POLITECNICO DI TORINO
Repository ISTITUZIONALE

The Role of Inter-Controller Traffic in SDN Controllers Placement

Original
The Role of Inter-Controller Traffic in SDN Controllers Placement / Zhang, T., Bianco, A., Giaccone, P.. - ELETTRONICO. - (2016), pp. 87-92. (IEEE Conference on Network Function Virtualization and Software Defined Networks (IEEE NFV-SDN) Palo Alto, CA, USA Nov. 2016) [10.1109/NFV-SDN.2016.7919481].

Availability:
This version is available at: 11583/2649808 since: 2018-02-28T14:36:52Z

Publisher:
IEEE

Published
DOI:10.1109/NFV-SDN.2016.7919481

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

30 November 2026

The Role of Inter-Controller Traffic in SDN Controllers Placement

Tianzhu Zhang, Andrea Bianco, Paolo Giaccone
Dept. Electronics and Telecommunications, Politecnico di Torino, Italy

***Abstract*—We consider a distributed Software Defined Networking (SDN) architecture adopting a cluster of multiple controllers to improve network performance and reliability. Differently from previous work, we focus on the control traffic exchanged among the controllers, in addition to the Openflow control traffic exchanged between controllers and switches. We develop an analytical model to estimate the reaction time perceived at the switches due to the inter-controller communications, based on the data-ownership model adopted in the cluster. We advocate a careful placement of the controllers, taking into account the two above kinds of control traffic. We evaluate, for some real ISP network topologies, the possible delay tradeoffs for the controllers placement problem.**

I. Introduction

The adoption of Software Defined Networking (SDN) paradigm in wide area networks (SDWANs), under a single administrative domain, poses severe technical challenges. Indeed, the centralized control of the network enables the development of complex network applications, but with two main limitations. First, the reliability is limited, due to the single point-of-failure. Second, the control traffic between the switches and the controller concentrates on a single server, whose processing capability is limited, creating scalability issues. Distributed SDN controllers are designed to address the above issues, while preserving a logically centralized view of the network state necessary to ease the development of network applications. In a distributed architecture, multiple controllers are responsible to interact with the switches, with two beneficial effects. First, the processing load at each controller decreases, because the control traffic between the switches and the controllers is distributed across many servers, with a beneficial load balancing effect. Second, resilience mechanisms are implemented to improve network reliability in case of controller failures.

Distributed controllers adopt coordination protocols and algorithms to synchronize their internal states and shared data structures and to enable a centralized view of the network state for the applications. The algorithms follow a consensus-based approach in which some coordination information is exchanged among controllers; thus, controllers reach a common network state only after some interaction. We show in Sec. III that this delay can heavily affect the reactivity perceived at the switches while interacting with the controllers, because any read/write of a shared data structure at the local controller is directed to a centralized "data owner" controller. In this case, the controller-to-controller delays must be added to the switch-to-controller delays when evaluating the latency perceived at the switches. The problem of supporting a responsive controller-to-controller interaction is of paramount importance for SDWANs, due to their geographical extension. Thus, the placement of the controllers must consider not only the delays between the switches and the controllers, but also the delays between controllers. Most of the past literature concentrated on the Openflow-based interaction and thus considered the switch-to-controller delays, and neglected the controller-to-controller delays, which are instead considered in our work.

In our paper, we provide the following novel contributions:

1) we provide some analytical models to evaluate the reaction time perceived at the switches when interacting with the controllers, due to the inter-controller control traffic, and we prove the relevant role of the adopted data-ownership model;
2) we show all the Pareto-optimal placements in terms of controller-to-switch and controller-to-controller delays for some real WAN topologies adopted in some real ISP networks.

In the extended version of our paper, available in [1], we discuss all the related work and validate *experimentally* our proposed analytical models in an operational SDWAN and showed their high accuracy. Furthermore, in [1] we propose a low-complexity algorithm to find the approximated Pareto frontier in large networks.

