintain multiple implementations. When changes need to be introduced, it is easier to evaluate them in an emulation environment with the actual software components prior to pushing those components out into the field.

EMANE (Navy Research Laboratory 2024) was the framework selected for the IST-161 experimentation. EMANE was selected for multiple reasons; the two primary technical reasons were scalability and flexibility, and other deciding factors included easy access and prior experience with some of the authors. EMANE is released as open source and is available for download on GitHub*, making it easy to be obtained and deployed.

The group has developed a deployable virtual machine (VM) that contains all the tools necessary to instantiate and run the Anglova scenario, which can be downloaded from the Anglova website†. The VM is preconfigured with EMANE and includes scripts to configure the network interfaces and radio models, tools to play back the scenario (including the mobility and path loss data), tools to generate

* Available at https://github.com/adjacentlink/emane/
† Available at https://www.anglova.net

network load, and tools to capture traffic for analysis. In addition, supplementary tools have been included to generate path loss data and to visualize the scenario during execution.

5.1.1 DAVC

The IST-161 RTG used ARL's DAVC management system to deploy the Anglova scenario (Figure 22) using the distributed EMANE emulation model. In this emulation model, the EMANE software is installed within VMs that execute the experiment's applications so their performance can be evaluated. The emulation environment is available as a cloud service to all IST-161 members.

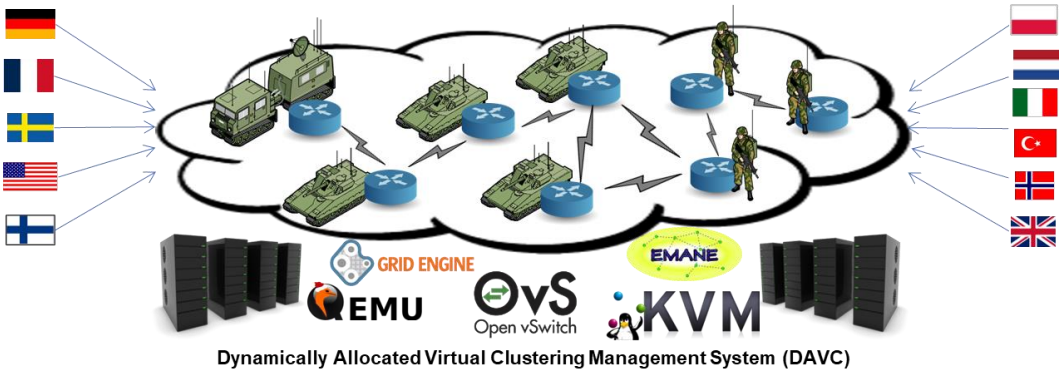


Figure 22. Emulation environment for IST-161.

DAVC is a web-based virtualization service and cloud-operating environment that creates complex virtual experimentation clusters that can be used for simulation-based, emulation-based, and hybrid field/emulation experimentation. DAVC deploys networked clusters composed of VMs tailored to user specifications. The DAVC management system abstracts away testbed infrastructure configuration through automated provisioning processes that configure the virtual networking for each VM. Clusters created by DAVC are heterogeneous, so each VM can have different operating systems (OSs), application sets, and hardware attributes such as RAM, CPU cores, hard disk, and network interfaces. DAVC provides a controlled, repeatable emulation environment for researchers to develop and verify concepts and algorithms designed for tactical networks.

Using DAVC, the Anglova scenario is distributed with a 1-to-1 mapping with each Anglova node running within a single DAVC virtual cluster node. The entire 269-node scenario can run within a 270-node DAVC cluster. When deployed in this manner, node 270 acts as the experimentation orchestration node and is responsible for executing the bootstrap scripting that launches the various applications and EMANE on the remaining 269 nodes.

The emulation environment provides the flexibility to deploy subsections of the entire 283-node scenario within a DAVC cluster. If a researcher is only interested in experimenting with the 159 mobile nodes contained in Vignette 2, the bootstrap scripting can map only those nodes contained within that portion of the Anglova scenario to 159 DAVC cluster nodes, where the corresponding EMANE configuration files will be run. This type of deployment model (i.e., using subsets of the Anglova scenario) was used in the experiments performed in Marcus et al. (2020) and is visualized in Figure 23. Marcus et al. were interested in evaluating the scalability and performance of a variety of group communications protocols on company- and multicompany-sized topologies ranging from 24 nodes to 96 nodes, and hence deployed several 96-node DAVC clusters and configured the experimentation scripting to run only a portion of the scenario that contained four 24-node companies.

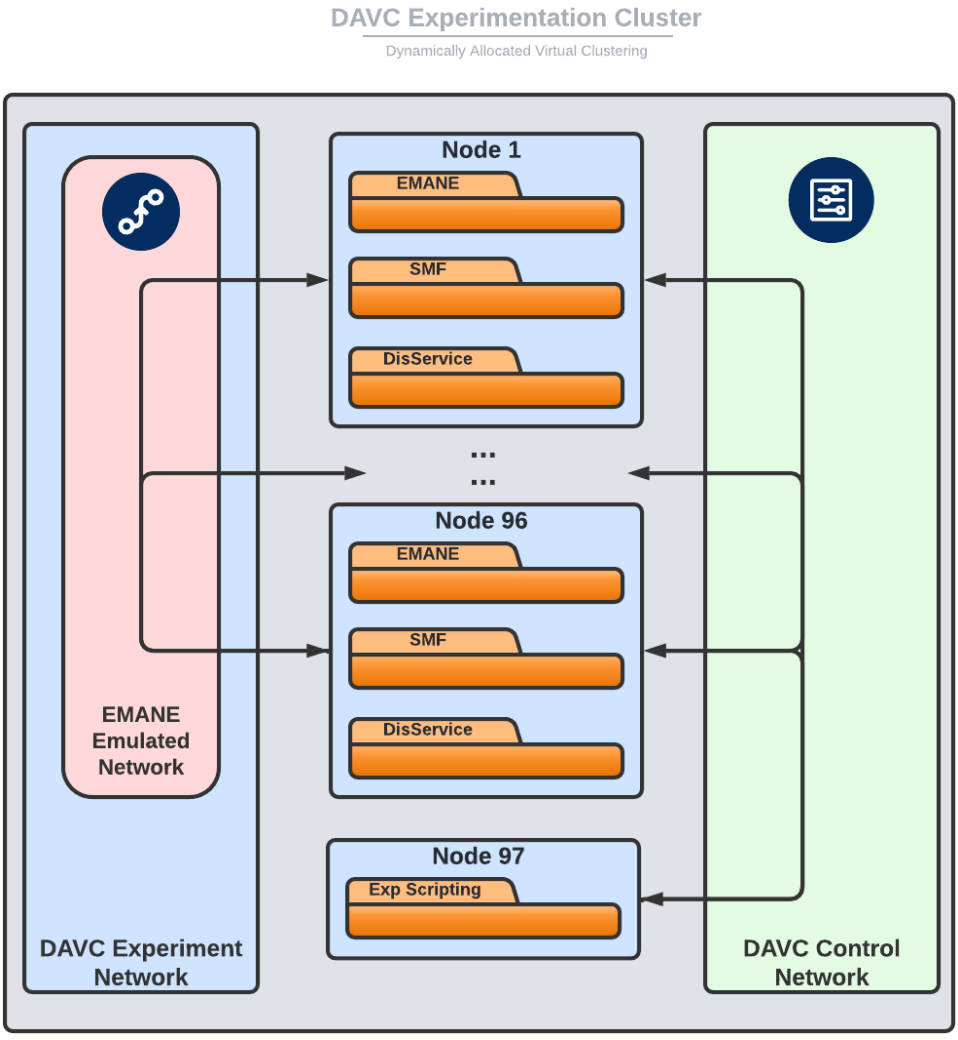


Figure 23. Example setup in DAVC of 96 nodes of Vignette 2 of the Anglova scenario.

5.1.2 VMWARE ESXi Testbed

In addition to the DAVC environment, the Anglova scenario was also deployed using a second, alternate configuration using the VMware ESXi virtualization platform. This deployment uses the hybrid model for EMANE configuration, where *n* VMs have one EMANE Server VM that runs all the emulation components for those *n* VMs. An example configuration is shown in Figure 24. The test VMs are designated test node-n (TN-n) and there could be multiple TNs per physical server, with a recommendation of one VM per CPU core. Each test VM is also configured with at least two network interfaces; one for the application data and one for control traffic (similar to the DAVC configuration). Each physical server also contains an EMANE server VM that runs all the EMANE components. One advantage of this deployment model is that the test VMs do not have to run any of the EMANE components. In fact, the network link to the test VM could also be connected to other computing and network devices (e.g., an embedded device, proprietary hardware, radio, a Wi-Fi access point), which allows hybrid experimentation with VMs and other physical hardware in the loop.

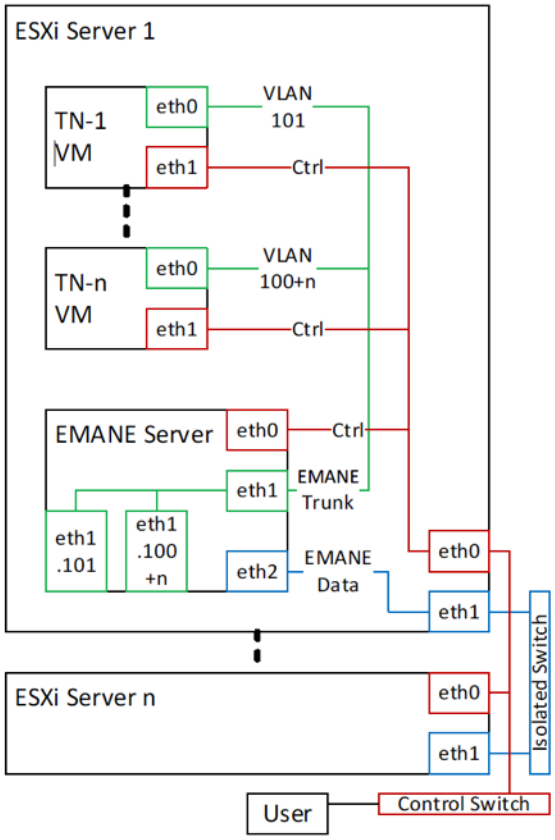


Figure 24. Anglova scenario running on the VMWare ESXi virtualization platform.

Regarding Figure 24, the network shown in green is configured as a virtual switch within ESXi, with each test VM connecting one or more interfaces to this virtual switch. Each connection is assigned a different VLAN ID. The EMANE server VM connects to this switch using a VLAN trunk that uses the 802.1q protocol. Multiple virtual interfaces (one per VLAN) are created within the EMANE server VM and then managed by EMANE. This configuration results in any traffic generated in the data interfaces of the TNs to be transferred to the corresponding virtual Ethernet interfaces inside the EMANE server, where each interface is mapped to one network emulation module (NEM). At runtime, EMANE reads from each interface, applies the necessary communications effects, and writes the data out to the interfaces that should receive the data. Figure 24's blue network is for EMANE data (OTA and events) exchanged between the different physical servers. Finally, the red network in Figure 24 is the control network used by a user to log into the VMs, start/stop the experiments, and collect data.

5.1.3 FKIE-VPN Environment for Distributed Experimentation

An additional approach for distributed experimentation has been developed and used in the context of the CWIX (NATO Allied Command Transformation 2021) where some of the IST-161-work has been tested. This section presents this environment in more detail since it can be used in combination with the Anglova scenario and DAVC environment. For testing at CWIX and for future STO research groups, a distributed VPN setup in general—particularly the one presented here—is expected to be useful.

The direct motivation for this distributed setup arose during the COVID-19 pandemic. The circumstances of this pandemic posed new challenges with respect to the execution of experiments involving different parties. In the past, participants of the involved parties could gather physically to conduct experiments, but this had not been easily possible in the first two years of the pandemic. Accordingly, new approaches to conduct network experiments in a distributed fashion have come into focus. In these setups, members and devices under test can be in different physical places in the world. This poses additional challenges to those already existing. An approach to enable experimentation by the different parties has been developed at FKIE in Germany and has already been tested in practice during CWIX in 2021 and 2022. The solution is based on VPN technology covering capabilities like basic network connectivity, featured data analysis functionality, and real-time radio network emulation (Barz et al. 2018). Furthermore, the solution allows for on-site experimentation of researchers as well as remote testing sites from distributed labs. The VPN approach was further enhanced in production to improve monitoring, performance, and management capabilities. In addition, the framework used for

data analysis was extended so live visualization and storage services could be provided to participants.

The following essential nonfunctional requirements can be identified regarding a testing infrastructure for this kind of event:

- Security
    - While unclassified, the network must still maintain a reasonable level of network security. Only eligible participants must be allowed and able to enter the network, and it should not be possible to tap into ongoing experiments without permission.
- Flexibility
    - It should be possible to draw connections between participating capabilities as needed. Ad hoc connections must be possible during the event.
- Reproducibility
    - The infrastructure should facilitate preplanning of connections and aid reproducibility of experiments.
- Scalability
    - The size of experiments should not be limited by the infrastructure (i.e., it should be possible to scale as required).
- Observability
    - It should be possible to monitor the state of the network so performance bottlenecks or other issues can be mitigated during the event. While these requirements are not trivial to support at an on-site event, it is still crucial to meet the same standards with a remote solution.

Three possible modes of execution were identified for CWIX. The first is on site (i.e., all capabilities must be present at the execution site and connections with external networks are possible but limited by the operators). Second is remote, when all capabilities participate in a remote fashion (i.e., using a VPN solution). There is no connectivity between the on-site network domain and remote participants. The third mode is hybrid, meaning it is possible to participate either on site or remotely and transparent connections between on site and remote sites are possible.

The VPN approach presented in this report has been designed to meet these requirements and simultaneously enable a hybrid mode of execution. This has been demonstrated successfully during the execution of NATO CWIX 2021 (Barz et al. 2021).

This section describes the architecture and technical components used to realize the distributed VPN infrastructure. First and foremost, VPN represents the de facto standard for overlay networks on the Internet. It can be both routed (Layer 3) or switched (Layer 2), and implementations support a variety of security measures that can be used to keep the data transmitted protected from external entities. A VPN is typically operated in a centralized fashion, with multiple clients connecting a single host (or a network site, in this case) to a VPN server. Clients usually connect using a signed certificate, issued by the server.

Figure 25 presents an overview of the VPN network architecture used at CWIX 2021. The experiment sites, presented at Figure 25's bottom, each use a dedicated client that connects to a VPN server. By establishing a connection to the server, the clients gain access to a VLAN trunk that holds the VLANs required at the experiment site. Typically, a client would use a set of experiment-specific VLANs as well as a common management VLAN that is also used for network analysis. A VLAN is shared by at least two experiment partners and basically represents a virtual Ethernet cable that can interconnect sites and systems. The server hosts a switching infrastructure that can bridge between clients so the correct VLANs are available at each trunk.

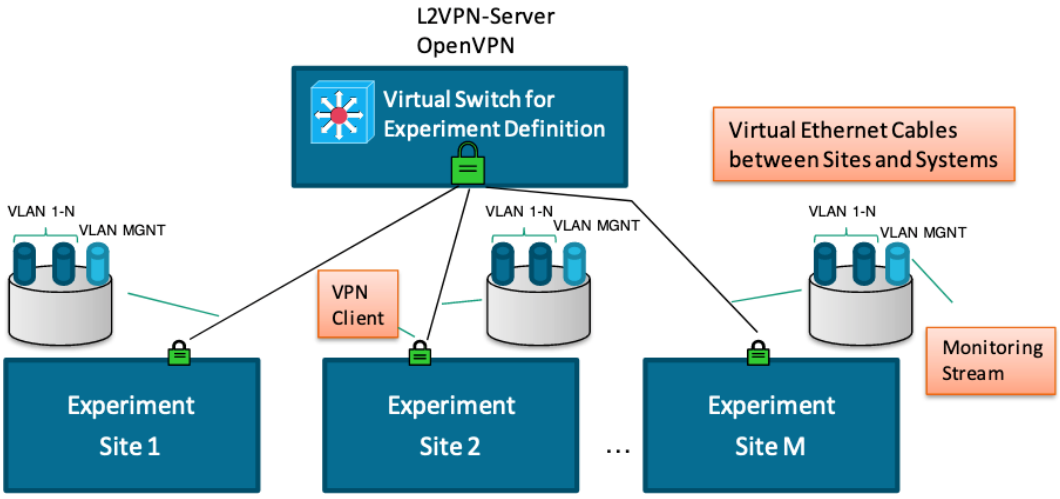


Figure 25. VPN architecture overview of different experiment sites managed by using different VLANs.

Regarding the VPN implementation to use, an early decision was made to focus on OpenVPN. This open-source software contains the required features in terms of security and Layer 2 access and is well-suited for automated reconfiguration. Access to the VPN (as well as encryption) is controlled using certificates (the most common approach). To ensure separation between the VPN clients, we use separate OpenVPN interfaces running on different ports. Each interface is assigned an internal tap interface that is then bridged via an Open vSwitch configuration. Since

the goal for CWIX 2021 was to support a hybrid mode, the network provided by the on-site operators needed to be connected into this Layer 2 infrastructure. To do this, a VPN client was set up at the on-site environment that would then bridge between VLANs on the network and the specific experiment sites requiring access to on-site services or test partners. This basic VPN architecture satisfied the requirement of security using state-of-the art encryption and authorization methods. In addition, experiment partners are provided only with those VLANs that are relevant to the experiments in which they actively participate. This allows the infrastructure to run many experiments in parallel, each within a well-documented and reproducible subsystem. It also allows for arbitrary connections between the attached network sites. In comparison to an on-site deployment with physical Ethernet connections, this VLAN-based approach can help significantly to reduce the effort in experiment preparation. Also, it greatly reduces the risk of configuration errors and aids debugging of possible issues. The system can be scaled as needed by three mechanisms:

- Additional client sites can be added by providing an OpenVPN instance. If required, these can be distributed over several hosts.
- If the range of available VLANs is exhausted, additional Open vSwitch tenants could be added.
- The system could be distributed over multiple network entry points, should a single uplink be insufficient. Therefore, the infrastructure can be scaled horizontally by increasing server resources.

To meet the requirement of reproducibility we use an approach that allows participants to specify their experiment's connections using a human-readable data-serialization language (yet another markup language; YAML). Using this configuration scheme, an automated process can generate (1) an assignment of VLANs for the experiment and (2) a configuration for VPN and switching in the back end. The fact that participants do not take part in the actual VLAN planning has shown to be a significant benefit. The system is designed in a way that allows for changes in the configuration (such as adding an experiment) during runtime of the infrastructure. This way, new experiments can be added on demand during the execution, meeting the requirement of flexibility. All configurations generated can be reused in upcoming CWIX events and serve as a documentation of connectivity between experiment partners. An example is shown in Figure 26.

```
CWIX-COMMS-EXP1:
NATION-A-NATION-B-EXP1 - 1:
- CWIX21-A
- CWIX21-B
NATION-A-NATION-C-EXP1 - 1:
- CWIX21-A
- CWIX21-C
NATION-A-NATION-D-EXP1 - 1:
- CWIX21-A
- CWIX21-D
NATION-B-NATION-C-EXP1 - 1:
- CWIX21-B
- CWIX21-C
CWIX-COMMS-EXP1- Analysis:
- CWIX21-A
- CWIX21-B
- CWIX21-C
- CWIX21-D
```

Figure 26. Example listing of a VLAN configuration for remote testing at CWIX.

This YAML structure specifies five network links that would result in an assignment of five distinct VLAN numbers and an automatic reconfiguration of the switching infrastructure. The last link shown is used for network analysis and shared by all four experiment partners. The requirement of scalability regarding the experiments is fulfilled, since the number of connections between the participants per experiment domain is limited by the amount of VLANs (up to 4095) and each additional connection requires an insignificant amount of resources. The number of participants is limited by the number of ports available (up to 65535) and not limited further by switching due to OpenVSwitch properties. This is a significant improvement over the limitations of physical or emulated switches.

During CWIX execution, it is important to keep track of the connectivity status of all nations, as defined by the requirement of observability. A custom dashboard was created to support this task. The dashboard reports information about the current connection status and time since the last status change. In addition to the connection status, the network infrastructure's capacity and availability at FKIE is constantly monitored.

The VPN approach enables researchers to interconnect their experiment sites in a well-defined fashion. In addition to these infrastructure-level services, the approach developed at FKIE includes an experiment and analysis stack, depicted in Figure 27. The core features provided by the environment are as follows:

- An orchestration platform "Analysis and Test Environment" (AuT) providing the following:

- Real-time network emulation using NS-3 (NS-3 Consortium 2024) and EMANE.
- Virtualized routers and applications.
- Scenario control and data distribution (including unit positions and C2 events).

- A data analysis stack including a processing back end and an Apache Kafka (Apache Software Foundation 2024) cluster for live streaming of measurement data and results, including the following:
  - A high-level experiment orchestration providing a user front end to coordinate experiment runs.
  - A persistent storage based on Amazon S3 that allows users to access the data produced during an experiment.
  - Live visualization services that visualize the processed data.
  - Data capture services that may be deployed to partner sites.

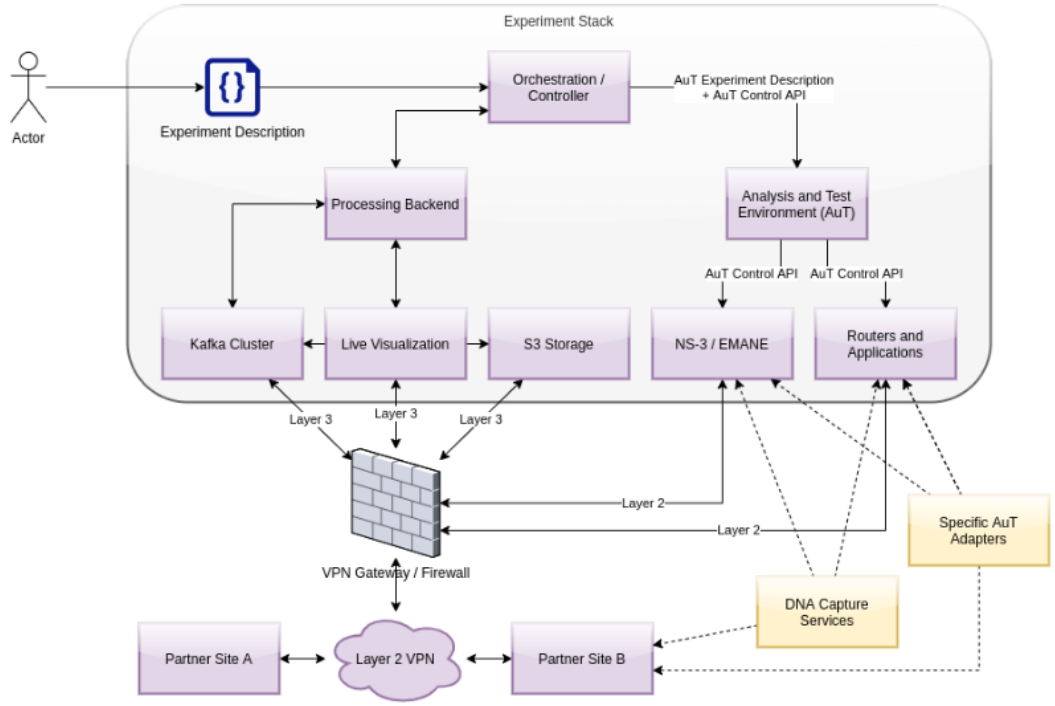


Figure 27. Pipeline diagram representing the different components and their interaction.

An instance of this stack is hosted for each experiment, enabling fine-grained control over computing and storage resources. Access to the services is limited through the access to the VLAN defined for the experiment.

The overall processing pipeline is organized as follows. In the first step, raw data is gathered using capture services. These are realized as lightweight components,

which (1) capture specific types of information, (2) are remote-controlled, and (3) can be deployed within an infrastructure or directly inside the device to be captured (depending on the type of information). At CWIX 2021, two capture services were distributed to experiment participants: packet capture (which captures and dissects network packets using tshark) and route capture (which captures changes in the systems' routing tables).

All services produce an output stream that is consumed and distributed by an Apache Kafka cluster. During the runtime of an experiment realized by the emulators and the AuT, a set of processing jobs is executed on the processing back end, producing analysis data. This data is then persisted to the S3 storage and used to produce a live visualization of the experiment. The general idea is that an analysis can incorporate information from multiple data sources to perform cross-layer analysis. Other sources of information, which have been implemented as data probes, include EMANE capture, which captures internal status tables from EMANE (Navy Research Laboratory 2024); NS-3 capture, which captures internal status tables from NS- 3 (NS-3 Consortium 2024); and NetJSONd capture, which captures the status of a NetJSONd service (an implementation of NetJSON [Capoano 2021; Capoano and Kaplan 2015] by FKIE).

The visualization component can construct graph-based representations based on gathered routing information captured by the deployed services on routers. These snapshots can be used to construct routing graphs to get a deeper understanding of the routing situation in each experiment at any point in time. Furthermore, data can be visualized using Layer 3 or Layer 2 information. The graph visualization has been a valuable tool for understanding and debugging test setups, as well as visually documenting test results. Another type of visualization that was used during CWIX 2021 showed a graph based on packet capture analysis (number of packets per second between nodes). Experiment partners were provided with access to both the live visualization and the storage service, making the experiment results available to partners without additional effort.

The interaction of the described CWIX components is based on a well-defined analysis workflow. The workflow is divided into a three-stage process with different tasks to be performed at each stage. These are to be performed either by the user or system components. This workflow is depicted in Figure 28. During experiment definition, the experiments are to be created and prepared by the user. The first step is to select the number and configurations of the components to be used in the experiments. The system supports routers, applications, and emulators as devices under test. Afterwards, connections of applications and routers must be configured as well as connections between routers and emulators. The final step during the definition stage is to select mobility tracks for the nodes and number of

replications. The execution phase is handled by the system in an automatic manner to reduce errors caused by manual steps. During each experiment execution, all components must be instantiated and configured first. Next, the experiment can be conducted by capturing data, replaying mobility tracks, and generating traffic. Finally, all components must be cleaned. The next experiment can be executed afterward. During the analysis stage, researchers must prepare the gathered data by filtering or mutating. Subsequently, the prepared data is used to calculate metrics. These metrics can be used for relative performance comparisons of different setups and to determine properties or behavior of the components under investigation. One can finally visualize the results by using plots.

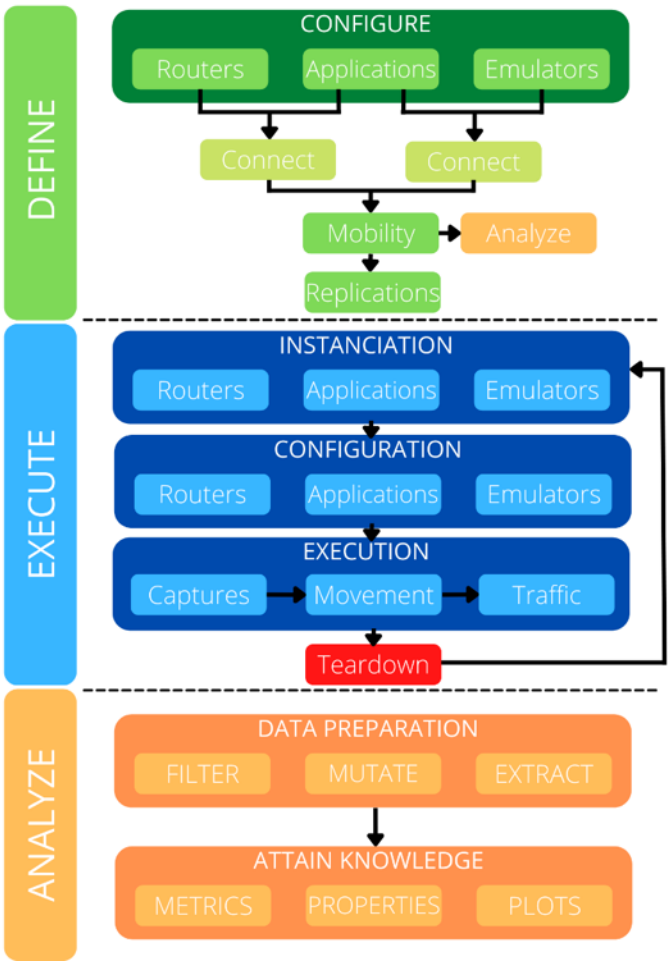


Figure 28. Workflow of analysis environment.

