Queuing over Ever-changing Communication Scenarios in Tactical Networks

Roberto Rigolin F. Lopes, Pooja Hanavadi Balaraju, Paulo H. L. Retore, and Peter Sevenich

Abstract—This paper introduces a hierarchy of queues complementing each other to handle ever-changing communication scenarios in tactical networks. The first queue stores the QoS-constrained messages from command and control systems. These messages are fragmented into IP packets, which are stored in a queue of packets (second) to be sent to the radio buffer (third), which is a queue with limited space therefore, open to overflow. We start with the hypothesis that these three queues can handle ever-changing user(s) data flows (problem \( A \)) through ever-changing network conditions (problem \( B \)) using cross-layer information exchange, such as buffer occupancy, data rate, queue size and latency (problem \( A|B \)). We introduce two stochastic models to create sequences of QoS-constrained messages (\( A \)) and to create ever-changing network conditions (\( B \)). In sequence, we sketch a control loop to shape \( A \) to \( B \) to test our hypothesis using model \( A|B \), which defines enforcement points at the incoming/outgoing chains of the system together with a control plane. Then, we discuss experimental results in a network with VHF radios using data flows that overflows the radio buffer over ever-changing data rate patterns. We discuss quantitative results showing the performance and limitations of our solutions for problems \( A, B \) and \( A|B \).

Index Terms—ever-changing communication scenarios, tactical networks, hierarchical queuing, robust control loop

1 INTRODUCTION

Tactical networks are heterogeneous radio networks used by the military to connect mobile nodes in the battlefield [1]. These nodes are supporting users collaborating to accomplish a mission using a set of Command and Control (C2) services, such as friendly/hostile force tracking, medical evacuation, chat and so on. Given the wide range of military missions (e.g. combat, reconnaissance and disaster response), the users will combine C2 services in different ways to achieve their objectives. Remember that these services may have different Quality of Service (QoS) requirements like priority, reliability and time of expire. The challenge is to meet the QoS requirements over radio links exposed to several sources of randomness.

For example, the source of randomness in tactical networks are the node/network mobility, physical obstacles on the way, physical attacks from hostile forces, the presence of an active adversary jamming the spectrum and so on. The combination of these random effects will influence the patterns of change in both user-generated data flows and network conditions observed during simulations or field experiments. Therefore, the distributions of change are unknown until the experiment is executed. We are using stochastic models to define the distributions of change before the experiment is executed.

This investigation assumes that the communication scenarios in tactical networks are composed by at least two parts changing independently, the user data flows and the network conditions, defined here as problem \( A \) and \( B \), respectively. Given the number of combinations between possible states of \( A \) and \( B \), we call it ever-changing communication scenarios [2], [3], [4]. Thus, the challenge is to design tactical systems that can deliver the user data flows, given certain network conditions. We call it problem \( A|B \) reusing the conditional probability notation [2].

These three problems are partially addressed in the literature reporting simulations and field experiments in tactical networks (e.g. [5], [6], [7], [8], [9], [10]). For example, it is common to find studies using non-stochastic user-generated data flows with few types of messages sent to the network within an uniform time window (problem \( A \)). Thus, we defined the stochastic model \( A \) to combine different types of messages creating sequences of messages with a known distribution of change that can be reproduced for quantitative verification or quantitative comparison. Moreover, the changes in the network conditions are defined by choosing the mobility model, communication technology, propagation model and so on. This is an indirect way of defining the degree of change of the network conditions, which is difficult to reproduce for quantitative comparisons (problem \( B \)). We introduced a direct approach using model \( B \) to create sequences of network states to configure the radios during run time. Our model uses conditional probabilities to define in precise terms the distribution of change to study the performance bounds of tactical systems.

We developed these two models to test our solution for problem \( A|B \), which is composed of a hierarchy of three queues (here called model \( A|B \)). Model \( A \) defines the user services as states of a Markov chain to generate sequences of QoS-constrained messages. Model \( B \) uses the radio data rates as states of a Markov chain to create sequences of network states. Theses two models, changing independently, compose the ever-changing communication scenarios to test
our model \( A|B \). The Markov assumption defining that the next state depends only on the current state, holds true because we are interested in testing the upper performance bounds of tactical system, implementing model \( A|B \), by combining all the possible states randomly.

The goal is to construct mathematical arguments explaining why systems, also called middlewares, brokers or proxies (e.g. [6], [7], [11], [12], [13], [14], [15], [16], [17], [18]), can handle changes in both \( A \) and \( B \) relying on cross-layer information exchange with the radios and routers (e.g. buffer occupancy, link data rate, latency, packet loss and so on). In our previous investigations, [2], [4] and [19], we discussed simulated and experimental results testing different ways of computing the Inter-Packet Interval (IPI) using cross-layer information to adapt the queuing mechanism to the current network conditions. Here, we study the performance of a hybrid solution (reactive/proactive) using Very High Frequency (VHF) radios by challenging our mechanism with ever-changing communication scenarios including network disconnections.

The contributions of this paper are the following.

- Definition of two stochastic models to create ever-changing communication scenarios: model \( A \) creates reproducible QoS-constrained data flows and model \( B \) creates reproducible data rate patterns implemented to control a VHF radio that supports five data rates along with link disconnection implemented with a relay;
- Introduction of model \( A|B \) defining enforcement and control points hosting our hierarchy of three queues that relies on both radio buffer occupancy (reactive) and link data rate (proactive) to dynamically control the user-generated data flows exceeding the available link data rate;
- Discussion over quantitative results from experiments testing our queuing mechanism over ever-changing communication scenarios in a VHF network; i.e. verifying our three models \( A, B \) and \( A|B \).

The remainder of this paper is structured as follows. Section 2 discusses the fundamental concepts supporting the present study together with the related investigations. Sections 3, 4 and 5 define the three problems and sketch three models to solve them (\( A, B \) and \( A|B \), respectively). Section 6 discusses experimental results in a VHF network with data rates changing in six different patterns and user data flows changing in three different patterns. Section 7 discusses the challenges and limitations of our three models and experimental results. Finally, Section 8 concludes the paper also listing future works. Table 1 defines the mathematical notation used in this paper.

## 2 Background and Related Works

### 2.1 Tactical networks

Tactical network is a Mobile Adhoc Network (MANET) used by the military to connect nodes in hostile environments like battlefields, potentially facing an adversary in both physical and cyber domains. Node mobility, environmental interference and jamming are examples of source of non-determinism (randomness) changing network metrics, such as link data rate, latency and packet loss. Thus, the tactical systems must dynamically adapt its behavior to the current network conditions, which may include network disruptions. Intermittent network conditions demands multi-layer, distributed and self-configurable tactical systems not relying on centralized control.

For example, Fig. 1 shows the structure of a tactical network where each VHF radio (up to 9.6 kbps, ∼20 km) is a gateway for a Ultra High Frequency (UHF) network (up to 240 kbps, ∼2 km) [20]. These networks connect together the head quarters (deployed node represented by a square), mobile vehicles (triangles) and dismounted nodes (circles). The dismounted nodes are soldiers, robots or sensors that may use a vehicle to move in the target area. When these nodes are not inside the vehicle, they are dismounted. In this investigation, we performed experiments in a subset of this tactical network with a VHF constellation composed by three radios, studying the performance of the sender (1) and receiver (2) also highlighted in the figure. Regarding scalability, our VHF network can scale up to 32 radios using the radios in our test setup, described later in Section 6.1. In such an extreme scenario, the probability of the user-generated data flows exceeding the ever-changing network capacity grows as a function of the number of nodes using the network at the same time.

