Joint Controller Placement and TDMA Link Scheduling in SDN-enabled tactical MANETs

Yiannis Papageorgiou, Merkouris Karaliopoulos, and Iordanis Koutsopoulos

Department of Informatics, Athens University of Economics and Business, Athens, Greece

***Abstract*—In this paper, we study the interplay of TDMA link scheduling and SDN Controller placement in SDN-enabled tactical mobile ad hoc networks (MANETs). The network needs to serve both user data flows, each with a (source, destination) pair, and the SDN control-plane messages exchanged between the Controller(s) and the rest of the nodes over SDN control paths. The delay suffered by those messages has to be minimized to help the SDN layer respond to the dynamic topological and traffic demand changes of those networks. This delay is affected by two network control functions: the placement of the SDN Controller(s), which determines the control traffic paths from the switches to the controller(s), and the TDMA scheduling of control path links. In this paper, we experiment with various Controller placement update policies and SDN-aware TDMA scheduling mechanisms and assess their impact on the average and worst-case communication delays over SDN control paths. We find that adding SDN awareness into TDMA scheduling can relax the need for frequent updates of the SDN Controller placement, thus achieving comparable delays with an ideal policy that updates the placement every time something changes in the network topology or user data traffic.**

***Index Terms*—Controller placement problem, Time division multiple access(TDMA), Cross-layer optimization, SDN-enabled tactical MANETs**

I. INTRODUCTION

Military communication systems become increasingly complex as the tactical operations they support become themselves more sophisticated, mobilizing many different agents (humans, manned and unmanned vehicles) and coordinating several tasks simultaneously (situational awareness, secure troop deployment, battle command). More often than not, these operations are carried out in areas without infrastructure and rely on mobile ad hoc networks (MANETs). One aspect of this complexity relates to the rich information that needs to be shared across different actors in the tactical network, including images, video and large files. High traffic demand scenarios favor collision-free Time Division Multiple Access (TDMA) schemes with spatial reuse [1] [2] over random access protocols such as Carrier Sense Multiple Access (CSMA) that need to invest significant resources to resolve collisions due to interference among neighboring nodes [3]. In scientific literature, several centralized and distributed algorithms have been proposed already in the 1980s for collision-free TDMA scheduling of node and link transmissions over multihop networks (see *e.g.*, [4] [5] [6]). In tactical networking, Link 16 has probably been the most widespread instance of a TDMA network, but it is designed for static network configurations with few nodes [7].

Moreover, modern military missions tend to involve coalitions of multiple groups, which operate under different command centers and may use in-field heterogeneous networking and computation devices. Defence agencies worldwide run research programs that focus on Software Defined Networking (SDN) as a solution for this challenge [8] [9]. SDN represents a paradigm shift in routing data, separating the data-plane (forwarding) tasks from the control-plane (routing intelligence) tasks. The routing data-plane resides locally at the network elements and can be programmed by a central entity (the SDN Controller) that implements the control-plane in software and runs on general purpose hardware. This functionality split makes SDN flexible in defining control policies that are highly relevant for military operations.

However, tactical MANETs are not an easy match for SDN. The critical performance metric here is how fast the single (or multiple) SDN Controller(s) can communicate with the rest of the network nodes. These delays determine how fast the Controller(s) can collect up-to-date information about the network state and how much time a new flow has to wait till the Controller computes a path for it and updates the forwarding rules at the nodes involved in this path. One network control mechanism that affects these delays is the SDN controller placement, *i.e.*, at which network node(s) do we place the SDN Controller(s) and how are the remaining nodes associated with it (them). The second network control function with an impact on those delays is the way links are scheduled for transmission at the TDMA level. Depending on how many slots are assigned to the SDN control-plane messages and the order of those slots in the frame, the SDN control-plane delays may be mitigated or exacerbated.

As the tactical network topology changes and data traffic flows come and go, the TDMA schedule also changes and so does the optimal SDN Controller placement. Yet, adapting the Controller placement too often incurs significant costs, in particular when the SDN control plane operates “in-band”, as the case typically is. Each time this happens, the tactical network needs to reserve time slots in TDMA links, depriving them from user data traffic, for messages needed to elect new optimal locations for the Controller(s) and establish new associations between them and the other nodes. Practically, the SDN Controller placement process cannot be executed at the time-scale of network topology changes. Ideally, we would like this to happen as rarely as possible. The question we pose in this paper is whether we can achieve this, *i.e.*,

978-1-6654-8534-0/22/$31.00

maintain communication delays at acceptable levels while reducing the frequency of SDN Controller placement updates, by making the TDMA scheduling SDN-aware. Intuitively, we ask *how much could we boost SDN by performing the TDMA scheduling in an SDN-aware manner*.

To address these questions, we set off to study the interplay between TDMA link scheduling and SDN Controller placement in SDN-enabled MANETs. The co-existence of SDN with legacy routing protocols in MANETs (*e.g.*, OLSRv2 [10]) has been explored in literature, see [11] [12] [13] [14]. In those studies, the main question is how to best share the routing responsibility between SDN and native MANET protocols and avoid duplication of effort (and message overhead). On the other hand, the SDN Controller placement problem has been studied in various network settings [15], including a cellular network with a TDMA link between the Controller(s) and its controlled nodes in [16]. Yet, to the best of our knowledge, there is no prior study of the SDN Controller placement problem in TDMA-scheduled MANETs.

