Computer Networks 197 (2021) 108326

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Network cost optimization-based capacitated controller deployment for SDN

Rong Chai *, Xizheng Yang, Chunling Du, Qianbin Chen

School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

ARTICLE INFO

Keywords:
Software-defined networking
Controller deployment
Response time
Network cost

ABSTRACT

As a novel network paradigm, software-defined networking (SDN) is capable of simplifying network management and offering flexible support to various user services. In order to meet the rapidly increasing transmission demands of SDN switches, the controller deployment strategy in an SDN scenario should be designed. In this paper, we investigate the capacitated controller deployment problem for SDN. Consider the signaling transmission and processing performance of switches and address the worst-case performance, we define network response time (NRT) as the maximum control plane response time of switches. Then aiming to achieve the tradeoff between NRT and the cost of controllers, we introduce the concept of network cost which is defined as the weighted sum of NRT and controller cost. The capacitated controller deployment problem is formulated as a constrained network cost minimization problem. To solve the optimization problem, we propose a two-stage heuristic algorithm, which first tackles the controller deployment subproblem under the unlimited capacity constraint, and then solves controller-type matching subproblem. Specifically, during the first stage, a minimum eccentricity-based controller deployment algorithm is designed to determine the number and location of controllers as well as the association strategy between controllers and switches. During the second stage, a greedy method-based controller-type matching strategy is proposed to determine the types of deployed controllers. Extensive simulations are performed and the results certify the effectiveness of the proposed algorithm.

1. Introduction

Software-defined networking (SDN) has emerged as a new network technology, and is expected to enable flexible configuration and management of network resources and offer efficient support to diverse user services [1]. By decoupling data plane and control plane, centralized network control and simplified flow forwarding can be achieved in SDN. More specifically, controllers in control plane are in charge of network control and management, e.g., determining forwarding strategies for user flows, whereas switches in data plane are merely executing flow forwarding according to the forwarding rules disseminated from the controllers [2].

To improve the efficiency of flow forwarding in SDN, switches may pre-cache certain number of forwarding rules in their ternary content addressable memory (TCAM). Upon receiving user flows, SDN switches search their TCAM for the corresponding forwarding rules. If the forwarding rules of user flows have been stored, the switches will forward user flows accordingly. However, it is probable that SDN switches fail to cache the forwarding rules of certain user flows due to their limited TCAM capacity, in this case, switches will send flow forwarding request messages, i.e., packet-in messages, to their associated controllers, and receive the forwarding rules specified in the packet-out messages sent from the controllers. The flow forwarding can then be executed accordingly. Apparently, the transmission performance of the packet-in and packet-out messages between switches and their associated controllers is of particular importance, since it may affect the quality of experience (QoE) of users significantly. In addition, as the packet-in messages are processed at controllers and the flow forwarding rules are determined by controllers, the processing capability of controllers may also have an impact on user QoE.

In a practical SDN scenario, controllers can be co-located with switches, and in particular, on account of the cost of controllers, the number of controllers is in general much smaller than that of switches, hence, we may associate one controller with multiple switches. Apparently, the number and location of controllers and the association relationship between controllers and switches may jointly affect the signaling transmission and process performance of the SDN network. Therefore, controller deployment problem should be investigated which jointly addresses the above controller placement related issues [3]. While traditional controller deployment studies assume that the deployed controllers are of ideal or desired characteristics, such as having enough capacity or processing capability [4], for practical SDN applications, it is probably that the SDN controllers to be deployed can only

* Corresponding author.
E-mail address: chairong@cqupt.edu.cn (R. Chai).

https://doi.org/10.1016/j.comnet.2021.108326
Received 8 April 2021; Received in revised form 10 June 2021; Accepted 9 July 2021
Available online 17 July 2021

be selected from a set of candidate controllers with certain features such as capacity or computation capability. In this circumstance, traditional controller deployment issue becomes a capacitated controller deployment problem [5,6].

Although capacitated controller deployment problem has been investigated in previous work, the authors aimed to achieve propagation delay optimization of switches or network cost minimization of controllers, and failed to consider extensively the signaling transmission delay between switches and controllers, the processing delay at controllers, and the cost of controllers. Furthermore, previous research work mainly focused on optimizing the overall network performance, and failed to stress the signaling transmission and processing performance of individual switches, hence, may lead to unfairness among switches and undesired control plane quality of service (QoS) of certain switches.

In this paper, we study capacitated controller deployment for SDN networks. Consider the signaling transmission and processing performance of switches and address the worst-case performance, we define network response time (NRT) as the maximum control plane response time (CPRT) of switches. Then, aiming to achieve the tradeoff between NRT and the cost of controllers, we introduce the concept of network cost which is defined as the weighted sum of NRT and controller cost. The capacitated controller deployment problem is formulated as a constrained optimization problem with the objective of minimizing network cost. It can be shown that the obtained optimization problem is an NP-hard problem and cannot be solved easily. In view of this, we propose a two-stage heuristic algorithm to obtain controller deployment strategy and controller-type matching strategy.

The major contributions of this paper are summarized as follows:

- While traditional controller deployment problem has been studied which assumes that the deployed controllers are of ideal or desired characteristics [7–13], the assumption may not hold for practical SDN applications. In this work, we consider a more practical assumption on controller deployment, i.e., the SDN controllers to be deployed can only be selected from a set of candidate controllers with certain features and different costs, and study capacitated controller deployment problem.
- Unlike the previous work that mainly focuses on optimizing the propagation delay between switches and controllers in SDN [5–8, 13–16], we examine the CPRT in a more comprehensive manner. In particular, we define CPRT of switches as the summation of the transmission and propagation delay required to transmit packet-in and packet-out messages between controllers and switches, as well as the processing delay at controllers. To stress the worst-case CRPT performance of switches, we define NRT as the longest CRPT of switches. By optimizing the worst case performance, the control plane transmission performance of individual switches can be enhanced.
- While the transmission performance of switches has been addressed in previous work, few work considered controller cost, e.g., the maintenance cost of controllers. In general, deploying a larger number of controllers may achieve better network transmission performance, however, higher cost for maintaining controllers will be resulted, which is highly undesired. In this paper, jointly consider the control plane transmission and process performance of switches, and the maintenance cost of controllers, we define network cost as the weighted sum of NRT and controller cost, and formulate the capacitated controller deployment problem of SDN as a constrained network cost optimization problem.
- The formulated minimization problem is an NP-hard problem which cannot be solved conveniently. In this paper, under some reasonable assumptions, the cost minimization problem is transformed into a min–max problem, and a two-stage heuristic algorithm is proposed. To be specific, in the first stage, an uncapacitated controller deployment subproblem is studied and a minimum eccentricity-based controller deployment algorithm is proposed to determine the number and location of controllers and the association relationship between controllers and switches. In the second stage, we address controller-type matching subproblem and propose a greedy method-based controller-type matching algorithm to determine the types of deployed controllers.
- Numerical simulations are conducted to verify the performance of the proposed two-stage algorithm. The results demonstrate that the proposed algorithm offers decreased network cost and achieve a tradeoff between NRT and controller cost.

The remainder of this paper is organized as follows. Section 2 states a summary of related work on controller deployment problem. Section 3 describes the system model considered in this paper. In Section 4, we formulate the capacitated controller deployment problem as a network cost minimization problem. The solution approach to the formulated optimization problem is presented in Section 5. Simulation results are described in Section 6. Finally, we conclude the paper in Section 7.

2. Related work

In this section, we present an overview of related work on controller deployment problem in SDN.

2.1. Traditional controller deployment algorithms

To design traditional controller deployment strategy, it is usually assumed that the characteristics of controllers are ideal or the capacity of controllers is large enough so that the processing delay of packet-in messages can be negligible.

The controller deployment problem in software-defined wide area networks was studied in [7]. In order to achieve communication delay minimization and to ensure the fairness of load distribution among controllers, a modified density peak clustering algorithm was proposed, which determines the required number and location of SDN controllers. In [8], the original SDN was divided into multiple sub-networks through applying a modified density peak clustering scheme. By adjusting the average degree and near centrality of the sub-networks, the controllers were located in the center of the sub-networks, so as to reduce the average propagation delay between switches and controllers. The authors in [9] studied the controller deployment problem and proposed a clustering-based network partitioning algorithm to minimize the overall delay. Ref. [10] formulated a dynamic controller assignment problem in SDN. In order to minimize the delay and control overhead, the authors proposed an online stable matching-based algorithm to assign the flow-requests of switches to controllers dynamically.

