Towards Optimal Network Planning for Software-Defined Networks

Shih-Chun Lin, *Student Member, IEEE,* Pu Wang, *Member, IEEE,* Ian F. Akyildiz, *Fellow, IEEE,* and Min Luo, *Senior Member, IEEE*

Abstract—Supporting on-line and adaptive traffic engineering in software-defined networks entails fast, robust forwarding of control messages from software-defined switches to the controller(s). In-band control using the existing infrastructure is cost-efficient but imposes a substantial barrier to timely, reliable transmissions of control messages. Moreover, because of the limited computational capability of a single controller, only the use of multiple controllers could ensure the efficacy of control message delivery. This paper investigates optimal network planning that jointly optimizes control traffic balancing and controller placement. We first formulate the optimal network planning as a nonlinear multi-objective optimization framework. The objective is to find the optimal controller placement and control traffic forwarding paths for each switch so that the required controllers and the control traffic delay are minimized. We solve this problem in a timely manner by partitioning the original problem into two sub-problems: multi-controller placement (MCP) and control traffic balancing (CTB) and solving each sub-problem by the proposed fast-convergent algorithms. We further propose an adaptive feedback control that iteratively solves MCP and CTB for dynamic network replanning, subject to the time-varying traffic volume and network topology. Simulations confirm that the proposed control scheme significantly reduces delay and gains throughput with minimum required controllers.

Index Terms—Traffic statistics, optimal network planning, in-band control, randomized rounding, control traffic balancing, software-defined networks.

1 Introduction

An emerging networking paradigm that separates the network control plane from the data forwarding plane is software-defined network (SDN). As a promising paradigm for dramatically improving network resource utilization, simplifying network management, reducing operating costs, and promoting innovation and evolution [1], [2], SDN has shown great potential for data center networks and the next-generation Internet [3], [4]. Recently, it has been extended to support 5G communication networks [5]. The main functions of SDN are (i) to separate the data plane from the control plane and (ii) to introduce novel network control functionalities based on an abstract representation of the network. In current instantiations of SDN, these functions are realized by (i) removing control decisions (e.g., routing) from the hardware (e.g., switches or routers), (ii) enabling programmable flow tables in the hardware through an open, standardized interface (e.g., Openflow [6]), and (iii) using a logically centralized network controller that defines the behavior and operation of the network forwarding-infrastructure.

Through SDN, when a new flow is initiated and no local forwarding policy is defined in the flow table, the switch must forward the first packet of the flow to the controller, which determines an appropriate forwarding path. As a result, the timely and reliably delivery of control messages (e.g., the first packet of every new flow, network traffic statistics, and flow instructions to all switches along the selected path) for each software-defined or Openflow switch largely impacts the efficiency and effectiveness of SDNs. Therefore, creating scalable and efficient SDN solutions by adopting a single controller is challenging. To address such a challenge, the placement of multiple controllers across the entire network can address the performance limitation of a single controller while retaining the benefit of network control centralization. In this case, several fundamental network planning problems have to be solved regarding (1) the minimum number of controllers, (2) their optimal deployment locations, (3) control domain assignments between switches and controllers, and (4) the optimal control traffic forwarding paths between switches and their corresponding controllers. When an in-band control channel [6] is used, both control and data traffic have to share the same forwarding infrastructure, and such a problem becomes even more prominent.

This paper proposes an optimal network planning framework for SDN. In particular, the objective of this framework is to find the optimal controller placement and control traffic forwarding paths for Openflow switches in such a way that the required number of controllers is minimized and at the same time, control traffic delay is shorter than the predefined threshold with a high probability. Towards this objective, we first formulate the optimal planning problem as a nonlinear multi-

- *S. C. Lin and I. F. Akyildiz are with the Broadband Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA (email: slin88@ece.gatech.edu; ian@ece.gatech.edu). P. Wang is with the Department of Electrical Engineering and Computer Science, Wichita State University, Wichita, KS 67260, USA (email: pu.wang@wichita.edu). M. Luo is with the Shannon Lab, Huawei Technologies Co., Ltd. Santa Clara, USA (email: min.ch.luo@huawei.com).*

objective (mixed integer and continuous) optimization in which the number of controllers is an integer-valued objective and the traffic delay belongs to continuous one. To solve such a complex optimization problem in a timely manner, we partition the original problem into two sub-problems: multi-controller placement (MCP) and control traffic balancing (CTB). More specifically, the MCP problem aims to find the minimum number of controllers and their respective deployment locations to cover all switches while the CTB problem aims at finding delay-optimal forwarding paths for control traffic between the switches and the deployed controllers. Finally, this paper proposes a feedback control scheme that iteratively solves the MCP and CTB problems until the desired delay performance, subject to the ever-changing network dynamics, is met.

After partitioning the original problem into the two sub-problems, we first prove that MCP is NP-Complete and show that it belongs to an integer programming (IP). Then, we propose an approximation algorithm via randomized rounding, which yields a feasible optimal solution in at most two iterations on average. Moreover, as the CTB sub-problem belongs to a non-separable nonlinear continuous optimization, its complexity is extremely high as a result of (i) its nonlinearity and (ii) massive variables of link traffic assignments for large-size networks (i.e., combinatorial explosion of flow-to-link traffic assignments). To solve the optimization problem, we first analyze the fundamental structure of CTB by proving its polynomial time complexity (i.e., its *polynomiality* [7]). Specifically, the optimization problem is proven that the problem is a strictly convex problem and that the solution can be approximated by a polynomial-time fast algorithm. Furthermore, motivated by the polynomiality analysis, we develop a polynomial-time approximation algorithm (PTAA) for the CTB problem that yields the optimal solution with fast convergence rate $O(1/c^m)$ with constant $c > 1$ and iteration number m. Such fast convergence is based on the adopted alternating direction method of multipliers (ADMM) [8], an emerging parallel and fast first-order method for solving large-scale convex optimization problems. Finally, we propose an adaptive feedback control scheme that iteratively solves the optimal controller placement (i.e., MCP) problem and the control traffic balancing (i.e., CTB) problem in such a way that the control traffic delay is shorter than the desired threshold with a predefined probability while requiring the deployment of a small number of controllers.

The major contributions of this paper are summarized as follows:

1) It formulates the optimal network planning problem for SDNs with respect to the minimization of required number of controllers as well as the network latency, such as the average network delay and the maximum link delay.
2) It partitions the original problem into two sub-problems, that is, MCP and CTB, and proposes two fast-solving algorithms that obtain the optimal placement and forwarding paths, respectively.
3) It proposes an adaptive feedback control for dynamic network replanning that is subject to the time-varying traffic volume and network topology. This is achieved by iteratively exploiting the fast-convergent algorithms of MCP and CTB.

Simulation results show that with minimum required controllers, the proposed control scheme successfully demonstrates communication efficiency with at least a delay reduction of 90% in the Sprint GIP backbone network [9], which is similar to the benchmark performance. To the best of our knowledge, this work is the first to address the optimal network planning problem in multi-controller SDNs along with the provably fast-convergent algorithms for near optimal solutions.

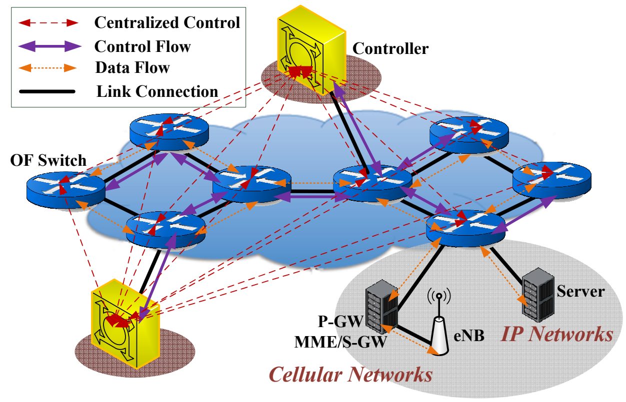
The rest of the paper is organized as follows. Section 2 provides related work, and Section 3 presents the system model. Section 4 concerns the optimal network planning problem. To solve this planning problem, Section 5 proposes the optimal multi-controller placement and Section 6 control traffic balancing, respectively. Section 7 provides performance evaluations, and Section 8 concludes the paper.

2 Related Work

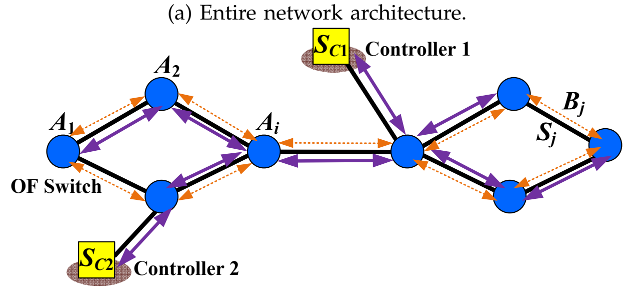
SDNs, which require many signaling events and control plane operations [2], [10], could easily generate a significant amount of control traffic that must be addressed together with data traffic. However, existing work [4], [11]–[13] all focuses on balancing data traffic in the data plane, such as prioritizing interactive, elastic, and background traffic in [4]. In [11], the authors propose an integration of dynamic load balancing, multi-path forwarding, and congestion control with the ability of per-flow and per-packet traffic splitting for data-center networks. A class-based traffic recovery with load balancing in [12] that supports SDN resilience for the automatic reconfigurability of traffic-path failures is introduced. Moreover, an adaptive resource management framework with controller and manager placement in [13] that supports adaptive load balancing and energy management is proposed. Different from data traffic balancing, which aims to evenly distribute data traffic flows among network links, control traffic balancing is much more challenging, particular for in-band control [2], [14]. It aims to find the control message forwarding paths of each switch in such a way that the control message delay can be minimized for the original data traffic, subject to acceptable performance. This control traffic forwarding problem is extremely critical in SDNs, because the timely delivery of control traffic initiated by Openflow switches (i.e., the first packet of every new flow and the traffic/congestion status) directly impacts the effectiveness of routing strategies determined by the controller.