The paper is organized as follows. In Sec. II we provide an overview of distributed SDN architectures. We describe the interaction in the control plane, highlighting the role of the controller-to-controller communications. In Sec. III we define the data-ownership models and the controller placement problem. We propose an analytical model to evaluate the reaction time for the different data-ownership models. In Sec. IV we present the numerical results obtained by considering realistic ISP topologies. Finally, in Sec. V we draw our conclusions.

II. Distributed SDN controllers

In distributed controllers, two control planes can be identified. First, the switch-to-controller plane, denoted as *Sw-Ctr plane*, supports the interaction between any switch and its controller (denoted as *master controller*) through the controller's "south-bound" interface. This interaction is usually devoted to issue data plane commands (e.g., through the OpenFlow (OF) [2] protocol) and to configure and manage network switches (e.g. through OF-CONFIG or OVSDB protocols). Second, the controller-to-controller plane, denoted as *Ctr-Ctr plane*, permits the direct interaction among the controllers

through the controller's "east-west" interface. Indeed, the controllers exchange heart-beat messages to ensure liveness and to support resilience mechanisms. Controllers need also to *synchronize the shared data structures* to guarantee a consistent global network view.

The traffic in the Sw-Ctr plane heavily depends on the network application running on the controller. For example, for a reactive application, a `packet-in` message with a copy of the first packet of a new flow is sent from the switch to the controller, which replies usually with a `flow-mod` to install a flow-specific forwarding rule. After such reply, the following packets of the same flow can be directly forwarded by the switch to the destined port without interaction with the controller. As a consequence, the reactivity of the controller, defined as the latency perceived by the switch to install the forwarding rule for a new flow, is lower bounded by the round trip time between the switch and its master controller.

A. Data consistency models

The traffic on the Ctr-Ctr plane is instead crucial to achieve a consistent shared view of the network state, which is a required condition to run correctly network applications. The network state is stored in shared data structures (e.g., topology graph, the mapping of the switches to their master controller, the list of installed flow rules), whose consistency across the SDN controllers can be either *strong* or *eventual*. Strong consistency implies that contemporary reads of some data occurring in different controllers always lead to the same result. Eventual consistency implies that contemporary reads may eventually lead to different results, for a transient period. Different levels of data consistency heavily affect the availability and resilience of the controller, as the well-known CAP theorem highlights [3], [4].

In both OpenDaylight (ODL) [5] and Open Network Operating System (ONOS) [6], two of the most relevant SDN controllers, strong consistency for the shared data structures is achieved by the recently proposed Raft consensus algorithm [7]. This algorithm is based on a logically centralized approach, since any data update is always forwarded to the controller defined as *leader* of the data structure. Then, the leader propagates the update to all the other controllers, defined as *followers*. The update is considered committed whenever the majority of the follower controllers acknowledges the update. Note that the role of master/follower controller for some data structure is in general independent of the role of master/slave controller for a switch.

In ONOS data can be also synchronized according to an eventual consistent model, in parallel to strong-consistent data structures. Eventual consistency is achieved through the so called "anti-entropy" algorithm [8] according to which updates are local in the master controller and propagate periodically in the background with a simple gossip approach: each controller picks at random another controller, compares the replica and eventual differences are reconciled based on timestamps.

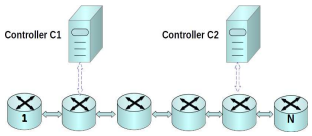


Fig. 1: Placement with minimum Sw-Ctr delay

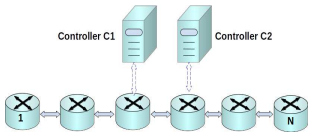


Fig. 2: Placement with minimum Ctr-Ctr delay

III. Data-ownership and reactivity in distributed controllers

The controller reactivity as perceived by a switch depends on the local availability of the data necessary for the controller. We can identity two distinct operative models.

