Wireless Networks
https://doi.org/10.1007/s11276-019-02077-5

The controller placement problem for wireless SDN

Amit Dvir[1] · Yoram Haddad[2] · Aviram Zilberman[1,2]

Abstract
The software defined networking paradigm decouples the network's control logic (the control plane) from the underlying routers and switches (the data plane), promoting centralization of network control. The controller placement problem is threefold in nature: the number of controllers to be placed in a network, the locations of these controllers and the assignment function of controllers to switches, with all of them important for the design of an efficient control plane. Most of the existing literature focuses on the placement problem assuming the medium between the controllers and the switches is wired. In this paper, we present a novel strategy to address the controller placement problem, which protects the latency, link failure probability and transparency in the case of a wireless SouthBound interface. We model the problem of determining the placement of wireless controllers in software defined networking. For this purpose, we present a heuristic solution, based on the simulated annealing genetic algorithm, which provides a fast and efficient solution.

Keywords Software defined networking · Controllers placement · SouthBound wireless medium · Simulated annealing heuristic · Wireless control plane · WiFi SDN · WiFi SBi · Wireless controller

1 Introduction

In SDN, the complexity of generating forwarding rules is offloaded to the controllers. In this way the wide view of the network gives the control plane the ability to manage the entire network resources and also the functional behavior of the network. This innovative paradigm has many advantages but also drawbacks; for instance, the control plane is now a bottleneck [16].

Figure 1 shows the classic SDN architecture [27]. The SouthBound interface (SBi) is the communication interface between the control plane and the data plane. The data plane is logically divided into clusters A, B and C, which are assigned with controllers A, B and C, respectively.

In the context of SBi optimization, one of the major (and still open) issues is the controller placement problem (CPP) which was first defined by Heller et al. [12]. It is a dominant research issue in the architecture of the control plane. The CPP focuses on the structure of the control plane, namely the number of controllers, their locations and the assignment function of the data plane switches to controllers. For evaluating a solution for the CPP, the average latency on the SBi is one of the most dominant metrics that are considered.

The SDN paradigm has implications for various types of communication networks between the switches and the users. However, the rapidly growing networks require a flexible architecture [26] between the controllers and the switches (SBi) which cannot be granted by wired networks.

The wireless medium can provide the required flexibility in installing the network infrastructure (SBi) and adjusting it to changes of loads. However, the shared wireless medium behaves differently than the wired one since it is vulnerable to radio interference, noise, fading signals and other RF phenomena. Since most of the existing solutions focus on wired SDNs, the latency is based on the Euclidean distance of the multi-hop path between the switch and its assigned controller.

However, models for latency in wireless networks require a good understanding of the wireless propagation effects and are generally very involved such as probabilistic models like Rayleigh or Rice's fading models [24].

✉ Amit Dvir
amitdv@g.ariel.ac.il

[1] Department of Computer Science, Ariel Cyber Innovation Center, Ariel University, Ramat Hagolan St., 40700 Ariel, Israel

[2] Jerusalem College of Technology, Jerusalem, Israel

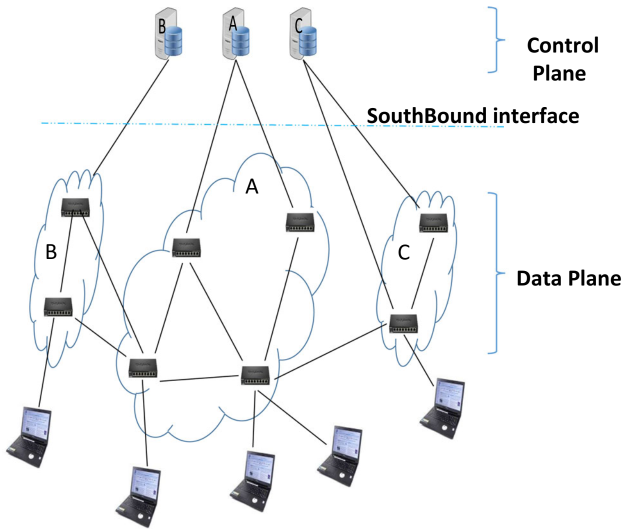

Fig. 1 Classic SDN

In addition, the performance of the wireless network is measured by factors that do not exist in a wired network, such as link failure probability that is negligible in wired networks.

Figure 2 shows a wireless SDN (WSDN) architecture based on Wi-Fi technology where the data plane Access Points (APs) are logically divided into clusters A, B and C and are assigned with controllers A, B and C, respectively. The SBi consists of the set of wireless links between controllers A, B and C and their assigned APs in groups A, B and C, respectively. The communication between the controllers as well as the links between the controllers and the APs are all Wi-Fi based. The inter data plane links may

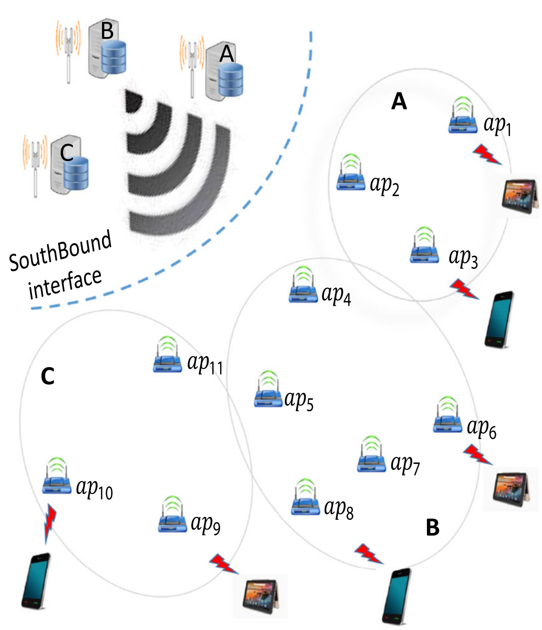

Fig. 2 Wireless SDN

be wireless too. Some research has already proposed a wireless control plane architecture in which inter control plane communication is wireless and the communication on the SBi is also wireless [1–3, 20] but none of them investigated the influence of the SBi wireless metrics on the solution of the CPP.

In this paper, we investigate the wireless controller placement problem(WCPP), taking into consideration the complexities of the wireless medium between the control plane and the data plane. We consider an SDN architecture in which the controllers communicate with the APs of the data plane through a Wi-Fi channel based on a CSMA protocol (see Fig. 2).

In order to investigate the efficiency of the wireless control plane, we propose an objective function which is a weighted sum of three objectives: latency, link failure probability and transparency. The transparency is a new metric defined as the marginal latency on the data plane caused by the co-channel interference that is added by the control plane. We show by simulations the relation between these metrics and the initial controllers placement. For the constraints, we model the average throughput on the SBi and a few other metrics which are not specific to the wireless medium.

Therefore, our main contributions are the following:

1. An objective function for the WCPP which is a weighted sum of three metrics.
2. A model for the metrics of the WCPP that are specific to the wireless SDN, e.g. propagation latency, throughput and link failure probability on the SBi.
3. We propose a new metric for the WCPP, namely the transparency defined as the marginal data plane average latency caused by the additional interference that the controllers add to the data plane.
4. A heuristic algorithm for solving the objective function which achieved very good results.

In this paper, we extend our previous work [8]. We extend the objective function by taking the transparency from the constraints to the optimization. This makes the objective function harder to solve but gives a more effective solution for the placement problem. Moreover, we added the controller's capacity constraint and the number of ports constraint. We also propose a heuristic algorithm for solving the objective function that is based on the simulated annealing algorithm. The proposed heuristic allows adding arbitrary objectives into the evaluation and is not limited with respect to the number of objectives that are taken into account during optimization [19]. We also extend the simulations to evaluate the extended objective function and algorithm.

The remainder of this paper is organized as follows. In Sect. 2, we list the existing solutions for the wired and wireless SDN placement problem. In Sect. 3, we introduce the wireless controller placement problem and propose a detailed model description. In Sect. 4, we present the algorithm for solving the objective function, while in Sect. 5 we show the simulations that we have conducted and discuss the outcomes. We conclude with a few remarks and suggestions for further research in Sect. 6.

2 Related work

The CPP occupies a major role in the control plane and SBi optimization. The performance and quality of service of the control plane can be measured by various metrics, such as the latency of packets transmission on the SBi and the resiliency of the control plane.