Ref. [11] takes into account the deployment cost of controllers and proposes a cost-optimized multi-controller deployment strategy. In [12], controller deployment problem was formulated as an optimization problem with the objective of maximizing the number of flows while minimizing the propagation delay of flow requests between controllers and switches under a given budget constraint. A global optimization solver was employed to solve the problem for small networks, and a heuristic algorithm was proposed to design the controller deployment strategy for large networks. In [13], SDN controller deployment problem was studied and a green centralized controller association algorithm was proposed to minimize the overall energy consumption of the control plane.

2.2. Capacitated controller deployment algorithms

The capacitated controller deployment problem of SDN was studied in [14–21]. In [14,17,18], it is assumed that the deployed controllers share the same capacity characteristics. The controller deployment strategy was proposed in [14], which minimizes the energy consumption of the network under the constraints on controller load and control path delay. In [17], the authors defined the concept of mutual-satisfaction degree, which is jointly determined by propagation delay, network failure probability and controller load, and then presented a controller deployment strategy to maximize the mutual-satisfaction degree. In [18], the authors proposed a controller deployment strategy so as to minimize the worst-case delay between controllers and switches and maximize the utilization of the controller.

Jointly consider the propagation delay of switches and the capacity of controllers, the authors in [15] formulated the capacitated controller deployment problem as a communication delay minimization problem, and designed a two-phase dynamic controller deployment strategy. Specifically, during the initial phase, a modified nearest neighbor propagation algorithm was proposed to partition the network into multiple sub-domains; then, during the second phase, the switches in different sub-domains are redistributed based on a breadth-first search algorithm to balance the load among controllers. In [16], a density-based controller deployment strategy was proposed based on clustering algorithm in order to optimize the transmission delay of switches and enhance network reliability. Ref. [19] examined the control plane delay of SDN, and proposed a heuristic algorithm which applies Dijkstra algorithm and K-means algorithm to minimize the average control plane delay.

Assuming that controller failures might occur in SDN, the authors in [20] addressed the problem of capacitated controller deployment. To enable reliable information interaction between switches and controllers, the authors proposed a controller selection scheme which assigns two controllers for individual switches, i.e., one master associated controller and one backup controller. The deployment strategy for both controllers was designed to minimize the overall signaling propagation delay. In [21], the authors studied the controller deployment problem in SDN-enabled edge networks, and formulated a mixed integer programming problem to minimize the weighted sum of the transmission delay of wireless nodes and node association cost. For small-scale edge networks, the formulated problem is solved directly by using software solver CPLEX. For large-scale edge networks, an approximate solution was proposed to obtain the sub-optimal controller deployment strategy.

The cost incurred due to controller deployment and maintenance was considered in [22–24]. In [22], the capacitated controller deployment problem in SDN was investigated and formulated as a network cost minimization problem. The optimization problem was transformed into two subproblems, i.e., controller deployment subproblem and controller-type matching subproblem, which are then solved, respectively. Consider the dynamic characteristics of user flows in SDN, the authors in [23] studied joint controller deployment and flow management problem and presented a joint strategy to optimize the maintenance cost of controllers. In [24], a load-aware controller placement strategy was proposed to minimize the cost for deploying and maintaining controllers.

In this paper, we extend our previous work in [22] and propose a more general and practical algorithm to determine controller deployment and controller-type matching strategy. The capacitated controller deployment problem is formulated as a constrained optimization problem which minimizes network cost. Under some reasonable assumptions, the optimization problem is transformed into a min–max problem, and a two-stage heuristic algorithm is proposed to solve the optimization problem. In the first stage, a minimum eccentricity-based controller deployment algorithm is proposed to determine the number and location of controllers and the association relationship between controllers and switches. Based on the results obtained during the first phase, we then design a greedy method-based controller-type matching strategy to determine the controller-type matching strategy in the second stage.

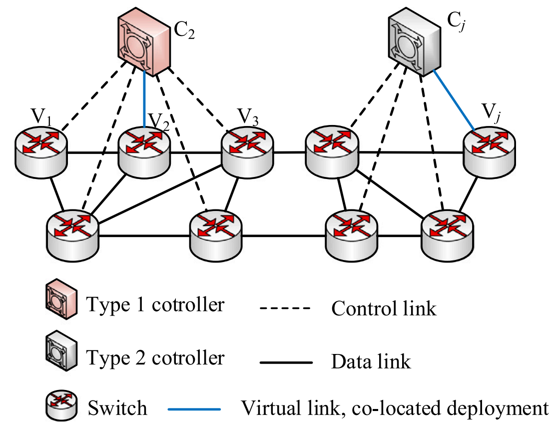

Fig. 1. System model.

3. System model

In this paper, we consider an SDN network consisting of N switches denoted by $\mathrm{V}=\{\mathrm{V}_1,\mathrm{V}_2,\ldots,\mathrm{V}_N\}$, where V_i represents the ith switch, $1 \le i \le N$. A binary variable $\beta_{i,j}$ is defined to characterize the connection status between V_i and V_j. If V_i and V_j are adjacent nodes, i.e., the direct link between V_i and V_j exists, we set $\beta_{i,j}=1$, otherwise, $\beta_{i,j}=0$. Let $\mathrm{E}_{i,j}$ denote the link connecting adjacent nodes V_i and V_j. We further denote the transmission rate of $\mathrm{E}_{i,j}$ as $B_{i,j}$ and the length of $\mathrm{E}_{i,j}$ as $d_{i,j}$, $1 \le i \ne j \le N$. Suppose individual switches may have various user flow transmission requirements and may need to transmit packet-in messages to their associated controllers in the case that the forwarding rules of certain incoming flows have not been pre-cached locally. Let λ_j denote the average arrival rate of the packet-in messages of V_j, $1 \le j \le N$.

We investigate the controller deployment problem of the considered SDN network. It is assumed that there are K types of candidate controllers which can be deployed in the SDN scenario. Various types of candidate controllers may have different processing capability which is characterized by controller capacity and different costs resulted from purchasing and maintaining controllers. We denote the capacity and cost of type k controllers respectively as ϕ_k and η_k, and the number of type k controllers as N_k, $1 \le k \le K$. Without loss of generality, we assume that $\phi_1 \le \phi_2 \le \ldots \le \phi_K$, and $\eta_1 \le \eta_2 \le \ldots \le \eta_K$.

It is assumed that controllers can only be co-located with switches and one switch is allowed to deploy at most one controller. For convenience, we refer the controller deployed at V_j as C_j, $1 \le j \le N$. In this paper, the capacitated controller deployment problem of the SDN network is addressed. To be more specific, we jointly design the deployment strategy of candidate controllers and the controller-type matching strategy of the deployed controllers. Fig. 1 shows the considered system model. In Table 1, we list some key notations used in this paper.

4. Capacitated controller deployment problem formulation

In this section, we jointly consider the control plane transmission and processing performance of switches as well as the cost of controllers, and define network cost function as the weighted sum of control plane performance and controller cost. Taking into account the constraints on controller deployment and capacity matching, we formulate a network cost minimization problem.

4.1. Network cost function formulation

In this subsection, we first define the NRT of switches and the cost of controllers and then formulate network cost function based on the two metrics.

Table 1
Summary of key notations.

Notation	Description
$T^{\mathrm{t}}_{i,j}$	Transmission delay for sending packet-in messages from V_j to C_i
$T^{\mathrm{p}}_{i,j}$	Propagation delay for sending packet-in messages from V_j to C_i
$T^{\mathrm{w}}_{i,j}$	Time required to process packet-in messages of V_j at C_i
T_j	CPRT of V_j
$\mathrm{E}_{i,j}$	Link connecting V_i and V_j
$B_{i,j}$	Transmission rate of $\mathrm{E}_{i,j}$
$d_{i,j}$	Length of $\mathrm{E}_{i,j}$
λ_j	Average arrival rate of packet-in messages at V_j.
N_k	Number of type k controllers
ϕ_k	Capacity of type k controllers
η_k	Cost of type k controllers
η	Controller cost
Ψ	Network cost function
x_i	Controller deployment variable
$y_{i,j}$	Association variable between C_i and V_j
$z_{i,k}$	Controller-type matching variable between C_i and type k

4.1.1. Network response time formulation

To characterize the control plane transmission and processing performance of switches, we define the CPRT of switches as the total time required for transmitting packet-in and packet-out messages, and that for processing packet-in messages, i.e., determining flow forwarding strategy at controllers.

