Cost-Effective Controller Placement Problem for Software Defined Multihop Wireless Networks

Afsane Zahmatkesh(✉), Thomas Kunz, and Chung-Horng Lung

Department of Systems and Computer Engineering, Carleton University,
Ottawa, ON, Canada
`afsane.zahmatkesh@carleton.ca`, {`tkunz,chlung`}`@sce.carleton.ca`

Abstract. In an SDN architecture, solving the controller placement problem (CPP) in a multi-controller environment plays an important role on network performance in terms of delay, reliability, control overhead, etc. In this architecture, control overhead, referred to as the network cost in this paper, consists of controller-device communications to discover the network topology, and exchange configurations and set up flow tables as well as inter-controller communications, if needed, to synchronize different network views and achieve the global view of the network. In software defined multihop wireless networking (SDMWN), because of the capacity limitation and the effect of interference on wireless links, and an in-band architecture in some types of networks to exchange both data and control traffic, it is important to solve the CPP while minimizing control overhead to reduce energy consumption, have lower packet losses and improve reliability. In this paper, the objective is to solve the CPP in SDMWN while minimizing the number of required control packets to be exchanged in the control plane. The novelty of our work is that we consider the characteristics of SDMWN and the capacity of wireless links to solve the CPP and select routes among network devices and controllers in the network. Our results demonstrate the impact of different factors such as the number of controllers, the capacity of wireless links and the arrival rate of new flows in devices on control overhead in SDMWN.

Keywords: Multihop wireless networks (MWNs) · Software defined multihop wireless networking (SDMWN) · Controller placement problem (CPP) · Control overhead

1 Introduction

Multihop wireless networks (MWNs) are self-configuring and self-organizing, in which network devices are able to communicate wirelessly with each other using one or more intermediate devices without the help of any infrastructure such as a base station or an access point. In addition to the common challenges of

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2021
Published by Springer Nature Switzerland AG 2021. All Rights Reserved
L. Foschini and M. El Kamili (Eds.): ADHOCNETS 2020, LNICST 345, pp. 130–146, 2021.
https://doi.org/10.1007/978-3-030-67369-7_10

all wireless networks such as unreliable and shared wireless medium, interference, etc., MWNs face additional challenges, including distributed management, device mobility, energy consumption, quality of service. In MWNs, since network management is distributed among network devices, it is challenging to optimize routing decisions and resource consumption globally, and to adjust to dynamic topology changes efficiently [1–3].

Software defined networking (SDN) [4] is a solution proposed to overcome some existing challenges in configuration and management of the traditional networks by decoupling the control plane and the data plane of the network. In SDN, the removed control plane from network devices is logically centralized in a controller that is responsible for managing the entire network. Although the SDN concept was first proposed for wired networks, applying SDN to MWNs can also be beneficial to overcome the MWN-specific challenges mentioned earlier. In the traditionally distributed management of MWNs, network devices are responsible for establishing the connectivity with each other using local, partial views of the network. While in software defined multihop wireless networking (SDMWN), the controller is able to optimize resource allocation globally and adjust to dynamic topology changes faster.

In SDMWN, a distributed control plane is used as an architecture in which the network is divided into different domains and multiple controllers are responsible for managing the domains separately. A distributed control plane is expected to address the challenges of a centralized control plane, i.e., only one controller manages the entire network, in terms of reliability, scalability, energy depletion issues, etc. However, it raises some new challenges including determining the number of controllers, assigning and placing controllers, referred to as the controller placement problem (CPP) [5], and achieving the global view of the network [6]. We describe these existing challenges using a distributed control plane in more detail in Sect. 2. Addressing these challenges, especially in SDMWN with shared and unreliable communications, has a significant impact on network performance metrics such as latency, cost, reliability and control overhead. To the best of our knowledge, in the related work in solving the CPP, all communications use the shortest paths and the authors have not considered the capacity of wireless links and the effect of interference on the capacity of links, which is crucial in SDMWN.

The objective of this paper is to find the placements of a given number of controllers and their assignments to the network devices in SDMWN with an in-band architecture to minimize the number of required control packets to be exchanged in the network. The control packets are exchanged between network devices and their assigned controllers, and among controllers to discover the network states and topology, exchange configurations and update flow rules in SDN forwarding devices. In an in-band architecture, the control and data traffic share the same band or channel. In SDMWN, minimizing the control packets flowing over links improves scalability and reduces energy consumption and packet losses, which also has an impact on the reliability of the control plane. In addition, to solve the problem, we consider the capacity of links and the effect of interference on controller placements, controller assignment and route selection among network devices and controllers as well as route selection among controllers. We formulate the CPP, which is an NP-hard problem [5], as a nonlinear problem. The formulated CPP is a nonlinear problem because we consider the capacity of links and route selection in the control plane. Therefore, we investigate the CPP with a small network to show the impact of the number of controllers, the arrival rate of new traffic flows, and the capacity of links on the control overhead in the network. In this paper, the terms control overhead and the network cost are used interchangeably.