In this section, we review the state of the art in solving the CPP. The difference between the various solutions is mainly in the investigated metrics that affect the control plane's performance. Therefore, we focus only on the solutions for the CPP that investigate the relation between these metrics and the control plane's performance. Comprehensive surveys of the CPP solutions can be found at [13, 29, 33].

The CPP for SDN was first introduced by Heller et al. [12]. They defined the CPP and the metrics for evaluating a solution for the initial placement of the control plane. The average propagation latency on the SBi, based on Euclidean distance, was the main objective. Yao et al. [34] showed that the messages capacity of the controller affects the performance. Therefore, they added the capacity constraint to the objective of lowering the latency. Sallahi and St-Hilaire [28] proposed a model for the financial cost of expanding the control plane or installing a new one. Their objective was to minimize the cost, including the cost of installing or removing network elements such as controllers and cables. Vizarreta et al. [31] proposed a new way of calculating the propagation latency on the SBi. They added switches-controllers backup paths to improve the resiliency of the control plane, and claimed that these backup paths should be taken into consideration in calculating the response time of the controllers.

Hu et al. [14] proposed the objective of minimizing the energy consumption of the network that serves for the control traffic under the constraints of the delay of control paths and the load of the controllers. Killi and Rao [17] proposed a controller placement strategy that considers reliability, capacity of controllers and also plans ahead for controller failures to avoid repeated administrative intervention. More recently, Tanha et al. [30] proposed a solution for the resilient CPP with the objective of maximizing the control plane resiliency. For this purpose, they take both the switch-controller and inter-controller latency requirements and the capacity of the controllers into account to meet the traffic load of switches.

All aforementioned works on the CPP, although providing valuable insights into performance optimization of latency, resiliency and reliability, have one common limitation: all of them considered only wired SDNs where the SBi is wired. When we take out the wires of the SBi, and use a wireless channel, the architecture of the SBi is totally different. Therefore, the available approaches in these works cannot be readily adopted for solving the CPP for wireless SBi, namely the WCPP.

So far, such a problem remains almost untouched. The use of a wireless link in the SBi raises many new research issues with regard to SDN. Therefore, the WCPP is a different problem than the classic CPP. The challenges and opportunities of WSDN were proposed by Chaudet and Haddad [6]. The WCPP was introduced by Abdel-Rahman et al. [1, 2]. They proposed two different deterministic solutions for the wired CPP focusing on the load of control messages on the controllers. Their objective was to minimize the number of SDN controllers constrained by the response time of the controllers. Based on these models, they introduced the WCPP and proposed an objective function that minimizes the number of SDN wireless controllers. Their proposed model is based on a TDMA mechanism where both the control plane and the SBi are wireless. However, they did not model the wireless channel and did not show the effect of the wireless metrics on the solution for the WCPP.

Johnston and Modiano [15] considered the impact of delayed state information on the performance of centralized wireless scheduling algorithms and proposed a dynamic controller placement framework, in which the controller is relocated using delayed queue length information at each node, and where transmissions are scheduled based on channel and queue length information.

Liu et al. [20] proposed a joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated networks and showed the advantage of a wireless control plane. Qin et al. [25] proposed an SDN-based fog architecture at the network edge to achieve an elastic resources and services to end-users, where the processing capacity resides at the network periphery as opposed to traditional data-centers.

Faragardi et al. [9] proposed an optimization for deployment of controllers and sinks in a wireless sensor network. Their objective is to optimize the location of sinks and controllers in the network, subject to reliability and timeliness as the prominent performance requirements in time-critical Internet of Things (IoT) systems through ensuring that each sensor node is covered by a certain number of sinks and controllers. In Wireless Sensor Networks (WSN), a sink node gathers data from neighbor nodes and sends it to the outside world via a gateway. The innovation of the proposed scheme is in considering sink placement and controller placement together. Fog computing is a promising alternative to host the SDN controller for avoiding the latency imposed by the Cloud computing. Thus, a Fog layer is a suitable location for hosting the SDN controller while guaranteeing timeliness.

Ashraf [4] proposed an algorithm for the CPP in Software-Defined Wireless Mesh Networks (SDWMNs). The proposed algorithm places the controllers at optimal positions in the network so as to minimize interference compared to random deployment of controllers. Yet they did not model the radio interference and did not show the influence of the interference on the solution for the CPP. In the domain of satellite networks Wu et al. [32] proposed the framework and mechanism of Software-Defined Satellite Networking (SDSN) [11] and also defined the dynamic controller placement problem (DCPP) as well as static controller placement problem (SCPP). However, these works did not consider the building blocks of the wireless channel. Namely, they did not consider the impact of the metrics of the wireless channel on the placement problem.

Recently, a new paradigm has emerged, named as software defined vehicular network (SDVN) which applies the concept of software defined networking (SDN) in vehicular ad-hoc network (VANET). This paradigm comes at a cost of higher operational delay because, the controllers are placed far away from the data plane in the existing SDVN architectures. Liyanage et al. [21] proposed, to bring the control plane down to Road Side Unit (RSU), and introduced a hierarchical distributed controller architecture where the top tier of controllers is regionally distributed and the bottom tier of controllers is placed in several selected RSUs closer to the vehicles. Thus, the latency induced by the system decreases. In addition, they presented a controller placement model for the RSU level controllers. This model is based on the p-median problem with the delay and the significance of the RSU location as the factors to achieve the optimization heuristically as an integer quadratic programming problem. For the latency calculations the wireless channel, which is based on a CSMA protocol, is modeled and their simulations show the influence on the latency metric on the placement. However they considered only the latency metric. Our work is different in two ways: first, we consider the latency and two other metrics which are the link failure probability and the transparency. Second, in our model we consider a wireless medium also between the controllers.

We conclude that the research of the CPP has started to penetrate to the wireless networks. Yet, most of state of the art works do not consider the building blocks of the wireless channel. In this research, we show the significance of modeling the wireless metrics on the SBi for an efficient solution of the WCPP.

3 System model

In this section, we first describe the architecture of the proposed network and then elaborate the WCPP.

3.1 Wireless controller placement architecture

In this paper, we consider a wireless SDN network architecture. As depicted in Fig. 2, the architecture is composed of two logical parts which are the data plane and the control plane. In the data plane, Mobile Stations (MS) connect to the network through APs. The APs are located randomly, since usually Wi-Fi networks are dense and deployed in urban areas, therefore the random spread of APs is reasonable. The APs communicate with other APs through a Wi-Fi based medium. In the control plane, there are numbers of SDN controllers which communicate with other controllers and with the data plane APs through a Wi-Fi based medium. We assume an unsaturated CSMA mechanism for all wireless links, and all the network paths are 1-hop, namely only direct paths from two network nodes are considered.

We further investigate the initial placement of the control plane. Obviously, the network loads change dynamically, and therefore the placement of the controllers has to adapt to the changes after the first initialization. The process of adaptation including the assignment of data plane APs to controllers, and the number and location of the controllers, is a different aspect of the CPP that is not in the scope of this research. Since moving a controller or changing the number of controllers is a heavy operation, therefore substantial parameters, such as the location of the controllers, do not change frequently after the initialization of the control plane. Hence, the initial placement, which is the domain of this research, is important. In addition we do not consider the MS since their locations change frequently, and therefore have less importance on the initial placement.

3.2 Wireless controller placement problem

Consider a network represented by an undirected graph G $(\mathbb{V}, \mathbb{E})$ where $\mathbb{V}$ is the set of network nodes controllers and APs. $\mathbb{E}$ is the set of wireless links, including the links between the controllers, the links between all APs and all links on the SBi, namely the wireless links between the APs and their assigned controllers. The set of k controllers is denoted by C where $C \subseteq \mathbb{V}$ and $C = \{c_1, \ldots, c_k\}$. The set of all APs is denoted by $A_p = \{ap_1, \ldots, ap_m\}$ where m is the number of APs and $A_p \subseteq \mathbb{V}$. Each controller c_i is placed at a location denoted by $\psi(c_i)$. A placement of controllers is denoted by:

$$\Omega = \{(c_1, \psi(c_1)), (c_2, \psi(c_2)), \ldots, (c_k, \psi(c_k))\} \quad (1)$$

The set of all possible placements of controllers is denoted by $\widehat{\Omega}$. Denote the controller assigned to AP ap_i by $\xi(ap_i)$. Denote by $S_\xi(c_i)$ the set of all APs that are assigned to controller c_i. An assignment function of access points to controllers is denoted by:

$$S = \{(c_1, S_\xi(c_1)), (c_2, S_\xi(c_2)), \ldots, (c_k, S_\xi(c_k))\} \quad (2)$$

The set of all possible assignments of controllers to switches is denoted by $\widehat{S}$.

$$\bigcup_{i=1}^{k} S_\xi(c_i) = A_P \quad (3)$$

$$\forall i, j \in \{1\ldots k\}, i \neq j \quad S_\xi(c_i) \cap S_\xi(c_j) = \varnothing \quad (4)$$

Equations 3 and 4 make sure that each AP is assigned to exactly one controller.