In a *single data-ownership* (SDO) model, a single controller (denoted as "data owner") is responsible for the actual update of the data structure, and any read/write operations on the data structures performed by any controller must be forwarded to the data owner. In this case, the Ctr-Ctr plane plays a crucial role for the interactions occurring in the Sw-Ctr plane, because some Sw-Ctr request messages (e.g., `packet-in`) trigger transactions with the data owner on the Ctr-Ctr plane. Thus, the perceived controller reactivity is also affected by the delay in the Ctr-Ctr plane. As discussed in Sec. II-A, this data-ownership model is currently adopted in ODL and ONOS, for all the strong-consistent data structures managed by Raft algorithm: a local copy of the main data structures is stored at each controller, but any read/write operation is always forwarded to the leader. With this centralized approach, data consistency is easily managed and the distributed nature of the data structures is exploited only during failures.

In a *multiple data-ownership* (MDO) model, each controller has a local copy of the data and can run locally read/write operations. A consensus algorithm distributes local updates to all the other controllers. This model has the advantage of decoupling the interaction in the Sw-Ctr plane from the one occurring in the Ctr-Ctr plane, thus improving the reactivity perceived by the switch. The main disadvantage is the introduction of possible update conflicts that must be solved with ad-hoc solutions and of possible temporary data state inconsistencies leading to network anomalies (e.g. forwarding loops) [4]. Thus, the model applies to generic eventual consistent data structures, as the ones managed by the anti-entropy algorithm in ONOS.

We concentrate our investigation on the delay tradeoff achievable in the Sw-Ctr and in the Ctr-Ctr control planes. For the MDO model, the two planes are decoupled, as shown later in Property 1. Thus, small Sw-Ctr delays imply high reactivity of the controllers (i.e. small reaction time), whereas small Ctr-Ctr delays imply lower probability of network state inconsistency. For the SDO model, Property 2 will show that the Ctr-Ctr delays affect not only the resilience but also the perceived reactivity of the controllers. Thus, reducing Ctr-Ctr delays is important as reducing Sw-Ctr delays; but, for topological reasons, reducing one kind of delays implies maximizing the other, and vice versa. Indeed, consider the toy

scenario depicted in Figs. 1-2, comprising N switches in a linear topology. We assume that each switch selects the closest controller as its master and that the delays between two nodes are directly proportional to their distance in terms of number of hops. We consider two specific controller placements. In Fig. 1, the two controllers are placed to minimize the average Sw-Ctr delay, which is (proportional to) $N/8$. The corresponding Ctr-Ctr delay is $N/2$. Instead, in Fig. 2, the controllers are placed to minimize the Ctr-Ctr delay, which is 1, whereas the Sw-Ctr delay doubles and becomes $N/4$.

We now derive the reactivity for the two data-ownership models. For simplicity, we consider only the propagation delays of the physical links, and neglect all the processing times and the queueing delays due to network congestion.

A. Reactivity model for MDO model

According to the MDO model, a generic event occurring at the switch (e.g. a miss in the flow table) generates a message (e.g., a `packet-in`) to its master controller, which processes the message locally and eventually sends back a control message to the switch (e.g., `flow-mod` or `packet-out` message). In the meanwhile, in an asynchronous way, the master controller advertises the update to all the other controllers. Thus, the reaction time of the controller perceived by the switch, defined as $T_R^{(m)}$, can be evaluated as follows:

Property 1: In a MDO model for distributed SDN controllers, the reaction time perceived at the switch is:

$$T_R^{(m)} = 2d_{\text{sw-ctr}} \tag{1}$$

being $d_{\text{sw-ctr}}$ the delay from the switch to its master controller.

B. Reactivity model for SDO model

In a SDO model, we assume the exchange of messages coherent with the detailed description of Raft algorithm available in [7] and devise a model to evaluate the reactivity of the controller as perceived by the switch. In [1] this analytical model was applied to a specific ODL network application and then experimentally validated in a real SDWAN, showing a high accuracy of the proposed model.