Similar to the gateway selection in mobile ad hoc network [7], i.e., some devices are selected as gateways in the network, the objective of the CPP in SDMWN is to select network devices in the network to host an SDN controller. The main concern in a gateway selection is to exchange data traffic and provide access to the internet with the aim of different metrics such as load-balancing, throughput and quality of service. While in solving the CPP in SDMWN in this paper, the objective is to minimize control traffic exchanged in the network, considering different characteristics of SDMWN.

The organization of this paper is as follows. Section 2 discusses existing challenges in a distributed control plane architecture and presents an overview of the related work in SDMWN. Section 3 introduces an optimization model to minimize the generated control overhead in the network while considering the characteristics of SDMWN and related constraints. Section 4 shows the evaluation of the proposed model and Sect. 5 is the conclusion.

2 Related Work

As mentioned earlier, SDN has the potential to address the MWN-specific challenges in terms of network management, device mobility, energy consumption, quality of service, etc. Several studies have demonstrated that applying SDN to MWNs can be beneficial [1–3,8–10]. In [1], various studies are reviewed that show the benefits of applying SDN to MWNs, in which the control logic of wireless devices is logically centralized in an SDN controller that programs the whole network.

In most studies on SDMWN, a physically centralized controller manages the network to satisfy the requirements of different types of applications such as routing, scheduling, task allocation, load-balancing, congestion control, etc. In the context of SDMWN, the results show that these approaches, compared to the traditional distributed management, can be beneficial in terms of network performance metrics such as packet delivery ratio, delay, etc. [1].

However, in addition to the existing drawbacks of a centralized control plane such as a single point of failure, scaling limitations [11], applying a physically centralized control plane to SDMWN has some disadvantages. In this case, due to the mobility of network devices and scalability issue, direct connections among network devices and a centralized controller are not reasonable or practical for some scenarios, e.g., constrained resources for network devices. Therefore, network devices need to communicate with a controller in a multihop manner using possibly an unreliable and shared wireless medium to update the general view of the network and receive flow rules. Consequently, network devices may face higher latency, especially devices farther away from the controller, which leads to higher flow setup time [12].

In a distributed control plane environment, controllers are able to control devices that are closer geographically which helps to reduce latency and reacts to topology changes faster. Moreover, in case of a controller failure, other controllers can manage the network. However, using a distributed control plane in an SDN architecture raises several new challenges. Among them are how to determine the number of controllers and their locations, and how to assign controllers to network devices. Heller et al. in [5] refer to those challenges in a distributed control plane as the controller placement problem (CPP) that is a NP-hard problem. The main objective of the CPP is to find the number of controllers and their placements and to assign controllers to network devices while considering different metrics and objectives [13]. In addition, it is challenging to integrate different local views and to achieve the global view of the network to provide inter-domain communications among devices. Most studies in SDMWN with a distributed control plane do not provide any details about inter-controller communications using the wireless medium to achieve the global view of the network in SDMWN [14–17].

Addressing the existing challenges in a distributed control plane has an impact on the generated control overhead in the network [18]. In SDN, in addition to the data packets exchanged among network devices, control packets need to be exchanged periodically among a controller and network devices (controller-device) to discover the network topology, exchange configurations and set up flow tables. Moreover, a number of control packets need to be exchanged periodically among controllers (inter-controller) to synchronize and integrate different network views and obtain a global view of the network to provide inter-domain communications. Some SDMWNs, such as WSNs, because of their characteristics, only have a single interface to forward both control and data messages. Therefore, in such networks, the capacity of links is used to exchange both data and control traffic. Moreover, because of the wireless nature of links, interference can influence the available capacity of links [19]. Therefore, it is important to consider the capacity of links as a constraint in solving the CPP in SDMWN to exchange control traffic.

Although various studies have been reported to solve the CPP in wired networks, only a few studies consider the impact of solving the CPP in wireless networks on network performance. This problem is introduced as the wireless CPP in [20], in which communication links among controllers and network devices are wireless. In this case, in addition to the metrics addressed in the wired CPP, the characteristics of unreliable and shared wireless medium should be considered in solving the wireless CPP, which plays a critical role on connectivity among controllers and network devices. Moreover, most studies do not provide any details about inter-controller communications using the wireless mediums to achieve the global view of the network in SDMWN.

The objective of [20] is to minimize the number of controllers and the total delay, and find optimal controller placements and assignments in a wireless network. The authors formulate the problem as a chance-constrained stochastic program (CCSP) with consideration of wireless communications, in which the total delay consists of the network access delay for the devices, transmission delay, propagation delay and the queuing delay at the controller. Results demonstrate that the proposed model is able to reduce the number of selected controllers and delay in the network. Dvir et al. [21] propose a multi-objective optimization problem to solve the wireless CPP, in which inter-controller communications and communications among controllers and access points are wireless. The authors formulate the problem to minimize propagation delay and link failure probability while considering throughput and a new metric called transparency as constraints. Transparency is defined as the latency in the data plane, which is caused by interference added by the proposed control plane. Then, the authors introduce two heuristic algorithms to solve the problem that are able to find the number of controllers to approximately minimize the objective function and satisfy the constraints. Moreover, the results demonstrate that with increasing the number of devices in the network, link failure probability and delay obtained from both algorithms increase.