A solution for the WCPP is a set $\{k, \Omega, S\}$ which consists of three elements:

1. minimizing the number of controllers k
2. optimizing the location of the controllers, Ω
3. the assignment function S of all APs to k clusters.

For example, consider the wireless network in Fig. 2:

$C = \{c_1 = A, c_2 = B, c_3 = C\}, A_p = \{ap_1, \ldots, ap_{12}\}$. Controller A is assigned with the group A of APs, Controller B with group B and controller C with group C. The set $\mathbb{E}$ is the sum of three groups of wireless links:

1. Wireless links between the controllers.
2. Wireless links between the APs.
3. Wireless links between each AP and its assigned controller.

The wireless SBi denote the wireless link between two network nodes x and y by (x, y). The set of wireless links on the SBi is defined as

$$\Lambda_{SBi} = \bigcup_{i=1}^{m} (\xi(ap_i), ap_i) \quad (5)$$

For example, the wireless SBi in Fig. 2 is

$$\Lambda_{SBi} = \left\{\bigcup_{i=1}^{3}(A, ap_i)\right\} \cup \left\{\bigcup_{i=4}^{8}(B, ap_i)\right\} \cup \left\{\bigcup_{i=9}^{11}(C, ap_i)\right\}$$

A list of abbreviations can be found in Table 1.

3.3 Placement metrics

The performance of the control plane is the objective of the CPP. Obviously, the structure of the network influences the control plane's performance. For example, we would expect the response time of a controller to be shorter if it is close enough to the data plane, but, on the other hand we would need more controllers to cover all the data plane, and this is not cost effective. The trade off between the different metrics in a wired SDN is well investigated, but the wireless channel is different and much more complex.

For example, the latency on a wired link is usually measured by the Euclidean distance, but in a wireless link, attenuation and fading are involved. Moreover, since a wireless channel suffers from interference, retransmissions and link failure, therefore, the latency is affected by other metrics than the Euclidean distance.

On the other hand, the proximity of the controllers to the data plane decreases the link failure probability on the SBi, but this increases the interference on the data plane, and the average latency within the data plane increases.

The abbreviations that are used for the metrics modeling are depicted in Table 1.

3.3.1 Average latency

We assume a Rayleigh fading channel with no line of sight, since usually Wi-Fi networks are deployed in urban areas. A transmission from node i to node j is successful if the SINR γ_{ij} is above a threshold Θ. Otherwise the interference that this signal creates is not considered. The SINR is given by

$$\gamma = \frac{Q}{N_0 + I} \quad (6)$$

Q is the received power, which is exponentially distributed with mean $\overline{Q}$. Over a transmission of distance d with an attenuation d^α we have $\overline{Q} = P_0 d^{-\alpha}$, where P_0 is the transmit power, α is the path loss exponent. N_0 denotes the noise power, and I is the interference power, namely, the sum of the received power from all the undesired transmitters. Note that even though we consider only the SBi links Λ_{SBi} for the average latency, other links such as intra control plane wireless links are considered for interference calculations.

Theorem 1 [7] *In a Rayleigh fading network with slotted ALOHA, where nodes transmit at equal power levels with probability p, the success probability of a transmission given a desired transmitter–receiver distance d_0 and n other nodes at distances $d_i (i = 1, \ldots, n)$ is*

Table 1 List of abbreviations

$G(V, E)$	Physical network with node set V and link set E, $V = C \cup A_p \cup U$
A_p	Set of all APs
C	Set of all controllers
Ω	A controller placement
$\widehat{\Omega}$	Set of possible controller placements
S	An assignment of APs to controllers
$\widehat{S}$	Set of possible assignments of controllers to APs
c_i	Controller
$\psi(c_i)$	PLacement (location) of controller c_i
ap	AP
$\xi(ap_i)$	Controller assigned to ap_i
$S_{\xi}(c_i)$	Set of all APs assigned to controller c_i
Λ_{SBi}	Set of all wireless links on the SBi
Λ_D	Set of all wireless links within the data plane
Λ_C	Set of all wireless links within the control plane
$\widehat{\Lambda}$	Set of all wireless links in the network
$\overline{\mathbb{P}r}$	Average link failure probability
$L(\theta)$	The capacity of controller θ
N_{ports}	Number of ports in a single controller
Q	Received power
N_0	Noise power
F	Frame size
R	Transmission rate
$G_{j,i}$	Channel gain on the link from node j to node i
$\sigma^p(i)$	Number of packets AP i sends
$\mu^c(c)$	Number of packets controller c can process
$\gamma_{i,j}$	SINR on the link from node i to node j
Θ	SINR threshold
W	Bandwidth
α	Path loss exponent
β	Packet size
$DIFS$, $SIFS$	Time intervals defined in 802.11
t_{bf}	Backoff time
t_{ack}	Acknowledge time
$Pr_{s,d}$	Link failure probability on the link from source s to destination d
$\lambda_1, \lambda_2, \lambda_3$	Coefficient parameters

$$\mathbb{P}_{s|d_0 \dots d_n} = exp\left(-\frac{\Theta N_0}{P_0 d_0^{-\alpha}}\right) \cdot \prod_{i=1}^{n}\left(1 - \frac{\Theta p}{\left(\frac{d_i}{d_0}\right)^{\alpha} + \Theta}\right) \tag{7}$$

where P_0 is the transmit power, N_0 the noise power, and Θ the SINR threshold.

Remark Theorem 1 applies also for the proposed architecture. Even though CSMA is different than ALOHA. However Eq. 7 expresses the fact that the transmission will be successful and this can happen only if no two concurrent transmissions occur. Not having two simultaneous transmissions is also required for successful transmission in CSMA. In other words, this equation comes after the listen before talk mechanism. However, it still does not guarantee complete protection from collisions since even in CSMA if sensing of the medium is performed exactly at the same time, then it will still result in a collision.

The set of all wireless links between the APs is defined as

$$\Lambda_C = \bigcup_{i,j \in \{1 \dots k\}, i \neq j} (c_i, c_j) \tag{8}$$

The set of all wireless links within the data plane is defined as

$$\Lambda_D = \bigcup_{i,j \in \{1 \ldots |A_P|\}, i \neq j} (ap_i, ap_j) \tag{9}$$

The set of all wireless links in the network is defined as

$$\widehat{\Lambda} = \Lambda_C \cup \Lambda_D \cup \Lambda_{SBi} \tag{10}$$

Consider a transmitter Tx and a receiver Rx. The set of all potential interfering transmissions are the transmissions on the links $L_P = \widehat{\Lambda} \setminus (Tx, Rx)$. The set A^I for the link (Tx, Rx) is defined as the set of transmitters that transmit on links from L_P, that are within reception range of Rx. More specifically, consider a network node N that is transmitting to AP ap_i such that $N \in \{C \cup AP \setminus ap_i\}$. The total interference received at ap_i from all interfering nodes in A^I is:

$$I_i = \sum_{z \in A^I} P_{zi} \tag{11}$$

where P_{zi} is the received signal power at ap_i from the z th interfering node at distance d_{zi}. As a result of interfering signal power received at ap_i from the interferers A^I, the SINR on the link between network node j and AP_i is:

$$\gamma_{ji} = \frac{P^r_{ji}}{(I_i + N_0)} \tag{12}$$

where P^r_{ji} is the received signal power from node j at AP_i over a distance d_{ji}. The received signal power at ap_i from node j is calculated as follows:

$$P^r_{ji} = P^t G_{ji} d_{ji}^{-\alpha} \tag{13}$$

where G_{ji} is the channel gain characterized by an exponential distribution, i.e. $G_{ji} \sim exp(P^t)$ to account for fading and shadowing effects, and α = 3 is used as the path loss exponent. The total end-to-end delay (latency) for a packet at a link between network nodes j and i can now be calculated as:

$$t_{ij} = \frac{F(bits)}{R_{ij}} \tag{14}$$

where F is the frame size, R_{ji} is the transmission rate, which is determined by SINR γ_{ji} experienced by AP_i when associated with controller j. The mapping between R_{ji} and SINR γ_{ji} in the context of 802.11 is provided by Oni and Blostein [23].

