

RESEARCH

Open Access



Control plane delay minimization-based capacitated controller placement algorithm for SDN

Rong Chai* , Qiongfang Yuan, Lei Zhu and Qianbin Chen

Abstract

Software-defined networking (SDN) is proposed as a new network paradigm which decouples control plane from data plane and provides flexible network management. In this paper, we consider the capacitated controller placement problem in SDN, which jointly determines the number, location, and the capacity matching strategy of SDN controllers. To stress the importance of control plane delay which is composed of both the transmission and processing delay between controllers and switches, and the inter-controller delay; we formulate control plane delay minimization problem subject to controller-switch association constraints, controller capacity constraints, etc. Since the formulated optimization problem is a complicated nonlinear integer programming problem which cannot be solved easily, we transform it into controller-switch association subproblem and controller capacity matching subproblem. To solve the controller-switch association subproblem, we propose a heuristic algorithm based on Dijkstra algorithm and K-means algorithm. Given the controller-switch association strategy, we then apply the Kuhn-Munkres (K-M) algorithm to solve the controller capacity matching subproblem and obtain the capacitated controller placement strategy. Simulation results are shown to demonstrate the effectiveness of the proposed algorithm.

Keywords: Software defined networking, Control plane delay, Controller placement, Capacity matching

1 Introduction

The rapid development of broadband services and the quick growth of user data traffic pose challenges to traditional network architectures and management technologies. It is highly difficult for traditional networks to meet the increasing user service requirements due to inflexible network architectures, complicated network management mechanisms, poor scalability, and low utilization of network resources [1, 2]. To tackle this problem, researchers proposed software-defined networking (SDN) technology [3].

As a promising novel network paradigm, SDN is capable of separating data plane from control plane. The controllers in the control plane are responsible for centrally controlling and managing the forwarding devices in the data plane, e.g., routers or switches [4]. On the other hand, the data forwarding devices only need to conduct

the flow forwarding policies received from the controllers and no longer participate in network control [5]. Through applying logically centralized control and efficient management, SDN is expected to greatly simplify the forwarding function of the underlying devices and the control process of the entire network, and promote the rapid deployment of complex network functions by offering application programming interface (API) for network administrators [6].

For an SDN scenario consisting of multiple switches, a certain number of controllers should be deployed [7]. The controller placement problem should be studied which typically refers to how to place controllers in the network and how to allocate the associated switches to those controllers, i.e., determining the number of controllers, their optimal location, and the association relationship between controllers and switches [8]. As the controllers are required to conduct signaling interaction with their associated switches as well as with other controllers, the problem of controller placement is of particular importance as the location and capability of the controllers may

*Correspondence: chairong@cqupt.edu.cn
School of Communications and Information Engineering Chongqing
University of Posts and Telecommunications, Key Lab of Mobile
Communication Technology, Chongqing, China

highly affect the signaling transmission performance in control plane, i.e., the signaling interaction performance between switches and controllers, and that between controllers.

Some previous research works addressed the energy consumption or the transmission cost required for transmitting signaling messages between switches and controllers and proposed controller placement strategies which aim to achieve the lowest energy consumption or the minimum transmission cost [9–18]. However, as a major performance metric characterizing service transmission performance, the signaling transmission delay should be considered when designing optimal controller placement strategies. While the signaling propagation delay between controllers and switches was mainly considered in [19–24], the inter-controller signaling transmission, which is mandatorily required for achieving controller synchronization over network states, failed to be considered extensively.

A general assumption in previous research works was that the controllers to be deployed are of exactly the same characteristics which can be arbitrarily defined; however, in a practical deployment scenario, the characteristics of the controllers which can be deployed may be subject to certain constraints. For instance, it is highly possible that the controllers can only be selected from a given candidate controller set, and the parameters of the controllers are also pre-defined in a certain parameter set. In this case, the capacitated controller placement problem, i.e., determining the number, location, and capacity of controllers based on a candidate controller set, should be considered [25–31].

In this paper, we assume that the controllers can only be selected from a given candidate controller set and study the capacitated controller placement problem of SDN. In particular, it is assumed that the capacity of the candidate controllers which characterizes the processing capability of the controllers is chosen from a certain set containing various pre-determined capacity values. To achieve efficient placement of the controllers, we propose an SDN hierarchical management architecture. Then, stressing the importance of control plane delay which is composed of both the transmission and processing delay between controllers and switches, and the synchronization delay between controllers, we formulate control plane delay minimization problem subject to controller-switch association constraints and controller capacity constraints, etc. Since the formulated optimization problem is a complicated nonlinear integer programming problem which cannot be solved easily, we transform it into controller-switch association subproblem and controller capacity matching subproblem. To solve the controller-switch association subproblem, we propose a heuristic algorithm based on Dijkstra algorithm and K-means algorithm.

Given the controller-switch association strategy, we then apply the Kuhn-Munkres (K-M) algorithm to solve the controller capacity matching subproblem and obtain the corresponding capacitated controller placement strategy.

We summarize the major contributions of this work as follows:

- In this paper, we study the capacitated controller placement problem of SDN. To achieve the efficient placement of the controllers, we design an SDN hierarchical management architecture based on which the proposed capacitated controller deployment scheme can be conducted.
- While previous research work [19–24] aimed to achieve the minimum propagation delay between controllers and switches when designing controller placement strategy for SDN, they failed to examine extensively the inter-controller delay, the signaling transmission delay between controllers and switches, and the signaling processing delay at the controllers which may jointly affect the transmission performance of the user services severely. In this paper, we examine the control plane delay of SDN comprehensively; formulate it as a sum of the transmission delay and propagation delay between controllers and switches, the processing delay at the controllers, and the inter-controller delay; and design an effective capacitated controller placement strategy which minimizes control plane delay.
- While most of the previous research works made a general assumption that the controllers to be deployed are of exactly the same characteristics which can be arbitrarily chosen, in practical deployment scenarios, the characteristics of the controllers which can be deployed may be subject to certain constraints and the parameters of the controllers might be pre-defined. In this paper, we assume that the controllers can only be selected from a given parameter set; in particular, the processing capability of each candidate controller is chosen from a certain set containing various pre-determined capacity values, and design the capacitated controller placement strategy which jointly determines the number and location of the controllers and the matching strategy between the controllers and the capacity values.
- We formulate the capacitated controller placement problem as control plane delay minimization-based optimization problem. Since the formulated optimization problem is a complicated nonlinear integer programming problem which cannot be solved easily, we transform it into controller-switch association subproblem and controller capacity matching subproblem. To solve the controller-switch

association subproblem, we propose a heuristic algorithm based on Dijkstra algorithm and K-means algorithm. Given the controller-switch association strategy, we then apply the K-M algorithm to solve the controller capacity matching subproblem and obtain the capacitated controller placement strategy.

The remainder of this paper is structured as follows. Section 2 reviews some related works. Section 3 describes the system model and the proposed SDN hierarchical management architecture. In Section 4, the capacitated controller placement problem is formulated. The solution to the formulated optimization problem is described in Section 5. Simulation results are presented in Section 6. Finally, Section 7 concludes this paper.

2 Related work

In this section, we present an overview of the related work.

2.1 Energy consumption minimization-based controller placement algorithms

As an important metric, energy consumption is often stressed when designing the optimal controller placement strategy for SDN. In [9], the authors proposed to switch off as many control links as possible in order to reduce the energy consumption of the network while ensuring all the switches are capable of connecting to one controller under maximum delay constraint. Jointly considering the propagation delay between controllers and switches, the load status of the controllers, and link utilization, the authors further proposed a controller-switch association algorithm to reduce network energy consumption. In [10], the authors considered the controller placement problem of SDN from the perspective of energy consumption of the control link between controllers and switches. Considering the energy consumption of the network that serves for the control traffic, the propagation delay, and the load of controllers, the authors formulated the controller placement problem as an energy consumption minimization-based binary integer programming problem and obtained the optimal strategy by solving the optimization problem.

2.2 Cost effective controller placement algorithms

While energy consumption minimization was addressed when designing controller placement strategy in SDN [9, 10], network signaling cost may also have a significant impact on the traffic transmission performance through the network; hence, some research works considered the cost for deploying SDN controllers and proposed cost-effective controller placement strategies [11–18].

The authors in [11] considered the signaling cost of transmitting flow setup requests from switches to their associated controllers in SDN. By defining the signaling

cost as the product of the number of flow setup requests and the signaling transmission delay between controllers and switches, the controller placement problem was formulated to minimize the signaling cost, and the optimal placement strategy of the controller was obtained by solving the optimization problem. In [12], a cost function was defined as a weighted sum of the state collection cost of switches, inter-controller synchronization cost, and switch reassignment cost. Aiming to minimize the cost function, the authors proposed a heuristic algorithm to determine the optimal number and placement strategy of the controllers. The authors in [13] jointly considered the communication cost between controllers, and that between controllers and switches. In order to reduce the two types of communication costs, a Pareto optimal solution for controller placement was proposed.