In addition, several recent studies [15]–[22] have focused on the controller placement problem from various planning perspectives. The pioneer work of Heller et al. in [15] first adopts the distance between a controller and switches as the performance metric and evaluates several well-known network topologies to find the optimal controller location. In [16], dynamic controller provisioning (i.e., changing controller placement according to time-varying flows) is addressed through an NP-hard integer-linear programming and two heuristic solving algorithms. In [17], NP-hardness of reliability-aware controller placement is proven, and two heuristic algorithms are examined in terms of the reliability of the control traffic path. In [18], capacitated controller placement (i.e., considering the loads of controllers) is studied to reduce the required number of controllers and the loads of the busiest controllers. Through the non-zero-sum game approach in [19], a placement framework is proposed with controllers' payoff functions to ensure the optimal controller number and their mapping to Openflow switches for cost savings and QoS improvement. Regarding small-scale SDNs for enterprises in [20], a simulation solution that simultaneously determines the optimal number, locations, types of controllers, and the interconnections with switches is provided. Differently, heuristic approaches proposed for large-scale SDNs in [21] provide operators with Pareto-optimal placement, with respect to planning-performance metrics. In sum, the existing efforts only examine heuristic schemes, which cannot yield guaranteed performance bounds.

More importantly, all of these solutions concern controller placement alone without involving control traffic forwarding. Lately, novel work in fast failover [22] considers both resilience-aware controller placement and control traffic routing to improve the resiliency of SDNs. In particular, exhaustive and greedy search algorithms are provided for controller placement; conventional routing tree in which each switch has only one path to the controller is adopted for control traffic forwarding. However, single-controller SDNs with in-band connections is limited, and no load balancing on control traffic exists, losing the capability of multi-path forwarding of control traffic. Different from existing solutions, we jointly investigate multi-controller placement and control traffic balancing for the design of optimal network planning and develop an efficient, adaptive control scheme that guarantees the optimum solution with fast replanning of controller placement and forwarding paths over time-varying QoS requirements, traffic statistics, and network topology in SDNs.



(a) Entire network architecture.



(b) Graph model with overlaid traffic flows.

Fig. 1: Network topology of SDNs.

3 System Model

A SDN generally consists of a number of Openflow enabled switches (i.e., OF-switches) and the centralized SDN controller(s) [6]. Figure 1 shows that each OF-switch forwards the traffic from a variety of networks, such as IP data networks and cellular networks. Whenever a new data flow is generated, the responsible OF-switch sends the routing request to the assigned SDN controller. Once the controller receives the control message, it calculates the optimal routing path(s) to the destination OF-switch and sets up the routing tables of switches along the optimal path(s). Two approaches develop control channels: dedicated out-band control or in-band control. Out-band control uses a dedicated network to establish direct control links between each OF-switch and their responsible controller. Such an approach is cost-prohibitive and not suitable for large-scale networks, such as metropolitan-area-networks that span a city. On the contrary, in-band control allows control and data messages to share the same forwarding infrastructure and thus is more cost-efficient. We consider in-band control throughout this paper.

The SDN is modeled by a graph $G = (V, J)$ as in Figure 1b, where V is the set of OF-switches with total n switches (i.e., $|V| = n$) and J is the set of links with total $|J|$ links. A controller can be placed at any location among all OF-switches. We denote the set of SDN controllers $K \subseteq V$ with a total of C controllers, and the serving time capability of the kth controller is modeled as an exponential distribution with mean time value $1/\mu_C^k$, $\forall k \in K$. Since the effectiveness and scalability of SDNs highly depend on the timely delivery of control messages from OF-switches to multiple controllers, optimal controller placement and control traffic balancing with regard to control and data traffic statistics are the focus of this paper. Note that to simplify the readability, we refer to switches as OF-switches in the remainder of the paper.

Regenerative stochastic processes [23] are generic traffic models that generalize several widely adopted Internet traffic models including alternating renewal processes, recurrent Markov chains, and reflected Brownian motion. Without loss of generality, both control and data flows are modeled by regenerative processes. In particular, the control traffic of each switch i is modeled by a regenerative arrival process A_i with mean value σ_i. For the jth link and $j \in J$, the existing data flow follows a regenerative arrival process B_j with mean value λ_j, and link serving time S_j follows another regenerative process with mean time $1/\mu_j$.

4 Optimal Network Planning Problem

We aim to provide a network planning scheme that determines the optimal placement for multiple controllers and balances link traffic loads according to the control and data traffic dynamics. In the following, we formulate this traffic-driven optimal network planning problem as a nonlinear multi-objective optimization.

4.1 Problem Formulation of Optimal Network Planning

As SDNs provide the centralized control capability with the global view of network status, we address the traffic-driven design of multi-controller planing with two objectives: (i) minimize the required controllers to reduce infrastructure cost via an efficient placement and (ii) minimize the link transmission delay via an optimized traffic scheduling. More specifically, the controller placement over SDNs should determine the number of required controllers, their individual locations, and the control domain assignments for each switch. From Section 3, each switch i's location can be a controller k's location (i.e., $k \in V$). The following variables are defined to address the placement problem. $\{y_k, \forall k \in V\}$ denotes the controller locations as

$$y_k = \begin{cases} 1, & \text{if a controller chooses switch } k\text{'s location;} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

and the number of total controllers C becomes $\sum_{k \in V} y_k$. $\{z_{ik}, \forall i \in V, k \in V\}$ denotes the control domain assignments between switches and controllers as

$$z_{ik} = \begin{cases} 1, & \text{if switch } i \text{ is assigned to controller } k; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Furthermore, to avoid the long-distance assignments for inefficient control message transmissions, indicator variables $\{I_{ik}, \forall i \in V, k \in V\}$ are introduced to enable the localized domain assignments. In particular,

$$I_{ik} = \begin{cases} 1, & \text{if } Distance(i,k) \leq dist; \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $Distance(i,k)$ denotes the distance between switch i and controller k, and $dist$ is the predetermined value by network operators or service providers from their specific concerns or requirements. From these variable definitions, $y_k = 1 - \prod_{i \in V}(1 - z_{ik} I_{ik})$ and

$$y_k \geq z_{ik} I_{ik} \quad \forall\, i \in V, k \in V. \tag{4}$$

To let each switch be assigned to a single dedicated controller, the following constraint is further provided as

$$\sum_{k \in V} z_{ik} I_{ik} = 1 \quad \forall\, i \in V. \tag{5}$$

Moreover, to avoid the infinite delay over controllers' incoming queues, serving capability of each controller should be enough for the arrival control messages of the assigned switches.

$$\sum_{i \in V} \sigma_i z_{ik} I_{ik} < \mu_C^k \quad \forall\, k \in K \subseteq V. \tag{6}$$

Eq. (6) facilitates the workload distribution among controllers and eliminates the queue congestion.

On the other hand, regarding multi-controller scenario, the traffic assignment matrix $\mathbf{x} = [x_{ij}^k]_{i \in V, j \in J, k \in V}$, where x_{ij}^k denotes the amount of control traffic on link j that originates from switch i to controller k, is obtained with respect to minimizing the network delay. To achieve load balancing, multi-path routing is adopted, in which the given P_{ik} is a set of available paths for the ith switch to the kth controller and $i \in V, k \in K$. This means the ith switch can forward the control messages to the kth controller via $|P_{ik}|$ available paths. To characterize possible multi-path routings of control flows, for the flow from the ith switch to the kth controller, we define a topology matric $\mathbf{T}_{ik}$ with size $|J| \times |P_{ik}|$ as follows:

$$\mathbf{T}_{ik}[j,p] = \begin{cases} 1, & \text{if the } j\text{th link lies on the } p\text{th path;} \\ 0, & \text{otherwise.} \end{cases}$$

The matrix $\mathbf{T}_{ik}$ maps the traffic from pathes to links and should always be full column-rank to avoid redundant paths. Its left-inverse matrix $\mathbf{T}_{ik}^{-1} = [t_{i1}^k, ..., t_{i|J|}^k]$ exists and has the size $|P_{ik}| \times |J|$, where t_{ij}^k is the column vector that maps the jth link to all possible paths of the ith switch's flow to the kth controller. t_{ij}^k is obtained by multiplying $\mathbf{T}_{ik}^{-1}$ with the jth standard basis e_j (i.e., $t_{ij}^k = \mathbf{T}_{ik}^{-1} e_j$). While each switch i brings a control flow with the mean value σ_i, the switch where the assigned controller sites can directly send its flow to controller without going through the network (i.e., if $z_{kk} = 1$, then $x_{kj}^k = 0$, $\forall k \in K, j \in J$). We set up the equalities for the control flow conservation of switches as $\|\mathbf{T}_{ik}^{-1}[x_{i1}^k, ..., x_{i|J|}^k]^T\|_1 = \sigma_i z_{ik} I_{ik}$, $\forall k \in K, i \in \widetilde{V} := V \setminus \{z_{kk} = 1, \forall k \in K\}$, where $\|\cdot\|^T$ and $\|\cdot\|_1$ denote the transpose and 1-norm of vector, respectively. Let $d_{ij}^k = \|\mathbf{T}_{ik}^{-1} e_j\|_1$, such equalities can be further simplified as

$$\sum_{j \in J} d_{ij}^k x_{ij}^k = \sigma_i z_{ik} I_{ik} \quad \forall\, i \in \widetilde{V}, k \in K, \tag{7}$$

which is the flow conservation constraint, implying that the control flow initiated by each switch is split into multiple outgoing flows on the selected transmission links. In addition, to balance the traffic loads among all links, every link should have finite transmission delay.

Such finite link delay conditions are equivalent to

$$\sum_{i\in\widetilde{V}}\sum_{k\in K} x_{ij}^k < \mu_j - \lambda_j \quad \forall\ j \in J, \tag{8}$$

which ensure the incoming traffic rates are less than the link service rates and link delays remain nonnegative. Moreover, regarding the second objective of minimizing network delay, two metrics are considered in terms of link's average delay. Specifically, D_{ave} denotes the average network delay and D_{max} denotes the maximum average delay among all links. With the aid of *Little's law* [24], D_{ave} and D_{max} for the control messages are respectively obtained as

$$D_{ave}=\frac{1}{\sum\limits_{i\in\widetilde{V}}\sigma_i+\sum\limits_{j\in J}\lambda_j}\sum_{j\in J}\frac{\sum\limits_{i\in\widetilde{V}}\sum\limits_{k\in K}x_{ij}^k+\lambda_j}{\mu_j-(\sum\limits_{i\in\widetilde{V}}\sum\limits_{k\in K}x_{ij}^k+\lambda_j)}; \tag{9}$$

$$D_{max}=\max_{j\in J}\frac{1}{\mu_j-(\sum\limits_{i\in\widetilde{V}}\sum\limits_{k\in K}x_{ij}^k+\lambda_j)}. \tag{10}$$

In particular Eq. (9) stands because for each link $j \in J$, new packets arrive with rate $(\sum_{i\in\widetilde{V}}\sum_{k\in K}x_{ij}^k+\lambda_j)$ and stay an average time of $1/[\mu_j-(\sum_{i\in\widetilde{V}}\sum_{k\in K}x_{ij}^k+\lambda_j)]$, whose product is the queue backlogs of the link j. Moreover, the total external arrival rate of control and data traffic in the network is $(\sum_{i\in\widetilde{V}}\sigma_i+\sum_{j\in J}\lambda_j)$. Therefore, by little theory, the average network delay in (10) is thus yielded. Alternatively, Eq. (10) simply selects the maximum of average link delay. Therefore, with the above definitions, we define the Optimal Network Planning Problem as follows.

