IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 19 August 2022, accepted 17 September 2022, date of publication 21 September 2022, date of current version 29 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3208283

RESEARCH ARTICLE

TNT: A Tactical Network Test Platform to Evaluate Military Systems Over Ever-Changing Scenarios

PAULO H. L. RETTORE[1], JOHANNES LOEVENICH[1], AND ROBERTO RIGOLIN F. LOPES[2], (Member, IEEE)
[1]Communication Systems (KOM), Fraunhofer FKIE, 53177 Bonn, Germany
[2]Secure Communications & Information Systems (SIX), Thales Deutschland, 92622 Ditzingen, Germany

Corresponding author: Paulo H. L. Rettore (paulo.lopes.rettore@fkie.fraunhofer.de)

This work was supported in part by Bundeswehr: BAAINBw—Federal Office of Bundeswehr Equipment, Information Technology and In-Service Support; and in part by the Bundeswehr Technical Center for Information Technology and Electronics—WTD81.

ABSTRACT This paper addresses the challenge of testing military systems and applications over different communication scenarios with both network conditions and user data flows changing independently. We assume that systems developed to handle ever-changing communication scenarios are more likely to be reliable and robust during real military operations. Therefore, we propose the Tactical Network Test (TNT) platform to automate the evaluation of military systems and applications over real military radios using a reproducible test methodology. TNT has four main goals (i) the creation of QoS-constrained data flows; (ii) the execution of models to change network conditions; (iii) the automation of experiments to quantify the performance of military systems over ever-changing communication scenarios; and (iv) the monitoring of quantitative metrics and performing data analysis. Our platform was used to execute experiments in a VHF network by sending uniformly distributed data flows during seven different communication scenarios, either generated by a stochastic model or mobility models. The experimental results are used to discuss the military system's performance by quantitative analysis using network metrics such as packet loss, delay, jitter, and data rate, and the test scenario characterization using mobility metrics such as speed, distance, and acceleration.

INDEX TERMS Testing tactical systems, military systems, ever-changing network scenario, mobility models, tactical networks.

I. INTRODUCTION

Tactical systems are multi-layer military systems with the means (e.g. control mechanisms and cross-layer information exchange) to deal with ever-changing communication scenarios, including unplanned link disconnections, and ever-changing user data flows. The communication scenarios are defined as ever-changing because the user data flows and network conditions are changing independently as a function of time during a mission. Therefore, tactical systems must support military operations by providing IP connectivity for network-centric warfare among mobile units in large areas. Moreover, tactical networks often combine different communication technologies (High Frequency (HF), Very High Frequency (VHF), Ultra High Frequency (UHF) and Satellite Communications (SatCom)), which are characterized by low bandwidth, high delay, and frequent link disruptions.

The scientific community has been designing military systems robust to frequent topology changes to meet Quality of Service (QoS) requirements from command and control (C2) services supporting the mission [1], [2], [3]. Therefore, test platforms for military systems must host models to generate communication scenarios with the user data flows and network conditions changing independently before using such

The associate editor coordinating the review of this manuscript and approving it for publication was Bidyadhar Subudhi.

a system on the battlefield. This fact motivated the present investigation to design and implement the Tactical Network Test (TNT) platform to support the development of robust military systems over real military radios.

Recent literature often uses static or periodic changes, reducing the test to a specific set of changes/scenarios, making the evaluation of the systems restricted to these contexts [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. Moreover, most of the literature has limited test scenario characterization, which is essential to ensure that the experiments are quantifying the performance bounds of the military systems. Therefore, TNT starts with the hypothesis that systems developed to handle both ever-changing network conditions and ever-changing user data flows are more likely to be reliable and robust during real military operations. TNT provides the means of systematically testing military systems before using them in the operational field.

The TNT platform includes two models ($\mathfrak{M}_A$ and $\mathfrak{M}_B$) to create different patterns of user data flows and network topology changes to test military systems over a wide range of communication scenarios. For example, in this article, we introduce a two-layer tactical system composed of a VHF radio and a queuing discipline shaping the user data flows as a function of the link data rate and the radio buffer usage. In addition, through a monitoring and data analysis approach, we are able to quantify the military system's performance using network metrics (e.g. link data rate, delay, jitter, and packet loss) and characterize the communication scenarios using a set of mobility metrics (e.g. traveled distance, speed, and acceleration). Furthermore, TNT is evaluated over seven different network scenarios created by model $\mathfrak{M}_B$ and data flows varying as a function of a distribution created by model $\mathfrak{M}_A$. In summary, the contributions of this paper are as follows:

- Design of a framework to automate the test of military systems/applications in tactical networks. The main design decisions are i) model $\mathfrak{M}_A$ to create data flows, (ii) model $\mathfrak{M}_B$ to create network conditions, and (iii) monitoring and data analysis;
- Introduction of two models to transform sequences of network states in a mobility pattern and vice-versa (models $\mathfrak{M}_{B_1}$ and $\mathfrak{M}_{B_2}$);
- Design and implementation of a mechanism to change the military radio modulation and to create link disconnections using a coaxial relay and controller between radios with wired antennas;
- Design, implementation, and test of an adaptive store and forward mechanism shaping the user data flow according to the network conditions, therefore, mitigating buffer overflow in military radios. This mechanism was tested with exemplary instances of TNT's models and the quantitative results illustrate the data analysis supported by the platform.

The rest of this paper is organized as follows, Section II discusses recent investigations also proposing test environments for military systems. Section III explains the design of TNT platform, describing the models and the methodology to create ever-changing communication scenarios in a test-bed with real military radios. Section IV compares experimental results using a set of patterns of link data rate change generated with stochastic models and mobility models. Lastly, Section V concludes the paper and lists future improvements.

II. RELATED WORK

This section discusses recent investigations introducing test platforms for tactical networks together with the literature using ever-changing communication scenarios as a methodology to evaluate military applications. Therefore, Table 1 compares the main aspects to test and reproduce the proposed methodologies, such as the methods used to create user data flows and network conditions in the test platforms and applications, the test environment deployed, and the reproducibility of the proposed methodology in terms of documentation and resources available.

A. EMULATED/SIMULATED TEST ENVIRONMENTS

The authors in [19] developed an emulation environment for heterogeneous tactical networks together with operation scenarios hosting several mobile units. The emulated test-bed, using Extendable Mobile Ad-hoc Network Emulator (EMANE), can support different communication technologies such as HF, UHF, VHF and SatCom due to its generic Radio Frequency (RF) propagation model. Moreover, they created different military scenarios called Anglova (Vignette 1 to 3) in the area of Fieldmont with 157 vehicles (network node), in total, moving over the course of two hours. This scenario was designed by military experts to be both realistic and publicly available.

In [4], the authors proposed a Tactical Network Integration Test Framework for simulation and emulation. Their solution is composed of three test environments, namely simulation, high-fidelity emulation, and scalable emulation. They started with a high-fidelity test-bed as a baseline for a comparative study. The study uses the same scenario for all three environments and tests them by comparing the performance and ensuring consistency in the experimentation. The authors also increased the network size to evaluate all test environments again comparing each environment against each other.

In [5], the authors proposed an emulated test-bed to host experiments with Software-defined Networks (SDN) solutions deployed in tactical scenarios [6], [20]. The emulation, also using EMANE, can execute various applications or traffic generators in diverse SDN configurations. The SDN layer can be configured to instantiate a specific network architecture hosting SDN-capable nodes. Additionally, the network scenarios change accordingly to the Anglova scenario [19]. The investigation reported in [21] developed a test-bed for Software-defined Tactical Network (SDTN) tests in an emulated environment using the Mininet-Wifi emulator. The experimental setup uses a specific SDN architecture composed of controllers in a distributed configuration, hosting one global controller and multiple local controllers collaborating to manage the network changes. The network can change as a result of node mobility and connectivity patterns together with user data flows, created by traffic generator tools. Moreover, the test-bed includes a visualization interface to show the network metrics. In the same direction, commercial solutions such as EXata [22] and iTrinegy [23] provide the means to create emulated networks to evaluate applications/services before deploying them in a real network environment.

These studies provide emulated/simulated test-beds, some of them focusing on SDN-capabilities, and some have general-purpose applicability. There are cases where the network changes are ensured by using Anglova scenario or mobility patterns, and for exploring the user data flows they use traffic generator tools or specific applications. However, most of these studies do not cover both network and user data flow changing independently, using limited means to characterize the test methodology by distinguishing and defining the user application, the military system, and the network topology. In other words, the limitations in the methodology description impose challenges to reproducing the results reported in most of these studies. In addition, other investigations [24], [25], [26], [27] have developed emulated test-beds (using OPNET, Joint Network Emulator (JNE) on top of EXata, among other simulators) for different purposes, therefore are not addressing the particular requirements of military systems designed to handle ever-changing communication scenarios at the edge of tactical networks.

B. TACTICAL SYSTEMS

Many studies reported in the literature have proposed military systems such as middlewares, brokers, or proxies to adapt the user data flow to the network constraints and challenges regarding resource-constrained radio devices in terms of buffer size, modulation, and power-constraints [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. Most of these investigations executed experiments in real test-beds or in emulated networks, using tools such as OPNET, NS 2 and 3, Omnet++, Net Emulator, EMANE, and Mininet. However, most of them have limited discussion about (i) how to adapt user data flows to the network conditions; (ii) military systems are evaluated mostly under stable or non-stochastic network and user data flow changes; (iii) they are not reproducible due to the lack of detailed methods description and resources used, or even (iii) the results are not quantified to be compared.

In [28] and [33], the authors proposed to distinguish the methodology of testing military systems in three problems A the user data flow, B the network changes and $A|B$ the military system that must handle both changes independently. For example, [32] solves problem A by introducing a model to generate ever-changing user data flows using a set of QoS-constrained messages. In [29] and [30], the authors introduced multi-layer control mechanisms doing flow control through hierarchical queuing in order to handle the network changes mitigating radio buffer overflow and packet loss. The proposed mechanism was evaluated over ever-changing communication scenarios using different patterns of data rate changes supported by real VHF radios [31].

TABLE 1. Related works comparison.