The distributed architecture presented in this report seems to be a promising approach to future experiments. The ability to connect experiment partners over different sites allows for flexible planning and could allow for testing that would not be possible without such an approach. While the architecture provides means for organizing the infrastructure level of a distributed experiment, one topic for

future work is to extend on the ability to perform measurements in a coordinated way. For example, it would be desirable to have a standardized method for distributed coordination of events throughout the system. There could be a need to synchronize different network emulators across the test sites (e.g., with mobility data) or to provide the systems under test with input (such as generating user input to C2 systems). Although this kind of control is partially realized by the AuT environment (hosting and controlling C2 systems), it is currently limited to controlling the local test site.

5.1.4 EMANE

EMANE (Navy Research Laboratory 2024) was developed by CenGen (now AdjacentLink) under sponsorship by the U.S. Naval Research Laboratory, the Office of the Secretary of Defense, and ARL. It is freely available on GitHub and provides a flexible framework for emulating multiple types of radios. EMANE primarily addresses the physical and medium access control (MAC) layers of the network stack and includes the following three different radio models: a generic RF pipe model, an 802.11abg model, and a time division multiple access (TDMA) model. Each of these models offer various customizations of the radio parameters, antennas, and error models. While models of some specific tactical radios exist, they are not available in the public domain and hence will not be discussed in this report.

EMANE supports a flexible deployment model; it may be centralized, fully distributed, or a hybrid of the two. Each radio interface of a network node in the emulated network is represented within EMANE by an instance of a NEM. The NEMs instantiate the physical and MAC layers of the radio being emulated and each EMANE instance can contain one or more NEMs. In the centralized model, all the NEMs are instantiated with a single instance of EMANE. In the fully distributed model that is used in the DAVC deployment, each EMANE instance contains only one NEM. In the hybrid model that is used in the Florida Institute for Human & Machine Cognition deployment, there are multiple instances of EMANE, each with multiple NEMs. In the hybrid and distributed deployment models, the NEMs communicate with each other using an OTA channel that uses UDP multicast over a control network (independent of the emulated network interfaces). The NEMs also react to control events that are sent to them over the EMANE event channel, typically also via UDP multicast. For example, when playing the Anglova scenario, the position and path loss updates are sent to EMANE via this control channel. See the EMANE documentation for more details (GitHub 2024).

5.1.5 SCB EMANE Waveform

SCB is a relatively new waveform that increasingly is being used in many types of scenarios. It provides robust multi-hop communication with low overhead and particularly is efficient for broadcast traffic. Therefore, given the interests of the IST-161 RTG, it was decided to attempt to incorporate SCB into the group's experimentation framework.

An initial approach to cooperative broadcasting at the symbol level called *opportunistic large array* was described in Suri et al. (2018). The name *barrage relay networks* for cooperative broadcasting is introduced in Blair et al. (2008). Barrage relay networks and SCB are similar as both use autonomous cooperation for time-synchronized nodes and a TDMA structure. The performance of a dynamic scheduling method for SCB is investigated in Grönkvist et al. (2016). A comparison of the network broadcast capacity obtained by SCB and traditional TDMA-based schemes is published in Komulainen et al. (2016). SCB is shown to be more efficient in mobile networks unless the network is large and dense (i.e., if the network is more or less a single-hop network).

5.1.5.1 SCB Background

SCB is designed for robust broadcast communication in mobile ad hoc networks. One or several separate broadcast streams are managed with cooperative relaying and an underlying TDMA structure with separate sets of time slots scheduled to each source (broadcast stream). All nodes that receive a new packet in a time slot simultaneously retransmit it in the next time slot that is scheduled to the same broadcast stream. For each broadcast stream, the protocol groups D time slots to a unit that we call a cooperative broadcast (CB) slot with size D. The CB slots are grouped into repeating frames, as shown in Figure 29. The example in Figure 29 has a CB slot size of four. A packet can be transmitted D hops from the source node with one CB slot, but with the next CB slot scheduled for the same broadcast stream, it propagates another D hops further out in the network and so on until all nodes are reached. The CB slot size D is also equal to the reuse distance that describes the minimum number of hops between two nodes that can transmit different packets simultaneously.

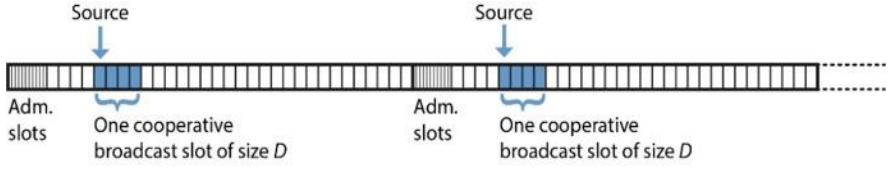


Figure 29. Example of SCB with dynamic scheduling.

In the CB slots allocated for a source node, only packets originating from the source node are handled. When the source node transmits a new packet, it will do so in the first time slot of its CB slot. All nodes that receive a packet simultaneously retransmit the packet in the following time slot in the same CB slot. This procedure is repeated until all nodes in the network have retransmitted the packet once. The reception of multiple copies of a packet in a time slot is assumed to be handled by the receiving nodes, similar to how multipath propagation is handled. For more detailed descriptions, see Blair et al. (2008), Grönkvist et al. (2016), and Komulainen et al. (2016).

The *network broadcast capacity* is the maximum number of packets per time slot that can be broadcast from the source to all destinations, where each packet perfectly fills a time slot. As nodes either can receive or transmit packets, the network broadcast capacity can never exceed one. A TDMA network with only one hop is an example of a network that can reach network broadcast capacity one. In an SCB network, we can broadcast one packet per CB slot (using D time slots). Hence, the network broadcast capacity is the inverse of D (i.e., $1/D$). To obtain the bit/s that can be transmitted in the network, the network broadcast capacity is multiplied with the link data rate R_L. The idea of the simplified modeling is to configure the parameters for the simplified approach in such a way that the same network broadcast capacity is obtained as it would be in a full-scale implementation of SCB. An example of dynamic scheduling with SCB is shown in Figure 29.

5.1.5.2 Approach Based on Simplified Multicast Forwarding

One approach to model SCB is to use simplified multicast forwarding (SMF) in combination with EMANE's RFPipe radio model. One of the most popular ways of implementing broadcast is to use the multipoint relay (MPR) method according to the SMF framework (Macker 2012). However, when emulating SCB, all nodes besides the source node are configured as MPRs. This configuration results in full flooding (but without simultaneous transmissions) as all nodes besides the source node relay a packet once. In SCB, all nodes besides the source node relay a packet once unless a restriction is set up on the number of hops through which a packet can be relayed.

Flooding a packet with SMF takes N transmissions in N time slots, while in SCB the N transmissions are only done in D time slots since they are performed simultaneously. Consequently, an SCB network has a capacity that is N/D times higher, thus requiring to artificially increase SMF capacity of the same factor when emulating SCB. When an RFPipe radio model is used without TDMA, each RFPipe transmits independently from each other. This means that when emulating SMF, full flooding the RFPipe data rate should be R_L/N. When emulating SCB, however,

the RFPipe data rate must be N/D times higher (i.e., R_L/D). Futhermore, since dynamic TDMA scheduling is not implemented, a fraction of the data rate representing the scheduling overhead O_S also should be deducted from the data rate. The RFPipe data rate is therefore set to

$$\frac{R_L(1 - O_S)}{D}$$

where D is set to four in the present investigation.

This approach mimics a dynamic SCB, meaning it is assumed that it adapts itself to different traffic loads in the source nodes. The traffic adaptation requires a set of administrative slots for the rescheduling mechanism in each time frame, as illustrated in Figure 29. The length of the administrative slots can be made shorter than the user time slots. It is possible to estimate the scheduling overhead O_S. The current dynamic schedule is transmitted in each node's administrative CB slot. Assuming the status of each slot is represented by two bits, up to four states can be defined. The following three states are needed at minimum: owned, blocked, and free. The *owned* state means that the node is the owner of the CB slot (i.e., it is the only node allowed to transmit new packets in the CB slot). The *blocked* state means that the time slot is owned by another node; this is used to acknowledge other nodes' allocation of a slot. The *free* state means that the time slot is free and can be allocated by a new node. It follows that the administrative slot must contain at least *2M* bits, where *M* is the length of the schedule that can be adapted. As the CB slots involve *D* transmissions per each frame, the required channel resource per node is *2DM/t*, where *t* is the time between a node's administrative CB slots. For an *N* node network with a link data rate of R_L, this means that a fraction

$$O_S = \frac{2DMN}{tR_L}$$

of the available time is used for administrative time slots carrying scheduling information. We denote this fraction as the scheduling overhead. In practice, an administrative time slot may need to be larger than *2M* bits because additional information such as node identity, traffic load, and priorities can be included. In addition, small time slot lengths may be inefficient due to forward error correction, guard times, and other factors. However, we do not consider this here.

For this approach, based on SMF and with at most 96 nodes in a network, we assume that the length of the schedule is $M = 2N$ (i.e., the number of CB slots that can be dynamically allocated), that *t* is 1 s, and that $D = 4$. To keep the scheduling overhead reasonably small for larger networks, *M* may have to be smaller than *2N* and *t* may have to be larger than 1 s. The scheduling overhead for the different network sizes of 24, 48, 72, and 96 nodes are approximately 1.1%, 4.2%, 9.5%, and

17%, respectively. For example, when emulating the 96-node network with R_L = 875 Kbit/s, the data rate of the RFPipe is set to 182 Kbit/s instead of 219 Kbit/s to include the scheduling overhead.

5.1.5.3 Approach Based on Precomputed SCB Topology

A second approach to model SCB is to include the cooperative effects when calculating the set of receiving nodes for each cooperative transmission. More precisely, the sum of received powers from the transmitting nodes is compared with a path-loss threshold $L_{b,\max}$ for each other node to decide if the transmission is successful or not. In this way, the total set of nodes that is reached from a source (in an arbitrary number of hops) can be computed based on the given path-loss values between the nodes.

Performing this calculation for each node as a source, we get a network topology description that we call an *SCB topology*. An SCB topology is precomputed for each second in the scenario. Assuming the required SCB transmissions takes less than 1 s and the topology does not change during this time, the precomputed SCB topologies can be used as a model of the SCB transmissions in the emulations.

One RFPipe is connected directly to all nodes in the set of receiving nodes given by the SCB topology. This approach is different from the SMF-based approach and mimics SCB with a fixed schedule. This means that each node only has one CB slot per frame, thus leading to a schedule of $M = N$ CB slots. The data rate in the RFPipe is set to

$$\frac{R_L}{ND}$$

where N is the number of nodes in the network. It is assumed all nodes send an equal amount of traffic. If only K nodes send traffic, N is simply replaced with K. If different nodes send a different amount of traffic, the RFPipe in the different nodes can be set differently, but the data rates need to add up to no more than a total data rate of R_L/D.

The delay can be modelled with fair accuracy. We assume for simplicity that the time slots of a CB slot are grouped together and the frame size is $M = N$ CB slots. Thus, on average a packet needs to wait a half frame on a CB slot for up to 4 hops, 1.5 frames for 5 to 8 hops, and so on. Only the waiting time for the source to transmit is included in the delay estimation, not the actual transmission times that could be up to four time slots in case of four hops. However, the delay caused by possible traffic queues in the ingress nodes need to be measured in the emulation and added to obtain the total delay of a transmission.

Let us now consider some of the pros and cons of the precomputed SCB topology-based approach. This approach models a static SCB that must be preconfigured after the expected traffic loads. It has the advantage of including cooperative effects, thus more accurately modelling the actual network connectivity obtained with SCB. Also, traffic queues only occur in the ingress/entry nodes. This makes it possible to obtain fairly accurate delay estimates. Furthermore, this approach supports any Layer 3 protocol that can run over EMANE, including TCP. The disadvantage with the approach is the preparation work required to generate the SCB topology data for each network size and parameter setting.

5.2 Experimentation Test Harness

A test harness was developed for the experiments in IST-161. This test harness is a test platform that is used to orchestrate the multitude of group communications experiments. It is a Java application that performs the core tasks of message generation and transmission, event logging, and collection. In addition, the platform ensures all the tests are run consistently with each other and the same number of messages of each type are generated at the same times by the same nodes within the Anglova scenario. This ensures a fair comparison among the examined protocols. Event logging is protocol-agnostic, meaning that regardless of the underlying protocol being used, the test platform tracks actions such as message transmissions and receptions, generating log messages that are immediately delivered to a centralized statistics server (also written in Java).

The test platform supports three types of messages that reflect the traffic that typically flows in tactical networks: Blue Force data (BFD), SD, and documents from headquarters (Docs). However, only BFT messages with a time-to-live (TTL) of four were used during the tests. BFT messages are short messages that identify the current position and status of the transmitting mobile node; each node typically sends BFT message updates periodically to every other node in the Company every X s. The exact traffic patterns that were used during the experiments are described in Section 7.

5.3 Anglova Scenario Emulation

5.3.1 Experiment Facilitation Scripts and Tooling

In this section, we provide a summary of the scripts and tools we developed to facilitate experimentation with EMANE, our SCB radio model, and OLSRv2 within the Anglova emulation scenario.

5.3.1.1 Emulation Initialization Scripts

We developed scripts to initialize and configure the experimentation scenario and testbed. The scripts would ensure the testbed and emulation were preconfigured with 4 companies each composed of 24 nodes for a total of 96 nodes and 5 emulated networks (i.e., 4 intra-company networks—1 for each company—and an overlay network interconnecting each of the 4 company networks). The experimentation scripting designated the first two nodes in each company as network gateways and configured them to run on both the overlay network and their respective intra-company network. The script designated the fourth node in each company to publish documents during the experiment (if required) when OLSRv2 was not being used.

5.3.1.2 Network and Bandwidth Constraint Scripts

We also developed tools to configure the MAC addressing and bandwidth constraints on the 96 nodes. To configure the MAC address, we developed a tool to change the original MAC address on each node to a custom MAC address of our choosing. We also developed a tool that leveraged the traffic control commands on Linux to set the bandwidth limits on the node's network interfaces. Two separate bandwidth constraints were set on the overlay nodes since they had two interfaces, one connected to the overlay network and the other connected to that node's company network. The rest of the nodes have only one company interface with the same bandwidth value. The script configured the following values that were set for the Anglova–SCB experiments:

- Company node’s bandwidth constraints:
  - tc qdisc add dev ens192 root tbf rate 7.292kbit limit 32768 burst 1540
- Company overlay node’s bandwidth constraints:
  - tc qdisc add dev ens192 root tbf rate 29.168kbit limit 32768 burst 1540
- Overlay node’s bandwidth constraints:
  - tc qdisc add dev ens224 root tbf rate 27.344kbit limit 32768 burst 1540

5.3.1.3 Emulation Event Scripts

We developed scripts to launch and terminate test protocols, the emulation scenario, TCP dump network captures, and the OLSRv2 routing protocol when it is required for the experimentation. As an alternative to playing back the scenario via

emulation event log files* and the EMANE Event Service Tool, we also developed a command line tool to orchestrate custom CommEffect† emulation events. The 96 nodes were distributed across three different Linux servers where each node runs an instance of EMANE and contains the configurations to drive the EMANE CommEffect model. These three servers also run the Anglova CommEffect driver via command line, translating the SCB scenario model files into EMANE CommEffect events for the nodes contained within that server. For instance, we execute two scripts for server 1; one generates the emulation events for the company nodes, and the other generates the emulation events for the overlay nodes. What follows is an example of running the Anglova CommEffect driver for the nodes on the first of the three servers:

- Overlay Node's CommEffect events
    - java -cp $CP -Xmx3g us.ihmc.anglova.AnglovaDriverCE -wbnem HopsNEMMapping-4C.txt-hop Comp_1_2_3_4_5_6_SCB_Frq300gaindB139.txt -bind 127.0.0.1 -startStep 5000 -endStep 7000 -startNode 1 -endNode 32
- Company Node's CommEffect events
    - java -cp $CP -Xmx3g us.ihmc.anglova.AnglovaDriverCE -wbnem HopsNEMMapping-4C.txt -hop Comp_1_2_3_4_5_6_SCB_Frq300gaindB139.txt -bind 127.0.0.1 -startStep 5000 -endStep 7000 -startNode 1 -endNode 24

where AnglovaDriverCE is the Anglova CommEffect driver, HopsNEMMapping-4C is the mapping script between the EMANE Nems and node ID, Comp_1_2_3_4_5_6_SCB_Frq300gaindB139.txt is the SCB scenario model file, startStep/endStep define the scenario timesteps that the experiment will run, and startNode/endNode define the nodes located on that server.

We keep a utility directory in a Linux console that is used to run all the scripts previously mentioned in this section. To start the scenario for a protocol using OLSRv2, there is a script for each company that runs all the necessary scripts to start each piece in the required order. For instance, for company 1, the order starts with OLSRv2, then SMF, next is Deutsch-Amerikanische Freundschaft (DAF; see Section 6.4 for more details), then the protocol instance, followed by the emulation scenario, and finally TCP dump for nodes 1 to 24. This process is repeated for the other three companies. There are also scripts to collect all the traffic control logs

* Available at https://coreemu.github.io/core/emane/eel.html

† Available at https://github.com/adjacentlink/emane/wiki/Comm-Effect-Model

and the DAF logs files from all 96 node network interfaces, as well as the EMANE status files from the three EMANE servers for debugging purposes.

5.3.2 Observations and Encountered Issues

We observed an issue using the CommEffect model with EMANE for realizing the precomputed SCB implementation. The CommEffect model can be used to enforce capacity limits between a sender and a receiver. However, we found CommEffect was allowing more outgoing traffic than should be allowed when the same transmitter is communicating with multiple different receivers. To fix this issue, we relied on the Linux Token Bucket Filter queuing discipline to ensure the bandwidth limit was properly enforced for each node transmitting on the network.

6. Protocol Development

This section describes the protocols and middleware solutions that have been investigated by the RTG and have been developed or improved by participants during IST-161. Other middlewares and protocols that were included in the experimentation but used as-is are not the focus of this section. The experiments with these solutions and their results are part of Section 7.

6.1 GDEM

The GDEM is a group communication protocol developed by TNO and is a further development of the JDSS-IEM (NATO 2017b) as specified in STANAG 4677 (NATO 2014). It was first introduced to the public in the International Conference on Military Communications and Information Systems (ICMCIS) 2021 report written by the IST-161 RTG (Suri et al. 2021). The JDSS-IEM is part of FMN Spiral 4 specifications.

6.1.1 Background

We attempted to evaluate the JDSS-IEM in our 2019 experimentation environment. We used a NATO open-source implementation provided by the Dutch Ministry of Defense for the evaluation. Initial results were quite good but never made it to publication, as both the JDSS-IEM specification and the implementation used had drawbacks that made it difficult to use within the experimentation environment.

First, the JDSS-IEM (or more specifically, the underlaying transport protocol NATO Friendly Force Information Interface Protocol 2 [NFFI IP2; NATO 2017c]), has a message size limit that is smaller than was needed for our experiments. For example, the message size we used in our experiments for SD messages, 128 kB, are not supported by NFFI IP2.

Secondly, protocol and payload within JDSS are tightly integrated. Protocol messages and the data model are tightly integrated and are part of an eXtensible Markup Language (XML) format. This makes it more difficult to send arbitrary payloads, like we do in our experimentation environment.

Thirdly, JDSS uses EXI (World Wide Web Consortium 2016) to compress messages; however, the test messages generated by the group communication test harness were not intended to be compressed. Because the generated data can be compressed very well, a protocol that has compression enabled could give better results than it would if the data was not so easy to compress.

Finally, the implementation of the protocol we used required a Microsoft Windows environment at that time, whereas our experimentation environment was Linux-based.

We still managed to do a small-scale experiment that consisted of 24 nodes sending only Blue Force message data. The promising results of this experiment triggered TNO to design and build a new protocol based upon the basic principles of JDSS-IEM that would resolve the issues we found. Having this implementation would also allow for experimenting with new features that can be added to the protocol.

6.1.2 Initial GDEM Version

The initial version of GDEM differs from JDSS in the following four ways:

- GDEM decouples the JDSS data model from the exchange mechanism by using binary message headers, whereas JDSS uses XML-based protocol messages, thus making GDEM agnostic for the payload it sends. See Figure 30 for an example.
- GDEM allows for evaluation the message header without having to decompress and parse the message payload.
- GDEM does not use NFFI IP2 for message segmentation and transport; instead, it uses a protocol called Simple UDP Segmentation Protocol (SUSP).
- Using compression is optional in GDEM and may be left to the application, whereas JDSS uses EXI or GZip compression for the full message payload (including the protocol headers).

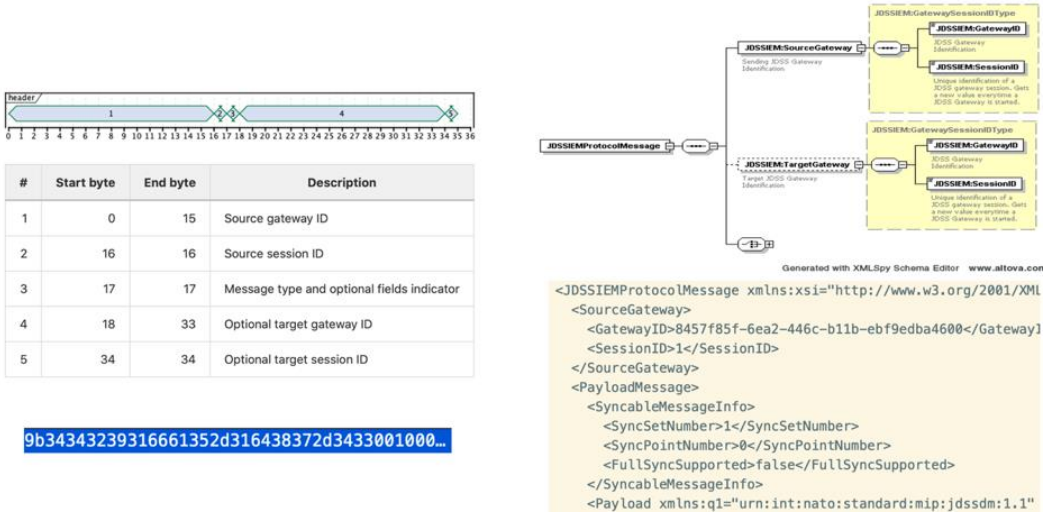


Figure 30. GDEM binary message envelope (left) holds similar information to the JDSS-IEM–based XML message (right).

The initial version of GDEM uses SUSP as a transport layer. SUSP is currently a proprietary protocol developed by TNO. It is comparable to NFFI IP2 as it is used by JDSS-IEM but allows messages with larger payloads. Like NFFI IP2, it uses UDP multicast to exchange message segments between participants. SUSP can decompose messages into a maximum of 65,536 segments, making the theoretical message size 96,141,312 bytes when a maximum transmission unit of 1,500 bytes is used. SUSP allows variable sizes for segments of the same message. The SUSP segment header is 5 bytes long.

GDEM, like JDSS-IEM, uses a repair window as a reliability mechanism. When a receiving gateway detects missing messages that still fall within the repair window, it will send out a synchronization request for the missing messages. To prevent network starvation, a mechanism is in place to limit synchronization requests and responses. When messages are missing outside of the repair window, the protocol has a full state synchronization feature that allows a participant to request the current state from another participant. We did not use full state synchronization in our experiments.

GDEM does not have specific features to accommodate the network compartmentation, which we wanted to experiment with in the context of this group; instead, the concepts GDEM borrows from JDSS were designed with only 4 to 8 nodes in mind. To support network compartmentation in the experiments we did in the context of the ICMCIS 2021 report (Suri et al. 2021), we introduced a relay function that sits between the GDEM and SUSP layers and relays messages between the company and overlay networks. This is an opportunistic way to support this network topology but does not scale very well. To prevent messages from looping around, we introduced a method that drops duplicate messages when they

are received within a preset threshold timer. Because GDEM uniquely identifies each message that is sent, each new message will be different regardless of the payload. Only repair messages may have duplicates; to prevent these from being dropped, the threshold value should not be more than a few seconds depending on the configuration of other timers used by the GDEM protocol.

6.1.3 Improving the Information Exchange Protocol

Like the JDSS-IEM, the initial version of GDEM could only repair full messages. When only a few segments of a message would go missing during transmission, the protocol does not have the ability to repair only these missing message segments. Adding the ability to repair missing message segments and further improvement on the message-relaying function were the main improvements we wanted to achieve during the first half of 2021. For the experiments done in the context of the MILCOM 2021 report (Tortonesi et al. 2021), we introduced the following changes to GDEM:

- GDEM is now able to repair partially received messages, whereas JDSS-IEM needs to resend the full message when a single segment of that message is not received.
- GDEM is now able to segment and reassembles messages, no longer relying on a separate transport layer (e.g., NFFI IP2, SUSP) for message segmentation.
- Each message segment has a full GDEM header, making it easier to analyze the protocol behavior in network traces.
- Next to the 16-byte gateway identifiers that are also used in JDSS, GDEM supports 4-byte gateway identifiers. The two types of identifiers can be used next to each other. The type of identifier is part of a set of header extension bits.
- A newly designed message relaying mechanism is used.

Adding a segment repair mechanism to the protocol was on the top of the priority list of changes to the GDEM protocol. To accommodate this, we added message segmentation to the protocol, thereby removing the dependency on SUSP to do it (Figure 31). The repair mechanism for message segments works like the message repair mechanism that was already in place. We added to notion of message segments to repair messages, making it possible to request individual segments as well as ranges of segments. Requesting segment ranges requires less space in the repair request, as multiple segments can be requested by specifying only the start and end segment of the range. Some changes to the back-off mechanism needed to be made so the protocol would not send a segment repair request before the sending gateway sent the full message in the first place.

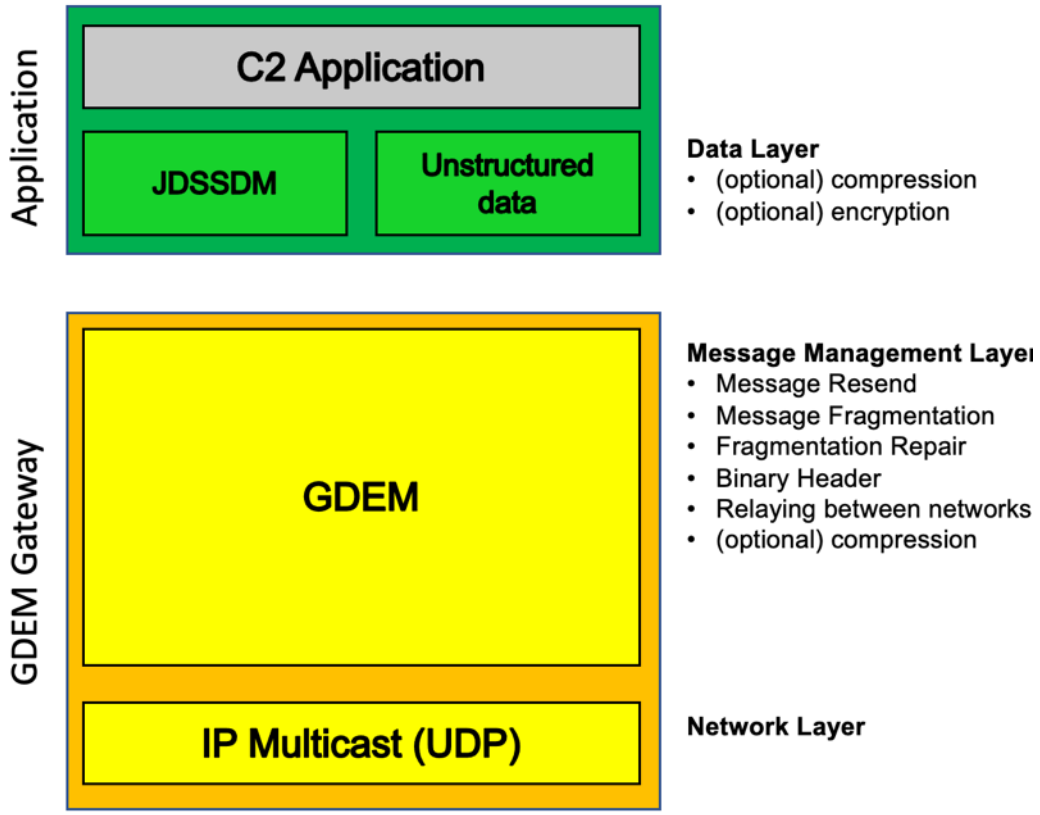


Figure 31. GDEM architecture.

