On Reliability-optimized Controller Placement for Software-Defined Networks

HU Yannan, WANG Wendong, GONG Xiangyang, QUE Xirong, CHENG Shiduan

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, PR. China

Abstract: By decoupling control plane and data plane, Software-Defined Networking (SDN) approach simplifies network management and speeds up network innovations. These benefits have led not only to prototypes, but also real SDN deployments. For wide-area SDN deployments, multiple controllers are often required, and the placement of these controllers becomes a particularly important task in the SDN context. This paper studies the problem of placing controllers in SDNs, so as to maximize the reliability of SDN control networks. We present a novel metric, called expected percentage of control path loss, to characterize the reliability of SDN control networks. We formulate the reliability-aware control placement problem, prove its NP-hardness, and examine several placement algorithms that can solve this problem. Through extensive simulations using real topologies, we show how the number of controllers and their placement influence the reliability of SDN control networks. Besides, we also found that, through strategic controller placement, the reliability of SDN control networks can be significantly improved without introducing unacceptable switch-to-controller latencies.

Key words: Software-Defined Networking; controller placement; reliability; network optimization

I. INTRODUCTION

Traditionally, both control and forwarding planes of packet networks have been highly integrated on the same boxes. Each network device in the network not only forwards packets, but also computes and maintains the states required for its operation. Although this architecture has made great success, control planes of traditional networks are increasingly becoming more and more complex to accommodate a richer set of control requirements. This high complexity makes the task of designing and managing networks more and more difficult and further creates a large burden and high bearer to new protocols and technology.

Having recognized the problem, a more recent trend, reflected by a range of previous works (e.g. 4D [1], RCP [2], Ethane [3], and OpenFlow-based technology [4-6]), is decoupling the control and data planes. In this mode, the intelligence of the network is moved onto one or more external servers, called controllers, which make up a network-wide logically centralized control plane that oversees a set of “dumb, and simply” forwarding elements. Control applications that reside in the control platform are then responsible for controlling and managing the network based on a global network view. Nowadays, this network control paradigm is often referred to as Software-Defined Networking (SDN).

A novel reliability metric, called expected percentage of control path loss, is presented to characterize the reliability of SDN control networks. Among the algorithms proposed in this paper, simulated annealing algorithm provides solutions that are close to optimal.

In SDNs, the characteristics of the control plane play an important role in the overall network performance. Generally speaking, the SDN control plane consists of three parts [6]: control platform, which handles state distribution (e.g. reading information from and writing control instructions to forwarding elements using a well-defined API, as well as coordinating the appropriate state among other control platform servers); control applications, which are developed upon a programmatic interface that the control platform provides; and control network (or connectivity infrastructure), which is used for propagating events to packet-forwarding devices or between multiple controllers.

Since network failures that disconnect the control and data planes could prevent the forwarding plane from requesting instructions from the control plane and may lead to severe packet loss and performance degradation, it is of great importance to improve the reliability of SDN control networks. Although the design choice can be affected by numerous factors, we find that the number of controllers and their placement (along with the assignment relationship between switches and these controllers) place fundamental limits on reliability of SDN control networks. Figure 1 illustrates four simple options when deploying one or two controllers in a five-node SDN, where the solid lines represent physical links. In this figure, each switch is controlled by exactly one controller (in case of two controllers, each switch could be assigned to either one). To reduce propagation latencies [7], for any placement of controller(s) and switch-to-controller assignment relationship, the control packets between each switch and its controller, as well as between controller pairs are transmitted along the shortest path between them. These paths compose the control network, and are represented by dotted lines in Figure 1. Clearly, the control network in Figure 1(a) is less reliable than the ones in Figure 1(b), 1(c) and 1(d) where two controllers are deployed, since the use of one controller could introduce the single point of failure problem. On the other hand, the placement of controllers also affects the reliability of control network. For example, the control network in Figure 1(c) is more reliable than the one in Figure 1(b), because if the physical link between switch A and B in Figure 1(b) fails, the communication path between node A and its controller (controller 1), as well as the one between the two controllers could be broken, whereas the failure of any physical link in Figure 1(c) could at most break one communication path. Moreover, the reliability of control network can also be affected by the switch-to-controller assignment relationship. In Figure 1(d), the two controllers are placed at exactly the same locations as in Figure 1(c). However, switch A is controlled by controller 1 in Figure 1(d) instead of controller 2 in Figure 1(c). As a result, the control network in Figure 1(d) is less reliable than the one in Figure 1(c), because if the physical link between switch B and C fails in Figure 1(d), the communication paths from both node A and B to their controller (controller 1) could be broken, whereas the same failure in Figure 1(c) could only break one communication path.

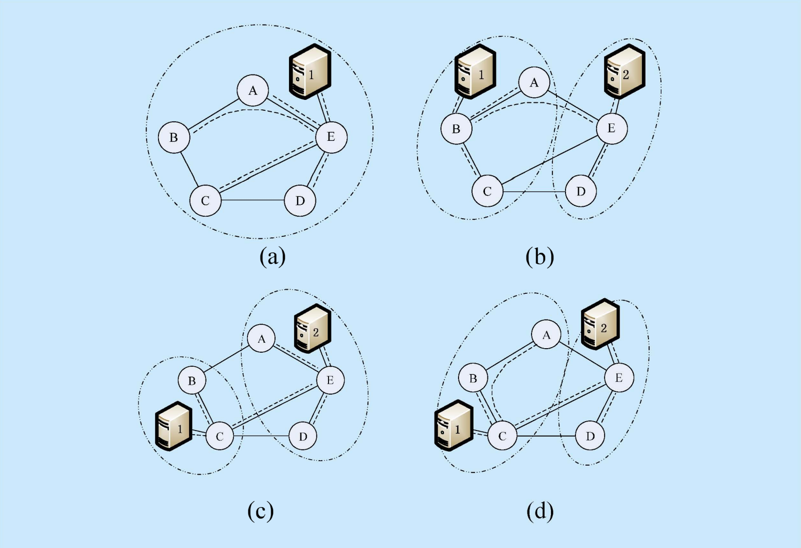


Fig.1 *Options in deploying controllers in a 5-node SDN*

Motivated by the above observation, in this paper, we are interested in finding the answers to the following questions: given a physical network and the failure probability of each network component (e.g. links and switches), how many controllers are needed and how to place them such that a pre-defined reliability objective is optimized. We call this design choice the reliability-aware controller placement (RCP) problem. The reasons that make this research necessary are three folds: (1) As stated previously, it is very meaningful to improve the reliability of SDN control networks. A reliable control network increases the network availability. (2) We need to investigate the impact of physical failures on the reliability of SDN control network. In traditional networks, where the control and data packets are commonly transmitted on the same path, previous works therefore assume that the control and data plane are equally affected when failure happens. However, this is not the case in SDNs as control packets may take different paths from data packets. How to model the reliability of the control network for SDNs is an important problem. (3) It is also very helpful to further give hints on reliable control network design for general network topologies. For example, intuitively, network topology affects the most reliable controller placement. However, is there any common trend that we can follow? Besides, as propagation delay bounds the control reactions with a remote controller in SDN [7], what are the tradeoffs between reliability and latencies?

To find the most reliable controller placements for SDNs, we propose a novel metric that reflects the reliability of the SDN control networks, called expected percentage of control path loss, where the control paths are defined as the route set that is used for communications between switches and their controllers, as well as between controllers themselves. After presenting our analysis framework for dependent control path failures, we formulate the RCP problem, which decides where to place the controllers and the assignment of switches to controllers to minimize the expected percentage of control path loss, and hence maximize the reliability. We prove that the RCP problem is NP-hard and develop two heuristic placement algorithms that automate the controller placement decision. These algorithms and the benefits of reliability-aware controller placement are evaluated through extensive simulations using real topologies. We also verify our proposal by quantifying the tradeoffs between metrics, e.g., reliability and latencies.

The rest of this paper is organized as follows. We review related work in Section II. Section III proposes our reliability metric, along with the analysis method. We also formulate the RCP problem and prove its NP-hardness in Section III. Section IV presents a number of placement algorithms. Section V covers the simulations and our findings. Finally, we conclude the paper in Section VI.

II. Related Work

This paper builds on our earlier work in Ref. [8]. We make the following extensions in this paper: we present the detailed reliability analysis framework for SDN control networks, and formulate the reliability-aware controller placement (RCP) problem; we relax the assumption which ignores the benefit of any control path protection mechanisms; we formally prove that the RCP problem is NP-hard; we present a complete simulation and evaluation of placement algorithms and tradeoffs between reliability and latencies.

There has been considerable work on reliability of traditional networks [9-12]. However, the models considered in these work assume that each device in the network either can function independently or consists of both control and forwarding planes. As a result, these models cannot be applied to the SDN architecture with decoupled control and forwarding planes [13].