### 2.2 Taxonomy of services and hierarchy of queues

Let us use the services taxonomy in Fig. 2 as a guideline to define the fundamental concepts related to the three problems being addressed in the present investigation and within the related literature. In this figure, the main functional blocks from NATO’s Consultation, Command and Control (C3) taxonomy [21] are placed side-by-side with a flow diagram listing the core services implementing our hierarchy of queues. There are annotations using letters for user(s) data flows (A) and for the tactical network (B), and the core services implementing our hierarchy of queues are numbered from (0) to (4), summing up seven components defined as follows:

(A) **User Applications:** is a set of QoS-constrained C2 services available to the users in tactical networks (e.g. the services in [22] and in Table 2). **Problem A**
defines the challenge of creating sequence of QoS-constrained messages that can be reproduced for quantitative comparisons (details in Section 3); (4) Management, Control and Security (vertical): defines the means to share cross-layer information among the different services to compile their current status, to enforce security and to provide feedback to control loops. We sketch control loops in Section 5 to handle ever-changing communication scenarios; (3) Community of Interest (COI) Services: hosts services to deal with particularities from the user applications (A) in a collaborative environment, such as coalition among different nations; (2) Core Enterprise Services: combine services for infrastructure, Service Oriented Architecture (SOA) and enterprise support, which are usually implemented within tactical systems. These systems are the main subject of literature review in the next section (Section 2.3); (1) Communication Services: combine services for transmission, transport and network access, which are partially implemented by software-defined radios together with tactical routers managing the heterogeneous network; (0) Radio(s): interface with the military radios composing the tactical network. This interface is usually implemented with standardized protocols like Simple Network Management Protocol (SNMP) and Dynamic Link Exchange Protocol (DLEP) [23] to control/share network metrics like link data rate, radio buffer occupancy, latency, packet loss and so on; (B) Tactical Network: heterogeneous radio networks with intermittent conditions and deployed in hostile environments also, potentially facing an adversary in both physical and digital domains. Problem B defines the challenge of creating reproducible patterns of network change (details in Section 4).

Given the network conditions B, the multi-layered tactical system (0-4) have to dynamically adapt the user(s) data flows A; here called Problem $A/B$ in reference to conditional probability notation. We start with the hypothesis that a hierarchy of three queues can complement each other exchanging cross-layer information while shaping the user(s) data flow to the current network conditions, therefore, solving Problem $A/B$. Our solution starts at the proxy/broker (3) intercepting and sending the user messages to the message queue (2), followed by a queue of IP packets implemented by the packet handler and Transport (1), and ends with radio buffer (0). Also, the cross-layer contextual monitoring (4) allows our hierarchy of queues to complement each other through the three control points annotated with roman letters in Fig. 2 and defined as follows:

(i) How long to admission? It is a function taking as input the size of the queue of packets (1), the buffer occupancy (0) and the current link data rate (0) to compute the waiting time at the queue of messages (2) before sending the next message to the queue of packets (0);

(ii) Is the radio buffer below $b\%$? The packet handler (1) continuously monitors the radio buffer occupancy checking if it is above a pre-defined $b\%$ threshold (reactive). If so, it pauses the flow of packets until the buffer occupancy is again lower than the threshold. For example, message-based data flows with the highest priority will be shaped as a function of the current network conditions. If the data flow exceeds the current network capacity, packets can miss their deadlines. However, the system is doing its best to deliver higher-priority packets even during extreme conditions like congestion in the network link;

(iii) Inter-packet interval (IP): It is the output of a function taking as input the link data rate (0), the current network usage (0) and the size of the IP packet (1) to compute the interval between packets to proactively shape the user data flow to the current network conditions.

These three control points $i$, $ii$ and $iii$ are defined in precise terms later in Section 5.2 introducing model $A/B$.

2.3 Related works

2.3.1 Tactical systems

Recent investigations are adapting technologies developed for the Internet, such as web services and transport protocols, to create reliable and secure message-based data flows in tactical networks. Back in 2007, the authors in [24] stated that Internet technologies lack the adaptive network management for QoS-constrained services competing for limited radio links and using large messages, such as Simple Object Access Protocol (SOAP) messages. More than a decade later, the recent literature are proposing proxies, brokers and middlewares, here called tactical systems, to adapt services data flows to tactical networks. But there is limited quantitative evidence suggesting that the proposed tactical systems can deliver QoS-constrained data flows (e.g. from command and control services) through ever-changing communication scenarios in tactical networks.

In [13], the authors introduce a mechanism to dynamically detect and adapt to the changing network conditions...
in tactical networks. Their non-intrusive solution identifies the communication technology providing the link but there is limited discussion about how to adapt user data flows to the network conditions. Similarly, the study in [6] discusses experiments with a broker implementing the publish/subscribe message exchange pattern using web services. The experiments were done in a network connecting real radios and emulated links. The authors compiled quantitative results exchanging Friendly Force Tracking (FFT) messages between two convoys of vehicles with four nodes each (non-stochastic data flows). The results were discussed with focus on packet loss, data rate and delay over limited network conditions with non-stochastic changes in the network conditions.

The investigation in [7] reports a quantitative evaluation of a proxy implementing three different application protocols tested over a tactical network with disconnected, limited and intermittent states [15]. The tactical network was composed by both simulated and emulated network technologies, such as SatCom, WiFi and Combat Network Radio (CNR). This heterogeneous testbed had links using different data rates ranging from 50 kbps to 2 mbps, delays between 5-550 ms, and packet error rate from 0 to 20% loss. Network disconnections were created by unplugging the cable for 60 seconds (non-stochastic) and plugging it back to check if the proxy could resume the transmission after the disruption. The study recommends HTTP together with GZIP to the majority of communication technologies suggesting that the network conditions were rather stable (non-stochastic) during their experiments.

The investigation in [25] extends the agile computing middleware to detect network states within a mobile node. The goal is to share the network states among other nodes and to adapt the nodes’ behavior (using the network) to the current network state. The definition of a node state is broad including the hardware and operating system, which are complemented by network specific characteristics such as the cluster of nodes, data flows (incoming/outgoing), network topology, various estimated metrics and alerts. The main contribution is an algorithm to compile and share the state of a cluster of nodes among other nodes in the network. The network overhead to compile and share the node’s status was discussed for 5, 10, 15 nodes, but the system’s ability to detect changes in the network was not quantified.

The investigation in [8] discusses field experiments with a proxy to exchange messages among four vehicles in a convoy moving through a hilly area in Germany. The four vehicles were using a VHF network during the whole experiment and the nodes were exchanging their positions (Friendly Force Tracking) and sending emails using the proxy to wrap Simple Mail Transfer Protocol (SMTP) messages to the bundle protocol originally designed for Disruption Tolerant Networks (DTN). The user(s) data flows are non-stochastic using two types of messages. The proxy is reactive without feedback from the radio network therefore not implementing the cross-layer information exchange. The quantitative results are discussed with focus on the user experience during the convoy movement. The network conditions during the experiments were not quantified.

| Notation | Definition |
|----------|------------|
| $A$ | Markov chain from model $A$ |
| $\bar{A}$ | Sequence of messages |
| $B$ | Markov chain from model $B$ |
| $f,m<\text{metric}>$ | Distribution function $f$ defining a metric \{size, reliability or ToE\} for messages $m$ |
| $\lambda$ | Arrival time distribution |
| $l$ | Layer number |
| $L_l^r(i_l,o_l)$ | Layer $l$ connected to radio network $r$ with two enforcement points $(i_l,o_l)$ and a control point $c_l$ |
| $\text{MTU}$ | Maximum Transport Unit (bytes) |
| $n$ | Number of command and control services |
| $N_r^r,i$ | Node $i$ connected to radio networks $r$ |
| $o_l$ | Enforcement point at outgoing chain at layer $l$ |
| $Q_{l,m}^{<\text{metric}>}$ | Message metric \{delay, reliability, size and ToE\} |
| $\omega$ | Time window (seconds) |
| $\Delta r_0$ | Current radio link metric \{data rate, latency, loss\} |
| $\Theta_c$ | A pipe, instance of model $A$/$B$, connected to the radio network $r$ |
| $X_0$ | Initial vector |

### 2.3.2 Previous works

In [19], we defined a hierarchy of queues to adapt the user data flows to the current network conditions to avoid radio buffer overflow. The control mechanism was reactive, pausing the data flow when a pre-defined threshold was crossed. We observed buffer underflow and overflow during experiments with different thresholds, which motivated further research. The experiments were performed with one type of message, unreliable and with a random payload, with size four times larger than the radio buffer. Therefore, our experiments were missing realistic diversity of messages with different QoS requirements. Moreover, the network conditions were stable throughout the whole experiment.

