Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 205 (2022) 147–156

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Military Communications and Information Systems (ICMCIS 2022)

A Cloud-based experimentation and analysis framework for services at the tactical edge

Mariann Hauge[a,*], Martin Asprusten[a], Tore J. Berg[a], André J. Bøhm[b]

[a]*Norwegian Defence Research Establishment (FFI), Instituttvn 20, 2007 Kjeller, Norway*
[b]*Sysint AS, Munkedamsveien 53B, 0250 Oslo, Norway*

Abstract

In this paper we present an experimentation and analysis framework that utilizes a public cloud infrastructure for most of the experimentation environment but also supports interaction with services installed on servers or dedicated hardware on-premises. The framework is enriched with a set of tools that allow for powerful scenario generation and modeling of radio communication in real time. We believe such experimentation and analysis frameworks can promote better cooperation between researchers, product developers and the military end users. We present an example experiment utilizing the framework, that studies the exchange of position updates between BMS applications on 4 moving vehicles and a forward command post, and share our lessons learned.

Keywords: emulation; cloud; tactical edge; military services; experimentation

1. Introduction

In order to ensure information superiority for the allied forces, the Information and Communications Technology (ICT) used must be continuously improved to include cutting edge technology. There will always be a desire to reduce the time needed to evolve a technology from research to use in the field. We believe that an important means to achieve this is to provide a flexible experimentation and analysis framework to both researchers, product developers and end users and bring these stakeholders together in an early stage of research and development. The framework should be able to represent realistic scenarios and include existing fielded services that the experimental software should interact with or be interoperable with.

An experimentation and analysis framework can be based on different types of experimentation environments. Typical alternatives include simulation-based environments, emulation-based environments, laboratory evaluation with actual hardware, limited field experimentation, and finally live military exercises. Each of these alternatives provides

* Corresponding author.
E-mail address: Mariann.Hauge@ffi.no

1877-0509
10.1016/j.procs.2022.09.016

respective advantages and disadvantages and has a role to play in the overall cycle from conceiving an algorithm to developing and deploying components on actual military hardware and systems. Some advantages and disadvantages with different types of environments are described in [24].

In this paper we present an experimentation and analysis framework deployed in a cloud environment and that utilizes emulation models for necessary components that cannot simply be represented with virtual functions (e.g. radio communication). The emulation-based environment provides a good compromise and stepping stone between simulations and using demonstrator software and hardware. In an emulation-based framework, the same software components may be used for test and experimentation as are used in demonstrators for field testing.

One challenge associated with emulation-based frameworks can be scalability. This could be an issue if the aim is to test large networks. However, efficient virtualization and cloud platforms that can access high performance servers, help mitigate this by making it easier to meet hardware requirements. Another challenge is the level of detail that is required of the emulation models used. This is particularly a challenge for radio models where channel interference, terrain, and medium access (MAC) methods can be complicated to emulate with adequate realism. It can also be more difficult to collect and process measurement results from an emulation-based environment running real software, compared to a purely simulation-based approach. In order for the experimentation and analysis framework to be easy to use, there is also a need for tools to create a scenario to drive the experiment.

We have built an emulation-based test and experimentation framework. In order to better understand the advantages and disadvantages of the framework, we performed a test experiment where we studied a military blue force tracking service in a mobile environment at the tactical edge. Cases in which communication systems are involved and create additional constraints, such as tactical communication links with limited communication capacity, are a major challenge for experimentation and analysis frameworks, due to the difficulty of emulating the communications systems in real-time. In order to overcome this challenge, our framework utilized a hybrid infrastructure consisting of a public cloud component in combination with high performance on-premises equipment that emulated the communications systems. In this paper we describe the framework as well as share our experience with the design and use of the framework.

In section 2 we present related work. Section 3 describes the test framework, tools and scenario that was used in the experiment. Section 4 presents the experiment used to validate the test framework. In section 5 we discuss the advantages and disadvantages with this environment, and give a conclusion in section 6.

2. Related work

Many approaches to creating a flexible experimentation and analysis framework exist. Most of the frameworks are not published, but here we include some of the published work that has inspired our work.

The Anglova scenario [24, 26, 25] was created by the NATO Science & Technology Organization's IST-124 task group on "Heterogeneous Tactical Networks - Improving Connectivity and Network Efficiency" [15]. The task group needed an easily accessible environment where collaborating researchers from different nations could combine their efforts in common test and experimentation with ICT in military tactical environments. The framework provides an emulation environment for radio communication and a realistic scenario with three vignettes. The first vignette focuses on Intelligence Preparation of the Battlefield, the second vignette on Troop Deployment, and the third vignette on an Urban Operation to Neutralize Insurgents. Documentation and much of the data needed to build the scenario (e.g., mobility pattern, transmission loss generation, traffic load etc.) is available at [1].

