Controller Placement and TDMA Link Scheduling in Software Defined Wireless Multihop Networks

Yiannis Papageorgiou, Merkouris Karaliopoulos, and Iordanis Koutsopoulos

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

Abstract—**In this paper, we iterate on Software Defined Wireless Multihop Networks (SDWMNs) and TDMA-scheduled links, where data and SDN control traffic compete for the same resources. Two control functions are key in ensuring adequate Quality of Service (QoS) for data flows and high responsiveness (low latencies) for the SDN control-plane messages exchanged between the network nodes: the SDN Controller placement, which determines the paths of SDN control messages across the network and their overlap with the data traffic paths; and the TDMA scheduling, which distributes time slots between these two types of traffic, prioritizing them in different ways.**

Our take, in this paper, is that by coordinating these two control functions, rather than executing them independently, we can deal more efficiently with the data QoS-SDN responsiveness trade-off. We, thus, pursue the joint optimization of controller placement and link scheduling, formulating the respective optimization problem and proposing a novel heuristic algorithm for it. Its main idea is to, first, determine maximal sets of simultaneously transmitting non-interfering links to serve the data traffic requirements and, then, seek a Controller placement that takes best advantage of the spare link transmission opportunities in those sets. We compare our algorithm with benchmark solutions that carry out the two control functions independently and find that it trades far better the rate that can be allocated to data traffic with the communication delays at the SDN control plane.

Index Terms—**Controller placement problem, Time division multiple access(TDMA), Cross-layer optimization, Software Defined Networks**

I. INTRODUCTION

Over the last fifteen years Software Defined Networking (SDN) has emerged as one of the most promising networking architectures. Explicitly separating the control and data plane functionalities, SDN delegates control tasks to dedicated nodes, the controllers, and lets SDN-enabled switch nodes handle data-plane forwarding. The controller nodes centrally coordinate the routing of data packets by computing routes and installing forwarding rules for individual flows at the switch nodes. With apparent advantages for the routing configuration and network management processes, SDN is widely adopted in wireline core networks and, over time, has been proposed for a broad range of wireless networking paradigms such as mobile cellular [1] and Internet-of-Things networks [2].

In the case of mobile ad hoc networks (MANETs) [3], the challenge is to make the centralized SDN architecture adequately responsive to the fast-varying topology of those networks and integrate it with the decentralized implementations of de facto standard algorithms that control the routing of traffic and the scheduling of node/link transmissions. Time Division Multiple Access (TDMA) with spatial reuse [4], in particular, schedules transmissions of multiple nodes or links at the same slot as far as they do not interfere with each other. For reasons of cost and complexity, these TDMA slots serve both the user data traffic and the SDN control-plane messaging so that the two traffic types need to compete for their use.

Our point of departure in this paper is the remark that the delays experienced by SDN control messages (hence, the SDN responsiveness) are primarily affected by two WMN control operations: the controller placement, determining the location of the SDN controller(s) and its(their) paths towards the rest of the nodes; and the TDMA scheduling, determining how the SDN control traffic is treated over the shared radio links. We then argue that by coordinating these two control functions, we can increase the responsiveness of the SDN control plane in the WMN (*i.e.*, achieve smaller delays for SDN messages) without proportionally penalizing the performance of user data flows (*i.e.*, rate of file transfers, voice/video delay).

Related work: The Controller Placement Problem (CPP) has been treated extensively in literature (interested readers are referred to related surveys such as [5]). It was first introduced in the context of ISP networks in [6], which demonstrated the fundamental trade-off between communication latency and control message overhead at the SDN level: as more SDN controllers are placed in the network, the average and worst-case latency to reach the switch nodes are reduced at the expense of increased network overhead in terms of exchanged messages. While a long thread of papers thereafter considered variations of the original problem statement and proposed improved controller placement algorithms, *e.g.*, [7], more relevant to our work is the study of CPP in wireless networks, which raises new challenges such as the reduced link reliability (non-negligible packet loss) due to radio propagation impairments and radio interference. Hence, in [8] the authors carry out a sensitivity analysis of various WiFi network performance metrics to the locations chosen for the placement of the controller(s), while in [9] the CPP is studied in the context of TDMA cellular wireless networks. The focus is more on the sensitivity of the controller placement to the radio link quality, whereas the TDMA access delay over the one-hop wireless link is assumed fixed and equal to half the TDMA frame duration. In a first study of the CPP in the context of wireless multihop networks in [10], we showed that adding SDN-awareness to TDMA scheduling reduces the communication delays between the controller and the associated switches.