Papers	Test Methodology of Military Systems: User data flow	Network changes	Environment	Reproduc.
Test Platform				
[4], [19]	-	-	Emulated SDTN/TN	✓
[5]	Stochastic	Periodic	Emulated SDTN	✗
[21]	Stochastic	Any	Emulated SDTN	✗
[22], [23]	-	-	Simulated/ Emulated	✗
[24], [25]	-	-	Simulated/ Emulated TN	✗
[26], [27]	-	-	Emulated and Real TN	✗
Application				
[7], [14]	Periodic	Static	Simulated/ Emulated TN	✗
[8]	Periodic	Static	Real TN	✗
[9]	Stochastic	Static	Simulated/ Emulated TN	✗
[10], [15]	Periodic	Periodic	Real TN	✗
[11], [12]	-	Static	Simulated/ Emulated	✗
[13]	Periodic	Periodic	Emulated and Real TN	✗
[16], [28] [29]–[31]	Periodic	Stochastic	Real TN	✓
[32]	Stochastic	-	Simulated/ Emulated	✓
[33]	Stochastic	Stochastic	Real TN	✓
[17]	Periodic	Periodic	Emulated SDTN	✓
Test Platform with an Exemplary Application				
TNT	Any	Any	Real TN	✓

In summary, recent literature has a limited discussion about testing military systems over ever-changing communication scenarios in emulated and real environments. Meaning that most studies use static or periodic changes and may not develop reproducible and well-defined methodologies, restricting/lacking the evaluation and characterization of test scenarios, as listed earlier in Table 1. Different from the literature, we designed a general-purpose platform, TNT, designed to decouple the test of military systems into three layers the application; the military systems; and the network layer. Therefore, through defining models we argue that the TNT provides a reproducible methodology allowing any variation over the test scenario. Thus, the network can change using stochastic/mobility models or real traces, and the user data flow can be defined by either specific applications or stochastic models. In addition, TNT also provided a prototype to create link disconnection in real stationary radios with wired antennas.

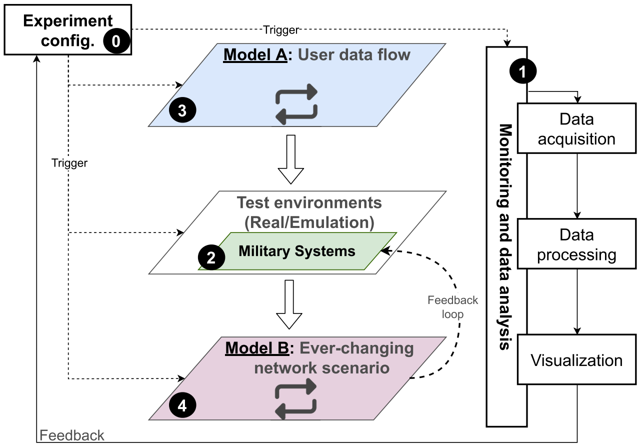


FIGURE 1. Design of the TNT platform.

C. METRICS FOR NETWORK SCENARIOS

The importance of categorizing the network scenarios to improve the description of realistic node mobility is discussed in [34]. Taking into account mobility metrics, the authors in [35] analyzed a list of available models that meet the requirements of tactical scenarios also identifying models that can be extended to meet such requirements. Regarding ever-changing communication scenarios, the present investigation advances our previous investigations [28], [31], [32] by introducing mobility models to create network scenarios that can be characterized by mobility metrics; modeling the communication area in order to convert any sequence of network states (link data rate) to a mobility trace and vice-versa; and wrapping these methods together with a monitoring and data analysis approach, providing a software platform to automate performance tests in tactical networks.

III. THE TNT DESIGN

In this section, TNTs platform is described aiming to automate tests to evaluate military systems and applications over a variety of network conditions and user data flows. The test results are composed of the system performance evaluation and the test scenario characterization. The TNT design is divided into three main components namely the military system being tested (**2**), the user data flow (**3**), and the network conditions (**4**), as illustrated in Figure 1. The experiments are orchestrated (configure/control and monitor) by two additional components, namely experiment configuration (**0**) and the monitoring and data analysis (**1**). The TNT's workflow is defined as follows:

- (**0**) *Experiment configuration*: defines the experiment configuration which triggers the monitoring of the network components through the monitoring and analysis (**1**); followed by the deployment of the military system (**2**), the creation of the user data flows (**3**) through Model $\mathfrak{M}_A$, and the network conditions (**4**) through Model $\mathfrak{M}_B$. After the end of the experiment (pre-defined by **0**), (**1**) is called again to process and analyze the experimental results.
- (**1**) *Monitoring and data analysis*: TNT collects the pertinent logs (sender and receiver IP packets, radio buffer, and radio modulation) during the experiment and processes them at the end of the experiment, triggered by (**0**), for quantitative analysis of the experimental conditions and the system performance.
- (**2**) *Military systems*: in this phase TNT deploys a middleware, broker, or proxy at the nodes in a test environment. This is the system subject of the performance tests executed by TNT;

In the application layer **(3)**, the user data flows are generated by a model that creates different patterns of data flows. From the network perspective **(4)**, TNT creates a topology that uses patterns of a data rate change or mobility traces, simulating a wide range of network scenarios in a communication area inspired by the wave propagation of omnidirectional antennas. These scenarios are deployed in real test-bed or emulated environments in order to test a given military system **(2)**. In this investigation, TNT is instantiated to deploy network scenarios in a test-bed with real military radios. The next sections discuss the main building blocks of the TNT platform in detail. Moreover, the models to create ever-changing network scenarios to test military systems, as well as the shaping mechanism and the disconnection prototype are available in a public repository.[1]

A. EXPERIMENT CONFIGURATION

The first step is to configure the experiment by using the two models $\mathfrak{M}_A$ defining the user data flow, and $\mathfrak{M}_B$ defining the network conditions and the test environment. The test environment can be either created by an emulation software or a real military test-bed; the experiments reported here were executed in a real test-bed, as described later in Section IV. At the application layer, TNT defines a model to create different patterns of data flows, allowing the users to input parameters such as the number of messages, size, and inter-message delays. Moreover, TNT uses model $\mathfrak{M}_B$ to configure the changes in the network by defining how the radio link data rate will vary during the experiments. This is done by selecting and providing one of the three inputs (i) a sequence of link data rates; (ii) a probability distribution (e.g. a Markov chain) for the corresponding network states; or (iii) a mobility trace (*x, y, time*) created by mobility models. Finally, TNT defines the network topology by setting the link data rates connecting them. The latter is set by parameters defining the communication technology, which has a range (kilometers), set of modulations describing nominal link data rates and the overall end-to-end delay.

B. CREATING USER DATA FLOWS

In order to introduce the element of chance at the application layer, TNT proposes a model to create user data flows that can simulate different applications through the generation of different traffic flows. For this purpose, we extend the

[1] https://github.com/prettore/TNT

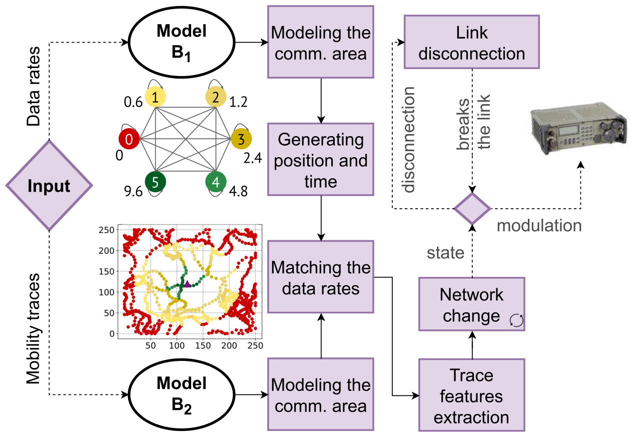


FIGURE 2. **Models for creating ever-changing network scenarios.**

user data flow model defined in our previous investigation [28], $Model_A(A, \lambda, f.m^{<\text{metric}>})$, where A defines the distribution of message type, the arrival time is defined by λ_A together with a distribution function f for all message metrics $f.m^{<\text{metric}>}$ such as {size, reliability, time of expire} to model $\mathfrak{M}_A$.

The extended version ($\mathfrak{M}_A$) also represents the variations of the user data flows created by different sequences of messages (modeled as a set of IP packets), message sizes, and sending rates. Formally, $\mathfrak{M}_A$ extends the data flow model by the quadruple $< \eta, \theta, \omega, \mu >$, where η defines the number of messages, θ the inter-message delay, ω the message size, and $\mu = < \rho, \mathfrak{R}, ToE, \cdots >$ again represents a m-tuple that brings m different QoS metrics such as ρ representing the message priority, $\mathfrak{R}$ addressing its reliability, ToE the time of expire and so on. In addition, η, θ and ω are defined as probabilistic functions sampling the respective metric by using a probability distribution chosen by the user. This representation increases the diversity of the outputs created by TNT and as a result, experiments can be conducted with more realistic user data flows instead of scenarios with static/periodic changes or specific applications only, meaning that the system performance will be measured based on these limited test conditions, and no further assumption regarding system robustness can be supported.

C. CREATING EVER-CHANGING NETWORK SCENARIOS

To create ever-changing network scenarios (model $\mathfrak{M}_B$ in Figure 1), TNT introduces two sub-models $\mathfrak{M}_{B_1}(\Sigma, S, \lambda, C, f, n_{ref})$ transforming sequences of data rates into mobility patterns and $\mathfrak{M}_{B_2}(\Gamma, C, n_{ref})$ transforming mobility patterns into sequences of data rates. The sub-models are independent of each other and are used to provide changes over the network based on a given input as shown in Figure 2. Moreover, both models also define parameters to configure the test environment in terms of topology, meaning that they can be used to initiate the changes in the network states through a radio interface and a link disconnection prototype. The input parameters for model $\mathfrak{M}_{B_1}$ are as follows: $\Sigma = (\sigma_0, \ldots, \sigma_{N-1})$ represents a sequence of data rate changes over the set of system states $S = \{s_0 = 0, s_1 = 1, s_2 = 2, s_3 = 3, s_4 = 4, s_5 = 5\}$ representing the nominal data rates $\{0.6, 1.2, 2.4, 4.8, 9.6\}$ kbps supported by the VHF radios (PR4G by Thales) in our laboratory and also shown in Figure 2. Notice that the system state space S can be defined in any desired way to match with the modulation of the radios used in the test environment.

Moreover, the sequence of states Σ can either be sampled using probability distributions, such as used in [28], [29], and [33], or manually defined by the user. The input parameter λ defines the time interval for updating one system state s_{t_n} at time t_n to the next system state $s_{t_{n+1}}$ with $n \in \{0, \ldots, N-1\}$, meaning that the model changes or disconnects the network link in the test environment according to the time distribution defined by λ. For example, assume that the movement starts at $t_0 = 0$ in system state $s_{t_0} = 2$, the next state $s_{t_1} = 0$ and $\lambda = 10$. Then, the link data rate between sender and receiver will be set to 1.2 kbps starting at time 0 and remains the same for exactly 10 seconds until the system changes to state 0, meaning that the link will be disconnected for the upcoming 10 seconds. Next, the parameter C represents the communication area with respect to the link quality defined by the current data rate $\sigma_n \in \Sigma$. Moreover, different distributions can be applied to sample the node positions over space using the probability density function f.