Dynamic controller placement problem of SDN was addressed in [14–16]. Under the assumption that controllers were statically deployed in the network and switches were statically associated to the controllers, the authors in [14] studied dynamic controller association problem. By formulating the problem as an online optimization problem which minimizes the overall cost caused by transmitting flow setup requests and the maintenance cost of the controllers, a hierarchical two-phase algorithm was proposed to obtain the suboptimal controller placement strategy. In [15], the authors proposed a two-level hierarchy (master and slave) control framework and designed a slave controller placement strategy to optimize control cost and control plane delay. In [16], dynamic controller placement and flow management problems were jointly considered for SDN and a joint optimal strategy was proposed to minimize maintenance cost and maximize controller utilization as well.

The financial cost for deploying controllers in SDN was considered in [17, 18]. In [17], the authors jointly considered the financial cost for installing controllers and setting up association relationship between switches and controllers and formulated the controller placement problem as an overall financial cost minimization problem. The optimal placement strategy of the controllers was obtained by solving the optimization problem. The problem of re-organizing an existing SDN was addressed in [18]. The financial cost for re-organizing the network was defined as the total cost for deploying or removing controllers, and setting up or removing network links, and an optimal re-organizing strategy was proposed which minimizes the financial cost.

2.3 Delay minimization-based controller placement algorithms

The previous works in [9–18] focused on minimizing energy consumption or designing cost-effective controller

placement strategies for SDN; however, it can be observed that the transmission delay or propagation delay between switches and controllers may affect network performance significantly; thus, it is highly desired to design delay minimization-based controller placement strategy.

The control plane delay between switches and controllers was considered in designing the optimal SDN controller placement strategy [19–24]. The authors in [19] studied the problem of placing controllers in SDN with wireless southbound interface. To achieve low signaling transmission latency and link failure probability, a heuristic solution was proposed based on simulated annealing genetic algorithm. In [20], the propagation delay from switches to controllers was examined and the propagation delay bound was set as a criterion to obtain the location and the number of controllers being deployed in the SDN scenario. In [21], the authors considered an SDN scenario consisting of a certain number of subnetworks and characterized the controller placement problem as a network partition problem. By employing K-means algorithm-based scheme, the maximum propagation delay between controllers and switches in each subnetwork was minimized. In [22], a controller placement approach based on community detection was proposed. The authors regarded the SDN topology as a network composed of multiple communities and designed controller placement strategy for each community. To further constrain the number of nodes in each community and achieve the balance of the controllers in different communities, a scale constraint factor was introduced. In [23], the authors formulated the flow setup time of various switches and optimally designed the controller placement strategy so as to minimize the average flow setup time. In [24], the authors divided the entire network into several sub-networks and proposed a modified density peaks clustering (MDPC) algorithm to design the controller placement strategy, so as to minimize the average propagation latency between switches and controllers.

In the case that network failures may occur, for instance, controllers, switches, or the links between switches may become unavailable, robust controller placement strategies are highly desired. To enhance the resilience of the controllers, the authors in [25] proposed to deploy redundant controllers at various resilience levels. By minimizing the propagation delay between the redundant controllers and switches, the optimal resilient controller placement strategy can be obtained. The research work in [26] considered the trade-off among various performance requirements, such as load balancing, failure tolerance, and transmission latency, and in order to achieve the trade-off, an optimal controller placement framework based on the Pareto game model is proposed. The authors in [27] considered the delay between controller and the associated switches in link failure state and proposed

a particle swarm optimization-based heuristic controller placement method to guarantee the delay and reliability in control plane.

While the authors in [20–27] stressed control plane delay optimization, they mainly considered a relatively simple delay formulation and ignored some important delay components, such as the signaling transmission delay, signaling processing delay, and the inter-controller delay. Furthermore, most of the previous research works assumed that the controllers to be deployed are of exactly the same characteristics which can be arbitrarily set; however, in practical deployment scenarios, the characteristics of the controllers may be subject to certain constraints and the parameters of the controllers might be pre-defined in a certain parameter set, in which case, the problem of capacitated controller placement or joint controller placement and capacity matching should be studied.

In [28], the authors explored and investigated more possible delay contributors, including the end-to-end latency and the queuing latency of controllers. The concept of network partitioning was introduced to decrease end-to-end delay. To further reduce the queuing latency of controllers, it is proposed to place appropriate multiple controllers in the sub-networks. The authors in [29] jointly took into account the propagation delay from switches to controllers and the load balancing among controllers and characterize the capacitated controller placement problem as a propagation delay minimization problem under controller load constraints. In [30], the authors designed a capacitated controller placement strategy which minimizes the worst case latency. Reference [31] considered the latency requirement of the links between switches and controllers, and those between controllers, and proposed a heuristic clique-based approach to determine the suboptimal controller placement strategy. In this paper, we extended our previous work on controller placement in SDN [32]. Under the assumption that the capacity of the controllers can only be chosen from a certain set containing various pre-determined capacity values, we study the capacitated controller placement problem based on control plane delay minimization and the optimal placement strategy of the controllers is obtained by solving the optimization problem.

3 System model and proposed SDN hierarchical management architecture

In this section, we first present the system model of this paper, then proposed an SDN hierarchical management architecture.

3.1 System model

In this paper, we consider an SDN system in which certain number of switches are deployed in a given

area. Assuming that switches may have different traffic transmission requirements and are allowed to choose controllers from a given specific set, we study the capacitated controller placement problem of the network. We assume that the co-location of switches and controllers is allowed, i.e., each controller can be deployed co-located with one switch. Without loss of generality, we assume that the transmission delay between one controller and its co-located switch can be ignored.

We denote the number of switches in the network as N . Let $V = \{V_1, V_2, \dots, V_N\}$ denote the set of switches, where V_i denotes the i th switch, $1 \leq i \leq N$. To characterize the connection status between switches, we define binary identifier α_{ij} , i.e., we set $\alpha_{ij} = 1$, if V_i is directly connected with V_j in the network; otherwise, $\alpha_{ij} = 0$. In this paper, we assume that the network topology is known; thus, α_{ij} is a known constant. Let C_i denote the i th controller, for convenience, we assume that C_i is co-located with V_i , $1 \leq i \leq N$. We assume that the maximum number of controllers that can be deployed is M , and the capacity of the controllers is specified. Let $\Phi = \{\phi_1, \phi_2, \dots, \phi_M\}$ denote the set of controller capacity, where ϕ_m represents the m th capacity value which can be assigned to one controller. Without loss of generality, the capacity of one controller is defined as the average rate of processing signaling message [33]. All symbols and variables used in this paper are summarized in Table 1.

Figure 1 shows the system model considered in this paper. As an example, in Fig. 1, we assume that the controller C_1 is deployed co-located with switch V_1 ; thus, the signaling interaction between C_1 and V_4 requires a multi-hop path from V_1 to V_4 via V_2 and V_3 . In the case that V_4 tends to transmit user flows to other switches and no flow forwarding rules have been pre-cached, V_4 should send packet-in message to C_1 and then forward user flows according to the received flow forwarding strategy from C_1 . As the signaling message interaction process between V_4 and C_1 results in control plane delay, it is obvious that shorter control plane delay is highly desired. From this example, we can also observe that given the topology of SDN switches, various controller placement strategies may result in different control plane delay. Hence, in this paper, we stress the importance of control plane delay and design capacitated controller placement strategy.

3.2 Proposed SDN hierarchical management architecture

In SDN, the available controller resources need to be managed in a coordinated manner to guarantee the latency requirements of switches and meet the capacity limitation of controllers. In order to achieve the efficient placement of the controllers and achieve centralized

Table 1 List of notations

Notations	Description
V	The set of switches
C	The set of controllers
Φ	The set of controller capacity
v_0	The propagation velocity of electromagnetic wave
β	The average size of signaling messages
μ_i	The capacity of C_i
l_i	The signaling load of C_i
D_{ij}^{\max}	The maximum tolerable control plane delay of V_j
Ψ_j	The candidate controller set of V_j
λ_j	The average arrival rate of the signaling messages sent from V_j
H_{ij}	The total hops of the link between V_j and C_i
L_{ij}^h	The length of the h th hop link between V_j and C_i
R_{ij}^h	The transmission rate of the h th hop link between V_j and C_i
α	Weighting factor for the importance of inter-controller delay
α_{ij}	Connection state variable between V_i and V_j
x_{ij}	Association variable between V_j and C_i
z_{im}	Capacity matching variable between C_i and ϕ_m
y_i	Controller placement variable at C_i
θ_{kn}	The route selection variable between adjacent switches V_k and V_n
D	The control plane delay of the network
D^s	The delay between controllers and switches
D^c	The inter-controller delay
D_{ij}^p	The propagation delay between V_j and C_i
D_{ij}^t	The transmission delay between V_j and C_i
D_i^w	The signaling processing time at C_i
$D_{ij}^{c,p}$	The propagation delay between C_i and C_j
$D_{ij}^{c,t}$	The transmission delay between C_i and C_j

network management, we propose an SDN hierarchical management architecture and design the capacitated controller deployment scheme based on the proposed architecture.