Our contributions: In this paper, we systematically study the coordination of the SDN Controller placement and the TDMA scheduling function. Our first contribution is the formulation of the Joint Controller placement and TDMA link scheduling (JCPTS) problem in Software Defined WMNs, which seeks to weigh the performance provided to user data flows against the SDN control-plane delays. The second and main contribution consists in a heuristic algorithm for JCPTS. The algorithm first determines maximal sets of non-interfering links and allocates them to TDMA slots in line with the user data traffic requirements. It then places the SDN Controller where it takes best advantage of spare link transmission opportunities that emerge in those slots. Our experimentation suggests that our algorithm can reduce SDN controller-to-switch delay communication without penalizing the QoS that is provided to data traffic.

The rest of the paper is organized as follows. In section II we describe our system model. Then, in section III we mathematically formulate the JCPTS problem and in Section IV we describe our heuristic algorithm for it. Numerical results are presented in Section V and we conclude our paper in section VI.

II. SYSTEM MODEL

A. Wireless Multihop Network

Let the digraph $G_t = (V, E_t)$ represent the network at time t, where V is the set of (mobile) nodes and E_t is the set of wireless links realized at time t between those nodes, depending on their locations and the radio propagation conditions across the network area. By $(u, v) \in E_t$ we denote the link, where u and v are the transmitting and receiving nodes, respectively. Node v is called an out-neighbor of node u and node u is called an in-neighbor of node v. 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 $(u, k), (k, v) \in E_t$.

B. SDN layer

Each network node is equipped with software that turns it to an SDN switch with local controller functionality, while one node is elected as the Master SDN controller, as shown in Fig. 1 (we focus here on the single SDN controller scenario, letting the multiple controller case as future work). The Master SDN controller periodically exchanges heartbeat messages with all other switches that are under its control (associated switches) and collects switch- and link-level statistics (bandwidth, failure probability) via Link Layer Discovery Protocol (LLDP) packets [11]. This information lets it acquire global knowledge about the network topology and dynamically update two types of information tables at the switches: the routing table, which logs network paths toward other nodes, and one or more flow tables listing packet forwarding rules for active network flows traversing the node.

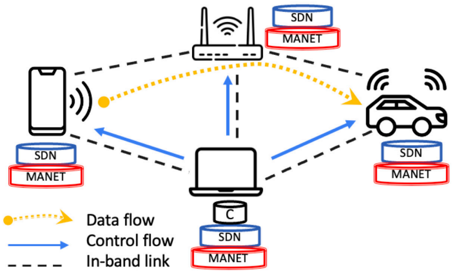


Fig. 1. Software Defined WMN with single SDN controller

When a new flow arrives at a switch, that switch first checks its flow table(s) for a matching entry. If such an entry is found, the flow packets are forwarded according to the corresponding rule; otherwise, the switch issues a PACKET-IN message to its Master SDN controller. The controller computes a path using its knowledge about the network and, via PACKET-OUT messages, installs forwarding rules at the switch that initiated the PACKET-IN message but also, proactively, to the other switches on the flow path. All these messages between the switches and the controller travel through paths called *SDN control paths*, which are typically shortest paths derived by native WMN routing protocols such as the Optimized Link State Routing protocol-version 2 (OLSRv2) [12], [13].

C. TDMA scheduling

The node transmissions 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.

1) Maximal Compatible Link Sets: The TDMA scheduler assigns slots to maximal compatible link sets (MCLSs), namely maximal sets of links that can simultaneously transmit without interfering with each other. Necessary and sufficient conditions for a link set to be *compatible* are defined in [14]: 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 $(k, l), l \in N_o(u)$. A compatible set is called *maximal* if it cannot grow any further without violating at least one of conditions (a)-(c)[1]. In [14], these conditions motivate a trivial greedy algorithm for the derivation of one, each time, MCLS. Hereafter, we denote the set of MCLSs by $\mathcal{M}$.

2) End-to-end communication delays over TDMA hops: A packet, either control or data, is generated randomly within the duration of a TDMA frame. Then, the end-to-end delay experienced by the packet packet depends on how different MCLSs are ordered within a TDMA frame. For example, in

---

[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.*, [15]. The analytical model is, however, acknowledged as a useful approximation of real-world interference.

the WMN example of Fig. 2a, consider a packet arriving at node ’3’ and having to traverse links (6) and (12) to reach its destination node ’7’. If its arrival at node ’3’ coincides with the 3rd slot of the TDMA frame (see top of 2b), it has to wait for 12.5 slots to traverse (6) and 3 slots for the last hop through (12). The best-case end-to-end delay, assuming no additional queuing delays at any link, is 15.5 slots. On the other hand, if the packet arrives at node ’3’ within the 11th slot of the frame, it takes 3.5 slots to traverse link (6) and 3 more for link (12), summing up to 6.5 slots. The expected end-to-end delay experienced by packets from node ’3’ to node ’7’, denoted by $D_{3,7}$ is calculated by

$$D_{3,7} = \frac{1}{N_s}\sum_{n=1}^{N_s} D_{3,7}(n) \tag{1}$$

where $D_{3,7}(n)$ is the end-to-end delay for packets generated within time slot n.