The following sections define two models, $\mathfrak{M}_{B_1}$ and $\mathfrak{M}_{B_2}$, to generate network scenarios. $\mathfrak{M}_{B_1}$ uses circular areas expanding around a reference node $n_{ref}(x, y) \in \mathbb{R} \times \mathbb{R}$ to define the communication area C. Using the communication area and inverse transform sampling, $\mathfrak{M}_{B_1}$ outputs an enriched mobility trace $\Gamma_E = \{(s_{t_0}, p_0, \sigma_0), \ldots, (s_{t_{N-1}}, p_{N-1}, \sigma_{N-1})\}$ composed by points $p_0, \ldots, p_{N-1}$ in a bi-dimensional vector space $V \times V$ distributed over time $t_0, \ldots, t_{N-1}$, matching the correspondent $\sigma_n \in \Sigma$. In contrast, $\mathfrak{M}_{B_2}$ is used to transform a mobility trace $\Gamma = \{(\lambda_0, p_0), \ldots, (\lambda_{N-1}, p_{N-1})\}$ into an enriched mobility trace $\Gamma_E = \{(s_{t_0}, p_0, \sigma_0), \ldots, (s_{t_{N-1}}, p_{N-1}, \sigma_{N-1})\}$ containing the respective link data rate σ_n. In other words, $\mathfrak{M}_{B_2}$ focuses on the transformation function, mapping the $p_0, \ldots, p_{N-1} \in \Gamma$ to the ring-shaped areas C with respect to a reference node $n_{ref}(x, y) \in \mathbb{R} \times \mathbb{R}$, describing a specific link data rate. It is important to notice that TNT assumes that the radios have an interface (e.g. Simple Network Management Protocol (SNMP)) to change the link data rates (radio modulation) or add an attenuation (e.g. using a channel emulator) to force the radio to change its modulation.

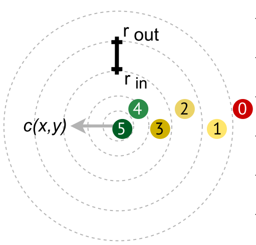
1) MODEL $\mathfrak{M}_{B_1}$: TRANSFORMING SEQUENCES OF DATA RATES INTO MOBILITY PATTERNS

Model $\mathfrak{M}_{B_1}$ creates mobility patterns from a given sequence of data rates (Σ), resulting in an enriched trace $\Gamma_E = \{(s_{t_0}, p_0, \sigma_0), \ldots, (s_{t_{N-1}}, p_{N-1}, \sigma_{N-1})\}$, that describes a given mobility pattern. TNT executes the following steps to transform a sequence of data rates into a mobility trace: First, we define a mapping of the system state S to a set

$A_0, \ldots, A_{N-1}$ of circular areas inspired by the wave propagation of omnidirectional antennas. After that, we use inverse transform sampling to create the positions $p_0, \ldots, p_{N-1}$ and the resulting enriched mobility trace Γ_E. Besides that, the mobility statistics from Γ_E are extracted and the resulting trace is used to change the radio link data rate or create link disconnections. The following paragraph describes this process in detail.

a: MODELING THE COMMUNICATION AREA

The communication area $C = A_0 \cup \cdots \cup A_{|S|-1}$ of a node that can be in $|S|$ different system states as the composition of $|S|$ nested circular areas $A_0, \ldots, A_{|S|-1}$. In this formulation, the radius r_i of the circular area $A_i \in \{A_0, \ldots, A_{|S|-1}\}$ is defined through the system state of the corresponding node, meaning that the area of communication can be described by slicing the circle for the maximum radius $r_1 < \infty$ (system state s_5), resulting in $|S|$ rings of communication. Assuming that a disconnection is mapped to the area beyond the maximum radius r_1, meaning that $r_0 = \infty$, the boundaries of each annular regions are defined by $\mathcal{B} = \{[0, r_{|S|-1}], \ldots, [r_2, r_1], [r_1, \infty]\}$. In this investigation, we require both boundaries of the annular regions to have a magnitude of M in meters, as defined in (1), allowing us to create different ranges of communication. Moreover, our test cases consider an environment area of 25 square kilometers and $M = 2$ km, therefore, modeling the node communication area with a radius of 20 km. We chose this configuration since the communication range depends on the respective radio technology, and thus we are inspired by the VHF radios in our laboratory having such capabilities.



$$\begin{aligned} A_5 &= \mathrm{ann}(c(x,y), 0 \cdot M, 2 \cdot M) \\ A_4 &= \mathrm{ann}(c(x,y), 2 \cdot M, 4 \cdot M) \\ A_3 &= \mathrm{ann}(c(x,y), 4 \cdot M, 6 \cdot M) \\ A_2 &= \mathrm{ann}(c(x,y), 6 \cdot M, 8 \cdot M) \\ A_1 &= \mathrm{ann}(c(x,y), 8 \cdot M, 10 \cdot M) \\ A_0 &= \mathrm{ann}(c(x,y), 10 \cdot M, 25 \cdot M) \end{aligned} \quad (1)$$

As a result from the previous observations, each system state $s_i \in S$ can be mapped to a ring-shaped area $A_{|S|-1} = \mathrm{ann}(c(x,y), 0, r_{|S|-1})), \ldots, A_0 = \mathrm{ann}(c(x,y), r_1, r_0))$ with center $c(x,y)$. The resulting communication area $C = A_0 \cup \cdots \cup A_{|S|-1}$ defines the sample space to create the points $p_0, \ldots, p_{N-1}$ of the raw mobility trace $\Gamma_R \subset \Gamma_E$ using the probability density function f. Note, that the representation for the communication area is inspired by the wave propagation of omnidirectional antennas related to the radios used in our laboratory. But, it can be replaced to work with any other wave propagation model by defining the mapping from the system states S to any arbitrary set $A_0, \ldots, A_{|S|-1}$ representing the area of communication C.

b: GENERATING POSITION AND TIME

Given a node with position $c(x,y)$, TNT implements inverse transformation sampling to create the points $p_0(x_0, y_0), \ldots, p_{N-1}(x_{N-1}, y_{N-1})$ of the raw mobility trace Γ_R, sampled from the circular areas A_i generated in the last step. For this purpose, let us assume that we have a reference node, which can be a base station or a neighboring node sharing its location, with position $n_{ref}(x,y)$. Depending on the link quality (system state) of the node with position $c(x,y)$ and the reference node $n_{ref}(x,y)$, TNT computes the boundaries of the annulus A_i, as shown in (1). In simple terms, this annulus describes the space of points matching the communication areas of the two nodes using their distance $|c(x,y) - n_{ref}(x,y)|$. Inverse transform sampling now enables TNT to sample a point from these areas of communication depending on the link quality between the two nodes.

Therefore, let X be a random variable describing the distance r of a point $p(x,y)$ to the center $c(x,y)$ (current node position) and r_{in} and r_{out} the lower and upper bounds of $b_i \in \mathcal{B}$ defining the inner and outer radius of the communication area A_i. Then, the Probability Density Function (PDF) describing the relative likelihood of the distance of a point to the center $c(x,y)$ is shown in Eq. (2). Furthermore, in Eq. (3), using the PDF, we get the value of the Cumulative Density Function (CDF) for distance d by calculating the integral of $f(r)$ from r_{in} to d.

$$f(r) = \frac{2r}{r_{out}^2 - r_{in}^2} \quad (2)$$

$$F(X = d) = \int_{r_{in}}^{d} f(r)dr = \frac{d^2 - r_{in}^2}{r_{out}^2 - r_{in}^2} \quad (3)$$

Moreover, the inverse of $F(X = d)$ in d is:

$$d = \sqrt{u\left(r_{out}^2 - r_{in}^2\right) + r_{in}^2} \quad (4)$$

Now, TNT uses inverse transform sampling, Eq. (4), to generate values of X which are distributed according to F. This works as follows: (1) Generate a random number u_1 from the standard uniform distribution in the interval [0, 1]; (2) find the inverse of the desired CDF $F^{-1}(d)$ and (3) compute $X = F^{-1}(u)$.

Once TNT sampled the distance $X = d$, it computes the x and y coordinates of the point $p(x,y)$ as shown in Eq. (5) by first sampling the corresponding angle α with respect to a uniformly distributed random number u_2 and calculating x and y using α afterwards. As a result, it generates a sequence of positions that we call *Raw trace* Γ_R. Moreover, TNT can be specified to vary this model by restricting position $p(x,y)$ at time t_n to be the nearest neighbor of the node position at time t_{n-1} with distance d from center $c(x,y)$, meaning that TNT chooses α such that it minimizes the *euclidean* distance of both positions. This approach allows the creation of new traces, named *Shortest trace*, restricting the node movements to the shortest distance using the same uniform coordinates distribution.

$$\alpha = 2 \cdot \pi \cdot u_2$$
$$x = d \cdot \cos \alpha$$
$$y = d \cdot \sin \alpha \tag{5}$$

Finally, in the last step, the model outputs a trace file with node positions and link data rate σ_n (node state) at a given time t_n, $node_{t_n} = [x, y, t_n, \sigma_n]$. In this sense, to create positions over time t_n TNT defines when the node starts to move s_{t_n} and how long the node will stay at the current state before moving to the next state $s_{t_{n+1}}$. For example, $s_{t_0} = 0$ means that the node starts moving from the beginning of the experiment and, on the other hand, $s_{t_0} = 60$ indicates that the movement will start after 60 seconds.

To create a new variation of the *Raw trace* Γ_R, TNT uses linear interpolation to connect the points in space-time. This process outputs a new trace file, named *Filled trace* Γ_E, with smooth movements. Then, for each node position and time $node_{t_n} = [x, y, t_n]$ and $node_{t_{n+1}} = [x, y, t_{n+1}]$, TNT interpolates n_p new positions, as defined in Eq. (6). The parameters used to generate the sequence of states as well as the stochastic model used as input to this model will be discussed later in Section IV together with the results and statistics of the experiments. It should be noted, that s_{t_n} is distributed by λ_n for Γ_R, but not for the *Filled trace* defined by Γ_E. Therefore, we cannot label the state of these new node positions in between by replicating the last state or interpolating them, because the state changes based on the communication area model, as discussed in *Modeling the communication area*, thus requiring one more step called *Matching data rates* to map the node trajectory within the communication area as described next.

$$p_x = x_n + (x_{n+1} - x_n)/n_p$$
$$p_y = y_n + (y_{n+1} - y_n)/n_p$$
$$p_t = t_n + (t_{n+1} - t_n)/n_p \tag{6}$$

c: MATCHING DATA RATES

Here, TNT matches the node position to the respective ring specified previously in (1) and assigns the positions to the correspondent network state. At this point, TNT ensures that any trace must follow the communication area modeled before.

d: TRACE FEATURES EXTRACTION

With the node's trace $node_{t_n} = [x, y, t_n, \sigma_n]$, TNT performs the features extraction in order to describe and characterize the node behavior based on its mobility. It computes metrics like distance from the base station (reference node), travel distance, speed, and acceleration between positions, adding these features to the final trace file. The goal is to enrich the visualization and analysis of test scenarios used in a particular experiment.