The Extendable Mobile Ad-hoc Network Emulator (EMANE) [5, 4] is an open source framework for emulating radio communication. EMANE supports real-time modeling of link and physical layer connectivity so that higher layer protocols (network protocols and application software) can be experimentally subjected to similar conditions that are expected to occur in real-world mobile, wireless network systems. A WiFi model, a basic time division multiple access (TDMA) model and a basic point-to-point radio channel model is included with the open source of EMANE. EMANE was used as the framework for radio emulation in the Anglova scenario.

The U.S Army Research Laboratory (ARL) has built a "Network Science Research Lab" (NSRL) to support research that aims to improve their ability to analyze, predict, design, and govern complex systems. This lab has a private data-center that powers their own cloud infrastructure, Dynamically Allocated Virtual Clustering Management System (DAVC) [18]. DAVC is a web-based virtualization service and cloud-operating environment capable of deploying

complex virtual experimentation clusters that can be used for simulation, emulation, and hybrid field/emulation experimentation. EMANE is one of the tools that is often used in the clusters. DAVC is designed to make it easy to create virtual clusters intended for experimentation with different network protocols and military services utilizing these networks. The environment is also able to perform experiments with flexible software defined networking (SDN) [19]. The Anglova scenario has been deployed on DAVC [15].

An analysis and test environment named AuT is described in [7]. AuT is a concept for creating fully virtual test beds for the integration of ICT systems in the tactical domain. This is very similar to both the purpose of NSRL/DAVC and the work presented in this article. AuT manages the virtual machines (VMs), the scenario and test execution, and includes the ability to do radio emulation. An environment that is capable of modeling a range of different radio types is described and integrated with AuT in [8].

Finally, a very flexible layer2 environment for connecting different sites together is described in [9]. This framework was built to cater for fully distributed experimentation between many nations for the NATO Coalition Warrior Interoperability eXercise (CWIX). This environment can be useful to connect virtual test beds with physical ones and potentially also connect dissimilar experimentation and analysis frameworks in different nations together, when needed. The paper also describes a logging and visualization tool that can ease the task of collecting and visualizing the results of the experiments.

3. The experimentation and analysis framework

The objectives of our experimentation and analysis framework was three-fold: 1) We wanted to explore if it was feasible to use a public cloud environment for the virtual elements of the experimentation and analysis framework in order to exploit the resource scalability that this provides and to avoid the cost of buying and maintaining a private data center. 2) We aimed to provide an environment that strives to build a digital copy of a set of typical military network infrastructures and fielded services. A software under test can then be installed in this environment and tested for interoperability with existing services in a realistic network infrastructure. 3) We also wanted to integrate a set of tools that we had access to for network emulation and test scenario generation. We expected that these tools could improve the flexibility and make the framework user-friendly. This flexibility and user-friendliness is important in order to create a framework that could cater for both researchers, product developers and end users.

Ideally, virtual machines or lightweight containers should be used to provide all protocols and services that are subject to experimentation and analysis, as well as services in operational use with which the tested software should interact. This would allow us to exploit the easy access and scalability of a cloud infrastructure. However, we expected that not all military services could be provided as virtual components. Therefore we also wanted to explore the ability to have a hybrid test framework with most of the functionality provided as virtual functions in the cloud and some of the equipment available in on-premises servers or specialized hardware.

We believe that an important role for an emulation-based experimentation framework is to provide a controlled and repeatable experimentation environment where demonstrator software can be tested. This can motivate the research communities to build prototypes of interesting technologies at an early stage. A second role for such environment is to enable for early integration testing of prototypes with other military services, in an emulated environment that can provide an abstracted, yet fairly realistic scenario. This would allow for more advanced testing, as well as large scale testing prior to experimenting with the software in field experiments. We expect that such an experiment framework can also bring together researchers, product developers and the soldier as the end user and thus motivate for more interaction between these groups.

3.1. Military scenario for the experiment

We chose to build a small land-based scenario at the tactical edge for our experiment. We expected this to be a difficult scenario to represent in a cloud-based infrastructure and would thus serve as a benchmark for what type of scenarios we could represent with the chosen design. We placed the scenario at the outskirts of Elverum in Norway. The reason for selecting this area was that we had detailed maps and terrain models easily available for the different tools that we wanted to integrate with the experimentation framework.