D. Traffic and slot allocation policy

The WMN delivers two types of traffic, data flows and control-plane messages between the controller and the other nodes. SDN messages are delivered in-band, namely they use the same resources (frequency channels, TDMA slots) with data flows. The WMN nodes maintain separate buffers/queues for data flows and SDN control messages at the MAC layer and keep record of their slot allocations for each type of traffic.

We assume that the scheduler has perfect information about data traffic flows. Namely, if F_t is the set of ongoing data flows at each point in time at the WMN (active flows), from the TDMA scheduler point of view, each flow $f \in F_t$ is characterised by the tuple (s_f, d_f, r_f, p_f), where s_f and d_f are the flow source and destination nodes, respectively, r_f is the flow rate in number of TDMA slots($\lceil$ f total rate$/R_s$ $\rceil$), and p_f is the $(s_f - d_f)$ routing path computed for the flow by the SDN controller. Indeed, such an assumption implies that there is some coordination of the TDMA scheduler with 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 inelastic data flows (*e.g.*, voice or video streaming), the number of TDMA slots assigned for each link on the flow path equals the flow rate, r_f. On the other hand, for elastic, *e.g.*, TCP-controlled, data flows, the scheduler allocates a minimum number of K slots to each link in the flow path. If there are spare slots in the frame after all voice and streaming traffic is served, they may be distributed between elastic flows till the frame slots are exhausted. Simultaneously, the scheduler should allow for TXOPs that serve SDN messages exchanged between the SDN controller and switch nodes.

III. FORMULATING THE JCPTS PROBLEM

Let $Y_m \in \{0, 1, ..., N_s\}$ be an integer decision variable denoting the number of times MCLS m appears in the TDMA frame. It should hold that

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

Let also binary variables

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

where the subscript t has been dropped from E_t and $\mathcal{M}_t$ to simplify notation, and

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

with

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

The placement of the controller at node v generates a distinct set of SDN control paths towards the other WMN nodes. Let SCP_{vu} denote the control path from v to associated switch u and D_{vu} be the delay v experiences when receiving messages from u, measured in frame slots over the current TDMA frame. If $\mathcal{F}$ is the set of active data flows in the network, the amount of slots each network link l needs to be assigned in the TDMA frame is

$$r_l^d = \sum_{f\in\mathcal{F}:l\in p_f} r_f \tag{6}$$

where $r_f = K$ for flows $f \in \mathcal{F}_e$, *i.e.*, in the set of elastic flows, and r_f equal to the slot-equivalent of the requested rate for $f \in \mathcal{F} \setminus \mathcal{F}_e$. Each network link l appears multiple times in frame, as part of different MCLSs, assuming TXOPs that can be used for either data or control traffic. If variable

$$q_{lmj} = \begin{cases} 1 & \text{if the TXOP of link } l \text{ in the } j^{th} \text{ occurence of} \\ & \text{MCLS } m \text{ is for data flows} \\ 0 & \text{if the TXOP of link } l \text{ in the } j^{th} \text{ occurence of} \\ & \text{MCLS } m \text{ is for control flows} \end{cases} \tag{7}$$

for $l \in E, m \in \mathcal{M}, 1 \leq j \leq Y_m$, denotes whether a given TXOP in a certain MCLS is reserved for data flows or control flows, the number of TXOPs that are reserved at link l for data flow transmissions are:

$$n_l^d = \sum_{m\in\mathcal{M}}\sum_{j=1}^{Y_m} Y_m \cdot x_{lm} \cdot q_{lmj} \tag{8}$$