In [2], we sketched the first version of the three models $A$, $B$ and $A$/$B$ to solve the problems being addressed in the present investigation. Complementing, in [3], we did an exploratory study instantiating model $A$ to create diverse sequences of QoS-constrained messages to discuss the implications of network disconnections in the queue of messages over numeric simulations. In [4], we reported experiments with an improved version of our queuing mechanism handling three patterns of data rate change.

Here, we improve model $A$ by adding a no service call state and model $B$ by adding a disconnected state and redoing the experiments over ever-changing communication scenarios. Model $A$/$B$ was extended to include a layer for the software defined radio, and we also reformulated the control functions using a more rigorous mathematical description. The multi-layer control loop uses both link data rate and buffer occupancy combining a reactive and proactive control to adapt the user data flows to ever-changing communication scenarios. Our new control mechanism is more precise and robust while differentiating QoS-constrained messages and managing the radio buffer occupancy.
3 Problem A

3.1 Ever-changing data flows

Let us assume that a set of command and control services are available to the user(s) through mobile nodes in a tactical network. Each service has its own set of QoS-requirements, predefined by experts or computed during run time, which are included in the message header. For example, Table 2 lists five exemplary services together with their frequency \( \lambda \), priority, reliability (yes or no) and Time of Expire (ToE). Priority is a number representing four different types from 0 (highest) to 3 (lowest), namely flash 0, immediate 1, priority 2 and routine 3. Reliability means that the receiver must acknowledge the sender when the message is successfully received. If the message is lost or partially received, the system must re-send the entire message or the lost packets. Notice that we also included the “no service call” \( s_0 \) to represent the state where no message is sent to the tactical network.

Given a mission or operation, the users will combine services with different QoS-requirements to achieve their objective(s). The frequency of messages sent to the network depends on the action in the battlefield therefore, we placed interrogation marks “?” in the lambda \( \lambda \) column listing the arrival time distribution in Table 2. For example, it is difficult to predict when an obstacle will be found or when a medical evacuation will be requested. However, some services (say \( s_5 \)) may send messages in a known frequency distribution (say \( \lambda_5 \)) like Friendly Force Tracking (FFT). Therefore, we seek to define a model to combine these different types of messages in different ways so as to include a wide range of military activities (e.g. disaster recovery, surveillance, combat and so on).

**Problem A:** How to combine a set of QoS-constrained messages to create a wide range of message combinations, including the element of chance (randomness) in extreme communication scenarios?

3.2 Modeling ever-changing sequence of messages

Problem A asks for a stochastic model with states representing messages from the user services listed in Table 2. Model A assumes that it is possible to define the probability of the service \( s_i \) being called after service \( s_j \) or \( p_{ij} \) creating a Markov chain defining the probability distribution among all service calls, as illustrated in Fig. 3. This figure shows a complete graph listing all transitions among the five exemplary services including the state 0 representing no message sent to the network. Thus, adding new service(s), say \( s_n \), to the model implies in adding new state(s) to the chain, which is represented by a \( n \times n \) matrix listing the conditional probabilities.

| Service            | \( \lambda \) | Priority | Reliability | ToE (sec) |
|--------------------|---------------|----------|-------------|-----------|
| \( s_0 \) No service call | ?            | -        | -           | -         |
| \( s_1 \) Chat     | ?            | 3 Routine| No          | 120       |
| \( s_2 \) Picture   | ?            | 2 Priority| Yes         | 3600      |
| \( s_3 \) Obstacle alert | ?     | 1 Immediate| Yes         | 150       |
| \( s_4 \) Medical evacuation | ?   | 0 Flash   | Yes         | 300       |
| \( s_5 \) FFT      | \( A_5 \)    | 3 Routine| No          | 120       |

Let us instantiate model \( A \) to create three sequences of messages \( A_1, A_2 \) and \( A_3 \) to illustrate the discussion throughout this paper. Each sequence is generated by its own Markov chain represented by the matrices in (1). This relation lists three \( 5 \times 5 \) matrices putting together the four services (\( s_1, s_2, s_3 \) and \( s_4 \)) including the “no service call” \( s_0 \). The numbers in these three matrices define the probability of a service \( s_i \) to be called if the service \( s_j \) was called before, thus the notation \( s_i|s_j \) or in short \( p_{ij} \). Taking matrix \( A_3 \) as an example, it states that there is a 1% probability of a medical evacuation \( s_4 \) to be called after a medical evacuation or \( A_3(p_{44}) = .01 \), but there is a 49% probability of an obstacle alert \( s_3 \) to be called after a medical evacuation or \( A_3(p_{34}) = .49 \) and so on. After twenty interactions \( A_3^{(20)} \) the matrix reaches the steady state with all lines converging to \( (.01 \ .26 \ .35 \ .29) \); therefore the probability \( A_3^{(20)}(p_{44}) \) starts in 49% and converges to 35%, and \( A_3^{(20)}(p_{34}) \) starts with 1% and converges to 29%.

The probabilities in these three matrices were arbitrarily chosen to create different sequences of messages. For example, \( A_1 \) defines high probability of no service call \( p_{00} = 40\% \), routine messages \( p_{12} = 20\% \) and priority messages \( p_{23} = 30\% \). In sequence, \( A_2 \) defines the same probability for all message types \( p_{ij} = 20\% \). Finally, \( A_3 \) defines zero chance of no service call \( p_{00} = 0 \) and high probability for messages with high priority (states \( s_3 \) and \( s_4 \)). The initial vector \( X_0 \) was also defined arbitrarily starting the process from state \( s_1 \) or \( p_{11} = 1 \).

**Model A:** the state machine \( A \) defines the distribution of message type, the arrival time is defined by \( \lambda_A \) together with a distribution function \( f \) for all message metrics/requirements \( f.m < \text{metric} > \) resulting in \( Model_A(A, \lambda, f.m < \text{metric} >) \), where metric is a set of metrics {size, reliability, time of expire}. Therefore, model \( A \) is a set of distributions defining the occurrence of different message properties.
Algorithm 1 Creating a sequence of states

Input: $X_0$, $A$, seed, length
Output: $\bar{A}$

**Initialization:**
1. $\bar{A} \leftarrow$ new array(length)
2. $n \leftarrow X_0.length$
3. for $i = 1$ to $\bar{A}.length$ do
4.   $\bar{A}_i \leftarrow X_0 * A^i$
5. end for
6. $x \leftarrow 0$
7. for $j = 1$ to $n$ do
8.   $x \leftarrow X_j + x$
9.   if (dice $\leq x$) then
10.      $\bar{A}[j] \leftarrow j$
11. end if
12. end for
13. return $\bar{A}$

3.2.1 Using model A

In this investigation, we are particularly interested in message priority for differentiating the messages in the queue of messages of a tactical system. Also, considering the time of expire to drop expired messages to shape the user(s) data flows using message’s QoS requirements. Thus, let us use the matrices in (1) to create three sequences of messages to illustrate the discussion throughout this paper. The algorithm 1 lists the steps starting with the instance of an array $\bar{A}$ with a given length to store the sequence of states and the number of states $n$ from the initial vector $X_0$ (lines 1 and 2). Then a loop goes through the array populating the sequence as follows (line 3). First, a random number (dice) is drawn from a random number generator using a given seed (line 4). Second, we multiply the initial vector $X_0$ by the given matrix $A$ power $i$, where $i$ is the current sequence number within the loop (line 5). Third, we match the dice with the range of probabilities in $X_i$ computing the next state and attributing it to the sequence array (lines 9-12). This procedure is repeated for every single state in the sequence $\bar{A}$, which is returned in the end (line 15).