2) MODEL $\mathfrak{M}_{B_2}$: TRANSFORMING MOBILITY PATTERN INTO SEQUENCES OF DATA RATES

Model $\mathfrak{M}_{B_2}(\Gamma, C, n_{ref})$ transforms mobility patterns into network states and is used to change the radio modulation or cause link disconnections during the experiment. It should be noted that TNT is designed to work with any tool that implements mobility models and exports a trace file, such as MobiSim [36] and BonnMotion [37]. For simplicity, this investigation focuses on the usage of BonnMotion due to its flexibility and support of a wide variety of models like Random Waypoint (RWP), Random Walk (RW), Probabilistic Random Walk (PRW), Gauss-Markov (GM), Manhattan Grid (MG), and Disaster Area (DA). Thus, TNT defines the mobility scenario and the corresponding parameters using BonnMotion, parses the resulting trace file Γ and uses the mobility trace to define the communication area C with respect to reference node n_{ref}, as discussed before. Then TNT uses the resulting set of communication areas $A_0, \ldots, A_{|S|-1}$ to match each position with a data rate representing the tactical radio modulation. Therefore, $\mathfrak{M}_{B_2}$ reuses the methods defined by $\mathfrak{M}_{B_1}$, namely *Modeling the communication area*, *Matching data rates*, and *Trace features extraction*, as discussed earlier in Section III-C1.

3) NETWORK CHANGE

To create network changes based on the mobility traces, TNT uses the radio's SNMP interface to change the nominal link data rates and the link disconnection prototype to create disruptions. Since our test-bed is using radios with wired antennas inside a laboratory, it is not possible to create disconnections moving nodes away from the communication range like in emulation platforms. Therefore, TNT proposes a low-cost and easy-to-use link disconnection (state $s_0 = 0$) prototype designed with a controller (Raspberry-Pi) and a coaxial relay. Notice that, TNT could also connect to a channel emulator creating link disconnections using a management interface. The documentation describing how to build this prototype and its codes are available in a public repository.[2] More precisely, the changes on the network are done by two different interfaces depending on the node state (i) *connected*: SNMP interface to the tactical radio (to change its modulation) and (ii) *disconnected*: interface to the link disconnection prototype *'cutting'* the wired antenna using a relay.

D. MILITARY SYSTEMS

After introducing TNT's mechanisms to create ever-changing user data flows and network conditions, the next step is to design a military system in order to test the platform over different scenarios as illustrated in Figure 1 (2). In previous investigations, we observed that the limited buffer size of military radios combined with the ever-changing network scenarios and user data flows (say message size and time

[2]Link disconnection prototype: https://github.com/prettore/link-disconnection-prototype.git

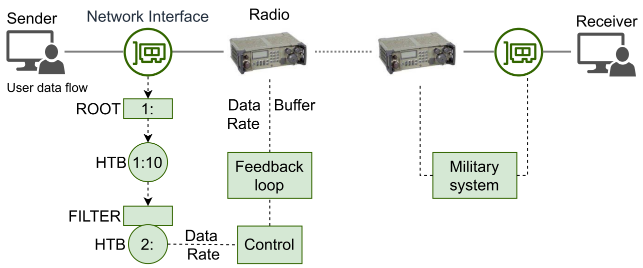


FIGURE 3. **Queuing discipline for military systems.**

distribution) results in buffer overflow [16], [38]. Therefore, we created a mechanism that can handle the radio buffer overflow by dynamically shaping the user data flow, and controlling the buffer usage within a given threshold. Such a mechanism is considered here as part of a military system supported by the TNT, but any other system could be either deployed on the network. The adaptive shaping mechanism works using Linux Queueing Discipline (QDISC) and information gathering from the radios, such as data rate and buffer usage through SNMP. Figure 3 shows the design of the hierarchical queues as well as the control mechanism to change the queue rules based on the data rates (which is an abstraction of the node's mobility for the real test-bed) and buffer occupancy reported by the radio.

TNT avoids buffer overflow using Traffic Control (TC) to dynamically shape the data flow within a control loop, which gathers the radio buffer occupancy and links data rate as feedback. Then, TNT creates two queues, where the first Hierarchical Token Bucket (HTB) queue is responsible to shape the data flow based on the maximum data rate supported by the radios, which is 9.6 kbps in our test-bed. This queue holds bursts of messages based on the radio communication boundary. Next, TNT filters all interested data traffic (sender and receiver IP address, UDP packets only, and even the application port), restricting the queues to have only the desired traffic. Notice that, these filters can be changed based on the test goals. Then, the second HTB queue is responsible to shape the data based on the current link data rate.

The control mechanism updates the dequeue rate DQ_r, resulting in increasing or decreasing the inter-packet delay at the sender. The computation of DQ_r uses the buffer occupancy ΔB and the data rate Δd reported by the radio. Moreover, the user can define the buffer threshold b, the buffer warning area w_b, and the percentage of data rate reduction r_d. Eq. (7) shows the conditions to change the QDISC rules shaping the data flow whenever the network changes, therefore, controlling the buffer usage.

$$DQ_r = \begin{cases} \max(\min(\Delta d), \Delta d * r_d) & \textit{if } \Delta B \geq b \\ \max(\Delta d) & \textit{elif } |\Delta B - b| > w_b \\ \Delta d & \textit{otherwise} \end{cases} \tag{7}$$

The proposed mechanism is explained below by instantiating three possible cases defined as follows: 1) suppose the current data rate is $\Delta d = 4.8$ kbps, the max data rate supported is max($\Delta d = 9.6$ kbps), the min($\Delta d = 100$ bps), the buffer threshold is $b = 50$ %, the warning area is $w_b = 10$ % and the current buffer occupancy is $\Delta B = 20$ %. Then, the second condition is satisfied ($|20 - 50| > 10$) and the HTB bitrate will be set to the maximum data rate supported by the radio. The goal is to fill up the buffer as fast as the radio supports before the network condition gets worse; 2) now, the buffer occupancy reached $\Delta B = 41$ %, then the third condition is satisfied ($|41 - 50| < 10$) and the dequeue rate is set to $DQ_r = 4.8$ kbps, reducing the delivery rate from the queue to the radio buffer; 3) finally, if $\Delta B \geq 50$ %, then the first condition will reduce the current DQ_r in $w_b = 10\%$ until the buffer occupancy keeps lower than the threshold. In short, the control mechanism changes the dequeue rate like $DQ_r = \{4.8$ kbps, 4.32 kbps, 3.88 kbps, 3.49 kbps, ..., 100 bps$\}$ until the buffer usage ΔB is close to the pre-defined threshold. If the link data rate changes, the system will adapt its dequeue rate DQ_r in a similar way.

TNT was designed to test such a control mechanism within military systems over a wide range of communication scenarios.

E. MONITORING AND DATA ANALYSIS

Having everything needed to execute an experiment, the next step is to deploy a monitoring and data analysis mechanism as shown in Figure 1 (1). TNT process the data gathered from both sender and receiver nodes and the trace files, analyzing and visualizing these data together. The monitoring and data analysis phase was designed to allow for quantitatively examining the experiment, helping to highlight issues regarding the test environment and quantifying the robustness of systems such as routing protocols and tactical middlewares. Including the monitoring of the end-to-end enforcement of QoS requirements from command and control applications. Next, we describe each step of this phase.

1) DATA ACQUISITION

This process was partly developed in our previous investigation [29] to collect all contextual data of both sender and receiver nodes such as IP packets, and radio features (buffer and modulation) and store them in a centralized database which TNT uses to conduct further analysis.

2) DATA PROCESSING

To quantify the military system performance, TNT process the IP packets from both sender and receiver nodes. The IP packets are sorted and combined as a function of time. In sequence, TNT process the time series extracting network metrics, such as *data rate*, *radio buffer*, *delay*, *jitter*, and *packet loss*. Finally, TNT exports the statistics in two different files: one compiling the overview of the experiment with basic statistics, such as *min, max, average, and standard*

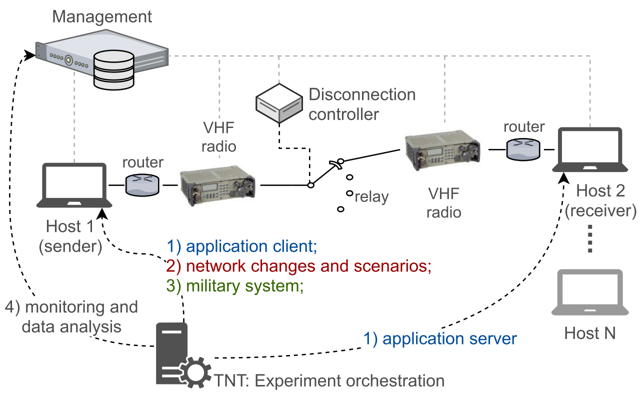


FIGURE 4. Test environment with real VHF radios.

TABLE 2. Experimental setup.

Components	Description
Radio	2 PR4Gs supporting {0.6, 1.2, 2,4, 4,8, 9.6}kbps
Nodes	2 i7, 8 GB RAM laptops
Management	TNT orchestrating the experiment and a data base
Link disconnection prototype	Raspberry-Pi 3; Relay (CX230L); Step-up DC-DC Converter; 50-ohm attenuator
User data flow	2.000 UDP packets of 1264 Bytes MTU each, following a uniform distributions $\eta_u = (a, b)$ where $a = 1$ and $b = 50$ packets per message/s
Network scenarios	7 scenarios during 170 min each using model $\mathfrak{M}_{B1}$ and model $\mathfrak{M}_{B2}$
Communication area	25 square kilometers and each data rate has 4 kilometers range as described in (1)

deviation, and the second file shows the entire log in a time series.

3) VISUALIZATION

Lastly, TNT automatically generates graphics plotting the quantitative results after each experiment. The goal is to support the user in making decisions regarding the test environment, the user data flow (application layer), the network condition (physical layer), and the military systems (application, transport, and IP layers). For example, all the plots discussed later in the evaluation section (Section IV) were generated using the *visualization* process.

IV. EVALUATION

This section discusses quantitative results from different experimental configurations selected to demonstrate TNT supporting the tests of an exemplary military system over ever-changing communication scenarios. The experiments are conducted in a VHF network over seven network scenarios generated by different instances of TNT's models. First, the test environment and the user data flows are described. Then, it is discussed the experimental results observed during experiments with different instances of model $\mathfrak{M}_{B_1}$ and model $\mathfrak{M}_{B_2}$, as introduced earlier in Section III.