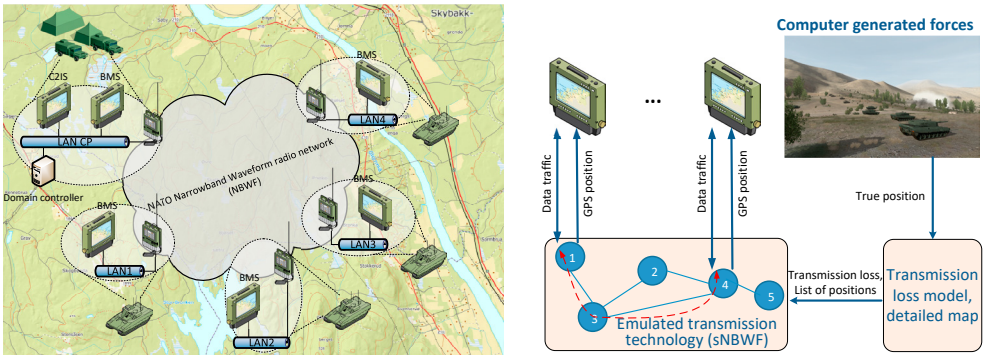

Fig. 1. (a) ICT to be tested in the experiment. (b) Interaction between the scenario generator, the transmission loss tool and the radio emulator.

In this terrain we deployed a company with four vehicles and a forward command post (CP) as shown in Fig. 1a. We wanted to run the tactical version of the Norwegian Battle Management System (BMS) on all military platforms in the scenario, and the Norwegian Command and Control system (C2IS) on the command post.

In order for these systems to interact we needed to model radio communication and included a model of a simplified version of the NATO Narrowband Waveform (NBWF) [20]. Since the scenario represents mobile devices at the tactical edge, we created a plan for how the participating vehicles should move in the terrain during the experiment, in order for the vehicles to move in and out of radio coverage. The actual movement was simulated by VR-Forces [17], a framework for generating Computer Generated Forces (CGF).

3.2. Tools

The following set of tools was used in order to build the experiment as described above.

Scenario generation. VR-Forces [17], a commercial off-the-shelf CGF system from MAK Technologies, was used to generate the routes for the vehicles in the terrain. The true position of the vehicles was shared over a High Level Architecture (HLA) [21] federation.

Emulation of NBWF. EMANE was used as the framework for radio emulation. The open-source version of EMANE only provides a WiFi model and two other basic radio models. We needed a low data-rate narrowband radio model in order to represent the most important radio communication for BMS and C2IS at the tactical edge for the Norwegian forces. For this, we developed a simplified NBWF EMANE model (sNBWF) for the experimentation and analysis framework. The EMANE model is based on a simulation model of NBWF described in [14]. The sNBWF EMANE performance (throughput, delay, etc.) were compared with the simulation model in order to validate the behavior of the model. The sNBWF EMANE model supports radio communication in real-time, which is necessary for an experimentation framework like this. More details about the sNBWF EMANE model can be found in [10].

Transmission loss generation. The sNBWF EMANE model needs the full matrix of transmission losses between all platforms at any given time in order to calculate the expected connectivity of the network. For a scenario with preplanned movements, this matrix of transmission losses can be precalculated with a given update frequency. This is the approach that is chosen for the Anglova scenario framework [1].

We, however, wanted to be able to calculate this matrix of transmission losses in real-time, in order to allow for the generation of new scenarios on the fly. This would also allow human-in-the-loop experiments were humans played the units involved, and their movements would therefore not be known in advance. We chose the WRAP technology of Altair's Feko application [6] to calculate transmission loss. This tool can use a range of channel propagation models combined with detailed 3-dimensional maps to calculate the transmission loss between senders and receivers with specified gain at specified heights in the terrain. For our experiment we chose to use the Detvag-90 [13] algorithm. The transmission losses calculated by WRAP were provided to the sNBWF EMANE radio model in order to calculate the signal strength at the receivers (see Fig. 1b).

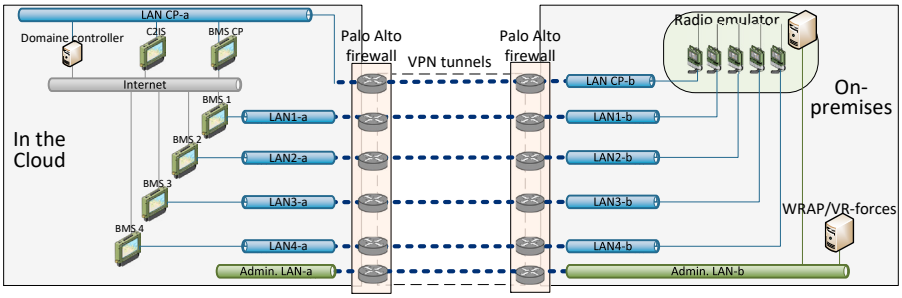

Fig. 2. The design of the experimentation and analysis framework utilizing both a public cloud and specialized on-premises hardware.

