Coordinating the SDN and TDMA planes in software-defined tactical MANETs with adaptive coding and modulation

Yiannis Papageorgiou[‡‖], Johannes F. Loevenich[†§‖], Merkouris Karaliopoulos[‡],
Paulo H. L. Rettore[*], Roberto Rigolin F. Lopes[§], and Iordanis Koutsopoulos[‡]
[‡]Department of Informatics, Athens University of Economics and Business, Athens, Greece
Email: {gpapageorgiou, mkaralio,jordan}@aueb.gr
[†]Department of Mathematics/Computer Science, University of Osnabrück, Bonn, Germany
Email: jloevenich@uni-osnabrueck.de
[*]Department of Communication Systems, Fraunhofer FKIE, Bonn, Germany
Email: paulo.lopes.rettore@fkie.fraunhofer.de
[§]Secure Communications & Information (SIX), Thales Deutschland, Ditzingen, Germany
Email: {johannes.loevenich, roberto.rigolin}@thalesgroup.com
[‖]These authors contributed equally to this work

***Abstract*—In this paper, we investigate a tactical mobile ad hoc network (MANET) enhanced with software defined networking (SDN) functionality. Radio transmissions of network links are orchestrated using time division multiple access (TDMA) and are subject to adaptive modulation and coding (AMC) to address the dynamic variations in the quality of MANET links. A significant challenge within these networks is meeting the quality of service (QoS) demands of end-user data traffic while maintaining the SDN control plane's responsiveness to MANET topology alterations. To address this challenge, we propose a novel approach that concurrently determines the optimal placement of the SDN controller and the transmission scheduling of links in the MANET. The management of residual resources has a direct impact on both performance metrics. Building upon previous research, we present a more comprehensive model of the quality variations of tactical MANET links, elucidating how these variations dictate transmission modes over each link through software-defined radio (SDR) technology and influence the slot allocation process. We contrast our joint optimization algorithm with alternative techniques that manage these two network control mechanisms independently. Numerical evaluations suggest that our proposed algorithm achieves superior performance in comparison with the techniques considered.**

***Index Terms*—Tactical Networks, Mobility, Network Modeling, Scheduling, Controller Placement Problem**

I. Introduction

Mobile Adhoc Networks (MANETs) have become increasingly prevalent in military and emergency response settings where traditional network infrastructure is absent or insufficient. Due to their dynamic and unpredictable nature, managing these networks remains a complex task. The "softwarization" of resource management is the main response to this complexity. Software-Defined Radio (SDR) provides the means to flexibly adapt the Physical (PHY) and Media Access Layer (MAC) layer parameters [1], whereas Software Defined Networking (SDN) has emerged as a promising paradigm for enabling dynamic control and management of network resources. Different traffic management policies are orchestrated by SDN controller entities that manage the network nodes (switches) via message exchange over the control plane. The actual location of those controllers directly impacts the controller-to-switch communication delays and the overall responsiveness of the SDN layer.

At the same time, radio resources in tactical MANETs are often allocated with Time Division Multiple Access (TDMA) techniques, which exploit spatial reuse. This study focuses on in-band MANETs, in which control and data planes share common resources. Spatial reuse TDMA schemes enhance network capacity by reusing time slots while taking link interference constraints into account in order to satisfy data Quality of Service (QoS) requirements

In [2], we demonstrate that coordinating the network decisions about the SDN controller placement and TDMA scheduling tasks can effectively improve network performance in terms of SDN-responsiveness and data QoS. This evidence was obtained under simplistic assumptions about link quality variation, *i.e.*, two MANET nodes could either communicate at maximum possible link speed or not at all. In the present study, we depart from that work in explicitly accounting for the time-varying link qualities that result from node mobility and challenging radio propagation conditions in tactical MANET environments. In parallel, we assume the existence of SDR capabilities that foster adaptive coding and modulation and let match the link transmission rate to the current link quality.

Starting point for our experimental work is the generation of network graphs with links of fluctuating quality. We use both real and synthetic mobility traces and a custom link model to this end. We then introduce to these networks different mixes of constant-bit-rate (CBR) and rate-elastic data traffic that compete for the TDMA resources with the SDN control plane traffic. The treatment of data and SDN control traffic by the TDMA scheduling task introduces a tradeoff between the QoS secured for data traffic and the responsiveness achieved at the SDN control plane, which can be further accentuated or mitigated by the placement of the SDN Controller. We propose an adapted version of the algorithm presented in [2] to resolve this trade-off between SDN-responsiveness and data QoS under varying network conditions. Our experiments show that our joint approach outperforms alternative solutions that carry out separately the two network control tasks.

The remainder of this paper is organized as follows. Section II provides an overview of the recent literature related to our work. In Section III, we introduce our system model, followed by the formulation of the problem that the tactical network faces when jointly determining the TDMA schedule and the SDN controller placement in Section IV. We then present our proposed heuristic algorithm for solving this problem in Section V. Next, in Section VI, we provide and discuss numerical results, before conccluding the paper in Section VII.

II. Related Work

A. SDN controller placement and MAC scheduling

The Controller Placement Problem (CPP) is extensively treated in the scientific literature. Interested readers are referred to surveys such as [3] and [4] for almost exhaustive accounts of the related work. The CPP was first introduced in the context of Internet Service Provider (ISP) networks in [5], which identifies and first explores the fundamental trade-off between communication delay and overhead at the SDN level: as more SDN controllers are placed in the network, the average and worst-case communication delay between SDN entities (controller, switches) is reduced, whereas the volume of exchanged messages increases. More relevant to our work in this paper is the thread that seeks to solve the problem in wireless environments accounting for the variations of the link quality [6], [7]. Hence, in [6], the authors consider various link metrics, such as the Signal-to-Interference-Ratio (SIR) throughput based on network topology state to find the ideal location of controller. Similar approaches in [8] focus on improving the network reliability in presence of link volatility and radio interference, by reducing the number of re-transmitted packets.

In this context, research work has also looked into the way the CPP interacts with the scheduling function at the MAC layer in determining the responsiveness at the SDN layer. In [9], with cellular networks in mind, the authors assume a round-robin TDMA scheme for scheduling transmissions and factor an approximate value for the TDMA access delay in the decision for the controller location. In earlier work on MANETs employing TDMA with spatial reuse [10], we made an argument in favor of *SDN-aware TDMA scheduling*, showing that the SDN control traffic delay can be significantly reduced by explicitly accounting for the SDN controller location during the derivation of the TDMA frame. In [2], on the other hand, we showed how the coordination of TDMA scheduling with the SDN controller placement decisions can more efficiently trade the SDN layer responsiveness with the QoS provided to the MANET data traffic flows. In this work, we investigate this trade off in the light of SDR-enabled MANETs, with adaptive modulation and coding capabilities, and accordingly adapt our algorithmic solutions to best leverage these capabilities.