The objective of [22] is to solve the wireless CPP in a VANET with a two-layer hierarchical control plane while minimizing the number of controllers and delay in the network, which consists of transmission delay, queuing delay, contention delay, processing and propagation delay. Results show that, compared to a random placement, using the proposed approach, the network experiences lower latency. Moreover, the proposed approach improves network performance in terms of delay and packet delivery ratio. Qin et al. [18], propose an optimization model to find controller placements in wireless edge networks while achieving a tradeoff between minimizing delay among devices and their assigned controllers, and minimizing control overhead (Mbps) in the network. Control overhead in [18] consists of communication among devices and their assigned controllers to set up flow rules, and inter-domain communication to discover the network topology along the shortest paths. To solve for large-scale networks, the authors propose a randomized greedy algorithm to find controller placements. Results show that the proposed algorithm is able to find near-optimal solutions and improve the network performance in terms of minimizing delay and control overhead. However, in [18], the authors do not consider the controller-device communication control overhead to discover the network topology and the characteristics of wireless medium to solve the problem.

The objective of most studies in solving the CPP in both wired and wireless networks is to minimize propagation delay among controllers and devices that is proportional to the distance among them [13]. Minimizing the distance among controllers and devices as well as among controllers has a direct effect on the reliability of the control plane, especially in wireless networks with shared and unreliable communications.

To the best of our knowledge, in the related work of solving the CPP in wireless networks, to minimize propagation delay, the capacity of wireless links and the impact of interference on the capacity of links are not considered when determining the number of controllers and controller placements. Moreover, in the related work, the authors assume that all communications use the shortest paths and they do not consider the capacity of links and the impact of limiting the capacity of links to exchange the control overhead on solving the CPP. Consequently, route selection to provide controller-device and inter-controller communications based on the capacity of links is still an open research area in solving the CPP. Therefore, in this paper, our objective is to solve the CPP to minimize the cost of the network while considering the characteristics of SDMWN and the capacity of wireless links.

3 System Model

We model an SDMWN as a directed graph $G = (V, E)$, where V represents the set of wireless network devices and E is the set of links between each pair of devices such that link (u, v) and link (v, u) are the members of E if and only if device u is within the transmission range of device v (R_T). In this paper, we use the protocol model formulated in [19] to find a set of links in the interference range of link (u, v).

Since in MWNs network devices are responsible for organizing the network, we consider all wireless network devices as SDN forwarding devices, communicating in a multihop manner, controlled by an SDN controller. Moreover, we assume that all network devices are stationary and candidate locations to place a controller. In addition, we have an in-band architecture, i.e., network devices use a single interface to forward data traffic and exchange control traffic with a controller placed on a device and control traffic among controllers.

3.1 Model Outputs

The notations used in the proposed model are listed in Table 1. The outputs of the proposed model are listed as follows while considering the capacity of links as a constraint to place controllers in the network and assign them to network devices.

- Optimal placements of N controllers in an SDMWN
- Controller assignments to network devices
- Route selections among controllers and network devices as well as among controllers
- Optimal cost of placing N controllers in the network

3.2 Objective Function

We formulate the problem as a nonlinear programming (NLP) problem and (1) shows the objective function that aims to minimize the total cost of the network.

$$Min(Cost_{TD} + \sum_{k=1}^{|V|} \sum_{i=1, i \neq k}^{|V|} (R_{Flow\_Rq}\ x_{k,i} [\sum_{\forall (u,v) \in E} f_{u,v}^{k,i} + \sum_{\forall (u,v) \in E} f_{u,v}^{i,k}])) \tag{1}$$

subject to: (3), (4), (5), (6), (7), (9)

Table 1. Notations used in this paper

Notation	Definition
y_k	Output (binary decision variable): The value equals one if and only if there is a controller placed on device k
$x_{k,i}$	Output (binary decision variable): The value equals one if and only if device i is assigned to a controller placed on device k
$f_{u,v}^{a,b}$	Output (binary decision variable): The value equals one if and only if link (u, v) is used to provide communication between device a and device b
$Cost_{TD}$	The total cost of topology discovery (*control packets/second*) running by all controllers on their own assigned network devices and among all controllers in the network calculated using (2)
R_{TD}	The rate of running topology discovery by each controller (*1/second*)
R_{Flow_Rq}	The arrival rate of new flows in each device that triggers a flow request message toward the assigned controller (*1/second*)
N	A given number of controllers
$neighbor[i]$	A set of neighbors of network device i in the network
$neighbor_{i,j}$	The j^{th} neighbor of network device i
$C_{u,v}$	The capacity of link (u, v) to exchange control Packets (*control packets/second*)
$C'_{u,v}$	The required bandwidth of link (u, v) to exchange control Packets (*control packets/second*)
$L_{u,v}^{Int}$	A set of links in the interference range of link (u, v)

The first part of (1) ($Cost_{TD}$) is the total cost of topology discovery in the network calculated using (2). The second part of (1) calculates the cost of controller-device communications using the best routes among controllers and their assigned network devices to exchange configurations and set up flow rules (*control packets/second*). In the case here, we will count control packet transmissions over each hop along their routes.