When a frame that is transmitted from AP_j to AP_i experiences collision, the transmission time is extended:

$$\tilde{t}_{ij} = DIFS + t_{bf} + t_{ij} + SIFS + t_{ack} \tag{15}$$

where $t_{bf} = \frac{CW_{max}}{2}$ x Slot-time is the backoff time, $t_{ack} = \frac{1}{r}$ is the time it takes to transmit ACK frame given basic data rate r (e.g. 1 Mbps in an 802.11b network) while SIFS and DIFS are time intervals defined in the 802.11 standard. The maximum contention window (CW_{max}) is 1024. We assume that backoff always occur since the only case where there is no backoff is only during the first packet transmission since even after successful transmission there is a post backoff mechanism in WiFi. Finally, the average latency on the SBi is:

$$\overline{L_{SBi}} = \frac{1}{|\Lambda_{SBi}|} \sum_{(i,j) \in \Lambda_{SBi}} \tilde{t}_{ij} \tag{16}$$

3.3.2 Transparency

The transparency metric is the marginal average latency in the data plane caused by the interference from the controllers, meaning, how the conversation between the controllers and the APs interfering in the data plane due to the fact that all the transmitting nodes use the same wireless medium and technology. The average latency on the data plane is:

$$\overline{L_D} = \frac{1}{|\Lambda_D|} \sum_{i,j \in \{1 \ldots |A_P|\}, i \neq j} \widetilde{t}_{i,j} \tag{17}$$

Note that for the calculation of $\overline{L_D}$ we consider the total interference as defined in Eq. 11. Next we calculate the total interference as in Eq. 11, but we exclude the interference that is caused by the controllers. Thus, the average latency on the data plane denoted by $\overline{L_{D_0}}$ is defined as in Eq. 17.

Therefore the transparency is

$$T = \frac{\overline{L_D} - \overline{L_{D_0}}}{\overline{L_{D_0}}} \cdot 100\% \tag{18}$$

The transparency is a measurement for the overhead by means of data plane latency, for installing controllers.

3.3.3 Throughput

For each two network nodes i and j, the throughput of a slotted unsaturated CSMA with Heterogeneous Traffic (1-hop paths) is:

$$Th_{ij} = \frac{1}{\tilde{t}_{ij}} \tag{19}$$

The average throughput on the SBi is

$$\overline{Th_{SBi}} = \frac{1}{|\Lambda_{SBi}|} \sum_{(i,j) \in \Lambda_{SBi}} Th_{ij} \tag{20}$$

3.3.4 Link failure probability

The scenario assumes a star topology structure where each AP has a single route to its exclusive assigned controller. The channel experiences a Rayleigh fading (no line of sight) and we assume that the link failure probability denoted $Pr_{i,j}$ (i and j are network nodes) is the probability that the SINR would go below a threshold Θ

$$\mathbb{P}r_{i,j} = 1 - \mathbb{P}_{j|d_0 \ldots d_n} \tag{21}$$

where $\mathbb{P}_{j|d_0 \ldots d_n}$ is defined in 7, and i, j are source and destination nodes. The average probability for a link failure on the SBi is

$$\overline{\mathbb{P}r_{SBi}} = \frac{1}{|\Lambda_{SBi}|} \sum_{(i,j) \in \Lambda_{SBi}} \mathbb{P}r_{i,j} \tag{22}$$

3.4 Objective function and constraints

Based on the proposed network architecture and the aforementioned metrics, the objective function for the WCPP can be modeled as can be seen in Eq. 25 where $\widehat{S}$ is the set of all possible assignments and $\widehat{P}$ is the set of all possible placements.

The metrics in the objective function are the average link failure probability $C_1(S, \Omega) = \Omega r_{SBi}$, the average latency $C_2(S, \Omega) = \Lambda_{SBi}$ and the transparency $C_3(S, \Omega) = T$, where $\Omega \in \hat{\Omega}, S \in \hat{S}$. Since the parameters C_1, C_2 and C_3 represent metrics that differ by nature we need the three coefficient parameters λ_1, λ_2 and λ_1.

These coefficient parameters are set by the operator such that constraint (26) guarantees that the number of APs that are assigned to controller c does not exceed c's port capacity N_{ports}. Constraint (27) guarantees that each AP is assigned with exactly one controller. Constraint (28) guarantees that the throughput does not go below threshold $Th_{threshold}$. Constraint (29) guarantees that the number of packets per second that does not match on the AP's look-up table and that must be sent to the controller - σ^p, does not exceed the number of packets/second a controller can process μ^c.

$$0 \le \lambda_i \le 1; \quad i \in \{1, 2, 3\} \tag{23}$$

$$\sum_{i=1}^{3} \lambda_i = 1 \tag{24}$$

$$Min_{S \in \hat{S}, \Omega \in \hat{\Omega}} \quad \lambda_1 \cdot C_1(S, \Omega) + \lambda_2 \cdot C_2(S, \Omega) + \lambda_3 \cdot C_3(S, \Omega) \tag{25}$$

subject to:

$$\sum_{c \in C_{set}} |S_\xi(c_i)| \le N_{ports} \tag{26}$$

$$\forall ap \in A_P |\xi(ap)| = 1 \tag{27}$$

$$\overline{Th_{SBi}} > Th_{threshold} \tag{28}$$

$$\forall c \in C \sum_{ap \in S_\xi(c)} \sigma^P(ap) \le \mu^C(c) \tag{29}$$

Note that these metrics give different aspects of the quality of the wireless SBi: The average latency is a measure for the response time of the controllers and the link failure probability relates to the throughput, since link failures cause retransmissions. The transparency of the control plane is a measure for the marginal average latency added to the data plane that is caused by the interference from the control plane. This is a metric that is exclusive for the WSDN.

The application of SDN concepts in the context of wireless networks poses many challenges. Consider a WLAN where each AP has to make decisions on its modulation format, power and channel based on SINR estimates and in addition has to route data packets to their destination. The wireless channel is noisy and retransmissions occur and therefore, the latency is high. In this case, a fully centralized network architecture such as SDN, imposes strict upper bounds on the latency between the SDN controller and the AP. The control decision should reach the AP before the channel state information, from which the decision request originated, has become obsolete.

Roughly, the latency should be on the order of the coherence time of the channel. Moreover, by the time a control decision is sent to the AP the structure of the network may change, since end users do hand-offs, they leave and join the network rapidly. Thus in wireless networks, it is not always clear as to which point in the design space one should operate.

We claim that the aforementioned metrics behave differently, for example, adding controllers shortens the distance between the AP and the controller and therefore, the link failure probability decreases but then the interference grows and the latency and the transparency increase. Moreover, the Euclidean distance is not as significant as in a wired network since the network signals suffer from fading which depend on probabilistic parameters. Hence, an AP may be assigned to a distant controller since the link to a closer controller suffers from high interference. We show that adding one controller may increase the transparency and at the same time adding two controllers may decrease the transparency since it requires different locations for the controllers or different assignment of APs to controllers. Therefore, the behavior of a wireless control plane is not predictable.

4 Wireless controller placement algorithms

In this section, we propose a solution for the objective function (Eq. 25). The objective function uses three decision variables and four constraints, some of which are functions of probabilistic variables. The placement problem is a K-Median problem which is already proved to be NP-Hard [22]. In this paper, we first use an algorithm that is K-Median based. This algorithm is very slow and inefficient when the network grows. In addition, the K-Median algorithm is based on a hill-climbing search and since the objective function uses three decision variables with different behavior, the function may have several extremum points which make it impossible to solve with such an algorithm. Therefore, we propose a heuristic that is based on the simulated annealing algorithm, called simulated annealing for the wireless controller placement problem (SAWCPP).