To keep things simpler, we consider a tactical network with a single SDN controller. The network carries user traffic flows generated by the tactical network nodes, each from a given source to a given destination. The routes of these flows are determined by the SDN Controller. Besides user traffic flows, the network carries SDN control-plane messages exchanged between the Controller and each network node (switch, in SDN terminology) that lies on a user traffic flow path. The paths for these control messages, called SDN control paths, are typically shortest paths, derived by the native MANET routing protocol.

Our main contribution to the literature is a study of how decisions about the SDN Controller placement and the scheduling of TDMA links jointly affect the communication delays between the SDN Controller and the other network nodes. On the one hand, we vary the frequency at which the SDN Controller placement is updated from zero (unchanged placement), to a intermediate value (periodic revision), up to the frequency of TDMA schedule revision. This last case involves the statement of the joint problem of controller placement and TDMA link scheduling and its joint solution with an iterative algorithm. On the other hand, we specify and assess three policies that assume SDN-awareness at the MAC level and render the TDMA scheduler friendlier to the SDN control traffic. The policies concern either how many TDMA slots are eventually allocated to SDN control flows, or the way link transmission opportunities are organized along the TDMA frame.

Our experimentation suggests that SDN-aware TDMA scheduling can have a significant compensatory effect upon the SDN control path delays: it can reduce them to the levels that would be achievable under a very aggressive SDN Controller update policy that would try to adapt at each change of the tactical network topology or traffic workload.

The rest of the paper is organized as follows. In section II, we outline the system model, insisting on the SDN-related and TDMA scheduling operations. In section III we present SDN and TDMA scheduling policies affecting the communication delays in the SDN control paths. We asses those policies in section IV and conclude the paper in section V.

II. SYSTEM MODEL

A. Tactical ad hoc network

We consider a tactical network comprising several types of mechanized nodes such as tanks, armored vehicles etc. The nodes may be involved in various tactical operations such as the troop deployment phase or the preparation for battle. Each operation induces different patterns in the mobility of nodes and the communication between them.

Let $G_t = (V, E_t)$ denote the digraph representing the network these nodes give rise to at time t. We write (u, v) to denote a link between nodes u and v, where u is the transmitting node and v the receiving one. Links are unidirectional: $(u, v) \in E_t$ does not necessarily imply that $(v, u) \in E_t$. The set of network links at time t, E_t, depends on the relative locations of the $|V|$ nodes and the radio propagation conditions across the area of the tactical operations. A node v is called an out-neighbor of node u and node u is called an in-neighbor of node v at time t if $(u, v) \in E_t$. The set of in(out)-neighbors of node v denotes its in(out)-neighborhood $N_{i(o)}(v)$. Likewise, node $v(u)$ is a two-hop in(out)-neighbor of node $u(v)$, if there is no direct link (u, v) between them, but there exists at least one node k so that $(u, k), (k, v) \in E_t$.

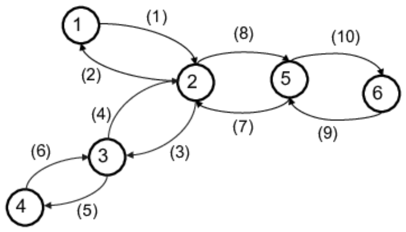
The routing of data traffic across the tactical network is controlled by Software Defined Networking (SDN), and the link transmissions are scheduled by a Time Division Multiple Access (TDMA) scheme.

B. SDN control plane

The tactical network is SDN-enabled. Each network node is equipped with software that turns it to an SDN switch with local controller functionality. These switches communicate with primary SDN Controllers that compute routes for traffic flows traversing the network and distribute them to all network nodes (switches). Each tactical network node maintains two primary information tables: a routing table, which indicates paths to other nodes in the network, and one or more flow tables with packet forwarding rules for active network flows traversing the node.

The primary SDN Controllers communicate with the switch nodes over *SDN control paths* (SCPs), as computed by legacy routing protocols in operation (*e.g.*, OLSRv2 [10]). Each Controller periodically exchanges heartbeat messages with switches under its control (its associated switches) to ensure that their connections are alive and collects information about the network topology via Link Layer Discovery Protocol (LLDP) packets. Through this process, the SDN Controller(s) can attain a global view of the network topology, compute routing paths for data traffic flows in the network and update the flow tables maintained at each node.