a) Cost of Topology Discovery: We assume that each device communicates only with its own assigned controller and each controller discovers a partial view of the network including its own assigned devices. Therefore, controllers communicate together to obtain the global view of the network.

$$\begin{aligned} Cost_{TD} = R_{TD}(\sum_{k=1}^{|V|} \sum_{i=1, i \neq k}^{|V|} [\sum_{\forall (u,v) \in E} f_{u,v}^{k,i} + (\sum_{m=1}^{|V|} \sum_{j=1, j \neq k,m}^{|neighbor[i]|} \\ \sum_{\forall (u,v) \in E} f_{u,v}^{neighbor_{i,j},m} \; x_{m,neighbor_{i,j}} \;) \;] \; x_{k,i}) \\ + R_{TD} \; [\sum_{k=1}^{|V|} \sum_{p=1, p \neq k}^{|V|} (y_k \; y_p \sum_{\forall (u,v) \in E} f_{u,v}^{k,p}) \;] \end{aligned} \tag{2}$$

The first and the second lines of $Cost_{TD}$ shown in (2) calculate the total number of control packets exchanged among controllers and devices per second to discover the network topology. The third line of (2) calculates the number of control packets exchanged among controllers per second to obtain the global view of the network.

To discover the network topology, each controller constructs and sends probes periodically to its own assigned devices. We calculate the total number of control packets generated by the controllers per second to their own assigned devices and count the number of hops ($\sum_{\forall (u,v) \in E} f_{u,v}^{k,i}$) in the best route while satisfying the constraints in 3.3.

When the assigned devices receive the probes from their controllers, they flood the probes. This cost is inevitable for any controller placement, resulting in a packet being transmitted over a wireless link to discover that link. Therefore, we do not need to model this cost explicitly in solving the CPP. After flooding the probes, their neighbors send the received probes to their own assigned controllers (the second line of (2)). Using these packets, each controller is able to find the links among its own devices and to devices outside of its partial view.

In addition, controllers need to communicate together in an interval of time to achieve the global view of the network while considering the number of hops between each pair of controllers ($\sum_{\forall (u,v) \in E} f_{u,v}^{k,p}$)). With controllers managing only relatively small parts of the network, we assume that the information to be exchanged fits into a single control packet. We consider the same interval of time for both running topology discovery by each controller and inter-controller communications.

b) Cost of Exchanging Configurations and Setting Up Flow Tables: The second part of (1) calculates the total number of control packets exchanged among controllers and their own assigned network devices to exchange configurations and set up flow tables, counting the number of hops in both directions among controllers and devices, and the arrival rate of flow requests in the controllers.

3.3 Constraints

The objective function presented in (1) is subject to the following constraints. The constraint defined in (3) avoids assigning a device to a controller that is not placed in the network.

$$x_{k,i} \leq y_k,\ \forall i, k \in V \tag{3}$$

The constraint defined in (4) ensures that each device is assigned to exactly one controller.

$$\sum_{k=1}^{|V|} x_{k,i} = 1,\ \forall i \in V \tag{4}$$

Equation (5) ensures that there is a given number of controllers in the network.

$$\sum_{k=1}^{|V|} y_k = N \tag{5}$$

Equation (6) defines the control flow conservation constraint, in which the constraint ensures that the total control flow entering each device in the network equals to the total control flow leaving the device except for the source and destination of a flow.

$$\sum_{(u,v)\in E} f^{a,b}_{(u,v)} - \sum_{(v,w)\in E} f^{a,b}_{(v,w)} = \begin{cases} 1, & \text{if } v = b \\ \text{-1}, & \text{if } v = a \\ 0, & \text{otherwise} \end{cases},\ \forall a, b \in V \tag{6}$$

In this model, we assume that, only a fraction of the link capacity is assigned to exchange control packets. The constraint defined in (7) ensures that each link can handle the total number of control packets flowing over the link per second. In this model, the total number of control packets flowing over link (u, v) in the network consists of the total number of control packets exchanged per second to discover the network topology, and exchange configurations and set up flow tables $(C'_{u,v})$ as well as the total number of control packets flowing over links in the interference range of link (u, v) per second $(C'_{u',v'})$.

$$C'_{u,v} + \sum_{\forall (u',v') \in L^{Int}_{u,v}} C'_{u',v'} \leq C_{(u,v)},\ \forall (u, v) \in E \tag{7}$$

Equation (8) calculates the total number of control packets flowing over a link per second to discover the network topology (the first, the second and the third lines) and to provide controller-device communications to exchange configurations and set up flow tables (the fourth line).