B. Link modeling in tactical MANETs

Recent literature has introduced various models for generating dynamic communication scenarios in tactical MANETs using stochastic or mobility-based link models. Stochastic models, for example [11], [12], aim to generate reproducible test scenarios that can be used to quantify the performance bounds of tactical networks. The use of probabilistic distributions is a flexible technique for configuring test communication scenarios [13], that are difficult to reproduce in real-world settings and allows testing the system performance before its deployment.

In contrast, mobility models generate mobility patterns for the nodes, track their geographical locations and compute the link states out of them. As far as the mobility models implement realistic mobility patterns of tactical network nodes, this approach has the potential to generate more realistic MANET communication scenarios [14]. stochastic models, It is also possible to map either stochastic link state models to mobility traces or the other way around. This approach provides high flexibility in testing tactical systems by enabling the switching between stochastic or mobility-based link models for evaluating system components [15].

III. System Model

A. The tactical MANET

Since the nodes move and the radio propagation conditions vary over time, the network topology is dynamic. At any point in time t we can model it with a directed network graph $G_t := (V, E_t)$, where $(u, v) \in E_t$ represents a link between nodes u and v at time t. Since the network links are directed, $(u, v) \in E_t$ does not necessarily imply that $(v, u) \in E_t$. We define a node v as an out-neighbor of node u and node u as an in-neighbor of node v if there is a directed link from node u to node v. The in(out)-neighborhood of node v is the set of all nodes that have a directed link to (from) node v. This neighborhood is denoted by $N_{i(o)}(v)$, where the subscript $i(o)$ stands for the in-neighborhood or out-neighborhood, respectively. Furthermore, a node v is a two-hop out-neighbor of node u (hence, node u is a two-hop in-neighbor of node v, if there exists at least one intermediate node k such that $(u, k), (k, v) \in E_t$ and no direct link (u, v) exists between them.

B. Link quality variation and adaptation

While communicating over wireless links, the tactical nodes move in various ways depending on their mission objectives.

Sometimes nodes engage in strictly coordinated group movements, such as convoy operations, while at other times, *e.g.*, in battlefield, nodes may move more unpredictably and pseudo-randomly. The tactical network monitors the quality of links over time and adapts accordingly the link transmission rate. Practically, this link adaptation process implies the existence of two mechanisms at the nodes' physical layer (PHY). First, there must be some link quality estimation process in place. For example, the two link endpoints could periodically compute and exchange some Channel State /Quality Indicator (CSI/CQI) value. Secondly, the PHY should be able to adapt its modulation and coding scheme according to this value to tune the transmission mode to the current link state.

Formally, the MANET links lie, at any point in time, in one of S link quality states, indexed from 0 to $S-1$. The link quality increases monotonically with the state index s and so does the transmission rate σ_s of the modulation and coding scheme chosen at state s, as shown in Table I

TABLE I: Link states based on modulation schemes

Link state index	Modulation & Coding scheme	Data rate (bps)
0	-	0
1	MCS1	σ_1
2	MCS2	σ_2
3	MCS3	σ_3
4	MCS4	σ_4
5	MCS5	σ_5

C. SDN plane

Each node $v \in V$ is equipped with software that transforms it into an SDN switch with local controller functionality. We also elect one single node as the Master SDN controller to coordinate the whole network. This controller communicates with all other switches under its control (associated switches), by periodically exchanging heartbeat messages with them. Additionally, it collects switch- and link-level statistics, such as bandwidth and failure probability, using Link Layer Discovery Protocol (LLDP) packets [16]. This way it acquires global knowledge about the network topology and dynamically updates two types of information tables at the switches. The routing table logs network paths toward other nodes, while one or more flow tables list packet forwarding rules for active network flows traversing the node.

Upon the arrival of a new flow, the switch first looks up its flow table(s) for a matching entry. If the switch finds a corresponding rule, it forwards the flow packets accordingly. However, if there is no matching entry, the controller uses its global network knowledge to compute a path for the flow and then installs forwarding rules at the switch that initiated the process, as well as, proactively, at other switches along the flow path. This mechanism of dynamic flow installation enables efficient use of network resources and globally efficient policy-aware routing of data traffic. To this end, PACKET-IN and PACKET-OUT packets are exchanged between the switches and the controller through SDN multihop control paths derived by native Wireless Mesh Network (WMN) routing protocols, such as the Optimized Link State Routing (OLSR) [17][1]. The routing of data traffic across the tactical multihop network is controlled by SDN, and the link transmissions are scheduled by a TDMA scheme.

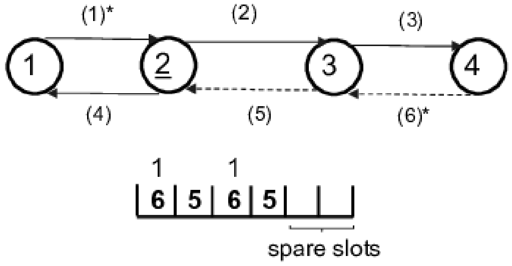


Fig. 1: Toy MANET example with 4 nodes and a single data traffic flow from node 4 to node 2 through node 3 at rate equivalent to two TDMA slots per frame. Links (5) and (6) on the flow path are assigned with two Transmission Opportunitys (TXOPs) each. For link (6), this involves the scheduling of MCLS $\{(1),(6)\}$ in slots 1 and 4 of the TDMA frame and results in two spare TXOPs for link (1). Two slots of the TDMA frame remain unused(spare slots).

D. TDMA Scheduling

The node transmissions over the network links are scheduled in collision-free manner over a fixed-length TDMA frame of N_s slots. If R_T is the TDMA physical rate and R_s the traffic rate corresponding to the periodic allocation of one slot per frame, then $N_s = R_T/R_s$ is the number of slots in the TDMA frame. This number also satisfies $N_s = T/T_s$, where T is the TDMA frame duration and T_s is the time slot duration.

The TDMA scheme we consider is centralized and employs spatial reuse: it schedules multiple links over each time slot as long as they are compatible with each other, *i.e.*, they can transmit simultaneously without interfering with each other. Necessary and sufficient conditions for a link set to be *compatible* are defined in [20]: the compatible link set of any link (u,v) should exclude (a) all links $(k,v), k \in N_i(v) \setminus u$ and links $(v,l), l \in N_o(v)$; (b) all links $(k,u), k \in N_i(u)$ and $(u,l), l \in N_o(u) \setminus v$; and (c) all links $(k,l), l \in N_o(u)$. We call a compatible set *maximal*, if no extra link can be added, without violating constraints (a)-(c)[2]. In [20], these conditions motivate a trivial greedy algorithm for the derivation of one, each time, Maximal Compatible Link Sets (MCLS). Since the network nodes move or the radio propagation conditions change or both, the set of MCLSs varies over time. Hereafter, we use $\mathcal{M}_t$ to denote the time-dependent set of MCLSs and $m \in \mathcal{M}_t$ to refer to one of them.