Figure 2 shows the proposed SDN hierarchical management architecture, which is composed of two types of functional entities, i.e., local management entity (LME) and global management entity (GME). The major functions of LME and GME can be summarized as follows.

3.2.1 LME

Being deployed at each switch, LME is responsible for collecting the state information of the switch, such as the average arrival rate of signaling messages, the transmission rate of the links between switches, and the maxi-

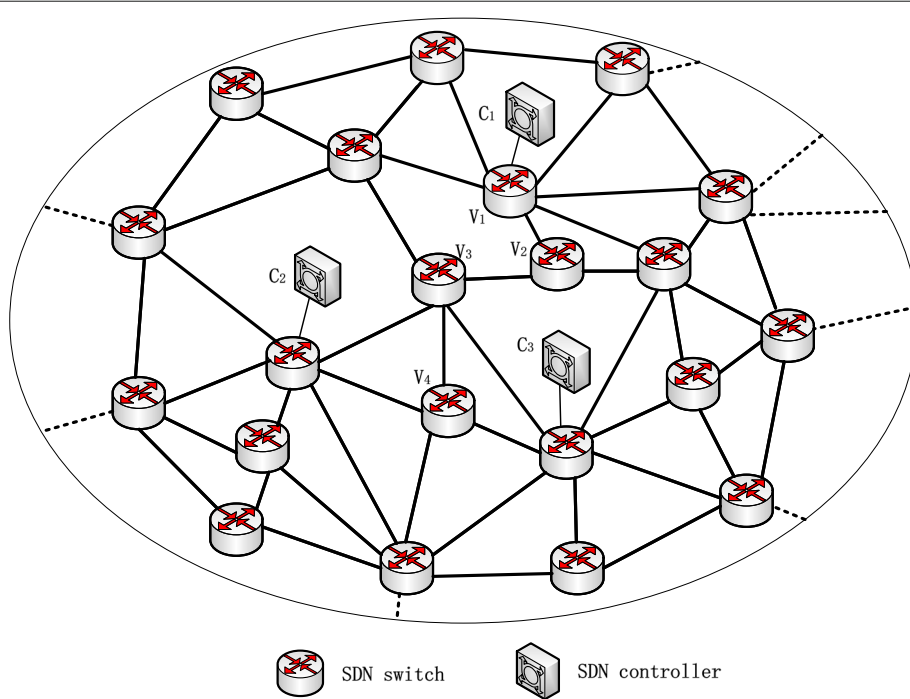


Fig. 1 System model. The system model considered in this paper

imum tolerable control plane delay of the switch. Through interacting with the GME, the LME sends the collected information to the GME. In addition, the LME also receives the obtained controller placement strategy from the GME.

3.2.2 GME

Being deployed over the switches and the controllers, GME acts as a centralized manager of the network. Through interacting with the associated LMEs, GME receives the state information of the switches. Based on

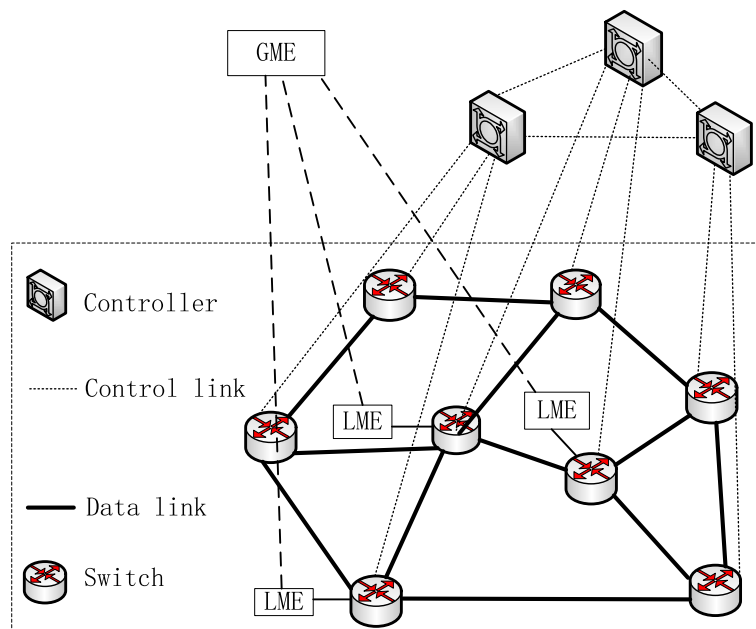


Fig. 2 Proposed SDN Hierarchical Management Architecture. The proposed SDN hierarchical management architecture based on which the proposed capacity controller deployment scheme can be implemented

the obtained information, GME is capable of conducting the proposed capacitated controller placement algorithm and obtaining the controller placement strategy.

4 Capacitated controller placement problem formulation

In this section, we first take into account the flow transmission requirement of the switches and determine the candidate controllers set for individual switches; then, we formulate the capacitated controller placement problem as a control plane delay minimization-based optimization problem.

4.1 Candidate controller set

The switches transmitting different user flows may have various requirement on signaling transmission performance. For instance, in the case of transmitting delay-sensitive user flows, the shorter end-to-end delay and control plane delay are highly desired. On the other hand, some delay-tolerant services may not have high requirement on flow transmission delay and control plane delay. In this paper, we assume that the switches in the SDN might be of different sensitivity on control plane delay, thus place constraints on their associated controllers. Let D_{ij}^s denote the control plane delay when V_j associates with C_i (we will give the mathematical expression of D_{ij}^s in the next subsection) and D_j^{\max} denote the maximum tolerable control plane delay of V_j ; we define the candidate controller set for individual switch to characterize the requirements of different switches, i.e., only the controllers belonging to the candidate controller set of one switch can be selected as the associated controller of the switch. Let Ψ_j denote the candidate controller set of V_j ; we can express Ψ_j as

$$\Psi_j = \left\{ C_i \mid D_{ij}^s \leq D_j^{\max}, 1 \leq i, j \leq N \right\}. \quad (1)$$

4.2 Control plane delay formulation

The control plane delay D of the SDN can be expressed as

$$D = D^s + \alpha D^c \quad (2)$$

where D^s denotes the delay between controllers and switches, D^c denotes the inter-controller delay, $\alpha \in [0, \alpha_0]$ is a weighting factor characterizing the importance of inter-controller delay, and $\alpha_0 \geq 1$ is a given constant.

4.2.1 Delay between controllers and switches

The delay between controllers and switches, i.e., D^s in (2), can be calculated as

$$D^s = \sum_{j=1}^N \sum_{\substack{i=1 \\ C_i \in \Psi_j}}^N x_{ij} D_{ij}^s \quad (3)$$

where x_{ij} denotes the binary association variable between V_j and C_i , i.e., we set $x_{ij} = 1$, if V_j associates with C_i ; otherwise, $x_{ij} = 0$, $1 \leq i, j \leq N$. D_{ij}^s in (3) can be expressed as

$$D_{ij}^s = D_{ij}^p + D_{ij}^t + D_i^w \quad (4)$$

where D_{ij}^p denotes the propagation delay between V_j and C_i , D_{ij}^t denotes the transmission delay between V_j and C_i , and D_i^w denotes the signaling processing time at C_i .

D_{ij}^p in (4) can be calculated as

$$D_{ij}^p = \sum_{h=1}^{H_{ij}} \frac{L_{ij}^h}{v_0} \quad (5)$$

where L_{ij}^h denotes the length of the h th hop link between V_j and C_i , H_{ij} denotes the total hops of the path between V_j and C_i , and v_0 denotes the propagation velocity of electromagnetic wave.

D_{ij}^t in (4) can be expressed as

$$D_{ij}^t = \sum_{h=1}^{H_{ij}} \frac{\lambda_j \beta T_0}{R_{ij}^h} \quad (6)$$

where λ_j denotes the average arrival rate of the signaling messages sent from V_j over time duration T_0 , β denotes the average size of the signaling messages, and R_{ij}^h denotes the transmission rate of the h th hop link between V_j and C_i .

In this paper, we assume that the signaling messages arrived at the controllers are processed in a first-come-first-serve (FCFS) manner; thus, D_i^w in (4) can be calculated as

$$D_i^w = \frac{1}{\mu_i - l_i} \quad (7)$$

where μ_i denotes the capacity of C_i , which can be expressed as

$$\mu_i = \sum_{m=1}^M z_{im} \phi_m \quad (8)$$

where z_{im} is the binary capacity matching variable, i.e., we set $z_{im} = 1$, if the capacity of C_i is set as ϕ_m ; otherwise, $z_{im} = 0$, l_i denotes the signaling load of C_i , which is defined as the total amount of signaling messages arrived at C_i [29], i.e.,

$$l_i = \sum_{j=1}^N x_{ij} \lambda_j \beta. \quad (9)$$

4.2.2 Inter-controller delay

The inter-controller delay denoted by D^c in (2) can be expressed as

$$D^c = \sum_{j=1}^N \sum_{\substack{i=1 \\ i \neq j}}^N y_i y_j D_{ij}^c \quad (10)$$

where y_i denotes the binary controller placement variable of C_i , i.e., we set $y_i = 1$, if C_i is placed at V_{ij} ; otherwise, $y_i = 0$.