A. TEST ENVIRONMENT

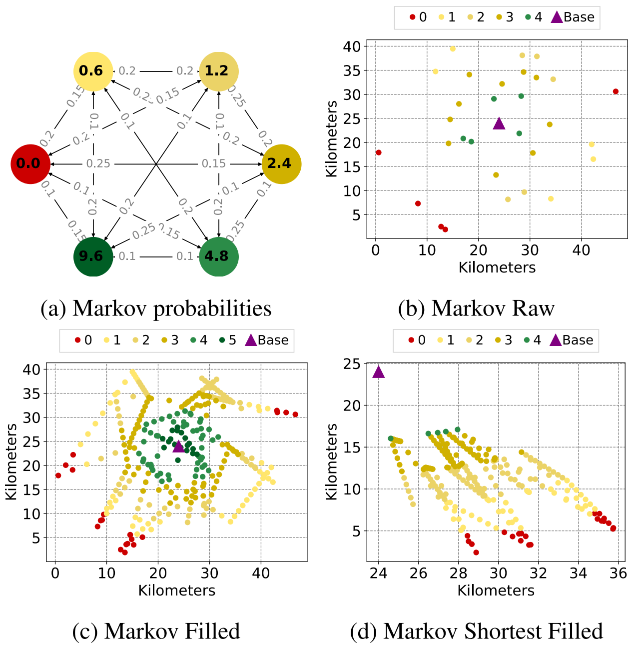
TNT was instantiated to perform experiments over real military radios, thus the infrastructure consists of VHF radios, Linux QDISC, the link disconnection prototype, and a data monitoring and analysis process to measure the data sent and received, computing the packets and the trace statistics, as illustrated in Figure 4. In this figure, TNT orchestrates the experiment by deploying all necessary scripts over the network nodes before the experiment's execution. After the user configures the experiment (as described earlier in Section III-A), TNT starts the experiment deploying (1) the application on the sender and receiver nodes (to generate the user data flows); (2) the mechanism to change the link data rate and the network scenarios based on mobility traces in the sender node, which has an interface with the radio and with the link disconnection prototype; (3) the military system (here the adaptive shaping mechanism for user data flows) on the sender node in order to avoid radio buffer overflow; and (4) the monitoring and data analysis scripts which collect, prepare and plot the experiment outputs.

The tactical network is composed of two VHF radios (PR4G), with 128kb of buffer size and supporting five data rates {0.6, 1.2, 2.4, 4.8, 9.6} kbps, each connected to a node (sender and receiver, respectively). The radio antennas are wired and connected to a link disconnection prototype in order to create link disconnections, state {0}. Then, the network condition is changed using mobility traces as described in Section III-C. Moreover, a server in a management network is used with methods to conduct the data monitoring and analysis, acquiring, processing, and plotting all logs. In the next sections, we describe the user data flow and the patterns of data rate changes used in the experiments, then discuss the experimental results. The experimental setup with all components used by TNT is listed in Table 2.

B. USER DATA FLOWS

The data flow can be generated in a variety of ways such as using specific applications in order to evaluate the enforcement of end-to-end QoS requirements or using a traffic generator designed to create different data flows. TNT uses the second approach to create traffic flows that can test military systems or simulate different applications based on different distributions of data flows. For example, some widely used traffic generators are *Iperf*, *Netperf*, *Distributed Internet Traffic Generator (D-ITG)*, *Multi-Generator (MGEN)* [39]. Traffic generators support stochastic models for packet size and inter-departure time to simulate user data flows. These features can be found in the D-ITG tool as described in [40] and [41] and for this reason, TNT used it to generate and measure the user data flow in all experiments reported in this section. However, TNT's modular design makes it simple to use different traffic generators, specific applications, or a new tool using model $\mathfrak{M}_A$.

The user data flow in the set of experiments discussed in this paper was generated with an instance of our model $\mathfrak{M}_A(\eta, \theta, \omega, \mu)$. Meaning that TNT configured D-ITG to send a total of $\eta =$ 2000 UDP (User Datagram Protocol) packets



(a) Markov probabilities (b) Markov Raw (c) Markov Filled (d) Markov Shortest Filled

FIGURE 5. From Markov chain to mobility pattern.

of $\omega = 1264$ Bytes (radio's MTU) each. Moreover, each message has a set of packets distributed following a uniform distribution and the delay between messages is $\theta = 1$ s. The distribution of packets per messages is defined as $\eta_u = (a, b)$ with the interval $a = 1$ and $b = 50$ packets, where given the number of packet x and $a \leq x \leq b$ the probability density function $PDF(x) = 0.020$, meaning equal probability to have different number of packet (between 1 to 50) sent per message per second. Notice that we defined η and ω in order to ensure that the data will be sent during the whole mobility pattern, which takes about 170 min. Besides, the goal is to demonstrate TNT's functionalities, testing the queuing discipline (part of a multi-layer military system) over a load of packets that surely demands a store-and-forward mechanism to successfully arrive at the receiver.

C. EXPERIMENTS OVER MODEL $\mathfrak{M}_{B_1}$

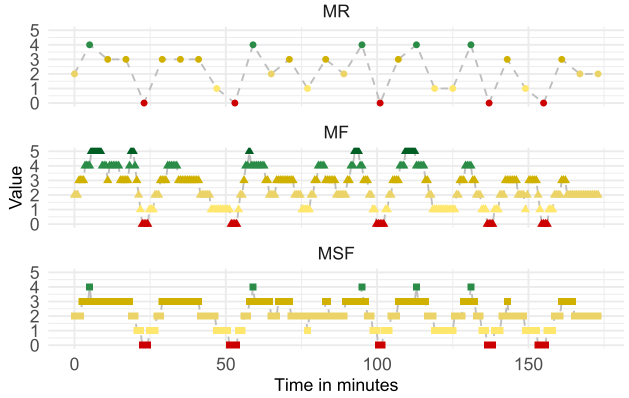
The model $\mathfrak{M}_{B_1}$ creates mobility patterns from a sequence of link data rates supported by our VHF radios. These data rates represent the boundaries of communication radius, meaning that by changing the modulation we are implicitly moving the node in space and time. In order to create the ever-changing network conditions, model $\mathfrak{M}_{B_1}$ is instantiated with the probability distribution defined by matrix (8) and illustrated in Figure 5a. The matrix configures a Markov chain with low probabilities at states {0, 4, and 5} (0.15, 0.1, and 0.1) and high probabilities at states {1, 2, and 3} (0.2, 0.2, and 0.25), where states{0, 1, 2, 3, 4, 5} correspond to {disconnection}, {0.6, 1.2, 2.4, 4.8, 9.6} kbps, respectively. Notice that this particular configuration was arbitrarily chosen to highlight TNT's features. Varying these probabilities will change the pattern of changes as well as the mobility trace.

$$
Markov = \begin{array}{c} \\ \end{array}\begin{matrix} & 0 & 1 & 2 & 3 & 4 & 5 \end{matrix}
$$
$$
Markov = \begin{pmatrix} 0.15 & 0.2 & 0.2 & 0.25 & 0.1 & 0.1 \\ 0.15 & 0.2 & 0.2 & 0.25 & 0.1 & 0.1 \\ 0.15 & 0.2 & 0.2 & 0.25 & 0.1 & 0.1 \\ 0.15 & 0.2 & 0.2 & 0.25 & 0.1 & 0.1 \\ 0.15 & 0.2 & 0.2 & 0.25 & 0.1 & 0.1 \\ 0.15 & 0.2 & 0.2 & 0.25 & 0.1 & 0.1 \end{pmatrix} \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \tag{8}
$$

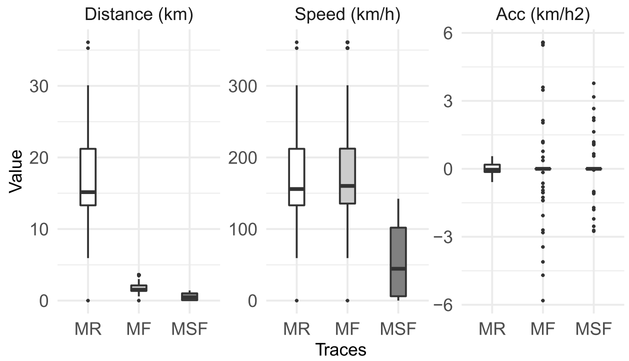
Thus, giving this matrix as an input to the model $\mathfrak{M}_{B_1}$ a sequence of methods are used to create three mobility patterns named Markov Raw (MR), Markov Filled (MF), and Markov Shortest Filled (MSF), as shown in Figure 5. To create the mobility patterns, model $\mathfrak{M}_{B_1}$ receives a sequence of states between s_0 and s_5 generated from the probability matrix (8), in this case study $s = 30$. Moreover, the node movement starts at $s_{st} = 300$ s and the time interval to state change was set to $s_{ti} = 360$ s seconds. These conditions directly impact the node mobility statistics. The MR, Figure 5b, shows how the node is distributed over space with its respective network states, similarly, the patterns MF and MSF are shown in Figure 5c and Figure 5d, respectively. For the last two traces, TNT interpolates the positions with $n_p = 10$, creating smooth movements over time. In addition, for the trace MSF, TNT modify the distributions of states over the communication rings, choosing the closest location based on the last position, then creating short movements.

In order to better understand these mobility patterns, Figure 6 shows the pattern of changes in a time series and the overall statistics of each trace. Taking a look at the changes over time in Figure 6a, we can see the difference among the patterns, even though, they were generated with the same probability matrix and sequence of states. This fact shows that mobility should be considered in the discussion of experimental results enriching the arguments to explain such test scenarios, otherwise the experiments can be interpreted or reproduced in different manners. In the MR trace, we can observe rapid state changes, jumping from one data rate to another. This pattern avoids any transitions between states, making it hard for systems to predict the network condition by monitoring the changes over time and acting to reduce packet loss. The MF trace is different showing smooth transitions between states by interpolating them. Based on smooth movements, tactical systems can sense small and slow network changes, that can be used as input to models designed to predict movements that may increase the probability of packet loss. Finally, the MSF trace combines smooth movements within short distances. Notice that this trace reduces the node transitions because the node position is chosen not considering all possible places inside the ring area but the closest one from its last position.

The difference among these mobility patterns can be observed from the trace statistics, as shown in Figure 6b. This figure plots mobility statistics such as the distance



(a) Link data rates as a function of time.



(b) Traces Statistics

FIGURE 6. **Traces created from the probability matrix in (8).**

between node positions, where the MR shows the largest traveled distances compared to the other patterns, reaching a maximum distance of 36 km and an average of 16.8 km traveled between positions. The speed metric of MR and MF traces show almost the same variation, reaching a max speed up to 360 km/h and averaged speeds about 169 km/h and 174 km/h, respectively. This is due to the interpolation process that divides the distance and time filling the transition states proportionally, resulting in the same node speed. Differently, the trace MSF is a result of the shortest distance approach, showing lower speed in average 58 km/h, compared to the other patterns. The difference can also be noticed in the acceleration metric, where the MR and MF traces have a high average acceleration of 0.01 km/h^2 compared to MSF with 0.001 km/h^2. Table 3 lists the overview of the three patterns of changes presented using the Markov chain. In addition to the mobility metrics, this table also shows the amount of variation in the nominal link data rate that each trace can support. For example, MF supports the highest data rate of 2.5 kbps on average, and MSF has the lowest rate of 1.5 kbps on average. Later, we discuss the impact of these link data rates on experimental results.