Fig. 4 plots the three sequences of messages in arrival order (left) and sorted by priority (right). The three sequences of messages, shown in figures (a)(c)(e), were generated with the same seed for the random number generator but with different distributions defined by the three matrices in (1). As a result, there are seven messages with high priority that appears in the same place (black squares at positions 10, 20, 23, 51, 63, 65, 77) and there are eighteen “no service call” (white squares at positions 6, 11, 12, 14, 19, 32, 33, 40, 47, 48, 49, 58, 68, 75, 80, 83, 84, 85, 90, 93 and 99) which are at the same place in (a) and (c). The sorted plots in figures (b)(d)(f), shows the message differentiation as a fundamental feature of the message queue (later discussed in section 5), which can be implemented with a queue for each message priority in a hierarchical token bucket. Each sequence, $A_1$, $A_2$, $A_3$ in the figure, has one hundred messages using the five exemplary services $s_{0,1,2,3,4}$ in Table 2. The messages are numbered from left to right indicating the sequence in which the messages will be sent to the tactical network. The colors indicate the message priority, the darker the higher: 0 flash (black), 1 immediate (dark red), 2 priority (red), 3 routine (light red) and no message (white).

3.2.2 Why ever-changing data flows?

We argue that these sequences of messages are creating ever-changing data flows to contrast with experiments reported in the literature (Section 2.3). The word ever indicates change through infinity but, here, we leverage the term to indicate an arbitrary but “big enough” combination of messages. For example, let us compute the possible permutations in Fig. 4 which has five types of messages with different priorities and is a sequence of one hundred messages long or $(100)^5 = 75,287,520$ or more than 75 million possible combinations. And in Fig. 5 each queue has twenty messages of five types then $(20)^5 = 15,504$ or more than fifteen thousand possible combinations.

Since these numbers can get larger in real communication scenarios in tactical networks (e.g. with more than four services) we call it ever-changing sequences of QoS-constrained messages. This diversity of messages is visible in figures 5(b)(d)(f) plotting twenty queues (columns) with
To the best of our knowledge, logs from military deployments are highly sensitive. Therefore, scientific publications are usually limited to synthetic and non-stochastic models to generate user data flows, as discussed earlier in Section 2.3.1. This fact motivated us to introduce model A to create a wide range of message-based data flows, which includes extreme scenarios, to test the performance bounds of tactical systems. Our model assumes that it is possible to define the conditional probabilities among different types of messages using a Markov chain. Since the literature makes a clear distinction between message-based and stream-based data flows, model A only creates the former.

For example, the experiments reported later in Section 6 uses the three pattern of messages discussed in this section to create data flows exceeding the nominal capacity of a VHF network (extreme scenario). The different sequence of messages test the system’s ability to differentiate the classes of message priority.

### 4 Problem B

#### 4.1 Ever-changing network conditions

Let us assume that a multi-homed node $N^r$ connected to $r$ radio networks will experience $r$ network states (e.g. connected, limited and disconnected) simultaneously (for $r > 1$). For example, a node $N^3_1$ is connected to three different radio networks $r = 3$ or namely $r = \{\text{VHF, UHF, WiFi, SatCom}\}$ varying independently. Fig. 6 shows a complete graph with the three network states, where state 0 is disconnected (red), state 1 limited (yellow) and state 2 connected (green). In the literature [7], [15], other authors define the network states as disconnected, intermittent and low-bandwidth, here we include the connected state and assume that intermittent is any variation of states in a given time window. The precise definition of these three network states depends on the network technology.

For example, Table 3 lists four exemplary network technologies (VHF, UHF, WiFi and SatCom) and particular ranges of metrics (data rate and latency) to define these three network states (columns); the manufacturer and model of military equipment should also be considered, but is omitted here for simplicity. In this table, the VHF network assumes the disconnected state with disruptions longer than few minutes and WiFi assumes the same state with shorter disruptions of just few seconds. And the limited state is squeezed between connected (less than state 2) and disconnected states (greater than state 0 in Fig. 6). In the connected state, a VHF radio can have up to 64 kbps of shared data rate (e.g. the radios used in this investigation have maximum nominal data rate of just 9.6 kbps) and latency >2 seconds, and disruptions longer than few minutes will be detected in the IP layer assuming the disconnected state (e.g. no routes at the operating system and loss of IP packets). As a result, everything between the bounds defined by the connected and disconnected states is classified as limited.

The metrics defining the bounds of the network states (nominal/current data rate, latency, link quality and so on) can be gathered during run time using protocols like DLEP [23] and using SNMP. Thus, tactical systems can use these metrics as feedback to adapt its queues to the current network conditions.

**Problem B: How to create sequences of network states including the element of chance (randomness) and ensuring a certain distribution of change to test store-and-forward mechanisms in**

---

**TABLE 3: Network states for different network technologies**

|           | Disconnected 0 | Limited 1 | Connected 2 | Range   |
|-----------|----------------|-----------|-------------|---------|
| VHF       | > 1 min        | < state 1 | < 64 kbps   | 20 km   |
| UHF       | > 10 sec       | < state 1 | < 240 kbps  | 2 km    |
| WiFi      | > 1 sec        | < state 1 | < 250 mbps, .01 sec | .2 km |
| SatCom    | > 5 min        | < state 1 | < 500 kbps  | -       |

---

**Fig. 6: The three network states: 0, 1 and 2**
disconnections. Using the matrices in (2), three without link state transitions among the six states in a complete graph.

4.2 Modeling the ever-changing network states

We noticed that our VHF radios have a SNMP interface to change the radio modulation supporting five data rates, namely 0.6, 1.2, 2.4, 4.8 and 9.6 kbps. Thus, we developed an algorithm to change the radio data rate in run time following a given pattern defined by our model B as follows.

Let us use the five data rates supported by our radios as states of a Markov chain along with a state for disconnection, as illustrated in Fig. 7. In this figure, state 0 is disconnected, state 1 represents the data rate .6 kbps, state 2 represents 1.2 kbps and so on until state 5 that represents 9.6 kbps, which is the highest nominal data rate supported by our radios. Notice that the edges of this state-machine are defined in general terms as $p_{ij}$ showing all the possible state transitions among the six states in a complete graph.

We can instantiate this model to create six sequences of network states using the matrices in (2), three without link disconnections ($B_1$, $B_2$ and $B_3$) and the other three using similar probabilities but with link disconnections ($B_4$, $B_5$ and $B_6$). These $6 \times 6$ matrices list the probability distribution connecting together the five data rates $b_i$ supported by our radios $b_i | p_{ij}$ plus the state 0 representing a disconnection. For example, given a data rate $b_1$ (say .6 kbps) the probability of changing to $b_2$ (say 1.2 kbps) is $Pr(b_2 | b_1) \sim p_{21}$ and so on. Notice that $B_1$ and $B_2$ have higher probabilities (> .3) for low data rates (< 4.8 kbps). $B_2$ and $B_3$ create a pendulum pattern with round 1 and round 2; round 1 decreases the data rate (i.e. swinging from the right to the left of Fig. 7) with $x_1 = 1$ and $x_2 = 0$ whereas round 2 increases the data rate with $x_1 = 0$ and $x_2 = 1$. And $B_3$ and $B_6$ have higher probabilities for states 3, 4 and 5, which represent high data rates (> 2.4 kbps).

Model B: the distribution of states is defined by the Markov chain $B$ and the time distribution for state change is defined by $\lambda$ or Model$_\lambda(B, \lambda)$. For example, the experiments in Section 6 use these six matrices in (2) to define the distribution of data rates together with a constant distribution of time $\lambda = 20$ seconds.

4.2.1 Using model B

Using the six matrices in (2) as input to algorithm 1 (explained earlier in section 3.2.1) we get the six sequences of states plotted in Fig. 8. In the left column of this figure, (a)(c)(e), there are three sequences of states without disconnections and in the right column (b)(d)(f) there are similar sequences but including link disconnections (zeros in red).