As a side effect of removing SUSP dependency, we are now able to decode GDEM messages in network traces, making it easier to analyze protocol behavior using network traces. We have created a decoder for this that allows us to inspect GDEM messages using the Wireshark network analyzer.

It is arguable that the 128-bits/16-byte address space for gateway identifiers, as specified in JDSS-IEM, is needed in all situations. We changed the protocol to support 4-byte identifiers without removing the 16-byte support. A bit in the header indicates whether the address in a message is either a 4- or 16-byte address.

The updated GDEM protocol now supports networking compartmentalization using a new relay function. The relaying behavior is now part of the GDEM protocol specification as an optional protocol feature. A relaying gateway that implements the relay function can be part of multiple networks. Relaying rules describe how messages should be relayed between these networks. A relaying gateway may be part of any number of networks; there is no theoretical limit.

When a message is relayed by a gateway a hop counter (i.e., part of the message envelope) will be updated. The gateway will also add its own address to the

message header as the last-hop address. This allows other gateways to detect whether a received message was relayed by a relay or not.

For each network to which a relaying gateway is connected, it will learn what other gateways are active on that network. As messages can find their way to the relaying gateways via multiple networks, the relaying gateway will consider the network(s) with the lowest message hop counter as the home network for a gateway. A relaying gateway will only relay messages to a network when that network is not the home network of the sender of the message. When a message has a specific target address, the message will only be relayed to the home network that belongs to the target address.

As multiple relaying gateways may be active in the same networks, messages are only relayed to networks where the message was not seen before. To accommodate this, a relaying gateway will wait for a random delay interval to listen for a duplicate message before it decides whether to relay the message. If frame an identical message is received in that time, that message will be dropped as that message was already relayed to that network by another relaying gateway.

To optimize message synchronization between nodes in different networks, the relaying gateway can answer synchronization requests on behalf of other nodes. When, according to the relaying rules, a synchronization request message may be relayed by a relaying gateway, that relaying gateway may answer that synchronization request if it has the requested data in its message cache.

The results of the updated protocol can be found in the report that was presented during MILCOM 2021 (Tortonesi et al. 2021). The implementation of the relay function was not fully finished during the experiments done for that report. Later experiments showed that the performance of the protocol improved upon the results presented in Tortonesi et al. (2021).

6.1.4 GDEM as Middleware

The experiments that were done with GDEM in the context of this RTG used a middleware function that uses the GDEM protocol to synchronize data between nodes. The middleware function, called GDEMGW, acts as a gRPC* server towards applications that have a communication need—in our case, the group communication testbed. In theory, multiple applications can use the same instance of the GDEMGW for their communication. The middleware function currently

* Available at https://grpc.io/

does not expose the full capabilities of the GDEM protocol. Full state synchronizations are not yet supported by this middleware.

6.1.5 GDEM's Future

TNO and the Dutch Ministry of Defense are exploring if GDEM can replace JDSS-IEM in a future version of STANAG 4677. Therefore, the GDEM specification will not be part of this report; the final version of the specification may end up in an updated version of the unclassified STANAG 4677. We currently target to make this part of FMN Spiral 6 and plan to do multinational experiments between GDEM implementations during upcoming editions of CWIX. TNO is exploring ways to make GDEM's specification and source code of the reference implementation available for organizations from NATO countries.

Our target was to make GDEM as simple as possible. In some cases, we had to make compromises between simplicity and making the protocol more reliable in some edge cases. We believe we managed to keep GDEM's inner workings simple and easy to understand and implement. We believe GDEM would be a great candidate for coalition and also national group communication in mobile tactical networks.

6.2 DisService

The Dissemination Service (DisService) (Suri et al. 2010) is a peer-to-peer, publish/subscribe, message-oriented information dissemination middleware designed to perform efficient, disruption-tolerant message delivery in tactical environments. Publishers can send messages in the context of a group; DisService will take care of delivering published messages to all subscribers of the group. DisService provides disruption tolerance by implementing the store, carry, and forward paradigm and leverages strategies to promote information availability in the network. Figure 32 shows the architecture of DisService.

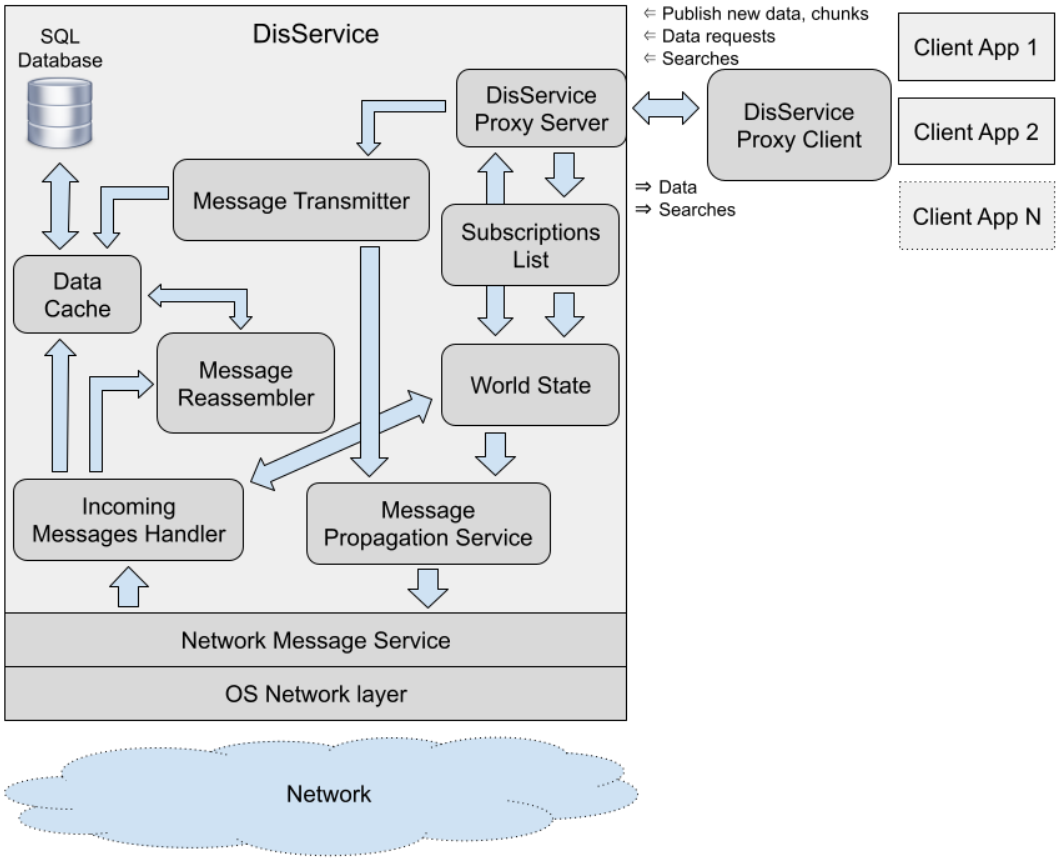


Figure 32. Architecture of DisService.

One of the fundamental assumptions in DisService is that, in a peer-to-peer wireless environment, the cost of transmitting a message using a local broadcast or multicast is the same as transmitting the same message using unicast. Therefore, DisService always prefers broadcast or multicast communications that enable all direct neighbors to receive and cache the message (or any fragment thereof). This feature, called *opportunistic listening*, significantly increases data availability and enables DisService nodes to retrieve missing fragments from any node in the network that is in their possession. Additionally, message replication strategies can be enabled via configuration, allowing nodes to opportunistically request or push messages with low network availability or marked as high priority whenever the channel is underutilized. Several cache expiration strategies can be configured to control the expiration of cached messages. For our experiments, message replication was turned off and cached messages never expired.

DisService messages are encapsulated within lower-level network messages produced by a library called *Network Message Service* (NMS). NMS interfaces with the OS to provide message reception, transmission, and network interface detection; additionally, NMS supports message fragmentation, encryption and

decryption, multicast message acknowledging, and payload checksum computation. The NMS message header is fixed to 27 bytes, including message type and version, source and target addresses, session ID, message ID, hop count, TTL, message reliability, queue status, metadata length, and data length.

DisService message headers have variable size, and all include message type, target node IDs, sender node ID, and session ID. Depending on the type of message, different fields may be present. Data messages carry an extra type field, group ID, publisher node ID, message sequence, chunk ID, object ID, instance ID, annotated object ID, annotation metadata, tag, client ID, client type, Multipurpose Internet Mail Extensions type, message length, fragment length, history window, priority, expiration time, and acknowledgment request. All IDs (with the exception of the client ID) and metadata are variable size strings, therefore their impact on bandwidth usage can vary significantly.

DisService subscribers are responsible for choosing how messages should be delivered. When a node subscribes to a group, it can choose whether messages should be delivered sequentially and/or reliably. Typically, sequencing increases latency while reliability increases bandwidth usage. Therefore, DisService clients are free to make tradeoffs to improve their efficiency while maintaining their functional requirements. For example, a node could prefer unreliable but sequenced delivery of Blue Force messages, while requiring reliable but unsequenced delivery of sensor reports. One current limitation of reliable subscriptions is that DisService nodes will not retrieve messages published in the group before the node subscribed to it. However, clients can still request specific messages, if they so desire, or issue queries to retrieve old messages.

The burden of requesting missing messages in reliable subscriptions (or parts of messages, called *fragments*) is left to subscribers that can generate missing fragment requests (MFRs) that other nodes can reply to if they have the requested messages or fragments in their cache. This design choice is consistent with the possibility within DisService that different subscribers require different delivery semantics for the same messages, completely unbeknownst to the original publisher. Furthermore, this approach exploits opportunistic listening to the fullest, where nodes may be able to obtain missing fragments from any or even more than one peer, not only the publisher.

DisService supports a variety of message dissemination and forwarding algorithms, including reliable flooding, probabilistic (epidemic) strategies, and heuristic strategies that can be selected based on the type of information being disseminated, network topology, and application goals. Reliable flooding is an expensive algorithm, but it is appropriate for high-priority messages that must be delivered reliably. Probabilistic protocols use attributes such as the number of neighbors of

nodes to determine the dissemination paths while maintaining bandwidth consumption under control. Heuristic protocols further enhance probabilistic approaches by exploiting domain-specific knowledge to guide the probabilistic dissemination (e.g., what groups nodes subscribe to, how many messages they have in their cache); however, such solutions typically have greater network footprints due to the additional information nodes need to exchange. In our experiments, we configured DisService not to relay messages and to use a purely probabilistic message-forwarding algorithm at the gateway nodes. This was in combination with some optimizations developed as part of the research conducted by the RTG that is described in Section 6.2.1.

DisService also provides an alternative approach to handle the dissemination of large messages, such as multimedia objects (pictures, video, etc.), called *chunking* or *information layering*. Chunking breaks large messages into independently intelligible, independent pieces (i.e., chunks) that are scattered across the network. When a node wants to retrieve a message, it queries its peers to find available chunks, retrieves one or more of them, and reassembles them locally to reconstruct a message. The more chunks that have been retrieved, the more faithful it is to the original. For example, large images can be broken up into multiple lower resolution images that are individually usable and available; when combined, chunks will recreate the full-resolution image.

6.2.1 Improvements to DisService

One of the goals of our efforts in IST-161 is to improve the bandwidth efficiency of DisService by avoiding forwarding messages that would result in duplicates. In the Anglova scenario, this was a concrete possibility because each company has two gateway nodes that forward messages from the company network to the overlay network and vice-versa.

To avoid both gateway nodes forwarding the same messages, we added a configuration option to specify messages that the DisService node should not forward based on their source address. We also introduced a random delay in the forwarding process and improved it by keeping track of messages already forwarded by other nodes that the local node can therefore skip during its forwarding phase. On every gateway node, we set the value of the aforementioned option to the addresses assigned to the company and the overlay network interface of the other gateway node in the company. Moreover, we configured a random message-forwarding delay on all gateway nodes. These changes have proved to work well in combination to reduce the probability that forwarding results in duplicate messages.

Another improvement we made for this effort was directed at improving the process DisService uses to generate and serve MFRs. We modified the way information is encoded in MFRs to reduce their size and added an option to further limit their size to fewer bytes than a configurable threshold. This change was in response to the observation that MFRs constituted most of the overhead in DisService traffic and that only a small fraction of them could be retransmitted, due to the number of missing messages increasing as the experiment progresses and the limited resources available in the network.

We then worked on optimizing the decision of whether to serve a specific MFR or not. We made the decision probabilistic, based on the number of neighbors of the node issuing the request (a piece of information that we added to the MFR), so that requests from nodes with a small number of neighbors are served with higher probability. We then introduced the following optimizations:

- When generating an MFR, a node does not request messages or fragments that another neighbor has recently requested.
- When deciding whether to reply or not to an MFR, a node does not serve messages or fragments from the same message that was recently retransmitted by another neighbor.
- When deciding whether to reply or not to an MFR, a node always responds to requests for messages or fragments for which it is the publisher, unless another node has already retransmitted them recently.

All these changes have proved successful in further reducing the bandwidth consumption of DisService, resulting in greater message delivery ratios (DRs) in our experiments.

6.2.2 DisService's Future

Several efforts will be directed towards reducing the network overhead of DisService messages. Optimizations will include removing the NMS encapsulating layer, removing unnecessary fields from message fragments and retransmissions, and adding message header compression.

Other efforts will be aimed at improving the channel's bandwidth usage, with the goal of reducing latency of retransmissions and not depending on correct bandwidth configuration inputs from the network administrators or the users. Another goal of these efforts will be to improve DisService efficiency in the presence of channels with fluctuating bandwidth availability or when the underlying network technology might change at runtime.

We also plan to support additional options when clients subscribe to a group. One such option will allow clients to restrict the reliability of the subscription to the *N*

last messages published by each subscriber, where *N* is application controlled. Another option will enable the client to request the retrieval of up to *M* messages published before the client joined the group, with *M* also being specified by the subscribing application (a special value will allow the client to request the retrieval of all messages ever published into the group and a node-level configuration will be added to restrict such value for clients, if needed).

Finally, we aim at improving the Application Programming Interface (API) that controls the retrieval and publication of chunks. The enhanced API will allow publishers to control the compression level of chunked media and subscribers to retrieve all missing chunks without having to request each single chunk individually. When subscribers request the retrieval of any extra chunk, we will introduce an optimization that enables DisService to request the chunk with the greatest number of fragments already available. This potentially can reduce bandwidth consumption for some use cases.

Currently, there are no plans to standardize DisService to enable wider use in the NATO community.

6.3 NDN

ICN represents a clean slate approach that does not depend on stable IP-based network connections to retrieve information and provides the tools to tailor the robustness of the information collection through different mechanisms for signaling redundancy and intermediate data storage (caching). ICN also seamlessly handles the change in traffic patterns for data meant for a single receiver and data meant for a group of receivers. This is also the case for the change in traffic patterns from a fairly stable network to a network that experiences intermittent connectivity.

Advantages of such an approach are robust network architecture, simple interface between the applications and the supporting network infrastructure, and simpler network management and control. The purpose of ICN is to make it easier to find relevant information in the network in a scalable manner. These benefits are also valuable for military operations. These characteristics demand a closer examination of ICN to determine whether this technology is interesting and relevant as a candidate architecture for future heterogeneous military networks.

Among the several ICN proposals with varying degrees of maturity, NDN represents a mature and interesting implementation based on content-centric networking—a concept that gained considerable attention from the perspective of military applications (Refaei and Afanasyev 2018; Burke et al. 2018; Wikipedia 2024; Named Data Networking 2024).

6.3.1 Advantages of NDN in Military Environments

NDN is built on the following two simple basic primitives: the request for specific content and the response with the matching data. The two packets that perform these primitives are called Interest and Data packets. These are similar to the request and response primitives for service-oriented middleware architectures. For any content to be transmitted, the consumer must issue an Interest that directly specifies the name of the content, thereby achieving host agnostic communication. This characteristic allows NDN to exhibit several advantages that can address the challenging characteristics of military networks.

The NDN forwarding daemon consists of multiple components as shown in Figure 33. A face for each type of transmission medium that is used (e.g., Bluetooth, Ethernet, application). A content store which functions as a local cache to store data packets, the Forwarding Information Base functions as a local routing table used by the daemon to decide on which interface to forward specific Interests. A Pending Interest Table keeps track of outstanding Interests, prevents duplicate transmission of Interests, and protects against Interest flooding.

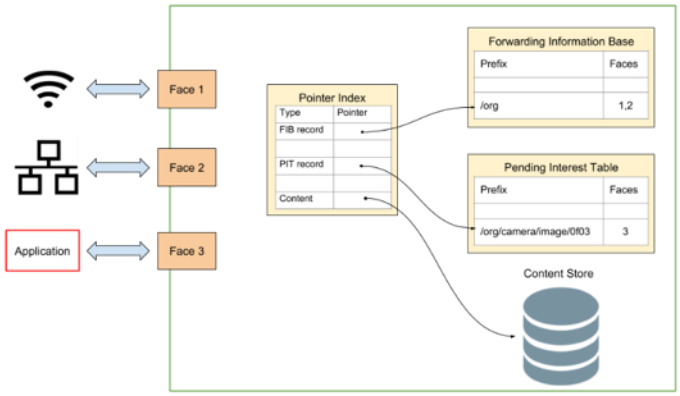


Figure 33. Overview of the forwarding engine in an NDN node.

6.3.1.1 Disruption Tolerance

NDN communication primitives decouple information producers and consumers and interaction between communication endpoints does not depend on a stable end-to-end connection due to its independence of the IP. In fact, if at some point in time an Interest message gets to the producer or a node that has cached the content, the responding Data packet travels its way back to the consumer and can be stored at each hop. If the Data message is lost due to broken links, the consumer can simply reissue the Interest after a time-out.

This approach provides a simple mechanism for reliable communication. In fact, after the first Interest, each successive Interest for the same content has a higher

likelihood of succeeding since the content is now likely stored in a content store closer to a consumer.

6.3.1.2 Node Mobility

Node mobility is a significant challenge for IP due to the dual nature of IP addresses being identifiers and locators. Instead, NDN copes significantly better with mobility and can be addressed by separately addressing the two main mobility scenarios (i.e., consumer mobility and producer mobility).

Consumer mobility is handled automatically by NDN. In fact, a consumer that has moved simply asks for new content in the standard way by issuing an Interest. Each time an Interest is issued, its transmission builds a path from the new position to the first node that holds a copy of the content, resulting in a process unaffected by the mobility of the consumer.

On the other hand, producer mobility is more challenging. In fact, when a mobile producer (MP) moves to a new position, a consumer might be unable to fulfill its Interest since the path to the MP might become invalid. To address this problem, the MP might accelerate the creation of a new Interest path by generating a "breadcrumb trail" to a rendezvous node to enable Interests to trace content at its new position (similar to the mobile IP approach). If the adoption of rendezvous nodes is unsuited for the specific use case, consumers and router nodes can also adapt their behavior for this situation by choosing more aggressive Interest flooding strategies to find the new location of the MP.

6.3.1.3 Multicasting and Multi-Homing

The NDN architecture seamlessly handles the dissemination of content to a single consumer or to a group of consumers. NDN manages all the interfaces over which data is sent or received via faces. Each face can be connected to higher-layer entities, such as an application, a physical network interface, or even a virtual link.

Furthermore, NDN overcomes the well-known problems that IP architectures present with multi-homing. In fact, since Interest routing is based on the content name, NDN works out of the box in heterogeneous network environments with multiple channel technologies and is capable of aggregating Interests received from different faces so only one Interest per content is sent over a shared communication link. Furthermore, nodes with multiple network interfaces are also profitable in terms of caching. When a Data packet is sent back to the consumer, NDN can leave a copy of the message in the local content store of all nodes along the reverse Interest path. Popular content can then be made automatically available close to its consumers.

6.3.2 Challenges of NDN in Military Environments

While NDN seems to offer some structural benefits for tactical environments, several challenges already have been identified such as information security, information naming, and reliability. Some of these challenges are fundamental to the adoption of NDN, thus several solutions already have been studied and addressed through different approaches.

6.3.2.1 Security

The NDN security architecture is very flexible but raises some challenges—particularly for mobile military networks. In NDN, information security is placed on the content. Each content chunk is signed and optionally encrypted. This is flexible but introduces overhead. The public key of the signer, a certificate for that public key, or a pointer to them must be sent with the packet and a trust chain must be in place. This can be problematic for low data-rate mobile connections. Traffic analysis is also a challenge. Out of simplicity, the Interests are sent in human readable text due to the designed naming schemes in most of the current experiments and evaluations with NDN but can be encrypted optionally (adding more complexity to the naming scheme). This is not adequate for military networks putting restrictions on the design of the naming schemes. The security model does not handle network security. There must be functions in place that perform network security. Some traditional solutions for mobile military networks (e.g., preloaded keys and link-level encryption) can solve some of these problems, but more research is needed.

6.3.2.2 Strategy Layer

The strategy layer is one of the components within the NDN architecture. The flexibility introduced by this layer enables one to tailor the forwarding policy of Interest and Data to meet application specific requirements. In fact, NDN allows for the implementation of a wide range of Interest dissemination strategies, from naive ones that simply mimic standard IP communication mechanisms (e.g., unicast, multicast) to more sophisticated ones that continuously analyze their performance and self-learn how to improve their performance. This is a powerful component within the NDN architecture that must be better studied and tuned to achieve its full potential.

For instance, strategies can be based on overheard traffic so that Interests might be sent on one or more faces to rapidly relocate the new position of the producer node. However, strategies that introduce high redundancy in the Interest distribution should be adopted carefully for networks where link capacity is limited.

Finding the best trade-off between complexity, efficiency, and robustness is crucial to exploit the flexibility of the strategy layer to adapt it to the network properties and communication requirements.

6.3.2.3 Reliability

NDN does not provide an integrated solution for reliable communication. As a result, unlike the TCP/IP model where reliability functions are provided by some transport layer protocols, the responsibility to recover a lost message in NDN relies by default on higher layers like protocols and applications on top of the NDN forwarding daemon. However, NDN also leaves the possibility open to solve this on the strategy layer of NDN (e.g., by adding/implementing a specific Interest dissemination strategy for reliable transmission).

The NDN interface between applications and the network layer allows applications to directly manage reliability and communication robustness by having control over the disseminated Interest messages. For example, an application can reissue a new Interest if the previous Interest for the same content was not resolved within a certain time-out period.

However, such mechanisms represent an important challenge similar to the definition of an effective strategy layer. In fact, Interest time-outs, Interest retransmission, and their multicast dissemination can improve data DR and QoS but can be costly in terms of network resources. Therefore, it is necessary to find the best tradeoff between Interest dissemination and network resource consumption to implement reliable communication within NDN.

6.3.2.4 Performance Tuning

Since NDN is a relatively new technology, the performance tuning of NDN applications is an aspect that still needs to be thoroughly investigated. For instance, cache replacement strategies for content stores and values for several time-outs (including Interest retransmission) are critical parameters for application performance and robustness (as they not only influence delay and jitter of the requested Data but availability as well). Cache replacement strategies of classic IP architectures are a mature research field and this knowledge can be reused here. However, more work is needed to learn what strategies are best to use for what routing strategy and different data types. Furthermore, applications might reissue an Interest if no Data has been received within a certain time-out. This parameter must be properly tuned to fit traffic requirements and network properties. More experience is needed that can result in guidelines for setting the parameters.

Chunk size is another very relevant performance-related research topic in NDN. Which maximum size of data elements should the consumer be able to ask for in

one Interest? The NDN architecture promotes the maximum transmission unit size of the transmission medium used as the preferred size of data chunks. The advantage of this is a very responsive network. The Interest can be routed a different way for each data chunk and thus be able to handle mobility and avoid network congestion (do flow control) very quickly. This could come at the cost of a larger overhead when sending Interest packets and Data packets (that include a security certificate) for each tiny chunk of data. Larger chunk sizes (e.g., whole documents or videos) reduce the overhead but also reduce some of NDN architecture's qualities.

6.3.2.5 Congestion Control

NDN adopts a connectionless communication model that is radically different from the connection-oriented TCP/IP model. Hence, it cannot easily leverage the immense knowledge base built over decades by researchers investigating end-to-end congestion control solutions such as those designed for TCP. As a result, NDN must leverage other congestion management solutions including receiver-based congestion control, hop-by-hop congestion control, and hybrid methods.

As of this writing, hop-by-hop congestion control methods (based on the automatic slowdown of Interest forwarding at the router level in case of overloading) seem to be the most proposed solution. However, those methods typically do not consider numerous important factors (e.g., the influences that caches have on transport traffic, multi-path transport). Overall, congestion management represents one of the aspects that is most in need of further investigation to foster practical adoption of NDN-based solutions.

6.3.3 IST-161 Specific Developments for NDN

NDN defines a pull-based communication model which contrasts with the more commonly adopted communication models in military environments, such as publish–subscribe, that have been studied within this RTG's research. To evaluate the effectiveness of NDN, we decided to define a few specific mechanisms to mimic publish–subscribe-like information dissemination for an NDN-based setup. We defined a specific naming scheme and implemented a proactive interest dissemination strategy with the assumption of isochronous (and known) generation of content.

We focused only on the dissemination of BFTs. This arguably represents the most challenging component of a tactical application for NDN, since BFT information is periodically published by each node and should be received by all other nodes within the same network. Therefore, each consumer must issue a different Interest for each producer.

With the assumption of isochronous generation of BFTs, it is possible to mimic a publish–subscribe-like communication model within NDN. In fact, each time producers generate a new content item, they will immediately receive Interests from all consumers for the newly generated data, after which they forward the data packet containing the generated information. Moreover, this mechanism can also be applied in two different ways. The first is to proactively disseminate Interests as soon as the data is available. The second is to disseminate the Interests before the content is produced to trigger immediate dissemination of the Data message as soon as it is generated due to the pending Interests (Campioni et al. 2019a).

The naming scheme is another crucial component to mimic the publish–subscribe-like communication model within NDN. In fact, since all content items are uniquely named, consumers must issue Interests related to the latest BFT data produced. Hence, we decided on the following naming scheme:

```
/anglova/<topic>/<company_id>/<node_id>/<seq>
```

where anglova is the name's root, topic identifies the type of content (in our case BFT), company_id is the identifier of the Anglova company in which the node is deployed, node_id is a unique identifier associated with each node, and seq is an incremental value that distinguishes each data item a producing node has published. This scheme allows the middleware to uniquely identify the data published by each node while allowing each consumer to request it without the need of a content discovery mechanism. For example, if a consumer needs the 16th BFT data item that is published by the node with id 4 deployed in the second company, it can simply send an interest specifying the following name:

```
/anglova/blue_force/2/4/16
```

Another aspect we focused on during the research on NDN was to improve our proactive Interest dissemination strategy with mechanisms to improve both network usage and DR. In fact, due to the nature of BFT dissemination (all nodes generating and disseminating at the same time) and the layout of the network, the high number of Interest required to satisfy all consumers can cause network congestion in large network environments.

To mitigate this problem and exploit interest aggregation, we developed a back-off–based Interest scheduler. When a consumer produces a large number of Interests, the scheduler divides the Interests in buckets of a fixed size that are sequentially sent after a random back-off time from one another. This way, each node will send only a limited amount of Interests at a time, thereby reducing traffic spikes that can cause congestion and packet drops.