In order to create a global view of the network, control signaling interaction between controllers is required. In this subsection, we characterize inter-controller delay as the sum of the propagation delay and the transmission delay of the communication signaling messages transmitted between controllers, i.e.,

$$D_{ij}^c = D_{ij}^{c,p} + D_{ij}^{c,t} \quad (11)$$

where $D_{ij}^{c,p}$ and $D_{ij}^{c,t}$ denote respectively the propagation delay and the transmission delay of the communication signaling transmitted between C_i and C_j . $D_{ij}^{c,p}$ can be calculated as the product of the normalized signaling load of the controllers and the propagation delay between C_i and C_j , i.e.,

$$D_{ij}^{c,p} = \frac{1}{2}(l_i + l_j)D_{ij}^p. \quad (12)$$

$D_{ij}^{c,t}$ can be expressed as

$$D_{ij}^{c,t} = \sum_{h=1}^{H_{ij}} \frac{(l_i + l_j)T_0}{2R_{ij}^h}. \quad (13)$$

4.3 Optimization constraints

The optimal capacitated controller placement strategy should be designed under a number of constraints including association constraints between controllers and switches, controller capacity constraints, and capacity matching constraints.

4.3.1 Controller-switch association constraints

In this paper, we assume that each switch can only associate with a unique controller, i.e.,

$$C1: \sum_{i=1}^N x_{ij} = 1.$$

As switches can only associate with their candidate controllers, we obtain:

$$C2: x_{ij} = 0, \text{ if } C_i \notin \Psi_j.$$

The association variable and controller placement variable should meet the following constraint:

$$C3: x_{ij} \leq y_i.$$

The signaling load arriving at one controller cannot exceed the capacity of the controller, i.e.,

$$C4: \sum_{j=1}^N x_{ij} \lambda_j \beta \leq \mu_i.$$

4.3.2 Controller placement constraints

As the total number of the controllers being placed should be less than M , we obtain

$$C5: \sum_{i=1}^N y_i \leq M.$$

We assume that the capacity of each controller should match one given value in the set Φ , and one given capacity value is assigned to one controller, we obtain

$$C6: \sum_{m=1}^M z_{im} = 1,$$

$$C7: \sum_{i=1}^N z_{im} = 1.$$

4.4 Optimization problem formulation

In this paper, we consider the capacitated controller placement problem in SDN. To stress the importance of control plane delay which is composed of both the transmission and processing delay between controllers and switches, and the inter-controller delay, we formulate controller placement problem as control plane delay minimization problem, which can be expressed as follows:

$$\begin{aligned} \min_{x_{ij}, y_i, z_{im}} \quad & D \\ \text{s.t.} \quad & C1 - C7. \end{aligned} \quad (14)$$

5 Solution to the optimization problem

The formulated optimization problem in (14) is a complicated nonlinear integer programming problem which cannot be easily solved. In this section, we transform it into controller-switch association subproblem and controller capacity matching subproblem and solve the two subproblems successively [14].

5.1 Controller-switch association subproblem

In this subsection, we first assume that the capacity of the controllers is large enough; thus, the signaling processing time at C_i , denoted by D_{ij}^w , is negligible in comparison with the propagation delay and transmission delay. We can then reduce the formulated optimization problem in (14) to a controller-switch association subproblem, i.e.,

$$\min_{x_{ij}, y_i} \sum_{j=1}^N \sum_{i=1}^N x_{ij} (D_{ij}^p + D_{ij}^t + y_i y_j D_{ij}^c) \quad (15)$$

s.t. C1–C5 in (14).

To solve the above controller-switch association sub-problem, we propose a heuristic algorithm based on Dijkstra and K-means algorithms [34, 35].

5.1.1 Dijkstra-based optimal route selection strategy

Examining the objective function in (15), we can see that for any two nonadjacent switches, V_i and V_j , the delay terms D_{ij}^p and D_{ij}^t may involve multi-hop links, hence may not be calculated easily. In this paper, we apply the Dijkstra algorithm, which is widely used to solve the shortest path problem in a weighted directed graph, to compute the optimal route between V_i and V_j , and to obtain the minimal D_{ij}^p and the minimal D_{ij}^t accordingly.

Considering the multi-hop transmission between V_i and V_j , we can express D_{ij}^p as

$$D_{ij}^p = \sum_{k=1}^N \sum_{n=1, n \neq k}^N \theta_{kn} D_{kn}^p, \text{ for } \alpha_{kn} = 1 \quad (16)$$

where $\theta_{kn} \in \{0, 1\}$, $k \neq n$ is the route selection variable between adjacent switches V_k and V_n , i.e., $\theta_{kn} = 1$, if V_k and V_n are adjacent switches and the data packets pass through the link between V_k and V_n when being transmitted from V_i to V_j . To guarantee the end-to-end transmission from V_i to V_j , θ_{kn} should meet the following constraints:

$$\sum_{n=1, n \neq k}^N \theta_{kn} = 1, \text{ if } V_k = V_i, \quad (17)$$

$$\sum_{k=1, k \neq n}^N \theta_{kn} = \sum_{n'=1, n' \neq n}^N \theta_{nn'}, \text{ if } V_n \neq V_i \neq V_j, \quad (18)$$

$$\sum_{k=1, k \neq n}^N \theta_{kn} = 1, \text{ if } V_n = V_j. \quad (19)$$

The optimization problem of determining the optimal route between V_i and V_j which achieves the minimal D_{ij}^p can then be formulated as

$$\min_{\theta_{kn}} D_{ij}^p \quad (20)$$

s.t. C1 : (17),
C2 : (18),
C3 : (19).

Applying the Dijkstra algorithm to solve the optimization problem formulated in (20), we create a weighted

directed graph $G = (V, E, W)$, where $E = \{E_{kn}\}$ is the link set, E_{kn} represents the link between V_k and V_n , $W = \{w_{kn}\}$ is the weight set, and w_{kn} is the weight of E_{kn} , which can be expressed as

$$w_{kn} = \frac{L_{kn}}{v_0}, \text{ for } \alpha_{kn} = 1 \quad (21)$$

where L_{kn} denotes the length between V_k and V_n .

By solving the shortest path problem for G , we can obtain the optimal route between V_i and V_j offering the minimum end-to-end propagation delay. Let θ_{kn}^* denote the optimal route selection strategy; the optimal propagation delay between V_i to V_j , denoted by $D_{ij}^{p,*}$, can be calculated accordingly.

Following the similar manner, we are capable of calculating the optimal transmission delay between V_i and V_j , denoted by $D_{ij}^{t,*}$. In addition, substituting $D_{ij}^{p,*}$ in (12), and re-calculating (11), we can also obtain the updated propagation delay between controllers C_i and C_j , denoted by $D_{ij}^{c,*}$.

5.1.2 K-means-based controller placement strategy

Given $D_{ij}^{p,*}$, $D_{ij}^{t,*}$, and $D_{ij}^{c,*}$, the controller-switch association subproblem formulated in (15) can be rewritten as

$$\min_{x_{ij}, y_i} \sum_{j=1}^N \sum_{i=1}^N x_{ij} (D_{ij}^{p,*} + D_{ij}^{t,*} + y_i y_j D_{ij}^{c,*}) \quad (22)$$

s.t. C1–C5 in (14).

Since one controller in the network may associate with a number of switches, we may form a cluster with the controller being the cluster head while its associated switches being the cluster members. Hence, the SDN network topology can be regarded as a cluster-based form. By applying clustering methods, the reasonable controller placement and controller-switch association strategy can be obtained.

In this paper, we apply K-means algorithm [21], which is an efficient clustering method to determine the number and location of the controllers and propose a K-means algorithm-based optimal controller placement strategy. The steps of the proposed algorithm can be described as follows:

1. Determine the first controller: for each switch, calculate the sum of the transmission delay and propagation delay when connecting with all other switches, i.e., for $\forall V_j$, compute $\sum_{i=1}^N (D_{ij}^{t,*} + D_{ij}^{p,*})$, and select the switch which offers the minimum sum of the transmission delay and propagation delay to place the first controller, i.e.,
 $C_{i_1} = \operatorname{argmin} \sum_{j=1}^N (D_{ij}^{t,*} + D_{ij}^{p,*}), y_{i_1}^* = 1;$
2. Set $m = 2;$
3. Determine the m th controller: Calculate the sum of the transmission delay and propagation delay

between each switch and deployed controllers, select the switch which offers the minimum propagation delay and transmission delay to deploy the m th controller, i.e., $C_{i_m} = \operatorname{argmin} \sum_{i_0=i_1}^{i_{m-1}} (D_{i_0 i_1}^{t,*} + D_{i_0 i_1}^{p,*})$, where C_{i_m} represents the m th controller which is deployed at V_{i_m} , $y_{i_m}^* = 1$;

4. Determine controller-switch association strategy, i.e., $\{x_{i_1}^*, x_{i_2}^*, \dots, x_{i_m}^*\} = \operatorname{argmin} \sum_{i_0=i_1}^{i_m} \sum_{j=1}^N (D_{i_0 j}^{p,*} + D_{i_0 j}^{t,*}) + \sum_{i_0=i_1}^{i_m} \sum_{i_0' \neq i_0}^{i_m} D_{i_0 i_0'}^{c,*}$;
5. Set $m = m + 1$; if $m < M$, repeat step 3 and step 4.