$$\begin{aligned} C'_{u,v} = R_{TD}[\sum_{k=1}^{|V|} \sum_{i=1,i\neq k}^{|V|} (f_{u,v}^{k,i} \\ + \quad \sum_{m=1}^{|V|} \sum_{j=1,j\neq k}^{|neighbor[i]|} f_{u,v}^{neighbor_{i,j},m} x_{m,neighbor_{i,j}}) x_{k,i}] \\ + \quad R_{TD}[\sum_{k=1}^{|V|} \sum_{p=1,p\neq k}^{|V|} (y_k \; y_p \; f_{u,v}^{k,p} \;)] + \\ \sum_{k=1}^{|V|} \sum_{i=1,i\neq k}^{|V|} (R_{Flow\_Rq} \; x_{k,i} \; [f_{u,v}^{k,i} + f_{u,v}^{i,k}]), \\ \forall (u,v) \in E \end{aligned} \tag{8}$$

Integrality constraints are presented in (9).

$$x_{k,i}, y_k, f_{u,v}^{k,i} \in \{0,1\}, \;\forall i,k \in V, \;\forall (u,v) \in E \tag{9}$$

4 Model Results and Analysis

We use AMPL (a mathematical programming language) [23] to implement our proposed optimization model running on an Intel Core i7 CPU (3.20 GHz) and 16.0 GB RAM. Moreover, we use the Baron solver 19.7.13 [24] which aims to find the optimal solutions globally for nonlinear optimization problems. We use NEOS server [25–27] to run the proposed model in AMPL running on an Intel Xeon E5- 2698 @ 2.3 GHz, 192 GB RAM and 300G SAS drives setup in RAID5.

Because of the high computational complexity of the proposed model, we consider a small network with 6 wireless network devices as shown in Fig. 1 to illustrate the performance of the proposed optimization model. The links in the interference range of each link in the topology shown in Fig. 1 are not displayed due to the space limit. Here is only an example of the links in the interference range of link *(1,2)*: $L_{1,2}^{Int} = \{$(1,3), (2,4), (3,4), (4,2), (4,5), (5,3), (6,4), (2,1), (3,1), (3,5), (4,3), (4,6), (5,4)$\}$. Our objective is to place N controllers in this network while minimizing the cost of the control plane defined in (1) and satisfying the defined constraints in 3.3. In this evaluation, we assume that each device receives a new flow every 2 s ($R_{Flow\_Rq} = 0.5$ (*1/second*)) and each controller runs topology discovery every 5 s ($R_{TD} = 0.2$ (*1/second*)) which is adopted from OpenDaylight [28], the most popular open source SDN Controller.

4.1 The Impact of the Number of Controllers on the Cost of the Network

To find the optimal cost of the network when placing different number of controllers, we run our proposed optimization model for different values of N from

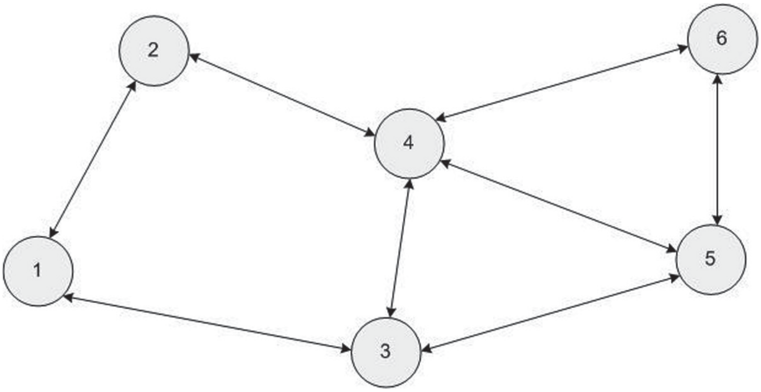


Fig. 1. An SDMWN with 6 wireless network devices

1 (placing a controller in one of the devices) to 6 (placing a controller on each device). As we mentioned earlier, the total cost of the network consists of the total cost of topology discovery and the total cost of controller-device communications to exchange configurations and set up flow tables in devices. Figure 2 demonstrates the optimal total cost of the network when placing different number of controllers in the network. As shown in this figure, when we solve the optimization problem to place three controllers in the network ($N = 3$), the minimum cost of the network is achieved. In this case, the optimal controller placements are devices 3, 4 and 5. Moreover, as demonstrated in Fig. 2, when we solve the problem to place one or six controllers, the network experiences the highest cost of the control plane defined in (1). Therefore, finding the right number of controllers in a network, depending on the network topology, has a direct effect on the cost of the network.

Figure 3 shows the total cost of topology discovery, which consists of the total cost of controller-device communications to discover the network state and topology, and the total cost of inter-controller communications to integrate views from each domain and obtain the global view of the network. As demonstrated in this figure, as the number of controllers increases in the network, the cost of controller-device communications decreases, while the network faces higher cost of inter-controller communications to integrate different network views. Although placing three controllers in the network results in the minimum total cost of the network as shown in Fig. 2, the optimal cost of topology discovery is obtained when we place two controllers in the network as demonstrated in Fig. 3. Therefore, a solution with the optimal cost of topology discovery does not necessarily result in the overall optimal cost of the network.