To assuage the scalability concerns raised when building SDNs with a logically centralized control plane, one of the practices is to improve the performance of a single controller by embracing well-known optimization techniques (e.g. multi-threads and I/O batching)

[14-15]. Alternative approaches are then to work on distributed implementations of the controllers (also valuable for fault tolerance), which include Onix [6] and HyperFlow [16]. Although these implementations use multiple controller instances, these work did not study the problem of placing controllers such that a given objective can be satisfied. We tackle this problem from the reliability perspective in this paper.

On the robustness of SDN, most prior work focused on eliminating network configuration errors or software bugs, which potentially stem from human initiated failures. For example, Kazemian et al. [17] proposed a general and protocol-agnostic framework to analyze the network configuration such that connectivity and isolation errors can be detected; Canini et al. proposed a NICE [18] way to test OpenFlow applications bugs by combining model checking and symbolic execution; Frenetic [19] abstracts aspects of SDNs to improve code correctness. Complementarily, our approach mitigates the impact of inevitable physical failures on the control networks.

Resiliency of the SDN architecture was studied in Ref. [13] and [20]. In Ref. [13] the authors considered the SDN architecture with a single controller, and proposed a resilience-aware controller placement algorithm and a control traffic routing algorithm to maximize the possibility of control traffic fast failover. However, their approach cannot be applied to environments where multiple controllers are required. To increase the resiliency of communication paths between switches and controllers, ZHANG et al. [20] proposed a min-cut based controller placement algorithm to partition the network, such that switches inside each partition are well connected and provide rich connectivity to the controller. Although this is similar to our goal, we propose a different optimization metric in this paper, which characterize the reliability of SDN control networks more precisely. More importantly, we formally establish that the problem of finding a reliable controller placement is NP-hard, and further study the relationship of controller placement and reliability using real topologies.

In Ref. [7], the authors motivated the controller placement problem and examined the impacts of placements on average and worst-case propagation latencies on real topologies. They find that most networks show diminishing returns for each added controller, and one controller location often suffices. While propagation latency is certainly a significant performance metric, we argue that reliability design is also an essential part for operational SDNs. This paper optimizes the reliability of SDN control networks, and further qualifies the tradeoffs between reliability and latencies.

III. Reliability-Aware Controller Placement

In SDNs, switches communicate with their controller via standard TLS or TCP connections. When multiple controllers are deployed, communications between these controllers are also required to achieve global consistency of network state (e.g. network topology and host address mapping) and coordinating control of the entire network. However, they are out of scope of this paper as we mainly focus on improving reliability through controller placement. We suppose that state synchronization and control coordination between controllers can be realized by adopting necessary techniques such as hierarchy.

In this paper, we make the following three assumptions:

1) There are only in-band connections between switches and their controllers, as well as between different controllers. That is, we assume that the control traffic uses the existing connections between switches in the network.

2) We envision a SDN network where primary paths are preinstalled for aggregated data traffic between switches. The controllers are thus not contacted for each single flow, but only in case of traffic engineering or outages to find backup paths. Consequently, the

amount of control traffic is limited. Therefore, we assume that there is no load balancing on the control traffic. That is, control traffic between each switch and its controller, or between each controller pair, is primarily transmitted over only one path, which we will refer to as control path (note that, protection mechanisms are still needed in case the primary path fails). If we view each control path as a logical link, control traffic is actually transmitted over an overlay network, named control network, above the physical network.

3) Control networks for SDNs may take any form [7]. To make the problem studied in this paper more concrete, we take hierarchy control network as an example. In other words, we assume that multiple controllers are connected in a full mesh in the control network, which connect to forwarding nodes below. Note that our method can easily be applied to other forms of control networks.

In this Section, we first define our reliability metric based on the failure of control paths. Due to the correlations among different control paths, the reliability of SDN control network is investigated using the dependent failure analysis model [21]. We finally give the mathematical formulation of the reliability-aware controller placement problem, and prove its NP-hardness.

3.1 Physical network and control network

The physical network is denoted as graph $G(V,E)$, where V is the set of nodes (switches) in the network, and E is the set of bidirectional edges (links) between nodes. (u,v)represents the physical link from node u to v. Given any two nodes s and t, let Pst denote the path from s to t provided by specific routing mechanism used in the network.

The SDN control network is overlaid on top of the physical network. We denote the control network as graph $G_c(V_c, E_c)$. V_c is the set of switches that send/receive or distribute the control traffic. Clearly, in a functional SDN, V_c=V. Ec is the set of control paths. In a SDN with hierarchy control network, controllers are attached to some switches, and the network is grouped into several clusters, each of which is controlled by the only controller it contains. Therefore, E_c includes the control paths between switches and their controllers and the full mesh of control paths among all controllers, i.e., $\boldsymbol{E}_c = \{\langle u, v\rangle | u, v \in \boldsymbol{V}_c$, there is a control path between u and $v\}$, where $\langle u, v\rangle$ denote the control path between u and v[1].

Given a control network and the specific routing mechanism for control traffic, we can map control path $\langle u, v\rangle$ to a sequence of physical links and nodes on the paths P_{uv} and P_{vu}. Moreover, we use $\boldsymbol{P}_{e_c}$ to denote the set of switches and links on control path e_c, where $e_c \in \boldsymbol{E}_c$.

3.2 Reliability metric

In literature, the network reliability can be broadly defined through deterministic or probabilistic metrics [11]. Two basic deterministic reliability criteria are the connectivity and cohesion of the graph underlying a network, which denote the minimum cardinality of a node cut-set and an edge cut-set, respectively. However, deterministic metrics usually fail to characterize the susceptibility of networks to disconnection since they do not account for the reliability of network components. In contrast, probabilistic metrics assume that components may fail with a certain probability, and thus provide more meaningful measures of network reliability. In general, probabilistic reliability metrics can be further divided into two categories, connectivity-based metrics and traffic-based metrics [22].

Connectivity-based measures focus on the network topology and connectivity. A common connectivity-based metric is the K-terminal reliability, which is defined as the probability that a subset K of nodes can communicate with each other [23]. More specifically, this measure considers a network to be operative if communication paths between each pair of the subset K of nodes exist. Traffic-based metrics, on the other hand, measure reliability with

[1]For convenience, we view each controller and the switch it attached to as a single element in control network. As a result, the existence of $<u,v>$ means that either u or v is controller attached, or u and v are attached to different controllers respectively.

the performance of the network, which can be defined as the fraction of traffic of the entire network that can be carried by the network [24]. Traffic-based measures consider the traffic-carrying capacity such that the degree of service disruptions due to failures can be quantified.

For a SDN, the main function of its control network is to distribute network control traffic along control paths between different devices of control plane and data plane. In a healthy SDN, there are valid control paths between switches and their controller, as well as between different controllers, such that the switch can behave in a proper way and global network state consistency can be achieved. In case that a control path fails, disconnections between a forwarding device and its controllers or between different controllers will partially invalidate the function of the control network. Consequently, the more control paths fail, the more control traffic interruptions happen, and the severer the forwarding service disruption is. Therefore, the traffic-based metrics are more meaningful for SDN control networks. However, since there is no detailed control traffic pattern available at present, in this work, we assume each control path carries one unit of traffic, and thus use a modified version of traffic-based metric to characterize the reliability of SDN control network. Our reliability metric is defined as the expected percentage of control path loss, where the control path loss is defined as the number of broken control paths due to network failures.

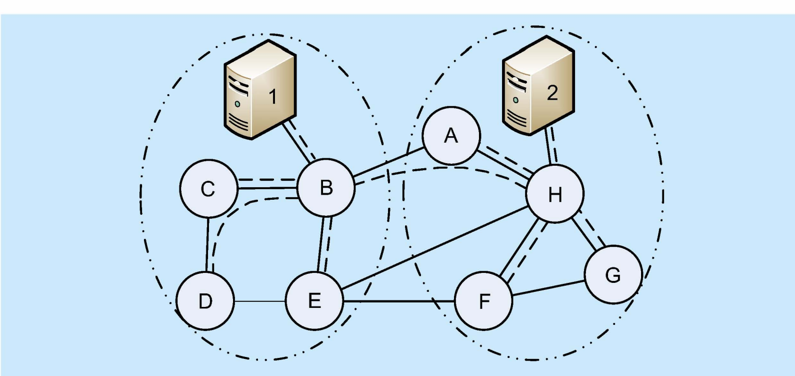


Fig.2 *An example of SDN control network. The network is grouped into two clusters. Solid lines represent physical links; dotted lines represent control paths*

The optimization target is then to minimize the expected percentage of control path loss.

3.3 Reliability analysis framework

In SDN control networks, the reliability analysis is complex due to the correlations between different control paths. In many scenarios, two overlaid control paths may share a common physical link or switch, and thus control paths fail in a dependent fashion. There are several approaches to studying network reliability with dependent component failures. In this paper, we adopt a commonly-used dependent failure model, the cause-based reliability analysis model [21], in SDN control networks. The basic idea is that failures of the control network components, e.g., control paths, have underlying physical causes that can be explicitly identified and are statistically independent.