Moreover, since the BFTs represent a common Interest by all consumers, the bucketing process is followed by an Interest shuffling process which is designed to exploit NDN caching. More specifically, the randomizing procedure increases the likelihood that nodes request different content in the same time window. This way, when an Interest is being queued to be forwarded to other nodes, NDN recognizes if the same Interest has been previously requested by another node and discards the Interest from the outgoing queue to avoid redundancy.

6.4 DAF, an Interface between olsrd2 and nrlsmf

As described by Macker (2012), SMF is a smart flooding technique for IP multicast packets in mobile ad hoc networks (MANETs) that can operate with either nonreduced or reduced relay set forwarding. In the former mode, also known as classic flooding, a network node forwards incoming multicast packets that have not been forwarded by that node before and meet other eligibility criteria, such as having a TTL greater than 1 and a multicast destination address that is not link-local. Although duplicate packet detection should prevent "packet storms," classic flooding may still cause many redundant copies of an IP multicast packet to coexist in the MANET. Reduced relay set forwarding aims to reduce the number of redundant multicast packets. Several distributed algorithms have been devised for this purpose; three examples of these can be found in Appendixes A through C of RFC 6621 (Macker 2012).

Appendix B of RFC 6621 describes source-based multipoint relay (S-MPR; Macker 2012). This algorithm leverages information from the OLSRv2 unicast MANET routing protocol (RFC 7181; Clausen et al. 2014) when an instantiation of that protocol is co-located on the nodes. It uses the current set of symmetric neighbors maintained by OLSRv2, as well as the subset of symmetric neighbors that are currently (flooding) MPR selectors of the node.

olsrd2 is an open-source implementation of OLSRv2. The main driver behind its development is IST-161's German partner FKIE. *nrlsmf* was developed by the U.S. Naval Research Laboratory and is an open-source implementation of SMF with support for S-MPR. The neighbor information that is required for the S-MPR algorithm of SMF can be obtained from olsrd2 through a telnet interface. This information is formatted as American Standard Code for Information Interchange (ASCII) strings. nrlsmf has an interface through which an external process can provide symmetric neighbor and MPR selector messages. However, this interface is based on the use of a Unix socket, and the messages consist of a short ASCII text header followed by a tail of concatenated MAC addresses in binary form.

To allow the use of the relevant information from olsrd2 by nrlsmf, the IST-161 RTG developed DAF, an interface and format conversion program. DAF runs as a separate process alongside olsrd2 and nrlsmf. It continually queries olsrd2 on its telnet interface for the latest symmetric neighbor and MPR selector sets. It adapts the responses to the format that is digestible by nrlsmf and sends it to nrlsmf's Unix socket. Additionally, it logs the information obtained from olsrd2 (see Figure 34).

```
# response set 164
00:0c:29:dd:02:13 true symmetric
00:0c:29:93:02:0a true symmetric
00:0c:29:c0:02:01 true symmetric
00:0c:29:60:02:14 true symmetric
00:0c:29:23:02:0b true symmetric
00:0c:29:77:02:0c true symmetric
00:0c:29:a3:02:15 true symmetric
00:0c:29:fe:02:03 true symmetric
00:0c:29:ad:02:16 true symmetric
00:0c:29:7e:02:0d true symmetric
00:0c:29:02:02:04 true symmetric
00:0c:29:77:02:0e true symmetric
00:0c:29:ef:02:17 true symmetric
00:0c:29:64:02:05 true symmetric
00:0c:29:ea:02:0f true symmetric
00:0c:29:56:02:18 false symmetric
00:0c:29:f4:02:10 true symmetric
00:0c:29:6a:02:06 true symmetric
00:0c:29:c8:02:07 true symmetric
00:0c:29:a7:02:11 true symmetric
00:0c:29:e2:02:08 true symmetric
00:0c:29:ab:02:12 true symmetric
00:0c:29:8a:02:09 true symmetric
00:0c:29:94:05:08 true symmetric
00:0c:29:c0:05:03 false symmetric
00:0c:29:48:05:01 true symmetric
00:0c:29:74:05:05 true symmetric
00:0c:29:da:05:02 true symmetric
00:0c:29:88:05:07 true symmetric
00:0c:29:c2:05:06 true symmetric
# summary: 30 neighbors, of which 30 symmetric, of which 28 MPR selector
```

Figure 34. Local DAF information (neighbor node MAC, if the node is an MPR selector, and neighbor status) obtained from olsrd2.

6.5 Tactical Group Interoperability Bridge

6.5.1 Background

The Tactical Group Interoperability Bridge is a plug-in–based software solution capable of providing intercommunication between several open and proprietary communication protocols. This component was designed with the objective of forwarding and controlling messages between different publish/subscribe solutions maintaining a common topic structure, enabling filtering and forwarding of messages.

The Interoperability Bridge achieves two primary objectives. First, it enables communication between clusters that base their communication on different protocols. Second, it enables clusters to remap their communication over middleware specifically designed to provide communication in a constrained environment. This can be particularly beneficial in tactical operations considering that enterprise solutions typically do not use multicast whereas solutions designed for tactical environments do.

The development of the Interoperability Bridge was based precedent work by the RTG that created a common plug-in–based infrastructure that the group used to evaluate the performance of several communication protocols in the Anglova scenario. The need for an Interoperability Bridge was highlighted when the group observed that even if several organizations would decide to communicate using a shared topic structure, communication was not easily attainable due to differences in the communication middleware that each organization would field. This called for a simple and extendable Interoperability Bridge capable of bridging different communication protocols, provide filtering utilities and topic manipulation to prevent loops and other undesired behaviors.

6.5.2 Architecture

The Tactical Group Interoperability Bridge architecture is based on a plug-in model that easily enables the introduction of a new communication technology. Each supported communication solution must implement a simple API that defines methods to send and receive messages. This API is then encapsulated in a plug-in by means of the PF4J library.* Each plug-in also hides protocol-specific initialization and configuration steps that are necessary to correctly use a solution.

A bridge endpoint can instantiate one or more plug-ins, enabling communication between them. Whenever a message is received from a plug-in, the Interoperability Bridge forwards the message to the other plug-ins that each provide protocol-specific functionalities and data sanitation to make the messages compatible with the supported communication middleware before forwarding the message.

The Interoperability Bridge can be configured to send reports about exchanged messages to a Stat Server that can be used to run diagnostics and evaluate performance.

Figure 35 shows a possible deployment of three clusters connected by different plug-ins. The bottom left cluster used DisService internally and GDEM to communicate with other clusters. The bottom right cluster uses DisService

* Available at https://pf4j.org/

externally and NATS internally. The top cluster uses Kafka to communicate within the cluster and DisService and GDEM to communicate with the two other clusters.

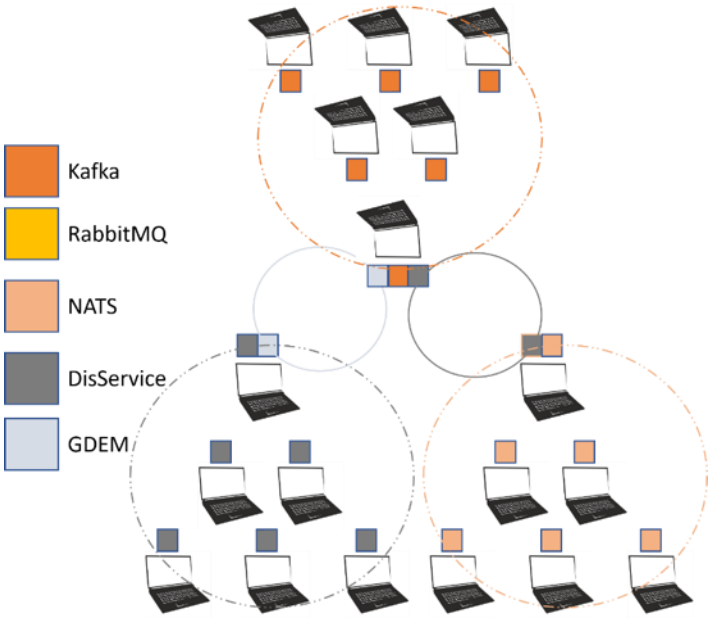


Figure 35. Tactical Group Interoperability Bridge deployment example.

The Tactical Group Interoperability Bridge was evaluated during CWIX 2021. An experiment showed that it was possible to bridge messages between DisService and GDEM.

6.6 Serverless Chat Using GDEM and DisService

During the CWIX 2021 exercise, members of the RTG executed an experiment for using multiuser chat in a tactical network over some of the best performing group communication protocols we evaluated. We used an existing implementation of eXtensible Messaging and Presence Protocol (XMPP)-overlay (XO) (Robert et al. 2010) for this experiment. We created XMPP-XO transports that used both GDEM and DisService as group communication protocol. The open-source implementation of XMPP-XO uses the NORM protocol (Adamson et al. 2009) by default.

Tactical multiuser chat is currently targeted for FMN Spiral 5 (Tactical Communication and Information Systems [TACCIS] only) and 6 (TACCIS and Operational Communications and Information Systems [OPCIS]). FMN Spiral 5 will use JDSS, an extension of the Joint Soldier Tactical System, to exchange tactical chat messages. There is a need for FMN Spiral 6 to interface with OPCIS

from TACCIS. OPCIS chat is based upon the XMPP protocol,* hence we looked at an option to use XMPP in TACCIS as well.

The experiments targeted a scenario where a tactical coalition network was connecting nodes of different nations. The XMPP-XO software running on each of the participating network nodes used the same setup. All nodes were using the NORM, DisService, or GDEM transport. Figure 36 shows the network architecture for the experiment executed between the Netherlands network (NLD) and the German network (DEU).

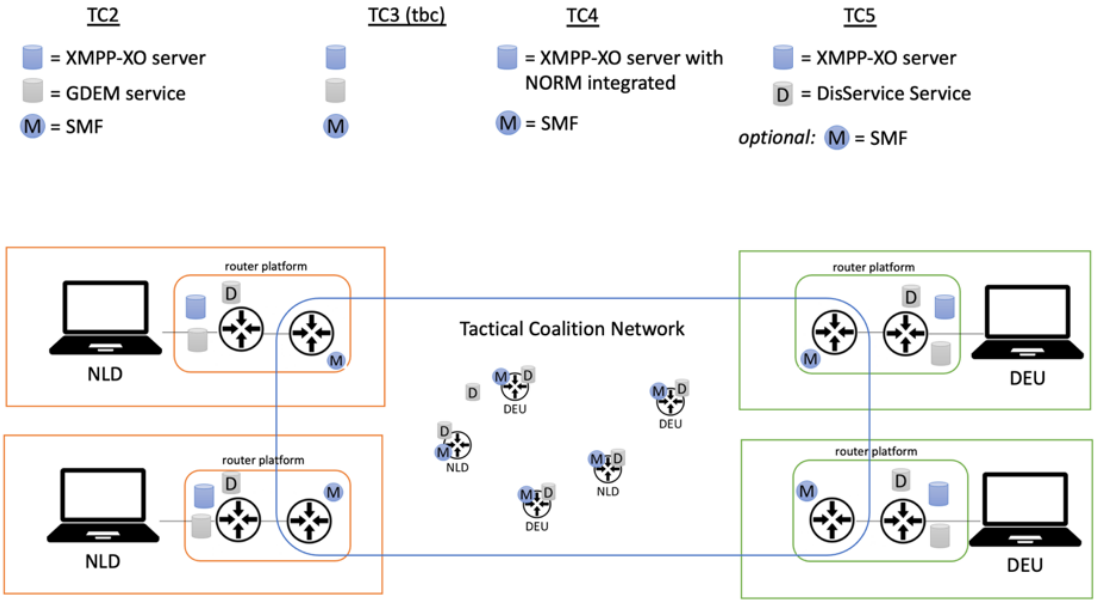


Figure 36. Example of a CWIX tactical chat experiment setup.

All three protocols were successfully used in experiments from a functional perspective during the CWIX exercise. No comparative (performance) studies between the protocols have been done.

7. Protocol Evaluation

7.1 Overview

In this section, we discuss the IST-161 RTG's evaluation of communication protocols and middlewares for group communications. The evaluation is largely based on the results of extensive experimentation using the IST-161 emulation environment and the Anglova scenario.

* Available at https://xmpp.org/

We first evaluated the performance of selected communication protocols in a single company using 802.11 as a baseline underlying network technology. We then switched to a multicompany scenario and adopted the more sophisticated SCB network technology for increased robustness and scalability, experimenting with both flat and hierarchical (overlay-based) network topologies. Next, we evaluated how the best performing protocols evaluated in the former two scenarios performed under degraded communication conditions, namely a reduced signal-to-noise ratio due to adversarial jamming.

In the experiments, we focused on broker-based (NATS), reliability-focused (ZEROMQ-NORM), and distributed (DisService and GDEM) communication protocols. Finally, we evaluated the performance of NDN with extensions developed specifically within the IST-161 RTG activity.

7.2 Company Level Evaluation

The first set of experiments were conducted to measure the performance of several selected protocols in the context of the Anglova scenario. The results have been published in Suri et al. (2019b). At the time of these experiments, GDEM and NDN were not yet included in the experiments.

When conducting these experiments, the RTG found most of the selected protocols were not suited for adoption in a tactical environment. We therefore chose to focus on a subset of the most promising protocols for further experimentation, obtained important feedback to guide us in further development of DisService, and further expanded the selection of protocols to consider eventually including GDEM to overcome the issues encountered in the experiments.

Experiments were used to identify performance using three measurements: DR, latency, and bandwidth usage. DR reports on the ability of the protocol to successfully deliver information to the intended recipients. A DR of 100 implies that every message that was supposed to reach a recipient did so. Latency measures the delay (in milliseconds) that occurred for all messages that were delivered. Bandwidth usage determines how much network bandwidth was utilized by the protocol or system during message delivery.

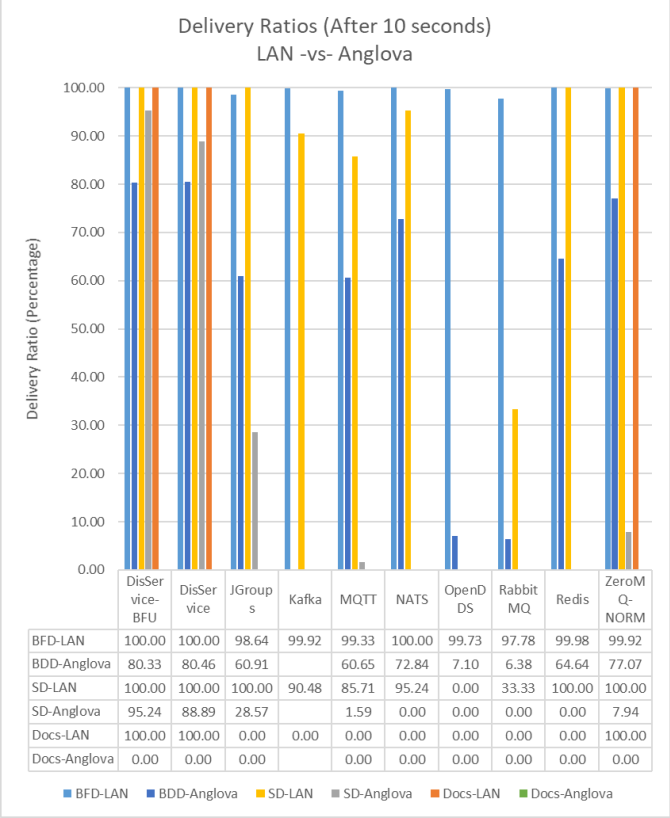
7.2.1 DR

The first set of results report on the DR. Three different types of messages were part of the experiment: BFD, SD, and Docs. BFD consists of position and status information of each node and is sent to every other node. This information is transmitted every 10 s and is 128 bytes in size. Given that the scenario runs for 20 min and each of the 24 nodes generates 6 messages each minute, there are a total

of 24 × 6 × 20 = 2880 messages generated. On the other hand, SD is generated by one node (the sensor gateway) once every 3 min and is 128,000 bytes in size. But this is sent only to three nodes, so there is a total of 3 × 7 = 21 messages. Finally, Docs are generated by one node (the notional Headquarters) once every 6 min and are 512,000 bytes in size. These messages are only sent to two nodes. A perfect DR of 100 implies that all these messages are received by their intended recipients.

DR results were collected with three different cutoffs with respect to time: –5 s, 10 s, and end of scenario. These cut-offs represent different tolerances for latency. For example, the policy might be that BFD with a latency over 10 s is too old and hence unusable. The results in Figure 37 show the DR with a cutoff of 10 s (i.e., a message is not counted if it arrived more than 10 s after being transmitted).

For each type of data (BFD, SD, and Docs), Figure 37 shows the DR for nine different protocols (i.e., DisService, JGroups, Kafka, MQTT, NATS, OpenDDS, RabbitMQ, Redis, and ZeroMQ-NORM). Furthermore, the results are broken down into two subcategories (i.e, local area network [LAN] and Anglova). The LAN results act as a baseline and show the performance that was observed when the protocols were running on an Ethernet LAN, whereas the Anglova results show the performance that was observed when the protocols were running under the Anglova scenario.



	DisService-BFU	DisService	JGroups	Kafka	MQTT	NATS	OpenDDS	RabbitMQ	Redis	ZeroMQ-NORM
BFD-LAN	100.00	100.00	98.64	99.92	99.33	100.00	99.73	97.78	99.98	99.92
BDD-Anglova	80.33	80.46	60.91		60.65	72.84	7.10	6.38	64.64	77.07
SD-LAN	100.00	100.00	100.00	90.48	85.71	95.24	0.00	33.33	100.00	100.00
SD-Anglova	95.24	88.89	28.57		1.59	0.00	0.00	0.00	0.00	7.94
Docs-LAN	100.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
Docs-Anglova	0.00	0.00	0.00		0.00	0.00	0.00	0.00	0.00	0.00

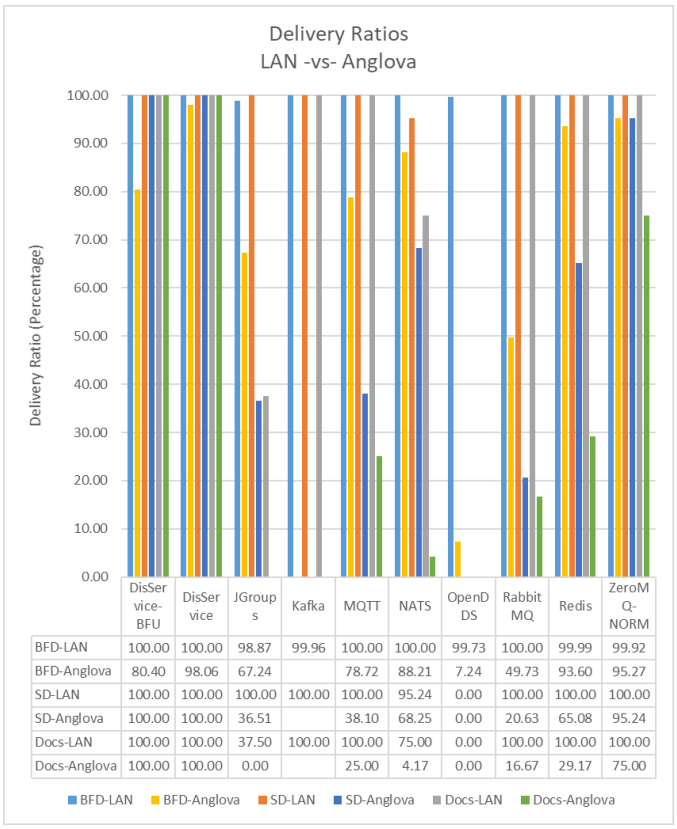
Figure 37. DRs with 10-s cutoff.

Two different sets of results are presented for DisService. The first set (denoted DisService-BFU) contains results when BFD was transmitted using an unreliable (but sequenced) delivery service, whereas the second set (denoted DisService) contains results when all three data types were transmitted using a reliable delivery service.

Several interesting observations can be made based on this data. All the protocols handle BFD well when running in a LAN environment. Most of the protocols, except OpenDDS and RabbitMQ, also handled SD well when running in a LAN environment. Finally, only two protocols, DisService and ZeroMQ-NORM, were able to handle Docs when running in a LAN environment. All results only count something as delivered if the delivery latency is less than 10 s.

The performance results are quite different when considering the Anglova environment. Kafka was the worst protocol and was unable to deliver anything. None of the protocols were able to deliver Docs within a 10 s interval. DisService was able to do a reasonable job delivering SD. DisService, ZeroMQ-NORM, NATS, Redis, JGroups, and MQTT were able to deliver between 60% to 80% of BFD.

Figure 38 shows the DRs at the end of the scenario (i.e., after running for 20 min). In this case, the protocols have had additional time to request retransmission of missing packets, thereby being able to deliver more of the SD and Docs. For BFD, DisService performs best (98.06%), followed by ZeroMQ-NORM (95.27%), Redis (93.60%), NATS (88.21%), and MQTT (78.72%). For SD, DisService is again the best (100%), followed by ZeroMQ-NORM (95.24%), NATS (68.25%), and Redis (65.08%). For Docs, DisService is again the best (100%), followed by ZeroMQ-NORM (75%).

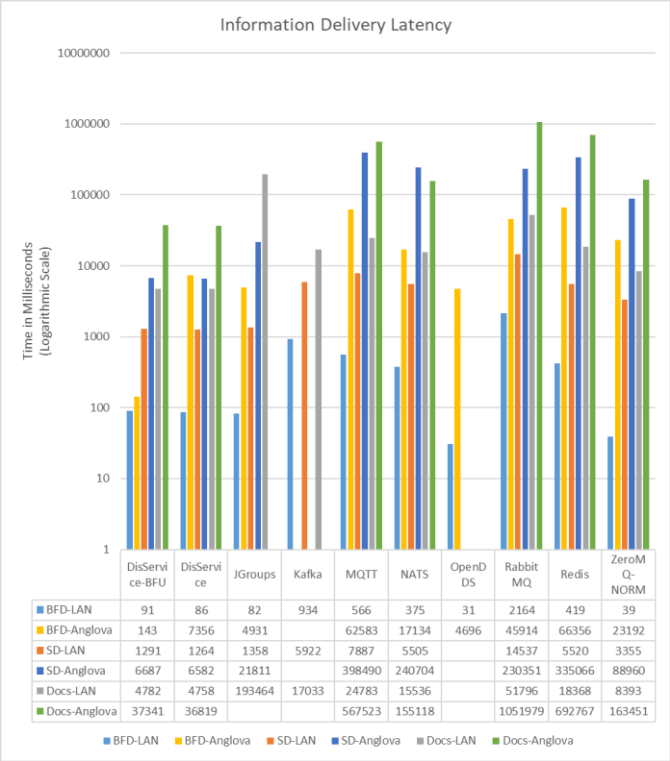


	DisService-BFU	DisService	JGroups	Kafka	MQTT	NATS	OpenDDS	RabbitMQ	Redis	ZeroMQ-NORM
BFD-LAN	100.00	100.00	98.87	99.96	100.00	100.00	99.73	100.00	99.99	99.92
BFD-Anglova	80.40	98.06	67.24		78.72	88.21	7.24	49.73	93.60	95.27
SD-LAN	100.00	100.00	100.00	100.00	100.00	95.24	0.00	100.00	100.00	100.00
SD-Anglova	100.00	100.00	36.51		38.10	68.25	0.00	20.63	65.08	95.24
Docs-LAN	100.00	100.00	37.50	100.00	100.00	75.00	0.00	100.00	100.00	100.00
Docs-Anglova	100.00	100.00	0.00		25.00	4.17	0.00	16.67	29.17	75.00

Figure 38. DRs at end of scenario.

7.2.2 Delivery Latency

The next measurement is delivery latency (Figure 39). This figure uses a logarithmic scale for the Y axis due to the large range in observed latency. An important note about the latency measurements is that it only records latencies for delivered messages (i.e., a protocol may have a very low DR, but if the latency is low for the few messages that were delivered, the overall reported latency is also low). All latencies were measured in milliseconds.



	DisService-BFU	DisService	JGroups	Kafka	MQTT	NATS	OpenDDS	RabbitMQ	Redis	ZeroMQ-NORM
BFD-LAN	91	86	82	934	566	375	31	2164	419	39
BFD-Anglova	143	7356	4931		62583	17134	4696	45914	66356	23192
SD-LAN	1291	1264	1358	5922	7887	5505		14537	5520	3355
SD-Anglova	6687	6582	21811		398490	240704		230351	335066	88960
Docs-LAN	4782	4758	193464	17033	24783	15536		51796	18368	8393
Docs-Anglova	37341	36819			567523	155118		1051979	692767	163451

Figure 39. Information delivery latency.

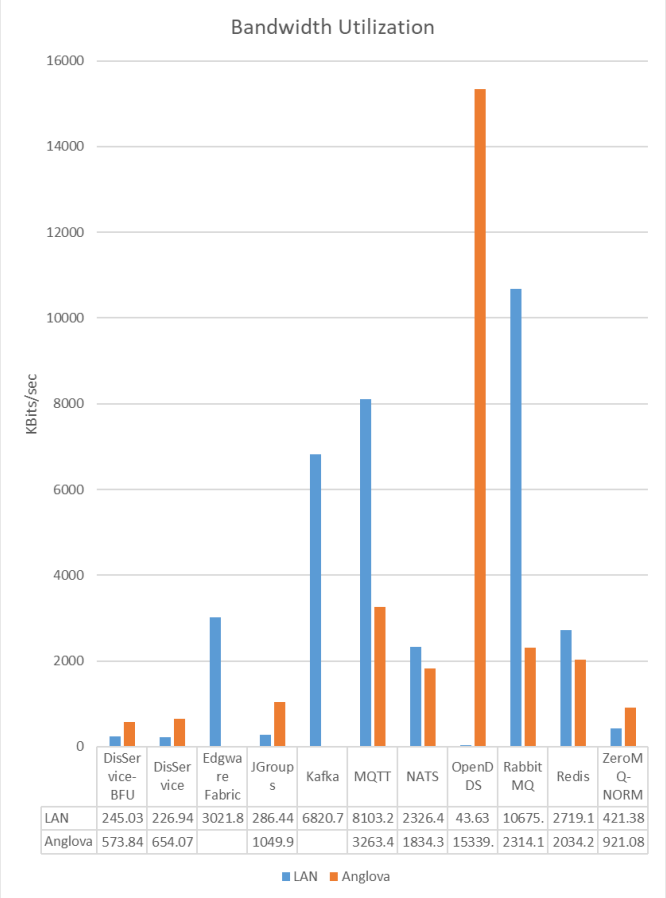
For BFD in a LAN environment, OpenDDS has the lowest latency (31 ms), followed by ZeroMQ-NORM (39 ms), JGroups (82 ms), and DisService (86 and 91 ms). In the Anglova environment, the variation of DisService that does not use the reliable service for BFD has the lowest latency (143 ms), followed by OpenDDS (4696 ms), JGroups (4931 ms), and the second variation of DisService (7356 ms). The two variations of DisService provide an interesting data point—when reliability is not enabled, the DR is 80.40% with an average latency of 143 ms, whereas when reliability is enabled, the DR goes up to 98.06% but the average latency also goes up to 7356 ms. These types of trade-offs are important to consider when designing solutions for information dissemination in tactical networking environments.

For SD in a LAN environment, DisService has the lowest latency (1264 ms) followed by JGroups (1358 ms), ZeroMQ-NORM (3355 ms), NATS (5505 ms), Redis (5520 ms), and Kafka (5922 ms). In the Anglova environment, DisService has by far the best performance (6582 ms), followed by JGroups (21811 ms) and ZeroMQ-NORM (88960 ms). However, the performance of JGroups must be discounted as it only delivers 36.51% of the messages compared with DisServce (100%) and ZeroMQ-NORM (95.24%).

For Docs in a LAN environment, DisService again has the lowest latency (4758 ms), followed by ZeroMQ-NORM (8393 ms), NATS (15536 ms), Kafka (17033 ms) and Redis (18368 ms). For Docs in the Anglova environment, unfortunately only DisService is able to deliver 100% of the Docs with a latency of 36819 ms. Second best is ZeroMQ-NORM, delivering 75% of the Docs with a latency of 163451 ms. The rest of the protocols have extremely low DRs for Docs.