The proposed controller-switch association algorithm is described in Algorithm 1.

5.2 Controller capacity matching subproblem

In this subsection, we assume that M controllers have been placed and all the switches have been associated with one controller, then solve the controller capacity matching subproblem. Assuming C_i is placed at V_i , i.e., $y_i^* = 1$, and

Algorithm 1 Proposed controller-switch association algorithm

Require: $\alpha_{ij}, L_{ij}^h, R_{ij}^h, \lambda_i, \beta$;

Ensure: x_{ij}, y_i ;

- 1: **for** $i = 1 : N$
 - 2: **for** $j = 1 : N$
 - 3: **if** $\alpha_{ij} = 1$
 - 4: **then** $D_{ij}^p = \frac{L_{ij}^h}{v_0}, D_{ij}^t = \frac{\lambda_j \beta}{T_{ij}^h}$;
 - 5: **end if**
 - 6: **end for**
 - 7: **end for**
 - 8: **for** $i = 1 : N$
 - 9: **for** $j = 1 : N$
 - 10: **if** $\alpha_{ij} \neq 1$
 - 11: **then** Applying Dijkstra algorithm to obtain D_{ij}^p , D_{ij}^t
 - 12: **end if**
 - 13: **end for**
 - 14: **end for**
 - 15: Applying K-means algorithm to select the C_{i_1} th switch to deploy the first controller, i.e., $C_{i_1} = \operatorname{argmin} \sum_{j=1}^N (D_{i_1 j}^{p,*} + D_{i_1 j}^{t,*})$.
 - 16: **for** $m = 2 : M$
 - 17: Deploy the k th controller: $C_{i_k} = \operatorname{argmin} \sum_{i_0=i_1}^{i_{m-1}} (D_{i_0 i_1}^{t,*} + D_{i_0 i_1}^{p,*})$;
 - 18: Determine controller-switch association strategy: $\{x_{i_1}^*, x_{i_2}^*, \dots, x_{i_m}^*\} = \operatorname{argmin} \sum_{i_0=i_1}^{i_m} \sum_{j=1}^N (D_{i_0 j}^{p,*} + D_{i_0 j}^{t,*}) + \sum_{i_0=i_1}^{i_m} \sum_{i_0' \neq i_0}^{i_m} D_{i_0 i_0'}^{c,*}$.
 - 19: **end for**
-

V_j has been associated with C_i , i.e., $x_{ij}^* = 1, 1 \leq i, j \leq N$, the optimal controller capacity matching subproblem can be formulated as

$$\min_{z_{im}} \sum_{j=1}^N \sum_{i=1}^N x_{ij}^* (D_{ij}^{t,*} + D_{ij}^{p,*} + D_{ij}^w) + y_i^* y_j^* D_{ij}^{c,0} \quad (23)$$

s.t. C6, C7 in (14)

where $D_{ij}^{c,0}$ denotes the inter-controller delay corresponding to the obtained controller placement strategy y_i^* and switch-controller association strategy x_{ij}^* . Since $D_{ij}^{p,*}, D_{ij}^{t,*}$, and $D_{ij}^{c,0}$ are all constants, the controller capacity matching optimization problem formulated in (23) can be converted into the following optimization problem:

$$\min_{z_{im}} \sum_{i=1}^N \frac{1}{\sum_{m=1}^M z_{im} \phi_m - l_i} \quad (24)$$

s.t. C6, C7 in (14).

Under the given constraints, the objective function of the above optimization problem can be rewritten as

$$\begin{aligned} \sum_{i=1}^N \frac{1}{\sum_{m=1}^M z_{im} \phi_m - l_i} &= \sum_{i=1}^N \frac{1}{\sum_{m=1}^M z_{im} (\phi_m - l_{i_m})} \\ &= \sum_{i=1}^N \sum_{m=1}^M z_{im} \frac{1}{\phi_m - l_{i_m}} \end{aligned} \quad (25)$$

where $l_{i_m} = \sum_{j=1}^N x_{i_m j}^* \lambda_j \beta$. Hence, the optimization model formulated in (25) is equivalent to the following problem:

$$\min_{z_{im}} \sum_{i=1}^M \sum_{m=1}^M z_{im} \frac{1}{\phi_m - l_{i_m}} \quad (26)$$

s.t. C6, C7 in (14)

where $l_{i_m} = \sum_{j=1}^N x_{i_m j}^* \lambda_j \beta$. Since the above optimization problem can be considered as an optimal matching problem in a weighted bipartite graph, it can be solved by the K-M algorithm [36].

Applying the K-M algorithm to solve the optimization problem in (26), we create a weighted network bipartite graph $G^0 = (V_1, \Phi, E')$, where the set of vertices V_1 denotes the collection of the controllers, i.e., $V_1 = \{C_{i_1}, C_{i_2}, \dots, C_{i_M}\}$ and $E' = \{E_{i_m, \phi_m}\}$ denotes the set of edge weight, and $E_{i_m, \phi_m} = \frac{1}{\phi_m - l_{i_m}}$.

The steps for solving the optimal controller capacity matching subproblem based on the K-M algorithm can be described as follows.

1. Find an initial feasible vertex labeling and determine G_0^l from G_0 ;
2. A distribution H is selected in G_0^l ;

3. If H is perfect, then the optimization problem is solved. Otherwise, the label having not being allocated by the distribution H is selected in G_0^l . Set $S = V_1$, and $T = \psi$, which denotes the empty set;
4. Let $N_{G_0^l}(S)$ denote the collection of points which connect with S in G_0^l . If $N_{G_0^l}(S) \neq T$, go to step 2. Otherwise, $N_{G_0^l}(S) = T$. Find

$$\Delta = \min\{l(u) + l(v) \geq w(u, v) | u \in S, v \in V_2 - T\} \quad (27)$$

and replace the existing labeling l with l^* by

$$l'(u) = \begin{cases} l(u) - \Delta, & u \in S \\ l(u) + \Delta, & u \in T \\ l(u), & \text{others.} \end{cases} \quad (28)$$

This process is conducted iteratively until an optimal match is obtained. The proposed controller capacity matching algorithm is described in Algorithm 2.

Algorithm 2 Proposed controller capacity matching algorithm

- 1: Create the weighted network bipartite graph $G^0 = (V_1, \Phi, E')$
 - 2: $V_1 = \{C_{i_1}, C_{i_2}, \dots, C_{i_M}\}$, $\Phi = \{\phi_1, \phi_2, \dots, \phi_M\}$, $E' = \{E_{i_m, \phi_m}\}$, where $E_{i_m, \phi_m} = \frac{1}{\phi_m - l_{i_m}}$
 - 3: Find an initial feasible vertex labeling and determine G_0^l from G_0
 - 4: A distribution H is selected in G_0^l
 - 5: **if** H is perfect **then**
 - 6: Obtain the optimal controller capacity matching strategy
 - 7: **else** the label having not being allocated by the distribution H is selected in G_0^l
 - 8: Set $S = V_1$, and $T = \psi$
 - 9: Let $N_{G_0^l}(S)$ denote the collection of points which connect with S in G_0^l
 - 10: **if** $N_{G_0^l}(S) \neq T$ **then**
 - 11: Go to Step 4
 - 12: **else** $N_{G_0^l}(S) = T$
 - 13: Find $\Delta = \min\{l(u) + l(v) \geq w(u, v) | u \in S, v \in V_2 - T\}$
 - 14: Replace the existing labeling l with l^* by

$$l'(u) = \begin{cases} l(u) - \Delta, & u \in S \\ l(u) + \Delta, & u \in T \\ l(u), & \text{others} \end{cases}$$
 - 15: **until** an optimal match strategy is obtained
 - 16: **end if**
 - 17: **end if**
-

It should be mentioned that the proposed Dijkstra-based optimal route selection strategy, K-means-based

optimal controller placement strategy, and the controller capacity matching algorithm should be executed based on the proposed SDN hierarchical management architecture. More specifically, the LMEs collect the status information of their associated switches, such as the maximum tolerable control plane delay of the switch D_j^{\max} , the average size of signaling messages β , and the average arrival rate of signaling messages λ_j . Through interacting with the associated LMEs, GME receives the state information of the switches. Based on the information obtained, the GME may conduct the proposed capacitated controller placement algorithm, obtain the controller deployment and matching strategy, characterized by $x_{i,j}$, y_i , $z_{i,m}$, and send the strategy to the LMEs.

6 Complexity analysis

In this section, we analyze the computational complexity of the proposed algorithm. As two subproblems, i.e., controller-switch association subproblem and controller capacity matching subproblem, are successively solved, we examine the complexity for solving the two subproblems, respectively.