These sequences can be examined in pairs starting from the top where we have the pair (a)(b) compiling the lowest data rates on average, with about 1.6 kbps for $B_1$ (a) and about 0.9 kbps for $B_4$ (b). In the middle, the $B_2/B_5$ sequences behave like a pendulum pattern from the highest data rate (state 5) to the lowest (state 0 or 1) and back and forth. At the bottom, is the pair $B_3/B_6$ with the highest data rates on average (states 3, 4 and 5), about 5.8 kbps for both $B_3$ (e) and $B_6$ (f) patterns of data rate change.

4.2.2 Why ever-changing link data rates?

Again, we use the term ever-changing to refer to these patterns of data rate change because each plot in Fig. 8 has one hundred states out of six possible states, which results in a combination of $\binom{100}{6} = 1, 192, 052, 400$. Thus, more than one billion combinations is big enough to call it ever-changing. Also, notice that Fig. 8 is annotated with three symbols: “x” for increment in data rate, “=” for equal data rate and “-” for decrease in data rate. Thus, one can verify the ever-changing property of these six pattern searching for the occurrence of only few equal signs between states.

4.3 Motivation and implementation B

We develop model $B$ to define in precise terms a wide range of distribution of data rate change during experiments in a VHF network. The motivation comes from our literature review, where we noticed that the variations in the network conditions are non-stochastic, especially during experiments creating network disconnections as a function of time. Moreover, the changes in the network conditions are generally unknown until the simulation or field experiment is executed.

We argue that any tactical system tested with the patterns of change generated by our model has a higher probability of success during field experiments (e.g. a convoy of vehicles moving through a contested region). Notice that this model can also be implemented using a channel
5 PROBLEM A|B

5.1 Given B, how to handle A?

We start with the hypothesis that a multi-layer queuing mechanism can cope with ever-changing communication scenarios, user data flows (A) and network conditions (B) changing independently, using the cross-layer information exchange within the control plane (e.g., anticipated by the services taxonomy earlier shown in Fig. 2). Remember that A defines the user-generated data flows with different QoS parameters (Section 3) which need to be adapted to the current network state B, here called Problem A|B and defined as follows.

\[ B_i \sim \text{random} \pm \text{standard deviation} \]

![Fig. 8: Six patterns of data rate change](image)

Fig. 8: Six patterns of data rate change

emulator to introduce different attenuation levels to create different link data rates. Then, these levels of attenuation would be the states of a new Markov chain.

Regarding the application of model B in a tactical network with several links, the VHF links can reuse the model described here. Each radio link will have its own matrix B defining the probability distribution of data rate changes. For other communication technologies (say SatCom or UHF) our model must be adapted by redefining the states to match the supported data rates.

5.2 Modeling the in/out chains and control plane

Recent software platforms have chains of enforcement points for input (receiving) and output (sending), so control and security mechanisms can transform both incoming and outgoing data flows. We have been evolving a model to solve problem A|B placing these in|control|out points within the layers of a services taxonomy, as illustrated in Fig. 9. In this figure, the in|control|out chains are organized in four layers \( L_l \), \( l = \{0, 1, 2, 3\} \), matching with the main functional blocks from the services taxonomy, namely Radio \( L_0 \), Communications \( L_1 \), Core Enterprise \( L_2 \) and Community of Interests \( L_3 \). This model can be further extended to include other layers/sub-layers allowing us to define in precise terms how the control mechanisms, sitting at \( c_{0..L} \), actuate in the layers \( L_l \) using two enforcement points \( i_l|c_l|o_l \), where \( i_l \) and \( o_l \) are the in/out chains. The mathematical notation is also defined in Table 1.

The in/out chains symmetrically complement each other receiving/sending, decrypting/encrypting, defragmenting/fragmenting messages and so on. This construction is also used to forward data flows from in to out within the same layer to model IP packet routing at layer 0 or 1 and message forwarding at layer 2 or 3. As a result, the four main layers have two enforcement points in/out and a control point or \( L_l(i_l|c_l|o_l) \). And a multi-homed node \( N_r \) has a pipe \( \Theta_r \) (set of layers \( L_l \)) for each radio network \( r \) with four layers as defined in our pipe model (3).

\[ \text{Model } A|B : N_r \sim \Theta_{1..r} \]

![Fig. 9: Pipe model: layers \( L_l \) with \( \langle i_l|c_l|o_l \rangle \)](image)

Problem A|B: How to define multi-layer enforcement and control points hosting a hierarchical queuing mechanism to adapt the ever-changing user data flows to the ever-changing network conditions?
Let us use the pipe model in (3) to define a multi-layer control loop to adapt the user(s) data flows to the network conditions. First, the user messages are intercepted at layer 3 \( c_3 \mid o_3 \) which queries the control point at the communications layer 1 retrieving routing information to queue the message in the pipe \( \Theta_1 \) connected to the radio network \( r \) that can reach the IP destination from \( o_2 \) (we omit name resolutions for simplicity). This functionality is often implemented in a proxy/broker in the literature which here is a layer shared (*) among all the pipes \( \Theta_1 \). At \( c_2 \mid o_2 \), the control point \( c_2 \) computes the time for admission for the \( L_1 \) below, holding the messages in a queue of messages at the outgoing chain \( o_2 \). In sequence, \( c_1 \mid o_1 \) computes the inter-packet interval to be enforced at \( o_1 \). \( Q \) shaping the user data flow to the current network state. Finally, at \( c_0 \mid o_0 \) the radio \( r \) keeps reporting its buffer occupancy \( Q_0 \) that is shared with the upper layers through the control plane.

Moreover, if one of the radio networks is disconnected (worst case), the messages/packets may move horizontally to be queued in a pipe that can reach its destination attaining the QoS requirements through a different radio network. We proceed discussing the fundamental computations used to control the pipe’s out chain namely, time for admission, computing expired messages and inter-packet interval in the following subsections.

### 5.2.2 Time for admission \( c_1 \mid o_1 \)
We assume that in a pipe with \( l \) layers it is possible to get the status of the underlying layers within a time window while controlling the user data flows through the cross-layer contextual monitoring in the control plane. For example, every layer \( l \) at the out chain has a queue \( Q_l \), its size is shared through the control plane \( c_l \) and its delay \( Q_l \) is controlled in \( c_l \). Therefore, we can compute the waiting time \( Q_l \) to allow the waiting layers to send the message, which is the time to empty the underlying layers in the pipe \( \Theta_1 \), here called time for admission. Time for admission is a fundamental feedback to manage the queue of messages at the out chain or \( o_2 \). In general terms, given a radio network \( r \), Eq. 4 computes the time to empty the underlying queues \( Q_l \).

\[
Q_l = \lim_{\Delta o_0 \to 0} \left( \frac{\sum_{i=1}^{l-1} Q_i}{\Delta o_0} \right)
\]

Where \( l \) is the layer from which time for admission is computed, \( Q_i \) is the queue size (e.g. kbits) at the outgoing chain \( o_i \) at layer \( l \), \( \Delta o_0 \) is the estimated data rate from the radio network \( r \) (e.g. kbps) which is tending to a number \( a \) other than zero (\( a \neq 0 \)). This is because zero indicates link disconnection therefore, undefined waiting time (division by zero).

### 5.2.3 Computing expired messages \( c_2 \mid o_2 \)
Given the link data rate \( \Delta o_0 \), we can go through the sequence of messages computing the time in the queue \( Q_{delay} \) for each message \( m_i \) using the relation in Eq. 5. This relation sums the size of the preceding messages to \( m_x \) (say in kbytes) and divides by the data rate \( \Delta o_0 \) (say in kbps) which is tending to a number \( a \) other than zero (\( a \neq 0 \)). If the time in the queue is larger than the time of expire, the message is dropped. Dropping expired messages is an effective way of adapting the user data flows to the current network conditions using the messages QoS requirements. Moreover, this computation must be done frequently to keep the queue updated (say every s seconds).

\[
Q_x = \lim_{\Delta o_0 \to 0} \left( \frac{\sum_{i=1}^{x} Q_i \cdot m_i}{\Delta o_0} \right)
\]

We can also compute the minimum data rate to deliver all the messages in the queue without expiration. The relation in Eq. 6 computes the minimum data rate in the queue, \( Q_2 \), using the cross-layer information shared by the radio (layer 0) and the routing protocol (layer 1). In previous investigations, we defined two mechanisms: (i) using the radio buffer occupancy \( o_0 \) and (ii) using the estimated link data rate compiled by the routing protocol \( Q_0 \). Here, we combine these two solutions using the following system of relations.