Referring to Fig. 3, the controller reaction time perceived by switch S1 is given by the time between the update event and the response event messages. Assume a cluster of C controllers. Let $d_{\text{sw-ctr}}$ be the communication delay between the switch and its master controller and $d_{\text{ctr-leader}}$ the communication delay from the master controller and the leader (being null whenever the master is also leader). Because of the majority-based selection, let $d_{\text{ctr*-leader}}$ be the communication delay between the leader and the farthest follower belonging to the majority (i.e. corresponding to the $\lfloor (C/2)+1 \rfloor$-th closest follower). Fig. 3 shows the detailed exchange of messages due to Raft consensus algorithm, whose detailed description is available in [1]. According to it, we can claim:

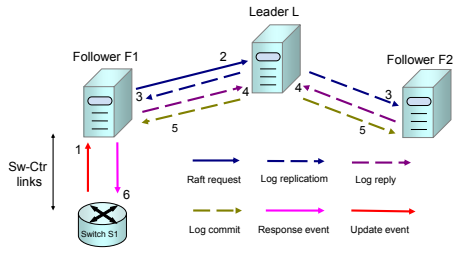


Fig. 3: Control traffic due to SDO model for an update event at the switch.

Property 2: In a SDO model (e.g. adopting Raft consensus algorithm) for distributed SDN controllers, the reaction time $T_R^{(s)}$ perceived at the switch is:

$$T_R^{(s)} = 2d_{\text{sw-ctr}} + 2d_{\text{ctr-leader}} + 2d_{\text{ctr*-leader}} \tag{2}$$

Thus, the reaction time is identical to the one for MDO model plus (roughly) 4 times the RTT between the controllers. Notably, this additional time may be dominant for large networks as SDWANs, as shown experimentally in [1].

C. The controller placement problem

The Sw-Ctr delays (between the switches and their master controller) and Ctr-Ctr delays (between controllers) have a direct impact on the reactivity of the controller perceived at switch level, as highlighted in Properties 1-2. This observation is particularly relevant for large networks, where propagation delays are not negligible. Thus the placement of the controllers in the network is of paramount importance and implies different tradeoffs between Sw-Ctr delays and Ctr-Ctr delays.

Let N be the total number of switches in the network and C be the total number of controllers to place in the topology. The output of any placement algorithm can be represented by the vector denoted as *placement configuration*: $\pi = [\pi_c]_{c=1}^{C}$, where $\pi_c \in \{1,\dots,N\}$ identifies the node at which controller c is associated with. We assume that all the controllers are associated to distinct nodes (equivalently, two controllers cannot be placed in the same node), i.e. $\pi_c \neq \pi_{c'}$ for any $c \neq c'$. Let $\Omega \subset \{1,2,..,N\}^C$ be the set of all placement configurations; thus, the total number of possible placements is $|\Omega| = \binom{N}{C}$.

The optimal controller placement problems consists of finding $\pi \in \Omega$ such that some cost function (e.g. the maximum or average Sw-Ctr delay) is minimized and it is in general a NP-hard problem for a generic graph, as discussed in [9].

IV. Results on the Placement of Controllers in ISP Networks

To explore all the possible tradeoffs on the Sw-Ctr and Ctr-Ctr planes, we adopt an optimal algorithm (denoted Exa-Place) to enumerate exhaustively all possible controller placements and get all *Pareto-optimal* placements[1] and thus the corresponding Pareto-optimal frontier. For small/moderate

[1]When considering two performance metrics x and y to minimize, a solution (x_p, y_p) is Pareto optimal if does not exist any other configuration (x', y') dominating it, i.e. better in terms of both metrics; thus, it cannot be that $x' \leq x_p$ and $y' \leq y_p$. The set of all Pareto-optimal solutions denotes the Pareto-optimal frontier.