[1] The exact way that native MANET routing protocols and the SDN-layer co-exist and complement each other is an independent research thread, see *e.g.*, [18] [19]. Our assumption is that the SDN control messages are routed over paths determined by the native MANET routing protocol and the data traffic is routed by the SDN layer.

[2] Indeed, the actual interference between two or more network links is not a binary property (*i.e.*, interfere or not) and capturing it precisely demands costly online field measurements, see *e.g.*, [21]

E. Network traffic and TDMA slot allocation policy

1) Network traffic: The network delivers two types of traffic, data flows and SDN control-plane messages between the controller and the other nodes. Depending on their QoS requirements, the data flows are further distinguished into two categories: *(a)* inelastic flows (*e.g.*, voice or video streaming), which require certain data rate and cannot tolerate delays beyond some upper bound, and *(b)* elastic, *e.g.*, TCP-controlled, flows, which only require from the network a minimum amount of data rate and may acquire more on a best effort basis, depending on the residual capacity of the TDMA frame.

2) TDMA slot allocation policy for data traffic: Both types of traffic are delivered in-band, namely data flows and SDN control-plane messages compete for the same resources (TDMA slots) at the same frequency band. Each node maintains separate buffers for data and SDN control traffic at the MAC layer and keeps record of the slot allocations for each type of traffic.

We assume that the scheduler has accurate information about data flows. If F_t is the set of ongoing data flows at each point in time at the network, for each flow $f \in F_t$, the TDMA scheduler has access to the tuple $(s_f, d_f, r_f, p_f, t_f)$, where s_f and d_f are the flow source and destination nodes, respectively, r_f is the required rate (in bps), p_f is the $(s_f \to d_f)$ routing path computed for the flow by the SDN controller and t_f denotes the category of data flow, inelastic (t_f=0) or elastic (t_f=1). Indeed, such an assumption implies some coordination between the TDMA scheduler and the SDN controller. This cross-layer approach is indeed central in our paper and it is elaborated further in the subsequent sections.

In light of this assumption, the scheduler follows a two-level slot allocation policy. For an inelastic data flow f, the number of TDMA slots assigned to each link $l \in p_f$ depends on its current state and equals

$$r_l(f) = \lceil \frac{r_f}{\sigma_{s_l(t)}} \rceil. \tag{1}$$

For elastic flows, on the other hand, the scheduler allocates a fixed minimum rate r_{min}, or $\lceil \frac{r_{min}}{\sigma_{s_l(t)}} \rceil$ slots, to each link in the flow path. Hence, depending on the link state and the subsequent transmission mode of each link, the slots that need to be reserved for different links in the TDMA frame differ, even if they serve identical sets of flows. If there are *spare slots* in the TDMA frame after all inelastic flows are served and elastic flows are assigned their minimum rate, these *may* be distributed between elastic flows till the frame slots are exhausted.

3) TDMA slot allocation policy for SDN control traffic: SDN control messages exchanged between the SDN controller node and the other (switch) nodes also need to be served by the TDMA frame. This can happen in two ways. The first one relates to the use of TDMA spare slots. Since these can also be allocated to the elastic flows, their management introduces a tradeoff between the data traffic QoS (elastic traffic rate) and the responsiveness of the SDN control plane. We are looking closer into this tradeoff in the evaluation section VI. The second way is through transmission opportunities (TXOPs) that become available as part of MCLSs, that are mapped to TDMA slots, dedicated for the data traffic, as shown in Fig. 1. In contrast to data flows, control plane messages regard only statistics collection or flow rule installation, scaling at the magnitude of few Kbps and do not require more than one slot to transmit the full amount of packets per control flow [22].

IV. Problem formulation

We formulate the problem that is faced by the tactical network when it jointly determines the TDMA schedule and the SDN controller placement, to best respond to the conflicting requirements of both the user data traffic and the SDN messages. This formulation is an adaptation of a model first proposed in [2].

We introduce an integer decision variable $Y_m \in \{0, 1, ..., N_s\}$ that represents the number of times MCLS m appears in the TDMA frame. Clearly, the sum of all integer variable values $Y_m \in \mathcal{M}$ for all MCLSss should equal N_s, namely:

$$\sum_{m \in \mathcal{M}_t} Y_m = N_s. \tag{2}$$

In addition, we define binary variables x_{lm} and z_v to capture the combination of compatible links into MCLSs (Eq. 3)

$$x_{lm} = \begin{cases} 1 & \text{if } l \in m \\ 0 & \text{otherwise} \end{cases} \quad l \in E_t, m \in \mathcal{M}_t \tag{3}$$

and the placement of the controller at node v

$$z_v = \begin{cases} 1 & \text{if the controller is placed at } v \\ 0 & \text{otherwise} \end{cases} \quad v \in V_t \tag{4}$$

s.t.

$$\sum_{v \in V} z_v = 1 \tag{5}$$

respectively. The placement of the controller at node v results in a unique set of SDN control paths to other network nodes. To quantify the expected delay of messages sent from node v and received at switch u over the TDMA frame, we use the variable D^c_{vu}, which represents the expected one-way delay in frame slots. Since messages may be generated within any time slot of the TDMA frame with the same probability, we can apply the law of total probability to express the expected delay as follows:

$$D^c_{u,v} = \frac{1}{N_s} \sum_{n=1}^{N_s} D_{u,v}(n) \tag{6}$$

where $D_{u,v}(n)$ represents the end-to-end delay of packets generated during the n^{th} time slot.

To determine the amount of slots required for each network link l in the TDMA frame, we consider the set $\mathcal{F}$ of active data flows in the network. Specifically, we define r^d_l (Eq. 7) as the sum of two terms: the first term accounts for the slot requirements of inelastic flows $f \in \mathcal{F}$ that traverse link l, while the second term sums the slot requirements over elastic flows $f \in \mathcal{F}$ that use link l.

$$r_l^d = \sum_{\substack{f \in \mathcal{F}: l \in p_f \\ t_f = 0}} \lceil \frac{r_f}{\sigma_{s_l}} \rceil + \sum_{\substack{f \in \mathcal{F}: l \in p_f \\ t_f = 1}} \lceil \frac{r_{min}}{\sigma_{s_l}} \rceil \tag{7}$$

Note that the actual number of slots (*i.e.*, TXOPs) that are needed per link depends on the current link state and monotonically decreases with the link state index (see Table I and Equation 1). .