and should satisfy

$$n_l^d \geq r_l^d \quad l \in E \tag{9}$$

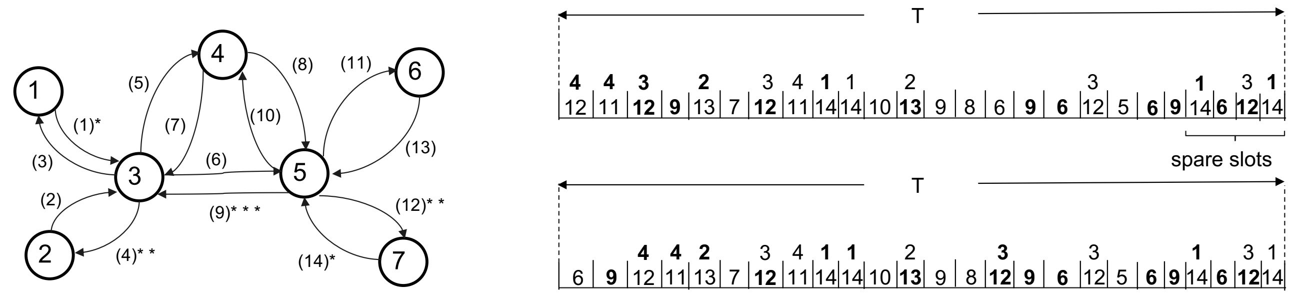
whereas each link should be assigned a minimum of one transmission opportunity

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

in frame for control traffic such as routing messages[2].

---

[2] These slots could also be used for TCP ACK packet traffic, which is not explicitly considered when allocating slots for elastic TCP-controlled traffic



(a) Network topology (nodes, links), inspired by Fig. 1 in [16]

(b) Frames before and after ordering

Fig. 2. TDMA schedule for given network topology and data flows of Table I: $N_s = 25$, whereas 21 slots are needed for serving data flows requirements and 4 are spare. Bold links in frame are dedicated to data flows and rest for control messages.

TABLE I
DATA FLOWS

Flow id	Type	(src,dest)	slot requirements	routing path
f1	CBR	6,1	1	$(13) \to (9) \to (3)$
f2	CBR	2,7	1	$(2) \to (6) \to (12)$
f3	CBR	5,2	2	$(6) \to (4)$
F4	TCP	1,7	1	$(1) \to (6) \to (12)$

The objective in our optimization problem combines two contradicting goals, the minimization of the mean delays over the SDN control paths and the maximization of the additional normalized rate (beyond the minimum K slots) allocated to elastic data traffic flows. Formally, we seek to

$$\min_{\mathbf{Y},\mathbf{z},\mathbf{q}} \quad w_1 \sum_{v\in V} z_v \frac{\sum\limits_{u\in V\setminus v} D^c_{vu}}{|V|-1} - w_2 \sum_{f\in\mathcal{F}_e} r_f\Big(\bigcup_{l\in p_f} (n^d_l - r^d_l)\Big)$$

$$s.t. \quad (2)-(9) \qquad (OPT)$$

In the objective function, w_1 and w_2 are constants (weights) that let prioritize the one goal over the other. The rate r_f in the second sum is given as an explicit function of the additional TXOPs reserved at links on the elastic flow paths for data traffic. This function is hard to describe analytically.

IV. PROPOSED ALGORITHM

The JCPTS problem in section III is an Integer Programming problem. We propose a heuristic solution for it, exemplifying its steps for the toy network example of Fig. 2a.

Step 1: First, for given network topology G, the scheduler computes a set of MCLSs, executing multiple times the greedy algorithm in [14]. For the 14 links in Fig.2a, three such sets are $((1),(14)),((4),(12))$ and (9).

Step 2: Then, it computes the total rate requirements in numbers of slots, r^d_l, that result for each network link $l \in E$ out of the data flows, as explained in section III, eq. (6). In Table I, we list those requirements when the example WMN serves four data flows.

Step 3: Next, it determines which MCLSs need to be used and how many times in the TDMA frame so that the rate requirements of all network links are satisfied, in line with (9), (10). This task is reminiscent of the Set Multicover problem and is solved through the greedy Algorithm in *e.g.*, [17]. Spare TDMA slots are distributed to elastic data flows, *e.g.*, F_4 in Fig 2b. Namely, our algorithm iteratively computes additional MLCSs that suffice to provide one or more slots to those flows till the frame capacity is exhausted. At each slot, the TXOPs assigned to data traffic are marked.

TABLE II
RANKINGS OF THE WMN LINKS IN FIG.2A ACCORDING TO: (A) (COL. 2) NUMBER OF SPARE TXOPS (IN BRACKETS) UNDER THE TDMA SCHEDULE OF FIG. 2B; AND (B) (COL. 3-9) SDN CONTROL PATHS TRAVERSING THEM (IN BRACKETS) FOR EACH OF THE 7 POSSIBLE LOCATIONS OF THE SDN CONTROLLER.