Let T_j denote the CPRT of switch V_j, $1 \le j \le N$. T_j is defined as the summation of the round trip transmission and propagation delay between V_j and its associated controller, and the processing delay at the controller, i.e.,

$$T_j = \sum_{i=1}^{N} x_i y_{i,j} (T^{\mathrm{t}}_{i,j} + T^{\mathrm{p}}_{i,j} + T^{\mathrm{w}}_{i,j}) \tag{1}$$

where $x_i \in \{0,1\}$ denotes the controller deployment variable, i.e., $x_i = 1$ indicates that controller C_i is deployed at V_i, otherwise, $x_i = 0$; $y_{i,j} \in \{0,1\}$ denotes the association variable between C_i and V_j, i.e., if V_j is associated with C_i, we set $y_{i,j} = 1$, otherwise, $y_{i,j} = 0$.

$T^{\mathrm{t}}_{i,j}$ in (1) represents the total transmission delay when transmitting packet-in messages from V_j to C_i, and transmitting packet-out messages from C_i to V_j. Let $\mathrm{P}_{i,j}$ denote the end-to-end path between C_i and V_j. Since C_i is co-located with V_i, if V_j and V_i are adjacent nodes, i.e., $\beta_{i,j} = 1$, $\mathrm{P}_{i,j}$ is reduced to $\mathrm{E}_{i,j}$. If $\beta_{i,j} = 0$, that is, the direct link between V_i and V_j does not exist, $\mathrm{P}_{i,j}$ becomes a multi-hop path between V_i and V_j. Under this circumstance, $T^{\mathrm{t}}_{i,j}$ can be calculated as the sum of the transmission delay of the links along $\mathrm{P}_{i,j}$. Jointly consider both cases, we express $T^{\mathrm{t}}_{i,j}$ as

$$T^{\mathrm{t}}_{i,j} = \begin{cases} \dfrac{2\alpha\lambda_j t_0}{B_{i,j}}, & \text{if } \beta_{i,j} = 1 \\ \displaystyle\sum_{h=1}^{H_{i,j}} \frac{2\alpha\lambda_j t_0}{B^h_{i,j}}, & \text{if } \beta_{i,j} = 0 \end{cases} \tag{2}$$

where α is the average size of packet-in messages and packet-out messages, t_0 is a fixed time duration, thus, the term $\lambda_j t_0$ represents the number of packet-in messages sent from V_j during time period t_0, $B_{i,j}$ denotes the transmission rate of $\mathrm{E}_{i,j}$, $B^h_{i,j}$ represents the transmission rate of the hth hop link along $\mathrm{P}_{i,j}$ and $H_{i,j}$ represents the total number of hops along $\mathrm{P}_{i,j}$.

$T^{\mathrm{p}}_{i,j}$ in (1) is the total propagation delay for sending packet-in messages from V_j to C_i and sending packet-out messages from C_i to V_j. $T^{\mathrm{p}}_{i,j}$ can be computed as

$$T^{\mathrm{p}}_{i,j} = \begin{cases} \dfrac{2d_{i,j}\lambda_j t_0}{v}, & \text{if } \beta_{i,j} = 1 \\ \displaystyle\sum_{h=1}^{H_{i,j}} \frac{2d^h_{i,j}\lambda_j t_0}{v}, & \text{if } \beta_{i,j} = 0 \end{cases} \tag{3}$$

where $d^h_{i,j}$ represents the length of the hth hop link of $\mathrm{P}_{i,j}$, and v is the propagation speed of electromagnetic wave. In general, $v = \frac{2}{3}c$, where c is the speed of light.

$T^{\mathrm{w}}_{i,j}$ in (1) is the time required to process the packet-in messages of V_j at C_i. We assume that packet-in messages are processed at the controllers based on the first come first served principle, therefore, $T^{\mathrm{w}}_{i,j}$ can be expressed as

$$T^{\mathrm{w}}_{i,j} = \frac{\lambda_j t_0}{\mu_i - l_i} \tag{4}$$

where μ_i and l_i represent respectively the capacity and traffic load of C_i. Since the capacity of C_i is determined by its capacity matching strategy, we express μ_i as

$$\mu_i = \sum_{k=1}^{K} z_{i,k}\phi_k \tag{5}$$

where $z_{i,k} \in \{0,1\}$ denotes the controller-type matching variable, that is, $z_{i,k} = 1$ indicates that C_i is a type k controller, otherwise, $z_{i,k} = 0$. As C_i may associate with various switches, and be in charge of the flow control and management of their switches, the load of C_i can be computed as $l_i = \sum_{j=1}^{N} y_{i,j}\lambda_j$.

Consider the CPRT of all the switches in the network, and stress the worst-case performance, we formulate the NRT of the SDN as the worst-case CPRT of switches. Let T denote the NRT, we obtain

$$T = \max_j \{T_j\}. \tag{6}$$

4.1.2. Controller cost formulation

It can be demonstrated that by deploying controllers efficiently, the CPRT of switches and the corresponding NRT may decrease which is mainly benefited from the reduced distance between switches and their associated controllers. However, deploying controllers in SDN may incur certain purchase, installation and maintenance cost. In general, the cost increases accordingly as the number of controllers increases [24]. For a practical SDN controller deployment scenario, the cost of controllers is an important factor worthy of consideration.

In this paper, controller cost is formulated as the overall cost resulted from purchasing, installing and maintaining various types of controllers. In particular, since deployed controllers might be of different cost, the controller cost is defined as a function of controller cost corresponding to various types and the number of controllers of certain types, i.e.,

$$\eta = \sum_{i=1}^{N}\sum_{k=1}^{K} x_i z_{i,k}\eta_k. \tag{7}$$

It can be observed from the above equation that controller cost is jointly determined by the controller deployment and controller-type matching strategy as well as the cost of various types of controllers. In general, as the number of controllers increases, the controller cost becomes higher. Consider an extreme case that all the controllers are of the same cost, the controller cost is reduced to the product of the number of controllers and the cost of individual controllers. Note that the communication overhead between controllers can also be considered as part of controller cost, which is not investigated in this paper for simplicity, and we may extend our current cost formulation by involving exact communication overhead between controllers in our future work. However, it should be mentioned that the communication overhead between controllers highly depends on the number of controllers, therefore, the controller cost defined in this paper can also be considered as being positively related to the communication overhead between controllers.

Let Ψ denote network cost function, which is defined as the weighted sum of the NRT and controller cost, i.e.,

$$\Psi = T + \omega\eta \tag{8}$$

where ω represents the weight of controller cost. It is noted that the trade-off between NRT and controller cost can be achieved by varying the value of weight ω. In particular, we may choose relatively large ω, when we tend to stress the cost of controllers compared to NRT. On the other hand, we may choose relatively small ω, if the performance of switches is mainly considered.

4.2. Optimization constraints

The capacitated controller deployment is subject to a number of constraints.

4.2.1. Controller deployment constraint

In order to guarantee that there are sufficient candidate controllers being deployed, the total number of candidate controllers must be greater or equal to the number of controllers deployed in the network, that is,

$$\mathrm{C1}:\ \sum_{i=1}^{N} x_i \le \sum_{k=1}^{K} N_k. \tag{9}$$

4.2.2. Controller-switch association constraints

To achieve efficient flow forwarding in the SDN, all the switches should attach at least one controller. Without loss of generality, in this paper, it is assumed that one switch can only access one controller for packet forwarding management, i.e.,

$$\mathrm{C2}:\ \sum_{i=1}^{N} y_{i,j} = 1, \quad 1 \le j \le N. \tag{10}$$

It is obvious that the association between switch V_j and controller C_i is effective only when C_i is deployed, therefore, we describe the constraint on switch association and controller deployment as follows:

$$\mathrm{C3}:\ y_{i,j} \le x_i, \quad 1 \le i, j \le N. \tag{11}$$

4.2.3. Controller capacity constraint

To ensure effective message processing at controllers, the capacity of one controller should be greater than its load, i.e.,

$$\mathrm{C4}:\ \sum_{j=1}^{N} x_i y_{i,j} \lambda_j \le \mu_i, \quad 1 \le i \le N. \tag{12}$$

4.2.4. Controller-type matching constraints

For certain type of controllers, the number of controllers being deployed cannot exceed their maximum number, i.e.,

$$\mathrm{C5}:\ \sum_{i=1}^{N} z_{i,k} \le N_k, \quad 1 \le k \le K. \tag{13}$$

If a controller is deployed, one specific controller type should be assigned, otherwise, no controller type is selected. Therefore, we obtain the following constraint:

$$\mathrm{C6}:\ \sum_{k=1}^{K} z_{i,k} = x_i, \quad 1 \le i \le N. \tag{14}$$

4.3. Optimization problem formulation

In an attempt to minimize network cost function, while taking into account the optimization constraints, we formulate the capacitated controller deployment problem as a constrained network cost minimization problem:

$$\begin{aligned} \min_{x_i, y_{i,j}, z_{i,k}} \quad & \Psi \\ \text{s.t.} \quad & \mathrm{C1}-\mathrm{C6}. \end{aligned} \tag{15}$$

By solving the above formulated minimization problem, the capacitated controller deployment strategy can be obtained and the tradeoff between NRT and controller cost can be achieved.