7.2.3 Bandwidth Usage

The last measurement used was bandwidth usage (Figure 40). The results are shown in kilobits per second (kbps) for both the LAN and Anglova environments. In the LAN environment, OpenDDS has the lowest bandwidth usage (43.63 kbps), but this is deceptive because OpenDDS only delivered BFD and failed to deliver any SD or Docs. Discounting OpenDDS, the next best performer was DisService (226.94 kbps), followed by JGroups (284.44 kbps) and ZeroMQ-NORM (421.38 kbps). Once again, the results of JGroups are skewed as it does not deliver as many documents (37.5%) as ZeroMQ-NORM (100%). RabbitMQ had the worst performance in terms of bandwidth, using up 10675.51 kbps.



	DisService-BFU	DisService	Edgware Fabric	JGroups	Kafka	MQTT	NATS	OpenDDS	RabbitMQ	Redis	ZeroMQ-NORM
LAN	245.03	226.94	3021.8	286.44	6820.7	8103.2	2326.4	43.63	10675.	2719.1	421.38
Anglova	573.84	654.07		1049.9		3263.4	1834.3	15339.	2314.1	2034.2	921.08

Figure 40. Bandwidth usage.

The bandwidth measurements in the Anglova environment show a distinct increase, presumably caused by the retransmissions necessary due to the unreliability of the network. In this case, the DisService variant that does not use reliable transmission for BFD has the lowest bandwidth usage (573.84 kbps), followed by the regular variant of DisService (654.07 kbps). The next closest is ZeroMQ-NORM (921.08 kbps), but ZeroMQ-NORM only delivers 75% of the Docs.

One protocol that appears in this list is Edgware Fabric, showing a bandwidth consumption of 3021 kbps. However, complete results for Edgware Fabric could not be obtained as the system failed to scale to more than 6 nodes. Furthermore, it was not able to operate (even with just 6 nodes) in the Anglova environment.

To facilitate a better comparison of the bandwidth usage, Figure 41 presents the relative efficiency of each protocol. This is done by computing the efficiency for each protocol to deliver all the messages that it successfully managed to deliver, then measuring the bandwidth that was utilized for the delivery. For each environment, the protocol that performed the best is assigned a score of 100%, and the remainder of the protocols are measured in terms of how close they get to the best performance.

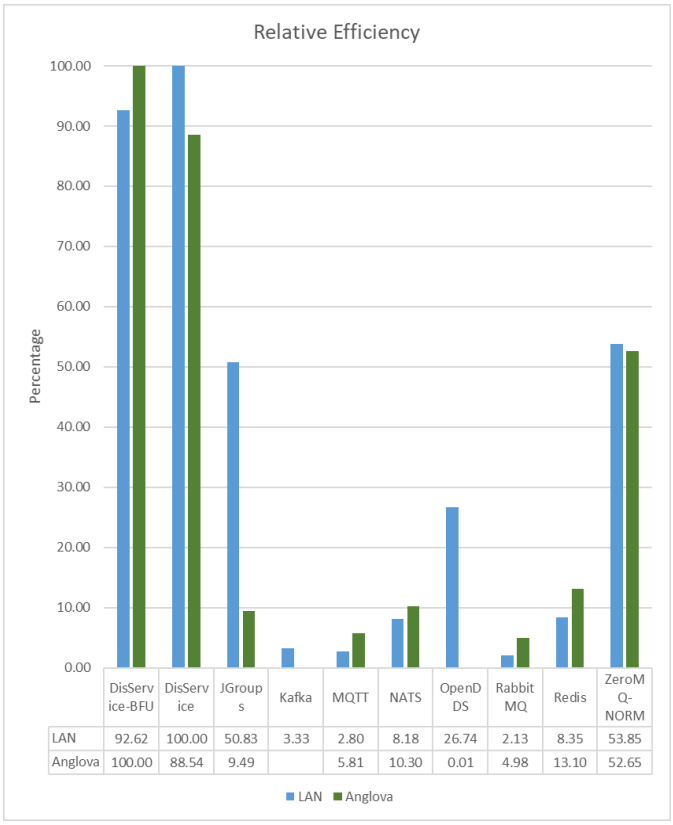


Figure 41. Relative efficiency (in terms of bandwidth).

In the LAN environment, the best performance is from DisService and hence it is assigned a score of 100. The next best performing protocol is DisService with unreliable delivery of BFD (92.62%), followed by ZeroMQ-NORM at 53.85%, and JGroups at 50.83%. Three protocols seem to perform the worst: RabbitMQ (2.13%), MQTT (2.8%), and Kafka (3.33%).

In the Anglova environment, the best performance is from DisService with unreliable delivery of BFD, and hence it is assigned a score of 100. The next best performing protocol is DisService (with reliable BFD) at 88.54%. Presumably, the required retransmission of BFD resulted in additional bandwidth that lowered the efficiency score. The third best performing protocol is ZeroMQ-NORM again at 52.65%. The worst performing protocol is OpenDDS at 0.01%. Given that OpenDDS performs relatively well in the LAN environment, we are currently investigating the observed performance issues with OpenDDS in the emulated network. While we expect some degradation in the performance, it should not be as bad as it is currently.

This relative efficiency measure does not take the delivery latencies into account. A protocol may have long latencies but could still be efficient in terms of bandwidth usage. Furthermore, it does not imply that the DRs are higher. For example, in the Anglova scenario, DisService with unreliable delivery of BFD has an efficiency score of 100% but its DR of BFD is 80.4%. On the other hand, DisService using the reliable delivery service for BFD had a DR of 98.06% but its efficiency score drops to 88.54%. Therefore, it is important to emphasize that this is not an all-encompassing measure; it is equally important to consider the other measures of latency and DRs.

7.3 Evaluating SCB Models

To test the scalability capabilities of communication protocols, we extended the experimental testbed to consider a multicompany communication scenario. The use of a SCB waveform was introduced at this point into the experiments of the group. SCB waveforms promise better scalability than traditional military waveforms and therefore are important to include when researching the performance of group communications in state-of-the-art military radio networks. Two approaches for modeling an SCB waveform in EMANE have been developed and compared (Marcus et al. 2020) by the IST-161 RTG: one based on SMF (Macker 2012), as described in Section 5.1.5.2; and one based on the precomputation of the network topology based on the principles of SCB, as described in Section 5.1.5.3.

The RTG introduced SCB as a transmission mechanism to the experiments since it is a promising concept to achieve the required transmission scalability for a

battalion-sized scenario. From the two aforementioned implementation approaches in EMANE (i.e., one based on SMF and the other using precomputed SCB topologies for the entire scenario), the latter was evaluated to fully allow us to model SCB's cooperative effects. The SMF-based approach allowed for dynamic SCB scheduling but resulted in more data transmissions than there would be in an actual SCB network.

We conducted three sets of experiments The first evaluated the performance of the SMF approach for SCB with 24, 48, 72, and 96 nodes and low, medium, and high levels of traffic load (where the high traffic load is near the maximum network capacity.) The second experiment was to compare the performance of the SMF approach with the precomputed SCB approach. Since the network topology has to be precomputed for every second of the scenario for the precomputed SCB approach, it was decided to do the comparison in only the 24-node scenario. The third experiment evaluated the performance of four different information dissemination middleware systems and a baseline UDP multicast using the precomputed SCB approach in a 24-node scenario.

The 24-node network consists of Company 1 and the first 20 min of the Anglova scenario. Another time segment, from 5500 to 6501 s into the Anglova scenario, is used for the larger networks. The 48-node network consists of Companies 1 and 3; the 72-node network consists of Companies 1, 3, and 4; and the 96-node network consists of all the companies.

Node traffic focused on BFD disseminations, where the BFT messages are sent to all nodes in the network. The messages are different sizes and are sent with different retransmission intervals to vary the traffic load (Figures 42 and 43).

SMF - 24 Nodes	Size	Frequency
Low	128 Bytes	2 seconds
Medium	1024 Bytes	2 seconds
High	1024 Bytes	1 second
SMF - 48 Nodes		
Low	128 Bytes	2 seconds
Medium	1024 Bytes	4 seconds
High	1024 Bytes	2 seconds
SMF - 72 Nodes		
Low	128 Bytes	2 seconds
Medium	1024 Bytes	6 seconds
High	922 Bytes	3 seconds
SMF - 96 Nodes		
Low	128 Bytes	2 seconds
Medium	1024 Bytes	8 seconds
High	1024 Bytes	5 seconds

Figure 42. BFD traffic loads (message sizes and frequency) used for the SMF–SCB approach.

Precomputed SCB - 24 Nodes	Size	Frequency
Low	128 Bytes	5 and 10 seconds
Medium	512 Bytes	5 and 10 seconds
High	1024 Bytes	5 and 10 seconds

Figure 43. BFD traffic loads (message sizes and frequency) used for the precomputed SCB approach.

We used the same wideband system configuration as in Suri et al. (2018) with $L_{b,max}$ set to 139 dB, a 1.25-MHz bandwidth, and an 875-Kbit/s link data rate.

We calculate the theoretical connectivity of the networks from the precomputed SCB topologies in Figure 44. The theoretical connectivity is defined as the fraction of node pairs that are connected via one or more hops. It should be close to the message DR in the emulations. The averages are 0.97 for the 24-node network and 1 for the larger network sizes.

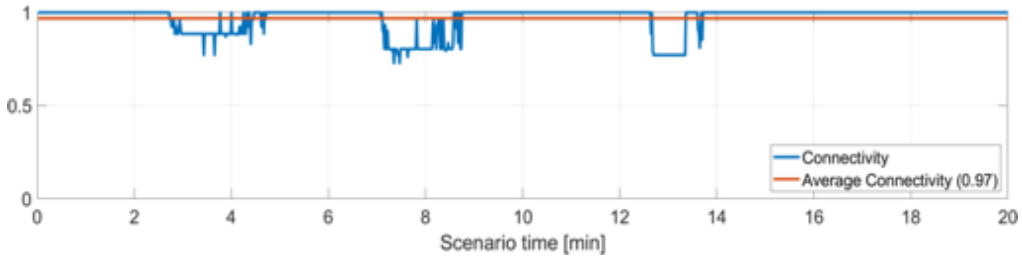


Figure 44. Theoretical connectivity for the 24-node Company 1 network.

As discussed in Section 7.2, three metrics were collected to evaluate the performance of each SCB method: DR, latency, and bandwidth usage. The DR reports on the ability of the method to successfully deliver the BFT messages to each recipient. A DR of 100 implies that every message was received by all recipients. The latency measures the delay (in milliseconds) that occurred for all the messages that were delivered. Finally, the bandwidth measurement determines how much network bandwidth was utilized by the system during message delivery.

7.3.1 Message DR

The following graphs show the average message DR throughout the scenario for the purpose of evaluating the two SCB models. The DR for the 24-node network should be roughly what is predicted in Figure 44 for both the SMF approach and the precomputed SCB approach. However, there may be some minor differences in the DR for the two approaches.

The are several drops in DR performance for the SMF-approach at minutes 4, 8, and 14 in the 24-node network for all traffic loads (Figure 45). These drops correspond to known sections of the Anglova scenario where node fragmentation

occurs, as indicated in Figure 44. These results also align with the DR when using the Precomputed SCB approach (Figure 46).

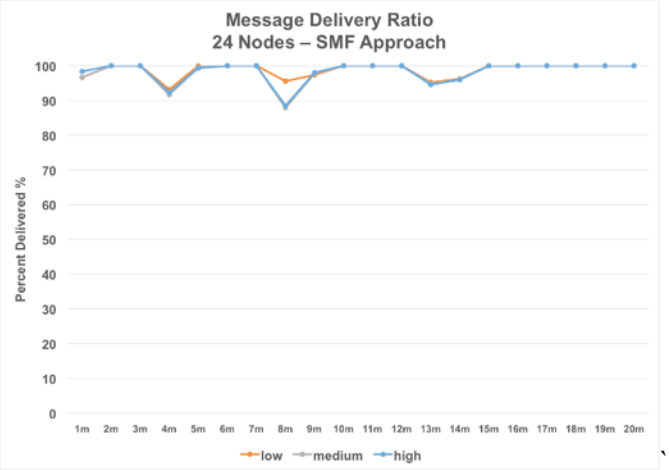


Figure 45. SMF approach message DR, 24 nodes.

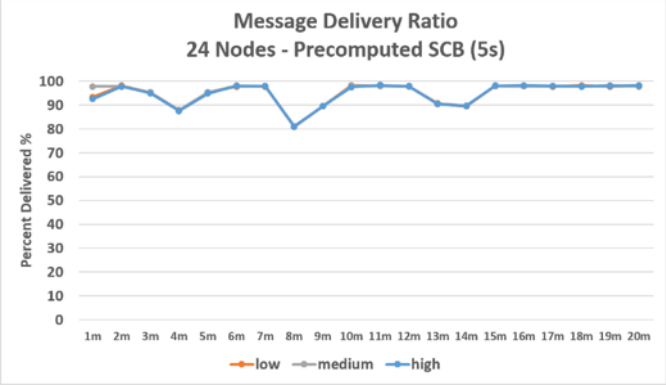


Figure 46. Precomputed SCB message DR (24 nodes, 5 s interval).

The networks are connected over the whole run for the larger networks, thanks to the rebroadcasting done by the additional nodes. The DR is the percentage of packets delivered and should be 100% (or close to 100%), as shown in Figure 47. The slightly lower than 100% DR for the first 2 min still needs to be investigated.

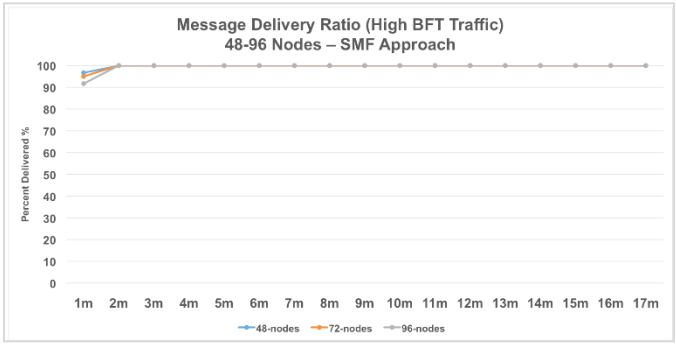
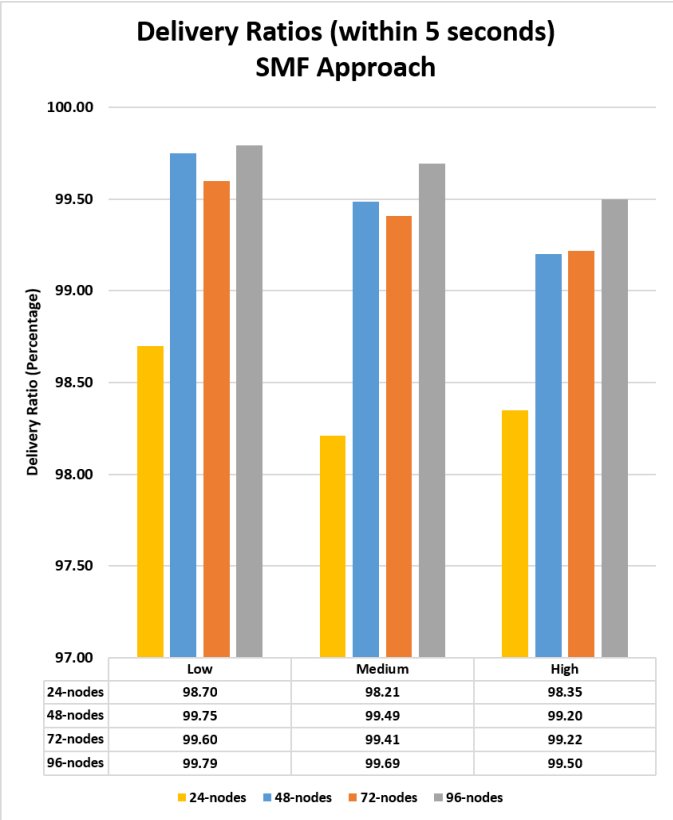


Figure 47. SMF approach message DR (48 to 96 nodes).

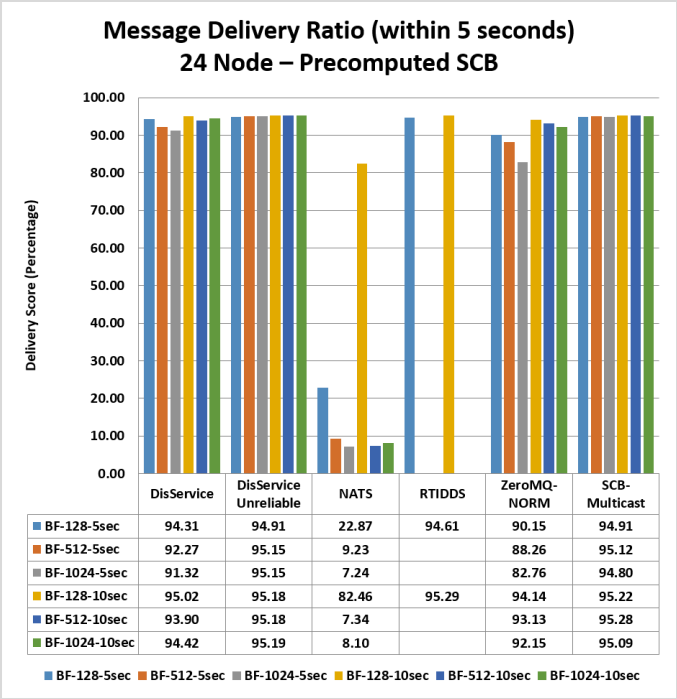
The message DRs for the SMF approach and different networks and traffic loads are shown in Figure 48. DR is slightly higher for the 24-node network than the theoretical bound for SCB based on connectivity (0.97). The reason may be an effect of SMF flooding, where the nodes resend the packet during a time interval when the network can overcome temporary connectivity breaks. The SMF approach uses a larger time diversity than an actual SCB implementation.



	Low	Medium	High
24-nodes	98.70	98.21	98.35
48-nodes	99.75	99.49	99.20
72-nodes	99.60	99.41	99.22
96-nodes	99.79	99.69	99.50

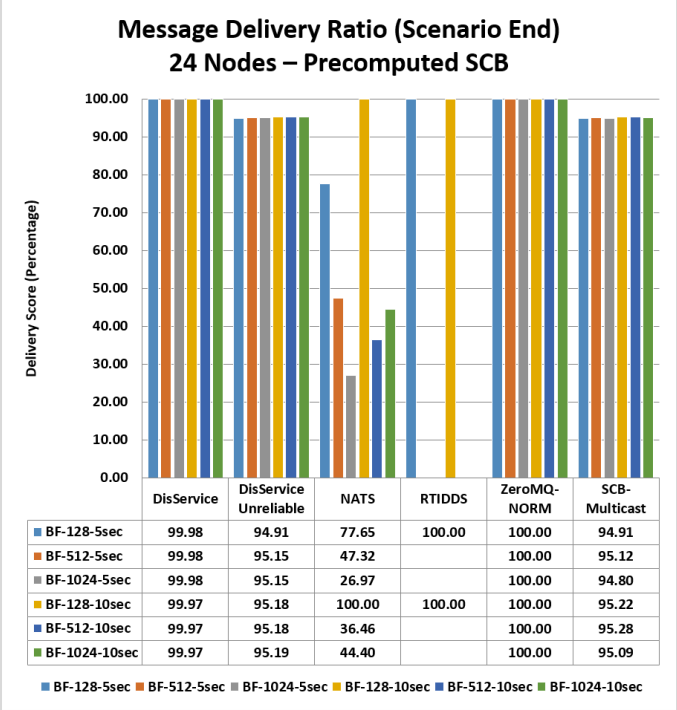
Figure 48. SMF approach DR for messages received within 5 s.

The message DRs for the precomputed SCB approach (labelled as SCB-Multicast) and different data dissemination protocols are shown in Figures 49 and 50. The dissemination protocols are run on top of precomputed SCB. The data dissemination protocols that were selected include DisService, NATS, a commercial implementation of DDS from Real Time Innovations (RTI), and ZeroMQ-NORM.



	DisService	DisService Unreliable	NATS	RTIDDS	ZeroMQ-NORM	SCB-Multicast
BF-128-5sec	94.31	94.91	22.87	94.61	90.15	94.91
BF-512-5sec	92.27	95.15	9.23		88.26	95.12
BF-1024-5sec	91.32	95.15	7.24		82.76	94.80
BF-128-10sec	95.02	95.18	82.46	95.29	94.14	95.22
BF-512-10sec	93.90	95.18	7.34		93.13	95.28
BF-1024-10sec	94.42	95.19	8.10		92.15	95.09

Figure 49. Message DR, 24 nodes, data dissemination protocols running on precomputed SCB.



	DisService	DisService Unreliable	NATS	RTIDDS	ZeroMQ-NORM	SCB-Multicast
BF-128-5sec	99.98	94.91	77.65	100.00	100.00	94.91
BF-512-5sec	99.98	95.15	47.32		100.00	95.12
BF-1024-5sec	99.98	95.15	26.97		100.00	94.80
BF-128-10sec	99.97	95.18	100.00	100.00	100.00	95.22
BF-512-10sec	99.97	95.18	36.46		100.00	95.28
BF-1024-10sec	99.97	95.19	44.40		100.00	95.09

Figure 50. Message DR, 24 nodes, data dissemination protocols running on precomputed SCB.

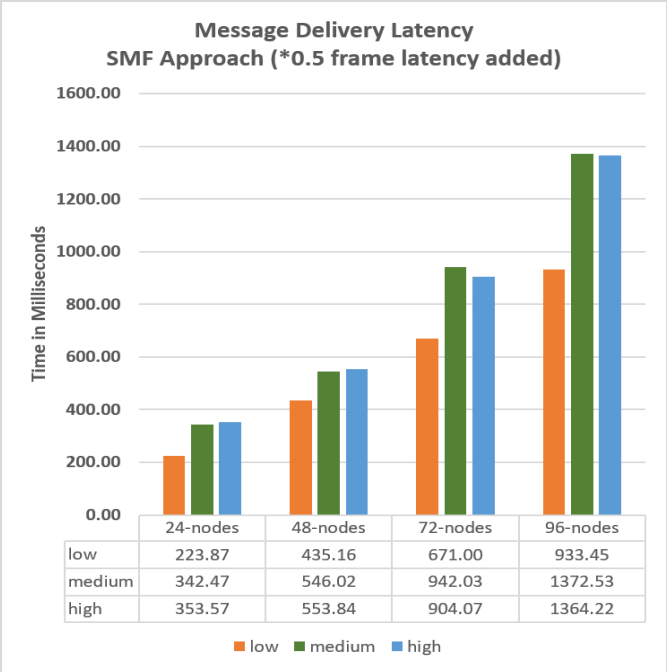
The results show that DisService Unreliable and SCB Multicast (which is basically a simple UDP multicast) perform approximately the same and are close to the theoretically computed DR of 97%. When reliability is turned on with DisService, the DR is close to 100% at the end of the scenario. This is also the case with ZeroMQ-NORM, which is another protocol providing a reliable multicast capability.

The performance of NATS differs considerably from the other protocols, as it is the only middleware that uses TCP. We see that the only configuration where NATS provides a reasonable DR (82%) is with the smallest size for the BFT message (128 bytes) and at the lowest update rate (once every 10 s).

The implementation of DDS from RTI provides results similar to the UDP multicast, but it was limited to only sending 128-byte messages. This is a limitation we hope to overcome in future work.

7.3.2 Message Delivery Latency

The results of the next measure for evaluating the two SCB modeling approaches is the delivery latency, as shown in Figures 51–57. An important note about the latency measurement is that it only records latencies for delivered messages (i.e., a protocol may have a very low DR, but if the latency is low for the few messages that were delivered, the overall reported latency is also low). All latencies were measured in milliseconds. Moreover, a node has to wait half a frame length on average in the SMF approach to obtain its SCB slot. Therefore, this latency of waiting half a frame on average is added. For example, this latency is 192 ms for the 24-node network. As can be seen in Figures 51 and 52, the latency is not so much higher than this waiting time for the low traffic load. The latency becomes a bit higher for the higher traffic loads due to some temporary queues in the nodes. The latency for the larger networks and different traffic loads is shown in Figures 53–55. The latency starts to increase at the end of the scenario for medium and high traffic loads in the larger networks. This is an indication that the queues in the nodes increase over time. This does not happen for the low traffic loads.



	24-nodes	48-nodes	72-nodes	96-nodes
low	223.87	435.16	671.00	933.45
medium	342.47	546.02	942.03	1372.53
high	353.57	553.84	904.07	1364.22

Figure 51. SMF approach message latency with 0.5-frame latency added.

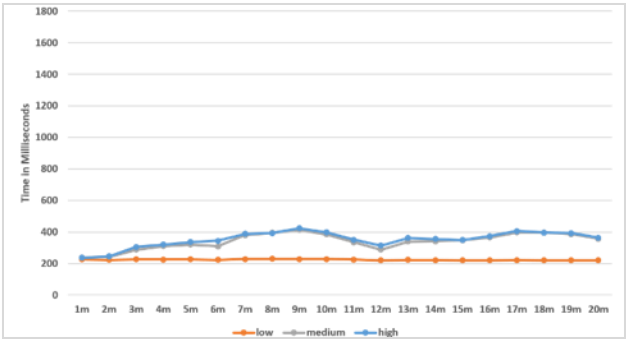


Figure 52. SMF approach message delivery latency for 24 nodes with 0.5-frame latency added.

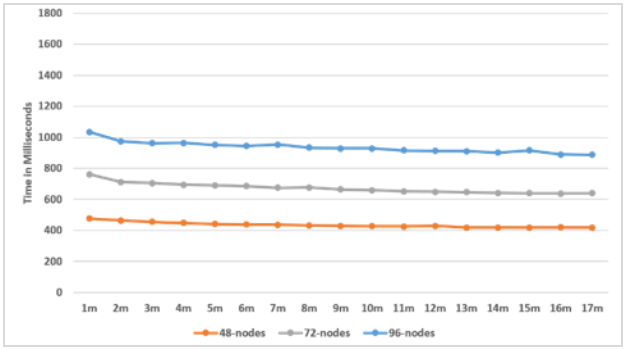


Figure 53. SMF approach message delivery latency for low traffic load with 0.5-frame latency added.

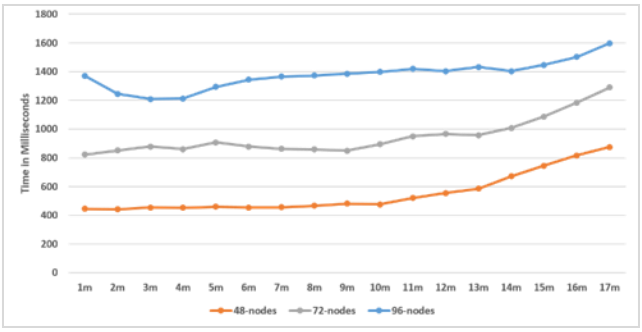


Figure 54. SMF approach message latency with 0.5-frame latency added.