Given a radio network \( r \), if the buffer occupancy \( \Delta Q_0 \) is zero, then the delay is also zero, see Eq. (7). If the buffer occupancy is below a threshold (\( \Delta Q_0 < Q_0 \)), the relation in (7) computes the inter-packet interval \( Q_1 \) using the time window \( \omega \) (seconds), the IP packet size \( \omega \) (bytes), the amount of bytes the radio will send within the time-window out. \( Q_0 \) (8a), how many bytes is left until the threshold in \( Q_0 \) (8b) and minimum latency \( o_0 \) (seconds). However, if the radio buffer is above a given threshold \( \Delta Q_0 \geq Q_0 \), then the system pauses the transmission by setting an undefined delay (say \( Q_1 = \infty \), that is the control becomes reactive) until the buffer is below the threshold as defined in (7). As a result, the mechanism goes switching between proactive and reactive behavior as function of both link data rate and buffer occupancy over time (multi-layer control loop).

\[
Q_1 = \begin{cases} 0, & \text{if } \Delta Q_0 = 0 \\ \omega \cdot Q_0 \text{MTU} / \omega + \text{out}, Q_0 + o_0, & \text{if } \Delta Q_0 \geq Q_0 \end{cases}
\]
5.3 Motivation and implementation $A/B$

Model $A/B$ was designed to host any multi-layer control loop acting in both incoming and outgoing data flows. The motivation comes from our literature review (Section 2.3) where recent literature lacks mathematical explanations on how the tactical systems adapt its behaviour to the changes in the communication scenarios. We also sought to complement the state-of-the-art of admission control and scheduling, such as the investigations reported in [26], [27], by introducing a hierarchy of queues complementing each other through multi-layer control loops.

Our model $A/B$ anticipates the possibility of moving packets horizontally among different pipes connected to different networks. This feature cannot be implemented at the radio level (layer 0 in Fig. 9) but can be implemented at the higher layers handling IP packets (layer 1) and messages (layer 2). For example, the tactical system used in the experiments reported in Section 6 uses the dynamic routes computed by Optimized Link State Routing (OLSR) [28] to trigger the horizontal movement among different pipes.

6 RESULTS

This section discusses the performance of our tactical system implementing model $A/B$ over ever-changing communication scenarios, created with models $A$ and $B$, in a VHF network inside a laboratory.

6.1 Experimental setup

Our experiments reused the 500 kB data flow from previous investigations for comparison purposes [4], [19] and monitored both sender and receiver (highlighted earlier in Fig. 1) at the three layers, namely radio buffer ($Q_0$ at layer 0), queue of packets ($Q_1$ at layer 1) and queue of messages ($Q_2$ at layer 2). Table 4 summarizes the experimental setup listing the radio type, time window, buffer threshold, data flows, network conditions and routing protocol used during the experiments discussed in the next sub-sections. OLSRv2 was used to create an overlay network putting all nodes in a single IP domain. The neighbor discovery was configured to use less than 2% of the maximum nominal data rate in our VHF network.

![Fig. 10: Radio buffer (%) and data rate (kbps)](image)

We can divide these patterns into two groups: one without link disconnections (state 0), \{$B_1, B_2, B_3$\}, and the other with disconnections, \{$B_4, B_5, B_6$\}. We observed packet loss during the experiments using the three patterns with link disconnections of about 98, 67 and 2% loss, respectively. Moreover, the frequent network disruptions in the patterns $B_4$ and $B_5$ broke the IP routes among the radios which were recovered after few minutes of stable connectivity. As a result, we learned that our system cannot handle these two patterns of variation within a 20 seconds time-window. However, pattern $B_6$ is manageable using a reliable transport protocol.

In the following sections, we discuss the experimental results layer by layer starting with the radio buffer $Q_0$, followed by the queue of packets $Q_1$ and ending with the queue of messages $Q_2$.

6.2 Layer 0: radio buffer $Q_0$

Remember that the radio buffer is the most constrained queue in the pipe therefore, we monitored the buffer occupancy as a function of time as plotted in Fig. 10. In this figure, the buffer occupancy (left vertical axis) is plotted together with the link data rate (right vertical axis) during the three experiments without disconnections, namely $B_1$, $B_2$ and $B_3$. The buffer usage (black) lies around 10% (pre-defined threshold) showing a similar pattern even with dissimilar patterns of data rate change. This implies that our solution is robust to data rate changes.

| Item            | Description                                                                 |
|-----------------|------------------------------------------------------------------------------|
| Radio           | 3 VHF radios: PR4G by Thales with wired antennas                            |
| Time-window $\omega$ | 20 seconds                                                                  |
| Buffer threshold | 10% of 128 kB                                                               |
| Data flows      | 500 kB data flows $A_{1,2,3}$ with random payload                           |
| OLSRv2          | 45 seconds for hello and aggregation intervals                             |
| Network conditions | $B_{1,2,3,4,5,6}$, the six patterns in Fig. 8                             |

We performed the experiments over the six patterns of data rate change, from $B_1$ to $B_6$, earlier plotted in Fig. 8.
The experiment plotted in Fig. 10a lasted for \( \sim 73 \) minutes because the data rates (red dots) were mostly among \( .6, 1.2 \) and \( 2.4 \) kbps, with the average buffer occupancy around \( 10\% \pm 1.66 \). In sequence, Fig. 10b plots a \( \sim 42 \) minutes long experiment with the data rate variation like a pendulum with average buffer occupancy of about \( 9.4\% \pm 2.01 \). Finally, Fig. 10c plots a \( \sim 31 \) minutes long experiment with data rates (green dots) mostly among \( 2.4, 4.8 \) and \( 9.6 \) kbps with average buffer occupancy of about \( 9\% \pm 2.02 \). The motivation to keep the radio buffer within the 10 % threshold comes from the fact that a packet cannot be removed/rescheduled when it has been sent to the radio buffer. Although, modern military radios can differentiate IP packets in the buffer using Differentiated Services Code Point (DSCP) field in the IP header to define the class.

### 6.3 Layer 1: queue of packets \( Q_1 \)

Moving one layer up, to the queue of packets, these patterns of buffer usage around a predefined threshold are the result of our hybrid solution that proactively introduces inter-packet interval \( Q_1^{\text{delay}} \) and re-actively pauses the data flow when the threshold is crossed, as previously discussed in Section 5.2.4. The difference in the data rate pattern also changes the way the system uses the queue of packets as plotted in Fig. 11. The queue size \( Q_1^{\text{size}} \) is plotted as a function of time showing that our solution relies more on the queue of packets during the more challenging network conditions in \( B_1 \) (red) followed by \( B_2 \) (blue) and then by \( B_3 \) (green) which is the best scenario, if compared to the other two.

Fig. 12 plots the inter-packet interval at both sender and receiver over \( B_1 \) (red), \( B_2 \) (blue) and \( B_3 \) (green). The sender plot shows the density of waiting time introduced by our solution (Fig. 12a) and the receiver side shows the effects of the transmission over the VHF radios following the three data rate patterns (Fig. 12b). Complementing, Fig. 12c plots everything together in a strip-chart to highlight the differences between the delay intervals observed at the sender (S) and receiver (R). There are at least two observations in this figure: (i) the intervals introduced by our solution at the sender are smoothed out by the radio buffer \( Q_0 \) and air time (interval between leaving the sender’s radio and arriving at the receiver’s radio computed at IP layer); (ii) the range of inter-packet intervals vary as an inverse function of the network conditions, lower data rates means greater IPI range; e.g. 0-30 seconds for \( B_3 \) (green) and 0-60 seconds for \( B_1 \) (red).