Definition 1: **[Optimal Network Planning Problem.]** Given a SDN modeled by $G = (V, J)$ with multiple controllers $k \in K \subseteq V$, control traffic arrival rates σ_i, a set of topology matrices $\mathbf{T}_{ik}$, $\forall i \in V, k \in K$, data traffic rates λ_j, and link serving rates μ_j, $\forall j \in J$, the network planning optimization problem is

$$\begin{array}{cc} \textbf{Find:} & x_{ij}^k, y_k, z_{ik} \quad \forall\ i \in V, j \in J, k \in V \\ \textbf{Minimize} & C = \sum_{k\in V} y_k \\ \textbf{Minimize} & D_{ave}(x_{ij}^k) \text{ or } D_{max}(x_{ij}^k) \\ \textbf{Subject to} & (4), (5), (6), (7), (8) \end{array}. \tag{11}$$

Note that the planning problem in Eq. (11) belongs to a mixed integer and continuous two-objective optimization, along with several nonlinear constraints and tremendous variables. The entire problem for the optimal values is very complicated to solve in a time-efficient manner (i.e., even finding a feasible solution will require a certain amount of computing time). Therefore, to provide fast solving strategy for this complex framework, we divide the original problem in Eq. (11) into two successive sub-problems as follows:

$$\begin{array}{cc} \multicolumn{2}{c}{\textbf{\textit{Multi-Controller Placement (MCP) Problem}}} \\ \textbf{Find:} & y_k, z_{ik} \quad \forall\ i \in V, k \in V \\ \textbf{Minimize} & C = \sum_{k\in V} y_k \\ \textbf{Subject to} & (4), (5), (6) \end{array}; \tag{12}$$

$$\begin{array}{cc} \multicolumn{2}{c}{\textbf{\textit{Control Traffic Balancing (CTB) Problem}}} \\ \textbf{Find:} & x_{ij}^k \quad \forall\ i \in \widetilde{V}, j \in J, k \in K \\ \textbf{Minimize} & D_{ave}(x_{ij}^k) \text{ or } D_{max}(x_{ij}^k) \\ \textbf{Subject to} & (7), (8) \end{array}. \tag{13}$$

Towards this, MCP problem in Eq. (12) first decides the optimal locations for multiple controllers ; CTB problem in Eq. (13) then examines the traffic scheduling for such a given placement. Moreover, to introduce adaptive feedback control for system QoS performance (e.g., network delay and throughput), the results obtained by executing CTB can further activate another round of MCP for different placement solution. Such iterations will continue until the QoS requirements are fulfilled. In the following, we first apply the randomized rounding technique to solve MCP problem and obtain the placement variables (i.e., y_k and z_{ik}) in Section 5. Next, with regards of the given solution of MCP, the Polynomial-Time Approximation Algorithm (PTAA) are proposed to solve CTB problem for control traffic scheduling parameters (i.e., x_{ij}^k) in Section 6. The design of adaptive feedback control is also included in Section 6.3.

5 Optimal Multi-Controller Placement via Randomized Rounding

In this section, we aim to solve MCP problem and determine the following: (i) the minimum required number of controllers, (ii) the locations of controllers, and (iii) the control domain assignments between switches and controllers. In particular, we first prove that MCP is NP-Complete. Next, by showing that MCP belongs to an integer programming (IP) [25], we propose an approximation algorithm that yields a feasible optimal solution through linear relaxation and randomized rounding techniques.

5.1 NP-Completeness of MCP Problem

The *decision version* [26] of the MCP problem is first introduced in the following Definition 2.

Definition 2 (Decision version of MCP problem): Given a network graph $G = (V, J)$ of SDN, in which switch set V provides a set of potential controller locations, and a positive integer b, the decision version of MCP problem determines whether there is such a subset $K \subseteq V$ with $|K| = b$ that (i) for each controller $k \in K$ its incoming queue has finite delay (i.e., $\sum_{k\in K}\mathcal{U}(\mu_C^k - \sum_{i\in V}\sigma_i z_{ik} I_{ik}) = |K|$ where $\mathcal{U}(\cdot)$ is a step function) and (ii) for each switch $i \in V$ there exists at least one controller $k \in K$ for $z_{ik}I_{ik} = 1$.

The NP-completeness of MCP problem is then provided.

Theorem 1: MCP problem is NP-complete.

Proof: First, we argue that the decision version of MCP problem belongs to NP. Given an instance of MCP, a verification algorithm can effectively check whether each switch has at least one controller in its neighborhood (i.e., within the range *dist* in Eq. (3) of the switch) and whether the minimum number of required controllers is b (i.e., $\sum_{k\in K}\mathcal{U}(\mu_C^k-\sum_{i\in V}\sigma_i z_{ik}I_{ik})=|K|=b$). Thus, MCP$\in$NP.

Next, we show that the Minimum Dominating Set (MDS) problem [26], which belongs to NP-complete, is polynomial time reducible to MCP problem (i.e., MDS$\leq_P$MCP). An instance of MDS is given by a graph $\bar{G}=(\bar{V},\bar{J})$ and a positive integer $b-1$. The objective of MDS is to determine whether there is such a dominating set $\bar{V}'\subseteq\bar{V}$ that $\sum_{k\in\bar{V}'}\mathcal{U}(\mu_C^k-\sum_{i\in V}\sigma_i z_{ik}I_{ik})=b-1$ and each element $i\in\bar{V}$ is a neighbor (i.e., with the range $dist$) of at least one element in $\bar{V}'$. Following the idea of MDS, we construct an instance of MCP problem from the one of MDS as follows. The sets V and J are defined as $V=\bar{V}\cup k'$ and $J=\bar{J}$, where k' is a new controller element with $\mu_C^{k'}>\sum_{i\in V}\sigma_i$ and is within the range $dist$ of at least one element in $\bar{V}$. Then, the instance of MCP is obtained as a graph $G=(V,J)$ and a positive integer b. Furthermore, we prove that the original instance of MDS is a "yes instance" if and only if the created MCP instance is also a "yes instance". On the one hand, suppose the created instance of MCP problem has a solution $K\subseteq V$ with $\sum_{k\in K}\mathcal{U}(\mu_C^k-\sum_{i\in V}\sigma_i z_{ik}I_{ik})=|K|=b$. By our construction, k' is one of the most powerful controllers within certain switches' neighborhoods and thus k' should be included in K. This implies that k' is the element of K that satisfies finite delay condition. While $\bar{V}=V-k'$, the instance of MDS then has a dominating set $\bar{V}'\subseteq\bar{V}$ with $|\bar{V}'|=b-1$. On the other hand, suppose there is a dominating set $\bar{V}'\subseteq\bar{V}$ with $|\bar{V}'|=b-1$ in the original MDS instance. Through the similar arguments, the minimum number of required controllers is b in the constructed MPS instance. Therefore, we have shown that MDS problem can be reduced to MCP problem by the proposed construction. As our construction takes polynomial time and MCP$\in$NP, we can conclude that MCP problem is NP-complete. □

5.2 Linear Relaxation (LP$_{\text{MCP}}$) and Randomized Approximation Algorithm for MCP

Considering MCP problem formulation in (12), the objective function is the number of required controllers in SDNs and three constraint functions are all linear. Thus, this integer programming IP$_{\text{MCP}}$ is also linear. By relaxing variables $y_k, z_{ik}\in\{0,1\}$ to $y_k, z_{ik}\in[0,1]$, we get the relaxed linear programming LP$_{\text{MCP}}$. That is, it follows IP$_{\text{MCP}}$ along with $y_k, z_{ik}\in[0,1]$, $\forall i\in V, k\in V$. The solution of LP$_{\text{MCP}}$ provides the optimal solution of IP$_{\text{MCP}}$.

Given an MCP instance modeled by IP$_{\text{MCP}}$, Algorithm 1 is proposed to solve such integer programming as follows. First, the relaxed linear programming LP$_{\text{MCP}}$ is solved to get an optimal fractional solution (OPT), denoted as y'_k, z'_{ik}, $\forall i\in V, k\in V$. Next, these fractional solutions are rounded to integer values, denoted as $\bar{y}_k, \bar{z}_{ik}$, $\forall i\in V, k\in V$, via a randomized rounding procedure. The rounding procedure consists of two steps: (i) set all $\bar{z}_{ik}$ to zero; then, (ii) let $\bar{z}_{ik}=1$ with probability z'_{ik} and execute this step for $\log(n)+2$ times, where n is the number of switches in the network. Step (ii) yields an integer solution $(\bar{C};\bar{y}_k,\bar{z}_{ik})$, where $\bar{C}=\sum_{k\in V}\bar{y}_k$. To ensure $(\bar{C};\bar{y}_k,\bar{z}_{ik})$ is a feasible solution to IP$_{\text{MCP}}$, Step (ii) is repeated until each controller has finite queue delay and the minimum number of required controllers $\bar{C}$ satisfies the condition that $\bar{C}\leq\alpha C'$, where $C'=\sum_{k\in V}y'_k$ and α is a constant provided in line 9 of Algorithm 1. The result of proposed algorithm for IP$_{\text{MCP}}$ is provided in the Theorem 2.

Algorithm 1: Randomized Rounding for MCP

Input : MCP problem in Eq. (12).
Output: $(\bar{C};\bar{y}_k,\bar{z}_{ik})$ % Optimal controller placement

```
1 Solve LP_MCP. Let (y'_k, z'_ik) be the optimum solution.
2 z̄_ik ← 0, ∀i ∈ V, k ∈ V
3 while t ≤ log(n) + 2 do
4     z̄_ik ← 1 with probability p̄_ik = z'_ik
5     t ← t + 1
6 end
7 repeat
8     line 3-6
9 until Σ_{i∈V} σ_i z̄_ik I_ik < μ_C^k, ∀ k ∈ V and C̄ ≤ αC',
  where y'_max := max_{k∈V} y'_k and α = log_{1/y'_max} 4C';
```

Theorem 2: Let OPT denote the optimal solution of MCP problem. Algorithm 1 yields a solution of $O(\log n)$OPT with high probability.