5. Controller deployment strategy

The network cost minimization problem formulated in (15) involves the coupling of controller deployment, controller and switch association and controller-type matching strategies, and the problem is defined using a set of nonlinear constraints, hence, is difficult to solve. As a matter of fact, we can observe that the formulated optimization problem is indeed an extended controller deployment problem, which can be regarded as a center location problem and has been demonstrated to be an NP-hard problem [25]. Therefore, the network cost minimization problem formulated in (15) is also an NP hard problem which cannot be solved conveniently. In this section, under some reasonable assumptions, the network cost minimization problem is transformed into a min–max problem, and a two-stage heuristic algorithm is proposed. In the first stage, a minimum eccentricity-based controller deployment algorithm is proposed for the formulated uncapacitated controller deployment subproblem. Based on obtained controller deployment strategy, we design a greedy method-based controller-type matching algorithm to determine the types of deployed controllers in the second stage.

5.1. Controller deployment subproblem formulation

To solve the minimization problem formulated in (15), we first analyze the objective function Ψ, which can be rewritten as

$$\Psi = \max_{j}\{T_j\} + \omega\eta. \tag{16}$$

According to the definition of T_j in (1), we can observe that if the following delay variation condition holds:

$$\begin{aligned} &\max_{j_1,j_2}\{\sum_{i=1}^{N} x_i y_{i,j_1} T^{\mathrm{w}}_{i,j_1} - \sum_{i=1}^{N} x_i y_{i,j_2} T^{\mathrm{w}}_{i,j_2}\} \le \\ &\quad \max_{j_1,j_2}\{\sum_{i=1}^{N} x_i y_{i,j_1}(T^{\mathrm{t}}_{i,j_1} + T^{\mathrm{p}}_{i,j_1}) - \sum_{i=1}^{N} x_i y_{i,j_2}(T^{\mathrm{t}}_{i,j_2} + T^{\mathrm{p}}_{i,j_2})\}, \end{aligned} \tag{17}$$

the worst-case CPRT of the switches will be merely determined by the summation of the transmission and propagation delay, and the impact of the signaling processing delay on the CPRT can be ignored. Let V_{j^*} denote the switch which is of the longest CPRT among all the switches, we obtain

$$V_{j^*} = \operatorname{argmax} T_j = \operatorname{argmax} \sum_{i=1}^{N} x_i y_{i,j}(T^{\mathrm{t}}_{i,j} + T^{\mathrm{p}}_{i,j}). \tag{18}$$

That is, by removing the term $T^{\mathrm{w}}_{i,j}$ in T_j, the switch offering the longest CPRT will not change, and accordingly, the designed controller deployment and type matching strategy is still effective. For convenience, we denote $T^{\mathrm{s}}_{i,j}$ as the summation of the transmission and propagation delay required to send the packet-in and packet-out messages between C_i and V_j, i.e.,

$$T^{\mathrm{s}}_{i,j} = T^{\mathrm{t}}_{i,j} + T^{\mathrm{p}}_{i,j}. \tag{19}$$

We now omit the processing delay at the controllers and recalculate the CPRT of V_j, denoted by $\bar{T}_j$, i.e.,

$$\bar{T}_j = \sum_{i=1}^{N} x_i y_{i,j} T^{s}_{i,j} = \sum_{i=1}^{N} x_i y_{i,j} (T^{t}_{i,j} + T^{p}_{i,j}). \tag{20}$$

Accordingly, the network cost minimization problem formulated in (15) can be equivalently transformed into the following problem:

$$\begin{aligned} \min_{x_i, y_{i,j}, z_{i,k}} \quad & \max_j \{\bar{T}_j\} + \omega\eta \\ \text{s.t.} \quad & \text{C1} - \text{C6}. \end{aligned} \tag{21}$$

It is apparent that deploying different types of controllers may cause different cost η and different network cost in turn. To further simplify the problem, we relax the constraints on controller deployment and suppose that all the deployed controllers are of the same type. For convenience, let k^* denote the type of deployed controllers, we set $z_{i,k^*} = 1$, if $x_i = 1$. The controller cost can then be computed as $\eta = \sum_{i=1}^{N} x_i z_{i,k^*} \eta_{k^*} = N\eta_{k^*}$, which is now a constant. As a result, the optimization problem (21) is reduced to the following controller deployment subproblem:

$$\begin{aligned} \min_{x_i, y_{i,j}} \quad & \max_j \{\bar{T}_j\} \\ \text{s.t.} \quad & \text{C1} - \text{C4}. \end{aligned} \tag{22}$$

5.2. Proposed minimum eccentricity-based controller deployment algorithm

In this subsection, a minimum eccentricity-based controller deployment algorithm is proposed for solving the problem in (22). More specifically, the SDN network is characterized by a complete graph wherein the CPRT between any two switches is calculated by applying the Dijkstra algorithm (D algorithm), then the nodes offering the minimum eccentricity are selected to deploy controllers, and the association strategy between switches and controllers is designed accordingly.

The proposed algorithm is discussed in detail in the following subsections.

5.2.1. Determining CPRT between two switches

We first examine the CPRT between any two switches, i.e., for switches V_i and V_j, we compute $T^{s}_{i,j}$ according to (2) and (3). For the simple scenario which V_i and V_j are adjacent nodes, i.e., $\beta_{i,j} = 1$, $T^{s}_{i,j}$ can be calculated directly. For the slightly complicated case where V_i and V_j are not adjacent nodes, i.e., $\beta_{i,j} = 0$, a multi-hop path is required to connect V_i and V_j. In general, the possible paths connecting V_i and V_j may not be unique, we may apply routing algorithms, for instance, the D algorithm, to determine the optimum path and the optimal CPRT accordingly.

To determine the optimal CPRT between switches by exploiting the D algorithm, we regard the switches in the SDN as network nodes and map the considered SDN network into a weighted graph $G = (V, E, W)$, where $E = \{E_{i,j}\}$ denotes the set of the links in the network, $W = \{W_{i,j}\}$ is the set of weights, $W_{i,j}$ is the weight of $E_{i,j}$, which is expressed as $W_{i,j} = T^{s}_{i,j}$.

Applying the D algorithm on G, we obtain the shortest path between V_i and V_j. Let $P^*_{i,j}$ denote the optimal path connecting V_i and V_j which offers the minimum CPRT. Let $B^{h,*}_{i,j}$ and $d^{h,*}_{i,j}$ represent respectively the transmission rate and the length of the hth hop of $P^*_{i,j}$, $H^*_{i,j}$ denote the total hops of $P^*_{i,j}$, and $T^{s,*}_{i,j}$ denote the response delay of $P^*_{i,j}$. $T^{s,*}_{i,j}$ can be calculated as

$$T^{s,*}_{i,j} = \begin{cases} 2\lambda_j t_0 \left(\dfrac{\alpha}{B_{i,j}} + \dfrac{d_{i,j}}{v} \right), & \text{if } \beta_{i,j} = 1, \\ 2\lambda_j t_0 \left(\displaystyle\sum_{h=1}^{H^*_{i,j}} \dfrac{\alpha}{B^{h,*}_{i,j}} + \sum_{h=1}^{H^*_{i,j}} \dfrac{d^{h,*}_{i,j}}{v} \right), & \text{if } \beta_{i,j} = 0. \end{cases} \tag{23}$$

5.2.2. Determining globally optimal controller deployment strategy

Apparently, the performance of controller deployment is closely related to the number of controllers being deployed. Due to the close coupling of various issues, it is highly difficult to jointly design the optimal number of controllers, the deployment strategy of individual controllers, and the association relationship between switches and controllers. However, consider a practical application scenario, where the number of controllers might be highly limited, we may first assume that the number of controllers are given and design the corresponding locally optimal controller deployment strategy. Then, the globally optimal controller deployment strategy can be obtained by comparing the obtained controller deployment strategy and choosing the one offering the minimum NRT.

Let m be the number of deployed controllers, where $1 \le m \le \min\{N, \sum_{k=1}^{K} N_k\}$, and denote $\Psi^{(m)}$ as the locally optimal network cost when the number of controllers is m, the globally optimal controller deployment strategy is given by

$$\{x^*_i,\ y^*_{i,j},\ m^*\} = \arg\min\{\Psi^{(m)}\}. \tag{24}$$

In the following subsection, we set the number of controllers as m, determine the locally optimal controller deployment strategy and obtain the corresponding $\Psi^{(m)}$.