Table 5 lists the experiment’s total time which is divided into two parts, namely time using the proactive and reactive mechanisms. The second part of the table lists the averages

### 6.4 Layer 2: message queue \( Q_2 \)

#### 6.4.1 How long to admission?

Another layer up, at the message queue, we can discuss the difference between using one large message or a sequence of messages to create a 500 kbps data flow. Remember that at this layer there are two control functions, one computing the time for admission (Eq. 4) and the other computing the expired messages (Eq. 5). Fig. 13 plots the time for admission as a function of time for the three patterns of data rate change without link disconnections. The peaks for \( B_1 \) (red) and \( B_2 \) (blue) reach more than 550 seconds and the peak for \( B_3 \) (green) less than 150 seconds. These numbers were computed with the relation in Eq. 4 using as input the data rate set by the script implementing the patterns of change (i.e. not using the data rate computed by the routing protocol). Notice
that the three curves go slowly converging to zero as a function of time, and the most challenging conditions in $B_{1 \max}$ implied a longer variation until reaching zero.

### 6.4.2 Expired messages

Given the waiting time in the queue $Q_3 \cdot m_x \cdot \Delta \overline{t}_{OE}$ (defined earlier in Section 5.2.3), we can compute the number of expired messages using the exemplary time of expire $Q_3 \cdot m_x \cdot \Delta \overline{t}_{OE}$ also listed in Table 2. Fig. 14 plots the number of expired messages for the three sequence of messages ($A_1$, $A_2$ and $A_3$ defined earlier in Section 3 to illustrate the use of model $A$) as a function of data rate (varying from 0 to 9.6 kbps) and two different message sizes (.5, and 2 kB). When the data rate is zero (worst case) all the messages are expired (100%) as the data rate improved (doubling from .5 to 9.6 kbps), lesser number of expired messages were observed. Notice that for small messages (.5 kB) these three sequences do not challenge the store-and-forward mechanism for data rates higher than 2.4; i.e. zeros on 4.8 and 9.6 kbps on the right hand side of Fig. 14.

### 6.4.3 Delivery rate and delay by priority class

Now we can move to the receiver side to compute the message delivery rate and message delay observed during our experiments. Table 7 compiles these two metrics for the three data flows $A_{1,2,3}$ (rows) against the three patterns of data rate change $B_{1,2,3}$ (columns). Remember that the data flows $A$ have four classes of message priority, namely 0 flash, 1 immediate, 2 priority and 3 routine. The numbers in this table show a high delivery rate (%) and low delay (seconds) for messages with higher priority (0, 1 and 2). And the routine messages, which are also unreliable as defined earlier in Table 2, were dropped during all experiments with data flows exceeding the network capacity (extreme scenario). For $B_1$, the pattern with lowest data rate, the priority (2) messages were also dropped because of their low time of expire, 150 seconds, when compared to 300 and 3600 seconds for flash 0 and immediate 1, respectively. These results suggest that our queuing mechanism is robust to changes in the network and can differentiate message-based data flows.

![Fig. 13: How long to admission? B1 (red), B2 (blue) and B3 (green)](image)

![Fig. 14: Number of expired messages in the three queues](image)

Complementing Fig. 14, Table 6 lists the number of expired messages in the three queues $A_1 | A_2 | A_3$ (with message size of .5, and 2 kB for illustration) against the range of nominal data rates from our VHF radios from 0 to 9.6 kbps. This table also lists the minimum data rate min($\Delta \omega$) (earlier defined in Eq. 6) for delivering each sequence of messages without letting the messages expire; i.e. no message is dropped in the queue. Notice that the sequences of messages with 2 kB challenged our queuing mechanism for all data rates because we observed expired messages even in the best conditions (9.6 kbps); except for $A_1$ which has 40% of no service call in Fig. 4b.

### 7 Discussion

The models developed in the present investigation explored the available metrics/requirements to define mathematical abstractions that were implemented in a tactical system/network and verified with experiments. Thus, our solutions for the three problems addressed in this investigation are bounded by the following challenges/limitations:

- Limited access to user-generated data flows from real military deployments or exercises. Instead of emulating realistic data flows, model $A$ was designed to create a wide range of randomly combined messages that can be reproduced for quantitative comparisons;
- Limited interfaces with military radios to dynamically retrieve network metrics in run time. This scenario is likely to improve with the growing adoption

---

**TABLE 6: Number of expired messages**

| $\Delta \omega$ kbps | $A_1$ | $A_2$ | $A_3$ | $A_1$ | $A_2$ | $A_3$ |
|----------------------|-------|-------|-------|-------|-------|-------|
| 0                    | 100   | 100   | 100   | 100   | 100   | 100   |
| .5                   | 100   | 100   | 100   | 100   | 100   | 100   |
| 1                    | 100   | 100   | 100   | 100   | 100   | 100   |
| 1.2                  | 100   | 100   | 100   | 100   | 100   | 100   |
| 2.4                  | 100   | 100   | 100   | 100   | 100   | 100   |
| 4.8                  | 100   | 100   | 100   | 100   | 100   | 100   |
| 9.6                  | 100   | 100   | 100   | 100   | 100   | 100   |

**TABLE 7: Delivery rate and message delay**

| Pattern | Priority | Delivery (%) | Delay (sec) | Delivery (%) | Delay (sec) | Delivery (%) | Delay (sec) |
|---------|----------|--------------|-------------|--------------|-------------|--------------|-------------|
| $A_1$   | 0        | 100          | 18.8 ± 12.3 | 100          | 12.8 ± 9.9  | 100          | -           |
|         | 1        | 100          | 104.6 ± 16.4| 100          | 47.0 ± 7.4  | 100          | -           |
|         | 2        | 100          | 303.3 ± 101 | 100          | 136.5 ± 45  | 100          | -           |
|         | 3        | 0            | 0 ± 0       | 0            | 0 ± 0       | 0            | -           |
| $A_2$   | 0        | 100          | 109.8 ± 6.2 | 100          | 49.4 ± 6.2  | 100          | 33.6 ± 10.6 |
|         | 1        | 100          | 230.1 ± 10  | 100          | 155.3 ± 35  | 100          | 105.6 ± 23  |
|         | 2        | 100          | 554.2 ± 47  | 100          | 249.4 ± 21  | 100          | 169.0 ± 14  |
|         | 3        | 0            | 0 ± 0       | 0            | 0 ± 0       | 0            | -           |
| $A_3$   | 0        | 100          | 83.3 ± 10.6 | 100          | 82.3 ± 10.6 | 100          | 56.0 ± 32   |
|         | 1        | 100          | 207.0 ± 26  | 100          | 207.0 ± 26  | 100          | 163.2 ± 31  |
|         | 2        | 100          | 810.4 ± 42  | 100          | 364.7 ± 28  | 100          | 248.0 ± 19  |
|         | 3        | 0            | 0 ± 0       | 0            | 0 ± 0       | 0            | -           |
of software defined radio platforms and standardized protocols (e.g. DLEP). Such developments will potentially increase the impact of the present investigation, allowing the improvement of our multi-layer control mechanisms using model $A/B$.

- The control mechanism introduced in this investigation was validated in a VHF network. Different communication technologies like UHF and Sat-Com, supporting higher data rates, will demand fine tuning of our control mechanism. However, model $A/B$ is a valid abstraction to define different control mechanisms acting in different pipes connected to different networks.