Proof: As C' is the optimal number of controllers and $\bar{C}$ is the corresponding solution after the randomized rounding, the expected value $\mathrm{E}(\bar{C})$ follows

$$\mathrm{E}\left(\sum_{k\in V}\bar{y}_k\right)=\mathrm{E}\left(\sum_{k\in V}\left[1-\prod_{i\in V}(1-\bar{z}_{ik}I_{ik})\right]\right)$$
$$=\sum_{k\in V}\left[1-\prod_{i\in V}\mathrm{E}(1-\bar{z}_{ik}I_{ik})\right]. \tag{14}$$

The second equality holds from the assumption that the controller selection of each switch is independent to other switches. Given α rounds of randomized rounding, the probability that the controller k is selected by the switch i (i.e., $\bar{z}_{ik}=1$) when the rounding is done is

$$\Pr\left[\bar{z}_{ik}=1\right]=\Pr\left[\bigcup_{t\leq\alpha}z'_{ik}=1 \text{ at round } t\right]$$
$$=1-(1-z'_{ik})^{\alpha}. \tag{15}$$

With Eq. (15) and Eq. (3), we have

$$\mathrm{E}[1-\bar{z}_{ik}I_{ik}]=\Pr\left[\bar{z}_{ik}I_{ik}=0\right]$$
$$=\begin{cases}1 & \text{when } I_{ik}=0\\ (1-z'_{ik})^{\alpha} & \text{when } I_{ik}=1\end{cases}. \tag{16}$$

Therefore, the upper bound of the expected number of controllers is obtained as

$$\mathrm{E}\left(\bar{C}\right)=\sum_{k\in V}\left[1-\prod_{i\in V}(1-z'_{ik}I_{ik})^{\alpha}\right]=\sum_{k\in V}\left[1-(1-y'_k)^{\alpha}\right]$$
$$\leq\sum_{k\in V}\left[1-(1-\alpha y'_k)\right]=\alpha\sum_{k\in V}y'_k=\alpha C'. \tag{17}$$

Next, we derive the probability that the number of controllers is large than $\delta\alpha C'$. Applying Chernoff bound [27], the following inequality is obtained as

$$\Pr\left[\sum_{k\in V}\bar{y}_k\geq\delta\alpha C'\right]\leq\left(\frac{e^{(\delta-1)}}{\delta^{\delta}}\right)^{\alpha C'}. \tag{18}$$

The best polynomial bound that can be achieved via the proposed randomized rounding is then

$$\Pr\left[\sum_{k\in V}\bar{y}_k\geq\delta\alpha C'\right]\leq\frac{e^{(\delta-1)}}{\delta^{\delta}}\leq\frac{1}{n^3}\leq\frac{1}{8}, \tag{19}$$

provided that $\alpha C'\geq 1$ and $\delta=\Theta(\log n/(\log\log n))$. It implies the approximation ratio $\Theta(\log n/(\log\log n))$ is achieved, when we apply at least two rounds for the rounding (i.e., $\alpha\geq 2$). Furthermore, we can achieve the better constant bound by considering the general value δ and applying more rounds. For $\delta>2e$, Eq. (18) becomes

$$\Pr\left[\sum_{k\in V}\bar{y}_k\geq\delta\alpha C'\right]\leq 2^{-(\delta-1)\alpha C'}; \tag{20}$$

for $\delta<2e$, the Chernoff bound is simplified as

$$\Pr\left[\sum_{k\in V}\bar{y}_k\geq\delta\alpha C'\right]\leq e^{-\frac{(\delta-1)^2}{4}\alpha C'}. \tag{21}$$

The value of optimal controller number C' is further considered in the following two cases. For $C'<1/e^2$, we let $\delta=1+1/C'>2e$ and get

$$\Pr\left[\sum_{k\in V}\bar{y}_k\geq\delta\alpha C'\right]\leq 2^{-\alpha}\leq e^{-\frac{\alpha}{2}}<\frac{1}{4}; \tag{22}$$

for $C'>1/e^2$, we select $\delta=1+\sqrt{(2/C')}<2e$ and obtain

$$\Pr\left[\sum_{k\in V}\bar{y}_k\geq\delta\alpha C'\right]\leq e^{-\frac{\alpha}{2}}<\frac{1}{4}. \tag{23}$$

For both above cases, it suggests that $\alpha\geq 4$; thus, the approximation ratio $O(1)$ is reached, when we apply at least four rounds for the rounding.

Finally, we consider the probability that some controllers have infinite queue delay after the randomized rounding. By the fact that a controller's queue delay cannot be infinite if it was not selected by any switch, the probability that a controller $k\in V$ has infinite queue delay at round t can be upper bounded by

$$\Pr\left[\sum_{i\in V}\sigma_i\bar{z}_{ik}I_{ik}>\mu_C^k\text{ at round }t\right]$$
$$=1-\Pr\left[\sum_{i\in V}\sigma_i\bar{z}_{ik}I_{ik}\leq\mu_C^k\text{ at round }t\right]$$
$$\leq 1-\Pr\left[\bar{z}_{ik}I_{ik}=0\ \forall\ i\text{ at round }t\right]. \tag{24}$$

Furthermore, from Eq. (16), the probability that the controller k is not selected by the switch i at round t is

$$\Pr\left[\bar{z}_{ik}I_{ik}=0\text{ at round }t\right]=\begin{cases}1 & \text{when } I_{ik}=0\\ 1-z'_{ik} & \text{when } I_{ik}=1\end{cases} \tag{25}$$

Therefore, the upper bound in Eq. (24) becomes

$$\Pr\left[\sum_{i\in V}\sigma_i\bar{z}_{ik}I_{ik}>\mu_C^k\text{ at round }t\right]\leq 1-\prod_{i\in V}(1-z'_{ik}I_{ik})$$
$$=1-(1-y'_k)=y'_k.$$

This implies the probability that a controller k has infinite queue delay after the rounding is upper bounded by

$$\Pr\left[\sum_{i\in V}\sigma_i\bar{z}_{ik}I_{ik}>\mu_C^k\right]\leq(y'_k)^{\alpha}. \tag{26}$$

By Eq. (26) and union bound [27], the probability that some controllers have infinite queue delay after the rounding is given as

$$\Pr[\text{Some controllers have infinite queue delay}]$$
$$\leq\sum_{k\in V}(y'_k)^{\alpha}\sum_{k\in V}(y'_{max})^{\alpha}=C'(y'_{max})^{\alpha}\leq\frac{1}{4}, \tag{27}$$

where $y'_{max}:=\max_{k\in V}y'_k$ and $\alpha=\log_{1/y'_{max}}4C'$. Combining Eq. (22) and Eq. (23) with Eq. (27), the proposed algorithm yields a solution that is $\delta(\log_{1/y'_{max}}4C')\approx O(\log n)$ times the solution of $\mathrm{LP}_{\mathrm{MCP}}$ with the probability at least $1/2$. That is,

$$\Pr\left[\bar{C}<\delta\alpha C'\wedge\text{Each controller has finite delay}\right]\geq\frac{1}{2} \tag{28}$$

This completes the proof. It is worth to note that both events in Eq. (28) can be verified in the polynomial time. Otherwise, we simply repeat the entire rounding process and the expected number of repetitions is at most two. □

6 Control Traffic Balancing for Multiple Controllers

In this section, we aim to solve the CTB problem based on the results from MCP and provide a coherent solution to traffic-driven network planning. We first prove the *polynomiality* [7] of the CTB problem. Based on this mathematical analysis, we propose the PTAA scheme with the aid of ADMM [8] (a fast first-order method), yielding fast convergence to the optimal solution, and further propose the adaptive feedback control in Section 6.3.

6.1 Polynomiality of CTB Problem

First of all, the minimization objective function in the formulated optimization problem (13) belongs to *a non-separable nonlinear continuous function*. As indicated by [7], to solve such problems, the leading methodology is to develop iterative and numerical algorithms, whose performance are characterized by the convergence rate. Moreover, different from linear problems, for the nonlinear problems, the length of the solution can be infinite (e.g., when a solution is irrational). Hence, the polynomiality of nonlinear optimization problems is determined by the existence of the polynomial-time converged algorithms which can approximate the optimal solution in the solution space.

Definition 3: **[*Polynomiality* of Nonlinear Problem.]** A nonlinear optimization problem is of *polynomiality*, if it has polynomial-time converged algorithm that provides optimal solutions with pre-specified accuracy in the solution space.

Theorem 3: CTB problem in (13) is of *polynomiality*.

Proof: We first prove that both average network delay D_{ave} in Eq. (9) and maximum average delay D_{max} in Eq. (10) are strictly convex. The basic form of D_{ave} is provided as $BD_{ave}(x) = (x+\lambda_j)/[\mu_j-(x+\lambda_j)]$, where μ_j and λ_j are the given constants and x is a single variable. Applying the second derivations to variable x, we obtain the following:

$$\frac{d^2BD_{ave}(x)}{dx^2} = \frac{2\mu_j[\mu_j-(x+\lambda_j)]}{[\mu_j-(x+\lambda_j)]^4}. \tag{29}$$

The convexity of Eq. (29) is determined by the sign of numerator, specifically $\mu_j-(x+\lambda_j)$. Furthermore, as the affine mappings preserve the convexity, we replace x in Eq. (29) with $\sum_{i\in\widetilde{V}}\sum_{k\in K} x_{ij}^k$ and get the multiple-to-one strictly convex function (i.e., $(\sum_{i\in\widetilde{V}}\sum_{k\in K} x_{ij}^k + \lambda_j)/[\mu_j-(\sum_{i\in\widetilde{V}}\sum_{k\in K} x_{ij}^k+\lambda_j)]$), where Eq. (8) guarantees the strictly convexity. Finally, since the summation of strictly convex function is still strictly convex, the average delay D_{ave} is a strictly convex function. Similar arguments are followed for the maximum average delay. In particular, the basic form of D_{max} is provided as $BD_{max}(x) = 1/[\mu_j-(x+\lambda_j)]$, and the corresponding second derivations to variable x is

$$\frac{d^2BD_{max}(x)}{dx^2} = \frac{2}{[\mu_j-(x+\lambda_j)]^3}. \tag{30}$$

With the affine mapping from x to $\sum_{i\in\widetilde{V}}\sum_{k\in K} x_{ij}^k$ and Eq. (8), the multiple-to-one strictly convex function $1/[\mu_j-(\sum_{i\in\widetilde{V}}\sum_{k\in K} x_{ij}^k+\lambda_j)]$ is obtained. Moreover, as the maximum of strictly convex functions is still strictly convex, the maximum delay D_{max} is proven as a strictly convex function. Thus, (13) belongs to a strictly convex optimization framework for either D_{ave} or D_{max} with linear constraint functions.