4.1 K-Median based algorithm

The previously proposed K-Median algorithm uses as an input a set A of APs located randomly in the plane where A is divided into $k = 3$ clusters such that all the APs in the same cluster are assigned with a controller which is co-located in one of the AP locations in the cluster. Then the K-Median algorithm chooses the clustering and the locations of the controllers such that the sum of weights of all links is minimized. The weight of a link fits with the objective function (Eq. 25) that is the sum of the latency, the probability for link failure on the link and the transparency. The pseudo-code for the proposed algorithm is presented in Algorithm 1. An enumeration process searches for the number k of controllers that would make the best performance according to the objective function. The output of the algorithm is k clusters with medoids. For each cluster, one of each of the cluster's members is assigned to be the medoid such that the sum of all weights in the output graph forest is minimized.

```
Data: K:number of controllers
Result: ControllersPlacement, Clustering, AverageLatency, AverageOutageProbability
initialization:
  networkArray← location of all access points
  Placement ← K-Medoids(K,edge weight←(Latency+OutageProbability))
  if constraintsCheck = True then
  |  return ControllersPlacement, Clustering, AverageLatency, OutageProbability
end
else
  |  return Null
end
```

Algorithm 1: Basic K-Median based enumeration algorithm

The proposed algorithm solves the WCPP for a specific and constant number of controllers but obviously by changing the number of the controllers we may achieve better solution, with lower objective value. Therefore, we extend the algorithm such that we find the number of controllers and their placement that minimize the objective value. The extended algorithm runs the basic algorithm on a different number of controllers and returns the best solution. EWCPP is described in Algorithm 2.

```
Result: Placement : ControllersPlacement,
OptimalControllersNumber, Clustering,
AverageLatency, AverageOutageProbability
initialization:
  K ← maximum number of controllers
  OptimalPlacement ← Inf
  networkArray← location of all access points
  while K > 0 do
  |  Placement ← K-Medoids(K,edge weight
  |    : (Latency+OutageProbability)
  |   if (constraintsCheck = True) and
  |   (Placement < OptimalPlacement) then
  |   |  OptimalPlacement ← Placement
  |   end
  |   K ← K-1
  |
end
return ControllersArray, AverageLatency,
OutageProbability
```

Algorithm 2: Extended K-Median enumeration algorithm - EWCPP

4.1.1 Analysis and discussions

EWCPP in Algorithm 2 is very simple and accurate, but it has a very high computational complexity since it has to enumerate all combinations of k controllers from n nodes.

It can be easily proved that the computational complexity of EWCPP is

$$O(k \cdot n \cdot C_n^k) \quad (30)$$

where k is the number of controllers and n is the number of network nodes.

4.2 Hill climbing with simulated annealing

The K-Median is a basic algorithm for clustering problems and its major drawback is that it cannot handle large graphs. Moreover, as discussed in the introduction, the proposed objective function is complicated and may include several extremum points and even a global minimum which is not an extremum point. Therefore, the K-Median algorithm cannot solve the WCPP efficiently [19].

In order to provide a viable and faster solution, we propose a heuristic in this section. Simulated annealing is a popular choice for finding a global optimum of problems that have large search space and many local optimums. Simulated annealing is a probabilistic method proposed independently by Kirkpatrick et al. [18] and Černỳ [5]. It was inspired by the metropolis algorithm for cooling of materials by slowly lowering the temperature. The main innovation of the algorithm is that worse solutions are also accepted with some probability. This way the algorithm will not stop at a local minimum where there is a different global minimum. This can be done by using a control parameter known as temperature such that the probability of accepting worse solutions decreases with the temperature. Hence, it allows the algorithm to explore the search space at higher temperatures and helps in the convergence at lower temperatures. Furthermore, the probability of accepting worse solutions decreases with the difference between objectives of current and new solutions. The input to the algorithm is the network graph and the annealing schedule. The algorithm returns the best solution encountered during the traversal.

The annealing schedule is limited by the starting temperature T_b and the ending temperature T_e. An I_{max} iterations are made at each temperature and the temperature decrement is Θ. The algorithm begins with an initial temperature and gradually decrements until it reaches an ending temperature. The starting temperature must be selected very carefully, i.e. it should not be too high or too low. Generally, the starting temperature is set to 1 and ending temperature is set to a small value such as 0.00001. In our case, the ending temperature is 10^{-10}. The ending temperature may not be zero because the probability of accepting the worst moves approaches zero as the temperature approaches zero. Next the starting point is chosen randomly. In each iteration, the objective value and the constraints are calculated and compared with the current best solution. A better solution is chosen but a worse solution is also accepted with a probability $\mathbb{P}(Pl', \Theta, T)$ as defined in Algorithm 3, or else another neighbor of the current solution is generated. The process continues until the temperature reaches an ending temperature. The simulated annealing parameters are depicted in Table 2. The pseudo-code for the simulated annealing is given in Algorithm 3.

```
Input: G(V,E), T_b, T_e, I_max, θ
Output: Pl^opt, val^opt
T ← T_b, val^opt ← inf, i ← 1
Pl ← RandomPlacement()
val ← PlacementEvaluation(Pl)
while T ≥ T_E do
    if (ConstraintsCheck(Pl) = true) and (val < val^opt) then
        | Pl^opt ← Pl, val^opt ← val
    end
    Pl' ← Neighbor(Pl)
    val' ← PlacementEvaluation(Pl')
    Θ ← val' − val
    r ← random[0..1]
    if ℙ(Pl', Θ, T) ≥ r then
        | Pl ← Pl', v ← val'
    end
    i ← i + 1
    if completed I_max iterations then
        | T ← T · θ, i ← 1
    end
end
return Pl^opt, val^opt
```

Algorithm 3: Clustering algorithm based on hill climbing with Simulated Annealing (SAWCPP)

One of the most sensitive stages in the simulated annealing algorithm is choosing the perturbation of the current placement. A perturbed placement may consist of a change in controller number or a change in their location. Obviously, this results in a new assignment function of APs to controllers. We perturb the current placement by randomly adding or removing one controller and then choosing a controller and altering its placement using Gaussian

Table 2 Simulated annealing parameters

Parameter	Value
Starting temperature T_b	10^{-4}
Ending temperature T_e	10^{-8}
Number of iterations I_{max}	550
Temperature decrement Θ	0.95

mutation. After the perturbation, all access points are reassigned to their nearest controller using K-NearestNeighbor algorithm. The pseudo-code for the perturbation operator is given in Algorithm 4.

```
Input: G(V,E), P(placement)
Output: P_neighbor
C ← set of all clusters
c_n ← randomly chosen controller
γ ← mutation factor close to zero
Ω ← Gaussian random number generator
Shift controller c_n by γ · Ω
Reassign all access points to controllers using
K-Nearest-Neighbor algorithm
P_neighbor ← new placement
return P_neighbor
```

Algorithm 4: Perturbation operator for SAWCPP

4.2.1 Analysis and discussions

Simulated annealing goes through O(log n) temperature steps. For each temperature, the search examines O(n) attempted and accepted changes. The computation rejects a change of the current tour in O(1) time. If a change is accepted, the average path reversal involves O(n) exchanges. Consequently, the run time T(n) of simulated annealing has the complexity:

$$T(n) = O((n^2 + n) \cdot \log n) \tag{31}$$

Since most steps take place at low temperatures, where most changes are rejected, the $O(n \cdot \log n)$ term is not negligible compared to the $O(n^2 \cdot \log n)$ term.

5 Experimental results

In this section, we show the advantages and disadvantages of solving the proposed objective function with SAWCPP. We compare the performance of SAWCPP and the performance of the K-Median based enumeration algorithm. In addition, we show the results of the SAWCPP in different sizes of SDNs, especially in large SDNs (more than 50 APs).

As mentioned in Sect. 2, to the best of our knowledge there is no other research that models the wireless channel for the optimization of the WCPP. Therefore, the objective of this section is not to compare the performance of the proposed solution with alternative solutions. In this section, we investigate the influence of the wireless metrics on the placement of a Wi-Fi based WSDN.

Moreover, we show the need to consider multiple metrics for an accurate solution for the WCPP. The simulation parameters are depicted in Table 3.

For the simulation we used Intel Core i5-2400 CPU @ 3.10 GHz, 3101 MHz, 4 Cores, running Matlab RS2015a. The solution algorithm uses the Matlab K-Medoid function to cluster the data plane APs.