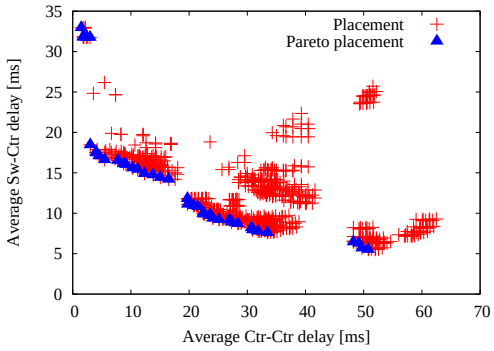


Fig. 4: Delay tradeoffs in HighWinds network

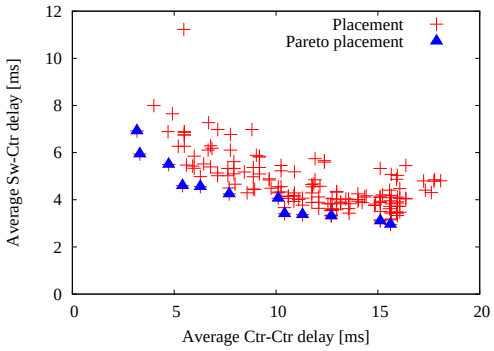


Fig. 5: Delay tradeoffs in Abilene network

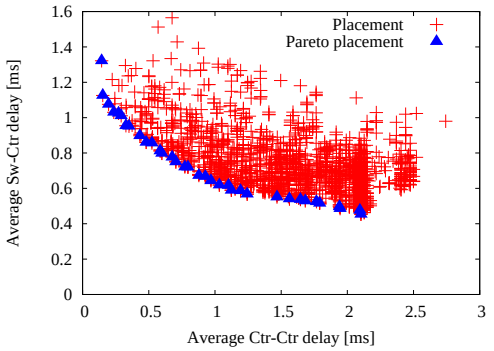


Fig. 6: Delay tradeoffs in York network

values of network nodes N and number of controllers C, as considered in this section, the number of possible placements is not so large and thus EXA-PLACE is computationally feasible. In [1] we propose an approximated algorithm to find the Pareto frontier for large networks and/or large number of controllers.

The network topology is described by a weighted graph where each node represents a switch; each edge represents the physical connection between the corresponding switches and is associated with a latency value. Each controller is connected directly to a switch. We assume that the master controller of a switch is the one with the minimum Sw-Ctr delay. We also assume that all the communications are routed along the shortest path. Coherently with previous work [9], we have considered specifically the topology available in the *Internet topology zoo* website [10]. This repository collects around 250 network topologies of ISPs, at POP level. For each ISP, the repository provides the network graph, with each node (i.e. switch) labeled with its geographical coordinates. From these, we computed the propagation delay between the nodes and associated it as latency of the corresponding edge. For any given controller placement, we compute both the Sw-Ctr delay (as the average delay between the switches and their master controllers) and the Ctr-Ctr delay (as the average delay among controllers).

A. *Tradeoff between Sw-Ctr and Ctr-Ctr delay*

We report only the analysis of three different ISP: (1) HighWinds, a world-wide network with 18 nodes, (2) Abilene, a USA-wide network with 11 nodes, (3) York, a UK-wide network with 23 nodes. Very similar results have been obtained for other topologies.