We first identify all major failure scenarios in physical network. For example, a failure scenario can be the case where a single switch or a single link fails. These failure scenarios happen independently. Each failure scenario may affect the states of several control paths in the control network. Let S be the set of all network scenarios, including the failure scenarios we identified and one special scenario, where no failure exists. In scenario $e_c \in \boldsymbol{E}_c$, let F_s denote the set of failed physical components, and Fs is a subset of $V \cup E$. Other components in the network, which are not in F_s, work properly. Moreover, we use p_s to denote the probability that scenario s happens, which can be derived from the historical network operation information, and it is easy to see that $\sum_{s \in S} p_s = 1$.

When scenario s happens, the control paths between the switches in F_s and their controllers fail. If Fs partitions the network[2], the control paths that traverse also fail. For example, in Figure 2, the failure of switch H isolates controller 2 from the network, making all control paths traverse switch H to controller 2 fail, and the failure of switch C isolates switch D from the network, which breaks the control path $\langle \boldsymbol{B}, \boldsymbol{D} \rangle$. On the other hand, if the network is not partitioned, the affected control paths may and may not fail, depending on the failover mechanisms used for control traffic, e.g. control path protection or rerouting. For example, in Figure 2, if IP link (A, H) fails, the control paths $\langle B, H\rangle$ and $\langle A, H\rangle$ fail with a certain probability. For example, as shown in [13], by carefully pre-configuring backup links in the switches, it is possible that the control traffic between a switch and its controller can be rerouted when there is a failure in the primary control path. We thus model the impact of network failures on the control paths as a probability variable. Specifically, we denotes qs as the conditional failure probability of the control paths that are affected by the network failures in scenario s, i.e.,

q_s=[control path e_c fails|F_s in scenario s fails]

where $e_c \in E_c$, $s \in S$, and $P_{e_c} \cap F_s = \emptyset$. As stated previously, q_s is related to the failover mechanisms used for control paths. To simplify the following discussion, we assume that qs is the same for all influenced control paths in scenario s. Moreover, in a given failure scenario, all components in physical network $G(V, E)$ are in deterministic states. Each switch detects failures and reroutes control traffic independently. Thus, the conditional failures of control paths are independent, and reliability analysis techniques in networks with independent failures can be applied.

For each failure scenario, we calculate the number of failed control paths, and our reliability metric, the expected control path loss, can be obtained. In a SDN control network $G_c(V_c, E_c)$, suppose m is the total number of control paths, i.e., $m=|E_c|$. We define θ_s as the total number of control paths that are possessed by the nodes in F_s [3]; let ϕ_s denote the total number of control paths that pass F_s, but which are not owned by F_s. θ_s and ϕ_s are determined by the control network topology and underlying physical network topology. For example, if C fails, i.e., $F_s = \{C\}$, in Figure 2, since control path $\langle B, C\rangle$ is owned by C, it definitely fails and θ_s=1; however, since control path $\langle B, D\rangle$ only passes C, it fails with a certain probability (depending on the specific failover mechanisms) and ϕ_s=1.

Finally, we have the expected percentage of control path loss δ defined as follows:

$$\delta = \sum_{s \in S} p_s \frac{\theta_s + \phi_s q_s}{m} \tag{1}$$

Generally speaking, the size of S can be very large if we want to cover all failure scenarios. However, in practice, we can get a satisfying statistical coverage (i.e., $\sum_{s\in S} p_s$ is very close to 1) by only analyzing the most probable failure scenarios. According to Ref. [25-26], the possibility that multiple physical components fail simultaneously in one administrative domain is extremely small, and most of the failures only involve single network component. Therefore, we can only take account of the failure scenarios where at most one physical component (link or node) fails at any time in one administrative domain. The failed links or switches can be repaired before the next failure event takes place. Under this situation, the number of failure scenarios is $|V|+|E|$, and this gives us enough precision for the purpose of designing SDN control networks.

3.4 Problem formulation

For network graph $G(V, E)$, let n be the number of nodes, i.e., $n=|V|$. Link weights represent propagation latencies. We denote the failure probabilities of switch $v(v \in V)$ and link $e(e \in E)$ by pv and pe, respectively. Also, let q_v and q_e denote the conditional failure probabilities of the control paths that are affected by the failures of v and e, respectively. We define $V_c \subseteq V$ to be the set of candidate locations for hosting controllers. Let M denote the set of controllers to be placed in the network, k and $P(M)$ denote the number and a possible topological placement of these controllers, respectively. Let c denote the total number of controller-to-controller adjacencies in the control network, i.e., the total number of control paths between controllers. Obviously, in a hierarchy control network, $c=k(k-1)/2$. Moreover, to re-

[2]By partition the network, we mean the situations where the failure of Fs isolates either some of the controllers or some of the switches from the network.

[3]If the switches in Fs are controller attached, θ_s should also include the control paths between the attached controllers and other controllers.

[4]Technically, the binary identifiers in the formulation are dependent variables that change with yj and xij. The reason is that, given a possible placement of controllers and the switch-to-controller assignment relationships, since each control path is deployed on the shortest path, the value of each binary identifier can always be exactly identified.

duce the propagation delay of control traffic, for any controller placement, each control path $\langle u, v\rangle$ is deployed on the shortest path between u and v (in terms of propagation delay). The notations are summarized in Table I.

We now formally state the problem studied in this paper, which decides the placement of a given number of controllers to maximize the reliability of SDN control networks.

Definition 1. (Reliability-aware Controller Placement (RCP) Problem): In network $G(V, E)$, given (1) Vc ($V_c \subseteq V$), set of candidate locations for hosting controllers; (2) p_v and p_e, the failure probabilities of each switch and each link; (3) q_v and q_e, the impact of switch failure and link failure on control paths; (4) set of shortest paths between node pairs; and (5) a positive integer k, the RCP problem is to find a controller placement, $P(M)$', of size k such that the percentage of expected control path loss δ is minimum.

Suppose $i \in V$, $j, h \in V_c$. Let $y_j = 1$ indicates that a controller is placed at location j; $x_{ij} = 1$ indicates switch i is allocated to controller at location j; $y_{jh} = 1$ if a controller-to-controller adjacency is identified between controllers at location j and h; otherwise these binary variables are 0. For example, in Figure 2, $y_B = 1$, $x_{BC} = 1$ and $y_{BH} = 1$. Moreover, let c_j be the number of switches that are controlled by the controller at location j. In other words, there are c_j controller-to-switch control paths at location j. For example, in Figure 2, $c_B = 3$ and $c_H = 3$. Then, for $v \in V$, θ_v, the number of control paths that are possessed by v, is

$$\theta_v = \begin{cases} \sum_{j \in V_c} y_{vj} + c_v, & y_v = 1 \\ \sum_{j \in V_c} x_{vj}, & y_v = 0 \end{cases} \tag{2}$$

where $\sum_{j \in V_c} y_{vj}$ represents the total number of control paths between the controller at location v and other controllers , and $\sum_{j \in V_c} x_{vj}$ represents the total number of control paths between node v and the controllers. Note that there are k controllers in the network, and each switch is assigned to exactly one controller, as a result,

$$\theta_v = \begin{cases} k - 1 + c_v, & y_v = 1 \\ 1, & y_v = 0 \end{cases}$$
$$\Leftrightarrow (k - 1 + c_v)y_v + (1 - y_v), \qquad y_v \in \{0, 1\}$$
$$= (k - 2 + c_v)y_v + 1, \qquad y_v \in \{0, 1\}$$

(3)

Moreover, for node $v \in V$, let binary identifier $g_{ijv} = 1$ denote the control path between switch i and controller at location j passes v; f_{jhv}=1 denotes the control path between two controllers at location j and h passes v; otherwise these identifiers are 0[4]. For example, in Figure 2, g_{DBC}=1 and f_{BHA}=1. Then, ϕ_v, the number of control paths that pass node v, but which are not owned by v, is

$$\phi_v = \sum_{\substack{i \in V, j \in V_C \\ i, j \neq v}} g_{ijv} x_{ij} + \sum_{\substack{j,h \in V_C, j<h \\ i, j \neq v}} f_{ihv} y_{jk} \tag{4}$$

Since $V_c \subseteq V$, the above equation can be expressed in a more efficient way. Suppose $i \in V$, $j \in V_c$. Let $y_j = 1$ indicate that a controller is placed at location j, 0 otherwise. $x_{ij} = 1$ indicates there is a control path between i and j, which means, $x_{ij} = 1$ exactly when either $y_j = 1$ and switch i is assigned to j, or $y_j = y_j = 1$ and there is an adjacency between controller i and j, 0 otherwise. For node $v \in V$, let binary iden-