5.2.3. Locally optimal controller deployment strategy

Assuming that the number of controllers is m, we propose a minimum eccentricity-based algorithm to determine the location of deployed controllers and the association strategy between controllers and switches.

The basic idea of the proposed algorithm can be summarized briefly as follows. We first calculate the eccentricity of individual switches in graph G, select the switch offering the minimum eccentricity to deploy a controller, then delete the selected switch in G. Repeat the above process until m controllers are deployed. Based on the initial controller deployment strategy, we associate switches with the controllers offering the smallest CPRT as their associated controllers. Define the subnetwork consisting of one controller and its associated switches as a subgraph, we update the controller deployment strategy in each subgraph by selecting the switch with the minimum eccentricity as the updated location of the controller. Given the updated controllers, switch reassociation process is executed. The process repeats until the set of controllers and the association strategy between controllers and switches no longer change.

For a given controller number m, the proposed minimum eccentricity-based algorithm can be summarized as below:

(a) Initialization: Given a weighted fully connected graph $\bar{G} = (V, \bar{E}, \bar{W})$, where $\bar{E} = \{\bar{E}_{i,j}\}$ is the set of links, $\bar{E}_{i,j}$ is the link connecting V_i and V_j. If $\beta_{i,j} = 1$, $\bar{E}_{i,j} = E_{i,j}$, otherwise, $\bar{E}_{i,j}$ is the logical link connecting V_i and V_j, which represents $P^*_{i,j}$ in physical network, $\bar{W} = \{\bar{w}_{i,j}\}$ denotes the set of weights and $\{\bar{W}_{i,j}\}$ denotes the weight of $\bar{E}_{i,j}$ and is defined as $\bar{W}_{i,j} = T^{s,*}_{i,j}$.

(b) Initial controller deployment:

b1) Set $m_0 = 1$ and the set of initial controllers $\Phi_c = \emptyset$;

b2) For $V_i \in V, 1 \le i \le N$, calculate the eccentricity, denoted by Θ_i, i.e.,

$$\Theta_i = \max_{V_j \in V,\ i \ne j} \{T^{s,*}_{i,j}\}.$$

b3) Select the switch which offers the minimum eccentricity to deploy a controller. Specifically, if

$$C_{i^*} = \arg\min_{V_i \in V} \{\Theta_i\},$$

we deploy a controller at V_i, and set $x^*_i = 1$.

b4) Update the set of controllers:

$$\Phi'_c = \Phi_c \cup \{C_{i^*}\}.$$

b5) If $m_0 < m$, remove V_{i^*} from V, i.e., $\mathrm{V} = \mathrm{V}/\{\mathrm{V}_{i^*}\}$, set $m_0 = m_0 + 1$, $\Phi'_{\mathrm{c}} = \Phi_{\mathrm{c}}$ and return to Step b3); else, go to Step c).

(c) Controller-switch association: For $\mathrm{V}_j \in \mathrm{V}/\Phi_{\mathrm{c}}$, select the controller offering the minimum CPRT as its associated controller, i.e., if $\mathrm{V}_{i^*} = \arg\min\{T^{\mathrm{s},*}_{i,j}\}$, and $\mathrm{C}_{i^*} \in \Phi_{\mathrm{c}}$, then, we set $y^*_{i^*,j} = 1$.

(d) Controller update: For $\mathrm{C}_i \in \Phi_{\mathrm{c}}$, create a subgraph G_i which is composed of C_i and the switches associated with C_i, i.e., $\mathrm{G}_i = (\mathrm{V}_0, \mathrm{E}_0)$, where $\mathrm{V}_0 = \{\mathrm{C}_i\} \cup \{\mathrm{V}_j | y^*_{i,j} = 1\}$ and $\mathrm{E}_0 = \{\bar{\mathrm{E}}_{i,j} | \mathrm{V}_i, \mathrm{V}_j \in \mathrm{V}_0\}$. Calculate the eccentricity of nodes in G_i, and choose the one offering the minimum eccentricity as the new location of controller, i.e.,

$$\mathrm{C}_{i^*} = \arg\min_{\mathrm{V}_i \in \mathrm{V}_0}\{\Theta_i\}.$$

Accordingly, update the controller set as $\Phi'_{\mathrm{c}} = \Phi_{\mathrm{c}} \cup \{\mathrm{C}_{i^*}\}/\{\mathrm{C}_i\}$.

(e) Check algorithm termination: If $\Phi'_{\mathrm{c}} = \Phi_{\mathrm{c}}$, and the obtained association strategy $y^*_{i,j}$ no longer changes, algorithm terminates, else, return to Step c).

5.3. Controller-type matching subproblem formulation

Executing the proposed minimum eccentricity-based controller deployment algorithm, we obtain the controller deployment and controller-switch association strategy, denoted by x^*_i and $y^*_{i,j}$, respectively, based on which, we devise the controller-type matching strategy.

Substituting x_i and $y_{i,j}$ in (20) by x^*_i and $y^*_{i,j}$, the optimal CPRT of V_j, denoted by $T^{(1,*)}_j$, can be expressed as

$$T^{(1,*)}_j = \sum_{i=1}^{N} x^*_i y^*_{i,j} T^{\mathrm{s},*}_{i,j}. \tag{25}$$

Taking into account the processing time of packet-in messages at controllers, we rewrite network cost as

$$\Psi' = \max_j\{T^{(1,*)}_j + \sum_{i=1}^{N} x^*_i y^*_{i,j} T^{\mathrm{w}}_{i,j}\} + \omega\eta. \tag{26}$$

Hence, controller-type matching subproblem can be formulated as

$$\begin{aligned} \min_{z_{i,k}} \quad & \Psi' \\ \text{s.t.} \quad & \text{C5, C6.} \end{aligned} \tag{27}$$

5.4. Greedy method-based controller-type matching strategy

To solve the network cost minimization problem formulated in (27), we present a greedy method-based controller-type matching algorithm. In particular, under the assumption that all the controllers are of identical type, the network cost corresponding to various controller types can be obtained. Let k^* denote the controller-type that leads to the minimum network cost, we first assign type k^* to all the deployed controllers, then examine the constraints on controller-type matching, i.e., the candidate controller number constraint specified in (13) and the delay variation constraint given in (17). If both constraints hold, the algorithm terminates, otherwise, we design controller-type matching strategy based on our proposed greedy method-based algorithm. The detail algorithm is discussed in the following subsections.

5.4.1. Initial controller-type matching strategy

While there are various types of candidate controllers being available, we start from a simple case, that is, we assume that the deployed controllers are of identical type, say, type k. Let Ψ'_k denote the corresponding network cost when type k controllers are deployed, and let m' denote the locally optimal number of controllers, we obtain

$$\Psi'_k = \max_j\{T^{(1,*)}_j + \sum_{i=1}^{N} x^*_i y^*_{i,j} T^{(2,\mathrm{w})}_{i,j}\} + \omega m' \eta_k \tag{28}$$

where $T^{(2,\mathrm{w})}_{i,j}$ is given by

$$T^{(2,\mathrm{w})}_{i,j} = \frac{\lambda_j t_0}{z_{i,k}\phi_k - \sum_{j=1}^{N} y^*_{i,j}\lambda_j}. \tag{29}$$

To obtain the controller type which offers the minimum network cost, we rank Ψ'_k in an ascending order. Suppose

$$\Psi'_{k_1} \le \ldots \le \Psi'_{k_i} \le \ldots \le \Psi'_{k_K}, \tag{30}$$

it can be understood in a straightforward manner that by deploying the k_1th type controllers, the resulted network cost will be minimized. For simplicity, we ignore the maximum number constraint of candidate controllers for the time being, and assign the k_1th type candidate controllers to all the deployed controllers. Let $\bar{z}_{i,k}$ denote local controller-type matching strategy, we set $\bar{z}_{i,k_1} = 1$, $\forall x^*_i = 1$, $1 \le i \le N$.

Since both the candidate controller number constraint and the delay variation constraint should be satisfied when designing controller deployment and type matching strategy, we need to further check the obtained controller-type matching strategy $\bar{z}_{i,k_1}$. For convenience, we refer the constraint on the maximum number of candidate controllers given in (13) as Constraint 1, and the delay variation constraint described in (17) as Constraint 2. If both constraints hold, the algorithm terminates. Let $z^*_{i,k}$ denote optimal controller-type matching strategy, we set $z^*_{i,k_1} = \bar{z}_{i,k_1}$, $\forall x^*_i = 1$, $1 \le i \le N$. In the case that any one of the constraints does not meet, we should modify the obtained controller-type matching strategy in order to satisfy the two constraints.