Figs. 4-6 show the scatter plot with the Sw-Ctr and Ctr-Ctr delays achievable by all possible placements of 3 controllers, for the three ISPs, respectively. In total, all the possible $\binom{18}{3} = 816$, $\binom{11}{3} = 165$ and $\binom{23}{3} = 1771$ different placements are shown; the corresponding Pareto-optimal placements are also highlighted. As observed when discussing the toy example of Figs. 1-2, high (or small) Sw-Ctr delays imply small (or high) Ctr-Ctr delays, respectively. The graphs show the large variety of Pareto-optimal placements. We denote by $P1$ the Pareto point with the minimum Sw-Ctr delay (i.e. the most right-low point), and by $P2$ the one with the minimum Ctr-Ctr delay (i.e. the most left-high point). Table I shows the delay reduction when we compare $P1$ with $P2$ and can be read as follows: if we allow the Sw-Ctr delay to increase by the factor shown in the second column, then the Ctr-Ctr delay decreases by the factor shown in the third column. Notably, in HighWinds if we allow the Sw-Ctr delay to increase by 6.0 times, then the Ctr-Ctr delay decreases by 34.8 times, which is very high gain. Also in York the gain is relevant, since an increase in the Sw-Ctr delay by 2.9 times corresponds to a Ctr-Ctr delay reduction of 15.0 times.

We can generalize these findings: Ctr-Ctr delays corresponding to Pareto points vary much more than Sw-Ctr delays in

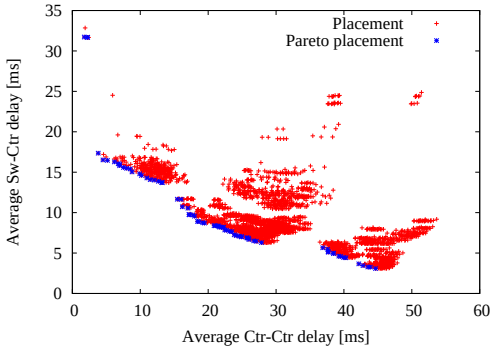


Fig. 7: Delay tradeoffs in HighWinds network with 4 controllers

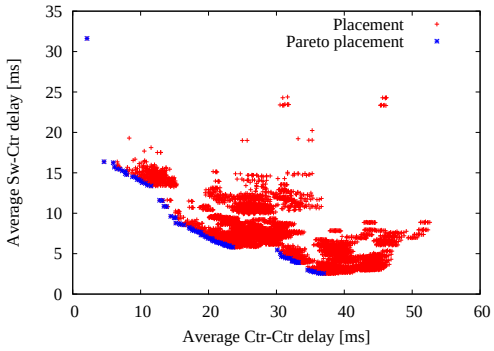


Fig. 8: Delay tradeoffs in HighWinds network with 5 controllers

TABLE I: Delay reductions for the extreme Pareto-optimal placements

ISP	$\frac{\text{Sw-Ctr delay in P2}}{\text{Sw-Ctr delay in P1}}$	$\frac{\text{Ctr-Ctr delay in P1}}{\text{Ctr-Ctr delay in P2}}$
HighWinds	6.0	34.8
Abilene	2.4	4.9
York	2.9	15.0

a generic network. Indeed, Ctr-Ctr delays are by construction between a minimum of 1-2 hops (when all the controllers are at the closest distance) and the maximum equal to the diameter of the network. The gains for the Sw-Ctr delays are lower, since the availability of multiple controllers decreases the maximum distance to reach the controller. We can conclude that larger Sw-Ctr delays with respect to the minimum ones are well compensated by much smaller Ctr-Ctr delays. This highlight the relevant role of the proper design of the Ctr-Ctr plane in SDN networks.

Figs. 7 and 8 show the delay tradeoff achievable for 4 and 5 controllers. Qualitatively the performance confirm our findings above for 3 controllers, even if now the absolute values of the delays for Pareto-optimal points are smaller, due to the larger number of controllers.

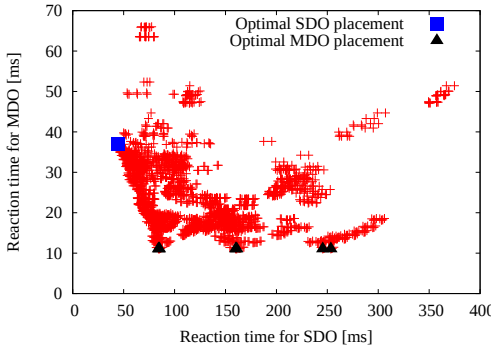


Fig. 9: Average reaction times in HighWinds network for all the placements. Optimal placements for the two data-ownership models are highlighted.