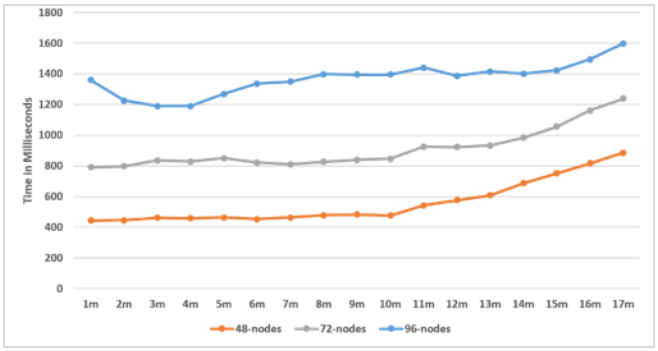
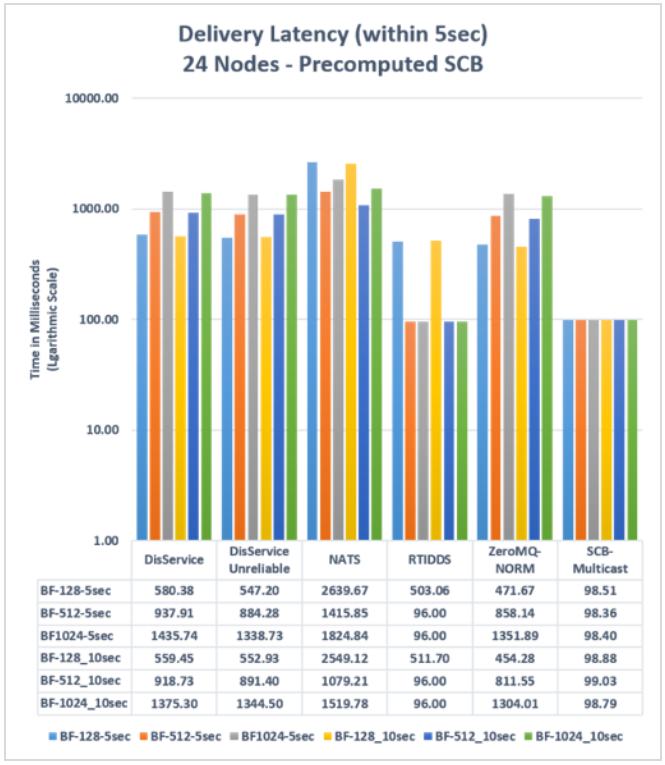


Figure 55. SMF approach message delivery latency for high traffic load with 0.5-frame latency added.



	DisService	DisService Unreliable	NATS	RTIDDS	ZeroMQ-NORM	SCB-Multicast
BF-128-5sec	580.38	547.20	2639.67	503.06	471.67	98.51
BF-512-5sec	937.91	884.28	1415.85	96.00	858.14	98.36
BF1024-5sec	1435.74	1338.73	1824.84	96.00	1351.89	98.40
BF-128_10sec	559.45	552.93	2549.12	511.70	454.28	98.88
BF-512_10sec	918.73	891.40	1079.21	96.00	811.55	99.03
BF-1024_10sec	1375.30	1344.50	1519.78	96.00	1304.01	98.79

Figure 56. Delivery latency for messages received within 5 s. Data dissemination protocols for 24 nodes running on precomputed SCB.

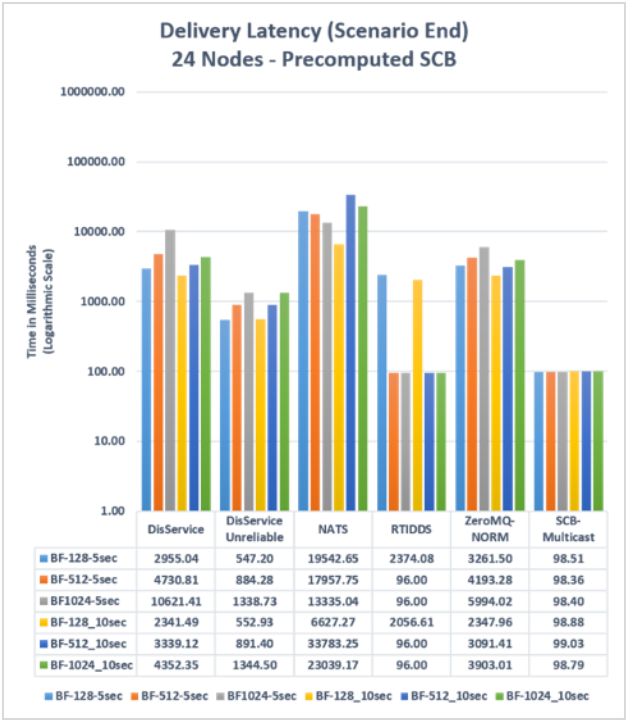


Figure 57. Delivery latency for messages received by the end of the scenario. Data dissemination protocols for 24 nodes running on precomputed SCB.

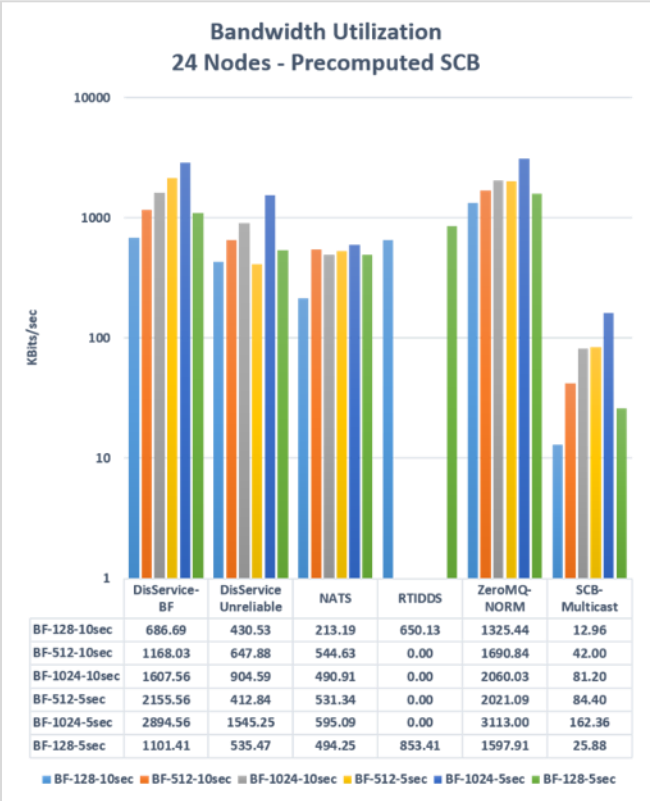
In the precomputed SCB approach, the latency of waiting half a frame on average is added in case of up to four hops; for 5 to 8 hops, 1.5-frame latency is added. The latency for the precomputed SCB approach (SCB multicast) is shown in Figures 56 and 57; also shown is the measured latency for the different data dissemination protocols that have been run in the 24-node scenario using the precomputed SCB model.

The latency of precomputed SCB does not depend on the traffic load as long as no queues build up in the ingress nodes. It is a fixed schedule where each node has to wait on its SCB slots. The latency is therefore the same for different traffic loads as long the network does not get overloaded, and every node must wait the same amount of time on average. In this case, the latency of waiting half a frame is 96 ms. As shown in Figure 56, the latency for precomputed SCB is only slightly higher than 0.96 ms. There are a few cases when more than four hops are needed for the 24-node network.

7.3.3 Bandwidth Usage

Bandwidth usage is the last measure that was collected for evaluating the two SCB approaches. The results are graphed for precomputed SCB in Figure 58 in terms of kilobits per second (kbps) and separated by BFT message pattern (size and frequency). The bandwidth usage is fixed in some sense for precomputed SCB.

Each node has one CB slot (four time slots, and the amount that each node can send is limited by the RFPipe data rate. Just one transmission is measured in the emulation (i.e., the one from the source node); the relay transmissions are not measured. Therefore, the actual bandwidth usage is four times higher than measured as four slots are allocated. In practice, some of these time slots may not be used at all if it is a single-hop or two-hop network, but they are allocated and no other node can use them, so a CB slot can be considered utilized all the time. All the dissemination protocols use substantially more bandwidth than a simple UDP multicast.



	DisService-BF	DisService Unreliable	NATS	RTIDDS	ZeroMQ-NORM	SCB-Multicast
BF-128-10sec	686.69	430.53	213.19	650.13	1325.44	12.96
BF-512-10sec	1168.03	647.88	544.63	0.00	1690.84	42.00
BF-1024-10sec	1607.56	904.59	490.91	0.00	2060.03	81.20
BF-512-5sec	2155.56	412.84	531.34	0.00	2021.09	84.40
BF-1024-5sec	2894.56	1545.25	595.09	0.00	3113.00	162.36
BF-128-5sec	1101.41	535.47	494.25	853.41	1597.91	25.88

Figure 58. Bandwidth usage. Data dissemination protocols for 24 nodes running on precomputed SCB.

Bandwidth usage increases with network size with the SMF approach, but also with traffic load, as can be expected (Figure 59). Of course, the traffic load is substantially higher than for the precomputed SCB model because SMF executes full flooding and every transmission is counted given that it occurs at the UDP level, whereas with the precomputed SCB, these retransmissions are factored out by virtue of computing all of the nodes that would have received the message ahead of time and simply enabling that to happen in one hop.

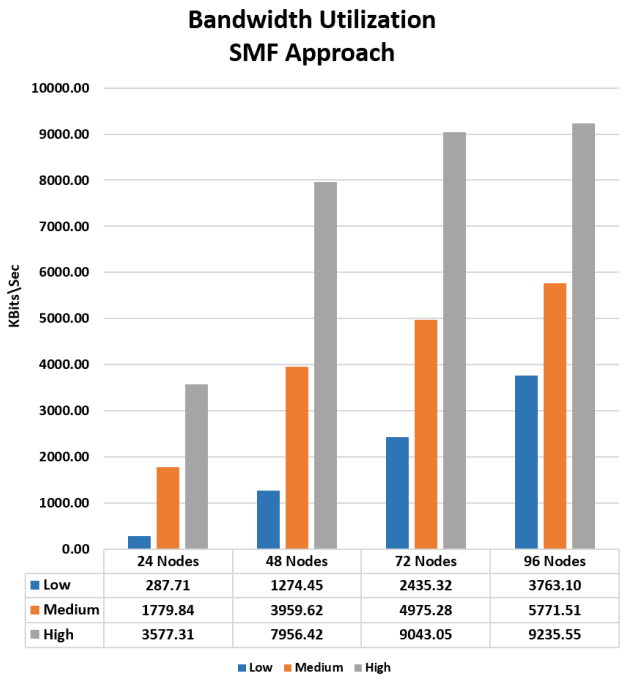


Figure 59. SMF approach bandwidth usage.

7.4 Scalability to Four Companies

The evaluation of the two SCB approaches in EMANE and an initial experiment with group communication protocols in a 24-node scenario with a precomputed SCB model were described in Section 7.3. After those experiments, the RTG executed a set of experiments to evaluate the performance of different group communication protocols using the precomputed SCB approach in a full scenario of all four companies. Additionally, GDEM was available for inclusion for the first time in the IST-161 experiments. As stated in Section 6.1, GDEM is based on the JDSS-IEM that is part of the FMN Spiral 4 specifications, and the results of these experiments have been published in Suri et al. (2021).

The experimental results show that centralized (or semi-centralized) broker-based approaches (in our case, represented by NATS) are not well-suited for group communications over wireless military networks where multiple nodes can simultaneously receive transmissions from a source. NATS clearly under-performed when compared with all other multicast or broadcast-based protocols in terms of having the worst DRs, the highest latency, and highest bandwidth usage. Amongst the other protocols, GDEM outperformed both DisService (except for a very small subset of metrics) and ZeroMQ-NORM. Furthermore, the scalability of these protocols appears to be linear with respect to the number of companies (and hence the number of nodes).

We used the troop deployment Vignette 2, where a mechanized battalion stages an attack against a hostile force, for these experiments. A subset of the vignette was used, consisting of the nodes forming two tank companies (Companies 1 and 2) and two mechanized infantry companies (Companies 3 and 4), with movements from scenario time 5000 s to 7000 s (see Figure 5 for more details on the movement patterns).

Each company had its own wideband network on a separate frequency, and a wideband overlay network was used to connect the company networks. Two nodes per company (respectively, nodes with ID 1, 2, 25, 26, 49, 50, 73, and 74) also participated in an overlay network, acting as gateways, to support inter-company communications. Each gateway essentially had two network interfaces—the first one to communicate with other nodes in the local company, and the second one to communicate with the other gateways that were also part of the overlay network. Each wideband network was configured with a total bandwidth of 875 kbits/s.

The selected group communication protocols for this experiment were NATS, DisService in unreliable (DisService-R) and reliable (DisService-U) modes, GDEM in unreliable (GDEM-U) and reliable (GDEM-R) modes, and ZeroMQ-NORM. For DisService and GDEM, the reliable mode indicates that the reliability measures of the respective protocol (such as retransmissions) are turned on.

Three measures were collected for comparison in these experiments: message DR, message latency, and protocol bandwidth usage.

7.4.1 Message DR

Figure 60 shows the results for the message DRs for the different protocols. The graphs show the results for different cut-off times, giving an impression of the delivery latency as well. While (as expected) NATS clearly underperforms since it is a broker-based solution, the DRs of GDEM and DisService are the highest for the one, two, and three company scenarios, with GDEM slightly outperforming DisService in the two and three company scenarios. ZeroMQ-NORM has a lower performance than GDEM and DisService. In the four company scenario, DisService-R performs significantly lower for the 5-s and 10-s cut-offs. When we measure all arriving messages without any time cut-off, all protocols except NATS perform well. Zero-MQ even hits the 100% mark for the one and two company scenarios.

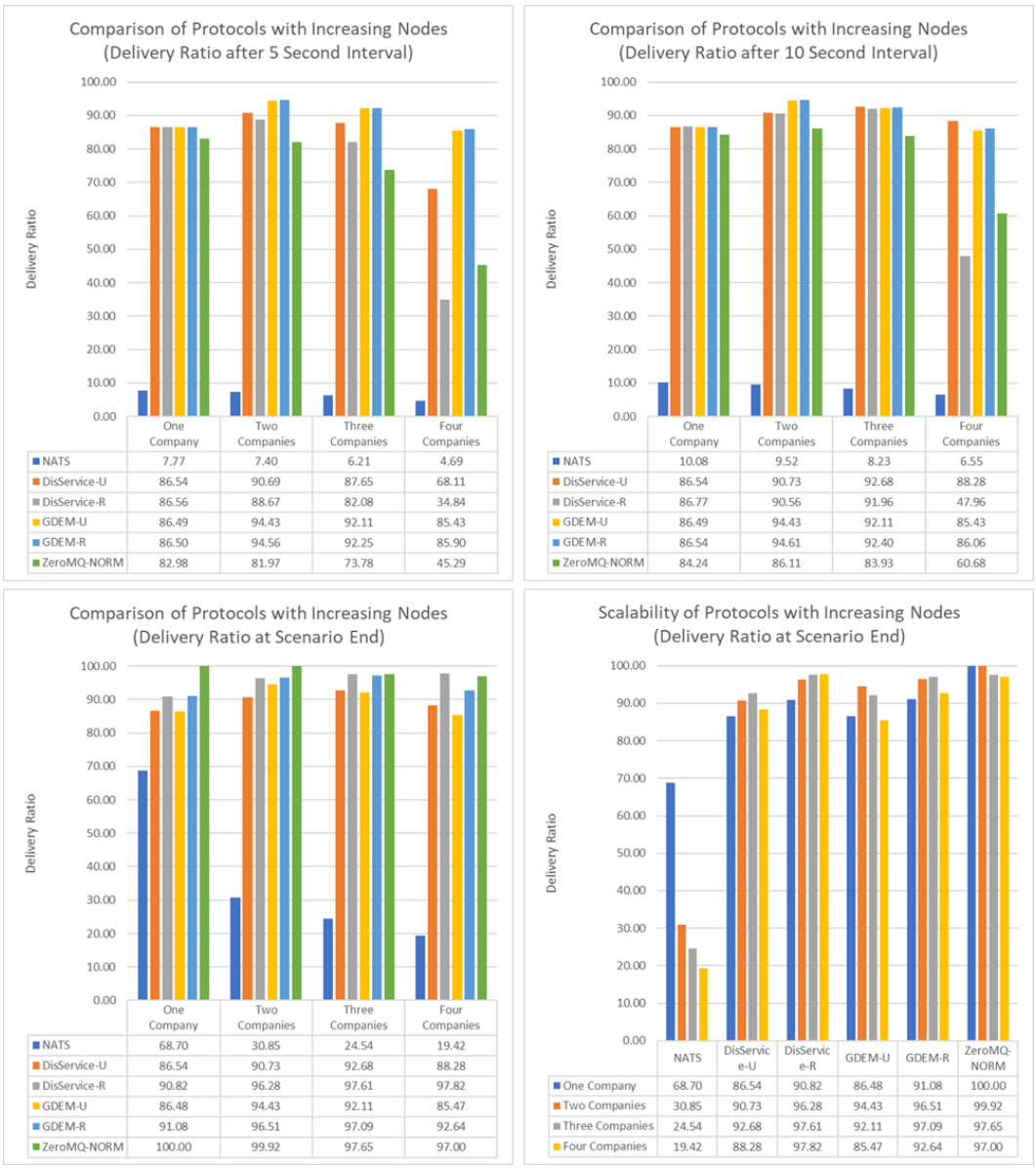


Figure 60. Comparison of the DR of different protocols with a different cut-off times and different company sizes.

7.4.2 Message Latency

Results concerning message latency for each protocol with 10-s cut-off are shown in Figure 61. GDEM clearly outperforms the other protocols when it comes to latency.

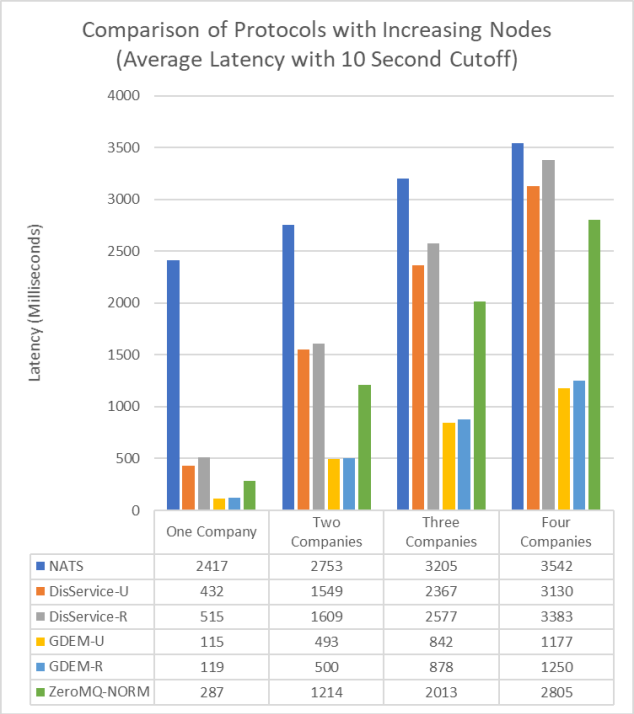
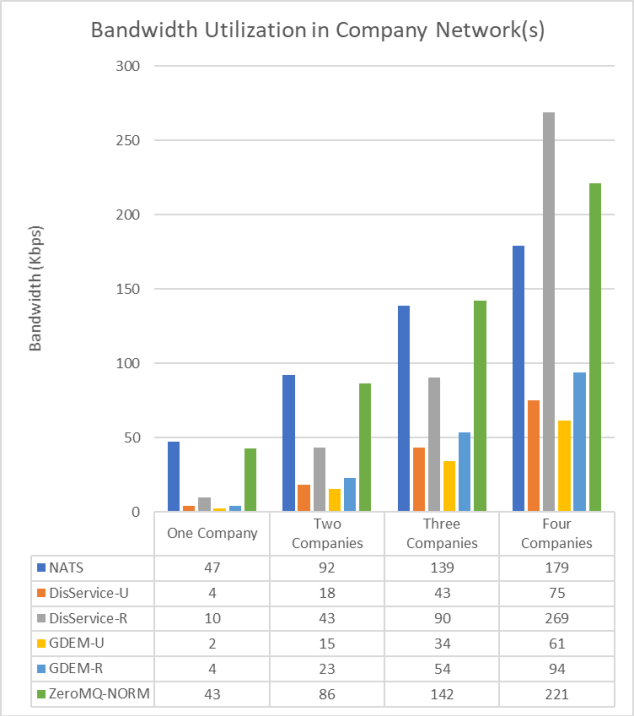


Figure 61. Comparison of the latency of different protocols with a 10-s cut-off and different company sizes.

7.4.3 Bandwidth Usage

Bandwidth usage of the group communication protocols in a military radio network is also very important. The results of its experiment are visualized in Figure 62. Again, GDEM clearly outperforms the other protocols. When looking at DisService and GDEM, GDEM-R outperforms DisService-R and GDEM-U outperforms DisService-U. For the four-company scenario, we see a large increase of bandwidth usage by DisService-R, going beyond even the bandwidth usage of ZeroMQ-NORM.



	One Company	Two Companies	Three Companies	Four Companies
NATS	47	92	139	179
DisService-U	4	18	43	75
DisService-R	10	43	90	269
GDEM-U	2	15	34	61
GDEM-R	4	23	54	94
ZeroMQ-NORM	43	86	142	221

Figure 62. Bandwidth usage in the company network with different company sizes.

7.4.4 Conclusions

GDEM has the best overall performance of the tested group communication protocols. GDEM especially outperforms the other protocols when it comes to the message latency and bandwidth usage. When it comes to message delivery, GDEM performs better in most cases, but there are cases where DisService and even Zero-MQ slightly outperform GDEM.

The difference between DisService-R and GDEM-R can essentially be explained by the applied reliability mechanisms of both protocols. GDEM-R has a configurable repair window size and therefore does not try to repair all messages, while DisService tries to retransmit every single lost message. DisService pays for this approach with a significant increase in latency and bandwidth usage while adding a relatively low increase of message DR at the end of the scenario when compared with GDEM-R. Therefore, the GDEM-R reliability mechanism is the better choice.

7.5 Jamming

A particularly compelling and sometimes overlooked criterion for the performance assessment of group communication solutions is resilience to jamming. In fact, it is not only conceivable but also probable that military forces in battlefield scenarios will have to deal with adversaries actively disrupting their communications.

Overall, we found that both DisService and GDEM are solid performers across the board, with the former exhibiting a relatively lower responsiveness (perhaps due to its more aggressive stance in trading off latency for a higher DR). ZeroMQ-NORM is a decent performer for BFT but is incapable of dealing with larger SD messages. NATS is basically out of its element in this scenario.

To evaluate the impact of jamming, we set up a dedicated emulation testbed based on a subset of Vignette 2 of the Anglova scenario. It reenacts the operations of a mechanized battalion staging an attack against a hostile force, involving 4 companies of 24 nodes each (96 nodes in total). We used this environment to evaluate the jamming resilience of six different protocols: GDEM (both in reliable and unreliable modes), DisService (both in reliable and unreliable modes), NATS, and ZeroMQ-NORM. The results of these experiments were published in Marcus et al. (2020).

For the experiments, a subset of the vignette was used, consisting of the nodes forming two tank companies (Companies 1 and 2) and two mechanized infantry companies (Companies 3 and 4), with movements from scenario time 5000 s to 7000 s. Each company had its own wideband network on a separate frequency and a wideband overlay network was used to connect the company networks. Two nodes per company (respectively, nodes with ID 1, 2, 25, 26, 49, 50, 73, and 74) also participated in an overlay network, acting as gateways to support inter-company communications. We used a precomputed SCB topology (as described in Section 5.1.5.3) that mimicked a static SCB network and allowed us to fully model SCB's cooperative effects.

We considered the company networks without jamming as a baseline. In that case, the company networks were connected and had a connectivity of one. We also considered special cases when the company networks were jammed during the whole scenario. As a rough model of a jammer quite far away with free line of sight to the nodes in the company networks, we reduced $L_{b;max}$ with values of 10, 20 and 30 dB. This resulted in more and more fragmented company networks. The connectivity of the scenario for different levels of jamming is shown in Figure 63.

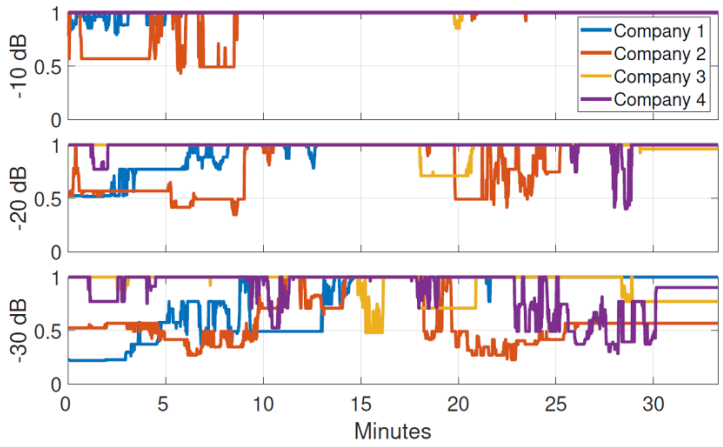


Figure 63. Connectivity of the company networks, with $L_{b;max}$ reduced by 10, 20 and 30 dB.

We first conducted an experiment to assess the capabilities of protocols to deliver messages in a denied environment, by using the DR as a reference metric (i.e., the percentage of destination nodes that received a message within the pool of intended recipients). Figures 64 and 65 show, respectively, the histograms of the DRs of BFT and SD messages for the six group communication solutions under four different conditions: no jamming (i.e., 0 dB), 10-, 20-, and 30-dB jamming. The x-axes of the figures indicate the DR of a message (which of course is limited between 0 and 1) and the y-axes indicate the number of messages that exhibited the corresponding DR. This plot format allows one to visually comprehend in a relatively straightforward fashion the performance of a protocol with respect to its capability to deliver messages. In fact, the ideal performance would be a single column at the right of the plot, corresponding to a 1.0-DR. The more a histogram differs from this ideal inverted-L shape, the worse its performance. Even from a quick glance at Figures 64 and 65, it is possible to identify the impact of jamming on the procotols' performance, with higher levels of jamming leading to histogram shapes that exhibit a higher density of low-DRs that look less and less like the ideal inverted-L.

A higher jamming value has an impact on the performance of all protocols, albeit to very different degrees. NATS exhibits poor performance across the board. Although it performs much better than NATS, ZeroMQ-NORM suffers considerably from jamming. Jamming impacts the performances of DisService and GDEM to a much lesser extent, with reliable versions of those protocols performing decently well even in cases of severe jamming.

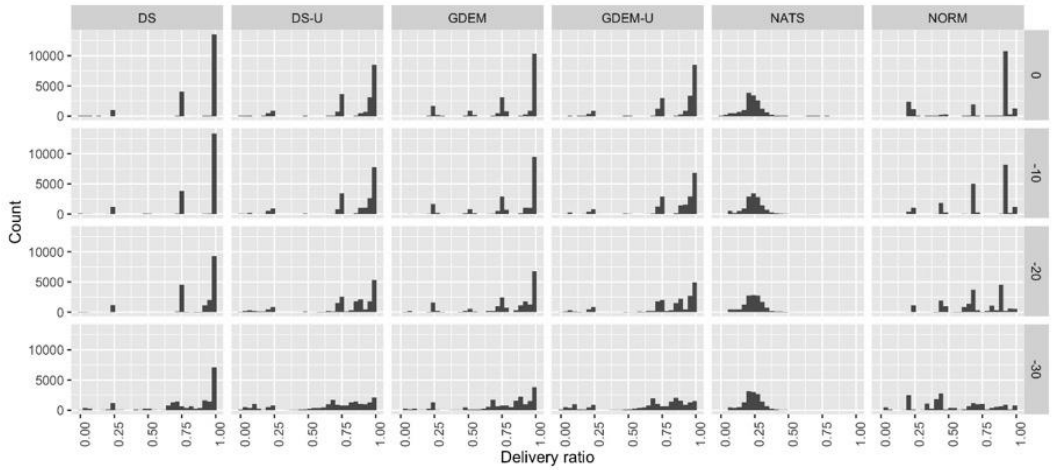


Figure 64. DR distribution for BFT messages with 0-, 10-, 20-, and 30-dB jamming levels.