Link Id	n_l^{sp}	1	2	3	4	5	6	7
14	1(4)	-	-	-	-	-	-	1(6)
3	2(3)	-	3(1)	2(1)	3(1)	2(1)	3(1)	3(1)
1	3(2)	1(6)	-	-	-	-	-	-
11	3(2)	3(1)	3(1)	2(1)	3(1)	2(1)	-	3(1)
12	3(2)	3(1)	3(1)	2(1)	3(1)	2(1)	3(1)	-
6	3(2)	2(3)	2(3)	1(3)	2(2)	-	-	-
2	4(1)	-	1(6)	-	-	-	-	-
4	4(1)	3(1)	-	2(1)	3(1)	2(1)	3(1)	3(1)
5	4(1)	3(1)	3(1)	2(1)	-	-	-	-
7	4(1)	-	-	-	1(3)	-	-	-
8	4(1)	-	-	-	-	-	-	-
9	4(1)	-	-	-	-	1(3)	2(3)	2(3)
10	4(1)	-	-	-	1(3)	2(1)	3(1)	-
13	4(1)	-	-	-	-	-	1(6)	-

Step 4: At this point, we turn to the controller placement question. To address it, we first rank the network links with respect to the number of spare TXOPs that are available to them at the end of step 3. For link l, these equal $n^{sp}_l = \sum\limits_{m\in\mathcal{M}} Y_m x_{lm} - r^d_l - 1$, see (6). Then, for each candidate controller location, we determine the resulting SDN control paths and compute a second ranking of the network links, according to the number of control paths traversing them, as shown in Fig. II. The idea then is to place the controller where the resulting distribution of control traffic load over the WMN links, assumed proportional to the number of control paths traversing each of them, better *matches* the distribution of spare TXOPs over those links. We adopt Kendall's tau (τ) rank correlation coefficient for this purpose [18], *i.e.*, the controller is placed at the node that yields the highest τ value between the two rankings.

Step 5: The end-to-end delays experienced by control messages over the network paths, can be reduced when those

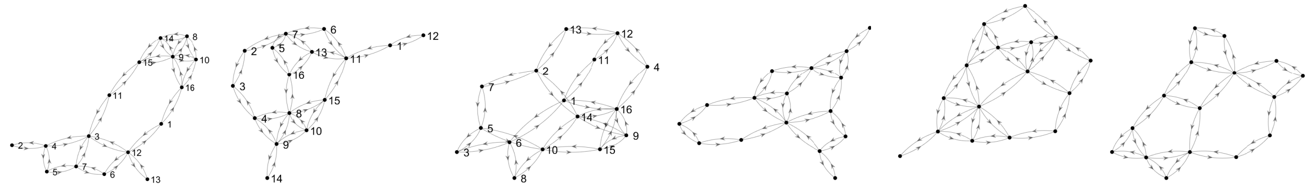


Fig. 3. Six sample WMN topology snapshots: the three leftmost ones are drawn from the Anglova mobility traces and the three rightmost ones are instances of $GRG(16, 0.4)$ in a 1x1 km^2 grid

MCLSs succeed each other in the TDMA frame in the same order that links succeed each other on the control paths. Hence, a greedy algorithm orders MCLSs in a way that is most attractive for all control paths, on average. The algorithm parses the TDMA slots sequentially, relating a pointer to each control path, which initializes it to its first link, and a counter to each MCLS, which is initialized to zero. For each slot, the score of MCLS m equals the number of pointers that point to a link in m. We assign the MCLS with the highest score (or one of those with the highest score if there are more than one) to the slot, we increase its counter by one, we forward the pointers in the respective path one link ahead, and we repeat the process. When a control path pointer reaches the final hop and needs to be forwarded, it is reset to the first link in the path. When the counter of MCLS m equals Y_m, m is removed from further consideration. Each cell of Table III, contains pairs of expected pairwise end-to-end delays when the MCLSs are packed randomly in the frame and when we apply the MCLS ordering taking into account control paths, as shown in Fig. 2b.

TABLE III
EXPECTED PAIRWISE NODE(ID) END-TO-END DELAYS (d_1, d_2) FOR THE NETWORK IN FIG. 2A UNDER A TDMA SCHEDULE WITH RANDOMLY ORDERED MCLSS (d_1) AND ONE WITH MCLSS ORDERED IN LINE WITH SDN CONTROL PATHS (d_2), AS SHOWN IN FIG. 2B.