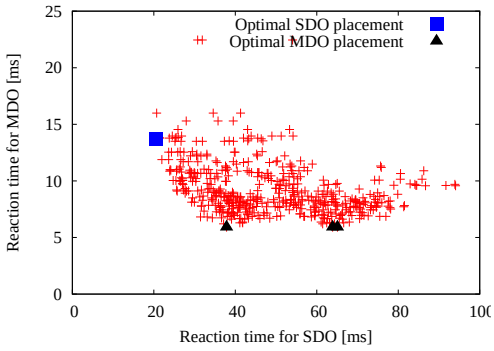


Fig. 10: Average reaction times in Abilene network for all the placements. Optimal placements for the two data-ownership models are highlighted.

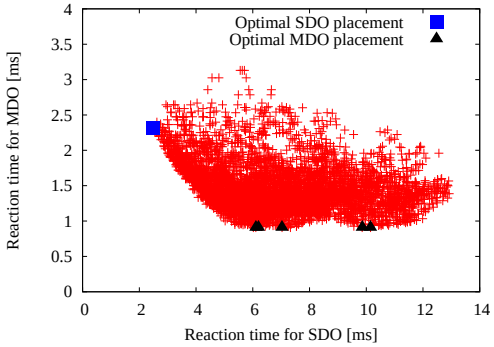


Fig. 11: Average reaction times in York network for all the placements. Optimal placements for the two data-ownership models are highlighted.

B. *Reaction time for SDO and MDO models*

We investigate the reaction times achievable for different data-ownership models, based on Properties 1 and 2. Given a controller placement, we study the effect of selecting the data owner among the controllers on the perceived controller reactivity.

In Figs. 9-11 we report the scatter plots of the average reaction times for the SDO and the MDO models when considering all possible controllers' placements and all possible selections

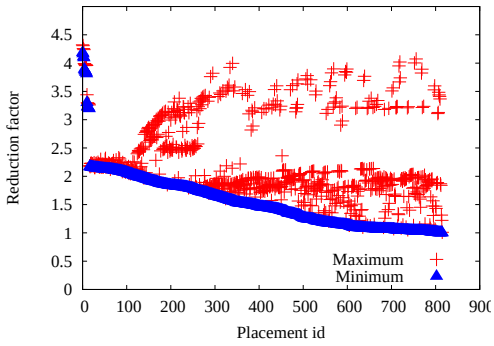


Fig. 12: Reaction time reduction in HighWinds network for the optimal selection of the data owner in the SDO model.

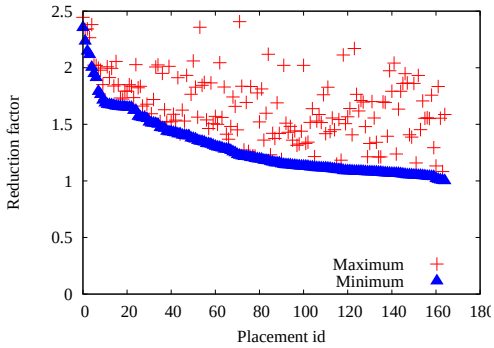


Fig. 13: Reaction time reduction in Abilene network for the optimal selection of the data owner in the SDO model.