In each occurrence of a MCLS m in the TDMA frame, a TXOP of link l, as far as it is part of m, may be used for either control or data traffic. We use a binary variable q_{lmj} to indicate this, namely:

$$q_{lmj} = \begin{cases} 1 & \text{if link } l \text{ has a TXOP in the } j^{th} \text{ occurence of} \\ & \text{MCLS } m \text{ and it is reserved for data flows} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Then, the total number of TXOPs that are reserved for data flows at link l is given by:

$$n_l^d = \sum_{m \in \mathcal{M}} \sum_{j=1}^{Y_m} Y_m \cdot x_{lm} \cdot q_{lmj} \tag{9}$$

Here, x_{lm} is a binary variable indicating whether link l is included in MCLS m. Each link must be assigned at least one TXOP in the TDMA frame for control traffic, such as routing messages. Therefore, we have an additional constraint

$$n_l^c \geq 1 \tag{10}$$

for all links l in the tactical network.

The derivation of the TDMA schedule then tries to best compromise two conflicting objectives: the minimization of the controller-to-switch delays at the SDN layer and the maximization of the additional rate, beyond the reserved minimum of r_{min}, that can be allocated to the ongoing elastic flows in the network $\mathcal{F}_e = \{f \in \mathcal{F}, t_f = 1\}$. Thus, the optimization problem we seek to solve can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{Y}, \mathbf{z}, \mathbf{q}} \quad & w_1 \sum_{v \in V} z_v \frac{\sum_{u \in V \setminus v} D_{vu}^c}{|V| - 1} - w_2 \sum_{f \in \mathcal{F}_e} \min_{l \in p_f} ((n_l^d - r_l^d)\sigma_{s_l} \\ s.t. \quad & (2) - (9) \qquad\qquad (OPT) \end{aligned}$$

Here, w_1 and w_2 are weight parameters that allow us to prioritize one performance metric over the other in the objective function.

V. Proposed Algorithm

We propose an extended version of the algorithm in [2] to address the (OPT) problem in section IV. The algorithm is heuristic and may be executed either periodically or upon changes in the network topology or data traffic. It proceeds in five sequential steps:

1. Derivation of MCLSs for the current topology: Initially, the scheduler computes a set of maximal MCLSs for the given topology. While finding the complete set of maximal MCLSs is NP-complete, we can obtain a subset of such MCLSs by executing the greedy algorithm in [20] multiple times. The complexity of this step can be further reduced by disregarding links in the lowest states (*e.g.*, states 1 and 2 in Table I) and focusing on MCLSs involving only links at higher-index states. This way we target the search towards MCLSs that can yield high throughout across the network. Additionally, the link state remains consistent for each slot of the frame, given that the distance between each pair of nodes fluctuates minimally during the TDMA frame.

2. Computation of slot requirements for each link: The algorithm then processes the paths of data flows and the states of links to calculate the number of slots, r_l^d, that must be allocated to each link, in accordance with Equation (7).

3. Determination of the MCLSs to include in the TDMA frame: Next, the algorithm determines which MCLSs should be utilized in the TDMA frame and how many times each MCLS is used so that the slot requirements of all network links, as computed in the previous step and taking Equation 10 into account, are met. We cast this problem as an instance of the Set Multicover problem and solve it using the greedy algorithm in [23].

4. Distribution of TDMA spare slots: At this step, the algorithm can adopt various policies for allocating the spare TDMA capacity. The default approach is to assign the entire spare TDMA capacity to elastic data flows. Specifically, our algorithm iteratively computes additional MCLSs that provide one or more slots to those flows until the frame capacity is depleted. TXOPs designated for data traffic in each slot are marked. Alternative policies are compared with the default one in Section VI.A.

5. Placement of the SDN Controller: The algorithm, then, turns to the controller placement problem. First, it ranks the network links with respect to the number of spare TXOPs available to them at the end of the previous step. For link l, these equal $n_l^{sp} = \sum_{m \in \mathcal{M}} Y_m x_{lm} - r_l^d - 1$. Then, for each candidate controller location, it determines the resulting SDN control paths and computes a second ranking of the network links, this time according to the number of control paths traversing them. The objective is to place the controller where the resulting distribution of control traffic load over the MANET links, assumed proportional to the number of control paths traversing each of them, better *matches* the distribution of spare TXOPs over those links. The Kendall's tau (τ) rank correlation coefficient is employed for this purpose [24], *i.e.*, the controller is placed at the node that yields the highest τ value between the two rankings.

6. Ordering of MCLSs in the TDMA frame: End-to-end delays experienced by control messages over the network paths can be further reduced when MCLSs succeed each other in the TDMA frame in the same order that links succeed each other on the control paths. We sketch an algorithm realizing

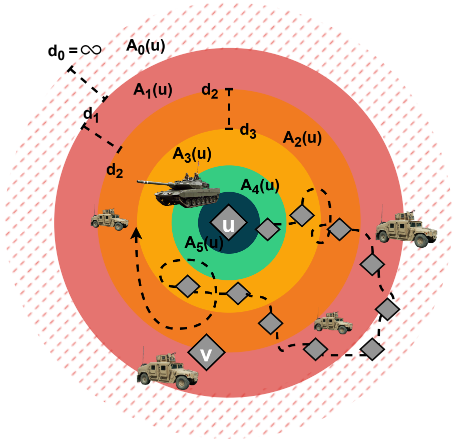


Fig. 2: Link mapping for 6 different modulations.

this in [2].

The provision for adaptive modulation and coding changes the algorithm in two steps, when compared to the original algorithm in [2]: in step 2, where it considers the current link state and the corresponding modulation and coding scheme to compute how many slots are needed for given traffic flow rate; and, in step 3, where MCLSs are not simply chosen according to how many links they can simultaneously serve but rather according to the cumulative throughput transmission rate they support.