Hence, when packets of a data flow arrive at a switch node, the switch seeks to match the relevant fields of the incoming packets, as extracted from packet header, with its currently installed flow forwarding rules/entries. If a match is



(a) Network topology (nodes, links), inspired by Fig. 1 in [1]

traffic flow id	(src, dst) nodes	rate requirement (in slots)	routing path
f_1	4, 5	2	4→3→2→5
f_2	6, 4	1	6→5→2→3→4
f_3	1, 6	4	1→2→5→6
f_4	3, 1	3	3→2→1

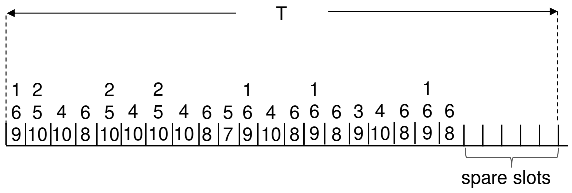
(b) Network traffic flows, their routing paths and flow requirements

Links	1	2	3	4	5	6	7	8	9	10
flows	f_3	f_4	f_2	f_1, f_4	f_2	f_1	f_2	f_1,f_3	f_2	f_3
Slots	4	3	1	5	1	2	1	6	1	4

(c) TDMA slot requirements per network link, as they result from the network flow requirements in (b)

(1, 6, 9) (2, 5, 10) (3, 9) (4, 10) (5, 7) (6, 8)

(d) Maximal compatible link sets for the topology in (a)



(e) TDMA schedule resulting from the per link TDMA slot requirements in (c) and the maximal link compatible sets in (d). Slots beyond those required from the current flows are shown as spare in the frame.

Figure 1: Steps involved in deriving the TDMA schedule for given network topology and traffic flows: $N_s = 25$. Twenty slots are used, five are spare.

not found (new flow), the switch issues a flow setup request to the Controller (within a PACKET-IN message). In response to this, the Controller issues a PACKET-OUT message to install flow rules at the switch that issued the request and may do so proactively for all other switch nodes that are included in the computed path (via PACKET-OUT messages).

These control messages that are exchanged between the SDN Controllers and their associated switches for acquiring information about the network state and updating the flow tables with new entries, constitute major part of the SDN control traffic. The delays experienced by these messages determine how responsive the SDN control plane is to the dynamic topology of the tactical network and can be affected by two main network control operations. The first one is the placement of SDN Controllers, *i.e.*, nodes that host the Primary Controllers and how are the other nodes associated with them. The second one is the way the SDN control traffic is treated over the shared medium. This is the responsibility of the TDMA scheduling process.

C. TDMA link scheduling

The node transmissions over the network links are scheduled in collision-free manner by a TDMA scheme. We consider a fixed-length TDMA frame. Letting R_{TDMA} be the TDMA physical rate and R_s be the minimum traffic rate corresponding to the periodic allocation of one slot per frame, $N_s = R_{TDMA}/R_s$ is the number of slots in each TDMA frame. Moreover, if T is the frame duration, $t_s = T/N_s$ is the duration of a single time slot.

The scheme we consider 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. For any link (u, v), its set of compatible links should exclude: (a) all links $(k, v), k \in N_i(v)$ and $(v, l), l \in N_o(v)$; (b) all links $(k, u), k \in N_i(u)$ and $(u, l), l \in N_o(u)$; and (c) all links $(k, l), l \in N_o(u)$. A *compatible link set* satisfies (a)-(c) for every link in the set and it is *maximal* when no other link can be added to it without violating those conditions[1].

The TDMA scheme then seeks to map each slot in the TDMA frame to a compatible link set so as to most effectively satisfy the rate requirements of the user data traffic flows. Each flow f is described by the tuple (src_f, dst_f, r_f, p_f), where src_f and dst_f are the flow source and destination nodes, respectively, r_f is the flow rate requirement in number of TDMA slots, and p_f is the $src_f - dst_f$ path, as computed by the network routing protocol. The TDMA schedule derivation proceeds along three steps, which are shown in the simple example scenario of Fig. 1:

- First, the scheduler computes maximal compatible link sets out of the current tactical network topology $G_t =$

---

[1] 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.*, [17]. The analytical model is acknowledged as a good useful approximation of real-world interference.

(V, E_t) [18]. For the network example topology in Fig. 1a, there are six such sets that are listed in Fig. 1d

- It then considers the user traffic flows currently served by the network, including their traffic rates[2] and routing paths and computes the total rate requirements r_l that result for each network link $l \in E_t$. In fig. 1c, we identify the subsets of the four flows in 1b that traverse each link in the network of Fig. 1a and compute the slot requirements they generate for the link.
- Finally, it determines which maximal compatible link sets need to be used and how many times in each frame, to cover the rate requirements of all network links. Viewing links as elements that set different coverage requirements and maximal compatible link sets as subsets of the full link set, we can reduce the TDMA schedule derivation to an instance of the Set Multicover problem and solve it through a heuristic algorithm (see *e.g.*, [19]). In Fig. 1e, we provide one such schedule, which maps compatible link sets to TDMA frame slots for the example of Fig. 1.