Table I *Notations of Reliability-aware Controller Placement Problem*

Notations	Implication
$G(V,E)$	Physical network with node set V and link set E
$G_c(V_c,E_c)$	SDN control network with node set V_c and control path set E_c
(u,v)	Link from switch u to switch v
$\langle u, v\rangle$	Control path between u and v
S	Set of all network failure scenarios
F_s	Set of failed components in scenario s, $F_s \subseteq V \cup E$
p_s	Probability that network failure scenario s occurs
q_s	Conditional control path failure probability in scenario s
n	Number of switches in the network
m	Number of control paths in the network, $m=\|E_c\|$
p_v	Failure probability of switch v
p_e	Failure probability of link e
q_v	Conditional control path failure probability of v
q_e	Conditional control path failure probability of e
V_c	Set of candidate locations for hosting controllers
K	Number of controllers to be placed
c	Total number of controller-to-controller adjacencies
θ_v	Number of control paths possessed by switch v
ϕ_v	Number of control paths pass switch v but not own by v
ϕ_e	Number of control paths pass link e
δ	Expected percentage of control path loss

tifier $h_{ijv} = 1$ denote the control path between i and j passes v, 0 otherwise. For example, in Figure 2, $h_{DBC} = 1$ and $h_{BHA} = 1$.Then, ϕ_v is

$$\phi_v = \sum_{\substack{i\in V, j\in V_c \\ V_c\subseteq V, i,j\neq l}} h_{ijv} x_{ij} \tag{5}$$

Similarly, for link $e \in E$, let binary identifier $h_{ije} = 1$ denotes the control path between i and j passes link e, 0 otherwise. Then, ϕ_e, the number of control paths that pass link e is

$$\phi_e = \sum_{\substack{i\in V, j\in V_c \\ V_c\subseteq V}} h_{ije} x_{ij} \tag{6}$$

Given the number of switch $n=|V|$, the total number of control paths in the network is

$$m = n + c = n + \sum_{i,j\in V_c, i<j} x_{ij} \tag{7}$$

Therefore, according to equation (1), the expected percentage of control path loss due to network failure is

$$\begin{aligned}\delta &= \sum_{e\in E} p_e \frac{\phi_e q_e}{m} + \sum_{v\in V} p_v \frac{\theta_v + \phi_v q_v}{m} \\ &= \frac{1}{m}\left[\sum_{e\in E} p_e \phi_e q_e + \sum_{v\in V} p_v(\theta_v + \phi_v q_v)\right] \\ &= \frac{1}{m}\left\{\sum_{e\in E} \sum_{\substack{i\in V, j\in V_c \\ V_c\subseteq V}} p_e h_{ije} x_{ij} q_e + \right. \\ &\left. \sum_{v\in V} p_v\left[(k-2+c_v)y_v + 1 + \sum_{\substack{i\in V, j\in V_c \\ V_c\subseteq V, i,j\neq l}} h_{ijv} x_{ij} q_v\right]\right\}\end{aligned} \tag{8}$$

We now present the integer programming formulation for the RCP problem as follows:

minimize δ

Subject to:

$$\sum_{i\in V_c} y_j = k \tag{9}$$

$$\sum_{j\in V_c} x_{ij} = 1 \ \{\forall i | y_i = 0, i \in V\} \tag{10}$$

$$\sum_{i,j\in V_c, i<j} x_{ij} = k(k-1)/2 \tag{11}$$

$$\sum_{i,j\in V_c, i<j} x_{ij} - (k-1)y_j/2 \leqslant 0 \tag{12}$$

$$y_j, x_{ij} \in \{0,1\} \quad \forall i \in V, j \in V_c \tag{13}$$

The object function is to minimize the percentage of expected control path loss due to physical failures. Equation (9) in the formulation is the constraint that the total number of controllers should be equal to k. Equation (10) ensures that a switch is assigned to exactly one controller. Equation (11) is the constraint that guarantees the correct number of controller-to-controller adjacencies in the object function. Equation (12) means that adjacencies for locations where a controller is not present are forbidden. Equation (13) indicates that y_j and x_{ij} are binary variables.

3.5 Complexity of the RCP problem

We now show that the RCP problem is NP-hard, even in a very simplified case. The proof sketch is as follows: first, we define a special case of the RCP problem, referred to as the *sRCP* problem; second, we prove that the sRCP problem is NP-hard by reducing it from the well-known Dominating Set (DS) problem which is NP-hard; finally, we conclude that the RCP problem is NP-hard as well.

The sRCP problem is a special case of the RCP problem that satisfies the following two additional conditions: (1) $p_v=0$, for $\forall v \in V$, i.e., there is no switch failure in the network; (2) $q_e=1$. As a result, the objective of the sRCP problem is to minimize $\delta' = \sum_{i\in V}\sum_{j\in V_c} c_{ij}x_{ij}$, where $c_{ij} = \sum_{e\in E} h_{ije}p_e$, which is defined as the cost of the control path between i and j.

Theorem 1. The sRCP problem is NP-hard.

Proof. Our proof is based on the reduction from the well-known the Dominating Set (DS) problem which is NP-hard.

The decision version of the sRCP problem is: Given a network graph $G(V,E)$ with cost on each link, V_c ($V_c \subseteq V$) set of candidate controller locations, an integer k and a real number C, does there exist a subset $V_c' \subseteq V_c$ of size k, such that $\delta' \leqslant C$?

The Dominating Set Problem: Given an undirected graph $G(V,E)$, a dominating set is a set V' such that for any $v \in V$, either $v \in V'$ or v is adjacent to a node in V'. The decision version of DS problem is: given a graph G and a number k, is there a dominating set of G of size at most k.

Obviously the decision version of sRCP problem is in NP. It's easy to count the number of controllers and to calculate the total control path cost given the controllers.

Given an instance of the DS problem, we construct an instance of the sRCP problem as follows.

1) Let the set of switches and the set of candidate controller locations of sRCP problem be the vertices V in the DS problem.

2) Let the cost of control path c_{ij} between switch i and controller j be the shortest path length from i to j.

3) Let the cost of control path between each controller pair be zero.

4) Let $C=|V|-k$.

Obviously, the construction can be finished in polynomial time. Now we prove that the sRCP problem has a total control path cost of at most C if and only if there is a dominating set of size at most k.

$\Rightarrow$: Assume the sRCP problem has a total control path cost of at most C. Then, there is a dominating set of size at most k. Since the total control path cost is at most $C=|V|-k$, the k controllers form a dominating set and each of other $|V|-k$ switch is adjacent to a node in the dominating set with a shortest path of length 1.

$\Leftarrow$: Suppose there is a dominating set of size k, let those vertices be the controller locations, and if the size of the dominating set is less than k, just add more vertices as controller locations to compensate the exact k. Since each of other $|V|-k$ vertices has to be assigned to exactly one controller, and is one hop away from any controller, we have the total control path cost is C.

We proved the sRCP problem is NP-hard.

Finally, since the sRCP problem is a special case of the original RCP problem, we know the RCP problem is also NP-hard.

IV. Evaluation

This section presents the results of computational experiments based on real topologies. In our simulations, we assume all nodes in the networks are capable of hosting controllers, i.e., $V_c=V$. In this section, we first evaluate two heuristic algorithms proposed in our earlier work [8], which are the l-w-greedy (l-w) algorithm and Simulated Annealing (SA) algorithm respectively, along with the random placement (RD) algorithm for comparison purpose, using the Internet2 OS3E topology [27]. We then expand our simulation to six more ISP network topologies obtained from Rocketfuel project [28], to characterize reliability performance against the size of controllers and further give hints on designing the most reliable control network. Finally, we analyze the controller placement tradeoffs between reliability and other design metrics, such as latencies. The key characteristics of network topologies used in this section are summarized in Table II.

Table II *Main characteristics of experiment networks*

Networks	OS3E	AS 1221	AS 1239	AS 1755	AS 3257	AS 3967	AS 6461
# nodes	34	104	315	87	161	79	138
# links	42	302	1945	322	656	294	744