ID	1	2	3	4	5	6	7
1	0,0	21, 21	7,7	22,22	18,14	36,17	21,16
2	19,19	0,0	13,13	20,20	16,27	34,30	19,29
3	4.92,4	13,13	0,0	13,13	13,13	31,16	16,15
4	14,14	15,15	13,13	0,0	13,13	32,28	17,17
5	18,18	33,32	13,13	13,12	0,0	13,9.63	7.56,7
6	26,25	41,40	21,21	19,18	13,13.5	0,0	26,26
7	21,20	36,37	16,16	14,14	5.2,5.2	21,17	0,0

V. NUMERICAL EVALUATION

A. Methodology

1) WMN topology: We assess our solution for the JCPTS problem over a set of snapshots capturing the WMN state at different points in time, namely a combination of the WMN topology (modeled as a directed graph) and the data traffic mix served by the WMN. Regarding the WMN topology, we experiment with two types of graphs, as shown in Fig. 3: (a) Geographical Random Graphs GRG(N, r) [19], whereby N nodes are randomly placed within a square grid and links are added between each pair of them as far as they lie within distance r from each other; and, (b) snapshots extracted from the publicly available Anglova mobility traces [3]. Capturing the deployment of mechanized battalions during tactical operations, they feature strong correlation in the nodes' mobility patterns.

2) Traffic and TDMA parameters: For each of those graphs, we generate 400 different traffic flow sets (traffic mixes) comprising inelastic and elastic flows. The default traffic split is 80%- 20%. Flow source and destinations nodes are picked randomly, rate requirements (in slots) of inelastic flows are uniformly sampled in [1,4] and their routing paths are computed as shortest paths over the graph. The TDMA frame runs at rate R_T = 1.5 Mbps and consists of N_s = 75 slots of 20kbps each. In our experiments, we distinguish between moderate- and high-demand scenarios, depending on how many TDMA slots are spare after we allocate the requested slots to the inelastic flows and K slots (see section II-D) to each elastic flow. Hence, spare slots vary from 15 to 30 under moderate-demand scenarios, whereas they do not exceed 5 slots under high-demand ones.

3) Performance metrics: The main two quantities of interest are the expected communication delays (in TDMA slots) experienced by messages on SDN control paths from the SDN Controller to the other 15 switches (CTR-SW delay) and the sum of (beyond-the-minimum) rates that are allocated to elastic flows in the TDMA frame. Where appropriate, we also report spare slots and TXOPs at TDMA frame level.

Our experimentation has been conducted in the MATLAB® environment.

B. Results

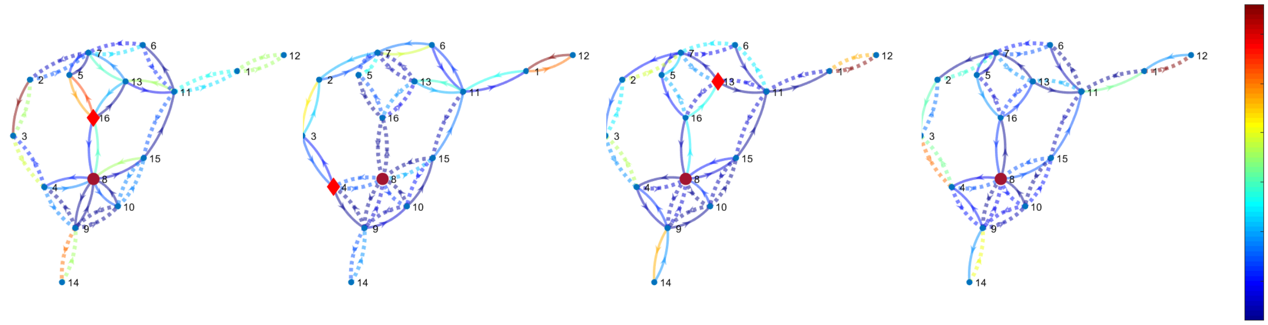
1) Sensitivity of the algorithm to data flows: The de facto approach to the SDN controller placement problem is to select the node that minimizes the shortest-path distance, either worst-case or average, towards other nodes. As a result, the controller tends to be located at a central network node[4].

With our approach the controller's location depends strongly on the data flow mix served by the WMN and the way this is routed through the network. We can see this more clearly in Fig. 4, where we have picked one WMN graph and generated

---

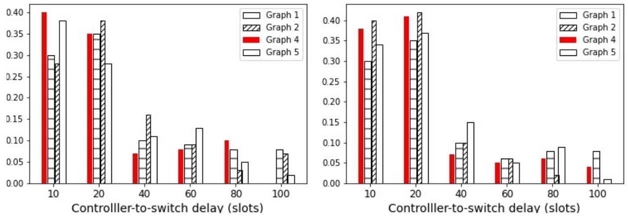
[3] Available: https://anglova.net

[4] Indeed, the node that minimizes the average hopcount towards the other network nodes is the node exhibiting the maximum closeness centrality [20].