D. SDN layer communication delays over TDMA hops

The delays that are experienced by the SDN control messages in the network depend on both the SDN placement and the TDMA scheduling. Assume, for instance, that in the network example of Fig. 1a there is one SDN Controller that is located at node 5. An SDN control packet generated at node 4 needs to traverse links (4,3), (3,2) and (2,5) to reach the SDN Controller. The delay to traverse these three hops depends on how the maximal compatible link sets are scheduled over the TDMA frame. Under the TDMA schedule of Fig. 1e, if the SDN control packet arrives sometime during the third slot of the TDMA frame, it can traverse the first radio hop (4,3) already in the next slot but it will need to wait 1.5 more slots, at node 3 to get access to link (3,2) and, even worse, 2.5 more slots at node 2. The overall delay is 5 slots. On the other hand, if the SDN control packet is generated at node 4 sometime during fifth slot of the TDMA frame, it would need to wait 3 slots for traversing (4,3) and 3.5 more to cross (3,2). It would still need to wait for another 1.5 slots for the final (2,5) hop, for an overall end-to-end delay of 8 slots. Since a packet may be generated with the same probability any time within the duration of a TDMA frame, the expected delay $D_{6,1}$ experienced by packets in the SDN control path $6 \rightarrow 5 \rightarrow 2 \rightarrow 1$ is given by

$$D_{4,5} = \frac{1}{N_s} \sum_{n=1}^{N_s} D_{4,5}(n) \tag{1}$$

where $D_{4,5}(n)$ is the packet end-to-end delay when generated within time slot n.

Table I reports the expected end-to-end delays $D_{x,y},\ \ x, y \in \{1, .., 6\}, x \neq y$ between all node pairs in the topology of Fig. 1a. It is clear that the SDN control path communication delays over the TDMA frame are affected by two factors. The first one is the number of slots assigned to path links. For example, $D_{3,2} < D_{3,4}$ in Table II since link (3,2) is allocated five slots in the TDMA frame and (3,4) is assigned four slots, although both nodes 2 and 4 are one-hop neighbors of Controller 3. The second factor is the distribution of assigned slots within the TDMA frame, when the assigned slots are two and more. In Table I $D_{1,2} < D_{3,4}$, although both (1,2) and (3,4) represent one hop paths and are assigned four slots in the TDMA frame. However, the slots assigned to link (1,2) are equidistantly distributed over the frame. The more uniform the distribution of slots over the frame is, the smaller the TDMA access is delay [1].

Table I: Expected pairwise node end-to-end delays (in TDMA slots) for the network in Fig. 1a under the TDMA schedule in Fig. 1e.

node id	1	2	3	4	5	6
1	0	4.16	15	26	22	28
2	8.76	0	13	24	13	19
3	10.92	4	0	6.72	15	21
4	12.92	6.12	2.32	0	19	25
5	12.08	3.28	14	25	0	4.8
6	13.08	4.28	15	26	2.36	0

III. COORDINATED SDN CONTROLLER PLACEMENT AND TDMA SCHEDULING

A. Controller placement revision policies

The SDN Controller placement process determines the location of SDN Controllers in the network and the resulting Controller-switch node associations that satisfy certain delay and reliability requirements. In the remainder of the paper, we focus on the single SDN Controller case, leaving the multiple Controllers' case as future work. With a single SDN Controller, the placement process is considerably simplified: the Controller is placed at the node that minimizes the average (*k-median* problem) or worst-case (*k-center* problem) delays over all Controller-to-Switch node pairs across the network.

However, as the network nodes move around and traffic flows enter/leave the network, they cause changes in the TDMA schedule and the delays experienced at the SDN layer, as discussed in section II-D. We may, hence, identify three different policies regarding the frequency at which the SDN placement is revised, each representing a different trade off between responsiveness to changes and overhead in terms of information exchange in the network:

- *Static placement (no adaptation)*: the SDN Controller placement does not change over time. This policy yields the highest delays and it is aimed as a reference for comparison, rather than a realistic alternative.
- *Joint SDN placement and TDMA scheduling*: On the other extreme, the SDN Controller placement could be updated every time the TDMA schedule changes. This is a policy that implies maximum adaptability to the changes happening in the network, but also the highest

---

[2] For voice and video streaming flows, the rates correspond to the voice coding and streaming rates, respectively; for elastic traffic flows, *e.g.*, TCP controlled, they could be a minimum guaranteed rate allocation.

overhead. It demands that the SDN Controller placement and the TDMA schedule are jointly derived every time there is a topological or traffic change in the network and sets a lower bound on the delays between the Controller and the other nodes.

- *Periodic revision of the SDN controller placement*: In practice, the two operations take place at different time scales. TDMA schedules track closely and adapt to the short-term changes in network topology and traffic, whereas the SDN controller placement may be revised over longer intervals to cope with possible high delays at the SDN control plane.

B. SDN-aware TDMA scheduling

On the other hand, the TDMA scheduler has some degrees of freedom in the way it treats SDN control plane traffic. These relate to:

The allocation of slots to SDN control paths: As a baseline practice, the schedule will ensure that at least one slot is allocated to all links in the network, irrespective of whether they serve data traffic flows or not, so that all SDN control messages can be served. A more SDN-aware TDMA scheduling would book additional slots for links that serve more SDN control traffic. For example, in the network of Fig. 1a, if the SDN Controller is placed at node 1, links (1,2) and (2,1) mediate the communication of the Controller with all nodes 2-6. A *one-slot-every-K* SDN control paths policy would book one slot for every K SDN control paths served by a link. With $K = 3$, for instance, links (1,2) and (2,1) would be assigned one more slot in the TDMA frame (since they serve 5 SDN control paths each). More precisely, with this policy, a link is assigned $\lceil P/K \rceil$, where P is the number of SDN control paths traversing the link.