4.1 Algorithm comparison

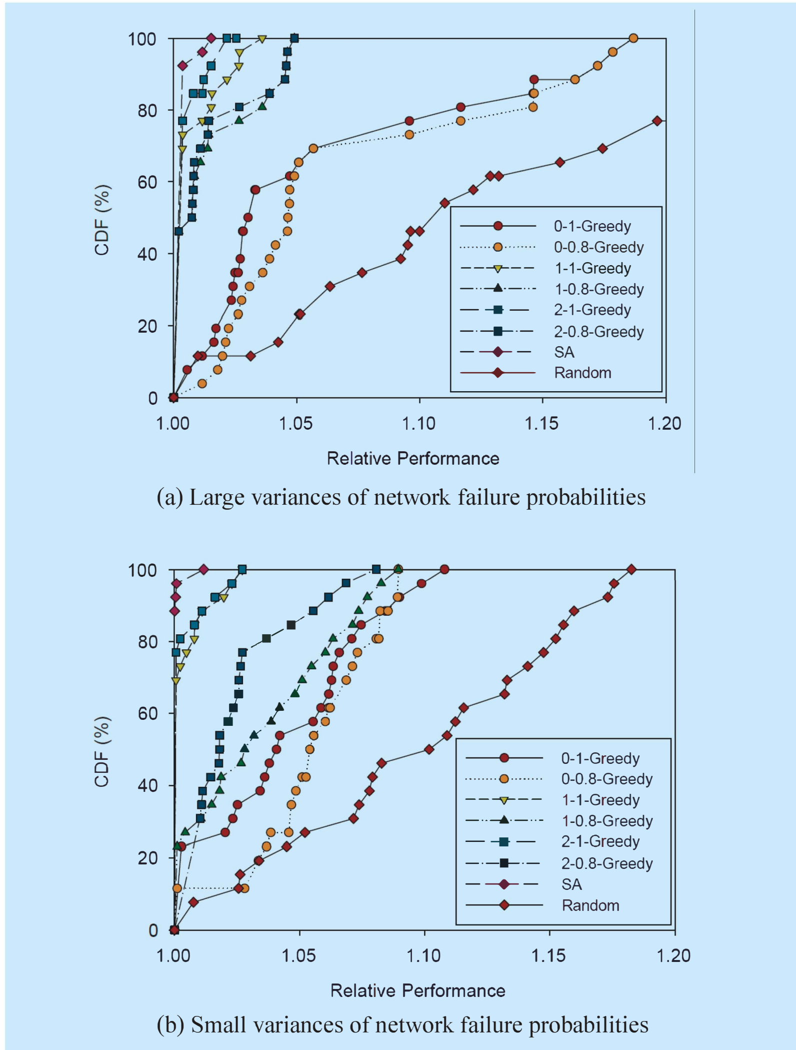
The algorithms are evaluated on the OS3E topology under different setups of the network component failure probabilities. For each failure probability setup, we first assign a value randomly chosen from {0,1} to each network component, representing the conditional control path failure probability that is affected by the failure of that network component, and then run a set of simulations. In each set of the simulation, we first pick k, the number of controllers to be placed. For the given k, we run one simulation for the following algorithms: 0-1-greedy, 0-0.8-greedy, 1-1-greedy, 1-0.8-greedy, 2-1-greedy, 2-0.8-greedy and simulated annealing. Since the random algorithm returns different results based on the locations selected, we execute random placements over 1000 times for a given number of controllers, and select the placement that yields the best performance. Then, we repeat all simulations for the next k. In our simulations, we experiment with k ranging from 1 to 27, where the upper bound of k is determined according to the setting of w (0.8, in our case).

To compare the performance of the algorithms on various failure probability setups and controller numbers, we use the *relative performance* of the algorithms as a metric, which is defined as the ratio between δ, the expected percentage of control path loss, of the feasible solution found by the algorithms to the one that is determined by the optimal placement. A relative performance of 1.0 indicates that the algorithm returns an optimal solution. To get the optimal placements for comparison, a natural approach is to use a brute force algorithm that generates and ranks all the possible placements. However, since this algorithm is exhaustive, we take a more time-efficient approach in our experiment. For each failure probability setup and controller number, we obtain the optimal controller placement by solving the integer linear program in Section 3.4 using CPLEX [29].

The comparison results are insensitive to variations in the failure probabilities, and it is easy to identify the algorithm that performs the best. For illustration purposes, we only report the results under the following two failure probability setups: the failure probability of each network component is randomly generated from interval [0,0.1] and [0.02,0.04], which stand for two scenarios where large and small variances of network component failure probabilities happen, respectively.

Figure 3 shows the cumulative distribution (CDF) of the relative performance across all simulation runs of different algorithms in both scenarios. As we can see, although the variances of network failure probabilities slightly affect the performance of the algorithms, it is easy to find out that the best three algorithms in both failure scenarios are: simulated annealing, 2-1-greedy and 1-1-greedy. Among these three, the SA outperforms the other two algorithms. On the other hand, we find that the random placement algorithm has the worst performance.

To quantify the differences between different algorithms, Figure 4 shows the minimum, maximum and median values of the relative performance of these algorithms in

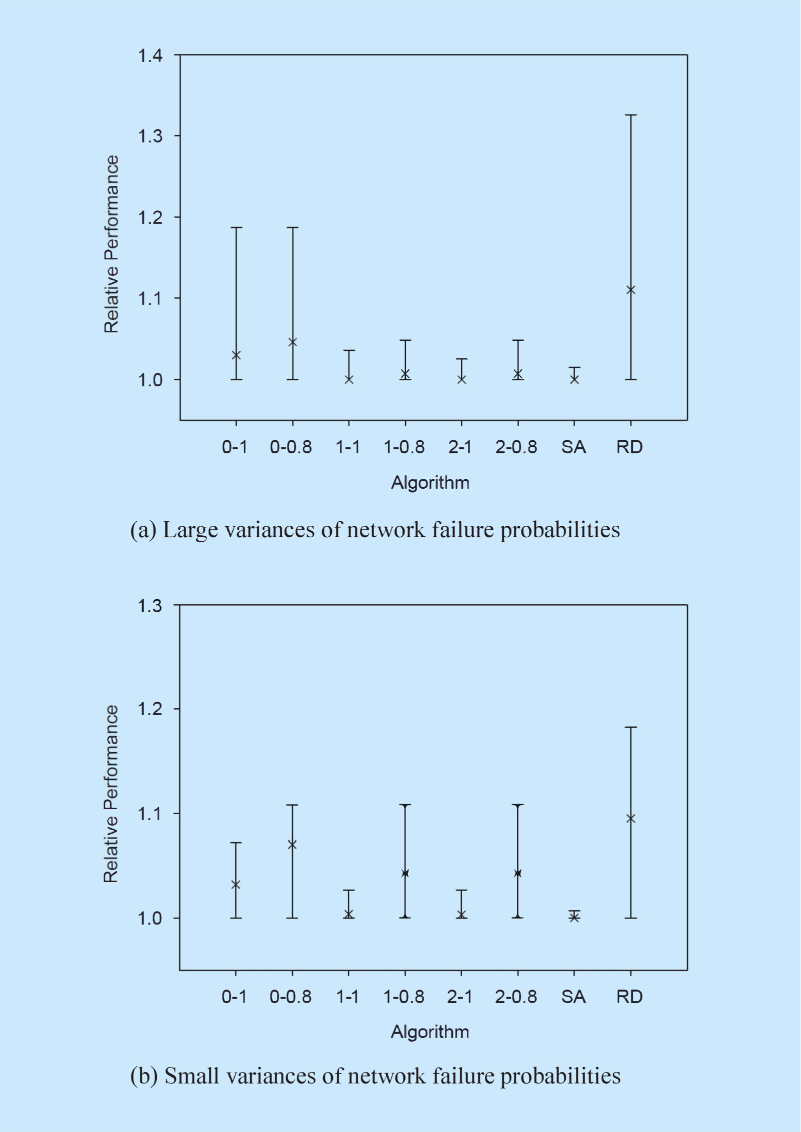


(a) Large variances of network failure probabilities

(b) Small variances of network failure probabilities

Fig.3 *The CDF of relative performance across all simulation runs of the placement algorithms on OS3E topology*

both scenarios over all simulation runs. For both scenarios, the SA algorithm performs the best, with only 0.02% worse than the optimal on average. The 2-1-greedy and 1-1-greedy algorithms perform within 0.15% and 0.37% worse than the optimal, respectively. The 1-0.8-greedy and 2-0.8-greedy algorithms perform roughly the same, with about 3.6% worse than the optimal results. The 0-1-greedy and 0-0.8-greedy have slightly worse performances that are respectively about 4.1% and 7.8% worse than the optimal. The performance of random placement is the worst among all algorithms. For the two experiment scenarios, we observe that up to 50% of the simulations



(a) Large variances of network failure probabilities

(b) Small variances of network failure probabilities

Fig.4 *A summary of the differences between the placement algorithms on OS3E topology using error bars.*

for the random algorithm have a relative performance of at least 1.2.

4.2 Impact of controller number on reliability

Having identified that the SA algorithm provides the solutions that are close to optimal, we now present the results of our simulations on the OS3E and Rocketfuel topologies, aiming to find the answer to the following question: if the controllers are carefully placed, how many controllers should we use in order to maximize reliability. To make the relationships between controller number and reliability metric on different networks comparable, for all the topologies, we set the failure probability and the conditional control path failure probability of each network component to the same values, 0.02 and 1, respectively. For Rocketfuel topologies, we use the reported latency values between vertices pairs as link weights, and consider only the SA algorithm in our experiments.