We decided to use an update frequency of 1 minute for the transmission loss matrix calculation. This is a computationally heavy task that would be difficult to do in real-time for very frequent updates and/or for a large network of nodes. For this experiment, which involved vehicles moving at a fairly slow speed in the terrain, we felt that an update rate of one minute was accurate enough. For other experiments, such as those involving fighter jets, the situation would be different, but for these platforms the channel propagation is often line of sight and thus does not need to involve very complex transmission loss calculation models.

Mediation service. Neither the WRAP tool nor EMANE supported the HLA interface used by VR-Forces to export information from the battle simulation. Therefore we created a mediation service that consumes position information from VR-Forces and creates transmission loss queries to the WRAP service. Further, the service translated the transmission loss matrices received from WRAP into a packet format that EMANE could consume and sent these together with the position information from VR-Forces to the sNBWF model in EMANE.

GPS service. Based on the positions received from VR-Forces, EMANE could generate GPS positions in the National Marine Electronics Association (NMEA) format. Each of the sNBWF radio instances in EMANE sent the GPS positions for each platform periodically on the local area network (LAN) on the platform. Services that needed GPS input (e.g. the BMS and C2IS service) could then consume the GPS information from their LAN.

Design. One of our goals was to use a public cloud as the infrastructure for much of the test and experimentation framework. We chose Microsoft Azure as the cloud provider. The services shown in Fig. 1a, ie. five instances of the BMS, one instance of the C2IS system and the domain controller together with the correct OS versions were installed on virtual machines in the Azure cloud.

The radio emulator requires very accurate timestamps on the messages. The clock accuracy is reduced and the jitter is increased when processes run in virtual machines in a datacenter environment [23]. Therefore we chose to run EMANE with the sNBWF model on a dedicated server on-premises. Another reason for doing this was to create a challenging hybrid setup with many network connections between the cloud and the on-premises equipment. We also chose to run the tool that generates the transmission loss matrices on-premises, since the tool required a physical license dongle. We could have set up a license server, making it possible to run the tool on a virtual machine in the cloud, but we did not explore this option for our example experiment. Since the radio emulator and the transmission loss tool were already located on-premises, it was practical to also run VR-Forces on-premises. The design of the experimentation and analysis framework is shown in Fig. 2.

As shown in Fig. 2 we needed to represent the small LANs on the vehicles and the forward command post in the test bed, and force the traffic generated by the services (BMS and C2IS) on these LANs through the emulated radio instance associated with each platform. The military BMS and C2IS services were located in the cloud and the emulated radio instances were located on-premises. Ideally we wanted to stretch a single layer-2 Ethernet connection for each of the emulated LANs between the cloud and the on-premises equipment. This design would give a close resemblance with the actual infrastructure on the emulated military platforms.

However, it turned out that the Software Defined Network (SDN) architecture under the hood of the Azure cloud environment did not provide a classical OSI stack layer-2 functionality in the Azure virtual networks. All network related Azure resources (e.g., virtual networks, route tables and network interface cards) available for setting up

the landing-zone for our experimentation and analysis framework in Azure represented layer-3 network functions. Therefore, we needed to use two network segments for each of the five LANs that we wanted to represent (Fig. 1a.) Each segment was allocated a class C address range, but this could have been smaller as long as each LAN segment had enough addresses available for connected devices. In our small test experiment, only the emulated radio node was connected to the on-premises segments, and a maximum of three nodes (at LAN CP) were connected to each of the segments in the cloud.

A set of VPN connections were used in order to connect the components in the cloud with the components on-premises. It turned out to be very difficult to keep the necessary traffic separation while using only a single VPN connection between the cloud and on-premises. It was also not possible to connect a single virtual machine to several virtual networks in Azure. The combination of these two limitations meant that we could not use Azure's own VPN solution. Instead, we used a firewall from Palo Alto Networks that could provide multiple VPN connections. A virtual machine containing this firewall was easily available on the Azure Marketplace for deployment in our landing zone. Clearly the chosen solution for connections between the cloud and on-premises equipment does not scale to a larger experiment. We discuss some possible solutions to this limitation in section 5.

It was quite cumbersome to configure the necessary Azure resources to build the small network seen in Fig. 2. Azure ignores all network configuration that is done in a guest VM and instead provides resources to create virtual networks, sub-networks, network interface cards and routing tables explicitly. The Azure platform is not designed to support dynamic routing experiments. But with the introduction of an emulated network environment this can be done in a layer above the virtual network that is created in the Azure landing-zone.