5.4.2. Constraint 1—oriented algorithm modification

Given the obtained controller-type matching strategy $\bar{z}_{i,k_1}$, we now examine Constraint 1. Let N' represent the number of deployed controllers, i.e., $\sum_{i=1}^{N} x^*_i = N'$. In the case that $N' \le N_{k_1}$, the constraint on the maximum number of candidate controllers holds, meaning that the type k_1 candidate controllers are sufficient for deploying all the required controllers, no controller modification is needed for ensuring Constraint 1.

In the case that $N' > N_{k_1}$, the number of deployed controllers is greater than that of type k_1 candidate controllers, hence, it is impossible to assign type k_1 to all the deployed controllers. On account of the performance optimization of type k_1 controllers, we should assign N_{k_1} controllers as type k_1, and assign a new type (referred to as destination type) to the remaining controllers, i.e., we should assign $N' - N_{k_1}$ controllers to other types. The emerging problem is that among all the deployed controllers, which should be assigned new types other than k_1 and how to determine the destination controller types.

According to the inequality given in (30), switching the type of controllers from k_1 to other types may result in the increase in network cost. In order to achieve network cost minimization, we should choose the feasible controller types which offer the minimum network cost. In particular, we may compare the difference between the network costs when assigning type k_1 and other types to the deployed controllers, and select the types resulting in the minimum increase in network cost.

Assume $\bar{z}_{i,k_1} = 1$, let $\delta_{i,k_1,k}$ denote the cost difference when shifting controller C_i from type k_1 to type k, we obtain

$$\delta_{i,k_1,k} = T^{\mathrm{w}}_{i,k} - T^{\mathrm{w}}_{i,k_1} + \omega(\eta_k - \eta_{k_1}). \tag{31}$$

Given type k_1, for $\forall\ k \ne k_1$, various $\delta_{i,k_1,k}$ is resulted. It is obvious that we should shift controller C_i from type k_1 to type k which offers smallest $\delta_{i,k_1,k}$. Furthermore, as shifting various controllers may lead to different $\delta_{i,k_1,k}$, we should jointly select the controllers and their destination types so that the minimum $\delta_{i,k_1,k}$ can be achieved, i.e., controller $\mathrm{C}_{i'}$ should be shifted from k_1 to k', provided that

$$\{i', k'\} = \arg\min_{i,k}\{\delta_{i,k_1,k} | z_{i,k_1} = 1\}. \tag{32}$$

Since we may need to shift the type of multiple controllers, the above process should be repeated until Constraint 1 meets. The algorithm is summarized in Algorithm 1.

Algorithm 1 Proposed modified controller-type matching algorithm: meeting Constraint 1

1: Initialization: given N_k, set $\bar{z}_{i,k_1} = 1$, $x_i^* = 1$, $N' = \sum_{i=1}^{N} x_i^*$, $\bar{N}_k = 0, \forall k \neq k_1$, $\boldsymbol{\Phi}_1 = \emptyset$
2: Check whether Constraint 1 on type k_1 holds
if $N' > N_{k_1}$
3: compute the minimum network cost difference $\delta_{i,k_1,k}$
4: determine the controller and type should be adjusted: $\{i', k'\} = \arg\min_{i,k \notin \boldsymbol{\Phi}_1} \{\delta_{i,k_1,k}\}, \forall \bar{z}_{i,k_1} = 1$
5: set $\bar{N}_{k'} = \bar{N}_{k'} + 1$
6: check whether Constraint 1 on type k' holds
if $\bar{N}_{k'} < N_{k'}$
set $z^*_{i',k'} = 1$, $z^*_{i',k_1} = 0$,
$N' = N' - 1$, return to Step 2
7: **else**
$\boldsymbol{\Phi}_1 = \boldsymbol{\Phi}_1 \cup \{k'\}$, return to Step 3
8: **end if**
9: **else**
algorithm terminates
10: **end if**

5.4.3. Constraint 2—oriented algorithm modification

In this subsection, we examine Constraint 2, and apply controller-type adjustment if Constraint 2 does not hold. Examining Constraint 2, we can observe that given controller deployment strategy, the right side of inequality (17) is a constant, and the left side is the maximum difference between the processing delay of individual switches, which depends on the types of the controllers. Therefore, in the case that the inequality does not hold, we may reduce the difference between the processing delay of switches by adjusting the types of controllers.

Let Δ denote the maximum difference of the processing delay of switches, i.e.,

$$\Delta = \max_{i,j}\{T^{\mathrm{w}}_{i,j}\} - \min_{i,j}\{T^{\mathrm{w}}_{i,j}\}, \ \forall\ y^*_{i,j} = 1. \tag{33}$$

To reduce Δ, we first determine the controllers and switches corresponding to the maximum and minimum processing delay. Let $\mathrm{V}^*_{j_1}$ denote the switch which is of the maximum processing delay when associating with $\mathrm{C}^*_{i_1}$. Similarly, denote $\mathrm{V}^*_{j_2}$ as the switch which achieves the minimum processing delay when associating with $\mathrm{C}^*_{i_2}$. We obtain

$$\{\mathrm{C}^*_{i_1},\ \mathrm{V}^*_{j_1}\} = \arg\max_{i,j}\{T^{\mathrm{w}}_{i,j}\},$$

$$\{\mathrm{C}^*_{i_2},\ \mathrm{V}^*_{j_2}\} = \arg\min_{i,j}\{T^{\mathrm{w}}_{i,j}\}.$$

The controllers $\mathrm{C}^*_{i_1}$ and $\mathrm{C}^*_{i_2}$ can be one controller or different controllers. Accordingly, we consider the following two cases, i.e., Case a: $i_1 = i_2$, and Case b: $i_1 \neq i_2$.

For Case a, i.e., $i_1 = i_2$, we obtain $\Delta = \frac{t_0(\lambda_{j_1} - \lambda_{j_2})}{\mu_{i_1} - l_{i_1}}$, where $\mu_{i_1} = \sum_{k=1}^{K} \bar{z}_{i_1,k}\phi_k$ is the only term related to controller types. To reduce the value of Δ, we should increase the value of μ_{i_1} by increasing ϕ_k, i.e., matching controller $\mathrm{C}^*_{i_1}$ to a destination type with larger capacity under the condition that the increased network cost is minimum. More specifically, suppose the current type of $\mathrm{C}^*_{i_1}$ is k_1, we should shift the type of $\mathrm{C}^*_{i_1}$ to k' provided that the following condition meets:

$$\{k'\} = \arg\min_{k > k_1}\{\Psi_k\}. \tag{34}$$

For Case b, i.e., $i_1 \neq i_2$, we may reduce Δ by matching controller $\mathrm{C}^*_{i_1}$ to a destination type with larger capacity or matching controller $\mathrm{C}^*_{i_2}$ to a new type with smaller capacity. Compare the resulted network cost of these two options, we may select the one leading to lower cost. Without loss of generality, we assume that $\mathrm{C}^*_{i_1}$ and $\mathrm{C}^*_{i_2}$ are of type k_1, we compute the cost resulted from shifting $\mathrm{C}^*_{i_1}$ to type k_1+1, denoted by δ_{i_1,j_1,k_1+1}, and shifting $\mathrm{C}^*_{i_2}$ to k_1-1, denoted by δ_{i_2,j_2,k_1-1}. If $\delta_{i_1,k_1,k_1+1} \le \delta_{i_2,k_1,k_1-1}$, we should shift $\mathrm{C}^*_{i_1}$ to type k_1+1. Similarly, if $\delta_{i_1,k_1,k_1+1} > \delta_{i_2,k_1,k_1-1}$, we assign type k_1-1 to $\mathrm{C}^*_{i_2}$. Checking Constraint 2, if the constraint holds, the algorithm terminates, otherwise, we need to further shift the types of controllers following a similar manner.

Table 2
Capacity and cost of candidate controllers.

Parameter	Type 1	Type 2	Type 3	Type 4
Capacity (ϕ_k)	3000	4800	6600	7800
Cost I (η_k)	0.850	1.760	2.970	4.430
Cost II (η_k)	1.850	3.060	4.570	6.430

Table 3
Message request rate.

Cases	λ_j
Case 1	$\lambda_j \in [100, 300]$
Case 2	$\lambda_j \in [50, 200]$
Case 3	$\lambda_j \in [200, 400]$

Table 4
Simulation parameters.