The proposed SAWCPP algorithm is evaluated on different sizes of networks and compared to the enumeration Algorithm 2. We assumed that the controller software runs on a server with maximum access bandwidth of 10 Gbps and capacity of 7.8×10^6 packets/s to serve 160 bytes packet size according to OpenFlow v 1.2 specifications [10]. All the simulations are performed such that each instance is executed for 10–100 times depending on the complexity of the simulation, and the average results are presented for statistical reliability. First we compare EWCPP and the SAWCPP.

Figure 3 shows the results of EWCPP and SAWCPP in different network sizes. On average the SAWCPP requires 1.3 controllers more than the EWCPP. This is due to the fact that SAWCPP uses an up-hill type search for an optimal objective value in contrast to the EWCPP that makes decisions from a global point of view. Figure 4 shows that the transparency in the placement of the SAWCPP is 11.67% higher on average than the corresponding transparency of the EWCPP placement.

Figure 5 shows the average link failure probability on the SBi that is calculated by the two algorithms. We observe that for both algorithms the average link failure probability increases with the growth of the network. This is because in dense networks transmissions suffer from higher interference, hence link failure probability increases.

Figure 6 shows the average latency on the SBi that is calculated by both algorithms. We observe that for both algorithms the average latency decreases with the growth of the network. This is because large networks require more controllers and therefore the controllers are closer to the APs.

Table 3 Simulation parameters

Parameter	Value
Network area	$1000 \times 1000\,\text{m}^2$
Controller transmission power	20 dBm
AP transmission power	12 dBm
Mean packet size	160 bytes
SIFS	10 µs
Noise N_0	90 dBm

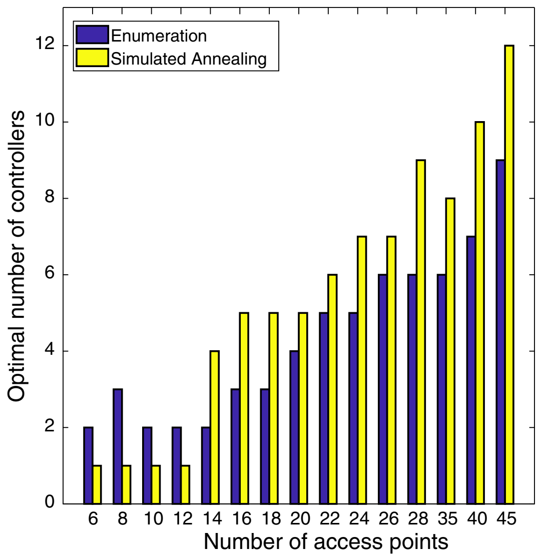

Fig. 3 A comparison of the optimal number of controllers placed by each algorithm

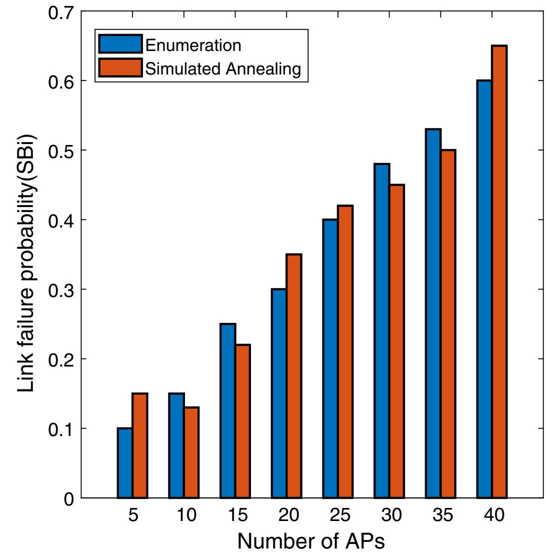

Fig. 5 A comparison of the link failure on the SBi for the placement that computed by each algorithm

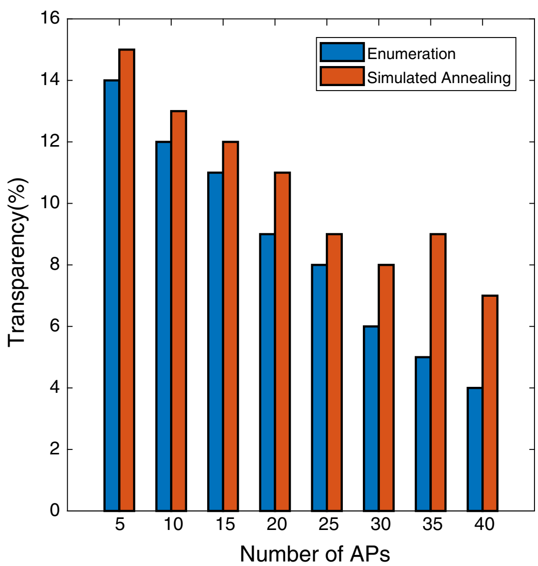

Fig. 4 A comparison of the transparency of the control plane that is placed by each algorithm

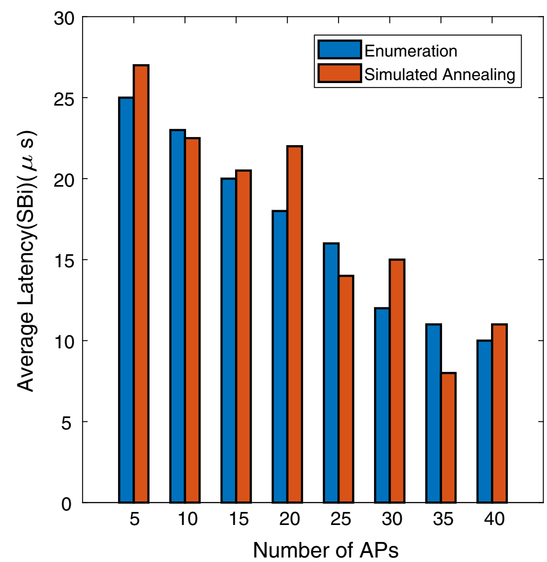

Fig. 6 A comparison of the average latency on the SBi for the placement that computed by each algorithm

This is due to the fact that SAWCPP uses an up-hill type search for an optimal objective value in contrast to the EWCPP that makes decisions from a global point of view. Therefore, even though the number of controllers that is decided by both algorithms is the same, the location of the controllers and the assignment function of APs to controllers is different.

Figure 7 shows the transparency that is calculated using the results of SAWCPP.

We observe that in some cases the SAWCPP results are less effective than the results of the EWCPP (i.e. Figs. 3 and 4. Obviously the EWCPP algorithm which is close to being a brute force algorithm, calculates close to optimal results. However, the cost is the running time of such algorithm. The EWCPP algorithm cannot run effectively on large networks.Therefore, in Fig. 7 the SAWCPP can calculate the transparency for a network with 100 APs, but it is very hard for the EWCPP algorithm.

We conclude that the cost in accuracy we pay by using the SAWCPP is reasonable because of the significant benefit of running time.

Figures 4 and 7 show that the transparency decreases when the network grows. We would expect that the transparency will increase since adding controllers adds

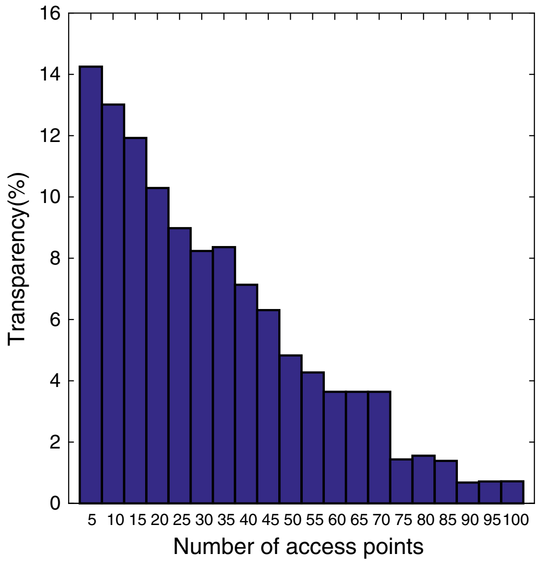

Fig. 7 Simulated annealing algorithm: the transparency of the control plane in different sizes of networks

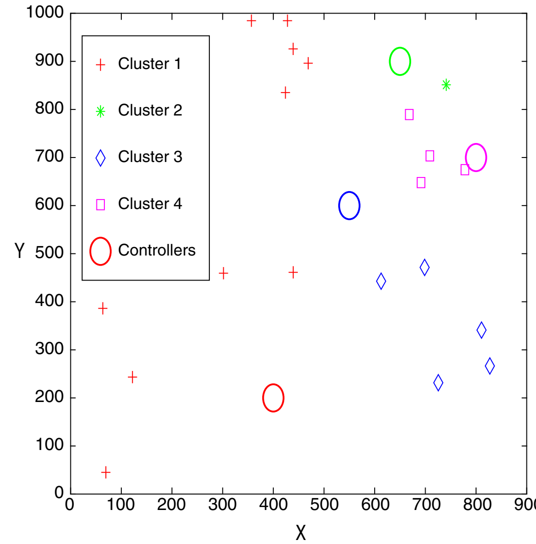

Fig. 9 Simulated annealing algorithm: a placement solution for a network that consists of 20 access points and 4 controllers

interference, nonetheless we observe that the transparency decreases. The reason is that extending the data plane with more APs increases the total interference within the data plane, hence the marginal interference and marginal latency that is added because of each additional controller decreases.