While the non-separable nonlinear optimization problem is in general hard, CTB problem that belongs to a non-separable convex continuous problem is solvable in polynomial time. In particular, based on the Ellipsoid method [28], a solution approximating the optimal objective value to the convex continuous problem is obtainable in polynomial time, provided that the gradient of the objective functions are available and that the value of the optimal solution is bounded in a certain interval [29]. In other words, any information about the behavior of the objective at the optimum can always be translated to a level of accuracy of the solution vector itself. The interest of solving such optimization problem is thus in terms of the accuracy of the solution rather than the accuracy of the optimal objective value. □

Theorem 3 motivates our following work that approximates the objective value of the balancing problem via a polynomial-time fast algorithm.

6.2 Polynomial-Time Approximation Algorithm (PTAA)

To exploit a fast and possible parallel solving approach for CTB problem with immense variables, we adopt ADMM [8] for the proposed optimization problem with the following two steps. First, we formulate the dual problem from the given primal problem in Eq. (13). Then, we alternatively solve both problems for the optimal solution. Note that while Eq. (13) contains two possible delay objectives, in the following we focus on the derivations for the case of average delay D_{ave}. Similar procedures can be done with maximum delay D_{max}; however, we omit the derivations and only include the results to simplify the readability.

Theorem 4: Base on the results of MCP (i.e., $(\bar{C};\bar{y}_k,\bar{z}_{ik})$), the dual problem of Eq. (13) with D_{ave} is given as follows:

$$\begin{array}{ll}
\textbf{Find:} & x_{ij}^k \text{ and } \beta_{ij}^k \quad \forall\, i\in\widetilde{V}, j\in J, k\in K\\
\textbf{Maximize} & \frac{-1}{\sum\limits_{i\in\widetilde{V}}\sigma_i+\sum\limits_{j\in J}\lambda_j}\sum\limits_{j\in J}\frac{\sum\limits_{i\in\widetilde{V}}\sum\limits_{k\in K}\beta_{ij}^k+\lambda_j}{\mu_j-(\sum\limits_{i\in\widetilde{V}}\sum\limits_{k\in K}\beta_{ij}^k+\lambda_j)}\\
\textbf{Subject to} & \begin{cases} \beta_{ij}^k = x_{ij}^k & \forall\, i\in\widetilde{V}, j\in J, k\in K\\ \sum\limits_{j\in J} d_{ij}^k\beta_{ij}^k = \sigma_i\bar{z}_{ik}I_{ik} & \forall\, i\in\widetilde{V}, k\in K\\ (8) & \end{cases}
\end{array} \tag{31}$$

Proof: A set of auxiliary variables is first introduced as $\boldsymbol{\beta} = [\beta_{ij}^k]_{i\in\widetilde{V},j\in J,k\in K}$ and $\beta_{ij}^k = x_{ij}^k$, $\forall i\in\widetilde{V}, j\in J, k\in K$. Then, the dual problem is obtained from the standard derivations and thus omits here. □

Given Eq. (31) and the *penalty parameter* $\rho>0$ for the *augmented Lagrangian* [8], we consider the update rules for primal variables x_{ij}^k, β_{ij}^k and dual variables γ_{ij}^k, $\forall i\in\widetilde{V}, j\in J, k\in K$. For x-update, the following iteration is obtained:

$$\mathbf{x}^{(m+1)} := \arg\min_{(8)} \frac{\rho}{2}\sum_{i\in\widetilde{V}}\sum_{j\in J}\sum_{k\in K}(x_{ij}^k-\beta_{ij}^{k(m)}+\gamma_{ij}^{k(m)})^2 \tag{32}$$

To simplify Eq. (32), let $\widetilde{n}=n-|\{z_{kk}=1,\forall k\in K\}|$, $\check{x}_j=\sum_{i\in\widetilde{V}}\sum_{k\in K}x_{ij}^k/\widetilde{n}\bar{C}$, $\check{\beta}_j^{(m)}=\sum_{i\in\widetilde{V}}\sum_{k\in K}\beta_{ij}^{k(m)}/\widetilde{n}\bar{C}$, $\check{\gamma}_j^{(m)}=\sum_{i\in\widetilde{V}}\sum_{k\in K}\gamma_{ij}^{k(m)}/\widetilde{n}\bar{C}$. Then, $x_{ij}^k=\beta_{ij}^{k(m)}-\gamma_{ij}^{k(m)}+\check{x}_j-\check{\beta}_j^{(m)}+\check{\gamma}_j^{(m)}$ and the x-**update** of ADMM for the primal problem (13) with D_{ave} and the corresponding dual problem (31) is

$$\begin{array}{ll}\textbf{Find:} & \check{x}_j \quad \forall\ j\in J\\ \textbf{Minimize} & \frac{\widetilde{n}\bar{C}\rho}{2}\sum\limits_{j\in J}(\check{x}_j-\check{\beta}_j^{(m)}+\check{\gamma}_j^{(m)})^2\\ \textbf{Subject to} & \widetilde{n}\bar{C}\check{x}_j<\mu_j-\lambda_j \quad \forall\ j\in J\end{array}. \tag{33}$$

Eq. (33) has $|J|$ single-variable problems and can be independently implemented in parallel for each link j, greatly decreasing the computation complexity.

For β-update, the iteration of $\boldsymbol{\beta}^{(m+1)}$ is

$$\underset{\sum\limits_{j\in J}d_{ij}^k\beta_{ij}^k=\sigma_i\bar{z}_{ik}I_{ik}\ \forall\ i\in\widetilde{V},k\in K}{\arg\min}\frac{\sum_{j\in J}\frac{\sum_{i\in\widetilde{V}}\sum_{k\in K}\beta_{ij}^k+\lambda_j}{\mu_j-(\sum_{i\in\widetilde{V}}\sum_{k\in K}\beta_{ij}^k+\lambda_j)}}{\sum_{i\in\widetilde{V}}\sigma_i+\sum_{j\in J}\lambda_j}$$
$$+\frac{\rho}{2}\sum_{i\in\widetilde{V}}\sum_{j\in J}\sum_{k\in K}(\beta_{ij}^k-x_{ij}^{k(m+1)}-\gamma_{ij}^{k(m)})^2, \tag{34}$$

and $\beta_{ij}^k=x_{ij}^{k(m+1)}+\gamma_{ij}^{k(m)}+\check{\beta}_j-\check{x}_j^{(m+1)}-\check{\gamma}_j^{(m)}$. To rewrite Eq. (34) in terms of $\check{\beta}_j$, $\forall j\in J$ as usual, we deal with the constraint functions by matrix operation and parameter rearrangement as follows. Given the intermediate variable $\hat{\beta}_j=\sum_{i\in\widetilde{V}}\sum_{k\in K}d_{ij}^k\beta_{ij}^k/\widetilde{n}\bar{C}$, the following equations hold

$$\begin{cases}\sum\limits_{i\in\widetilde{V}}\sum\limits_{k\in K}d_{ij}^k\beta_{ij}^k=\widetilde{n}\bar{C}\hat{\beta}_j\\ \sum\limits_{i\in\widetilde{V}}\sum\limits_{k\in K}\beta_{ij}^k=\widetilde{n}\bar{C}\check{\beta}_j\end{cases}. \tag{35}$$

Rearranging Eq. (35), we obtain $\check{\beta}_j=g_j^1\hat{\beta}_j+g_j^2\check{\beta}_j$, where g_j^1 and g_j^2 is from Eq. (36) and $(\cdot)^{\dagger}$ denotes the pseudo-inverse of matrix. The β-**update** of ADMM is then obtained by

$$\begin{array}{ll}\textbf{Find:} & \check{\beta}_j \quad \forall\ j\in J\\ \textbf{Minimize} & \frac{1}{\sum\limits_{i\in\widetilde{V}}\sigma_i+\sum\limits_{j\in J}\lambda_j}\sum\limits_{j\in J}\frac{\widetilde{n}\bar{C}\check{\beta}_j+\lambda_j}{\mu_j-(\widetilde{n}\bar{C}\check{\beta}_j+\lambda_j)}\\ & +\frac{\widetilde{n}\bar{C}\rho}{2}\sum\limits_{j\in J}(\check{\beta}_j-\check{x}_j^{(m+1)}-\check{\gamma}_j^{(m)})^2\\ \textbf{Subject to} & \sum\limits_{j\in J}\frac{1-g_j^2}{g_j^1}\check{\beta}_j=\frac{\sum\limits_{i\in\widetilde{V}}\sum\limits_{k\in K}\sigma_i\bar{z}_{ik}I_{ik}}{\widetilde{n}\bar{C}}\end{array}. \tag{37}$$

Instead of having multiple single-variable problems, (37) is a $|J|$-variables problem due to the coupled constraint function among $\check{\beta}_j$, $\forall j\in J$. However, such a constraint function is simply a linear combination of $|J|$ variables and can be easily solved. Finally, the iteration of **dual-update** of ADMM is obtained:

$$\begin{aligned}\gamma_{ij}^{k(m+1)} &:= \gamma_{ij}^{k(m)}+x_{ij}^{k(m+1)}-\beta_{ij}^{k(m+1)}\\ \Rightarrow\check{\gamma}_j^{(m+1)} &= \check{\gamma}_j^{(m)}+\check{x}_j^{(m+1)}-\check{\beta}_j^{(m+1)}\ \forall\ j\in J.\end{aligned} \tag{38}$$

On the other hand, regarding the maximum average delay D_{max}, the x-**update** of ADMM is given as

$$\begin{array}{ll}\textbf{Find:} & \check{x}_j \quad \forall\ j\in J\\ \textbf{Minimize} & \frac{\widetilde{n}\bar{C}\rho}{2}\sum\limits_{j\in J}(\check{x}_j-\check{\beta}_j^{(m)}+\check{\gamma}_j^{(m)})^2\\ \textbf{Subject to} & \begin{cases}\widetilde{n}\bar{C}\check{x}_j<\mu_j-\lambda_j & \forall\ j\in J\\ \frac{1}{\mu_j-(\widetilde{n}\bar{C}\check{x}_j+\lambda_j)}\le t & \forall\ j\in J\end{cases}\end{array}; \tag{39}$$

the β-**update** is provided as follows:

$$\begin{array}{ll}\textbf{Find:} & \check{\beta}_j \quad \forall\ j\in J\\ \textbf{Minimize} & t+\frac{\widetilde{n}\bar{C}\rho}{2}\sum\limits_{j\in J}(\check{\beta}_j-\check{x}_j^{(m+1)}-\check{\gamma}_j^{(m)})^2\\ \textbf{Subject to} & \sum\limits_{j\in J}\frac{1-g_j^2}{g_j^1}\check{\beta}_j=\frac{\sum\limits_{i\in\widetilde{V}}\sum\limits_{k\in K}\sigma_i\bar{z}_{ik}I_{ik}}{\widetilde{n}\bar{C}}\end{array}. \tag{40}$$

The **dual-update** is the same as Eq. (38).