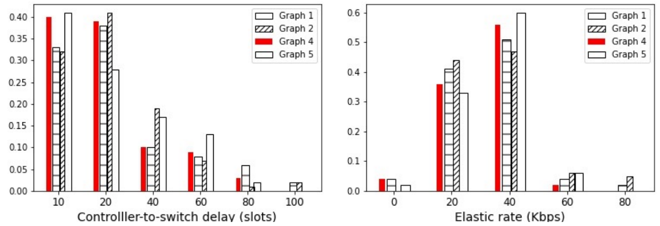
perimetrically routed centrally routed uniformly routed across the WMN

Fig. 4. SDN Controller placement as a function of data traffic routing patterns (dashed lines) in a sample WMN topology snapshot (2nd from left in Fig. 3). Links are coloured according to their spare TXOPs in the TDMA frame and dashed if traversed by data flows. Diamond and circular markers point to the location of the SDN controller under our algorithm and the de facto TDMA-unaware approach.



(a) moderate demand (b) high demand

Fig. 5. CTR-SW delay differences between our algorithm and the TDMA-unaware controller placement with data traffic prioritization



(a) CTR-SW delay (b) bonus rate per elastic flow

Fig. 6. CTR-SW delay differences and elastic flow rate differences between our algorithm and the TDMA-unaware controller placement with SDN control traffic prioritization: high demand

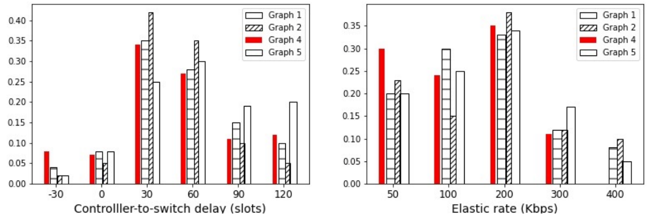
four distinct traffic patterns through it. In all four cases, links are colored in the blue-red scale depending on their spare TXOPs in the TDMA frame, which range from 0 to 30. Now the controller (diamond marker) tends to be located at nodes surrounded by links with more TXOPs in frame. The selected node is different each time and typically not the most central one (circular marker).

2) Performance of the algorithm: We compare our algorithm against two alternative solutions. In both, the SDN controller is placed at the node that minimizes the average minimum hopcount to all other nodes, independently from the TDMA scheduling process. The TDMA scheduler executes the slot allocation policy in section II-D but differentiates from our algorithm in the management of spare slots. Under the first alternative, fully prioritizing the data traffic, it distributes all spare TDMA capacity to TCP flows and serves SDN control messages best-effort through the spare TXOPs in each slot. Under the second alternative, which fully prioritizes the SDN control messages, any spare TDMA capacity is granted to SDN traffic and only spare TXOPs that emerge for links not in SDN control paths are made available to elastic traffic. These two approaches represent two extremes in the way spare capacity may be distributed between data and SDN control traffic by the TDMA scheduler.

We compare our algorithm against those two alternatives over 4 WMN graphs and 3200 flow sets (800 sets/graph, in two sets of 400 for moderate and high demand, respectively.

(a) Comparison with TDMA-unaware controller placement and data traffic prioritization: Intuitively, our algorithm achieves lower CTR-SW delays as depicted in Fig. 5. Whereas the treatment of the elastic flows is the same and the rates allocated to them are identical, our algorithm places the SDN controller at a node that can better leverage the spare TXOPs in the frame. Histograms of mean delay differences between our algorithm and its alternative in Fig. 5 suggest that CTR-SW delays under our algorithm are always smaller, for both graph types, and that the delay savings exceed half a frame duration for every third data flow. The largest difference recorded was 105 slots (220 slots with our algorithm against 325 slots with the alternative). Notably, distributions of the delay difference are similar under both moderate and high traffic load. However, the average delay reduction across all (topology, traffic mix) snapshots is moderately higher under high demand (23% *vs.* 18%).

(b) Comparison with TDMA-unaware controller placement and control traffic prioritization:. In Fig. 6 we plot distributions of the CTR-SW delay reduction and the gains in extra rate that can be ensured for elastic flows with our algorithm under high demand. The plots mark a dominance relationship, *i.e.*, our algorithm outperforms its alternative in both performance metrics. Hence, in such resource-constrained settings, the smart selection of the SDN controller placement by our algorithm is more rewarding than the pure allocation for spare slots to SDN control paths. There are always delay savings, even if they are overall smaller than in Fig. 5b.