As expected, the answer depends on the topology. However, we see a range of common trends. Figure 5 shows the relationship between the number of controllers and the reliability of control network. The horizontal axis is the ratio between k and the total number of nodes in the network, n. The vertical axis is the expected percentage of control path loss due to network failures. We are surprised to find that reliability optimizations on different topologies provide similar benefits. Placing too many or too few controllers reduces the reliability of control network; whereas when k is the number in the middle the most reliable controller placement can be obtained. The reason is that, if the number of controllers is too small, some switches have to use long paths to connect to controllers, which increases the possibility of control path loss; on the other hand, when placing too many controllers, the control paths between controllers become the main determinant of the control network reliability, and this makes the reliability low. In our experiments, the best controller number, k', is in between $[0.035n, 0.117n]$, as shown in Table III. Furthermore, although the number of controllers that maximizes the reliability of SDN control network increases with the network node number, we find that the ratio between k' and n decreases with the topology scale. For example, to maximize reliability, the 34-node OS3E topology requires 4 controllers and the ratio is 0.117, whereas the AS 1239 topology, which consists of 315 nodes, requires 11 controllers and the ratio is 0.035. However, there are no observable n-dependent patterns.

4.3 Tradeoffs between reliability and latencies

Like all optimization problems, designing an appropriate control networks for SDN generally requires tradeoffs between metrics. In Ref. [7], Heller et al. argued that the "best" controller placement for wide-area networks is the one that minimizes propagation delays. Since the focus of this paper is to maximize the reliability of control networks, a natural question is: what are the tradeoffs between reliability and latencies? Specifically, we are interested in finding the tradeoffs between reliability and average latency, as well as between reliability and worst-case latency. The simulations are executed on the OS3E and the Rocketfuel topologies. We set the network component failure probabilities of each topology to the same values as in Section 5.2.

In the OS3E topology, for each number of controllers, the optimal placement for each metric consists of a different set of locations, with sets overlap in some cases. The cloud of points in Figure 6(a) shows the reliability and average latency metrics for all possible placements of up to five controllers. Since we are only interested in the points in the lower-left region, Figure 6(b) shows only the points that are on the pareto frontier. To more easily evaluate the tradeoffs, Figure 6(c) is the normalization of the pareto frontiers. We can now directly evaluate the tradeoffs between reliability and average latency. Similarly, Figure 7 illustrates the placement tradeoffs between the reliability and worst-case latency metrics. As shown in Figure 6 and Figure 7, the tradeoffs between reliability and latencies vary with the number of controllers. For example, when placing one controller, the average latency optimized placement is also the reliability optimized placement; however, in the worst case, we see a 15.1% reduction in reliability when k = 4. We then look at the data at the value of k around 3-4, which are the points of diminishing returns found by Heller et al. [7]. On average, when optimizing for average and worst-case latency, we see 9.17% and 9.03% decreases in reliability, respectively; on the other side, when optimizing for reliability, the

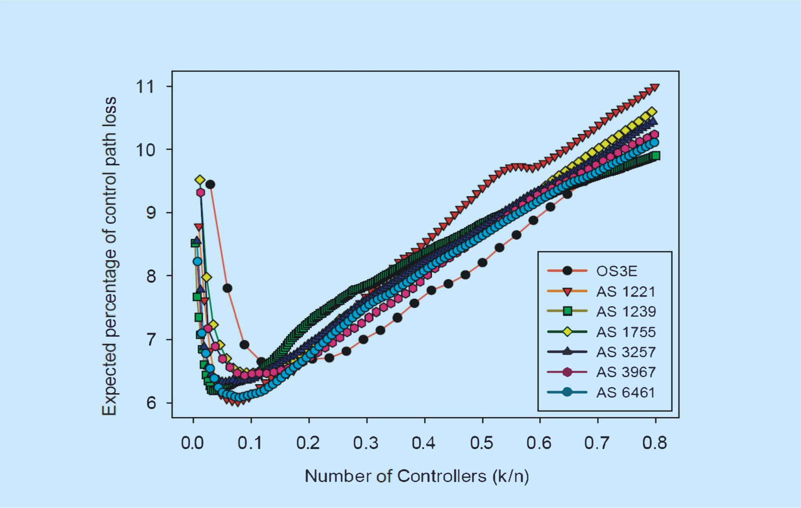


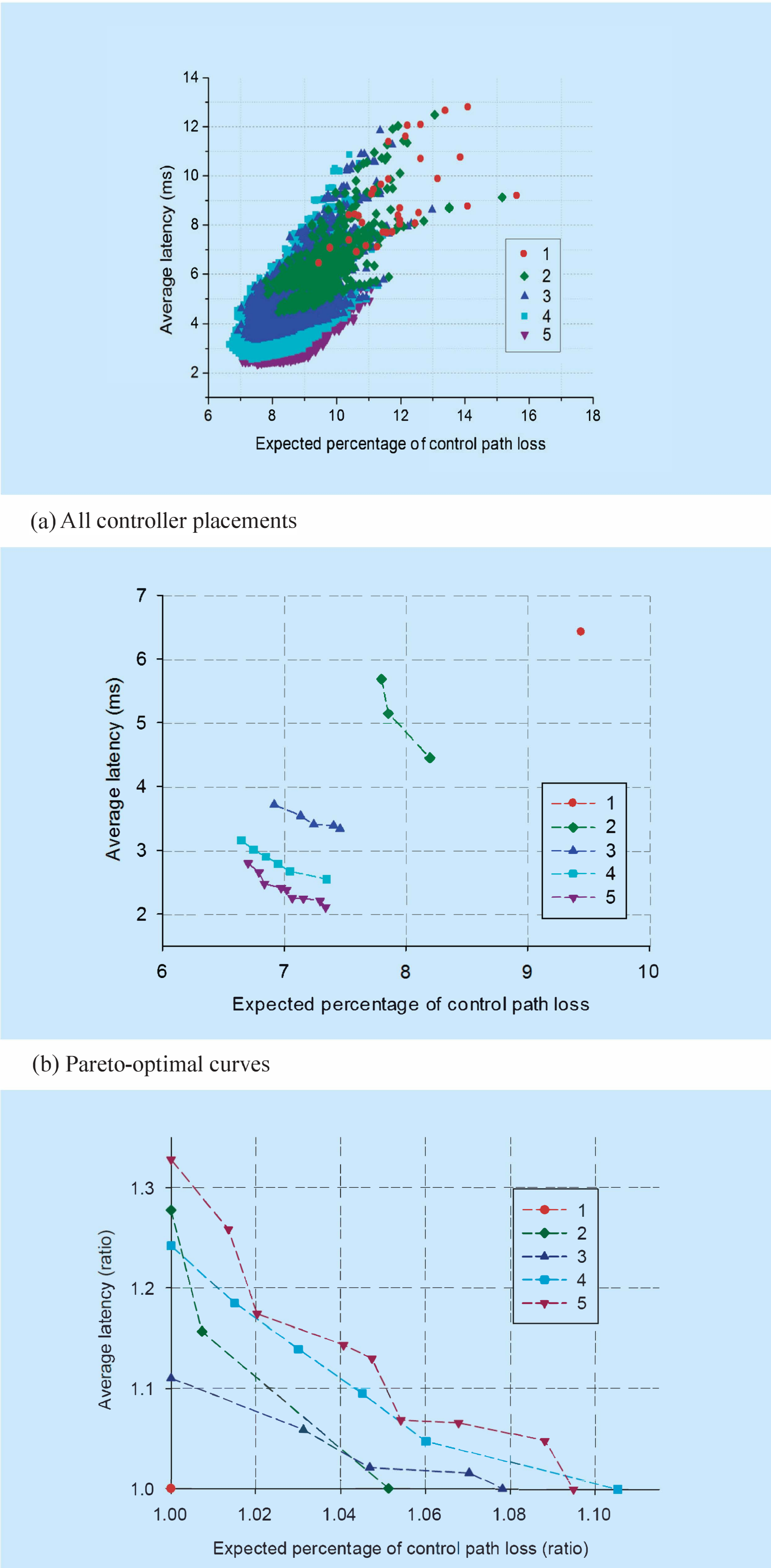
Fig.5 *Reliability metric vs. the number of controllers on different topologies. Smaller is better.*

Table III *The best number of controllers on different topologies*

Networks	OS3E	AS 1221	AS 1239	AS 1755	AS 3257	AS 3967	AS 6461
n	34	104	315	87	161	79	138
k'	4	8	11	9	12	7	11
k'/n	0.117	0.077	0.035	0.103	0.075	0.089	0.08

average and worst-case latency increase by 17.3% and 13.2%, respectively.

It is clearly that optimal values for both reliability and latency metrics are often impossible to achieve at the same, a nature question is then to ask: if one is willing to trade latency for reliability, are the resulting latencies acceptable? To answer this question, we use Figure 8 to illustrate the corresponding one-way latencies when controllers are placed at the locations that optimizes reliability on both OS3E and Rocketfuel topologies. The latencies for different topologies are represented by different curves. Moreover, to get a full knowledge of these delays, we compare them to the three delay bounds that are used in Ref. [7] (shown as horizontal lines). Obviously, for all topologies, when reliability is given high priority, we observe that it is quite sufficient for the corresponding average latencies to meet the medium delay bound (25 ms). Actually, for most of the experiment topologies, placing controllers at the reliability-optimized locations presents



(a) All controller placements

(b) Pareto-optimal curves

(c) Normalized pareto-optimal curves

Fig.6 *Placement tradeoffs between reliability and average latency for k = 1 to k = 5; (b) shows the best placements from (a); (c) normalizes the curves in (b).*

no fundamental limit for average latencies to meeting even the strictest delay bound, which is 5 ms. Besides, even when worst-case latency is more concerned, we see no unacceptable node-to-controller worst-case latencies comparing to the mesh restoration delay bound (100 ms).