The ordering of maximal compatible link sets in the TDMA frame: The delay experienced by data over each single radio hop may change considerably depending on the way the required maximal compatible link sets are mapped to TDMA frame slots. This delay decreases as the slots reserved for each link are spread over the TDMA frame rather than when they are assigned back-to-back [1]. Moreover, for data flows traversing multiple TDMA hops on their path, the end-to-end delay is reduced when the order of the maximal compatible link sets in the frame is aligned with the flow path, *i.e.*, the sets succeed each other in the frame, ideally back-to-back, giving rise to flow paths. This way, once a packet gets a slot to traverse the first TDMA hop, it can traverse the remaining ones without additional "transit" delay. Therefore, an SDN-aware TDMA scheduler could seek to align the mapping of maximal compatible link sets to TDMA frame slots with SDN control paths,Fig. 2 provides such a schedule and Table II reports the smaller pairwise end-to-end delays that result from it. We have used a greedy algorithm to derive such a mapping. Input to the algorithm are the maximal compatible link sets and the times each has to appear in the TDMA frame to cover the rate requirements of each network link (see section II-C). The algorithm is executed sequentially for each slot. All remaining link sets are scored according to the number of SDN control paths they can "progress" with. A maximal compatible link set progresses with an SDN control path when it includes the next hop link on that path. For instance, in Fig. 2, the link set $\{((2),(5),(10)\}$ progresses two SDN control paths, the $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ path from the Controller 1 to node 4 over links $\{(1),(3),(5)\}$ and the $1 \rightarrow 2 \rightarrow 5 \rightarrow 6$ path from Controller 1 to node 6 over links $\{(1),(8),(10)\}$. The maximal compatible link set that holds the maximal score in the given slot (or one of those with the highest score, if they are more than one) is assigned to the slot and we increase by one a counter that logs how many times it has been used in the frame. When a maximal compatible link set reaches the number of times it has to appear in the frame, it is removed from further consideration. The algorithm terminates when this limit is reached by all input sets.

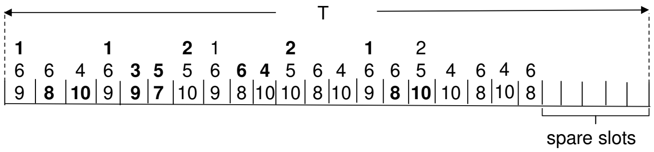


Figure 2: Alternative way to schedule maximal compatible link sets over the TDMA frame for the example of Fig. 1 in line with SDN control paths when node 1 is the SDN Controller. Numbers in bold mark SDN control paths, *e.g.*, $(1) \rightarrow (8) \rightarrow (10), (1) \rightarrow (3) \rightarrow (5), (6) \rightarrow (4) \rightarrow (2), (9) \rightarrow (7) \rightarrow (2))$.

Management of spare capacity: When the user demand is low/medium and does not fully utilize the TDMA frame slots, the spare capacity (*e.g.*, the five slots in Fig. 1e or Fig. 2) are typically distributed in round-robin fashion to ongoing user data traffic flows; for instance, to elastic TCP-controlled traffic flows that could benefit from additional capacity and scale up their throughput. An SDN-friendly TDMA could prioritize links serving SDN control paths when doing so.

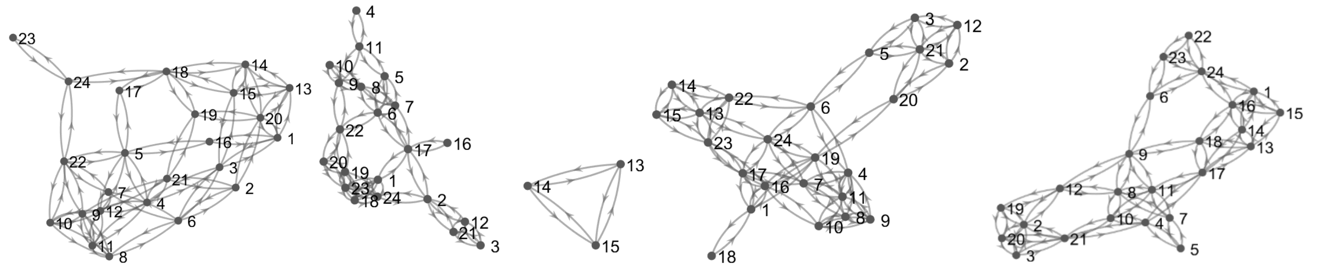
IV. NUMERICAL EVALUATION

A. Methodology

We study how the SDN controller placement interacts with the TDMA scheduling function in tactical MANETs. The three SDN Controller placement policies in section III-A and the three TDMA mechanisms for prioritizing SDN control-plane traffic in section III-B are tested over a number of

node id	1	2	3	4	5	6
1	0	4.6	14	15	15	16
2	6.44	0	13	14	13	14
3	9.2	3.68	0	5.84	16	17
4	10.2	4.76	2.2	0	17	18
5	8.28	3.08	16	17	0	5.32
6	9.28	4.2	17	18	2.44	0

Table II: Expected pairwise node end-to-end delays (in TDMA slots) for the network in Fig. 1a under the TDMA schedule in Fig. 2.