6.1 Controller-switch association subproblem

For controller-switch associated subproblem, the Dijkstra algorithm is applied to calculate the optimal route between V_i and V_j , $1 \leq i, j \leq N$. As the number of switches in the network is denoted by N , then computational complexity can be calculated as $O(N \log N)$. Applying K-means algorithm to solve the controller-switch association strategy, the computational complexity can be computed as $O(NMT)$, where M denotes the number of cluster heads (controllers) and T denotes the number of iterations.

6.2 Controller capacity matching subproblem

Based on the controller placement strategy obtained from previous controller-switch association subproblem, we formulate the controller capacity matching subproblem and obtained the matching strategy through applying the K-M algorithm. Given the number of controllers M , it can be shown that the complexity of the algorithm is $O(M^4)$.

7 Performance evaluation

In this section, we provide numerical results of the proposed algorithm and compare them with the previously proposed controller placement algorithms. In the simulation, we consider an SDN scenario consisting of a number of SDN switches which are randomly deployed in 2000 km \times 2000 km square region. That is, the position of the switches follows uniform random distribution in the given simulation region. We assume that there is a random connection between any two switches in the region. The transmission rate of links is randomly selected from a set

of values. To examine the characteristics of transmission links on the control plane delay, we consider three link characteristics cases, as summarized in Table 2.

In the simulation, the size of the packet-in message β is set as 160 bytes [29], and the arrival rate of the packet-in message requests of the switches is represented by λ_j , which are distributed in different ranges. In order to examine the difference of λ_j , we set three message request scenarios in the simulation. More specifically, different scenarios contain various percentage of the switches with λ_j being chosen from certain range; the specific simulation parameters are shown in Table 3. Simulation results are averaged over 400 independent processes involving independent instances.

Figure 3 shows the relationship between the control plane delay and the number of controllers obtained from different message request scenarios for link status case 1. We also compare the results obtained from the proposed scheme and the scheme proposed in [21]. We can see that for case 1, the control plane delay initially decreases as the number of controllers increases for three message request scenarios. However, when the number of controllers deployed in the network reaches a certain value, the control plane delay begins to increase as the number of controllers increases. This is because when the number of controllers increases, the inter-controller delay increases rapidly. Comparing the three curves in the figure, we can see that the proposed scheme outperforms the scheme proposed in [21]. The reason is that the algorithm proposed in [21] mainly aims to minimize the propagation delay between controllers and switches and ignores the transmission delay between controllers and switches, the processing delay at the controllers, and the inter-controller delay. In addition, the simulation results obtained for the three message request scenarios show that scenario 1 provides the best latency performance, while scenario 3 provides the worst latency performance. The reason is that the service request of the switches in scenario 1 is relatively low compared to the other two scenarios; hence, a minimum control plane delay can be obtained.

In Fig. 4, we examine the relationship between the control plane delay and the number of controllers for different link cases, i.e., case 2 and case 3 under scenario 3 of message requests. Similar to Fig. 3, we can see that for scenario 3, the control plane delay initially decreases as the number of controllers increases for two

Table 2 Transmission rate of links

Link cases	Transmission rate of links (Mbps)
Case 1	$R_{ij}^h \in \{50, 100, 150, 200, 300, 500\}$
Case 2	$R_{ij}^h \in \{50, 75, 100\}$
Case 3	$R_{ij}^h \in \{150, 200, 300\}$

Table 3 Message request scenarios

Scenarios	$\lambda_j \in [1k, 10k]$	$\lambda_j \in [10k, 100k]$	$\lambda_j \in [100k, 200k]$
Scenario 1	100% switches	0	0
Scenario 2	80% switches	20% switches	0
Scenario 3	60% switches	20% switches	20% switches

transmission rate cases. However, when the number of controllers deployed in the network reaches a certain value, the control plane delay begins to increase with the increase of controller number. This is because the inter-controller delay increases as the number of controllers increases. Comparing the results obtained from our proposed algorithm and the algorithm proposed in [21], we can see that the proposed scheme offers lower control plane delay than the scheme proposed in [21]. Simulation results obtained for different link cases indicate that case 3 provides a lower control plane delay and case 2 provides a higher control plane delay; this is benefited from the better transmission performance of the switches in case 3.

Figure 5 shows the relationship between the control plane delay and the number of controllers for different average size of the signaling messages (denoted by β). The link status and message request scenario are chosen as case 1 and scenario 2, respectively, in plotting the figure. We also compare the results obtained from the proposed scheme and the scheme proposed in [22] and [31]. We can see that the proposed algorithm offers a smaller controller plane delay than the scheme proposed in [22] and [31]. The reason is that the algorithm proposed in [22] failed to consider the processing delay at the controllers and the inter-controller delay, and the algorithm proposed in [31] assumed that the controllers are of the same capacity and failed to consider the capacity matching problem of the controllers, thus resulting in undesired controller plane delay. In addition, comparing the results obtained from different β , we can see that smaller β offers better delay performance, which is benefited from a lower load of message requests.

In Fig. 6, we consider the controller placement in different message request scenarios when the link status is case 1, and examine the relationship between the control plane and the number of switches. It can be seen from the figure, as the complexity of the network topology increases, the control plane delay increases accordingly. For comparison, we also plot the simulation results of the algorithm proposed in [21]. It can be seen from the figure that our proposed algorithm provides better latency performance. Furthermore, the simulation results obtained for the three message request scenarios show that scenario 1 provides the best latency performance, while scenario 3 provides the worst latency performance.

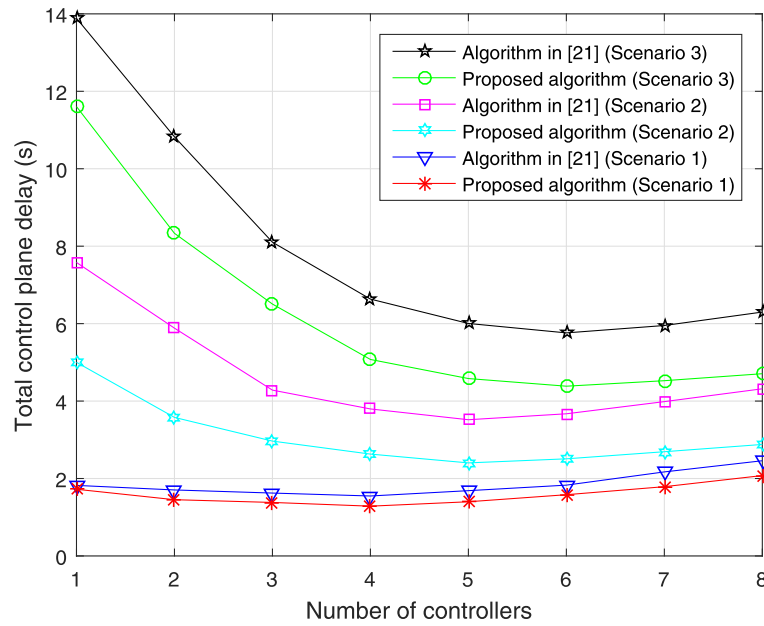


Fig. 3 Control plane delay versus the number of controllers (different scenarios). The control plane delay versus the number of controllers for different scenarios

Figure 7 examines the relationship between control plane delay and the number of switches by varying the transmission rate of the switches, i.e., case 2 and case 3 under scenario 3 of message requests. As can be seen from the figure, the increase in the number of switches leads to an increase in control plane

delay, and the delay performance obtained from case 3 is better than the delay performance obtained from case 2.

In Fig. 8, we examine the control plane delay versus the number of switches obtained from proposed scheme and random matching scheme which matches

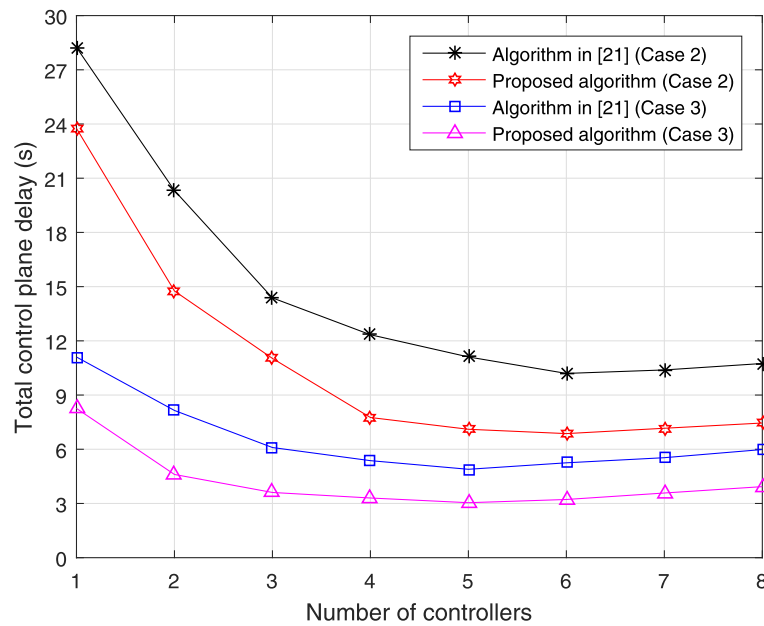


Fig. 4 Control plane delay versus the number of controllers (different cases). The control plane delay versus the number of controllers for different cases