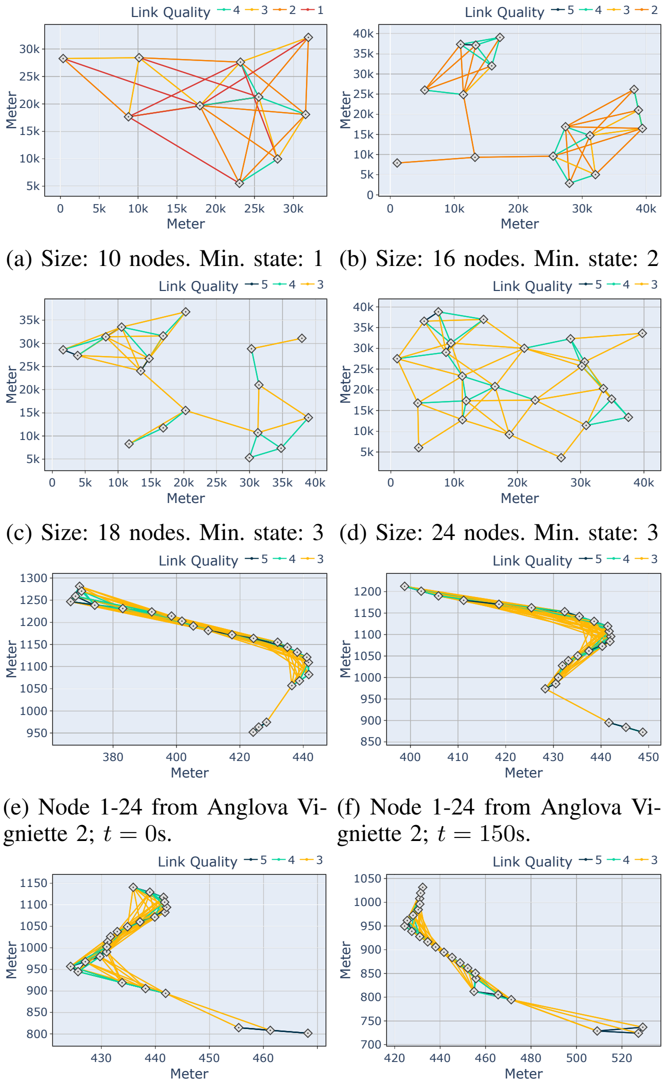
VI. Numerical results

A. *Tactical MANET snapshots*

We develop graph models to capture the dynamic topology of tactical MANETs. Our graph model incorporates various configurable parameters, including the size of the mission area, the number of network nodes, and the velocity intervals for low, medium, and high mobility. The output of our model is a weighted directed graph, where edge weights represent the transmission rates of the corresponding network links.

Fig. 3 shows exemplary outputs of our graph model. Four graphs (a)-(d) are generated as 10-24 nodes move in a 40km × 40km mission area under the Random Waypoint (a), Group Mobility (b), and Gauss-Markov mobility models (c), (d) with speeds uniformly distributed in intervals [(0.1, 1.5), (4, 10), (10, 25)] in m/sec for low, medium, high mobility, respectively. Graphs (e)-(h), on the other hand, track the topology changes due to the movement of the Company 1 nodes 1-24 in the Anglova Vignette 2 from $t = 0s$ to $t = 300s$.

To determine the state of link (u, v) at each time t, we partition the area around node u into S annular areas denoted by $A_0(u), \ldots, A_{S-1}(u)$, as illustrated in Fig. 2. In this figure, each annular area $A_s(u)$ is bounded by the circles of radii d_{s-1} and d_s, with $d_s > d_j$ for $s < j$, $d_S = 0$, d_1 equal to the radio transmission range of node u, and $d_0 = \infty$. Hence, A_{S-1} corresponds to a circular area of radius d_{S-1} around node u and $A_0(u)$ denotes the area beyond the radio communication coverage of u. Then, a link between node u and another node v is at state $s, 0 \leq s \leq S-1$, when node v lies within the annular area $A_s(u)$, *i.e.*, when their Euclidean distance falls within the interval $[d_s, d_{s+1}]$.



(a) Size: 10 nodes. Min. state: 1

(b) Size: 16 nodes. Min. state: 2

(c) Size: 18 nodes. Min. state: 3

(d) Size: 24 nodes. Min. state: 3

(e) Node 1-24 from Anglova Vignette 2; $t = 0$s.

(f) Node 1-24 from Anglova Vignette 2; $t = 150$s.

(g) Node 1-24 from Anglova Vignette 2; $t = 300$s.

(h) Node 1-24 from Anglova Vignette 2; $t = 500$s.

Fig. 3: Exemplary outputs of graph model $\mathcal{G}$.

The different link states are indicated with colors ranging from blue (state 5) to red (state 1) in Fig. 2. These states are mapped to the radii that bound the respective geographical areas as follows. We compute the histogram of pairwise distances between the MANET nodes and map the four radii $d_1, \ldots, d_5$ to percentiles of this distribution. Hence, d_5 is mapped to the 10^{th} percentile, d_4 to the 25^{th} percentile, d_3 to the 45^{th} percentile, d_2 to the 60^{th} percentile and d_1 is mapped to the 75^{th} percentile, implying that 25% of the node pairs are outside the radio coverage of each other. We used this strategy for mapping the system states, because it presents a systematic/controlled way to map distances to plausible link states in the absence of real field measurements or precise radio propagation and PHY models in emulators

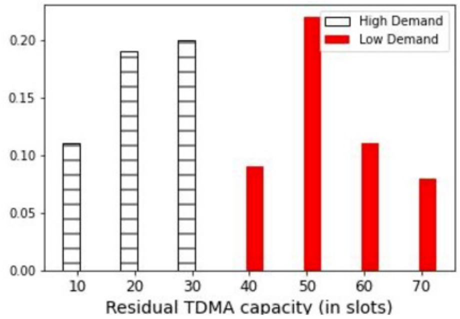
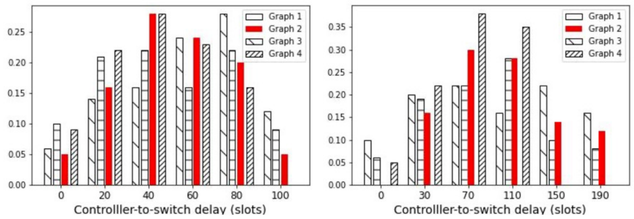
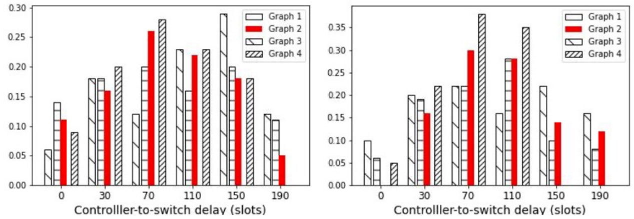


Fig. 4: Distribution of unused slots across the TDMA frames under low and heavy demand over the network snapshots



(a) moderate demand (b) high demand

Fig. 5: Mean CTR-SW delay differences between our algorithm and the TDMA-unaware controller placement with data traffic prioritization($RES_d = 1$)



(a) moderate demand (b) high demand

Fig. 6: Worst CTR-SW delay differences between our algorithm and the TDMA-unaware controller placement with data traffic prioritization ($RES_d = 1$)

like CORE/EMANE. Moreover, we varied the minimum acceptable link state δ, which denotes connection status, from 1 to 3, see Fig. 3(a)-(d).