Topology snapshot 1 Topology snapshot 12 Topology snapshot 21 Topology snapshot 66

Figure 3: Four network topology snapshots out of the Anglova mobility traces that were used in our numerical evaluation.

distinct *network snapshots*. Each network snapshot is uniquely specified by the union of the network topology, as this results from the mobility of the MANET nodes, and the set of user data traffic flows that are served by the network. For the MANET node mobility, we have used the publicly available traces of the Anglova scenario [20]. In particular, we track the movement of a company of 24 nodes (company 1) during the troop deployment phase (vignette 2) that lasts 7800 seconds. We capture snapshots of the nodes' locations in (latitude, longitude) pairs every 100 seconds and construct the related digraphs (*topology snapshots*) after converting these coordinates to pairwise node distances and factoring in radio link parameters. Four of those graphs are shown in Fig. 3.

Then, for each of these 78 graphs, we generate 12 different traffic flow sets, distinguishing between low/moderate and heavy demand scenarios. The difference between the two lies in the number of spare slots in the TDMA frame, as shown in Fig. 4. The number of traffic flows in each set follows a uniform distribution in [3,5] for moderate demand and in [7,8] for heavy demand. In both cases, the flow source and destinations nodes are picked randomly, their rate requirements (in slots) are also uniformly sampled in [1,4] and their routing paths are computed as shortest paths over the graph. Overall, we get approximately 936 (78 graphs x 12 flow sets) distinct network snapshots for each of the two demand scenarios.

Regarding the TDMA parameters, the TDMA frames consist of $N_s = 75$ slots and the TDMA PHY rate is $R_s = 1.5$ Mbps so that 1 TDMA slot corresponds to a rate of 20kbps. We compute four different TDMA schedules for each of the 936 network snapshots: one (called baseline) that does not implement any of the SDN-friendly mechanisms in section III-B and three more implementing one mechanism each. For ease of reference, we list these four scheduling policies together with the notation subsequently used in plot legends in Table III.

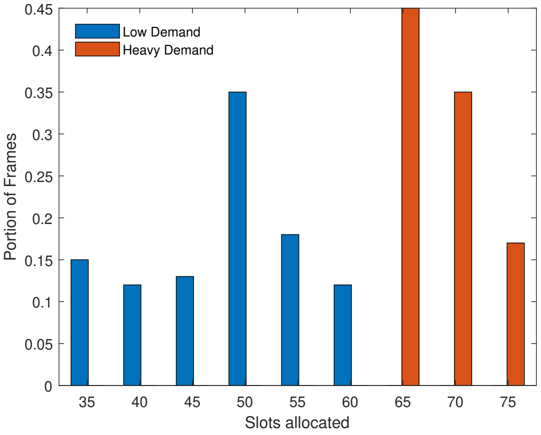


Figure 4: Distribution of reserved slots across the TDMA schedules corresponding to the 78x12 network snapshots, under low and heavy demand

On the other hand, Under the joint Controller placement-TDMA scheduling policy and the periodic revision policy, we simultaneously determine the SDN Controller placement and the TDMA schedule for each network snapshot or every K network snapshots, respectively. We do this through a heuristic iterative process: starting from an initial placement of the SDN Controller, we compute the SDN-aware TDMA schedule, which in turn may change the SDN Controller placement and generate a new TDMA schedule. The iterations end when there is convergence, *i.e.*, the SDN Controller location and the TDMA schedule do not change any more. The solid mathematical formulation of the joint SDN Controller placement and TDMA scheduling problem, its characterization and a more exhaustive search for efficient solution algorithms is an interesting direction for future work.

Irrespective of how the SDN Controller placement and TDMA schedule are derived, we measure the expected communication delays (in TDMA slots) in the 23 SDN control paths from the SDN Controller to the other twenty three nodes so that we come up with two sets of 21.5K such values, called *measurement points*, for the 936 network snapshots, one for each demand scenario. We compare the different policies with respect to statistics of these delays, including their distribution, averages and worst-case values. We also record the worst delay (in TDMA slots) among every network snapshot, under different combinations of scheduling and controller placement policies, concerning both low/moderate and high demand scenarios, depicted in Table IV. We used Python scripts for generating the network snapshots out of the original Anglova

Legend notation	Description
baseline	no SDN-aware policies
spare	baseline + prioritization of SDN control messages when spare capacity is available
bonus(K)	baseline + additional slots to links in SDN control paths with priority: 1 more slot every K served paths
path-ordered	baseline + ordering of maximal compatible link sets in the TDMA frame in line with SDN control paths

Table III: TDMA scheduling policies and notation

traces in EMANE and the MATLAB simulation environment for the remaining analysis.