The test can be done over different conditions even though it follows the same sequence of states. Therefore, a test scenario characterization using mobility metrics supports the

TABLE 3. **Trace statistics created from a Markov model.**

Traces	Features						
	Data rate - Kbps	Distance - km		Speed - km/h		Acc. - km/h^2	
	(avg ± sd)	Max	Avg	Max	Avg	Max	Avg
MR	1.9 ± 1.59	36	16.8	360	169	0.55	0.01
MF	2.5 ± 2.35	3.6	1.7	360	174	5.5	0.01
MSF	1.5 ± 0.91	1.4	0.57	142	58	3.7	0.001

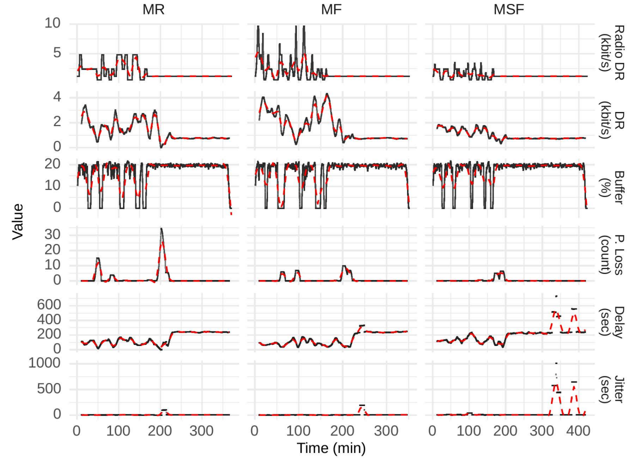


FIGURE 7. **Network statistics over Markov mobility traces.**

comprehension of the experiment's goals and limitations. For example, these patterns can simulate the mobility of different entities in a field, such as vehicles and humans patrolling a given area. The MSF trace could represent the movement of a vehicle with an average speed of 58 km/h in an uneven area, requiring high variation on acceleration. The MR could characterize a non-human movement, e.g. an unmanned aerial vehicle, covering larger distances and reaching high speeds. Notice that these models can generate different test scenarios by modifying a set of parameters that can be reproduced for independent verification.

1) TEST-BED EXPERIMENT

Before starting the experiment, TNT deploys the configuration files for the traffic generator creating the IP data flows, and the instance of model $\mathfrak{M}_{B_{1,2}}$ creating changes in the link data rates in our VHF network with real radios. After the experiment ends, TNT plots six network metrics in Figure 7. The metrics are the radio link data rate (*Radio DR*), the data rate computed at the receiver side (*DR*), the radio buffer (*Buffer*), the packet loss (*P. Loss*), the end-to-end delay (*Delay*), and jitter (*Jitter*). Notice that the *Radio DR* and *Buffer* metrics were acquired from the VHF radio through SNMP and compiled together with the other metrics to show the overall view of the experiment.

These experiments were designed to test the adaptive shaping mechanism, as part of a multi-layer military system described earlier in Section III-D, over different communication scenarios. The *Radio DR* metric shows how the network changes during a 170 min long experiment, which ends when the node stops "moving." However, the data flow goes through the radio link until the packets' queue and the buffer are empty. This behavior is explained by the limited link data rate supported by the network scenarios, but also by the shaping mechanism adding inter-packet delays to avoid buffer overflow. Therefore, each network scenario takes different time intervals to end the experiment. For example, MF takes about 346 min, MR about 365 min, and MSF takes about 417 min. Notice that the link data rate observed at the receiver *DR* is lower than the link data rate reported by the radio *Radio DR* because the radio reports the nominal capacity of its modulations. Thus, MF can support a data rate of 1 kbps on average, while MR supports 0.98 kbps, and MSF 0.8 kbps on average.

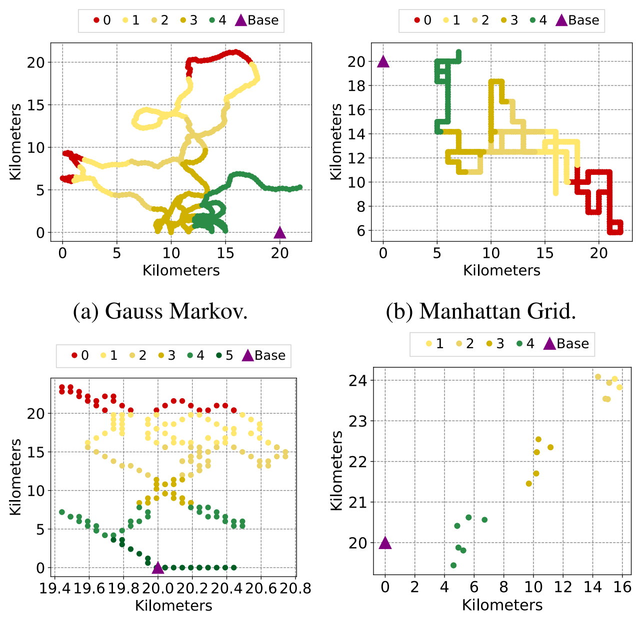
The performance of the adaptive shaping mechanism is better visualized in the *Buffer* metric, where the buffer occupancy is kept very close to the pre-defined threshold of $b = 20$ %, avoiding buffer overflow and reducing as much as possible packet loss. Moreover, the metrics *Delay* and *Jitter* also reflect the behavior of the mechanism, showing delays added between packets and how it varies over time. The minimum delay observed in our VHF network is higher than 5.1 s for all scenarios. The average delay observed in MF was 157 s, for MR about 171 s, and around 221 s in MSF. Moreover, MSF experienced the highest jitter variation of about 62 s, while MR had the lowest value about 11 s. These metrics describe the boundaries of QoS for a given communication scenario, helping to identify which type of application can be used in each scenario.

These experiments show the capabilities of our radios and how much variation in the link data rate (mobility) it can support. The changes on the MR happen every $s_{ti} = 360$ s jumping from one state to another therefore loosing 10 % of the IP packets sent through the radio link. MF and MSF experience lower packet loss, 7.4 % and 6.5 %, due to the smoother state transitions. This is explained by the mechanism implemented in our adaptive shaping, aiming to detect abnormal buffer behavior, such as flushing all data stored in the buffer after a link disconnection. Thus, these experiments show that our VHF radios cannot support too many and long disconnections over time, about 360 s, thus requiring system support, like the adaptive shaping mechanism, to minimize packet loss.

We also noticed that these changes are strongly related to the routing protocol used in the radios to perform neighbor discovery and its configuration parameters, in our case OLSR, which cannot be manipulated without a radio manager system with proper permission. Once the flush happens, resulting in packet loss, the radio also takes time to resume IP data flow, adding more delays and degrading QoS. Therefore, the military system needs to act to minimize the impact of link disconnections and the resulting flush in the radio buffer. The shaping mechanism minimizes this issue by defining two rules (i) a low buffer threshold of $b = 20$ % (could be less), in order to reduce the packet loss in case of flushing; (ii) identifying these flushes on the buffer and prevent more loss reducing the dequeue delivery rate to a minimum as possible.

TABLE 4. Statistics from experiments over Markov traces.

Metrics		Traces MR	MF	MSF
Experiment duration (min)		365.25	346.19	417.25
Data Rate in kbps (avg ± sd)		0.98 ± 0.72	1 ± 0.95	0.8 ± 0.7
Packets	Sent	2000	2000	2000
	Recv.	1796	1851	1869
	Drop.	10.2%	7.4%	6.5%
	Rate (avg/s)	0.08	0.09	0.07
Delay (sec)	Min	5.8	6.2	5.1
	Max	2899.7	5878.13	19519.66
	Avg ± sd	171 ± 101	157 ± 157	221 ± 673
Jitter (sec)	Avg ± sd	11.2 ± 96.3	13.4 ± 185	62.7 ± 943



(a) Gauss Markov. (b) Manhattan Grid.

(c) Probabilistic Random Walk. (d) Random Waypoint.

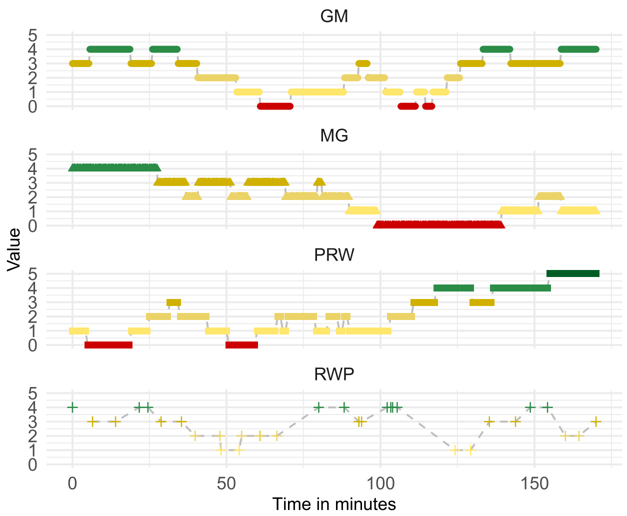
FIGURE 8. Mobility pattern from mobility models.

Table 4 summarizes the experimental results discussed in this section. Thus, after the experiment ends TNT exports a summary as shown in this table in addition to the result visualized in a time series.

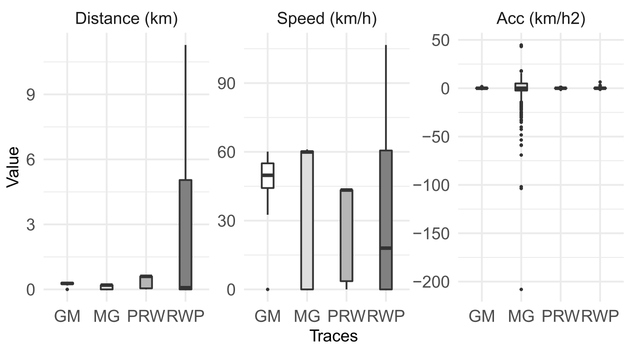
D. EXPERIMENTS OVER MODEL $\mathfrak{M}_{B_2}$

This section describes the experiments using model $\mathfrak{M}_{B_2}$, which reuses well-known mobility models to create patterns of link data rate change, as described earlier in Section III-C2. TNT was set to use BonnMotion to generate four exemplary mobility patterns, from four different models, namely GMs, RWPs, MGs, and PRWs. Figure 8, shows how the node movement is distributed over space and its respective network states. Notice that model $\mathfrak{M}_{B_2}$ is independent of the mobility model generator BonnMotion and it supports any trace file with a format $node_i = [x, y, time]$.

Next, we briefly describe each model and the set of parameters used to create a given mobility pattern. The goal is to



(a) Link data rates as a function of time.



(b) Traces Statistics.