Depending on the network topology, placing different numbers of controllers may result in the same network cost. For example, in the topology as shown in Fig. 4a, in case of placing 2 or 3 controllers in the network ($N = 2$ or $N = 3$), the total network cost is 2.8 (*control packets/second*). Moreover, depending on

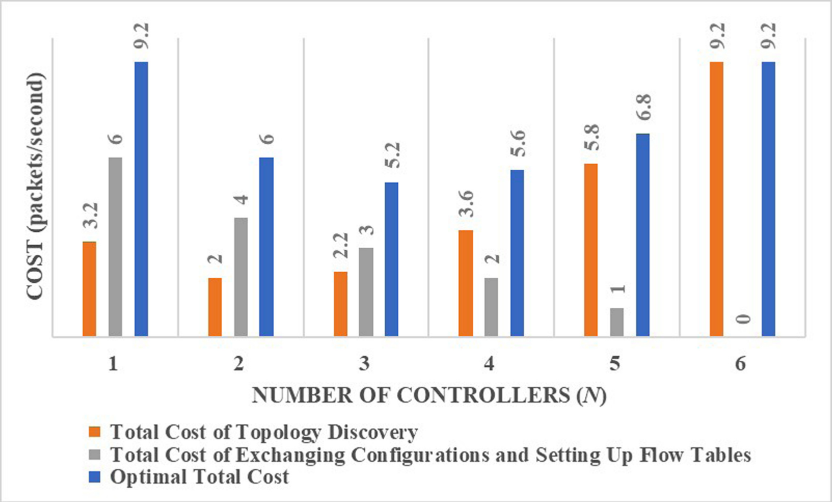


Fig. 2. The total cost of the network (*control packets/second*)

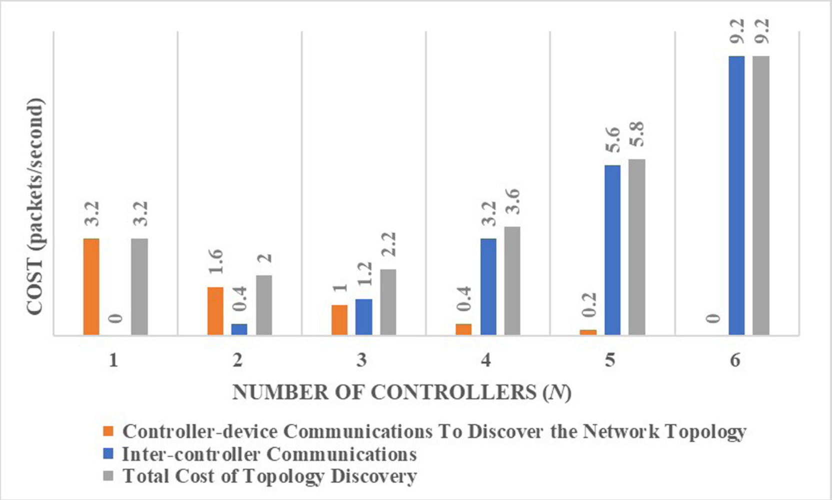
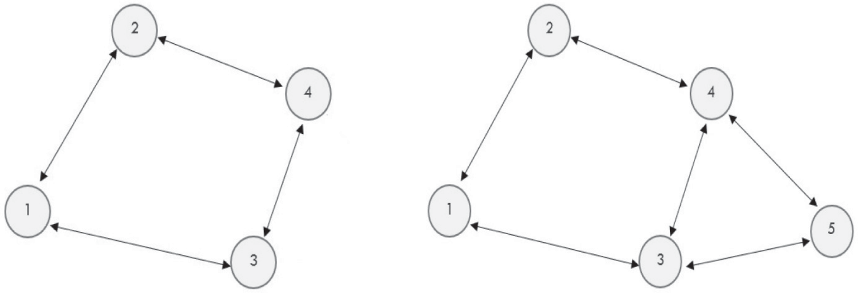


Fig. 3. The cost of topology discovery in the network (*control packets/second*)

the topology, it is possible to have more than one placement to be selected as the optimal placement. For example, in the topology as shown in Fig. 4b, when $N = 1$, both device 3 and device 4 can be selected as the optimal placement by the solver.



(a) network with 4 devices (b) network with 5 devices

Fig. 4. Impact of number of controllers and controller placements on the network cost

4.2 The Impact of R_{Flow_Rq} On the Network Cost

Figure 5 demonstrates that if we increase the arrival rate of new flows in each device (R_{Flow_Rq}) in the defined scenario, since the number of control packets exchanged per second to set up new flow rules increases, the optimal solution to minimize the total cost of network is to place one controller on each device ($N = 6$). Therefore, in case of increasing the value of R_{Flow_Rq}, placing more controllers in the network decreases the cost of communications among devices and controllers and the total cost of the network.