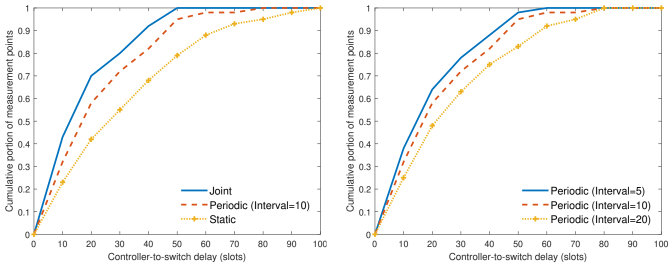
B. Numerical results

1) Controller placement policies with baseline TDMA scheduling: The first set of experiments sets a reference for the achievable communication delays on SDN control paths under the three SDN Controller placement revision policies. Figures 5 and 6 plot the empirical cumulative distribution of these delays when these policies are applied over the baseline TDMA scheme, under low/moderate and high demand, respectively. In line with intuition, when the tactical network undergoes the complexity of updating the SDN Controller placement every time the TDMA schedule is revised, the encountered SDN control path delays are the smallest. Notably, the delays under this policy are less sensitive to the demand scenario, exhibiting worst-case values in the order of one TDMA frame (75 slots).

On the contrary, under the static case, the worst-case delay almost doubles, compared to joint case, as shown in Table IV, for both low demand scenario (52, 102 slots) and high demand scenario (65, 120). The performance of the periodic Controller placement update is better shown in Fig. 5b and Fig. 6b. Tuning the update frequency of the periodic policy (every 5, 10, 20 topology snapshots corresponding to 60, 120 and 240 network snapshots, respectively) we can span the delay space that is bounded by the joint policy (at the low end) and the static one (at the high end).

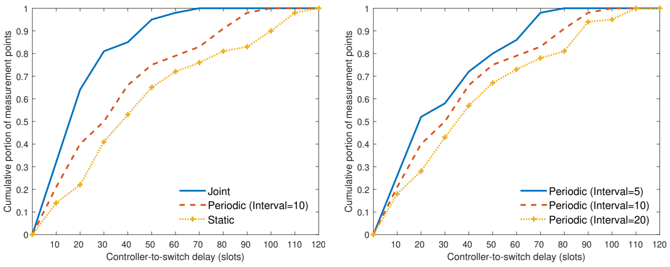
2) SDN-aware TDMA scheduling policies over fixed controller placement: In the second set of experiments, we fix the location of the Controller throughout the network snapshots and compare the three SDN-aware TDMA scheduling policies in Table III. The implication is that, instead of struggling to update the Controller placement every time something in the network changes (topology or user data traffic), we could stick to what we have at the SDN plane and try to generate TDMA schedules that are SDN-friendly (SDN-aware). We plot the empirical cumulative distribution of the delays experienced by the controller in Fig.7. Path-ordered/ordering scheduling policy outperforms the rest scheduling policies in every demand scenario. The other two policies have a less dramatic impact, which is also sensitive to the amount of data traffic. Hence, under high demand, the TDMA capacity is scarce and there is little margin to allocate more slots to links involved in SDN control paths. The results essentially confirm the compensatory effect that path-ordered scheduling policy can have upon the SDN control path delays, when in worst cases results to minimum delay against the other policies, as recorded at fourth row of Table IV.

3) Periodic Controller placement revision over TDMA scheduling policies: The final set of experiments presented in Fig.8 combines the periodic update of SDN Controller placement with the different SDN-aware TDMA scheduling policies. The main take away from these experiments, is that ordering the maximal compatible link sets in the TDMA frame in line with SDN control paths and updating periodically controller's placement yields delays that are comparable to



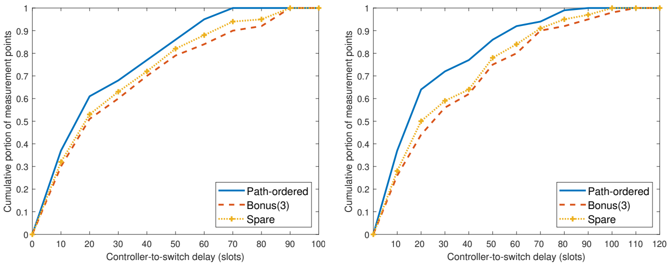
(a) all three policies (b) periodical update policy

Figure 5: SDN control path delays under the three Controller placement revision policies and baseline TDMA scheduling: low/moderate traffic demand



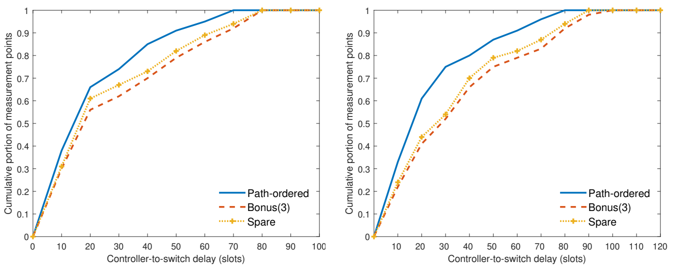
(a) all three policies (b) periodical update policy

Figure 6: SDN control path delays under the three Controller placement revision policies and baseline TDMA scheduling: high demand



(a) moderate traffic demand (b) high traffic demand

Figure 7: SDN control path delays under SDN-aware TDMA scheduling and static SDN Controller placement.