As seen in Fig. 2 we planned to provide an administrative network inside the experimentation and analysis framework and use this network to access all the nodes in the experiment (e.g. establish remote connections to the virtual machines in the cloud and bringing other information such as HLA data and time synchronization between the cloud and the on-premises equipment). This design would seal the test framework off from the Internet. However due to Covid and the need for flexible connections to the experiment we enabled access to all the VMs in the cloud via the Internet as well.

3.3. Automation

In order to create our small test experiment we needed to deploy more than 100 different Azure resources and do manual configuration of many of these. We needed to install the military software that we wanted to utilize in the experiment on the VMs, and configure these properly. We also needed to set up the firewall for the VPN connections, prepare the tools for the scenario as well as create the scenario and setup the emulation environment for the chosen network size. This was a time consuming task, and if the experimentation and analysis framework is to be used by researchers, product developers and end users, they can not be expected to have the time and competence to build this environment manually.

Fortunately, it is possible to automate the creation of a test environment almost completely. With an Infrastructure as code (IAC) orchestration tool such as Terraform [12], all the resources needed in the landing zone for a specific

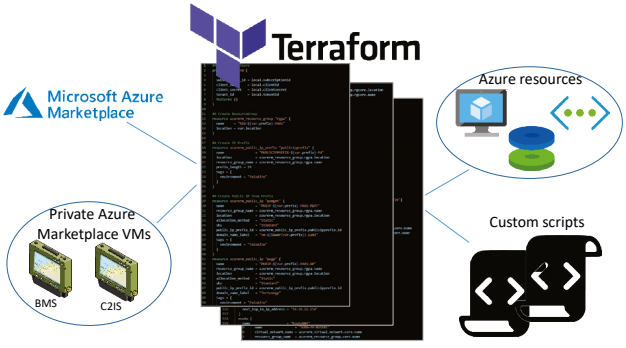

Fig. 3. Terraform and the four groups of resources needed to fully automate the creation of a test environment

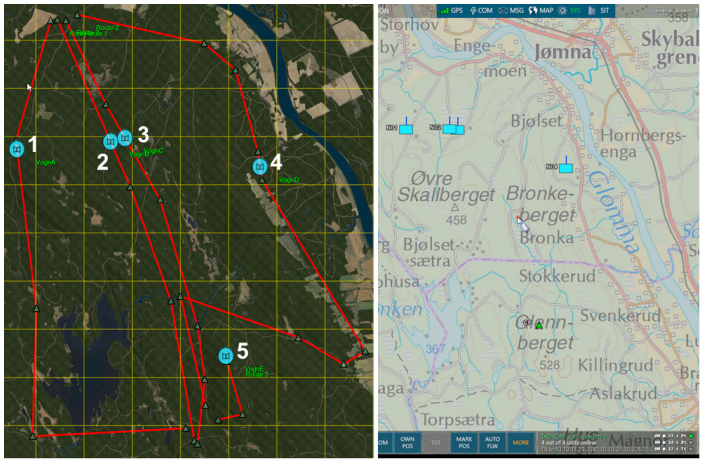

Fig. 4. The mobility pattern generated in VR-Forces and the corresponding blue force picture in the BMS application of the CP about 10 minutes into the experiment

experiment can be deployed using a simple ”Experiment x” *apply* command. The experiment can later be removed from the cloud with an ”Experiment x” *destroy* command when it is not needed any more.

In order to fully automate the establishment of a test environment like the one we built in our example experiment, a Terraform script rely on access to the following resources (see Fig. 3):

- VMs available in Microsoft Azure Marketplace that the script can deploy. E.g. the Palo Alto firewall VM template that we used to create the VPNs between the Azure cloud and our on-premises equipment.
- VMs available in a private Azure Marketplace. A selection of Azure VM templates with preinstalled important military services could be available as VMs in a private Marketplace (e.g. the BMS service and the C2IS service in our experiment.)
- Azure IAAS resources. Terraform scripts can create and configure the necessary Azure resources such as networks, routing tables, storage space, IP addresses, key-vaults, etc. for the experiment.
- A set of custom scrips that configures the instantiated VMs with user information such as setting up the VPN tunnels in the Palo Alto VM and setting call signs and a communication plan in the BMS application. The VMs that are created from the templates in the Marketplace does not have any individual configuration. The custom scrips can be called by Terraform.

Some initial effort is required in order to create a toolbox of VMs, custom scripts and Terraform scripts to build a set of scenarios, but with this in place, a deployment of one test environment can be created quickly and in as many copies as needed. With such standard environments in place, tailoring the environment for a planned experiment, for instance by installing experimental software, would require minimal effort.