(a) CTR-SW delay (b) bonus rate per elastic flow

Fig. 7. Negative CTR-SW delay differences and positive elastic flow rate differences between our algorithm and the TDMA-unaware controller placement with SDN control traffic prioritization: moderate demand

TABLE IV
TDMA-AWARE *vs.* TDMA-UNAWARE SDN CONTROLLER PLACEMENT: AVERAGE DELAY SAVINGS AND ELASTIC FLOW RATE GAINS

TDMA-unaware controller placement	**CTR-SW** moderate demand	high demand	**bonus rate** moderate demand	high demand
spare slots → data	18%↓	23%↓	-	-
spare slots → SDN	27%↑	11%↓	62%↑	20%↑

The alternative policy becomes, expectedly, more rewarding in terms of CTR-SW delay under moderate demand, when there is much more spare TDMA capacity to take advantage of. Fig. 7a now plots the *reverse* delay difference (delay under the alternative minus the delay under our algorithm) showing that the generous-to-SDN alternative yields lower delays in all but a very few cases. Yet, this happens at the expense of elastic flows that stick to the minimum K-slot allocation, while our algorithm lets them cumulatively gain up to 400Kbps more in a TDMA frame of 1.5Mbps rate, as shown in Fig. 7b. Table IV summarizes how our algorithm compares with the two TDMA-unaware approaches to the SDN controller placement problem in terms of average values. At high demand, our algorithm performs at least as well as either of the two approaches with respect to both metrics. At moderate demand, it realizes a more favorable trade-off between what data flows get and the responsiveness at the SDN layer.

VI. CONCLUSIONS

In this paper, we have considered the placement of the SDN controller in a TDMA-scheduled wireless multihop network and formulated it as a joint controller placement and TDMA scheduling (JCPTS) problem. We have proposed a heuristic algorithm for solving the problem and experimentally evaluated it against policies that determine the location the controller independently from the TDMA scheduling decisions. Our results suggest that our, essentially cross-layer, approach trades better the rate that can be allocated to data traffic with the responsiveness of the SDN control layer. Whereas the quantitative aspects of this comparison are sensitive to specific algorithmic choices we made, *e.g.*, for the TDMA scheduling, we expect its qualitative trends to persist more generally.

A natural extension of this work is to consider larger WMN networks with multiple SDN controllers, each having associations with a subset of the network nodes. That would technically require to include the controller-to-controller and controller-to-switches message overhead into the optimization problem formulation and solution.

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] I. F. Akyildiz, S.-C. Lin, and P. Wang, "Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation," *Computer Networks*, vol. 93, pp. 66–79, 2015.
[2] N. Bizanis and F. A. Kuipers, "SDN and Virtualization Solutions for the Internet of Things: A Survey," *IEEE Access*, vol. 4, pp. 5591–5606, 2016.
[3] P. Bellavista, A. Dolci, and C. Giannelli, "MANET-oriented SDN: Motivations, Challenges, and a Solution Prototype," in *Proc. 19th IEEE WoWMoM*, Chania, Greece, June 2018, pp. 14–22.
[4] L. Kleinrock and J. Silvester, "Spatial reuse in multihop packet radio networks," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 156–167, 1987.
[5] 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.
[6] 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.
[7] G. Wang *et al.*, "A k-means-based network partition algorithm for controller placement in software defined network," in *Proc. IEEE ICC*, Kuala Lumpur, Malaysia, May 2016.
[8] A. Dvir, Y. Haddad, and A. Zilberman, "Wireless controller placement problem," in *Proc. 15th IEEE CCNC*, 2018, pp. 1–4.
[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. 40th IEEE MILCOM*, Washington, USA, 2022.
[11] K. Poularakis *et al.*, "Flexible sdn control in tactical ad hoc networks," *Ad Hoc Networks*, vol. 85, pp. 71–80, 2019.
[12] T. Clausen *et al.* The Optimized Link State Routing Protocol Version 2. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7181
[13] J. F. Loevenich, R. R. F. L. Paulo H.L. Rettore *et al.*, "A bayesian inference model for dynamic neighbor discovery in tactical networks," *Elsevier Procedia Computer Science*, vol. 205, pp. 28–38, 2022.
[14] 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.
[15] 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.
[16] 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.
[17] V. V. Vazirani, *Approximation algorithms.* Springer, 2001.
[18] M. G. Kendall, "The treatment of ties in ranking problems," *Biometrika*, vol. 33, no. 3, pp. 239–251, 1945.
[19] M. Penrose, "Random Geometric Graphs," *Oxford University Press*, 2003.
[20] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.