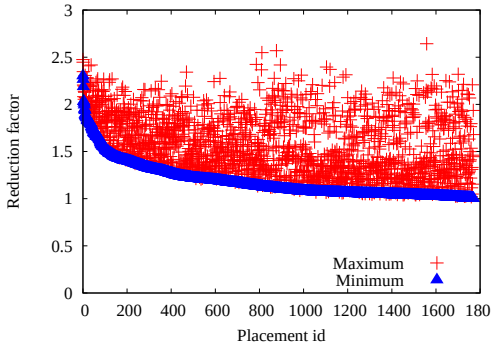


Fig. 14: Reaction time reduction in York network for the optimal selection of the data owner in the SDO model.

for the data owner, in the case of 3 controllers. Each controller placement appears with 3 points aligned horizontally, one for each data owner, since the data owner selection does not affect the MDO reaction time. In the plots we have highlight the placements with the minimum reaction time according to the SDO and MDO models. By construction, the minimum reaction time for the MDO is always smaller than the one for SDO model. From these results, the optimal placements are shown to be very different for the two data-ownership models and this fact motivates the need for a careful choice of the controllers placement and the owner, based on the adopted data-ownership model.

To highlight the role of the proper selection of the data owner for the SDO model, in Figs. 12-14 we investigate the benefit achievable when considering the best data owner among the 3 available controllers, for the three ISPs under consideration. Assume that a given controller placement corresponds to three values of reaction times: d_1, d_2 and d_3, sorted in increasing order. The minimum reduction factor is defined as d_2/d_1 and the maximum reduction factor as d_3/d_1. We plot the delay reduction factor due to the optimal choice of the data owner, for any possible placement. For the sake of readability, the placements have been sorted in decreasing order of minimum reduction factor. Figs. 12-14 show that a careful choice of the data owner in the SDO model decreases the reaction time by a factor around 2 and 4.

These results show that the selection of the data owner in the SDO model has the largest impact on the perceived performance of the controller, and can be easily optimally solved with an exhaustive search, after having fixed the controller placement.

V. Conclusions

We considered a distributed architecture of SDN controllers, with an in-band control plane. We investigated the problem of choosing where to place the controllers across the network nodes. Differently from previous work, we highlighted the importance of the interaction among the controllers in the placement problem. We identified two possible models for the shared data structures: the single and the multiple data-ownership models, which are both implemented in state-of-art controllers. We evaluated analytically the controllers reactivity as perceived by the switches for the two models. We studied the optimal controllers placement problem taking into account all the communications in the control plane (from the switches to the controllers, and among the controllers). We computed the optimal Pareto frontier for some realistic ISP topologies. Based on our numerical results, the choice of the placement of the specific controller with the role of data owner is of paramount importance for the single data-ownership model, since the reactivity of the controller depends heavily on the delay between the controllers and the leader controller.

We believe that our investigation provides a solid methodology to design the network supporting the control plane in large networks, as in the scenario of SDWANs.

References

[1] A. Bianco, P. Giaccone, S. De Domenico, and T. Zhang, "The role of inter-controller traffic for placement of distributed SDN controllers," *CoRR*, vol. abs/1605.09268, 2016. [Online]. Available: http://arxiv.org/abs/1605.09268

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

[3] E. Brewer, "Pushing the CAP: Strategies for consistency and availability," *Computer*, vol. 45, no. 2, pp. 23–29, Feb. 2012.

[4] A. Panda, C. Scott, A. Ghodsi, T. Koponen, and S. Shenker, "CAP for networks," in *HotSDN*, New York, NY, USA, 2013.

[5] OpenDaylight: A Linux foundation collaborative project. [Online]. Available: http://www.opendaylight.org

[6] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an open, distributed SDN OS," in *ACM HotSDN*, New York, NY, USA, 2014.

[7] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annual Technical Conference*, Philadelphia, PA, 2014, pp. 305–320.

[8] A. Muqaddas, A. Bianco, P. Giaccone, and G. Maier, "Inter-controller traffic in ONOS clusters for SDN networks," in *IEEE ICC*, Kuala Lumpur, Malaysia, May 2016.

[9] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *ACM HotSDN*, 2012, pp. 7–12.

[10] "The Internet Topology Zoo." [Online]. Available: http://www.topology-zoo.org/dataset.html