Parameters	Value
Number of switches	20
Number of candidate controllers	3∼10
Link bandwidth ($B_{i,j}$, $B^h_{i,j}$)	30∼80 Mbps
Average size of packet-in messages (α)	160 bytes
Weighting factor(ω)	0.03
Duration (t_0)	1 ms

6. Simulation results

In this section, we evaluate the performance of our proposed algorithm through numerical simulations. We consider an SDN region with the size being 300 km × 300 km where various number of switches are randomly located and the connection status between any two switches is randomly set. The number of candidate controller types is set as 4, and the capacity of various types of candidate controllers is shown in Table 2. For comparison, we consider two cases of candidate controller cost, i.e., Cost I and Cost II, as shown in Table 2. From the table, we can see that for either Cost I or Cost II, different cost of individual types of controllers is set. To evaluate the performance of network, we either choose Cost I or Cost II as controller cost. We also consider four cases of message request rate as shown in Table 3 [6,26]. Other parameters used in the simulation are listed in Table 4, unless otherwise mentioned. The simulation results are averaged over 1000 independent experiments.

We compare the performance of our proposed algorithm with that of the algorithm proposed in [12], where a controller deployment strategy was designed to maximize the number of flows and minimize the propagation delay. To implement the algorithm proposed in [12] in our simulation scenario, the propagation delay of the links between switches and potential controllers is ranked in an ascending order, and the potential controllers which offer the smallest delay are then selected under budget constraint.

Fig. 2 displays the relationship between network cost and the number of controllers. To plot the figure, we set the number of candidate controllers as 12. For individual types, the number of candidate controllers is set as 3. We also set the candidate controller cost as Cost I and the message request rate as Case 3. For comparison, we evaluate the performance of two network scenarios which consist of 20 and 50 switches, respectively. The algorithm performance corresponding to different values of the controller cost weight (ω) is also evaluated. From

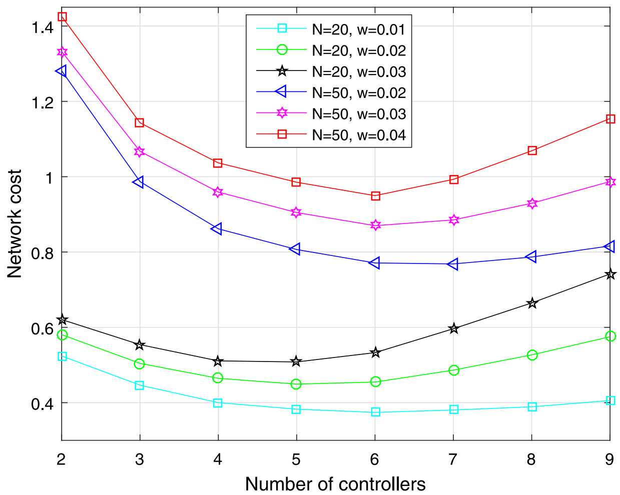

Fig. 2. Network cost versus the number of controllers (different network scales and weight factors).

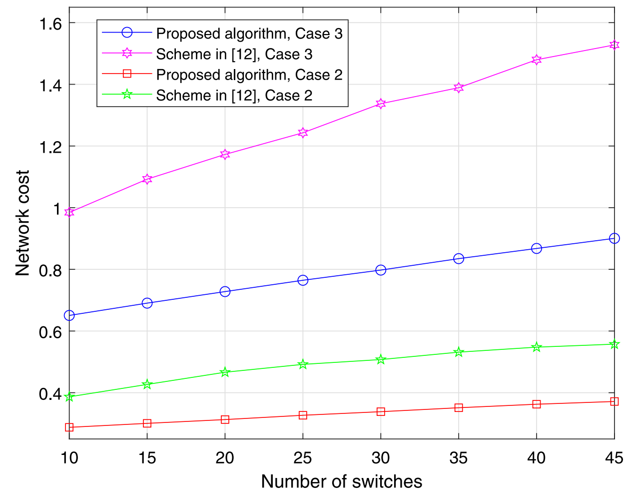

Fig. 4. Network cost versus the number of switches (different message request rates).

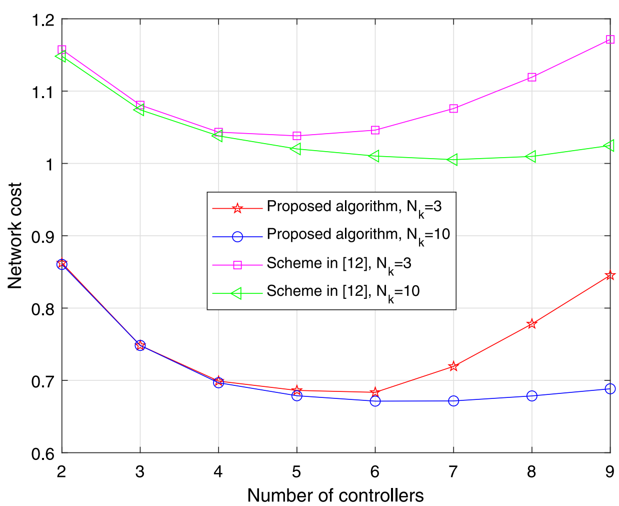

Fig. 3. Network cost versus the number of controllers (various maximum number of candidate controllers).

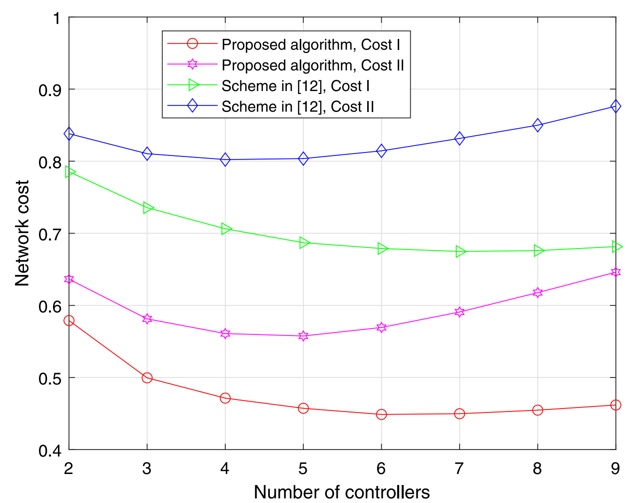

Fig. 5. Network cost versus the number of controllers (different controller costs).

the figure we can see that as the number of controllers starts to increase, network cost decreases slightly and achieves a minimum value. This is mainly because the NRT decreases when increasing the number of controllers. However, further increasing the number of controllers, the corresponding network cost increases. The reason is that controller cost increases as the number of controllers increases. Examining the network cost resulted from difference network scales characterized by the number of switches, we can observe that network cost increases for larger network scale. Moreover, when the weighting factor of controller cost increases, the network cost increases as well.

Fig. 3 depicts the network cost obtained from our proposed algorithm and the comparison algorithm. To plot the figure, we set the candidate controller cost as Cost I and the message request rate as Case 2. Different numbers of the maximum candidate controllers of certain types are considered in the simulation. As can be observed from the figure, when the number of deployed controllers is greater than 4, lower network cost is achieved for the scenario with sufficient candidate controller. This is because as the number of deployed controllers increases, the load on each controller decreases. When the number of candidate controllers is sufficient, it is more likely to select controllers with low cost; however, in the case that the number of candidate controllers of certain types is highly limited, the controllers with higher capacity and higher cost might be selected, resulting in larger network cost. In addition, the performance of the proposed algorithm outperforms the one proposed in [12].

In Fig. 4, we evaluate network cost versus the number of switches for different message request rates. In the simulation, we set the number of candidate controllers for individual types as 3 and the candidate controller cost as Cost II. It is obvious that network cost increases with the increase in the number of switches. The reason is that for a larger number of switches, the worst-case CPRT may increase, resulting in higher NRT. It can be observed from the figure that as the messages request rate increases, the network cost increases accordingly. This is because the increase in the message request rate leads to the higher load of the controllers, resulting in the increased processing delay and network cost. Compared to the reference algorithm, our proposed algorithm can obtain better performance.