V. Conclusions and Future Work

This paper studies the problem of placing controllers in Software-Defined Networks (SDNs) to maximize the reliability of SDN control networks in detail. It presents a novel reliability metric (expected percentage of control path loss), and analyzes the dependent control path failures in SDN control networks. After formulating the reliability-aware controller placement problem, the paper proves its NP-hardness. Several placement algorithms and their benefits are examined using real topologies. The paper also qualifies the tradeoffs between reliability and latencies.

The main results and conclusions of this paper are as follows. First of all, placement performance depends on the specific algorithm used. Among the algorithms proposed in this paper, simulated annealing algorithm provides solutions that are close to optimal. In addition, even when controllers are strategically placed, the number of them should be chosen properly. Placing too many or too few controllers reduces reliability. Finally, simulation results show tradeoffs between metrics. However, the corresponding latencies when optimizing for reliability is sufficient to meet existing response-time requirements.

In the future, we plan to extend our work in two directions. First of all, we want to consider other design choices that could not be covered by this paper. For example, besides the placement of controllers, reliability of SDN control networks can also be improved by carefully constructing control networks (e.g., configuring primary control paths as well as backup paths, or assigning switches to some backup controllers besides their primary controllers). We will extend our methods by jointly considering these design choices and provide a comprehensive solution. In addition, our analysis is based on the assumption that each switch only connects to one controller using one path. While this assumption can be practical for some networks where primary paths are preinstalled for aggregated data traffic between switches, it could become invalid in other network environments. For example, in a WAN that connecting multiple datacenters, if the controllers are responsible for installing forwarding paths for every single flow, due to the high flow arrival rate, load balancing for control traffic is quite necessary. As a result, we will relax this assumption and study the resulting reliability improvement problem in our future work.

ACKNOWLEDGEMENT

This work was supported in part by the National High Technology Research and Development Program (863 Program) of China under Grant No. 2011AA01A101, the National High Technology Research and Development Program (863 Program) of China under Grant No. 2013AA013301, and the National High Technology Research and Development Program (863 Program) of China under Grant No. 2013AA013303.

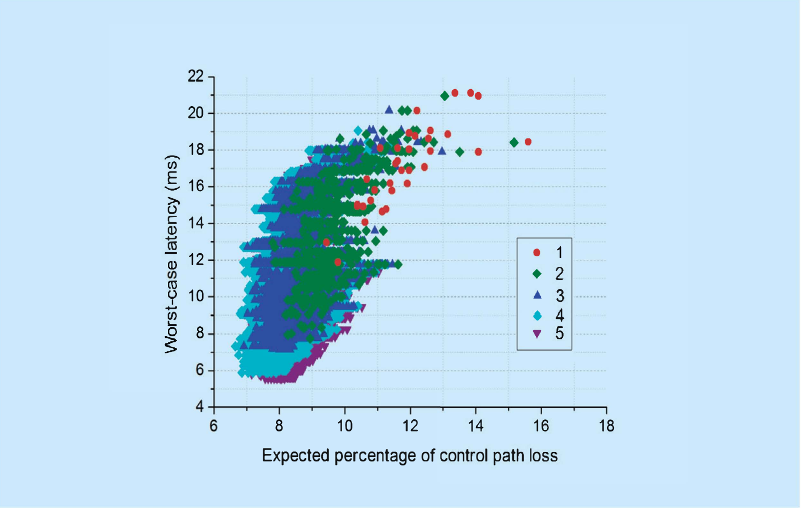
References

[1] GREENBERG A, HJALMTYSSON G, MALTZ A D, *et al.* A clean slate 4D approach to network control and management[J]. SIGCOMM Computer Communication Review, 2005, 35(5): 41-54.

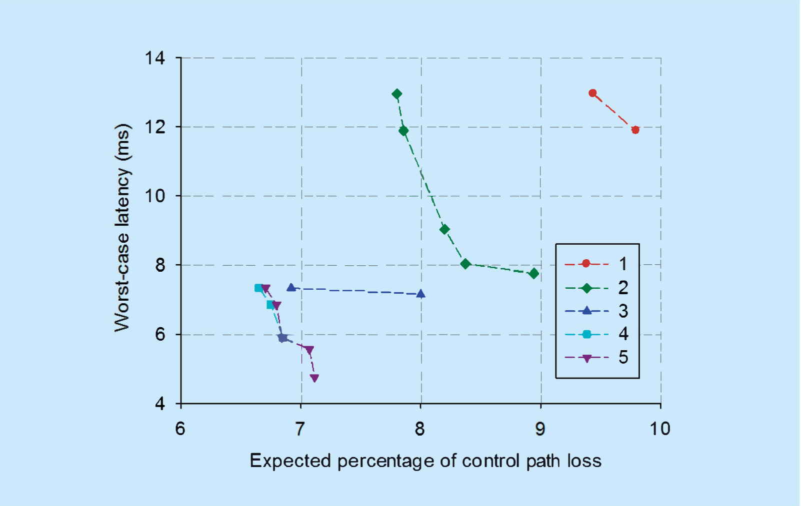
[2] CAESAR M, CALDWELL D, FEAMSTER N, *et al.* Design and implementation of a routing control platform[C]// Proceedings of the 2nd conference on Symposium on Networked Systems Design and Implementation (NSDI 2005) Volume 2: May 2-4, 2005, Boston, MA, USA. USENIX Association, 2005: 15-28.

[3] Casado M, Freedman J M, Pettit J, *et al.* Ethane: taking control of the enterprise[J]. SIGCOMM Computer Communication Review, 2007, 37(4): 1-12.

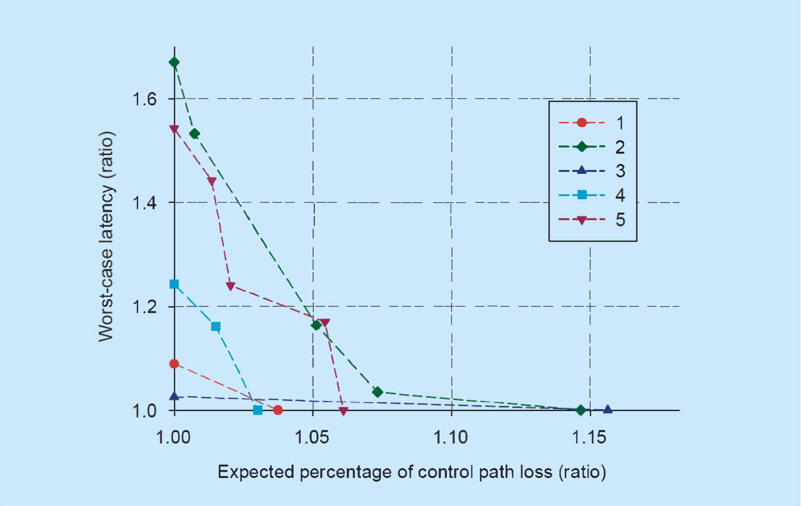
[4] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, *et al.* OpenFlow: enabling innovation in cam-