With the above accomplishments, we propose PTAA in Algorithm 2 through update rules of ADMM to solve CTB problem for multi-controller SDNs. The convergence analysis of proposed PTAA can be further conducted by following the similar arguments in our previous study [14] for single-controller SDNs. In particular, due to the strictly convex framework of CTB problem for multi-controller SDNs, the convergence rate of Algorithm 2 to the optimal solutions is $O(1/c^m)$, where $c>1$ is a constant and m is the number of iterations. Therefore, upon this stage, we have solved CTB problem via the proposed fast and parallel PTAA.

Algorithm 2: Polynomial-Time Approximation Algorithm (PTAA) for CTB

Input : CTB problem in Eq. (13) with D_{ave} or D_{max}.
Output: $x_{ij}^k,\forall\ i\in\widetilde{V},j\in J,k\in K$

1. **Set** $\check{x}_j^{(0)}=0,\check{\beta}_j^{(0)}=0,\check{\gamma}_j^{(0)}=0,\forall\ j\in J$
2. **for** $m=0,1,\ldots$ **do**
3. **Compute** $\check{x}_j^{(m+1)},\forall\ j\in J$ according to Eq. (33) for D_{ave} or according to Eq. (39) for D_{max}
4. **Compute** $\check{\beta}_j^{(m+1)},\forall\ j\in J$ according to Eq. (37) for D_{ave} or according to Eq. (40) for D_{max}
5. **Compute** $\check{\gamma}_j^{(m+1)},\forall\ j\in J$ according to Eq. (38)
6. **Set** $x_{ij}^{k(m+1)}$ from $\check{x}_j^{(m+1)},\forall\ i\in\widetilde{V},j\in J,k\in K$
7. **end**

6.3 Dynamic Replanning via Adaptive Feedback Control

As network size and traffic flow dramatically changes, the controller placement, switch-controller association, and control traffic forwarding paths may need to to replanned. In this section, we assume that the network operator has already deployed servers at particular locations throughout the network. The controllers are running on these servers, which can be in either active mode or inactive mode. A controller is active if at least one

$$\begin{pmatrix} g_j^1 & g_j^2 \end{pmatrix} = \begin{pmatrix} [1\cdots 1] & [1\cdots 1]\cdots & [1\cdots 1] \end{pmatrix} \begin{pmatrix} [d_{1j}^1\cdots d_{1j}^{\bar{C}}] & [d_{2j}^1\cdots d_{2j}^{\bar{C}}]\cdots & [d_{\tilde{n}j}^1\cdots d_{\tilde{n}j}^{\bar{C}}] \\ [1\cdots 1] & [1\cdots 1]\cdots & [1\cdots 1] \end{pmatrix}^{\dagger} \tag{36}$$

OF-switch is assigned to it and the controller is inactive otherwise. Towards this, we aim to propose an adaptive feedback control scheme that adaptively activate the controllers at the optimal locations and select the optimal forwarding paths for control traffic according to time-varying traffic volume.

6.3.1 Statistical delay guarantee

Because of the randomness features of control and data flows, it is more practical to provide *statistical* guarantees (i.e. the probability that the packet violates its delay constraint is bounded) in QoS control over SDNs. That is,

$$\Pr\left[W(t) \geq W^B\right] \leq \tau, \tag{41}$$

where $W(t)$ is the queueing delay, W^B is the requisite bound, and τ characterizes the degree of guarantees. The violation probability can be upper-bounded as $\Pr\{W(t) \geq W^B\} \leq f(W(t), W^B)$. We aim to formulate the function f with respect to the average delay D_{ave} and maximum delay D_{max} from previous sections, and obtain the network throughput of control messages within the statistical delay guarantees. More specifically, this system throughput characterizes the allowable control traffic from switches, which satisfies the delay constraint: $\Pr\left[W(t) \geq W^B\right] \leq f(W(t), W^B) \leq \tau$.

First, regarding the average transmission delay D_{ave}, we formulate the upper-bounded function f by Markov inequality [27]. In particular,

$$\Pr\left[W(t) \geq W^B\right] \leq \frac{E(W(t))}{W^B} = \frac{D_{ave}}{W^B} \leq \tau_{ave}^*. \tag{42}$$

where τ_{ave}^* is the predefined probability threshold. Moreover, by the network model in Section 3, the achievable throughput under QoS guarantee (W^B, τ_{ave}^*) is yielded as $\sum_{i\in V}\sigma_i$.

With regards of the maximum average delay D_{max}, let $W_j(t)$ denote the transmission delay of link $j \in J$ and $W_{max}(t)$ denote the maximum link delay (i.e., $W_{max}(t) = \max_{j\in J} W_j(t)$). Then, this maximum delay can be upper-bounded as

$$\begin{aligned}\Pr\big[W_{max}(t) \geq W_{max}^B\big] &= 1 - \Pr\left[W_1(t) \leq W_{max}^B\right] \times \ldots \\ &\qquad \times \Pr\left[W_{|J|}(t) \leq W_{max}^B\right] \\ &\leq 1 - \left[1 - \frac{E(W_1(t))}{W_{max}^B}\right]\cdots\left[1 - \frac{E(W_{|J|}(t))}{W_{max}^B}\right] \\ &\leq 1 - \left(1 - \frac{D_{max}}{W_{max}^B}\right)^{|J|} \leq \frac{|J| D_{max}}{W_{max}^B} \leq \tau_{max}^*. \end{aligned} \tag{43}$$

where τ_{max}^* is the predefined probability threshold. The first equality comes from the assumption of independence among $W_j, \forall j \in J$; the second inequality follows Markov inequality. Now, the achievable throughput $\sum_{i\in V}\sigma_i$ satisfies the QoS guarantee $(W_{max}^B, \tau_{max}^*)$.

6.3.2 Feedback control loop

Now, we develop the feedback control loop for dynamic network replanning. Informally speaking, the idea is first to derive the optimal multi-controller placement (i.e., activate the controllers at proper servers), then to assign switches to proper controllers, next to optimize the traffic scheduling upon this switch-controller assignment, and finally to active the feedback control loop if the scheduling result doesn't meet the QoS requirements. More specifically, the required QoS guarantee from upper applications gives three specific parameters: the delay metric (i.e., the average packet latency or the maximum link transmission latency), the corresponding requisite delay bound W^B and the degree of guarantee τ. Setting these parameters and the network planning framework as the input, Algorithm 3 iteratively exploits MCP solution in Algorithm 1 and CTB solution in Algorithm 2, until QoS guarantee is fulfilled. In particular, **NSQG** function in line 2 enables the feedback control that gives the true value if the statistical QoS guarantee is not satisfied in the current round. In that case, the algorithm enables the successive round and stops whenever it reaches the optimal solutions. Therefore, the fast-convergent features of proposed algorithms for both MCP problem (i.e., at most two rounds in average) and CTB problem (i.e., $O(1/c^m)$) allow us to reconfigure entire system efficiently according to time-varying traffic statistics and network topology.

Algorithm 3: Adaptive Feedback Control

Input : QoS Guarantee (W^B, τ) and Traffic-Driven Network Planning in Eq. (11)

Output: $\sum_{i\in V}\sigma_i^*; (y_k^*, z_{ik}^*), x_{ij}^{k*}, \forall i \in V, j \in J, k \in K$

1 **Set** $\sigma_i = \infty, \forall\, i \in V; D = \infty$ % Initialization
2 **while** **NSQG**$\big(D, W^B, \tau\big)$ **do**
3 $\quad\|\{\sigma_i\}\|_1 \leftarrow \|\{\sigma_i\}\|_1 - 1$
4 $\quad(\bar{C}; \bar{y}_k, \bar{z}_{ik}) \leftarrow$**Algorithm 1**$(\{\sigma_i\}, \text{Eq. (12)})$ % MCP
5 $\quad(D, x_{ij}^k) \leftarrow$**Algorithm 2**$(\bar{y}_k, \bar{z}_{ik}, \text{Eq. (13)})$ % CTB
6 **end**
7 $\sum_{i\in V}\sigma_i^* = \|\{\sigma_i\}\|_1; \{y_k^*, z_{ik}^*, x_{ij}^{k*}\} \leftarrow \{\bar{y}_k, \bar{z}_{ik}, x_{ij}^k\}$

7 Performance Evaluation

We evaluate the proposed solving algorithms, including Algorithm 1 for the MCP sub-problem, Algorithm 2 for the CTB sub-problem, and Algorithm 3 for the network replanning problem in a practical network scenario: the Sprint GIP backbone network [9]. Specifically, Sprint

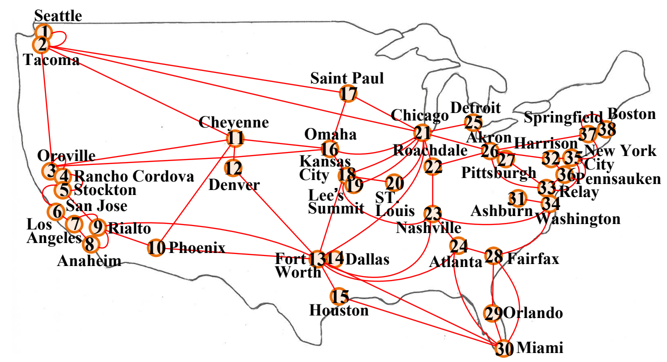


Fig. 2: The Sprint GIP network topology of North America with 38 nodes and 66 links [9].