Figures 8 and 9 show an example of placement results of EWCPP and SAWCPP, respectively: 20 data plane APs are located similarly in both figures in a 1000 m^2 area. Each circle represents a controller and the rest of the shapes represent APs in different clusters according to the shape. Note that in a wired network the latency depends mainly on the Euclidean distance, therefore, each cluster is organized such that all the switches and their assigned controller are geographically close. But this is not the situation in a wireless network since the network's behavior is influenced by RF phenomena such as channel fading and the Euclidean distance is less significant. For example, an AP can be assigned to a distant controller to which it has a line of sight, and not to a close controller where the channel is noisy or faded and the SINR is low.

Figure 10 shows the behavior of the energy, that is the objective value relative to the temperature as used in the simulated annealing process. Note that the objective value

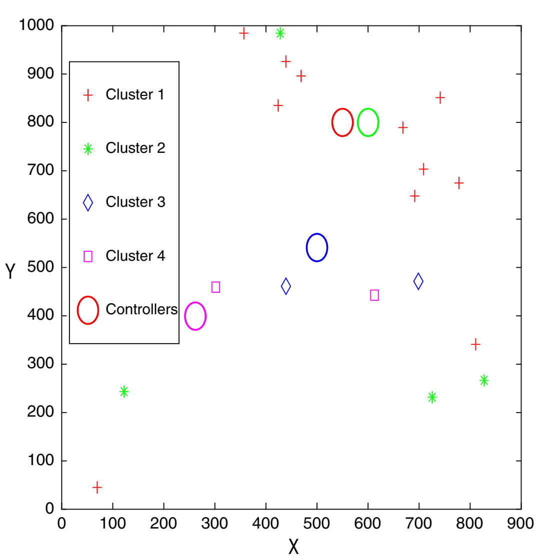

Fig. 8 Enumeration algorithm: a placement solution for a network that consists of 20 access points and 4 controllers

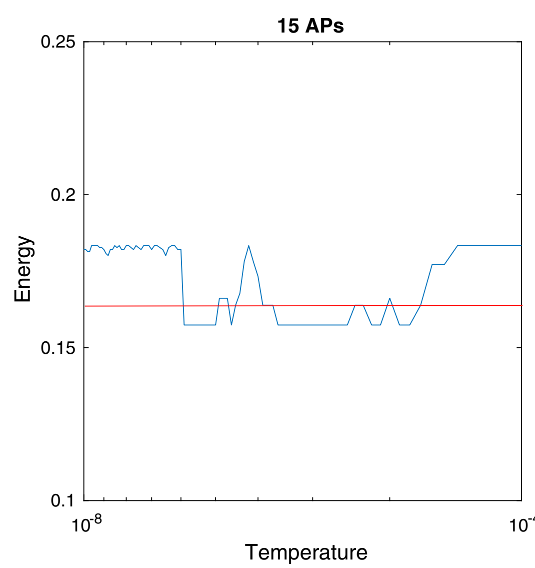

Fig. 10 The energy(objective value) versus the temperature for 15 APs. The horizontal line shows the enumeration algorithm result

Table 4 Simulated annealing versus K-Median

Number of APs	Critical range of temperatures	Global energy minimum	K-Median result
10	$10^{-8}, 10^{-4}$	0.105	0.110
15	$10^{-8}, 10^{-4}$	0.160	0.165
20	$10^{-8}, 10^{-4}$	0.210	0.215

does not depend on the temperature but during the process of the simulated annealing the placement changes, therefore, the objective value changes and the probability of choosing the next placement with worse energy depends on the current temperature. The horizontal line shows the corresponding result of EWCPP. Table 4 summarizes the same comparison but for different sizes of data plane that is the number of APs. The critical range of temperatures shows the range of temperatures in which the energy changes significantly. Global energy minimum is the objective value for the best placement that was calculated by the SAWCPP and the K-Median result is the objective value for the best placement that the EWCPP calculated.

The simulations show the importance of choosing the right parameters for the simulated annealing algorithm, namely the values for T_b, T_e and Θ. We observe that choosing $T_b = 10^{-4}, T_e = 10^{-8}$ is sufficient, which makes the algorithm run for a shorter time. Moreover, the frequent change of the objective value in this range of temperatures requires Θ to be close to 1. In addition, we observe that the behavior of the objective function is much more complex than in a wired network.

The simulation results emphasize the following observations:

- The placement for wired SDN depends mainly on the Euclidean distance between the network nodes. Put differently, a wireless link suffers from RF phenomenon such as radio interference, background noise, fading signals, shadowing, the exposed and hidden terminals, some of which are probabilistic. Therefore, the placement solution depends on the RF behavior of the medium. The simulations show that the clustering of the data plane APs and their assignment to controllers have no prominent relation to the geographical distance and the APs may be assigned to more distant controllers since the link to a close controller may have low SINR.
- Increasing the number of controllers does not necessarily increase performance. Adding controllers reduces the size of the clusters. Moreover, the distances between the controllers and the access points are shortened. In a wired network, proximity to the controller lowers the latency and improves the performance, while in a wireless network this adds more interference on the data plane and may increase the transparency. Furthermore, it decreases the SINR within the data plane and the control plane, hence latency increases. On the other hand, the link failure probability may diminish.
- The probabilistic behavior of the Rayleigh fading channel causes unpredictable behavior of the network—adding one controller may lower the performance but adding 2 controllers may improve the performance. Adding controllers changes the clustering of the network and the SINR can change dramatically both ways. Therefore, a minimum number of controllers no longer achieves maximum performance but the opposite may happen.
- Due to the probabilistic behavior of the Rayleigh fading channel, closer APs do not necessarily get better service from the controller as it is in a wired network. Therefore, the structure of the clusters is not proportional with the Euclidean distance from the controllers.
- In larger networks the transparency of the control plane improves.
- The transparency metric is important for the solution of the WCPP. Installing controllers imposes interference on the data plane and changes the SINR, but when the network grows this overhead is less significant and the transparency of the control plane improves. This is a tradeoff that the operator would have to consider when installing a wireless control plane.

6 Conclusions

SDN with a wireless control plane is a challenging new concept that has not yet been thoroughly investigated. WSDN requires different way of thinking than SDN for wired networks. In this paper we introduced the WCPP and showed the complexity of implementing SDN for wireless networks. We modeled the different metrics that affect the solution for the WCPP and proposed an objective function and constraints for the problem. We solved the objective function with two different algorithms and our simulations show the different aspects of the solution. The next step may be considering alternative technologies for the wireless links such as cellular. In addition, the combination of such technologies is also a challenge.

Acknowledgements This research was (partly) funded by the Israel Innovations Authority under the Neptune generic research project. Neptune is the Israeli consortium for network programming.