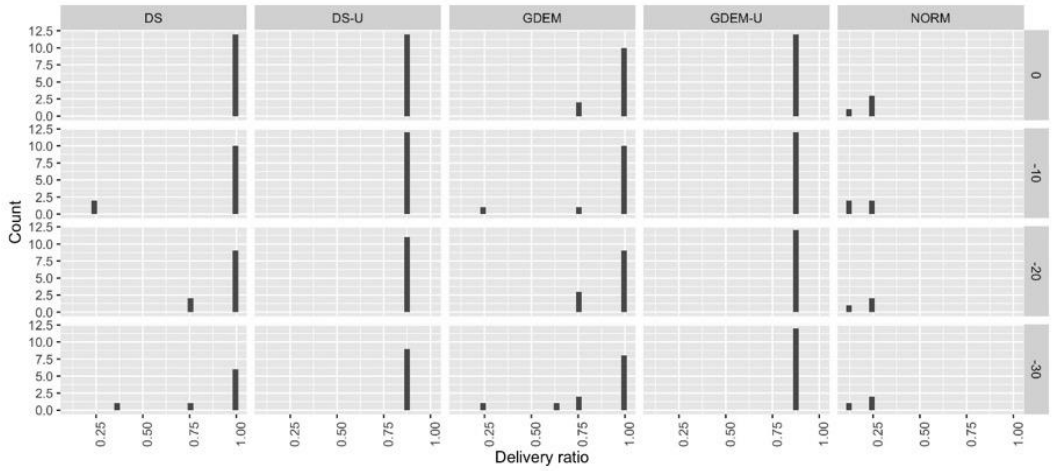


Figure 65. DR distribution for SD with 0-, 10-, 20-, and 30-dB jamming levels.

Table 7 enables a less immediate but more in-depth analysis of the protocols' performance. Table 8 displays the DR recorded for BFT messages after 5 s from their initial transmission, after 10 s, and overall with 0-, 10-, 20-, and 30-dB jamming levels. The results indicate that DisService outperforms its competitors either in reliable or unreliable modes, with GDEM being a close second overall and in short-term DR. However, this discrepancy might be caused by GDEM's new message relay implementation that is not yet working optimally.

Table 7 shows the DR and throughput (in bits per second [bps]) used at the company-level network (BCN) and at the overlay network (BON) recorded for SD messages with 0-, 10-, 20-, and 30-dB jamming levels. The data indicate that GDEM has the lowest footprint from the bandwidth consumption perspective. DisService comes in second place, exhibiting a relatively aggressive tendency in

trading off bandwidth consumption (for retransmissions) to achieve the highest possible message DR. Both NATS and ZeroMQ-NORM show very poor performance, likely due to their poor handling of large messages such as those used to carry SD.

Table 7. DR and bandwidth (in bps) recorded for SD messages.

Protocol	Jamming level											
	0 dB			10 dB			20 dB			30 dB		
	DR	BCN	BON	DR	BCN	BON	DR	BCN	BON	DR	BCN	BON
DS	100	201294	47029	93,75	233501	50868	78,12	271490	53365	59,38	296559	47976
DS-U	87,5	169594	47290	87,5	162507	44412	79,7	165612	44585	65,62	141728	40120
GDEM	94,2	104516	31291	89,58	104185	29908	87,5	116593	31454	87,5	136230	31500
GDEM-U	87,5	120397	26598	87,5	115891	25722	87,5	126825	27968	87,5	128875	27168
NATS	0	289497	85296	0	285352	82784	0	279064	83614	0	278285	81880
NORM	7,29	464799	95343	6,25	463313	92233	5,2	444529	86560	5,21	409464	75103

Table 8. DR for BFT messages.

Protocol	Jamming level											
	0 dB			10 dB			20 dB			30 dB		
	At 5 s	At 10 s	Overall	At 5 s	At 10 s	Overall	At 5 s	At 10 s	Overall	At 5 s	At 10 s	Overall
DS	36,15	40,31	88,31	25,9	27,77	87,85	20,41	22,04	84,14	22,59	24,86	77,47
DS-U	43,92	46,96	84,43	33,86	36,49	83,49	30,22	32,13	78,7	36,91	39	66,2
GDEM	34,5	34,95	81,65	34,77	35,31	81,68	33,18	33,78	78,63	32,11	33,4	74,68
GDEM-U	65,52	66,56	85,84	66,01	67,11	84,35	64,08	66,4	79,98	55,48	57,32	67,24
NATS	4,1	5,82	17,89	4,2	5,83	17,94	4,2	5,81	17,78	4,01	5,69	17,49
NORM	25,62	30,2	75,66	23,76	27,07	75,23	20,02	22,2	64,6	16,35	17,71	49,9

Finally, we analyzed the delivery latency (i.e., the time elapsed between a packet* transmission and its receival). To investigate the dynamic behavior of protocols and see how they reacted to changes in the emulation environment, we sliced the dataset in different time windows that each correspond to 5 min, and then separately analyzed the protocol performance in those windows.

Using box and whiskers plots, Figures 66 and 67 show the distribution of delivery latencies obtained for BFTs and SD for the six group communication solutions under four different conditions: no jamming (i.e., 0 dB), 10-, 20-, and 30-dB jamming. Of course, lower values are better as they correspond to quicker delivery times. The better performance that protocols seem to demonstrate as the jamming level intensifies needs to be offset by the fact that the protocols are indeed delivering a significantly lower number of messages to their destination. Protocols basically exhibit bipolar behavior: they either deliver messages quickly, or don't deliver them at all. As anticipated by the data in Table 8, GDEM has very good

* Because a message has many recipients, each message will generate many packets.

performance. DisService, its closest competitor, is also quite good but exhibits higher latencies (especially in reliable mode), perhaps because it aggressively leverages store-and-forward communication semantics to deliver messages.

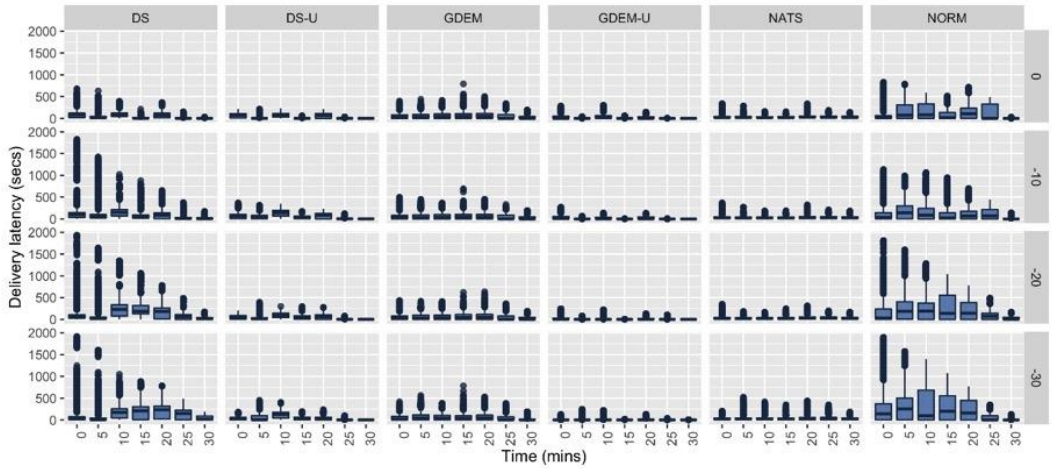


Figure 66. Delivery latency distribution of BFT messages aggregated within 5-min intervals, respectively reduced by 0, 10, 20, and 30 dB.

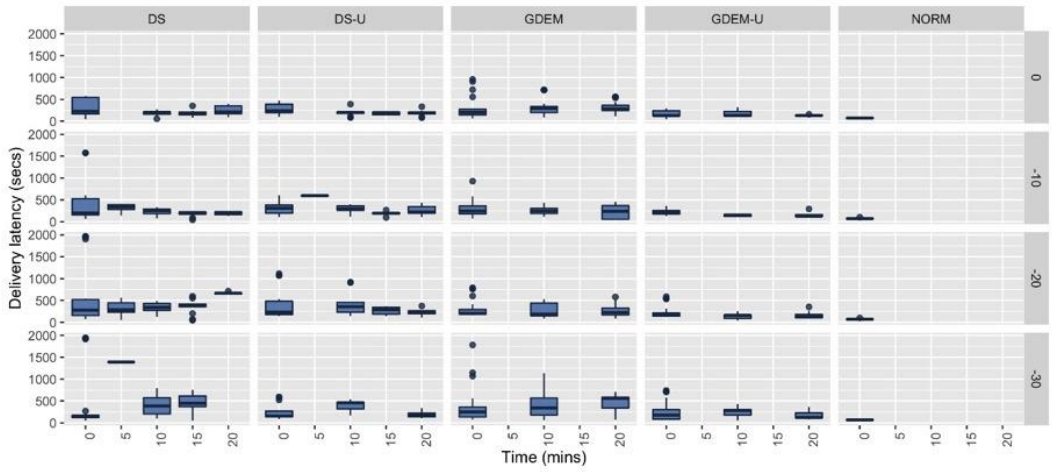


Figure 67. Delivery latency distribution of SD aggregated within 5-min intervals, respectively reduced by 0, 10, 20, and 30 dB.

7.6 NDN

Following an extensive evaluation of broker-based, reliable, and distributed protocols, we proceeded to start evaluating NDN. It pursues a clean-slate approach that could bring potential to military applications based on our design decisions and adjustments (described in Section 6).

NDN leverages UDP multicast for message transmission for both our experimental evaluations (i.e., single-company and multi-company environments). With this method, applications will broadcast Interests to all reachable nodes in the network that, if they have cached the requested content, will broadcast the requested Data message back.

We focused on the investigation of three performance indicators in the NDN experiments: DR (cumulative number of Data packets received is included in the single-company evaluation), delivery latency, and bandwidth usage. These three performance metrics also have been evaluated for other group communication protocols to enable easy comparison of NDN against other relevant baseline references. NDN has been compared with DisService and NATS for our single-company experiments, while we also introduced GDEM results in the multicompany experiments.

The results of the RTG's NDN experiments have been published at MILCOM 2019 (Campioni et al. 2019b) and MILCOM 2022 (Campioni et al. 2022).

7.6.1 Single-Company Results

In a single-company environment, our NDN configuration showed a DR similar to DisService configured in best-effort QoS mode (Figure 68), highlighting that NDN can be used as a reliable communication protocol for medium-size networks.

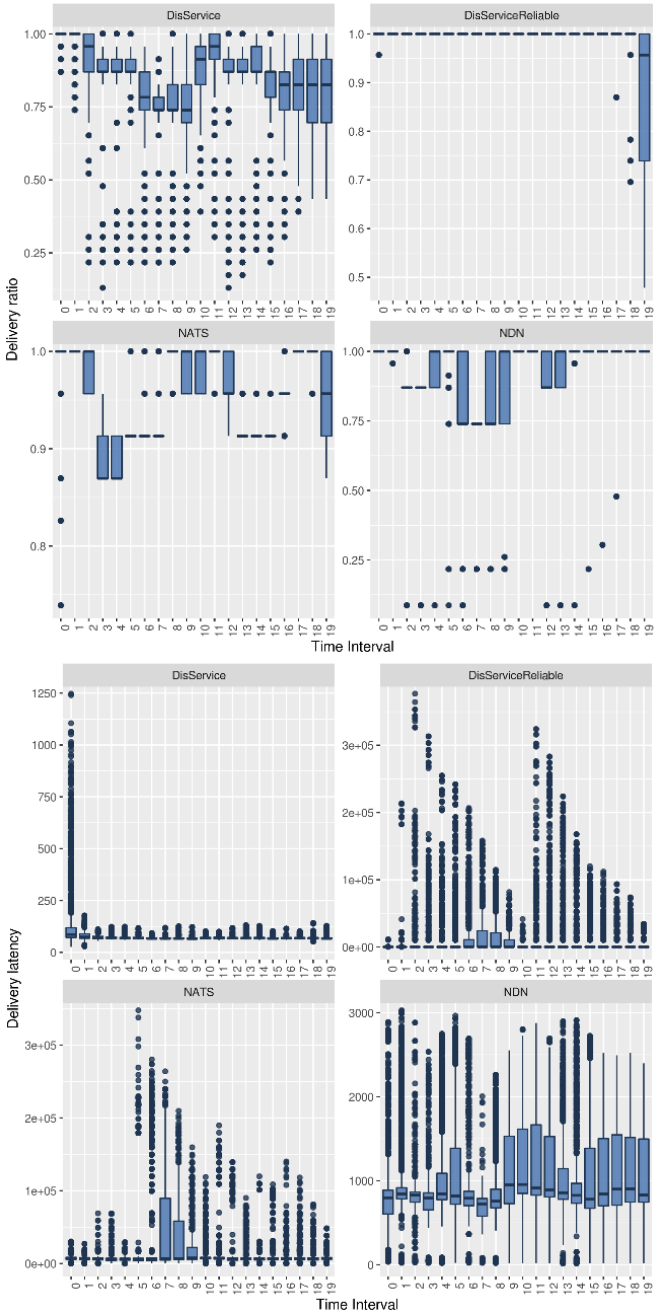


Figure 68. DR per minute, delivery latency per minute, and cumulative number of data received per node.

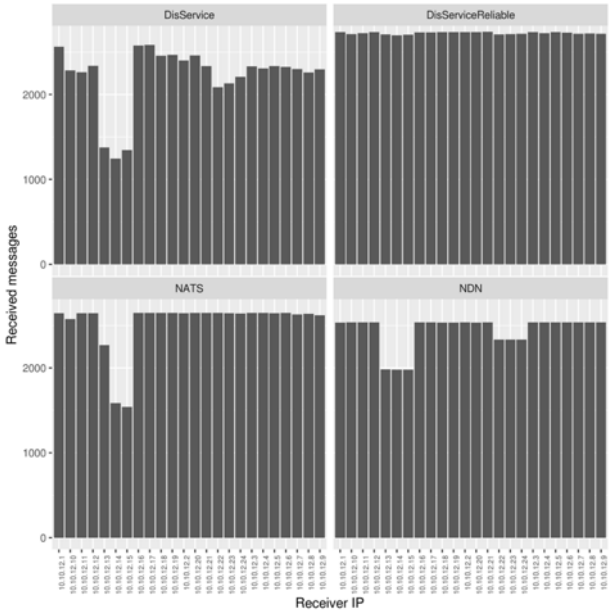


Figure 68. DR per minute, delivery latency per minute, and cumulative number of data received per node (continued).

However, the average latency of our NDN configuration highlights the challenges of a pull-based communication model offered by the protocol for this push-based scenario of BFT data. Since we configured NDN nodes to issue Interests for these experiments only after the referred BFT content was generated, the link latency includes the consumer waiting during both the transmission of the Interest and the retrieving of the Data.

Another negative aspect we registered with our NDN configuration is the inefficient use of network resources. As highlighted in the cumulative bandwidth consumption in Figure 69, our NDN configuration uses 10× more bandwidth compared with DisService though achieving a similar DR to DisService. This is because most of the consumers in our configuration request the same BFT data at roughly the same time, minimizing the use of content caching within the nodes in the network. After all, most of the consumers already sent out their Interest (for the data of the same producer) before that data has been sent by a producer. Even though we already optimized this a bit (as described in Section 6), more effort is needed to optimize this further, possibly by implementing a proper synchronization model.

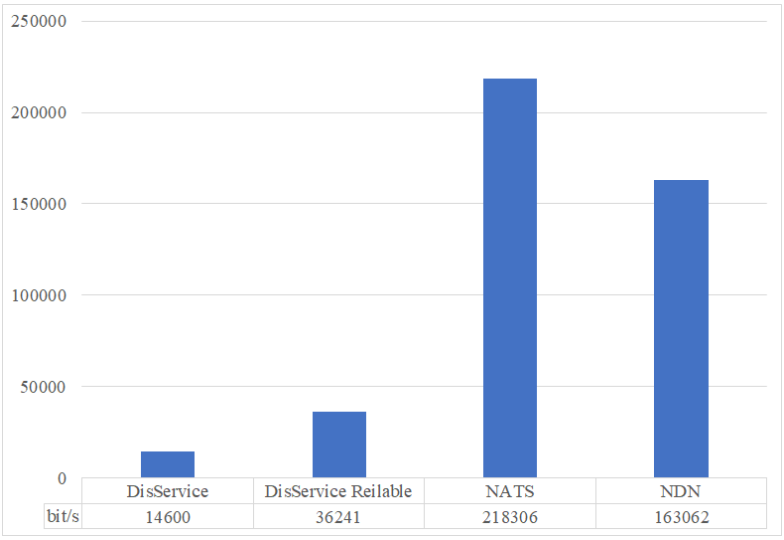


Figure 69. Cumulative bandwidth consumption per protocol (bits/s).

7.6.2 Multicompany Results

The NDN configuration that we used for the single-company setup is less effective within a multicompany setup (Figures 70 and 71). With this configuration, NDN barely delivers 40% of the generated BFTs successfully to the consumers within the different companies which is insufficient for its consideration in military environments, even if content is transmitted with a best-effort policy.

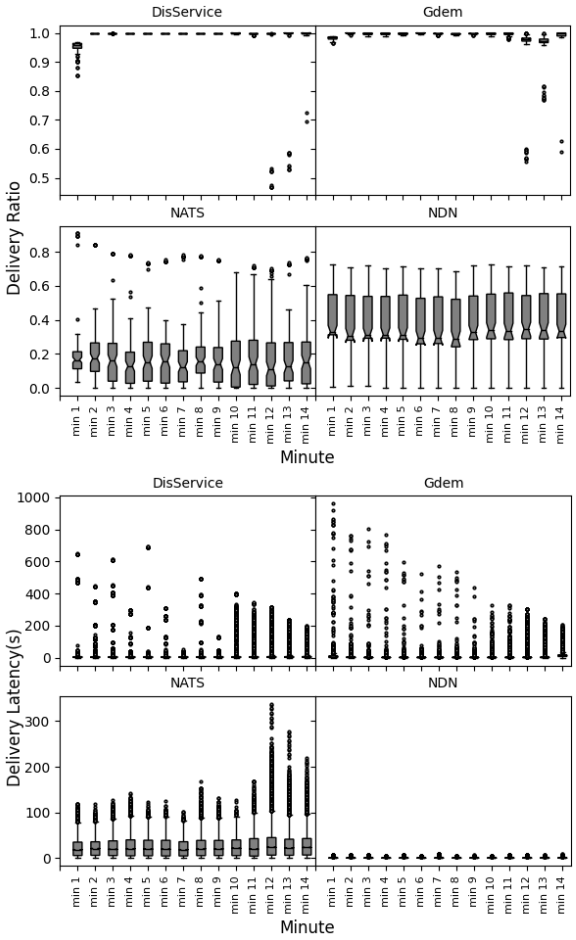


Figure 70. DR per minute, delivery latency per minute.

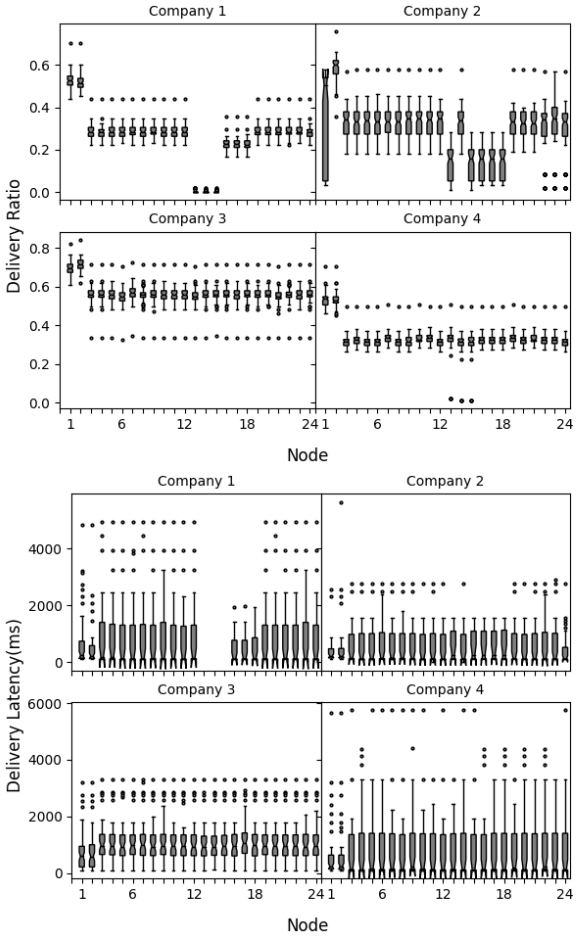


Figure 71. NDN DR per node (left) and NDN delivery latency per node (right).

To mitigate that, we adopted a more aggressive publish–subscribe-like strategy wherein each node issued the next Interest immediately after receiving the data packet of the previously requested content. This mechanism was purely timer-based with the single-company setup. This adjusted strategy shows an improvement in terms of data delivery latency, however there was not much improvement in terms of data DR. It shows that this configuration still cannot cope very well with nodes disconnecting from the group for some time, after which they need to catch up on the missed BFT data. These results imply that a better synchronization mechanism needs to be implemented to improve the data DR.

Moreover, while we optimized the data request pattern of the different nodes by introducing randomization in the Interest requests to gain more benefits from in-network caching, our NDN configuration showed a higher bandwidth consumption compared with the other protocols (Figure 72). One of the causes of this higher bandwidth consumption is that each node still sends our Interest messages for all other nodes within the network, while some of these nodes are unavailable at that specific time (as the nodes do not yet have a mechanism to synchronize on node

availability). We speculate that we can optimize bandwidth consumption by implementing such a synchronization mechanism.

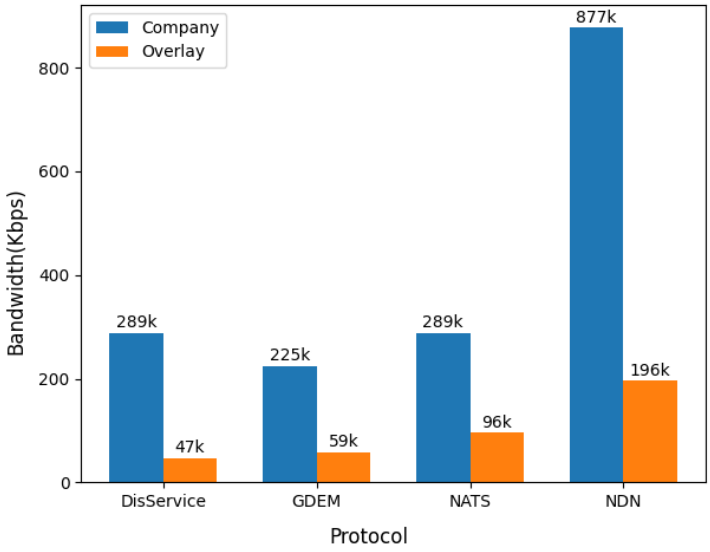


Figure 72. Cumulative bandwidth consumption.

7.6.3 NDN Conclusions

Our NDN configuration has shown promising performance results in terms of DR and delivery latency for the single-company scenario. This is in contrast with the multicompany scenario in which our NDN configuration was seemingly unable to deliver sufficient BFT data to the consumers within the different companies, despite the Interest dissemination upgrades we implemented.

Our NDN configuration also presented shortcomings compared with some of the other evaluated protocols; these included having higher bandwidth consumption for delivering the same BFT data and the need for implementing additional adjustments to facilitate a publish–subscribe-like mechanism on top of the Interest/Data request model. This assumed isochronous (and known) generation intervals of BFT data that could prove impractical for some applications and were poorly scalable with our multicompany scenario. Thus, migrating a naturally push-based BFT application towards a pull-based communication model proves to be challenging.

This shows it is important to design a proper content synchronization protocol on top of NDN that can be used by applications to share and synchronize their data, instead of trying to perform this kind of synchronization from within the BFT applications themselves, as our current NDN-based experiment configuration seems unfit for multicompany (e.g., larger size) networks. However, it is too early to state if the NDN concept itself is a viable candidate for military networks; a proper content synchronization mechanism needs to be designed for this to be

evaluated. As of this writing, first initiatives for such concepts are showing up within the ICN community (e.g., the NDN State-Vector-Sync concept).

8. Conclusions and Recommendations

The main contributions of the IST-161 RTG are in the domain of group communication protocols for mobile military networks.

8.1 Anglova and Emulation Environment

The IST-161 RTG enriched the Anglova scenario and experimentation environment by several contributions, including making available a detailed description of the Anglova battalion of Vignette 2, developing a test harness for traffic generation and data analysis, modelling SCB waveforms, and testing with a jamming model.

The Anglova scenario can still be extended in the future (i.e., by adding additional traffic scenarios and waveform models). Alternative mobility scenarios should be considered, for instance, in an additional vignette. When it comes to the emulation testbed, the primary way to share testbed configurations is sharing VMs (e.g., via anglova.net). Multiple partners are setting up experimentation environments and a VPN setup can be used in the future for multi-premises testing by different partners (as used on CWIX). Future groups should still strive to run experiments jointly on the same testbed, not just connecting individual setups. This allows for a much closer level of cooperation.

8.2 Architecture

Group communication solutions cover a wide area of solution space, especially when solutions for static networks are included. Usually, these solutions are suites of protocols and have their own design. The IST-161 RTG developed architectural approaches that enabled fair comparison between solutions for group communications. The first approach involves using the concept of protocol stack comparison rather than just protocol comparison. This allows one to think more precisely about what functionality we are comparing. The second approach is the group communication taxonomy that describes all functionalities that may be part of a group communication solution without linking it specifically to a technical layer where it must be implemented.

When it comes to future research, a protocol stack comparison using the group communication protocol taxonomy allows for efficient (cross-layer) design of group communication solutions (e.g., by preventing redundant functionality that is

implemented in multiple layers). Such a terminology framework can reduce the amount of misunderstanding when it comes to group communications research.

8.3 Security

The security section highlighted the challenges faced in tactical radio networks, defined typical assets and security objectives, and outlined available standards and how they can be applied to achieve different security configurations. It provided insight into the challenges within the field of security in tactical radio networks and group communications. We recommended future IST activities that focus on mobile military networks to include these security insights in their research.

8.4 Group Communications Protocols

Regarding group communication middleware, our results show that broker-based solutions fail in a mobile, bandwidth-constraint, intermittent, network. This makes sense, since they have centralized architecture. The RTG found that distributed protocols for group communications can scale up to 96 nodes in the Anglova scenario.

GDEM and DisService clearly outperform the other group communication solutions, with GDEM providing the best results. GDEM is an extension of the NATO standard JDSS (STANAG 4677; NATO 2017a). It is therefore advised to pursue standardization of key elements of GDEM in a new version of STANAG 4677 so the improvements of GDEM can also find their way into FMN.

While GDEM is the best performing solution in the RTG experiments, there could still be other use cases and traffic scenarios for which DisService may be a good alternative to GDEM. It is valuable to pursue this for other network and traffic scenarios. Although DisService is open source, the protocol itself has not been standardized.

SCB has been evaluated as a good approach on the transmission layer for multicast/broadcast data dissemination due to its scalability and robustness. It is recommended to investigate how SCB performs when unicast traffic also needs to be transported. Since SCB is optimized for broadcast traffic, SCB may be less efficient for unicast.

The RTG found that NDN in its current form is not well-suited for distributing BFD in the Anglova scenario Vignette 2. The group recommends that NDN in its current state should not be pursued for NATO standardization. Many developments in NDN would be needed to make it useful for mobile tactical networks. In the future it could be explored for other types of user traffic (e.g., not-regular multicast of

large files). One could argue if this is worth the effort, as the results of such development could potentially result in a close copy to GDEM or DisService that already exist.

8.5 Routing and Group Communications

This study showed that functionality required for group communications is implemented differently in different solutions. When viewing mobile military networks as mobile ad hoc IP networks, a prominent concern regards how network routing should be implemented. Since group communications protocols are focusing on multicast traffic, and traditional multicast routing is typically too cumbersome for bandwidth-constraint mobile environments, the RTG investigated IP multicast forwarding approaches. However, the resulting report was published after the finalization of IST-161 and therefore was not included in this report. The results of that report show that SMF in combination with the OLSRv2 MPR mechanism to enable forwarding has been evaluated to be efficient in many (but not all) cases. Forwarding with SMF is desirable in larger, more complex networks.

Another approach that the RTG discussed is implementing forwarding between networks inside the group communication middleware. In this case, radio technology (such as SCB) takes care of multi-hop forwarding within the radio network. At the edges of the radio network, the group communication middleware would make decisions on forwarding the data onto other radio networks. Such an approach may give more control over data forwarding; however, it requires a middleware implementation on intermediate nodes that is not needed when using an end-to-end IP-multicast forwarding model. Considerations such as these need to be investigated more thoroughly and be supported with experimentation. The IST-161 RTG recommends including IP-multicast–related topics in a future IST activity.