B. Evaluation Methodology

1) Experimentation setup: We evaluate our proposed algorithm for solving (OPT) in the MATLAB environment. We experiment with the topology snapshots of the 4 directed weighted graphs in Fig.3e-3h. To each of these graphs, we inject 400 different mixes of data traffic flows, each comprising inelastic and elastic flows at a 80%-20% split. We randomly select the source and destination nodes of the flows and their required rates (in Kbps) are uniformly sampled from $\{20kbps, 40kbps, 60kbps, 80kbps\}$. The corresponding flow paths are computed as shortest paths over the topology.

Our TDMA frame consists of 150 slots. Assignment of one slot in the frame corresponds to data rate of 20 kbps under link state $s_l = 5$. The other four (s_l, σ) pairs are $(4, 10kbps)$, $(3, 6.6kbps)$, $(2, 5kbps)$, and $(1, 4kbps)$. Regarding the spare TDMA capacity policy (see section V), we determine a controllable parameter $RES_d \in [0, 1]$, which represents the percentage of the spare TDMA capacity that is distributed to elastic data flows. Finally, in our experiments we distinguish between high and low traffic load scenarios, depending on the amount of spare slots that remain available after reservations are made for the data traffic flows, as shown in Fig. 4.

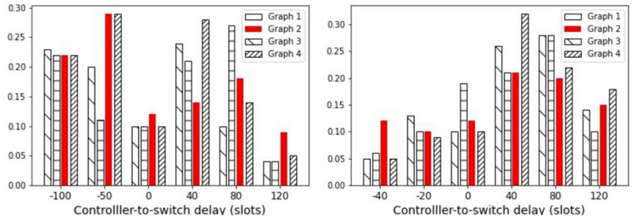
2) Performance metrics: We emphasize two Key Performance Indicators (KPIs) when evaluating the effectiveness of our algorithm across different network snapshots. The first one is the communication delay between the single controller and the 23 switches over the SDN control paths, measured in time slots. We record both the mean and the worst-case value of the controller-to-switch delay. The second KPI is the sum of (beyond-the-minimum) rates allocated to elastic flows in the TDMA frame, measured in Kbps.

C. Comparison alternatives

We compare our approach in section V against two existing methods: one that places the SDN controller at the most central node using an SDN-unaware TDMA scheduling policy, and another that uses an SDN-aware TDMA scheduling policy. The former allocates slots for user data traffic and distributes the full amount of residual capacity among elastic flows with $RES_d = 1$, while the latter allocates slots for both data traffic and control flows and distributes the residual capacity among the control paths. The two policies represent extremes in the distribution of residual capacity Intuitively, the effect of the compared methods on both KPIs is scaled by the amount of spare slots. Yet, assigning the residual capacity to data flows does not prevent control paths from acquiring additional resources; instead, they acquire additional spare TXOPs in MCLSs used to deliver bonus rates for elastic flows.

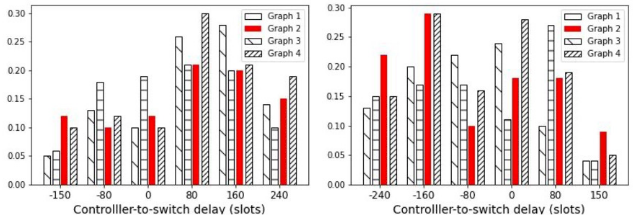
D. Evaluation results

1) Comparison with SDN-unaware TDMA scheduling ($RES_d = 0$): The first set of experiments demonstrates that our algorithm is capable of achieving lower CTR-SW delays in both average and worst-case terms, under varying levels of traffic demand. As depicted in Figs. 5 and 6, our joint approach consistently outperforms the alternative, which considers control functions separately. These figures plot histograms of delay differences between the two approaches, with positive differences indicating higher delays with the SDN-unaware TDMA scheduling. Notably, both approaches adopt the same TDMA scheduling policy, and distribute the same amount of residual capacity to elastic data flows. Our joint approach leverages spare TXOPs to serve control traffic and prioritize their order, aligned with control paths, over the TDMA frame.



(a) moderate demand (b) high demand

Fig. 7: Mean CTR-SW delay differences between our algorithm and the SDN-aware TDMA scheduling policy with control traffic prioritization($RES_d = 0$)



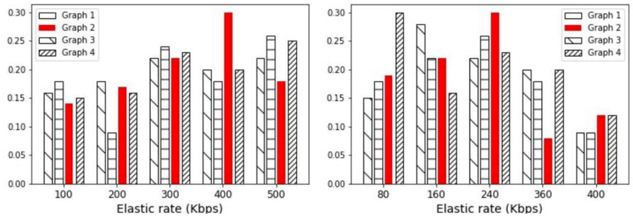
(a) moderate demand (b) high demand

Fig. 8: Worst CTR-SW delay differences between our algorithm and the SDN-aware TDMA scheduling policy with control traffic prioritization($RES_d = 0$)

This effectively reduces the mean delay up to 100% of the frame duration, and the worst-case delay by 200%.

Furthermore, we observe that a non-negligible percentage of cases result in zero delay differences between the two approaches, indicating that they place the SDN controller at the same position and hence do not differ in terms of delay. However, in approximately 28% of cases across all graphs and demand scenarios, the worst-case delay exceeds three-quarters of the frame duration. This highlights the significance of our joint approach in reducing CTR-SW delays, particularly in high-traffic scenarios where delay can have a significant impact on system performance.

2) Comparison with SDN-aware TDMA scheduling($RES_d = 1$): The second group of experiments includes the comparison of our algorithm to scheduling TDMA slots accounting for the controller location. The histograms depicted in Figs. 7, 8, 9, point out the trade-off between SDN-responsiveness(CTR-SW delay) and data QoS (additional elastic rate). Specifically, in Figs. 8b, 7b, 9b, we observe that our algorithm outperforms the other policy at both metrics, in the case of high demand scenario. Under limited residual frame capacity, up to 30 slots, allocating extra slots to control paths does not suffice to achieve comparable delays with those our algorithm yields. For a certain percentage of cases, overall graphs and data traffic mixes up to 30-37%, the difference is negative, meaning that our algorithm augments the delay (mean and maximum) to 80 slots on average, see the histograms of Figs. 7,8. In the general case, our algorithm achieves smaller delays: the mean delay difference is approximately in the order of half frame and the worst-case delay scales even to one frame. Nevertheless the additional rate, dedicated to elastic flows, climbs to 400 Kbps, rapidly increasing the data QoS provided by our algorithm.