In Fig. 5, we examine the impact of controller cost on network performance. In the simulation, we set the number of candidate controllers of individual types as 10 and the message request rate as Case 1 . For comparison, we set the parameters of controller cost respectively as Cost I and Cost II (as given in Table 2). From the figure, we can see that when the cost of controllers is set as Cost II, much higher network cost is resulted compared to that of Cost I, which illustrates the severe

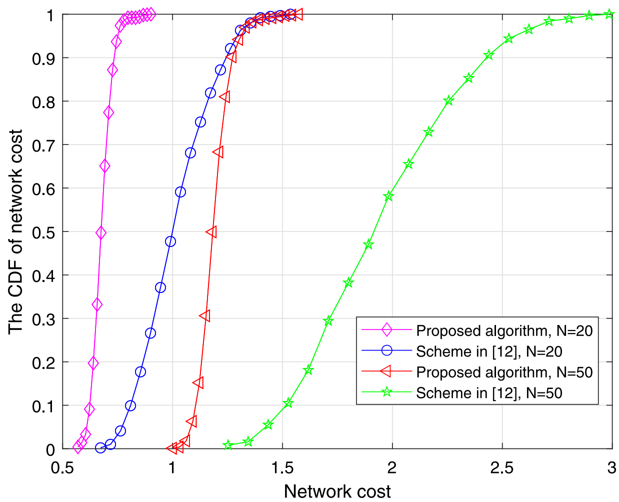

Fig. 6. CDF of network cost.

impact of controller cost on network cost. Comparing the performance obtained from our proposed algorithm and the one proposed in [], we can see our proposed algorithm achieves much lower network cost, and requires less number of controllers. This is because our proposed algorithm intends to minimize network cost while the algorithm proposed in [12] aims to minimize the propagation delay and maximize the number of user flows, and fails to consider the transmission delay and processing delay, as well as the cost of controllers.

Fig. 6 depicts the cumulative distribution probability (CDF) of network cost in two scenarios with different number of switches. To plot the figure, we set the number of candidate controllers of each type as 10, the controller cost as Cost I and the message request rate as Case 2. The results are computed based on 1000 experimental outcomes corresponding to randomly chosen simulation parameters. Given a specific set of simulation parameters, the possibility that network cost is less than a certain value is computed. Obviously, the network cost resulted from our proposed algorithm is considerably smaller than that resulted from the one proposed in [12], which certifies the effectiveness of the proposed algorithm.

7. Conclusions

In this paper, we investigated the capacitated controller deployment problem in SDN networks, and proposed a two-stage heuristic algorithm to minimize the defined network cost. Specifically, a minimum eccentricity-based controller deployment strategy and a greedy method-based controller-type matching algorithm were proposed in the first stage and the second stage, respectively. In order to examine the performance of our proposed algorithm, we conducted simulations in comparison with a previously proposed algorithm. From the simulation results, we could observe that as the number of deployed controllers increases, network cost first decreases and then increases. Hence, an optimal number of deployed controllers offering the minimum network cost could be obtained. We also examined the impacts of network scale, weighting factor and controller cost on network performance. Simulation results verified that the proposed algorithm could achieve better performance than previously proposed scheme.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] W. Xia, Y. Wen, C.H. Foh, D. Niyato, H. Xie, A survey on software-defined networking, IEEE Commun. Surv. Tutor. 17 (1) (2015) 27–51.

[2] D. Kreutz, F.M.V. Ramos, P.E. Verlssimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: A comprehensive survey, Proc. IEEE 103 (1) (2015) 14–76.

[3] J. Lu, Z. Zhang, T. Hu, P. Yi, J. Lan, A survey of controller placement problem in software-defined networking, IEEE Access 7 (2019) 24290–24307.

[4] T. Das, V. Sridharan, M. Gurusamy, A survey on controller placement in SDN, IEEE Commun. Surv. Tutor. 22 (1) (2020) 472–503.

[5] X. Hou, M. Wu, B. Lv, Y. Liu, Multi-controller deployment algorithm in hierarchical architecture for SDWAN, IEEE Access 7 (2019) 65839–65851.

[6] K. Yang, D. Guo, B. Zhang, B. Zhao, Multi-controller placement for load balancing in SDWAN, IEEE Access 7 (2019) 167278-167289.

[7] C.N. Sminesh, E.G.M. Kanaga, A. Roy, Optimal multi-controller placement strategy in SD-WAN using modified density peak clustering, IET Commun. 13 (20) (2019) 3509–3518.

[8] Y. Qi, D. Wang, W. Yao, H. Li, Y. Cao, Towards multi-controller placement for SDN based on density peaks clustering, in: 2019 IEEE International Conference on Communications (ICC), 2019, pp. 1-6.

[9] 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) (2018) 344–355.

[10] S. Bera, S. Misra, N. Saha, Traffic-aware dynamic controller assignment in SDN, IEEE Trans. Commun. 68 (7) (2020) 4375–4382.

[11] J. Xu, L. Wang, C. Song, Z. Xu, Minimizing multi-controller deployment cost in software-defined networking, in: 2019 IEEE Symposium Computers and Communications (ISCC), 2019, pp. 1-6.

[12] Z. Guo, W. Chen, Y. Liu, Y. Xu, Z. Zhang, Joint switch upgrade and controller deployment in hybrid software-defined networks, IEEE J. Sel. Areas Commun. 37 (5) (2019) 1012–1028.

[13] A. Ruiz-Rivera, K. Chin, S. Soh, Greco: An energy aware controller association algorithm for software defined networks, IEEE Commun. Lett. 19 (4) (2015) 541–544.

[14] Y. Hu, T. Luo, N.C. Beaulieu, C. Deng, The energy-aware controller placement problem in software defined networks, IEEE Commun. Lett. 21 (4) (2017) 741–744.

[15] G. Li, X. Wang, Z. Zhang, SDN-Based load balancing scheme for multi-controller deployment, IEEE Access 7 (2019) 39612–39622.

[16] J. Liao, H. Sun, J. Wang, Q. Qi, K. Li, T. Li, Density cluster based approach for controller placement problem in large-scale software defined networkings, Comput. Netw. 112 (2017) 24–35.

[17] Y. Liu, H. Gu, F. Yan, X. Yu, K. Wang, Multi-controller placement based on two-sided matching in inter-datacenter elastic optical networks, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), 2020, pp. 1-6.

[18] S. Petale, J. Thangaraj, Failure-based controller placement in software defined networks, IEEE Trans. Netw. Serv. Manag. 17 (1) (2020) 503–516.

[19] R. Chai, Q. Yuan, L. Zhu, Q. Chen, Control plane delay minimization-based capacitated controller placement algorithm for SDN, EURASIP J. Wirel. Commun. Netw. (282).

[20] B.P.R. Killi, S.V. Rao, Capacitated next controller placement in software defined networks, IEEE Trans. Netw. Serv. Manag. 14 (3) (2017) 514–527.

[21] Q. Qin, K. Poularakis, G. Iosifidis, S. Kompella, L. Tassiulas, SDN controller placement with delay-overhead balancing in wireless edge networks, IEEE Trans. Netw. Serv. Manag. 15 (4) (2018) 1446–1459.

[22] C. Du, Q. Chen, J. Li, L. Zhang, Network cost optimization-based capacitated controller deployment for SDN, in: 2020 IEEE/CIC International Conference on Communications in China (ICCC), 2020, pp. 1045-1050.

[23] M.T.I. ul Huque, W. Si, G. Jourjon, V. Gramoli, Large-scale dynamic controller placement, IEEE Trans. Netw. Serv. Manag. 14 (1) (2017) 63–76.

[24] M. Khorramizadeh, V. Ahmadi, Capacity and load-aware software defined network controller placement in heterogeneous environments, Comput. Commun. 129 (2018) 226–247.

[25] M. Davoodi, K-balanced center location problem: A new multi-objective facility location problem, Comput. Oper. Res. 105 (2019) 68–84.

[26] M. Tanha, D. Sajjadi, R. Ruby, J. Pan, Capacity-aware and delayguaranteed resilient controller placement for software-defined wans, IEEE Trans. Netw. Serv. Manag. 15 (3) (2018) 991–1005.

Rong Chai received her Ph.D. degree in electrical engineering from McMaster University, Ontario, Canada, in 2008. She received the M.S. degree and the B.E. degree from University of Electronic Science and Technology of China, Chengdu, China, in 1998 and 1995, respectively. In 2008, she joined the School of Communication and Information Engineering, Chongqing University of Posts and Technology, where she is currently a professor. Her research interest is in wireless communication and network theory and she has authored or coauthored 95 research papers.

Xizheng Yang is currently working toward the M.S. degree in Chongqing University of Posts and Technology, Chongqing, China, where he received the B.E. degree in 2019. His research interests include software-defined networking, and access control of blockchain.

Chunling Du is currently working toward the M.S. degree in Chongqing University of Posts and Technology, Chongqing, China, where she received the B.E. degree in 2018. Her research interests include software-defined networking, and network resource management.

Qianbin Chen received his Ph.D. degree in electrical engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2006. In 1988, he received the B.S. degree from Sichuan University, Chengdu China, and joined the School of Communication and Information Engineering, Chongqing University of Posts and Technology, where he is currently a professor. He has been working in the areas of wireless and mobile networking for more than 20 years and has authored more than 120 international journals and conference articles. His research interest is in wireless communication, network theory and multi-media technology.