FIGURE 9. Traces created from mobility models.

understand the mobility models and the trace characteristics generated by them. The GM model [42] was designed to generate node movement based on its current location, speed, and direction. At given time intervals, these parameters are updated creating the node movement as a function of time. For example, Figure 8a shows a mobility pattern created by an instance of GM to simulate node movement through uneven terrain. This trace was generated using the following parameters: frequency to update the movement $f_u = 20$ s, angle deviation $a_{std} = 0.39$, speed deviation $speed_{std} = 0.5$ m/s, and maximum speed $max_s = 16.7$ m/s (60 km/h). MG model [43] uses a grid of roads to simulate modern urban areas, as shown in Figure 8b. This model is mostly used for tests in vehicular networks, where mobile nodes move on streets of a city, following a given probability to turn or keep straight on at each intersection. We configured the model with the following probabilities: the mobile node goes straight with a probability of $f = 0.5$ and turn left with $t_l = 0.25$ or turn right with $t_r = 0.25$. We also defined the minimum speed of $min_s = 0.1$ m/s (0.36 km/h), the maximum averaged speed about $max_{as} = 16.7$ m/s, a pause probability of $p_p = 0.1$ (if there is no speed changes) and a pause of about $p_t = 30$ s.

PRW [37] chooses new directions and speeds for the node to follow based on pre-defined probabilities. For example, Figure 8c shows the output of an instance of PRW using a Markov probability matrix to define probabilities distribution to move in any direction at a fixed speed or to remain in the same direction. This model also creates erratic movement with a given probability, simulating the unpredictable movement of many entities in nature. Lastly, we used an instance of model RWP [44] to generate the mobility trace shown in Figure 8d. In this figure, the node starts without a change in direction for a certain period of time $p_t = 600$ s, then it chooses at random the next destination. The node speed follows an uniform distribution and we defined the minimum speed around $min_s = 10$ m/s (36 km/h) and the maximum speed of $max_s = 30$ m/s (108 km/h). Once the node reaches the pre-defined checkpoints, such as cp_1(5 km, 20 km), cp_2(10 km, 22 km), and cp_3(15 km, 24 km) it pauses for p_t seconds before starting the process again.

1) TRACE ANALYSIS

Complementing Figure 8, Figure 9a, shows the distribution of link data rates over time for the four mobility models. These patterns can represent different entities moving in a battlefield/scenario as described before, enriching the discussion and explanation of given system behavior based on a network scenario (node mobility). The difference among these patterns is also observed in the trace statistics shown in Figure 9b. Comparing the distance, speed, and acceleration features, it is noticed that the GM trace shows the most stable movements compared to the other patterns, with small variation around the mobility metrics with an average speed of about 47 km/h, acceleration about 0.005 km/h^2 averaged and distance traveled between positions about 0.27 km in average.

The node at MG can reach 40 km/h and traveled distance about 0.12 km averaged with high averaged deceleration about 0.9 km/h^2, simulating a vehicle behavior in an urban area. The PRW trace shows long traveled distances about 0.44 km, compared with the last traces, at low speed about 31 km/h with no much variation in acceleration about 0.0003 km/h^2, all in average. For example, these attributes can be used to describe an unmanned patrol in a plateau terrain. Finally, the RWP trace introduces more variability in the traveled distances about 2.59 km averaged and reaching maximum speed of 108 km/h. Moreover, this trace has specific checkpoints (areas) where it stays for a while before starts moving again, which can be characterized as vehicular movements, based on the speed and distances the entity can reach. Table 5 shows the numeric overview regarding the mobility traces MG, PRW, GM and RWP discussed in this section.

2) TESTS WITH THE ADAPTIVE SHAPING MECHANISM

Following the same methodology described in the experimental results with model $\mathfrak{M}_{B_1}$. Here, the adaptive shaping mechanism is tested over the network conditions provided

TABLE 5. Trace statistics created from mobility models.

Traces	Features							
	Data rate - Kbps	Distance - km		Speed - km/h		Acc. - km/h^2		
	(avg ± sd)	Max	Avg	Max	Avg	Max	Avg	
GM	2.1 ± 1.7	0.33	0.27	60	47	1.7	0.005	
MG	1.6 ± 1.6	0.2	0.12	61	40	44	-0.9	
PRW	2.4 ± 2.7	0.6	0.44	43	31	0.8	0.0003	
RWP	2.7 ± 1.6	11.2	2.59	106	30	6.4	0.31	

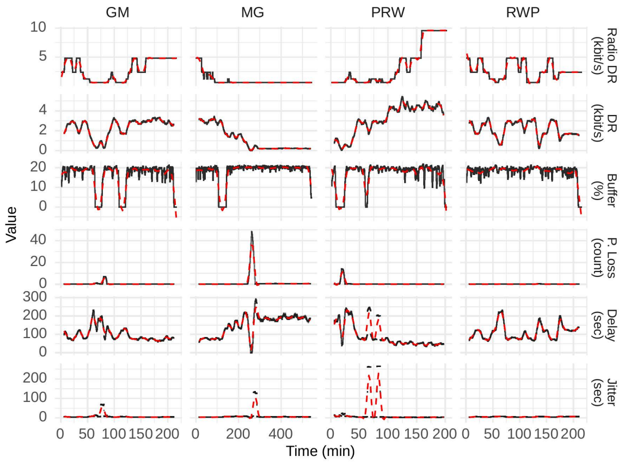


FIGURE 10. Network statistics over mobility traces.

by the model $\mathfrak{M}_{B_2}$, as illustrated in Figure 10 by different network metrics. The first observation is the experiment duration, where MG took about 536 min (more then 8 hours) to end the experiment, the GM and glsrwp took about 210 min, and PRW just 197 min. The difference in time duration of each experiment is explained by the average link data rate and the duration of link disconnections generated by the models when their mobility traces are used to set the radio modulation in our test-bed. It is also visible that our shaping mechanism can hold the user data flow in order to keep the radio buffer occupancy (*Buffer*) close to the pre-defined threshold of $b = 20\,\%$. Moreover, in all mobility patterns, except RWP, an abnormal radio behavior flushing its buffer is observed, converging to the same explanation reported earlier in Section IV-C regarding the limitations of our VHF radios. Even though this behavior could negatively impact the packet delivery, our shaping mechanism was able to sense these abrupt changes in the buffer occupancy to adapt the dequeue rate to the minimum as possible, reducing packet loss while the route is not recovered. Therefore, the minimum packet loss (*P. Loss*) observed in MG reached about 24 %, PRW with 4.8 % and GM reaching about 3.7 % of packet loss. Although RWP does not contain disconnections, we observed packet loss of about 1.6 %. The low packet loss happens because the probability of losing packets increases as the link data rate decreases, as calculated in [38] using experimental results with the same VHF radios.

Regarding the end-to-end delay (*Delay*), the minimum delay observed in our VHF network was 4.17 s. On average, PRW experienced a minimum delay of about 95 s, compared

TABLE 6. Statistics from experiments over mobility traces.

Metrics		Traces			
		GM	MG	PRW	RWP
Experiment duration (min)		210.86	536.59	197.26	210.061
Data Rate in kbps (avg ± sd)		0.97 ± 0.75	0.79 ± 1.1	0.87 ± 0.96	2.1 ± 1.5
Packets	Sent	2000	2000	2000	2000
	Recv.	1925	1513	1904	1968
	Drop. %	3.75%	24.35%	4.8%	1.6%
	Rate (avg/s)	0.15	0.04	0.16	0.15
Delay (sec)	Min	10.04	5.29	4.17	14.24
	Max	2023.4	3889.2	7900.5	344.02
	Avg ± sd	101 ± 67	199 ± 144	95 ± 260	112 ± 54
Jitter (sec)	Avg ± sd	7.4 ± 59	12 ± 131	21 ± 355	5.7 ± 5.9

to GM with 101 s, RWP 112 s, and MG reaching 199 s. The *Jitter* metric of PRW reached the highest variation, about 21 s, followed by MG with 12 s, GM with 7.4 s and RWP reaching about 5.7 s, all on average. Table 6 summarizes the quantitative results discussed in this section.

V. CONCLUSION

This paper introduced and evaluated the TNT platform designed to test military systems over ever-changing communication scenarios. TNT generates scenarios with different patterns of change for network conditions and user data flow, also collecting and analyzing data from experiments, and showing quantitative performance metrics. TNT wraps the necessary models to automate the evaluation of military systems and applications over real military radios. Therefore, TNT supports the creation of a variety of data flows by the definition of a general model ($\mathfrak{M}_A$) instantiated with a traffic generator tool, models ($\mathfrak{M}_{B_1}$ and $\mathfrak{M}_{B_2}$) for changing network conditions through stochastic models, transforming the sequence of network states (link data rates) in mobility patterns and mapping mobility traces to link data rates. As a result, mobility traces can be used to change the radio link data rates in stationary radios in a laboratory. TNT also includes a prototype to create link disconnections in radios with wired antennas.

Addressing the problem of frequent changes in the link quality that may lead to a buffer overflow in military radios, TNT introduced an adaptive mechanism to shape the user data flow to the network conditions. Lastly, TNT's monitoring and data analysis scripts supported the discussion of experimental results over seven different scenarios, including plots and statistics, characterizing both the military system's performance using network metrics and the changes in the communication scenario using mobility metrics.

We plan to extend TNT to support emulated network environments with SDN-capable devices and military communication technologies, such as VHF, UHF, and SatCom. The goal is to increase the scale of the test environment and to improve military systems before deploying them in a close to a real network environment. Moreover, we plan to enhance the network-changing model $\mathfrak{M}_{B_1}$ by introducing physical interference that simulates barriers along the mobility traces and extending the model to support mobility metrics as input, such as speed, distance, and non-linear movements.

REFERENCES

[1] G. Elmasry, "A comparative review of commercial vs. Tactical wireless networks," *IEEE Commun. Mag.*, vol. 48, no. 10, pp. 54–59, Oct. 2010.

[2] R. Amin, D. Ripplinger, D. Mehta, and B.-N. Cheng, "Design considerations in applying disruption tolerant networking to tactical edge networks," *IEEE Commun. Mag.*, vol. 53, no. 10, pp. 32–38, Oct. 2015.

[3] R. R. F. Lopes, P. H. Rettore, S. M. Eswarappa, J. Loevenich, and P. Sevenich, "Performance analysis of proactive neighbor discovery in a heterogeneous tactical network," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, May 2021, pp. 1–8.

[4] L. Prior, C. Fossa, D. Ward, J. Sun, P. Boehm, E. Kuczynski, J. Cain, and T. Mak, "Tactical network integration test framework," in *Proc. Mil. Commun. Conf. (MILCOM)*, Nov. 2011, pp. 2158–2163.

[5] K. M. Marcus, K. S. Chan, R. L. Hardy, and P. L. Yu, "An environment for tactical SDN experimentation," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2018, pp. 1–9.