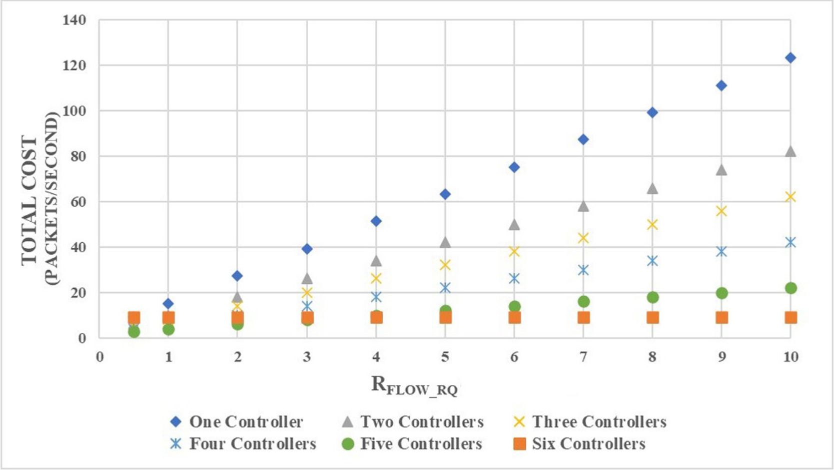


Fig. 5. The impact of increasing the value of R_{Flow_Rq} on the cost of the network

4.3 The Impact of Link Capacity on the Number of Controllers

To show the impact of the capacity of links assigned to exchange control packets on the number of controllers in the network while considering different values of arrival rate of flow requests, first, we limit the assigned capacity of all links to 6 (*control packets/second*). In this case, the solver is not able to solve the problem for $N = 1, 5$ and 6 while satisfying the capacity constraint defined in (7). In this scenario, depending on the network requirements, only two (total cost = 6 (*control packets/second*)), three (total cost = 5.2 (*control packets/second*)) or four controllers (total cost = 5.6 (*control packets/second*)) can be placed on the network while satisfying the capacity constraint.

On the other hand, as shown in Fig. 5, if we assume that $R_{Flow\_Rq} = 4$ (*1/second*), with an increase of the number of controllers in the network, the total cost of the network decreases such that placing six controllers has the minimum cost (10 (*control packets/second*)). In this case, e.g., if we limit the assigned capacity of all links to 10 (*control packets/second*) to exchange control packets, the solver is not able to solve the problem for $N < 6$ while satisfying the capacity constraint defined in (7). In other words, the problem can be solved only for placing six controllers in the network to satisfy the capacity constraint. Therefore, depending on the capacity of links and the value of R_{Flow_Rq}, the number of possible controllers to be placed in the network varies. In addition, in our proposed model, communications among controllers and devices as well as among controllers may not use the shortest paths to ensure the capacity limit of links. Therefore, a solution that would enforce or only consider communication along the shortest path would either violate the capacity constraint or not be a feasible solution.

Finally, results show that when $N = 1$, placing a controller in one of the possible controller placements in the topology shown in Fig. 1 instead of solving the optimization problem does not minimize the cost of the network. We compare the optimal cost of the network when placing one controller in the topology shown in Fig. 1 with the average cost of placing a controller in one of the six possible controller placements each time in the network. Each device k in Fig. 1 can be a candidate to be placed a controller ($y_k = 1$). The optimal cost of placing one controller in the network is 9.2 (*control packets/second*), while the average cost of placing a controller in one of the possible placements is 12.6 (*control packets/second*). The results demonstrate that the proposed model indeed finds a placement that minimizes the cost of the network while considering different factors including the capacity of links to solve the problem.

4.4 The Impact of Increasing the Number of Devices on the Execution Time of the Optimization Problem

The results demonstrate that when we increase the number of devices in the network, it takes a long time to solve the proposed optimization problem and find the minimum number of controllers that aims to minimize the network cost. For example, it takes 0.497515 (*seconds*) to solve the problem in a network with

3 devices, while the solver needs 153.67783 (*minutes*) to solve the problem in a network with 6 devices. Further, as shown in Sect. 4.1, the total cost for different solutions may be close and there may be more than one optimal solution for a specific network topology and a given set of constraint. Therefore, investigating the heuristic algorithms to the proposed optimization model helps to find the near-optimal solutions in large-scale networks.

5 Conclusion and Future Research

In this paper, we proposed an optimization model to find the placement of N controllers in SDMWN and their assignment to network devices while minimizing the total control packets required to be exchanged in the network to discover the network state and topology, and to exchange configurations and set up flow tables in network devices. To solve the CPP in SDMWN, we considered the capacity limits of wireless links and the impact of interference on the capacity of links to select routes among devices and controllers as well as among controllers. The results obtained from the optimization model demonstrate the impact of different factors such as the number of controllers, the capacity of wireless links and the arrival rate of new flows in devices on the generated control overhead in the network. Moreover, our proposed optimization problem is able to minimize the control overhead compared to a random placement.

In this paper, due to the high computational complexity of the optimization model, only a small network was demonstrated for evaluation. Investigating heuristic algorithms to find a near-optimal solution in a large-scale SDMWN in a reasonable time is being conducted. In addition, adjusting the number of controllers and controller placements based on topology changes to minimize the number of required control packets exchanged in the network per second should be addressed in the future.