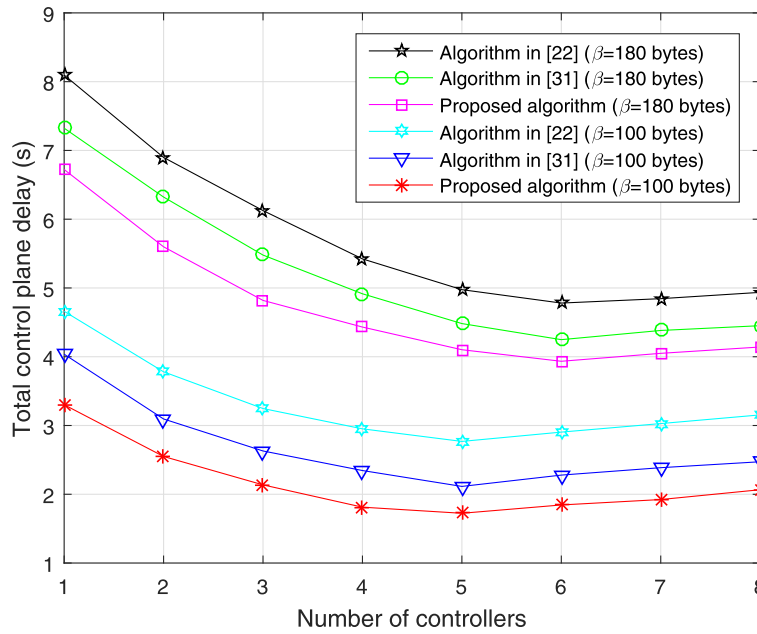


Fig. 5 Control plane delay versus the number of controllers (different β). The control plane delay versus the number of controllers for different β

the capacity of the controller to the values in the given capacity set randomly. For comparison, we also plot the control plane delay obtained for different average size of the signaling messages denoted by β . The link status and message request scenario are chosen as case 1 and scenario 3, respectively, in plotting the figure.

We can see that the proposed algorithm provides a smaller controller plane delay than the random matching algorithm. The reason is that the proposed controller matching scheme aims at minimizing the controller plane delay, thus resulting in desired delay performance. Comparing the results obtained from different β , we

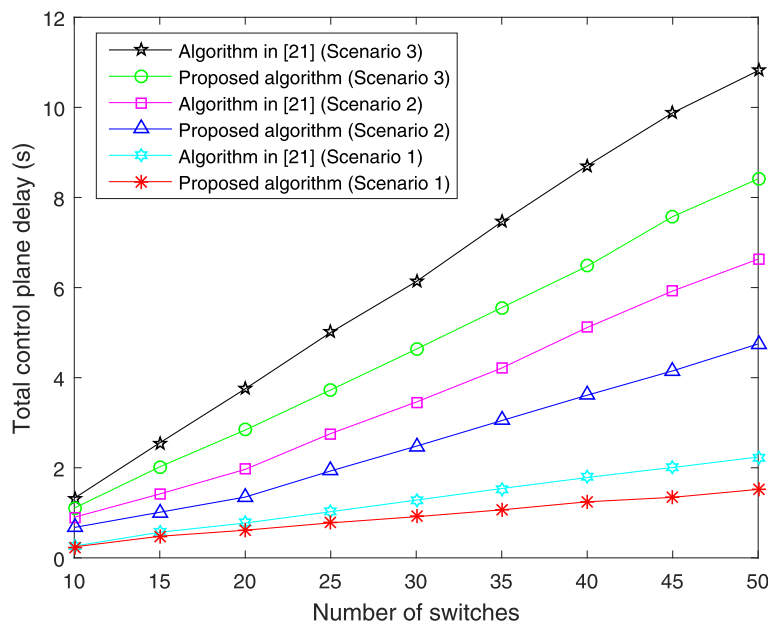


Fig. 6 Control plane delay versus the number of switches (different scenarios). The control plane delay versus the number of switches for different scenarios

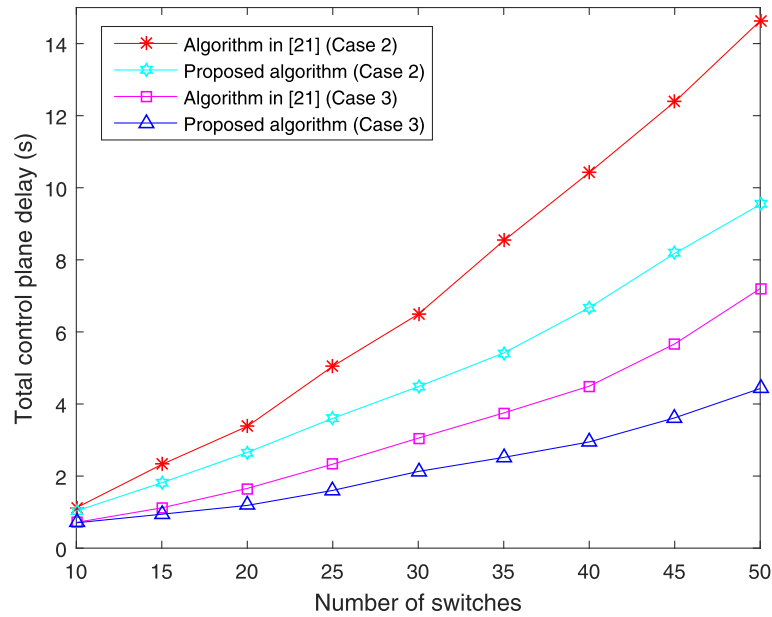


Fig. 7 Control plane delay versus the number of switches (different cases). The control plane delay versus the number of switches for different cases

can see that smaller β offers better delay performance, which is benefited from a lower load of message requests.

8 Conclusion

In this paper, we consider the capacitated controller placement problem in SDN. To achieve the efficient placement

of the controllers, we first propose an SDN hierarchical management architecture. Stressing the importance of control plane delay of the network, we formulate a control plane delay minimization problem subject to controller-switch association, controller capacity constraints, etc. To solve the formulated optimization problem, we transform it into controller-switch association subproblem and

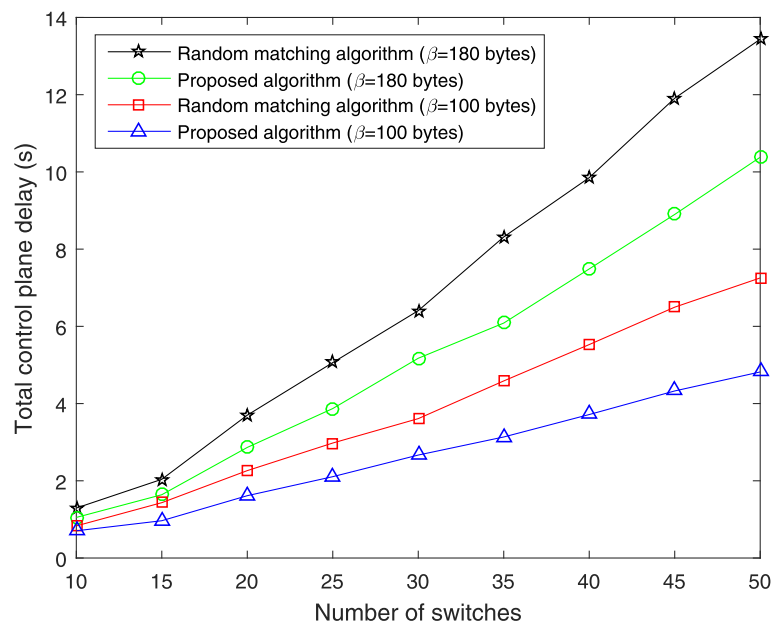


Fig. 8 Control plane delay versus the number of switches (different β). The control plane delay versus the number of switches for different β

controller capacity matching subproblem. To solve the controller-switch association subproblem, we propose a heuristic algorithm based on Dijkstra algorithm and K-means algorithm. Given the controller-switch association strategy, we then apply the Kuhn-Munkres (K-M) algorithm to solve the controller capacity matching subproblem and obtain the corresponding capacitated controller placement strategy. Simulation results validate the effectiveness of the proposed algorithm.

Abbreviations

API: Application programming interface; GME: global management entity; K-M: Kuhn-Munkres; LME: local management entity; QoS: quality of service; SDN: software-defined networking

Acknowledgements

We gratefully acknowledge the anonymous reviewers for taking time reviewing our manuscript.

Authors' contributions

RC proposed the idea and revised this paper. QY and LZ wrote the manuscript and participated in the simulation. QC gave some suggestions and participated in the paper revision. All authors have contributed to this research work. All authors have read and approved the final manuscript.

Funding

This work was supported in part by the National Science and Technology Specific Project of China (2016ZX03001010-004), in part by the National Natural Science Foundation of China (No.61571073) and in part by the Joint Scientific Research Fund of Ministry of Education and China Mobile (MCM20160105).

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests

The authors declare that they have no competing interests.