References

1. Abdel-Rahman, M. J., Mazied, E. A., MacKenzie, A., Midkiff, S., Rizk, M. R., & El-Nainay, M. (2017). On stochastic controller placement in software-defined wireless networks. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, (pp. 1–6) https://doi.org/10.1109/WCNC.2017.7925942
2. Abdel-Rahman, M. J., Mazied, E. A., Teague, K., MacKenzie, A. B., & Midkiff, S. F. (2017). Robust controller placement and assignment in software-defined cellular networks. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)* (pp. 1–9).
3. Akyildiz, I. F., Lin, S. C., & Wang, P. (2015). Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation. *Computer Networks*, *93*, 66–79.
4. Ashraf, U. (2018). Placing controllers in software-defined wireless mesh networks. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1–4). https://doi.org/10.1109/ICOMET.2018.8346386
5. Černỳ, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, *45*(1), 41–51.
6. Chaudet, C., & Haddad, Y. (2013). Wireless software defined networks: Challenges and opportunities. In *2013 IEEE International Conference on Microwaves, Communications, Antennas and Electronics Systems (COMCAS)* (pp. 1–5).
7. Dams, J., Hoefer, M., & Kesselheim, T. (2015). Scheduling in wireless networks with Rayleigh-fading interference. *IEEE Transactions on Mobile Computing*, *14*(7), 1503–1514.
8. Dvir, A., Haddad, Y., & Zilberman, A. (2018). Wireless controller placement problem. In *Consumer Communications and Networking Conference (CCNC), 2018 15th IEEE Annual* (pp. 1–4).
9. Faragardi, H. R., Vahabi, M., Fotouhi, H., Nolte, T., & Fahringer, T. (2018). An efficient placement of sinks and SDN controller nodes for optimizing the design cost of industrial IoT systems. Software: Practice and Experience.
10. Foundation, O. N. Interoperability event technical issues report. Report June 2013. https://3vf60mmveq1g8vzn48q2o71a-wpengine.netdna-ssl.com/wp-content/uploads/2014/10/openflow-spec-v1.2.pdf
11. Gopal, R., & Ravishankar, C. (2014). Software defined satellite networks. In *32nd AIAA International Communications Satellite Systems Conference*.
12. Heller, B., Sherwood, R., & McKeown, N. (2012). The controller placement problem. In *Proceedings of the first Workshop on Hot Topics in Software Defined Networks* (pp. 7–12). ACM
13. Hu, T., Guo, Z., Yi, P., Baker, T., & Lan, J. (2018). Multi-controller based software-defined networking: A survey. *IEEE Access*, *6*, 15980–15996.
14. Hu, Y., Luo, T., Beaulieu, N. C., & Deng, C. (2017). The energy-aware controller placement problem in software defined networks. *IEEE Communications Letters*, *21*(4), 741–744. https://doi.org/10.1109/LCOMM.2016.2645558.
15. Johnston, M., & Modiano, E. (2017). Controller placement in wireless networks with delayed CSI. *IEEE/ACM Transactions on Networking*, *25*(3), 1775–1788. https://doi.org/10.1109/TNET.2017.2651808.
16. Karakus, M., & Durresi, A. (2017). A survey: Control plane scalability issues and approaches in software-defined networking (SDN). *Computer Networks*, *112*, 279–293.
17. Killi, B. P. R., & Rao, S. V. (2017). Capacitated next controller placement in software defined networks. *IEEE Transactions on Network and Service Management*, *14*(3), 514–527. https://doi.org/10.1109/TNSM.2017.2720699.
18. Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., et al. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.
19. Lange, S., Gebert, S., Zinner, T., Tran-Gia, P., Hock, D., Jarschel, M., et al. (2015). Heuristic approaches to the controller placement problem in large scale SDN networks. *IEEE Transactions on Network and Service Management*, *12*(1), 4–17.
20. Liu, J., Shi, Y., Zhao, L., Cao, Y., Sun, W., & Kato, N. (2018). Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network. *IEEE Journal on Selected Areas in Communications.*, *36*, 221–232.
21. Liyanage, K. S. K., Ma, M., & Chong, P. H. J. (2018). Controller placement optimization in hierarchical distributed software defined vehicular networks. *Computer Networks*, *135*, 226–239.
22. Megiddo, N., & Supowit, K. J. (1984). On the complexity of some common geometric location problems. *SIAM Journal on Computing*, *13*(1), 182–196.
23. Oni, P. B., & Blostein, S. D. (2017). Decentralized AP selection in large-scale wireless LANs considering multi-AP interference. In *2017 International Conference on Computing, Networking and Communications (ICNC)* (pp. 13–18).
24. Patzold, M. (2001). *Mobile fading channels: Modelling, analysis and simulation*. New York: Wiley.
25. Qin, Q., Poularakis, K., G. I., & Tassiulas, L. (2018). *SDN controller placement at the edge: Optimizing delay and overheads*. IEEE Infocom.
26. Saadon, G., Haddad, Y., & Simoni, N. (2019). A survey of application orchestration and OSS in next-generation network management. *Computer Standards & Interfaces*, *62*, 17–31. https://doi.org/10.1016/j.csi.2018.07.003.
27. Sallahi, A., & St-Hilaire, M. (2015). Optimal model for the controller placement problem in software defined networks. *IEEE Communications Letters*, *19*(1), 30–33. https://doi.org/10.1109/LCOMM.2014.2371014.
28. Sallahi, A., & St-Hilaire, M. (2016). Expansion model for the controller placement problem in software defined networks. *IEEE Communications Letters*, *19*(1), 30–33.
29. Singh, A. K., & Srivastava, S. (2018). A survey and classification of controller placement problem in SDN. *International Journal of Network Management*, *28*(3), e2018.
30. Tanha, M., Sajjadi, D., Ruby, R., & Pan, J. (2018). Capacity-aware and delay-guaranteed resilient controller placement for software-defined WANs. *IEEE Transactions on Network and Service Management.*, *15*, 991–1005.
31. Vizarreta, P., Machuca, C. M., & Kellerer, W. (2016). Controller placement strategies for a resilient SDN control plane. In *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)* (pp. 253–259). https://doi.org/10.1109/RNDM.2016.7608295
32. Wu, S., Chen, X., Yang, L., Fan, C., & Zhao, Y. (2018). Dynamic and static controller placement in software-defined satellite networking. *Acta Astronautica*. https://doi.org/10.1016/j.actaastro.2018.07.017.
33. Xie, J., Guo, D., Hu, Z., Qu, T., & Lv, P. (2015). Control plane of software defined networks: A survey. *Computer Communications*, *67*, 1–10.
34. Yao, G., Bi, J., Li, Y., & Guo, L. (2014). On the capacitated controller placement problem in software defined networks. *IEEE Communications Letters*, *18*(8), 1339–1342.

Amit Dvir received the B.Sc., M.Sc., and Ph.D. degrees from Ben-Gurion University, Beer Sheva, Israel, all in communication systems engineering. He is currently a Faculty Member with the Computer Science Department and the Cyber Center, Ariel University, Israel. From 2011 to 2012, he was a Post-Doctoral Fellow with the Laboratory of Cryptography and System Security, Budapest, Hungary. His research interests include enrichment data from encrypted traffic.

Yoram Haddad received his B.Sc., Engineer diploma and M.Sc. (Radiocommunications) from SUPELEC (leading engineering school in Paris, France) in 2004 and 2005, and his Ph.D.; in computer science and networks from Telecom ParisTech in 2010 all awarded with honors. He was a Kreitman Post-Doctoral Fellow at Ben-Gurion University, Israel between in 2011–2012. In 2019 he received his Habilitations (HDR) from Sorbonne University. Since 2011 he is a faculty member at the Jerusalem College of Technology (JCT)—Lev Academic Center in Jerusalem, Israel and Head of FTNet Lab. Yoram's published about 50 papers in international conferences (e.g. SODA,…), book chapters and journals. He served on the Technical Program Committee of major IEEE conferences and was the chair of a working group at European COST action on Autonomous Control for a Reliable Internet of Services (ACROSS). He was the recipient of the Henry and Betty Rosenfelder outstanding researcher award for year 2013. Yoram's main research interests are in the area of Wireless Networks and Algorithms for networks, Wireless Software Defined Networks (SDN) and Machine Learning for 5G cellular networks and beyond.

Aviram Zilberman received the bachelor's degree in software engineering from the Technion—Israel Institute of Technology, Haifa, Israel, in 1999 and the master's degree in computer science and mathematics from Ariel University, Ariel, Israel, in 2018. He is currently working towards the Ph.D. degree in the area of security in mobile ad-hoc networks in the department of computer science at the Ariel University, Israel.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.