References

1. Zahmatkesh, A., Kunz, T.: Software defined multihop wireless networks: promises and challenges. J. Commun. Netw. **19**(6), 546–554 (2017)
2. Chahal, M., Harit, S., Mishra, K.K., Sangaiah, A.K., Zheng, Z.: A survey on software-defined networking in vehicular ad hoc networks: challenges, applications and use cases. Sustain. Cities Soc. **35**, 830–840 (2017)
3. Kobo, H.I., Abu-Mahfouz, A.M., Hancke, G.P.: A survey on software-defined wireless sensor networks: challenges and design requirements. IEEE Access **5**, 1872–1899 (2017)
4. Kreutz, D., Ramos, F.M.V., Verıssimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. Proc. IEEE **103**(1), 14–76 (2015)
5. Heller, B., Sherwood, R., McKeown, N.: The controller placement problem. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, pp. 7–12 (2012). https://doi.org/10.1145/2342441.2342444

6. Zhang, Y., Cui, L., Wang, W., Zhang, Y.: A survey on software defined networking with multiple controllers. J. Netw. Comput. Appl. **103**, 101–118 (2018)
7. Mahiddin, N.A., Sarkar, N.I., Cusack, B.: An Internet access solution: MANET routing and a gateway selection approach for disaster scenarios. Rev. Socionetw. Strat. **11**, 47–64 (2017). https://doi.org/10.1007/s12626-017-0004-3
8. Haque, I.T., Abu-Ghazaleh, N.: Wireless software defined networking: a survey and taxonomy. IEEE Commun. Surv. Tutorials **18**, 2713–2737 (2016)
9. Rademacher, M., Jonas, K., Siebertz, F., Rzyska, A., Schlebusch, M., Kessel, M.: Software-defined wireless mesh networking: current status and challenges. Comput. J. **60**, 1520–1535 (2017)
10. Jaballah, W. B., Conti, M., Lal, C.: A survey on software-defined VANETs: benefits, challenges, and future directions. CoRR, vol. abs/1904.04577 (2019)
11. Oktian, Y.E., Lee, S., Lee, H., Lam, J.: Distributed SDN controller system: a survey on design choice. Comput. Netw. **121**, 100–111 (2017)
12. Ur Rahman, S., Kim, G., Cho, Y., Khan, A.: Deployment of an SDN-based UAV network: controller placement and tradeoff between control overhead and delay. In: International Conference on Information and Communication Technology Convergence, pp. 1290–1292 (2017). https://doi.org/10.1109/ICTC.2017.8190924
13. Das, T., Sridharan, V., Gurusamy, M.: A survey on controller placement in SDN. IEEE Commun. Surv. Tutor. **22**(1), 472–503 (2020)
14. Alioua, A., Senouci, S.-M., Moussaoui, S.: dSDiVN: a distributed software-defined networking architecture for infrastructure-less vehicular networks. In: International Conference on Innovations for Community Services (2017). https://doi.org/10.1007/978-3-319-60447-3_5
15. Mora, S., Vera, J.: RDSNET: a proposal for control architecture for software defined MANETs. Int. J. Eng. Technol. **10**(3), 816–827 (2018)
16. Kazmi, A., Khan, M. A., Akram, M. U.: DeVANET: decentralized software-defined VANET architecture. In: IEEE International Conference on Cloud Engineering Workshop, pp. 42–47 (2016). https://doi.org/10.1109/IC2EW.2016.12
17. Elzain, H., Yang, W.: Decentralizing software-defined wireless mesh networking (D-SDWMN) control plane. In: Proceedings of the World Congress on Engineering (2018)
18. Qin, Q., Poularakis, K., Iosifidis, G., Kompella, S., Tassiulas, L.: SDN controller placement with delay-overhead balancing in wireless edge networks. IEEE Trans. Netw. Serv. Manage. **15**(4), 1446–1459 (2018)
19. Gupta, P., Kumar, P.R.: The capacity of wireless networks. IEEE Trans. Inf. Theor. **46**, 388–404 (2000)
20. Abdel-Rahman, M.J., et al.: On stochastic controller placement in software-defined wireless networks. In: IEEE Wireless Communications and Networking Conference, WCNC, pp. 1–6 (2017). https://doi.org/10.1109/WCNC.2017.7925942
21. Dvir, A., Haddad, Y., Zilberman, A.: The controller placement problem for wireless SDN. Wirel. Netw. **25**, 4963–4978 (2019)
22. Liyanage, K.S., Ma, M., Joo Chong, P.H.: Controller placement optimization in hierarchical distributed software defined vehicular networks. Comput. Netw. **135**, 226–239 (2018)
23. Fourer, R., Gay, D.M., Kernighan, B.W.: AMPL: A Modeling Language for Mathematical Programming, 2nd edn. Duxbury Press, Scituate (2002)
24. BARON Solver, Version 19.7.13. https://minlp.com/baron-downloads. Accessed Sep 2019

25. Gropp, W., More, J. J.: Optimization environments and the NEOS server. In: Buhman, M.D., Iserles, A. (eds.)Approximation Theory and Optimization, pp. 167–182. Cambridge University Press (1997)
26. Dolan, E. D.: The NEOS server 4.0 administrative guide. Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory (2001)
27. The neos server for baron/ampl. https://neos-server.org/neos/solvers/go:BARON/AMPL.html. . Accessed Sep 2020
28. OpenDaylight (ODL). https://www.opendaylight.org