Received: 6 May 2019 Revised: 3 November 2019 Accepted: 28 November 2019

Published online: 26 December 2019

References

- P. Mell, T. Grance, The NIST definition of cloud computing. *Commun. ACM.* **53**(6), 50 (2011)
- J. Zhao, Y. Liu, Y. Gong, C. Wang, L. Fan, A dual-link soft handover scheme for C/U plane split network in high-speed railway. *IEEE Access.* **6**, 12473–12482 (2018). <https://doi.org/10.1109/ACCESS.2018.2794770>
- W. Xia, Y. Wen, C. H. Foh, D. Niyato, A survey on software-defined networking. *Communications Surveys Tutorials IEEE.* **17**(1), 27–51 (2015)
- D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: a comprehensive survey. *Proc IEEE.* **103**(1), 14–76 (2015). <https://doi.org/10.1109/JPROC.2014.2371999>
- T. Das, V. Sridharan, M. Gurusamy, A survey on controller placement in SDN. *Commun. Surv. Tutor. IEEE.* **1**–1 (2019). <https://doi.org/10.1109/COMST.2019.2935453>
- J. Lu, Z. Zhang, T. Hu, P. Yi, J. Lan, A survey of controller placement problem in software-defined networking. *IEEE Access.* **7**, 24290–24307 (2019). <https://doi.org/10.1109/ACCESS.2019.2893283>
- A. Ksentini, M. Bagaa, T. Taleb, in *2016 IEEE Global Communications Conference (GLOBECOM)*, On using SDN in 5G: the controller placement problem, (2016), pp. 1–6. <https://doi.org/10.1109/GLOCOM.2016.7842066>
- G. Wang, Y. Zhao, J. Huang, W. Wang, The controller placement problem in software defined networking: a survey. *IEEE Netw.* **31**(5), 21–27 (2017). <https://doi.org/10.1109/MNET.2017.1600182>
- A. Ruiz-Rivera, K. W. Chin, S. Soh, Greco: an energy aware controller association algorithm for software defined networks. *IEEE Commun. Lett.* **19**(4), 541–544 (2015). <https://doi.org/10.1109/LCOMM.2015.2394457>
- Y. Hu, T. Luo, N.C. Beaulieu, C. Deng, The energy-aware controller placement problem in software defined networks. *IEEE Commun. Lett.* **21**(4), 741–744 (2017). <https://doi.org/10.1109/LCOMM.2016.2645558>
- L. Yao, P. Hong, W. Zhang, J. Li, D. Ni, in *2015 IEEE International Conference on Communication Workshop (ICCW)*, Controller placement and flow based dynamic management problem towards SDN, (2015), pp. 363–368. <https://doi.org/10.1109/ICCW.2015.7247206>
- M. F. Bari, A. R. Roy, S. R. Chowdhury, Z. Qi, in *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*, Dynamic controller provisioning in software defined networks, (2013). <https://doi.org/10.1109/cnsm.2013.6727805>
- A. Ksentini, M. Bagaa, T. Taleb, I. Balasingham, in *IEEE International Conference on Communications (ICC)*, On using bargaining game for optimal placement of SDN controllers, (2016), pp. 1–6. <https://doi.org/10.1109/ICC.2016.7511136>
- T. Wang, F. Liu, H. Xu, An efficient online algorithm for dynamic SDN controller assignment in data center networks. *IEEE/ACM Trans. Netw.* **25**(5), 2788–2801 (2017). <https://doi.org/10.1109/TNET.2017.2711641>
- W. Ren, Y. Sun, H. Luo, M. Guizani, A novel control plane optimization strategy for important nodes in SDN-IoT networks. *IEEE Internet Things J.* **6**(2), 3558–3571 (2019). <https://doi.org/10.1109/JIOT.2018.2888504>
- M. T. I. Huque, W. Si, G. Jourjon, V. Gramoli, Large-scale dynamic controller placement. *IEEE Trans. Netw. Serv. Manag.* **14**(1), 63–76 (2017). <https://doi.org/10.1109/TNSM.2017.2651107>
- A. Sallahi, M. St-Hilaire, Optimal model for the controller placement problem in software defined networks. *IEEE Commun. Lett.* **19**(1), 30–33 (2015). <https://doi.org/10.1109/LCOMM.2014.2371014>
- A. Sallahi, M. St-Hilaire, Expansion model for the controller placement problem in software defined networks. *IEEE Commun. Lett.* **21**(2), 274–277 (2017). <https://doi.org/10.1109/LCOMM.2016.2621746>
- A. Dvir, Y. Haddad, A. Zilberman, The controller placement problem for wireless SDN. *Wirel. Netw.* (2019). <https://doi.org/10.1007/s11276-019-02077-5>
- T. Y. Cheng, M. Wang, X. Jia, in *2015 IEEE Global Communications Conference (GLOBECOM)*, QoS-guaranteed controller placement in SDN, (2015), pp. 1–6. <https://doi.org/10.1109/GLOCOM.2015.7416960>
- G. Wang, Y. Zhao, J. Huang, Q. Duan, J. Li, in *2016 IEEE International Conference on Communications (ICC)*, A k-means-based network partition algorithm for controller placement in software defined network, (2016), pp. 1–6. <https://doi.org/10.1109/ICC.2016.7511441>
- W. Chen, C. Chen, X. Jiang, L. Liu, Multi-controller placement towards SDN based on Louvain heuristic algorithm. *IEEE Access.* **6**, 49486–49497 (2018). <https://doi.org/10.1109/ACCESS.2018.2867931>
- M. He, A. Basta, A. Blenk, W. Kellerer, in *2017 IEEE International Conference on Communications (ICC)*, Modeling flow setup time for controller placement in SDN: evaluation for dynamic flows, (2017), pp. 1–7. <https://doi.org/10.1109/ICC.2017.7996654>
- Y. Qi, D. Wang, W. Yao, H. Li, Y. Cao, in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Towards multi-controller placement for SDN based on density peaks clustering, (2019), pp. 1–6. <https://doi.org/10.1109/ICC.2019.8761814>
- M. Tanha, D. Sajjadi, J. Pan, in *2016 IEEE Global Communications Conference (GLOBECOM)*, Enduring node failures through resilient controller placement for software defined networks, (2016), pp. 1–7. <https://doi.org/10.1109/GLOCOM.2016.7841786>
- S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, M. Hoffmann, Heuristic approaches to the controller placement problem in large scale SDN networks. *IEEE Trans. Netw. Serv. Manag.* **12**(1), 4–17 (2015). <https://doi.org/10.1109/TNSM.2015.2402432>
- Z. Fan, J. Yao, X. Yang, Z. Wang, X. Wan, in *2019 28th Wireless and Optical Communications Conference (WOCC)*, A multi-controller placement strategy based on delay and reliability optimization in SDN, (2019), pp. 1–5. <https://doi.org/10.1109/WOCC.2019.8770551>
- G. Wang, Y. Zhao, J. Huang, Y. Wu, An effective approach to controller placement in software defined wide area networks. *IEEE Trans. Netw. Serv. Manag.* **15**(1), 344–355 (2018). <https://doi.org/10.1109/TNSM.2017.2785660>
- G. Yao, G. Bi, Y. Li, L. Guo, On the capacitated controller placement problem in software defined networks. *IEEE Commun. Lett.* **18**(8), 1339–1342 (2014)

30. B. P. R. Killi, S. V. Rao, Capacitated next controller placement in software defined networks. *IEEE Trans. Netw. Serv. Manag.* **14**(3), 514–527 (2017). <https://doi.org/10.1109/TNSM.2017.2720699>
31. M. Tanha, D. Sajjadi, R. Ruby, J. Pan, Capacity-aware and delay-guaranteed resilient controller placement for software-defined WANs. *IEEE Trans. Netw. Serv. Manag.* **15**(3), 991–1005 (2018). <https://doi.org/10.1109/TNSM.2018.2829661>
32. L. Zhu, R. Chai, Q. Chen, in *WCSP*. Control plane delay minimization based SDN controller placement scheme, (2017), pp. 1–6. <https://doi.org/10.1109/ICC.2017.7996654>
33. Y. Zhao, L. Iannone, M. Riguidel, in *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*. On the performance of SDN controllers: a reality check, (2016). <https://doi.org/10.1109/nfv-sdn.2015.7387410>
34. P. Bogdan, in *International Carpathian Control Conference (ICCC)*. Dijkstra algorithm in parallel-case study, (2015), pp. 50–53. <https://doi.org/10.1109/ICCW.2015.7247206>
35. Z. Junhui, Y. Tao, G. Yi, W. Jiao, F. Lei, Power control algorithm of cognitive radio based on non-cooperative game theory. *China Commun.* **10**(11), 143–154 (2013). <https://doi.org/10.1109/CC.2013.6674218>
36. X. Zhou, L. Yang, D. Yuan, Bipartite matching based user grouping for grouped OFDM-IDMA. *IEEE Trans. Wirel. Commun.* **12**(10), 5248–5257 (2013). <https://doi.org/10.1109/TWC.2013.090413.130097>

Publisher's Note

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

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