4. Example experiment and evaluation

In order to validate the interaction between the different tools and the design of the experimentation and analysis framework, we performed an experiment that studied the packet loss of communications between moving units in a virtual terrain. The units exchanged position updates between the five BMS terminals on the military platforms of Fig. 1a, and moved according to a pattern created in VR-Forces (Fig. 4a). The scenario lasted for about 35 minutes and the BMS terminals were configured to send their position updates every 30s. It is difficult to see in Fig. 4a, but there is a small ridge called Bronkeberget between vehicle number 2 and vehicles number 1-3. The forward command post (vehicle number 5) in the bottom of the picture is placed on an elevated spot called Glennberget. The scenario was created to give challenging conditions for the radio communication between the rightmost vehicle and the three leftmost vehicles.

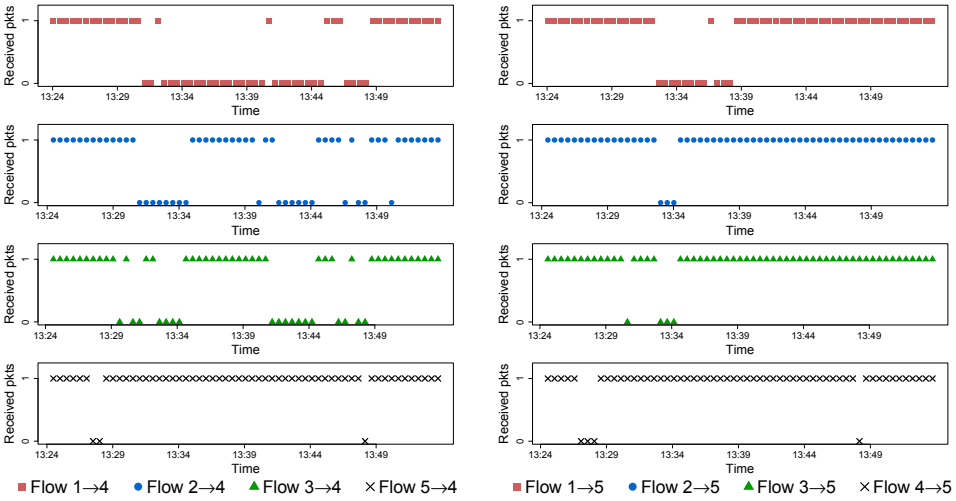

Fig. 5. (a) BMS position update packets received at vehicle number 4; (b) BMS position update packets received at the forward CP.

All the tools, VR-Forces, WRAP, the sNBWF model in EMANE, the mediation service and the GPS service, were used in this experiment. In addition to these tools we used Wireshark [3] on all the BMS terminals in the cloud to capture both sent and received BMS packets. Fig. 4b shows a screen capture of a remote desktop connection to the BMS terminal in the cloud that represented the forward command post.

We ran a series of repetitions of the scenario shown in Fig. 4 as well as some experiments with slightly different paths for the vehicles. We processed the captured Wireshark logs with R [2] and also observed the position updates on the BMS terminals during different stages of the scenario. In all experiments, the packet arrival rate seemed to be according to what we expected in the chosen terrain. Fig. 5a shows BMS packets received at vehicle 4 during a full scenario run. As expected, the highest packet loss happened on data-flow from vehicle 1. The packet loss from vehicle 2 and 3 that traveled close together was similar, and almost all packets that came from the forward command post in the elevated position were received. Fig. 5b shows BMS packets received at the forward command post. The CP received almost all BMS messages due to its elevated position.

The results were similar, but not fully identical, between different runs. Some of the reasons for the small differences were due to different random number seeds used in the radio emulator, and slightly different timings of the BMS position updates. In addition to the BMS tests, we also ran a successfull 48 hour long test with varying amounts of Poisson traffic generated by the multi generator (MGEN) [22] network test tool, to study the stability of the system.

5. Lessons learned

The experimentation and analysis framework we have built and studied in this paper can be used to test ICT software in different stages of development, from experimental software developed by researchers to new services or protocols under development by industry. The framework is enriched with virtual machines running existing fielded military services and emulation of radio communication as well as tools for scenario generation.

Our experience with utilizing Microsoft Azure's public cloud infrastructure as one of the building-blocks for the framework was good, with the exception of Azure's inability to host specialized virtual network infrastructure. Azure had powerful mechanisms for governance of the cloud environment and provided the level of control, configuration and easy reconfiguration that we needed. Documentation of how to deploy and use the different Azure resources was also plentiful. Azure requires that there is an Azure specific software layer installed on all VMs that are deployed in the cloud. This can potentially hinder some of the fielded military services from being installed on VMs in the Azure cloud.