(a) moderate traffic demand (b) high traffic demand

Figure 8: SDN control path delays under SDN-aware TDMA scheduling and periodical revision of the Controller placement every 10 topology snapshots

Table IV: Worst-case controller-to-switch delays (in TDMA slots) under different combinations of scheduling and controller placement policies for each demand scenario (low, high)

TDMA schedule	Joint	Periodic(10)	Static
Baseline	52, 65	85, 101	102, 120
Spare	-,-	78, 85	87, 92
Bonus(3)	-,-	81, 90	91, 101
Path-ordered	-,-	64, 74	73, 85

what we get with the joint SDN Controller placement policy for both demand scenarios. We notice a significant reduction of those delays, in particular under high demand and in terms of worst-case values depicted in Table IV.

V. CONCLUSIONS

In this work, we studied the interplay between the SDN Controller placement and TDMA scheduling functions in tactical TDMA mobile ad hoc networks. We described three ways in which the TDMA scheduler could become SDN-friendly and reduce the experienced delays over SDN control paths. This helps SDN become more responsive to the dynamic topology and workload of the network, while reducing the overheads of frequently updating the SDN Controller placement. Being experimental, our study has considered concrete algorithmic solutions for the actual controller placement problem and the derivation of the TDMA schedule. Although we might see some differentiation in the precise numbers when using alternative solutions, we expect the qualitative conclusions of this study to persist.

One direction for future work, already outlined in section IV-A, is the mathematical formulation and characterization of the joint SDN controller placement and TDMA scheduling problem. A second direction, with both analytic and experimental goals, is the extension of the study to scenarios with multiple SDN Controllers, each controlling a subset of the tactical network nodes. In that case, controller-to-controller delay should also be taken into consideration.

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.

REFERENCES

[1] R. Nelson and L. Kleinrock, "Spatial TDMA: A Collision-Free Multihop Channel Access Protocol," *IEEE Transactions on Communications*, vol. 33, no. 9, pp. 934–944, 1985.

[2] L. Kleinrock and J. Silvester, "Spatial reuse in multihop packet radio networks," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 156–167, 1987.

[3] I. Rhee, A. Warrier, J. Min, and L. Xu, "Drand: Distributed randomized tdma scheduling for wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 10, pp. 1384–1396, 2009.

[4] D. Baker, A. Ephremides, and J. Flynn, "The design and simulation of a mobile radio network with distributed control," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 1, pp. 226–237, 1984.

[5] I. Chlamtac and A. Lerner, "Fair algorithms for maximal link activation in multihop radio networks," *IEEE Transactions on Communications*, vol. 35, no. 7, pp. 739–746, 1987.

[6] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, 1988.

[7] J. Shaio, J. Piper, and B. J., "A slot allocation algorithm for survivability of tactical tdma networks," The MITRE Corporation, Technical Report, Tech. Rep., 2011. [Online]. Available: https://www.mitre.org/sites/default/files/pdf/10_3948.pdf,

[8] J. Nobre, D. Rosario, C. Both, E. Cerqueira, and M. Gerla, "Toward software-defined battlefield networking," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 152–157, 2016.

[9] K. Poularakis, G. Iosifidis, and L. Tassiulas, "SDN-Enabled Tactical Ad Hoc Networks: Extending Programmable Control to the Edge," *IEEE Communications Magazine*, vol. 56, pp. 132–138, 2018.

[10] C. Barz, C. Fuchs, J. Kirchhoff, J. Niewiejska, and H. Rogge, "Olsrv2 for community networks: Using directional airtime metric with external radios," *Computer Networks*, vol. 93, pp. 324–341, 2015.

[11] 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.

[12] 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.

[13] K. Poularakis, Q. Qin, K. M. Marcus, K. S. Chan, K. K. Leung, and L. Tassiulas, "Hybrid SDN Control in Mobile Ad Hoc Networks," in *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, Los Alamitos, CA, USA, June 2019, pp. 110–114.

[14] I. Koutsopoulos, "Learning the optimal controller placement in mobile software-defined networks," in *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2022, pp. 70–79.

[15] 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.

[16] M. J. Abdel-Rahman, E. A. Mazied, A. MacKenzie, S. Midkiff, M. R. Rizk, and M. El-Nainay, "On stochastic controller placement in software-defined wireless networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[17] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '05. USA: USENIX Association, 2005, p. 28.

[18] 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.

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

[20] NATO IST-124 Research Task Group on Heterogeneous Networks: Improving Connectivity and Network Efficiency. Anglova tactical network emulation scenario. [Online]. Available: https://anglova.net