(a) All controller placements

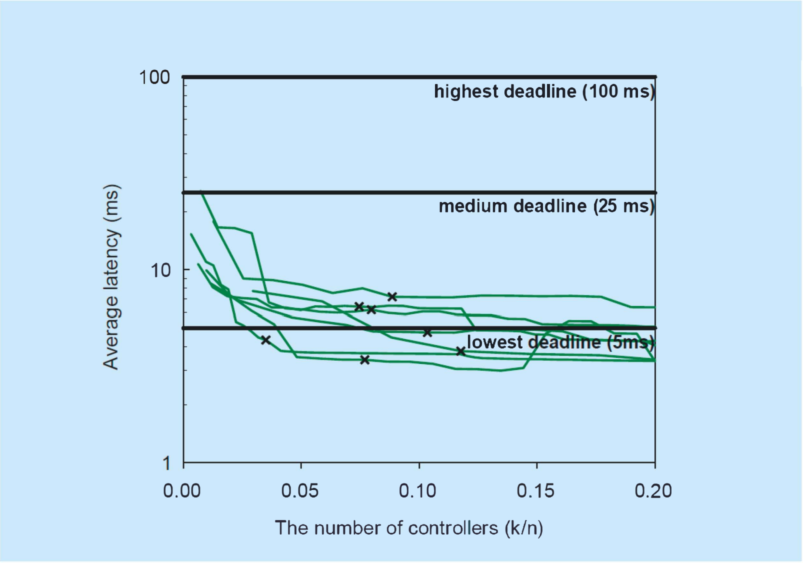


(b) Pareto-optimal curves

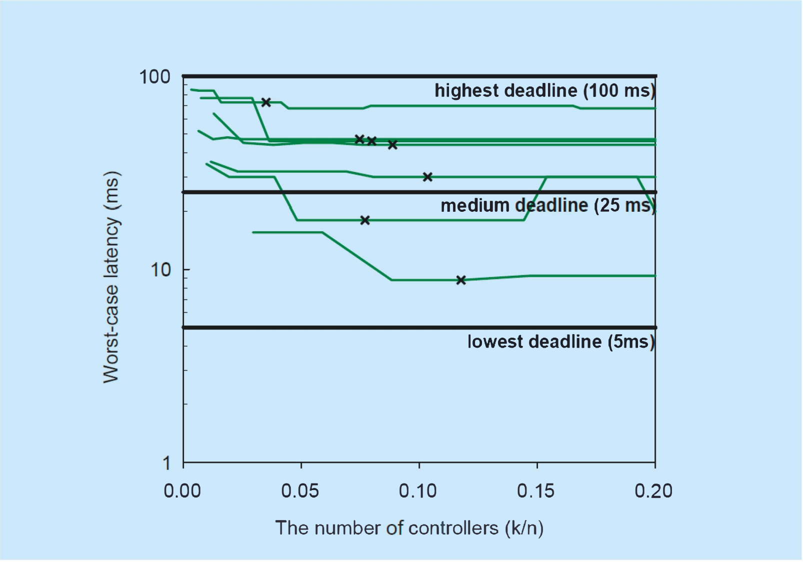


(c) Normalized pareto-optimal curves

Fig.7 *Placement tradeoffs between reliability and worst-case latency for k = 1 to k = 5; (b) shows the best placements from (a); (c) normalizes the curves in (b).*



(a) Average latencies



(b) Worst-case latencies

Fig.8 *One-way latencies when reliability is optimized on both OS3E and Rocketfuel topologies (log-scaled). The X mark on each curve represents the corresponding latencies of the reliability-optimized placement.*

pus networks[J]. SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.

[5] GUDE N, KOPONEN T, PETTIT J, *et al*. Nox: towards an operating system for networks[J]. SIGCOMM Computer Communication Review, 2008, 38(3): 105-110.

[6] KOPONEN T, CASADO M, GUDE N, *et al*. Onix: A Distributed Control Platform for Large-scale Production Networks[C]// Proceedings of the 9th conference on Symposium on Operating Systems Design and Implementation (OSDI 2010): October 4-6, 2010, Vancouver, BC, Canada. USENIX Association, 2010.

[7] HELLER B, SHERWOOD R, MCKEOWN N. The controller placement problem[C]// Proceedings of the first workshop on Hot topics in software defined networks (HotSDN 2012): August 13-17 2012, Helsinki, Finland. ACM, 2012: 7-12.

[8] HU Yannan, WANG Wendong, GONG Xiangyang, *et al*. Reliability-aware controller placement for software-defined networks[C]// Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013): May 27-31, 2013, Ghent, Belgium. IEEE Computer Society, 2013: 672-675.

[9] ALBERT R, JEONG H, BARABÁSI A L. Error and attack tolerance of complex networks[J]. Nature, 2000, 406(6794): 378-382.

[10] COLBOURN C J. Reliability issues in telecommunications network planning[M]// Telecommunications network planning. Springer US, 1999: 135-146.

[11] WEICHENBERG G, CHAN V W S, MÉDARD M. High-reliability topological architectures for networks under stress[J]. IEEE Journal on Selected Areas in Communications, 2004, 22(9): 1830-1845.

[12] LI Guangzhi, YATES J, Wang D, *et al*. Control plane design for reliable optical networks[J]. IEEE Communications Magazine, 2002, 40(2): 90-96.

[13] BEHESHTI N, ZHANG Ying. Fast failover for control traffic in Software-defined Networks[C]// Proceedings of the Global Communications Conference (GLOBECOM 2012): December 3-7, 2012, Anaheim, CA, USA. IEEE Computer Society, 2012: 2665-2670.

[14] TOOTOONCHIAN A, GORBUNOV S, GANJALI Y, *et al*. On controller performance in software-defined networks[C]// Proceedings of the 2nd USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE 2012): April 24, 2012, San Jose, CA, USA. USENIX Association, 2010.

[15] CAI Zheng, Cox L A, Ng E S T. Maestro: A System for Scalable OpenFlow Control, Technical Report TR10-11, Department of Computer Science, Rice University, 2010.

[16] TOOTOONCHIAN A, GANJALI Y. HyperFlow: A distributed control plane for OpenFlow[C]// Proceedings of the 2010 internet network management conference on Research on enterprise networking: April 24, 2012, San Jose, CA, USA. USENIX Association, 2010.

[17] KAZEMIAN P, VARGHESE G, MCKEOWN N. Header space analysis: Static checking for networks[C]// Proceedings of the 9th conference on Symposium on Networked Systems Design and Implementation (NSDI 2012): April 25-27, 2012, San Jose, CA, USA. USENIX Association,

2012.
[18] CANINI M, VENZANO D, PERESINI P, *et al*. A NICE way to test OpenFlow applications[C]// Proceedings of the 9th conference on Symposium on Networked Systems Design and Implementation (NSDI 2012): April 25-27, 2012, San Jose, CA, USA. USENIX Association, 2012.
[19] FOSTER N, HARRISON R, FREEDMAN M J, *et al*. Frenetic: A network programming language[J]. ACM SIGPLAN Notices, 2011, 46(9): 279-291.
[20] ZHANG Ying, BEHESHTI N, TATIPAMULA M. On resilience of split-architecture networks[C]// Proceedings of the Global Telecommunications Conference (GLOBECOM 2011): December 5-9, 2011, Houston, Texas, USA. IEEE Computer Society, 2011: 1-6.
[21] LE K V, LI V O K. Modeling and analysis of systems with multimode components and dependent failures[J]. IEEE Transactions on Reliability, 1989, 38(1): 68-75.
[22] LIU Guanglei, JI Chuanyi. Scalability of network-failure resilience: analysis using multi-layer probabilistic graphical models[J]. IEEE/ACM Transactions on Networking, 2009, 17(1): 319-331.
[23] NG T. K-terminal reliability of hierarchical networks[J]. IEEE Transactions on Reliability, 1991, 40(2): 218-225.
[24] SHI J J, FONSEKA J P. Analysis and design of survivable telecommunications networks[J]. IEE Proceedings Communications, 1997, 144(5): 322-330.
[25] MARKOPOULOU A, IANNACCONE G, BHATTACHARYYA S, *et al*. Characterization of failures in an operational IP backbone network[J]. IEEE/ACM Transactions on Networking (TON), 2008, 16(4): 749-762.
[26] ZHOU Dongyun, SUBRAMANIAM S. Survivability in optical networks[J]. IEEE Network, 2000, 14(6): 16-23.
[27] Internet2 open science, scholarship and services exchange[EB/OL]. [2013-09-17]. http://www.internet2.edu/network/ose/.
[28] SPRING N, MAHAJAN R, WETHERALL D. Measuring ISP topologies with Rocketfuel[J]. ACM SIGCOMM Computer Communication Review, 2002, 32(4): 133-145.
[29] IBM ILOG CPLEX Optimization Studio[EB/OL]. [2013-09-17]. http://www-03.ibm.com/software/products/us/en/ibmilogcpleoptistud/.

Biographies

HU Yannan, received the B.S. degree of electronical and information Engineering from Harbin Institute of Technology (HIT), Harbin, China in July 2008. He is currently a Ph.D. candidate at Broadband Network Research Center, State Key Laboratory of Networking and Switching Technology at Beijing University of Posts and Telecommunications. His research interests include next generation Internet, Software-Defined Networking and network optimization.

WANG Wendong, received the B.S. and M.E. degrees, both in computer science, from Beijing University of Posts and Telecommunications (BUPT), Beijing, China in 1985 and 1991, respectively. He is currently a professor of State Key Lab of Networking and Switching Technology at BUPT. His research interests are IP QoS, next generation Internet, and next generation internet services.

GONG Xiangyang, received the B.S. and M.E. degrees, both in computer science, from Xi'an Jiaotong University(XJTU), Xi'an, China in 1992 and 1995, respectively, and the Ph.D. degree in communication and information system in 2011 from Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He is currently a professor of the State Key Lab of Networking and Switching Technology at BUPT. His research interests are IP QoS, network security, advanced switching technologies and next generation Internet.

QUE Xirong, received the B.E. and M.E. degrees, both in computer science, from BUPT, Beijing, China in 1993 and 1998, respectively. She is currently an associate professor of State Key Laboratory of Networking and Switching Technology at Beijing University of Posts and Telecommunications (BUPT). Her research interests include: IP QoS, next generation internet services and next generation Internet.

CHENG Shiduan, professor of State Key Laboratory of Networking and Switching Technology at Beijing University of Posts and Telecommunications (BUPT). Her research interests include: next generation internet, IP QoS, network measurement and wireless network technology.