8 Conclusion

This investigation introduced two models to create ever-changing communication scenarios in tactical networks including the element of chance (randomness) from both user data flows (model $A$) and network conditions (model $B$). In sequence, we sketched a third model $A/B$ to define multi-layer control loops to handle the ever-changing conditions from model $A$ given the current network conditions from model $B$. These three models were developed to test the performance bounds of tactical systems and are reproducible for quantitative comparisons. We verified our three models doing experiments in a VHF network using real military radios. Model $A$ was used to create data flows with 500 kbps of random payload known to overflow the radio buffer (four times the radio buffer) over the best network conditions (radios with wired antennas). Model $B$ was used to create six different patterns of ever-changing network states with data rates varying from 0.6 to 9.6 kbps and also including link disconnections. We used model $A/B$ to define a combination of a reactive/proactive control to shape the user-generated data flow coping with current network conditions. Our hierarchy of queues was continuously monitoring the radio buffer occupancy to pause the transmission when a predefined threshold was crossed (reactive). At the same time, it proactively inserted an interval between IP packets which was computed using the link data rate (compiled by the routing protocol), buffer occupancy and latency. Thus, the feasibility of our control mechanism has been validated in communication scenarios with both message types and link data rates changing independently.

As future work, we plan to study the performance of reliable transport protocols and proactive routing protocols over ever-changing network conditions, including disconnections, and also delivering ever-changing QoS-constrained data flows. Further research is needed to extend model $A$ to include stream-based data flows like video and VoIP. We plan to use model $A/B$ to define different scheduling mechanisms like weighted fair share and active queuing that may have better results during experiments over faster communication technologies (e.g. UHF radios with up to 240 kbps of nominal data rate). Lastly, we also plan to use statistical learning algorithms, sitting at the control plane of model $A/B$, to compute/predict ever-changing communication scenarios from different layers.

Acknowledgments

This investigation was funded by BAAINBw (Federal Office of Bundeswehr Equipment, Information Technology and In-Service Support) and WTD 81 (Bundeswehr Technical Center for Information Technology and Electronics).

Parts of this investigation were published at DRCN 2019 [2], ICMCIS 2018-2019 [3], [19], and MILCOM 2019 [4].

References

[1] G. Elmasry, “A comparative review of commercial vs. tactical wireless networks,” IEEE Communications Magazine, vol. 48, no. 10, pp. 54–59, October 2010.

[2] R. R. F. Lopes, P. H. Balara, and P. Sevenich, “Creating and handling ever-changing communication scenarios in tactical networks,” in 15th International Conference on the Design of Reliable Communication Networks (DRCN), Coimbra, Portugal, March 2019, pp. 67–74.

[3] R. R. F. Lopes, P. H. Balara, and P. Sevenich, “Creating ever-changing QoS-constrained dataflows in tactical networks: An exploratory study,” in International Conference on Military Communications and Information Systems (ICMCIS), Budva, Montenegro, May 2019, pp. 1–8.

[4] R. R. F. Lopes, P. H. Balara, A. T. Silva, P. H. Rettore, and P. Sevenich, “Experiments with a queuing mechanism over ever-changing data rates in a VHF network,” in IEEE Military Communications Conference (MILCOM), Norfolk VA, USA, November 2019.

[5] N. Suri, M. R. Breedy, K. M. Marcus, R. Fronteddu, E. Cramer, A. Morelli, L. Campioni, M. Provosty, C. Enders, M. Tortonesi, and J. Nilsson, “Experimental evaluation of group communications protocols for data dissemination at the tactical edge,” in 2019 International Conference on Military Communications and Information Systems (ICMCIS), May 2019, pp. 1–8.

[6] 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 International Conference on Military Communications and Information Systems (ICMCIS), May 2017, pp. 1–8.

[7] J. J. Lindquist, F. T. Johnsen, and T. H. Bloebaum, “Proxy pair optimizations for increased service reliability in DIL networks,” in IEEE Military Communications Conference (MILCOM), Oct 2017.

[8] M. Małowski, P. Kaniwski, R. Matyszkiewicz, and P. Berezinski, “Standard tactical services in a military disruption-tolerant network: Field tests,” in IEEE Military Communications Conference (MILCOM), Oct 2017, pp. 63–68.

[9] N. Suri, J. Nilsson, A. Hansson, U. Sterner, K. Marcus, L. Misirlioglu, M. Hauge, M. Peuhkuri, B. Buchin, R. in’t Velt, and M. Breedy, “The angloval tactical military scenario and experimentation environment,” in International Conference on Military Communications and Information Systems (ICMCIS), May 2018, pp. 1–8.

[10] M. Manso, J. M. A. Calero, C. Barz, T. H. Bloebaum, K. Chan, N. Jansen, F. T. Johnsen, G. Markarian, P. P. Meiler, I. Owens, J. Sliva, and Q. Wang, “SOA and wireless mobile networks in the tactical domain: Results from experiments,” in IEEE Military Communications Conference (MILCOM), Oct 2013, pp. 593–598.

[11] R. Fronteddu, A. Morelli, M. Mantovani, B. Ordway, L. Campioni, N. Suri, and K. M. Marcus, “State estimation for tactical networks: Challenges and approaches,” in IEEE Military Communications Conference (MILCOM), Oct 2018, pp. 1042–1048.

[12] S. Rufieux, C. Gisler, J. Wagen, F. Buntschu, and G. Bovet, “TAKE — tactical ad-hoc network emulation,” in International Conference on Military Communications and Information Systems (ICMCIS), May 2018, pp. 1–8.

[13] 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 IEEE Military Communications Conference (MILCOM), Nov 2016, pp. 970-975.

[14] S. G. Pease, I. W. Phillips, and L. Guan, “Adaptive intelligent middleware architecture for mobile real-time communications,” IEEE Transactions on Mobile Computing, vol. 15, no. 3, pp. 572–585, March 2016.
[15] K. Scott, T. Refaei, N. Trivedi, J. Trinh, and J. P. Macker, “Robust communications for disconnected, intermittent, low-bandwidth (DIL) environments,” in IEEE Military Communications Conference (MILCOM), Nov 2011, pp. 1009–1014.

[16] A. Ghosh, S. 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 IEEE Military Communications Conference (MILCOM), Oct 2010, pp. 178–183.

[17] K. Lund, E. Skjervold, F. Johnsen, T. Hafsoe, and A. Eggen, “Robust web services in heterogeneous military networks,” IEEE Communications Magazine, vol. 48, no. 10, pp. 78–83, October 2010.

[18] 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 IEEE Military Communications Conference (MILCOM), 2011, pp. 1586–1591.

[19] 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 International Conference on Military Communications and Information Systems (ICMCIS), May 2018.

[20] A. Diefenbach, R. R. F. Lopes, T. A. Lampe, C. Prasse, J. Śliwa, R. Goniacz, and A. Viidanoja, “Realizing overlay xcast in a tactical service infrastructure: An approach based on a service-oriented architecture,” in International Conference on Military Communications and Information Systems (ICMCIS), May 2018, pp. 1–8.

[21] J. Hannay, “Architectural work for modeling and simulation combining the NATO architecture framework and C3 taxonomy,” vol. 14, pp. 139–158, 11 2016.

[22] J. Evans, B. Ewy, M. Swink, S. Pennington, D. Siquiers, and S. Earp, “TIGR: the tactical ground reporting system,” IEEE Communications Magazine, vol. 51, no. 10, pp. 42–49, October 2013.

[23] D. S. R. T. S. Ratliff, S. Jury and B. Berry, “Dynamic Link Exchange Protocol (DLEP),” Internet Requests for Comments, RFC Editor, RFC 8175, June 2017. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8175.txt

[24] K. Lund, A. Eggen, D. Hadzic, T. Hafsoe, and F. T. Johnsen, “Using web services to realize service oriented architecture in military communication networks,” IEEE Communications Magazine, vol. 45, no. 10, pp. 47–53, October 2007.

[25] 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 IEEE Military Communications Conference (MILCOM), Oct 2017, pp. 430–435.

[26] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, “Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges,” IEEE Communications Surveys Tutorials, vol. 14, no. 2. pp. 607–640, 2012.

[27] S. Jain and M. Chawla, “Survey of buffer management policies for delay tolerant networks,” The Journal of Engineering, vol. 2014, no. 3, pp. 117–123, 2014.

[28] P. J. T. Clausen, C. Dearlove and U. Herberg, “RFC 7181: The optimized link state routing protocol version 2,” Apr. 2014. [Online]. Available: https://tools.ietf.org/html/rfc7181