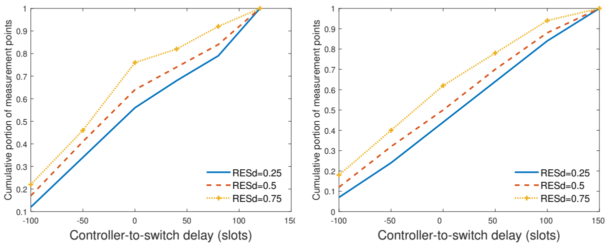
(a) moderate demand (b) high demand

Fig. 9: Elastic data traffic rate difference between our algorithm and the SDN-aware TDMA scheduling policy with control traffic prioritization($RES_d = 0$)

The increased residual capacity, under the moderate demand scenario, seems to act as a gamechanger to the balance of the trade-off between the two main metrics. As depicted in Figs.8a, 7a, our algorithm is less efficient in terms of delay, both mean and worst, but blasts the additional rate difference to higher levels, reaching even to 400 or even 500 Kbps as shown in Fig. 9a for moderate demand scenario. Furthermore, the states of elastic flow path links influences the total additional rate acquired. Specifically, we expect elastic flows served through links with $s_l = 3$ to provide less additional rate. Consequently, the share of the proportion of residual capacity, *the added TDMA awareness in the TDMA frame*, affects the resulting measurements.

3) Comparison with SDN-semiaware TDMA scheduling ($RES_d \in (0, 1)$): With the final group of experiments, we explore several ways to distribute the residual capacity, other than the two extremes described in previous experiments. We plot the empirical cumulative distribution of the delays and the additional elastic rate allocated overall graphs and data mixes, for each of the demand scenarios, letting RES_d assume values in $\{0.25, 0.5, 0.75\}$. Notably, the distribution of residual capacity to control and data traffic, has a compensatory effect on both KPIs.

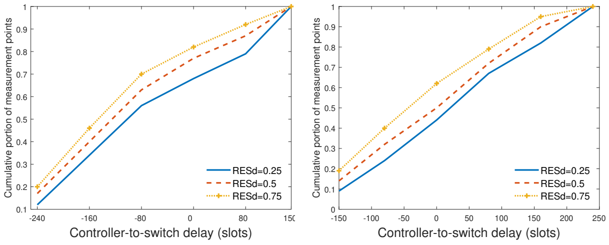
Specifically, in Figs.10b,11b,12b we observe that in high demand scenarios, the resulting differences among the scheduling policies are small since the residual resources are limited. The average difference between the policies $RES_d = 0.25, RES_d = 0.75$ in terms of mean delay is 9%, 12% for worst case and 10% for additional allocated rate to elastic flows. Yet, more significant differences are noted, when more residual resources remain unallocated. The trade-off between delay and additional rate extends its gap, up to 22%, 25 % in terms of delay (mean and worst) and 22% of additional data rate. Each cell of Table II contains the total percentage gains or loss our algorithm achieves against the alternative policies. We observe a non-linear correlation between adding SDN-awareness to TDMA schedule on both KPIs. Specifically,



(a) moderate demand

(b) high demand

Fig. 10: Mean CTR-SW delay differences between our algorithm and the SDN-aware TDMA scheduling policy for various values of RES_d

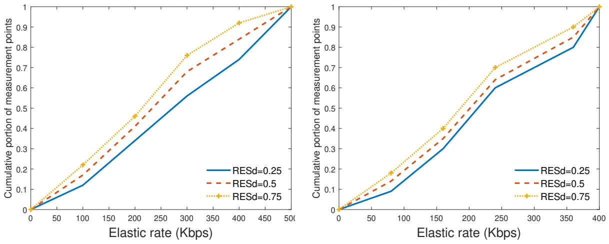


(a) moderate demand

(b) high demand

Fig. 11: Worst CTR-SW delay differences between our algorithm and the SDN-semiaware TDMA scheduling policy for various values of RES_d

doubling the SDN-awareness (from by 50% to by 100%), the delay loss gap is furthered by 5% (27% → 32%), yet the bonus rate achieved are increased by 27% (31% → 48%) in the case of moderate demand.



(a) moderate demand

(b) high demand

Fig. 12: Elastic data traffic rate difference between our algorithm and the SDN-aware TDMA scheduling policy with control traffic prioritization for various values of RES_d

TABLE II: Proposed algorithm against TDMA policies of various awareness levels in terms of delay and bonus rate

SDN-(%) aware TDMA scheduling	Delay		Bonus rate	
	moderate demand	high demand	moderate demand	high demand
0%	22%↓	28%↓	-	-
50%	27%↑	11%↓	31% ↑	17%↑
100%	32%↑	9%↓	48%↑	28%↑

E. Discussion and limitations

The proposed algorithm is sensitive to topology and user data traffic flows traversing the MANET. Hence, whenever the topology or traffic matrix of the network changes, both the optimal location of the SDN Controller and TDMA frame may change. In previous work [10], we explored the perpective of relaxing the frequency of the SDN controller relocation, by deriving a smarter *SDN-aware* TDMA frame, that could help reduce the delay over the control plane. The algorithm is centralized in the sense that control plane coordination, and TDMA frame derivation are performed by the single SDN controller and the TDMA scheduler. There is direct communication between those two. Specifically, the SDN Controller possesses and updates the necessary information needed by the TDMA scheduler. In that way, TDMA scheduler is not responsible for collecting network state information (NSI) and significant overhead is saved.

The amount of total network overhead grows significantly, as the size of MANET scales. When scaling the MANET size, one SDN controller is not capable of coordinating the network efficiently. Hence, we need multiple SDN controllers at the price of an additional synchronisation overhead. The multiple CPP introduces a tradeoff between the Ctr-Sw delays and the overall SDN control overhead. The induced overhead raises significantly, due to the required synchronization overhead among controllers, while the Ctr-Sw overhead remains in the order of few Kbps per flow, as shown in [25], [26]. In our case of study, we place one single controller, thus, the amount of signaling information and processing load do not prevent the frequent SDN controller relocation at the time scale of network state changes.

An interesting direction for investigation would be to devise the distributed version of such algorithm, which would reduce the amount of coordination between nodes and would be easier to implement in real time. The dominant challenge is to efficiently adapt with the setting of heterogeneously delayed Network State Information(NSI), because of the distributed nature of the network.

VII. Conclusion

In this study, we compared joint and separated controller placement and TDMA scheduling approaches in SDN enabled tactical MANETs with adaptive coding and modulation. We evaluated our approach against two existing methods and showed that our joint approach achieved lower CTR-SW delays, both in average and worst case scenarios, compared to the separate consideration of the two control functions.