9. References

Adamson B, Bormann C, Handley M, Macker J. RFC 5740: NACK-oriented reliable multicast (NORM) transport protocol. Network Working Group; 2009 Nov. Retrieved April 17, 2025, from https://tools.ietf.org/html/rfc5740.

Antipolis S. ETSI releases cryptographic standards for secure access control. European Telecommunication Standards Institute; 2018 Aug [accessed 2020 May]. Retrieved April 17, 2025, from https://www.etsi.org/newsroom/news/1328-2018-08-press-etsi-releases-cryptographic-standards-for-secure-access-control

Apache Software Foundation. Apache Kafka. Apache Software Foundation; c2024. Retrieved April 17, 2025, from https://kafka.apache.org/

Babiarz J, Chan K, Baker F. RFC 4594: Configuration guidelines for DiffServ service classes. The Internet Society; 2006 Aug. Retrieved April 17, 2025, from https://www.rfc-editor.org/info/rfc4594

Baker F, Polk J. RFC 4542: Implementing an emergency telecommunications service (ETS) for real-time services in the internet protocol suite. The Internet Society; 2006 May. Retrieved April 17, 2025, from https://www.rfc-editor.org/info/rfc4542

Barz C, Fuchs C, Kirchhoff J, Krzyzek M, Brück N. An approach to measurement-based tactical radio modeling and real time emulation. Proceedings of the 2018 International Conference on Military Communications and Information Systems (ICMCIS); 2018 May 22–23; Warsaw, Poland. p. 1–5.

Barz C, Kirchhoff J, Niewiejska J, Prahl-Kamps M, Stavrou I, Werberich E. Distributed network experimentation in the NATO CWIX 2021 communications focus area. Proceedings of MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM); 2021 Nov 9–Dec 2; San Diego, CA. p. 360–365. https://doi.org/10.1109/MILCOM52596.2021.9653134

Blair A, Brown T, Chugg KM, Halford TR, Johnson M. Barrage relay networks for cooperative transport in tactical MANETs. Proceedings of the MILCON 2008 – 2008 IEEE Military Communications Conference; 2008 Nov 16–19; San Diego, CA. p. 1–7. https://doi.org/10.1109/MILCOM.2008.4753655

Boonstra D, Hartog T, Schotanus H, Verkoelen C. A secure NEC-enabling architecture disentangling infrastructure, information and security. Proceedings of NATO STO-MP-IST-111; 2012 Sep 13.

Burke J, Afanasyev A, Refaei T, Zhang L. NDN impact on tactical application development. Proceedings of MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM); 2018 Oct 29–31; Los Angeles, CA. p. 640–646. https://doi.org/10.1109/MILCOM.2018.8599725

Campioni L, Hauge M, Landmark L, Suri N, Tortonesi M. Considerations on the adoption of named data networking (NDN) in tactical environments. Proceedings of 2019 International Conference on Military Communications and Information Systems (ICMCIS); 2019a May 14–15; Budva, Montenegro. p. 1–8. https://doi.org/10.1109/ICMCIS.2019.8842778

Campioni L, Tortonesi M, Wissingh B, Suri N, Hauge M, Landmark L. Experimental evaluation of named data networking (NDN) in tactical environments. Proceedings of MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM); 2019b Nov 12–14; Norfolk, VA. p. 43–48. https://doi.org/10.1109/MILCOM47813.2019.9020843

Campioni L, Wissingh B, Tortonesi M, Suri N, Stefanelli C. NDN experimental evaluation multi-domain tactical environments. Proceedings of MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM); 2022 Nov 28–Dec 2; Rockville, MD. p. 1–6. https://doi.org/10.1109/MILCOM55135.2022.10017463

Capoano F, Kaplan LA. NetJSON: data interchange format for networks. NetJSON; 2015 July Retrieved June 11, 2024, from https://netjson.org/rfc.html

Capoano F. What is NetJSON? NetJSON; c2021. Retrieved April 17, 2025, from https://netjson.org/docs/what.html

Chan K, Babiarz J, Baker F. RFC 5127: Aggregation of DiffServ service classes. The Internet Society; 2008 Feb. Retrieved April 17, 2025, from https://www.rfc-editor.org/info/rfc5127

Clausen T, Dearlove C, Jacquet P, Herberg U. RFC 7181: the optimized link state routing protocol version 2. Internet Engineering Task Force; 2014 Apr. Retrieved April 17, 2025, from https://datatracker.ietf.org/doc/html/rfc7181

Farsund B, Hegland AM, Lillevold F. LTE for military communication – business models and vulnerabilities. Proceedings of International Conference on Advanced Communications Technology (ICACT) 2017; 2017 Feb 19–22; Pyeongchang, South Korea.

Fronteddu R. Computer networks in tactical and disaster recovery environments [dissertation]. University of Ferrara; 2022.

GitHub. EMANE wiki. GitHub; c2024. Retrieved June 11, 2024, from https://github.com/adjacentlink/emane/wiki

Gong J, Wee H. Adaptively secure ABE for DFA from k-Lin and more. In: Canteaut A, Ishai Y, editors. Advances in cryptology – EUROCRYPT 2020. Lecture notes in computer science; vol. 12107. 2020. p. 278–308.

Grönkvist J, Komulainen A, Sterner U, Uppman U. Dynamic scheduling for cooperative broadcasting in tactical ad hoc networks. Proceedings of MILCOM 2016 – 2016 IEEE Military Communications Conference (MILCOM); 2016 Nov 1–3; Baltimore, MD. p. 1034–1040. https://doi.org/10.1109/MILCOM.2016.7795466

Komulainen A, Grönkvist J, Nilsson J. Comparing the capacity of cooperative broadcast to spatial reuse TDMA in ad hoc networks. Proceedings of the 2016 International Conference on Military Communications and Information Systems (ICMCIS); 2016 May 23–24; Brussels, Belgium. p. 1–7. https://doi.org/10.1109/ICMCIS.2016.7496581

Lin H, Luo J. Compact adaptively secure ABE from k-Lin: beyond NC1 and Towards NL. In: Canteaut A, Ishai Y, editors. Advances in cryptology – EUROCRYPT 2020. Lecture notes in computer science; vol. 12107. 2020. p. 247–277.Macker J, editor. RFC 6621: simplified multicast forwarding. Internet Engineering Task Force; 2012 May. Retrieved April 17, 2025, from https://datatracker.ietf.org/doc/html/rfc6621

Marcus KM, Nilsson J, Suri N, Hansson A, Breedy M, Fronteddu R, Cramer E. Emulating synchronized cooperative broadcast in the Anglova scenario and EMANE. Proceedings of the 2020 Military Communications and Information Systems Conference (MilCIS); 2020 Nov 10–12; Canberra, Australia. p. 1–9. https://doi.org/10.1109/MilCIS49828.2020.9282377.

Named data networking. Wikipedia; c2024. Retrieved June 11, 2024, from https://en.wikipedia.org/wiki/Named_data_networking

NATO Allied Command Transformation. Federated interoperability - CWIX. NATO; 2021. Retrieved April 17, 2025, from https://www.act.nato.int/activities/federated-interoperability

NATO Science and Technology Organization. Heterogeneous tactical networks – improving connectivity and network efficiency. NATO; 2019 Sep. Report Nos.: STO-TR-IST-124-Part-I, STO-TR-IST-124-Part-II.

NATO Security Committee. AC/35-D/2004-REV3: primary directive on CIS security. NATO; 2013 Nov 15.

NATO. IMSM-0149-2019: data centric security vision and strategy proposal for the alliance federation, including the NATO enterprise. NATO; 2019a May 17.

NATO. MC 0640: minimum level of communication and information systems capabilities at land tactical level. NATO; 2019b Jul 24.

NATO. SCIP extension for interoperability with TSVCIS release 1.1. NATO; 2020 Sep.

NATO. STANAG 4677 AEP-76 Vol I Ed A Ver 3: specifications defining the Joint Dismounted Soldier System Interoperability Network (JDSSIN) - security. NATO; 2017a Dec 15.

NATO. STANAG 4677 AEP-76 Vol IV Ed A Ver 3: specifications defining the Joint Dismounted Soldier System Interoperability Network (JDSSIN) - information exchange mechanism. NATO; 2017b Dec 15.

NATO. STANAG 4677: dismounted soldier systems standards and protocols for command, control, communications and computers (C4) interoperability (DSS C4 interoperability). NATO; 2014 Oct 3.

NATO. STANAG 4711: Interoperability point quality of service (IP QoS). NATO; 2018 Jan 25a.

NATO. STANAG 4787 AComP-4787 Edition A: Networking and information infrastructure (NII) internet protocol (IP) network encryptor – interoperability specification (NINE ISPEC). NATO; 2018 Jan 25b.

NATO. STANAG 5527 Ed 1: friendly force tracking systems (FFTS) interoperability. NATO; 2017c Mar 20.

Navy Research Laboratory. Extendable mobile ad-hoc network emulator (EMANE). Navy Research Laboratory (US); c2024. Retrieved June 11, 2024, from https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/EMANE/

NS-3 Consortium. NS-3 network simulator. NS-3 Consortium; c2024. Retrieved June 11, 2024, from https://www.nsnam.org/

Refaei T, Afanasyev A. Enabling a data-centric battlefield through information access gateways. Proceedings of MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM); 2018 Oct 29–31; Los Angeles, CA. p. 634–639. https://doi.org/10.1109/MILCOM.2018.8599860

Robert NL, Macker J, Millar D, William CR, Taylor I. XO: XMPP overlay service for distributed chat. Proceedings of MILCOM 2010 Military Communications

Conference; 2010 Oct 31–Nov 3; San Jose, CA. p. 1116–1121. https://doi.org/10.1109/MILCOM.2010.5680094

Sahai A, Waters B. Fuzzy identity-based encryption. In: Cramer R, editor. Advances in cryptology – EUROCRYPT 2005. Lecture notes in computer science; vol. 3494. 2005. p. 457–473.

Suri N, Benincasa G, Tortonesi M, Stefanelli C, Kovach J, Winkler R, Kohler R, Hanna J, Pochet L, Watson S. Peer-to-peer communications for tactical environments: observations, requirements, and experiences. IEEE Communications Magazine. 2010 Oct;48(10);60–69. https://doi.org/10.1109/MCOM.2010.5594678

Suri N, Breedy MR, Fronteddu R, Morelli A, Cramer E, Nilsson J, Hansson AA, Marcus KM, Martens A. Evaluating the scalability of group communication protocols over synchronized cooperative broadcast. Proceedings of the 2021 International Conference on Military Communication and Information Systems (ICMCIS); 2021 May 4–5; The Hague, Netherlands. p. 1–9. https://doi.org/10.1109/ICMCIS52405.2021.9486407

Suri N, Breedy MR, Marcus KM, Fronteddu R, Cramer E, Morelli A, Campioni L, Provosty M, Enders C, Tortonesi M, et al. Experimental evaluation of group communications protocols for data dissemination at the tactical edge. Proceedings of 2019 International Conference on Military Communications and Information Systems (ICMCIS); 2019b May 14–15; Budva, Montenegro. p. 1–8. https://doi.org/10.1109/ICMCIS.2019.8842801

Suri N, Hauge M, Tortonesi M. Technology Watch Card – value-of-information-based prioritization and filtering. IST-TW-15, version 2019a Mar.

Suri N, in 't Velt R, Breedy M, Cramer E, Nilsson J, Marcus KM, Morelli A, Fronteddu R. To forward or not to forward: considerations on and experiments with network layer forwarding in combination with middleware services for group communications. Procedia Computer Science. 2022;205:78–87.

Suri N, Nilsson J, Hansson A, Sterner U, Marcus K, Misirlioğlu L, Hauge M, Peuhkuri M, Buchin B, in ‘t Velt R, Breedy M. The Angloval tactical military scenario and experimentation environment. 2018 International Conference on Military Communications and Information Systems (ICMCIS); 2018 May 22–23; Warsaw, Poland. p. 1–8. https://doi.org/10.1109/ICMCIS.2018.8398729

Swanson M, Hash J, Bowen P. NIST SP 800-18 rev.1: guide for developing security plans for federal information systems. National Institute of Standards and Technology; 2006 Feb.

Tomida J, Kawahara Y, Nishimaki R. Fast, compact, and expressive attribute-based encryption. Des Codes Cryptogr. 2021;89:2577–2626.

Tortonesi M, Breedy M, Campioni L, Cramer E, Hansson A, Morelli A; Nilsson J. Peuhkuri M. Suri N. Evaluating the impact of jamming on group communication protocols in tactical environments. Proceedings of MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM); 2021 Nov 9–Dec 2; San Diego, CA. p. 677–683. https://doi.org/10.1109/MILCOM52596.2021.9653039

Tsabary R. Fully secure attribute-based encryption for t-CNF from LWE. In: Advances in cryptology–CRYPTO 2019: 39th annual international cryptology conference proceedings, part I. 2019 Aug 18–22; Santa Barbara, CA. p 62–85.

World Wide Web Consortium. Efficient extensible interchange working group. W3C; c2016. Retrieved June 11, 2024, from https://www.w3.org/XML/EXI/

Zickau S, Thatmann D, Butyrtschik A, Denisow I, Küpper A. Applied attribute-based encryption schemes. Proceedings of the 19th International Innovations in Clouds, Internet and Networks (ICIN) Conference; 2016 Mar 1–3; Paris, France.

Appendix A. Anglova Vignette 2 Network Plans

This appendix provides the two example network plans for the Anglova scenario (Figures A-1 and A-2), as provided on www.anglova.net.

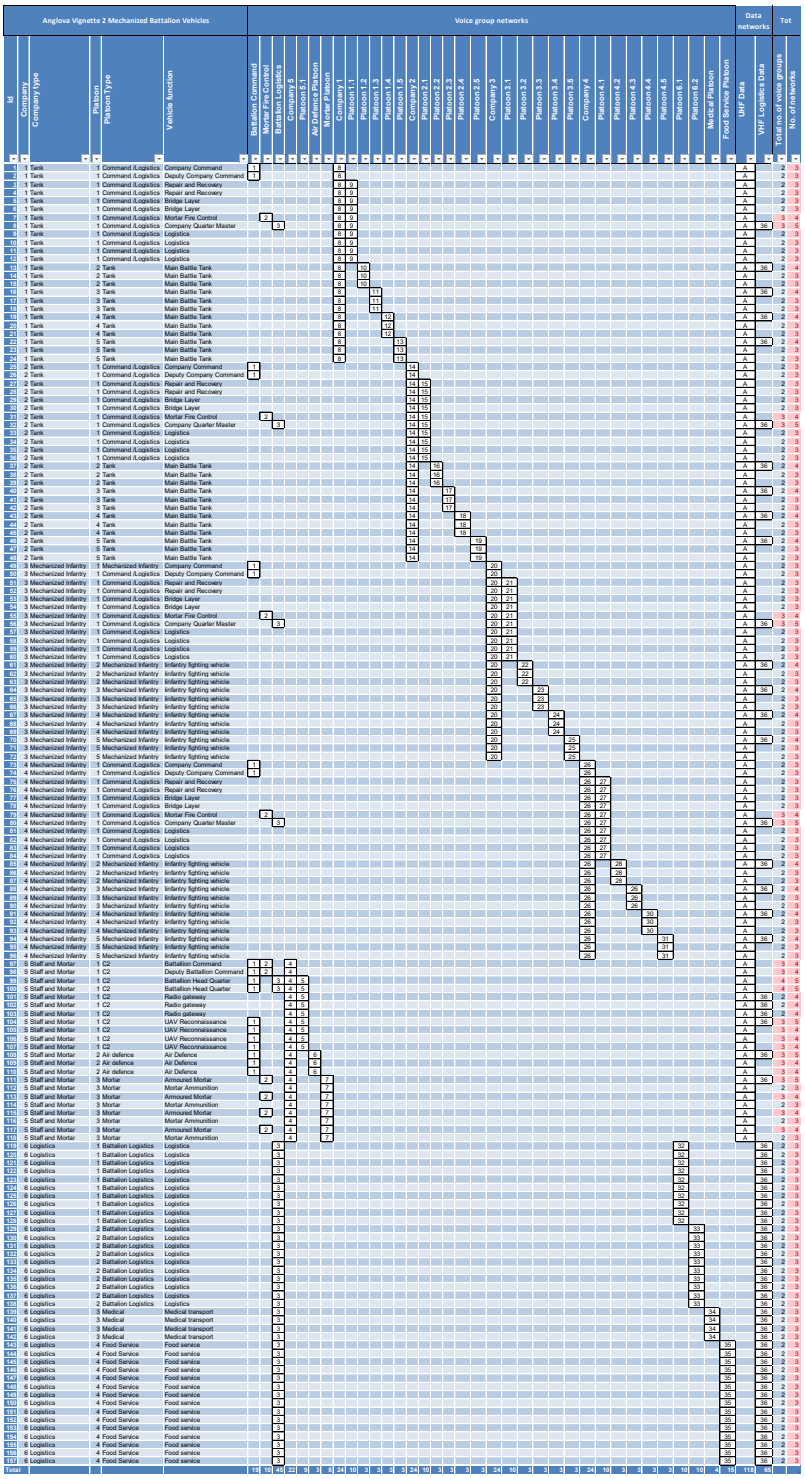


Figure A-1. Network plan 1 for Anglova Vignette 2.

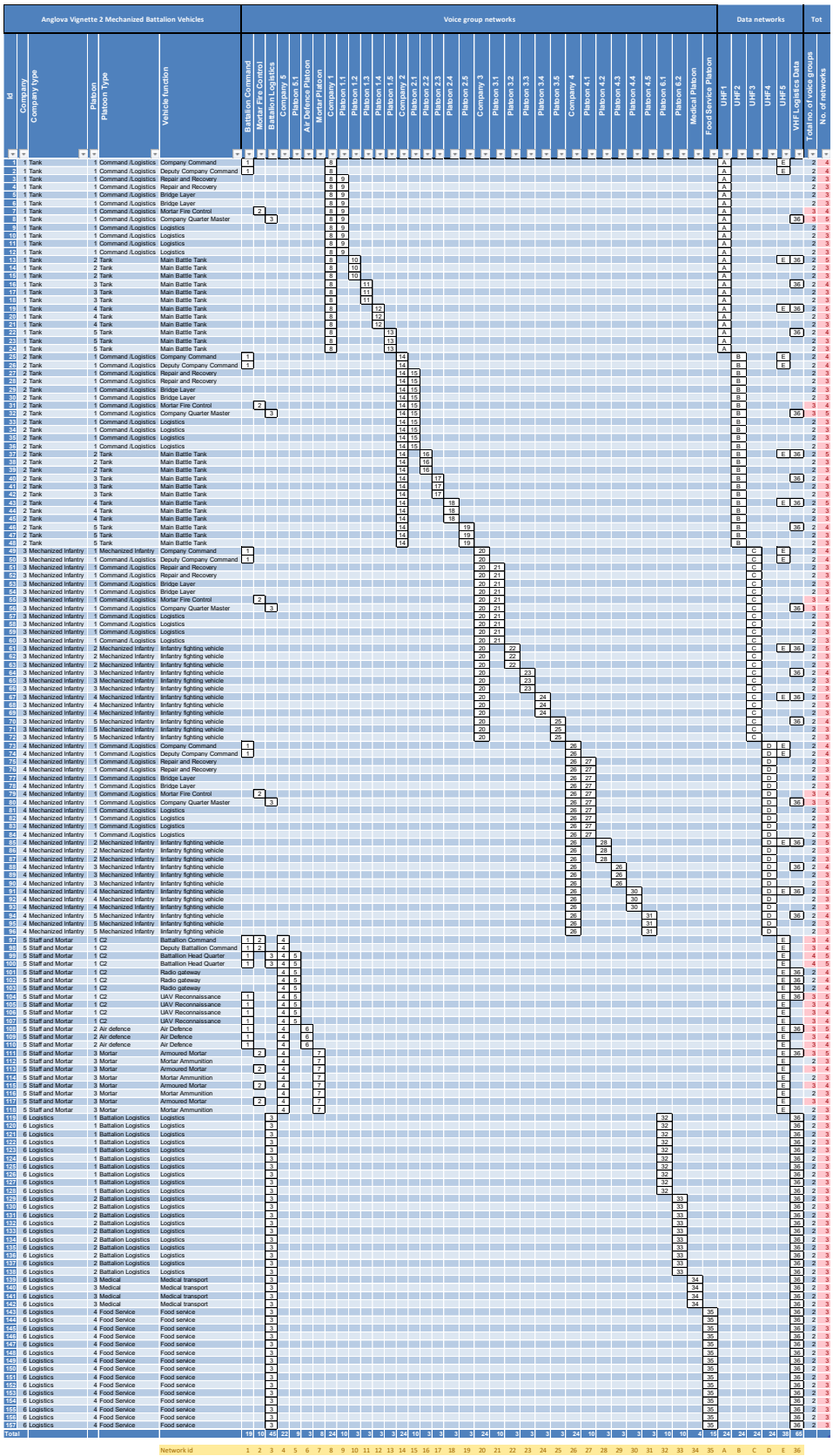


Figure A-2. Network plan 2 for Anglova Vignette 2.

Appendix B. NATO Security Documents and Standardization Efforts

This appendix gives an overview of relevant NATO documents, standards, and standardization initiatives within NATO. Some of the ones to consider when defining a security architectures and solutions for heterogeneous networks are as follows:

- Secure Communications Interoperability Protocol (SCIP) is an application-layer protocol that enables secure end-to-end communication over heterogeneous networks. The minimum bandwidth required is 2.4 kbps. A main benefit of SCIP is its secure cryptographic suite's auto-negotiation. This feature makes it possible to use SCIP equipment for multiple levels of security. SCIP is primarily designed for peer-to-peer communication, but it also supports point-to-multipoint communication. Since 2018, the work is organized under a partnership program administered by the NATO Communications and Information Agency (NCIA).
- Networking and Information Infrastructure (NII) Internet Network Encryption (NINE) is NATO's version of IP security. The NINE protocol operates at Layer 3 and protects all datagrams between peer NINE entities. It is geared towards peer-to-peer communication in fixed networks with sufficient bandwidth and good connectivity but also supports multicast/group communication. A NINE profile for tactical use was expected in 2020.
- Key Management Interoperability Specification (KMISpec) is an initiative from NCIA to provide a new NATO joint standard for key management. A draft specification has been provided with input from several nations. The KMISpec aims to standardize key management interfaces of end crypto units (e.g., SCIP and NINE devices), key fill devices, and other key management infrastructure components.
- Tactical Secure Voice Cryptographic Interoperability Specification (TSVCIS) is a specification that evolved from the Tactical Secure Voice Working Group formed by the U.S. National Security Agency in 2008 to ensure that modernized tactical secure voice devices will be interoperable across the DOD. The specifications define the voice encoding, encryption, framing, synchronization, key management, and other functions for tactical secure voice and data radio communications. TSVCIS is geared towards 16 kbps synchronous voice communication. NATO is defining Secure Tactical Communications–Interoperability Specification (STaC-IS) that will provide a mode that makes TSVCIS interoperable with SCIP.
- STaC-IS is the Interoperability Specification provided by the working group that seeks to provide interoperability between SCIP and TSVCIS. The target is to provide requirements for an interoperable secure tactical communication mode covering both voice and data. STaC-IS has now been

integrated in SCIP via the SCIP 214.6 module (not interoperable with the SCIP-210/233 modules) and in TSVCIS.

- NATO's Data Centric Security, Vision & Strategy document has been endorsed by NATO's Consultation, Command, and Control Board. Its vision is the enhanced ability to protect, control, and share data in accordance with evolving operational requirements strengthening the information superiority of the NATO Enterprise and Alliance Federation ("Delivery of shareable Alliance information, protected at source, controlled for life"). The document received multiple objections from several nations but was endorsed after all, as it is only a vision document.

List of Symbols, Abbreviations, and Acronyms

ABE	attribute-based encryption
ACK	acknowledgment
API	Application Programming Interface
ARL	Army Research Laboratory
ASCII	American Standard Code for Information Interchange
AuT	Analysis and Test Environment
BCN	company-level network
BFD	Blue Force data
BFT	Blue Force tracking
BMS	battlefield management systems
BON	overlay network
C2	Command and Control
CB	cooperative broadcast
CIS	communication and information systems
CoI	Community of Interest
CPU	central processing unit
CWIX	Coalition Warrior Interoperability eXercise
DAF	Deutsch-Amerikanische Freundschaft
DAVC	dynamically allocated virtual clustering environment
dB	decibel
DCS	data-centric security
DDIL	Denied, Degraded, Intermittent, Limited
DDS	Data Distribution Service
DEU	the German network
DEVCOM	U.S. Army Combat Capabilities Development Command
DisService	Dissemination Service
Docs	documents from headquarters
DR	delivery ratio

EMANE	Extendable Mobile Ad-hoc Network Emulator
EXI	Efficient XML Exchange
FKIE	Fraunhofer Institute for Communication, Information Processing and Ergonomics
FMN	Federated Mission Networking
FOI	Swedish Defence Research Agency
GDEM	Generic Data Exchange Mechanism
GPS	global positioning system
ICMCIS	International Conference on Military Communications and Information Systems
ICN	Information-centric Networking
ID	identifier
IER	Information Exchange Requirement
IP	Internet Protocol
IST	Information Systems Technology
JDSS	Joint Dismounted Soldier System
JDSS-IEM	Joint Dismounted Soldier System Information Exchange Mechanism
kbps	kilobits per second
KMI	key management infrastructure
LAN	Local Area Network
MAC	medium access control
MANET	mobile ad hoc network
MC	Military Committee
MFR	missing fragment requests
MILCOM	Military Communications Conference
MILS	multiple independent levels of security
MLS	multilevel security
MP	mobile producer
MPR	multipoint relay
MQTT	Message Queuing Telemetry Transport

NACK	negative acknowledgment
NATO	North Atlantic Treaty Organization
NATS	Neural Autonomic Transport System
NBWF	narrowband waveform
NDN	Named Data Networking
NDNLPv2	Named Data Networking Link Protocol version 2
NEM	network emulation module
NFFI IP2	NATO Friendly Force Information Interface Protocol 2
NII	Network and Information Infrastructure
NINE	Network and Information Infrastructure Network Encryptor
NLD	the Netherlands network
NMS	Network Message Service
NORM	NACK-Oriented Reliable Multicast
OLSRv2	Optimized Link State Routing protocol version 2
OPCIS	Operational Communications and Information Systems
OS	operating system
OTA	over-the-air
PID	Process ID
pps	packets per second
PT	plaintext
PTT	push-to-talk
QoS	quality of service
RAM	random access memory
RF	radio frequency
RFC	Request For Comment
RPC	Remote Procedure Call
RTG	Research Task Group
RTI	Real-Time Innovations
SA	security association
SAD	SA Database

SCB	synchronized cooperative broadcast
SCIP	Secure Communications Interoperability Protocol
SD	sensor data
SMF	simplified multicast forwarding
S-MPR	source-based multipoint relay
STaC-IS	Secure Tactical Communications Interoperability Specification
STANAG	Standardization Agreement
STO	Science and Technology Organization
SUSP	Simple UDP Segmentation Protocol
SVS	State-Vector-Sync
TA	treatment aggregate
TACCIS	Tactical Communication and Information Systems
TBD	to be determined
TCP	Transmission Control Protocol
TDMA	time division multiple access
TEK	traffic encryption key
TN	test node
TNO	Netherlands Organization for Applied Scientific Research
TSVCIS	Tactical Secure Voice Communications Interoperability Specification
TTL	time-to-live
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UHF	ultra high frequency
v	version
VHF	very high frequency
VLAN	virtual local area network
VM	virtual machine
VPN	virtual private network
WB	wideband

WBWF	wideband waveform
XML	eXtensible Markup Language
XMPP	eXtensible Messaging and Presence Protocol
XO	overlay
YAML	Yet Another Markup Language

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CI TECH LIB