As mentioned in section 3.2, the SDN design of Azure does not support hosting of specialized virtual network infrastructures, which hinders experiments that study dynamic routing and other protocols that operate on layer-3 or

below in the network stack. The network settings of the virtual machines in Azure is overridden by Azure specific resources to create virtual networks and static routing tables. It might be possible to utilize Virtual eXtensible Local Area Network (VXLAN) [16] to provide a virtual layer-2 network above the Azure virtual networks and do network experimentation over this virtual layer-2 network. We did not pursue this option for our test experiment. Another option can be to move the complete network part of the experiment into EMANE, however this could potentially reduce the flexibility and increase the complexity of the experiment design. Scalability might also be a problem. Currently Azure does not fully support multicast, which can also be a limitation for some experiments.

Utilizing a public cloud for the experimentation and analysis framework is appealing in order to exploit the scalability that this provides and avoid the cost of buying and maintaining a private data center. The drawback is that many military services might not be made available for deployment on a public cloud due to security limitations. Clearly no classified services can be made available, but this might also be the case for other services.

One of the ambitions behind our experimentation and analysis framework, was to provide an environment that strives to build a digital copy of a set of military network infrastructures and fielded services. The software under test can then be installed in this environment and tested for interoperability with existing services in a network infrastructure that resembles a realistic infrastructure. One challenge is that all existing services cannot be installed in the cloud, either due to the mentioned security issues, due to hardware requirements or other limitations. One solution to this can be to provide the services or network elements at servers or specialized hardware on-premises, but this might complicate network design. Another option is to create a model to emulate the service.

In our framework we included a model of a military radio network. Specialized hardware such as different types of sensors, or other components could also be modeled. However, the use of models means that the behavior of the environment will not be exactly like in the field. When we introduce models that emulate services there will be an associated modeling error that differs depending on the type of service that is modeled and the fidelity of the model. It is important to be aware that the digital copy of an environment that we represent in the framework will not be identical to the real system. However our claim is that an experimentation and analysis framework for emulation-based experiments has fewer sources for modeling errors than simulation experiments.

We chose to run the radio emulation on a dedicated server on-premises. The sNBWF EMANE model requires clocks to be synchronized to within 10 ms between the distributed processes. The clock accuracy we observed in Azure was not close to this requirement. We might be able to improve the clock synchronization by utilizing a short path to a Stratum 1 clock, but transmission delay between elements that might be deployed in different sections of the data center will still be a problem.

We also experienced that it was important to have a close dialog with the manufacturer of service that we wanted to represent on VMs in the framework. Support from the manufacturer for this job ensured that the services executes correctly in the cloud environment and helped to minimize the differences between the service behavior in the framework and in a real deployment.

Our effort to combine a tool for computer generated forces (VR-Forces) with the tool for real-time transmission loss generation (WRAP) and the radio emulation model (in EMANE) gave us a powerful and user friendly mechanism for scenario generation. This also allows for human-in-the-loop experiments using the framework where the virtual platforms are played by soldiers in real-time, using software such as Virtual Battle System (VBS) [11].

Finally, as discussed in subsection 3.3, we believe it is necessary to create a toolbox of resources such as models, virtual machines with pre-installed services, predefined scenarios with different sized units and mobility patterns (for mobile scenarios), and scripts to automate the deployment of the scenario in the cloud as much as possible. The toolbox must be actively maintained and support should be available to use the experimentation and analysis framework. This is particularly important in order to leverage the framework for better cooperation between researchers, product developers and the end users. Some governance must also be in place to administer the access and use of the public cloud and on-premises environment.

6. Conclusion

We have presented an experimentation and analysis framework that leverages a combination of public cloud infrastructure and specialized on-premises hardware. The intent was to build a digital copy of a set of military network infrastructures and fielded services. ICT software in different stages of development, from experimental software

developed by researchers to new services or protocols under development by industry, can be installed in this environment and tested for interoperability with existing services in a realistic network infrastructure.

The framework is enriched with an emulation model of a simplified version of an NBWF radio network and a mediation service that provides interoperability between a tool for computer generated forces, a tool for transmission loss generation and the radio model. This creates a powerful environment for scenario generation on the fly, and could also support simulation of realistic radio communication in experiments where soldiers play the battle progress in real time.

We claim that the presented experimentation and analysis framework can promote better cooperation between researcher, product developers and the end users in an early stage of research and development. We further believe that this is key to shorten the time from research to fielded products.

We observe that there are also challenges associated with such a framework that must be mitigated, such as the potential difficulties involved in representing all interesting services in the framework, modeling errors, the ability to do routing experiments, the need to develop a toolbox for automated test generation, as well as the need to maintain the toolbox and provide support for the use of the experimentation and analysis framework.