Future investigations would include the expansion of the problem of placing multiple controllers on the network, which brings more challenges in the foresight. Firstly, the deployment of multiple controllers requires their frequent message exchange to synchronize, which results to significant overhead. Secondly, the association of switches with the most appropriate controller to minimize CTR-SW delay with the minimum overhead cost. And thirdly, the optimal number of controllers and associations to minimize total CTR-SW delay considering overhead constraints.

Acknowledgment

This work has been carried out in the context of the project entitled Software defined Mobile Tactical Ad hoc NeTwork (SMOTANET), which has received funding from the European Defence Industrial Development Programme (EDIDP) under grant agreement No EDIDP-CSAMN-SDN-2019-038-SMOTANET. This paper reflects only the authors' views and the European Commission is not responsible for any use that may be made of the information contained herein.

The authors would like to thank Bundeswehr (BAAINBw and WTD81), for the several national and international grants supporting the development of the ideas converging to the present paper.

References

[1] D. Kafetzis, S. Vassilaras, G. Vardoulias, and I. Koutsopoulos, "Software-defined networking meets software-defined radio in mobile ad hoc networks: State of the art and future directions," *IEEE Access*, vol. 10, pp. 9989–10 014, 2022.

[2] Y. Papageorgiou, M. Karaliopoulos, and I. Koutsopoulos, "Controller Placement and TDMA Link Scheduling in Software Defines Wireless Multihop Networks," in *Proc. IEEE International Conference on Communications (ICC) (to be published)*, Rome, Italy, 28 May.-1 Jun. 2023. [Online]. Available: https://mm.aueb.gr/publications/9c432f11-6d40-4930-8c74-90175bdea7d7.pdf

[3] T. Das, V. Sridharan, and M. Gurusamy, "A survey on controller placement in sdn," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 472–503, 2020.

[4] B. Isong, R. R. S. Molose, A. M. Abu-Mahfouz, and N. Dladlu, "Comprehensive Review of SDN Controller Placement Strategies," *IEEE Access*, vol. 8, pp. 170 070–170 092, 2020.

[5] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop on Hot Topics in Software Defined Networks*, ser. ACM HotSDN '12, Helsinki, Finland, 2012, p. 7–12.

[6] A. Dvir, Y. Haddad, and A. Zilberman, "Wireless controller placement problem," in *Proc. 15th IEEE CCNC*, 2018, pp. 1–4.

[7] K. Poularakis *et al.*, "Flexible sdn control in tactical ad hoc networks," *Ad Hoc Networks*, vol. 85, pp. 71–80, 2019.

[8] Q. Zhong, Y. Wang, W. Li, and X. Qiu, "A min-cover based controller placement approach to build reliable control network in sdn," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 481–487.

[9] M. J. Abdel-Rahman *et al.*, "On stochastic controller placement in software-defined wireless networks," in *Proc. IEEE WCNC*, San Francisco, CA, 2017, pp. 1–6.

[10] Y. Papageorgiou, M. Karaliopoulos, and I. Koutsopoulos, "Joint Controller Placement and TDMA Link Scheduling in SDN-enabled Tactical MANETs," in *Proc. IEEE Military Communications Conference (MILCOM)*, Rockville, MD, USA, 28 Nov.-1 Dec. 2022, pp. 125–132.

[11] 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. 188 015–188 035, Sept. 2020.

[12] Rukaiya, S. A. Khan, M. U. Farooq, M. A. K. Khattak, I. Matloob, and I. Nosheen, "A stochastic distribution based methodology to estimate control phase time for software-defined radios in tactical manets," *IEEE Access*, vol. 9, pp. 71 687–71 698, 2021.

[13] J. F. Loevenich, P. H. Rettore, R. R. F. Lopes, and A. Sergeev, "A bayesian inference model for dynamic neighbor discovery in tactical networks," *Procedia Computer Science*, vol. 205, pp. 28–38, 2022, 2022 International Conference on Military Communication and Information Systems (ICMCIS).

[14] S. Kaviani, B. Ryu, E. Ahmed, K. A. Larson, A. Le, A. Yahja, and J. H. Kim, "Robust and scalable routing with multi-agent deep reinforcement learning for manets," 2021.

[15] P. H. L. Rettore, J. Loevenich, and R. R. F. Lopes, "Tnt: A tactical network test platform to evaluate military systems over ever-changing scenarios," *IEEE Access*, vol. 10, pp. 100 939–100 954, 2022.

[16] L. Liao and V. C. M. Leung, "LLDP based link latency monitoring in software defined networks," in *12th International Conference on Network and Service Management (CNSM)*, Montreal, Quebec, Canada, Oct. 31 - Nov. 4 2016, pp. 330–335.

[17] T. H. Clausen, C. Dearlove, P. Jacquet, and U. Herberg, "RFC7181: The Optimized Link State Routing Protocol version 2," *The Internet Engineering Task Force*, April 2014.

[18] A. Detti, C. Pisa, S. Salsano, and N. Blefari-Melazzi, "Wireless Mesh Software Defined Networks (wmSDN)," in *Proc. IEEE 9th Int'l Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2013, pp. 89–95.

[19] H. C. Yu, G. Quer, and R. R. Rao, "Wireless SDN mobile ad hoc network: From theory to practice," in *Proc. IEEE Int'l Conference on Communications (ICC)*, 2017, pp. 1–7.

[20] I. Cidon and M. Sidi, "Distributed assignment algorithms for multihop packet radio networks," *IEEE Transactions on Computers*, vol. 38, no. 10, pp. 1353–1361, 1989.

[21] J. Padhye *et al.*, "Estimation of link interference in static multi-hop wireless networks," in *Proc. 5th ACM SIGCOMM IMC*, Berkeley, CA, 2005, pp. 28–33.

[22] K. Choumas, D. Giatsios, P. Flegkas, and T. Korakis, "The sdn control plane challenge for minimum control traffic: Distributed or centralized?" in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2019, pp. 1–7.

[23] V. V. Vazirani, *Approximation algorithms*. Springer, 2001.

[24] M. G. Kendall, "The treatment of ties in ranking problems," *Biometrika*, vol. 33, no. 3, pp. 239–251, 1945.

[25] K. Poularakis, Q. Qin, L. Ma, S. Kompella, K. K. Leung, and L. Tassiulas, "Learning the optimal synchronization rates in distributed sdn control architectures," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 1099–1107.

[26] D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, "Logically centralized? state distribution trade-offs in software defined networks," *HotSDN'12 - Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks*, 08 2012.