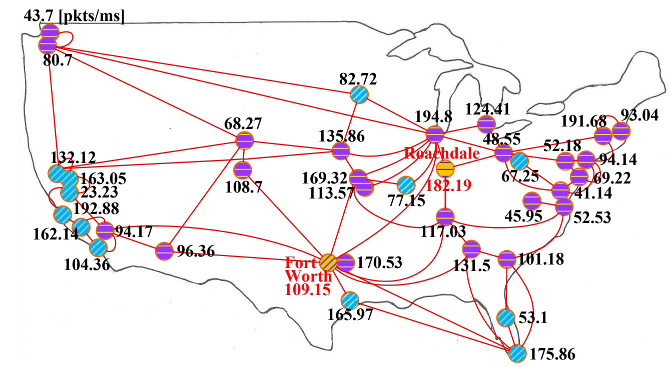


Fig. 3: Optimal MCP in **13 Fort Worth** and **22 Roachdale** and two respective switch groups with controller serving capability $\mu_C^k = \sigma_k + 200 * \text{rand}$ [pkts/ms], $\forall k \in K \subseteq V$ (actual values are listed in the figure).

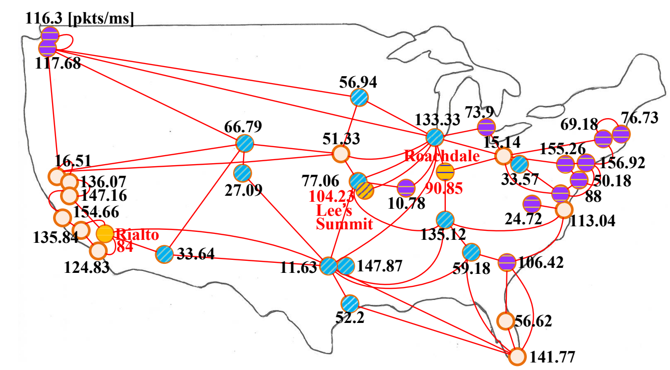


Fig. 4: Optimal MCP in **9 Rialto**, **19 Lee's Summit**, and **22 Roachdale** and three respective switch groups with controller serving capability $\mu_C^k = \sigma_k + 150 * \text{rand}$ [pkts/ms], $\forall k \in K \subseteq V$ (actual values are listed in the figure).

Corporation [9] provides the real backbone network topology and the actual link delay of data traffic. Such delay information is utilized to estimate the corresponding data traffic arrival and serving rates. As shown in Figure 2, the GIP network topology of North America with 38 nodes and 66 links is adopted for our evaluations. Moreover, as our objective is to minimize the control traffic latency (e.g., either average network delay or maximum link delay), we focus on evaluating the former D_{ave} performance while the latter D_{max} can be examined in a similar way. In the following sections, we first separately evaluate the MCP and CTB solutions and then examine the adaptive feedback control.

7.1 Performance for the MCP and CTB

Given control traffic rate σ_i from switches, Figure 3 and Figure 4 show the optimal multi-controller placement in the Sprint GIP network with respect to controller serving capabilities in which rand denotes an uniformly distributed random variable between 0 and 1. Note that to enable localized control domain assignments (i.e., I_{ik} in Eq. (3)), it is assumed that each controller can regulate its three-hop switch neighbors. In particular, with better controller-serving capability, Figure 3 shows that two controllers in Fort Worth and Roachdale are selected with their respective switch groups. Controllers with greater capabilities are not selected in optimal MCP, as not only controller computation capabilities but their topological attributes are concerned with the minimum controller requirement. Moreover, Figure 4 further shows the optimal MCP with less controller serving capability, in which three controllers in Rialto, Lee's Summit, and Roachdale are selected with their respective switch groups. Similarly, these controllers are favored to serve as traffic hubs because of their great serving capabilities and their central locations with many direct links to switches.

According to yielded controller placement by solving the MCP problem, we study the delay performance of control traffic under the proposed control traffic balancing solution shown in Algorithm 2. We compare PTAA with the shortest-path control traffic forwarding [15]. In particular, such a scheme adopts hop-counts as routing metric and employs the shortest path strategy to guide control traffic from a switch to the controller. In addition, the benchmark solution is also implemented, which solves the control traffic balancing problem in Eq. (13) through the brute-force exhaustive search. Moreover, two network delay bounds are adopted from [30]. Specifically, (i) ring protection, 50 [ms], concerns the target restoration time of a ring topology (e.g., SONET ring). It covers the time from fault detection to when flowing traffic in the opposite direction along the ring. (ii) Shared-mesh restoration, around 200 [ms], serves as the point at which voice calls start to drop, or ATM circuit rerouting may be triggered.

Given link serving rate 1000 [pkts/ms], Figure 5 shows the average delay of proposed PTAA solution for control traffic balancing and several possible solutions with respect to control. Specifically, in Figure 5a, PTAA always has lower delay, which is similar to the benchmark, than shortest-path routing. With increasing control traffic from switches, shortest-path routing brings the dramatic delay increase that incurs link overflow;

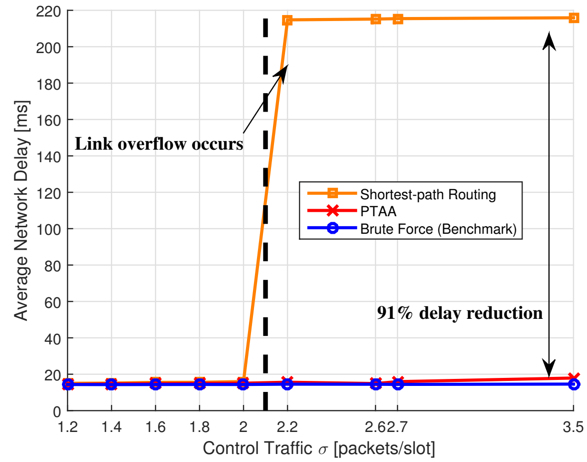
however, our solution can tolerate such higher loads by distributing extra control traffic over links with lighter data loads. Figure 5b further shows that PTAA can maintain network delay lower than the latency of ring protection, and retain the slightly increased delay for increasing traffic. With better link-serving capability (i.e., the serving rate 1200 [pkts/ms]), the results in Figure 6 show that the proposed adaptive control method greatly surpasses shortest-path routing with the delay close to the benchmark. Our solution provides the significant delay reduction, particularly for large control traffic arrivals illustrated in Figure 6a, while shortest-path routing induces link overflow even for limited control traffic volume. Similarly, as shown in Figure 6b, PTAA leads to the delay performance lower than the latency of ring protection. Above observations suggest that by employing information of traffic statistics, our solution brings better controller placement and link resource utilizations and outperforms the existing scheme with at least a delay reduction of 90%.

In addition, to demonstrate the fast convergence of Algorithm 2, Figure 7 shows that the rapid convergence rate of PTAA follows our theoretical finding $O(1/c^m)$ under optimal MCP in Figure 3, where y-axis is our targeted D_{ave} and the penalty parameter ρ in PTAA is set to 10^{-3} after an empirical examination. The results imply that PTAA provides the satisfactory values after 150 iterations, which serve as a desired stopping point. Therefore, yielding optimal controller placement and traffic balancing, the above results confirm that the proposed randomized rounding provides suitable MCP in a timely manner, and the proposed PTAA significantly reduces delay via a fast and parallel computation approach, thus favored in practical implementation in large-scale SDNs.

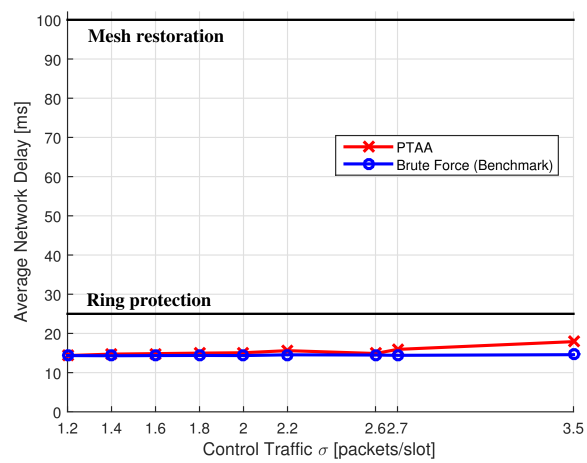
7.2 Dynamic Network Replanning via Adaptive Feedback Control

To evaluate the effectiveness of dynamic replanning with regards of feedback control strategy and QoS provisioning, we examine statistical delay guaranteed throughput via Algorithm 3. In particular, the required delay bound is given as $W^B = 100$ [ms], the delay degree assignment $\tau = 0.16$, and the same controller serving capability in Figure 3. First, as the control traffic rate of each switch is doubled compared with configuration in Figure 3, the replanning of controller placement is executed to cope with the increased control traffic, where two additional controllers are activated at the servers in Chicago and Tacoma, respectively, as shown in Figure 8. More specifically, newly activated controllers, particularly the one in Chicago with great serving capability, considerably share and alleviate heavy traffic loads from two original controllers in Figure 3 (i.e., in Fort Worth and Roachdale). The above observations validate the capability of dynamic replanning subject to the possibly time-varying traffic volume and network topology.

Moreover, we investigate delay-guaranteed throughput, varying the delay-guarantee degree and link serving



(a) Comparisons of control traffic forwarding schemes.



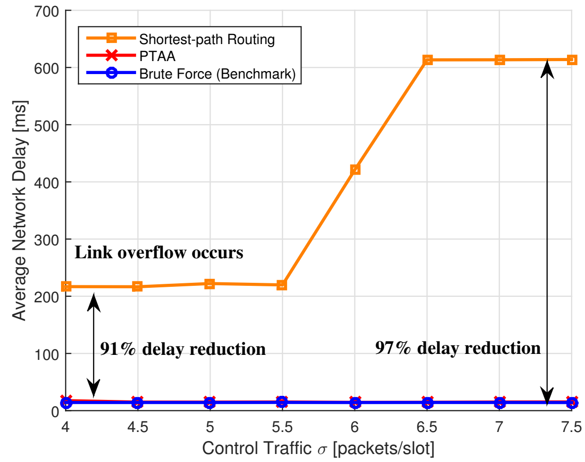
(b) Comparisons of the proposed control with benchmark and bounds relevant to today's networks.

Fig. 5: The average network delay in the Sprint GIP network under optimal MCP in Figure 3 with respect to control traffic arrivals. The link serving rate $\mu_j = 1000$ [pkts/ms], $\forall j \in J$.