References

[1] Anglova scenario, . URL: https://anglova.net/.

[2] The R project for statistical computing, . URL: https://www.r-project.org/.

[3] Wireshark, . URL: https://www.wireshark.org/.

[4] Adamson, B., Claypool, D., 2013. Mobile network emulation – experiences and challenges, in: proc. IEEE MILCOM, pp. 1081–1086.

[5] Adjacent Link, . Extendable Mobile Ad-hoc Network Emulator (EMANE). URL: https://github.com/adjacentlink/emane/wiki.

[6] ALTAIR, . ®Feko Applications. URL: https://www.altair.com/feko-applications/.

[7] Angelstorf, F., Becker, A., Jansen, N., Noth, F., 2017. Analysis and test framework for the integration of ICT systems the tactical domain, in: proc. ICMCIS.

[8] Barz, C., Fuchs, C., Kirchhoff, J., Krzyzek, M., Brück, N., 2018. An approach to measurement-based tactical radio modeling and real time emulation, in: proc. ICMCIS.

[9] Barz, C., Kirchhoff, J., Niewiejska, J., Prahl-Kamps, M., Stavrou, I., Werberich, E., 2021. Distributed network experimentation in the NATO CWIX 2021 communications focus area, in: proc. IEEE MILCOM, pp. 360–365.

[10] Berg, T.J., 2020. Running norBMS over virtual radios. FFI-External Note 20/02876. Forsvarets forskningsinstitutt. URL: https://www.ffi.no/en/Publications.

[11] BISim, . Virtual Battle System (VBS). URL: https://bisimulations.com/.

[12] HashiCorp, . Terraform. URL: https://www.terraform.io/.

[13] Holm, P., Lundborg, B., Waern, A., Dec. 2003. Parabolic equation technique in vegetation and urban environments. Report FOI-R–1050–SE. FOI - Swedish Defence Reserach Agency. URL: https://www.foi.se/en/foi/reports.html.

[14] Libæk, B., Solberg, B., Oct. 2011. A simulator model of the NATO Narrowband Waveform physical layer. Report 2011/00533. FFI External-note.

[15] M. Hauge (ed.), Sept. 2019. Heterogeneous tactical networks – improving connectivity and network efficiency, Final Report. Report STO-TR-IST-124-PART-I. NATO STO. URL: https://www.sto.nato.int/publications/.

[16] Mahalingam, M., Dutt, D., Duda, I., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., Wright, C., Aug. 2014. Virtual eXtensible Local Area Network (VXLAN). Electronic Article RFC7348. IETF. URL: http://www.ietf.org.

[17] MAK Technologies, . VR-Forces. URL: https://www.mak.com/products/simulate/vr-forces.

[18] Marcus, K., 2014. Application of the dynamically allocated virtual clustering management system to emulated tactical network experimentation, in: proc. SPIE Defense + Security, pp. 26–38. URL: https://doi.org/10.1117/12.2054771.

[19] Marcus, K.M., Chan, K.S., Hardy, R.L., Yu, P.L., 2018. An environment for tactical SDN experimentation, in: proc. IEEE MILCOM.

[20] NATO, . Narrowband Waveform for VHF/UHF Radios - AComP-5630-5633 Edition A. STANAG 5630, Ed. 1.

[21] NATO, 2015. Modelling and simulation architecture standards for technical interoperability: High level architecture (HLA). STANAG 4603, Ed. 2.

[22] Naval Research Laboratory (NRL), . Multi-generator (MGEN) traffic generation tool. URL: https://github.com/USNavalResearchLaboratory/mgen.

[23] Parker, T.W., 2011. Experiences managing a parallel mobile ad-hoc network emulation framework, in: proc. ICWIN, pp. 449–455.

[24] Suri, N., Hansson, A., Nilsson, J., Lubkowski, P., Marcus, K., Hauge, M., Lee, K., Buchin, B., Mısırhoğlu, L., Peuhkuri, M., 2016. A realistic military scenario and emulation environment for experimenting with tactical communications and heterogeneous networks, in: proc. ICMCIS.

[25] Suri, N., Marcus, K.M., C. van den Broek, Bastiaansen, H., Lubkowski, P., Hauge, M., 2019. Extending the anglova scenario for urban operations, in: proc. ICMCIS.

[26] Suri, N., Nilsson, J., Hansson, A., Sterner, U., Marcus, K., Mısırhoğlu, L., Hauge, M., Peuhkuri, M., Buchin, B., R. in't Velt, Breedy, M., 2018. The angloval tactical military scenario and experimentation environment, in: proc. ICMCIS.