[6] P. H. L. Rettore, M. Von Rechenberg, J. F. Loevenich, R. R. F. Lopes, and P. Sevenich, "A handover mechanism for centralized/decentralized networks over disruptive scenarios," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2021, pp. 836–842.

[7] A. Ghosh, S.-W. Li, C. J. Chiang, R. Chadha, K. Moeltner, S. Ali, Y. Kumar, and R. Bauer, "QoS-aware adaptive middleware (QAM) for tactical MANET applications," in *Proc. Mil. Commun. Conf. (MILCOM)*, Oct. 2010, pp. 178–183.

[8] A. Poylisher, F. Sultan, A. Ghosh, S.-W. Li, C. J. Chiang, R. Chadha, K. Moeltner, and K. Jakubowski, "QAM: A comprehensive QoS-aware middleware suite for tactical communications," in *Proc. Mil. Commun. Conf. (MILCOM)*, Nov. 2011, pp. 1586–1591.

[9] A. S. Peng, D. M. Moen, T. He, and D. J. Lilja, "Automatic dynamic resource management architecture in tactical network environments," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2009, pp. 1–7.

[10] T. Chen, S. Eswaran, M. A. Kaplan, S. Samtani, D. Shur, J. Sucec, and L. Wong, "Enhancing application performance with network awareness in tactical networks," in *Proc. Mil. Commun. Conf. (MILCOM)*, Nov. 2011, pp. 1158–1163.

[11] R. Fronteddu, A. Morelli, M. Tortonesi, N. Suri, C. Stefanelli, R. Lenzi, and E. Casini, "DDAM: Dynamic network condition detection and communication adaptation in tactical edge networks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2016, pp. 970–975.

[12] R. Fronteddu, A. Morelli, E. Casini, N. Suri, B. Jalaian, and L. Sadler, "A content and context-aware solution for network state exchange in tactical networks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2017, pp. 430–435.

[13] F. T. Johnsen, T. H. Bloebaum, J. M. A. Calero, Q. Wang, J. Nightingale, M. Manso, and N. Jansen, "WS-notification case study and experiment," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, May 2017, pp. 1–8.

[14] J. J. Lindquister, F. T. Johnsen, and T. H. Bloebaum, "Proxy pair optimizations for increased service reliability in DIL networks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2017, pp. 870–875.

[15] M. Małowidzki, P. Kaniewski, R. Matyszkiel, and P. Bereziński, "Standard tactical services in a military disruption-tolerant network: Field tests," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2017, pp. 63–68.

[16] R. R. F. Lopes, A. Viidanoja, M. Lhotellier, A. Diefenbach, N. Jansen, and T. Ginzler, "A queuing mechanism for delivering QoS-constrained web services in tactical networks," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, May 2018, pp. 1–8.

[17] S. M. Eswarappa, P. H. L. Rettore, J. Loevenich, P. Sevenich, and R. R. F. Lopes, "Towards adaptive QoS in SDN-enabled heterogeneous tactical networks," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, May 2021, pp. 1–8.

[18] J. F. Loevenich, A. Sergeev, P. H. L. Rettore, and R. R. F. Lopes, "An intelligent model to quantify the robustness of tactical systems to unplanned link disconnections," in *Proc. 18th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Mar. 2022, pp. 1–8.

[19] N. Suri, A. Hansson, J. Nilsson, P. Lubkowski, K. Marcus, M. Hauge, K. Lee, B. Buchin, L. Misirhoglu, and M. Peuhkuri, "A realistic military scenario and emulation environment for experimenting with tactical communications and heterogeneous networks," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, May 2016, pp. 1–8.

[20] M. Von Rechenberg, P. H. L. Rettore, R. R. F. Lopes, and P. Sevenich, "Software-defined networking applied in tactical networks: Problems, solutions and open issues," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, May 2021, pp. 1–8.

[21] Q. Zhao, A. J. Brown, J. H. Kim, and M. Gerla, "An integrated software-defined battlefield network testbed for tactical scenario emulation," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Norfolk, VA, USA, Nov. 2019, pp. 373–378.

[22] EXata. *Exata Network Emulation Software*. Accessed: Feb. 10, 2022. [Online]. Available: https://www.scalable-networks.com/products/exata-network-emulator-software

[23] iTrinegy. *Software Defined Test Networks*. Accessed: Feb. 10, 2022. [Online]. Available: https://itrinegy.com/solution-by-industry/military/

[24] A. Poylisher, C. Serban, J. Lee, T. Lu, R. Chadha, C.-Y.-J. Chiang, K. Jakubowski, and R. Orlando, "Virtual ad hoc network testbeds for high fidelity testing of tactical network applications," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2009, pp. 1–7.

[25] S. Doshi, U. Lee, B. Bressler, R. Bagrodia, M. DiGennaro, J. Oleksa, and Y. Chen, "Operationally realistic testing of network centric tactical applications in a lab environment," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2012, pp. 1–6.

[26] N. Jansen, D. Kramer, and M. Spielmann, "Testbeds for IT systems in tactical environments," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2014, pp. 1293–1298.

[27] C. Serban, A. Poylisher, A. Sapello, Y. Gottlieb, C. J. Chiang, and R. Chadha, "Testing Android devices for tactical networks: A hybrid emulation testbed approach," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2015, pp. 489–494.

[28] R. R. F. Lopes, P. H. Balaraju, P. H. L. Rettore, and P. Sevenich, "Queuing over ever-changing communication scenarios in tactical networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 291–305, Jan. 2022.

[29] R. R. F. Lopes, P. H. Balaraju, A. T. Silva, P. H. Rettore, and P. Sevenich, "Experiments with a queuing mechanism over ever-changing data rates in a VHF network," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Norfolk, VA, USA, Nov. 2019, pp. 131–136.

[30] P. H. Balaraju, P. H. L. Rettore, R. R. F. Lopes, S. M. Eswarappa, and J. Loevenich, "Dynamic adaptation of the user data flow to the changing data rates in VHF networks: An exploratory study," in *Proc. 11th Int. Conf. Netw. Future (NoF)*, Oct. 2020, pp. 64–72.

[31] R. R. F. Lopes, J. Loevenich, P. H. L. Rettore, S. M. Eswarappa, and P. Sevenich, "Quantizing radio link data rates to create ever-changing network conditions in tactical networks," *IEEE Access*, vol. 8, pp. 188015–188035, 2020.

[32] R. R. F. Lopes, P. H. Balaraju, and P. Sevenich, "Creating ever-changing QoS-constrained dataflows in tactical networks: An exploratory study," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, Budva, Montenegro, May 2019, pp. 1–8.

[33] R. R. F. Lopes, P. H. Balaraju, and P. Sevenich, "Creating and handling ever-changing communication scenarios in tactical networks," in *Proc. 15th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Coimbra, Portugal, Mar. 2019, pp. 67–74.

[34] A. Bujari, C. T. Calafate, J.-C. Cano, P. Manzoni, C. E. Palazzi, and D. Ronzani, "Flying ad-hoc network application scenarios and mobility models," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 10, pp. 155–172, 2017.

[35] S. Kumar, S. Sharma, and B. Suman, "Mobility metrics based classification & analysis of mobility model for tactical network," *Int. J. Next-Gener. Netw.*, vol. 2, no. 3, pp. 2565–2573, 2010.

[36] S. M. Mousavi, H. R. Rabiee, M. Moshref, and A. Dabirmoghaddam, "MobiSim: A framework for simulation of mobility models in mobile ad-hoc networks," in *Proc. 3rd IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2007, p. 82.

[37] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "BonnMotion: A mobility scenario generation and analysis tool," in *Proc. 3rd Int. ICST Conf. Simul. Tools Techn.*, 2010, pp. 1–10, doi: 10.4108/ICST.SIMUTOOLS2010.8684.

[38] J. F. Loevenich, R. R. F. Lopes, P. H. Rettore, S. M. Eswarappa, and P. Sevenich, "Maximizing the probability of message delivery over ever-changing communication scenarios in tactical networks," *IEEE Netw. Lett.*, vol. 3, no. 2, pp. 84–88, Jun. 2021.

[39] Naval Research Laboratory and PEANR Group. *MGEN user's and Reference Guide Version 5.0*. Accessed: Feb. 10, 2022. [Online]. Available: https://github.com/USNavalResearchLaboratory/mgen/blob/master/doc/mgen.pdf

[40] A. Botta, A. Dainotti, and A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios," *Comput. Netw.*, vol. 56, no. 15, pp. 3531–3547, Oct. 2012.
[41] A. Botta, A. Dainotti, and A. Pescape, "Do you trust your software-based traffic generator?" *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 158–165, Sep. 2010.
[42] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for PCS networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun. 18th Annu. Joint Conf. IEEE Comput. Commun. Societies*, vol. 3, Mar. 1999, pp. 1377–1384.
[43] ETSI, SMG, *Universal Mobile Telecommunications System (UMTS); Selection Procedures for the Choice of Radio Transmission Technologies of the UMTS*, document TR 101 112, V3.2.0, 1997.
[44] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*. Boston, MA, USA: Springer, 1996, pp. 153–181.

PAULO H. L. RETTORE received the B.Sc. and M.Sc. degrees in computer science, in 2009 and 2012, respectively, and the Ph.D. degree in computer science from the Federal University of Minas Gerais (UFMG), in 2019. He is currently a Scientist at Fraunhofer FKIE, Bonn, Germany. He is with the Communication Systems Department (KOM), he has been focused on measuring the performance bounds of tactical systems over ever-changing scenarios. His research interests include computer networks, mobile *ad-hoc* networks, tactical networks, software-defined networking, ubiquitous computing, the Internet of Things, intelligent transportation systems, and smart mobility.

JOHANNES LOEVENICH received the bachelor's degree in computer science from Rheinische Friedrich-Wilhelms-Universität Bonn, in 2017, and the bachelor's degree in mathematics from Rheinische Friedrich-Wilhelms-Universität Bonn, in 2018. He is currently pursuing the master's degree in computer science with the University of Bonn, Germany. He is also working as a Scientist with the Communication Systems Department (KOM), Fraunhofer FKIE, Bonn, Germany. His research interests include computer systems, distributed systems, tactical networks, data science, and game theory.

ROBERTO RIGOLIN F. LOPES (Member, IEEE) received the B.Sc. degree in computer science from UFMT, Brazil, the M.Sc. degree in computer science from UFSCar, Brazil, and the Ph.D. degree in computer science from USP, Brazil. He is currently a Scientist at Thales Deutschland, Ditzingen, Germany. He is with the Secure Communications & Information Systems (SIX), he has been attacking problems in computer networks and distributed systems with a particular interest in the performance bounds of tactical systems over ever-changing communication scenarios. He also visited universities at UTwente, The Netherlands, and NTNU, Norway. He worked as a Scientist at Fraunhofer FKIE, Germany. He has been rebuilding his own education following curiosity freely by reading books on physics, mathematics, and philosophy.