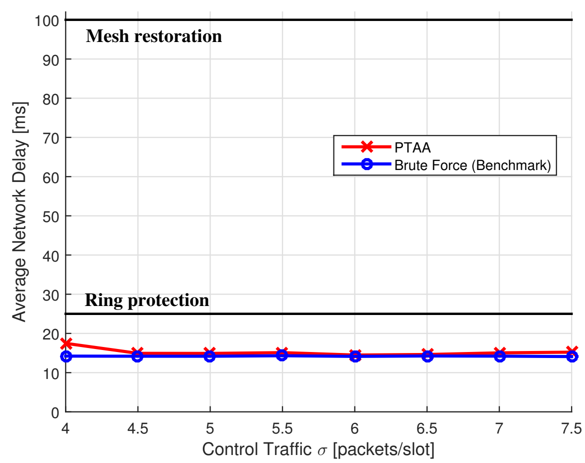
capabilities. It is shown in Figure 9 that as expected, both the loose delay guarantee degree τ and the higher serving capability can induce higher delay-guaranteed throughput. Moreover, with the given $\tau = 0.16$, Figure 10 shows achievable throughput with respect to control traffic arrivals. As traffic arrivals increase, the higher serving capability per controller can lead to a higher control traffic throughput. All of the above evaluations suggest that our solution can instantly react to various network dynamic, including time-varying QoS requirements, network topologies and serving capabilities, and traffic statistics.

8 Conclusion

This study addressed traffic-driven network planning of in-band control traffic as an nonlinear multi-objective (mixed integer and continuous) optimization problem.



(a) Comparisons of control traffic forwarding schemes.



(b) Comparisons of the proposed control with benchmark and bounds relevant to today's networks.

Fig. 6: The average network delay in the Sprint GIP network under optimal MCP in Figure 3 with respect to control traffic arrivals. The link serving rate $\mu_j = 1200$ [pkts/ms], $\forall j \in J$

This complex optimization was solved in a timely manner by partitioning the original problem into two subproblems (the placement and traffic balancing) and iteratively solving the sub-problems through the proposed adaptive feedback control for the global optimal solution. Performance evaluations confirmed that the proposed control scheme, based on the minimum number of required controllers, successfully demonstrated communication efficiency via a fast and low complexity approach with at least a delay reduction of 90%, which was similar to the benchmark of in-band control traffic. Thus, a novel paradigm facilitated on-line configurations of centralized multi-controllers in practical SDN implementations.

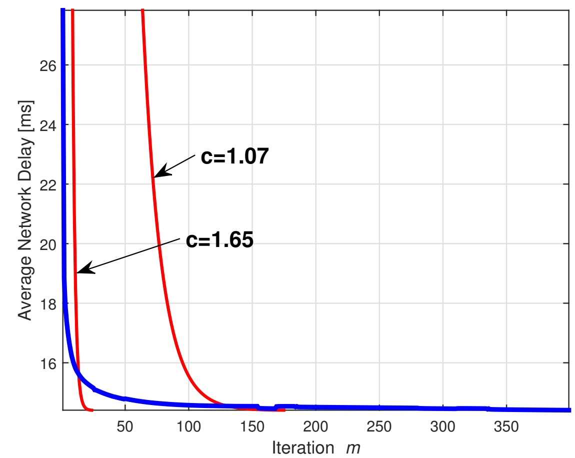


Fig. 7: Linear-fast convergence of Algorithm 2: PTAA with rate $O(1/c^m)$.

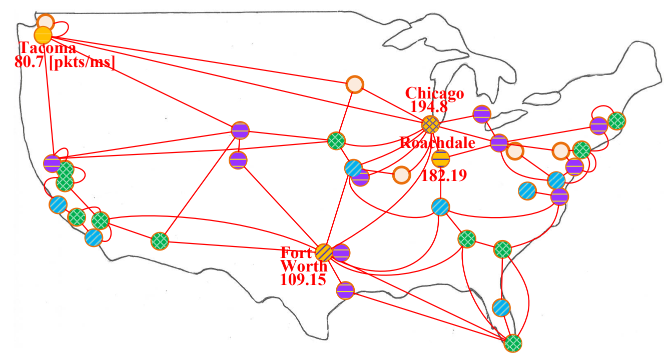


Fig. 8: Dynamic replanning of optimal MCP in **13 Fort Worth**, **22 Roachdale**, **21 Chicago**, and **2 Tacoma** and four respective switch groups under same controller serving capability in Figure 3 but doubled control traffic rates from switches.

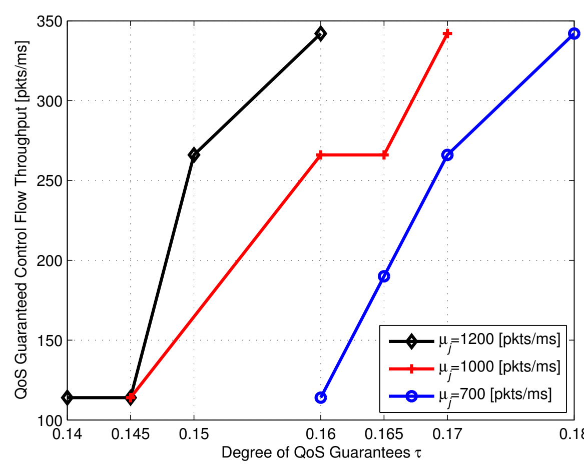


Fig. 9: Statistical QoS guaranteed control flow throughput with respect to the degree of QoS guarantee τ.

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, , and J. Turner, "Openflow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[2] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap

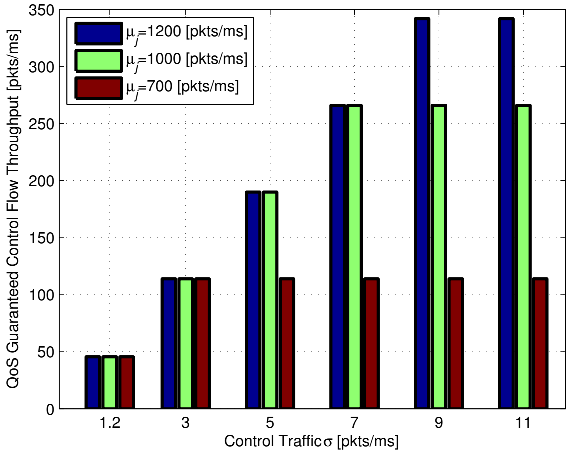


Fig. 10: Statistical QoS guaranteed control flow throughput with respect to control traffic arrivals.

for traffic engineering in sdn-openflow networks," *Computer Networks*, vol. 71, pp. 1–30, Oct. 2014.

[3] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Holzle, S. Stuart, and A. Vahdat, "B4: Experience with a globally-deployed software defined wan," in *ACM SIGCOMM*, Aug. 2013, pp. 3–14.

[4] C. Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven wan," in *ACM SIGCOMM*, Aug. 2013, pp. 15–26.

[5] I. F. Akyildiz, P. Wang, and S. C. Lin, "Softair: A software defined networking architecture for 5g wireless systems," *Computer Networks*, vol. 85, pp. 1–18, July 2015.

[6] ONF, "OpenFlow Switch Specification," version 1.4.0. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

[7] D. S. Hochbaum, "Complexity and algorithms for nonlinear optimization problems," *Annals OR*, vol. 153, no. 1, pp. 257–296, 2007.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[9] Sprint, Overland Park, KS, "Sprint IP network performance," 2011, online available: http://www.sprint.net/performance.

[10] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 136–141, Feb. 2013.

[11] A. Mallik and S. Hegde, "A novel proposal to effectively combine multipath data forwarding for data center networks with congestion control and load balancing using software-defined networking approach," in *International Conference on Recent Trends in Information Technology (ICRTIT)*, Apr. 2014.

[12] D. Adami, S. Giordano, M. Pagano, and N. Santinelli, "Class-based traffic recovery with load balancing in software-defined networks," in *IEEE Globecom Workshops (GC Wkshps)*, Dec. 2014, pp. 161–165.

[13] D. Tuncer, M. Charalambides, S. Clayman, and G. Pavlou, "Adaptive resource management and control in software defined networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 1, pp. 18–33, Mar. 2015.

[14] S.-C. Lin, P. Wang, and M. Luo, "Control traffic balancing in software defined networks," *Computer Networks*, 2015.

[15] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *ACM HotSDN*, Aug. 2012.

[16] M. F. Bari, A. R. Roy, S. R. Chowdhury, Q. Zhang, M. F. Zhani, R. Ahmed, and R. Boutaba, "Dynamic controller provisioning in software defined networks," in *9th International Conference on etwork and Service Management (CNSM)*, Oct. 2013, pp. 18–25.

[17] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, "On reliability-optimized controller placement for software-defined networks," *China Communications*, vol. 11, no. 2, pp. 38–54, Feb. 2014.

[18] G. Yao, J. Bi, Y. Li, and L. Guo, "On the capacitated controller placement problem in software defined networks," *IEEE Commun. Letters*, vol. 18, no. 8, pp. 1339–1342, Aug. 2014.

[19] H. K. Rath, V. Revoori, S. Nadaf, and A. Simha, "Optimal controller placement in software defined networks (sdn) using a non-zero-sum game," in *IEEE 15th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2014, pp. 1–6.

[20] A. Sallahi and M. S.-Hilaire, "Optimal model for the controller placement problem in software defined networks," *IEEE Commun. Letters*, vol. 19, no. 1, pp. 30–33, Jan. 2015.

[21] S. Lange, S. Gebert, T. Zinner, P. T.-Gia, D. Hock, M. Jarschel, and M. Hoffmann, "Heuristic approaches to the controller placement problem in large scale sdn networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 1, pp. 4–17, Mar. 2015.

[22] N. Beheshti and Y. Zhang, "Fast failover for control traffic in software-defined networks," in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2012, pp. 2665–2670.

[23] S. Karlin and H. M. Taylor, *A First Course in Stochastic Processes*. New York: Academic Press, 1975.

[24] D. Gross, J. F. Shortle, and C. M. Harris, *Fundamentals of Queueing Theory*. Wiley-Sons, 4th ed., 2008.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ. Press, 2004.

[26] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms, second edition*. MIT Press, 2001.

[27] A. Gut, *Probability: A graduate course*. Springer Texts in Statistics, Springer-Verlag, 2005.

[28] S. A. Vavasis, *Nonlinear Optimization: Complexity Issues*. New York: Oxford University Press, 1991.

[29] A. S. Nemirovsky and D. B. Yudin, *Problem complexity and method efficiency in optimization*. New York: Wiley, 1983.

[30] Ansi, "Enhanced network survivability performance, Tech. Rep. T1.TR.68-2001.