On Multiple Controller Mapping in Software Defined Networks with Resilience Constraints

Article *in* IEEE Communications Letters · August 2017

DOI: 10.1109/LCOMM.2017.2696006

Tram Truong-Huu
National University of Singapore

On Multiple Controller Mapping in Software Defined Networks with Resilience Constraints

Vignesh Sridharan, Mohan Gurusamy, *Senior Member, IEEE*, and Tram Truong-Huu, *Senior Member, IEEE*

***Abstract*—We propose an effective switch-controller mapping scheme for distributed controller architectures in Software Defined Networks. Our scheme maps a switch to multiple controllers and distributes flow setup requests among them to minimize flow setup time, satisfying the resilience constraint which requires that a specified fraction of setup requests at each switch is not affected upon a controller failure. We develop an optimization formulation for the problem and compare our scheme against the single controller mapping. The results show that our scheme reduces flow setup time, provides better fairness among switches and that it is more stable against dynamic traffic fluctuations.**

***Index Terms*—Software Defined Networks, switch-controller mapping, resilience, fairness.**

I. Introduction

SOFTWARE Defined Networking (SDN) has emerged as a promising paradigm in networking whose main principle is the separation of control plane and data plane. SDN leads to the development of a programmable and flexible network, thus increasing the scope for innovation in network management and services provided. The separation between the control and data plane is accomplished by having a centralized controller that is responsible for managing the flow tables at the switches, which contain the forwarding rules for the arriving packets. When a new flow arrives at a switch and no matching forwarding rule is found in the flow table, the switch sends a flow setup request in the form of a packet-in message to the controller. The controller determines the route for the flow and responds back with a packet-out message that contains the forwarding rule, and completes the flow setup process.

However, an SDN architecture with a centralized controller faces multiple issues such as scalability and resilience [1]. The controller can be overwhelmed when it handles a large amount of flow setup requests. The controller also represents a single point of failure in the network. Further, the propagation delay between the switches and the controller in wide area networks is non-negligible, leading to long latency in switch-controller communication. Hence, distributed controller architectures have been proposed to overcome these issues. These are implemented by having physically distributed controllers; each managing several switches [1], [2]. Yang *et al.* also presented a distributed controller architecture in the context of data center inter-connection [3].

Distributed controller architecture introduces an important issue that is the mapping of switches to the controllers. This is a significant problem since the mapping affects the flow setup time of new flows in the network. The two major factors that influence the delay experienced in setting up a new flow are: (i) the response time of the controller to the packet-in message, which in turn depends on the load at the controller and the complexity of the algorithm used to generate the forwarding rules, and (ii) the propagation delay between the switch and controller. Controller load and switch-controller propagation delay depend upon the mapping decision of the switches to the controllers. Thus, an effective mapping scheme is needed to minimize such delay in flow setup.

There are two broad approaches for switch-controller mapping in the context of distributed controller architecture. In the first approach, each switch is mapped to only one of the controllers in the network and all the flow setup requests are sent to that controller. We refer to this approach as single mapping (SM) scheme. Cheng *et al.* [4] adopted this approach and they proposed heuristic algorithms for controller placement and mapping under quality of service constraint, which places an upper bound on flow setup time in the network. Wang *et al.* [5] used game theory to dynamically assign switches to controllers to minimize average response time of the controller in the context of data centers. Gao *et al.* [6] proposed load balancing schemes for devolved controllers in mega data centers. The above works raise the resilience issue since flow setup requests and flow monitoring are affected until a new mapping is done.

The second approach allows each switch to be mapped to multiple controllers. Thus, the flow table is managed by one or more controllers and the flow setup requests are distributed among them. We refer to this approach as multiple mapping (MM) scheme. BalanceFlow [7] and COLBAS [8] adopted this approach and proposed heuristics for controller load balancing with consideration for preventing extreme propagation delay between the switches and controllers. The distribution of flow setup requests among the controllers is done by a coarse-grained approach based on source-destination switch pairs of the flows. We adopt a fine-grained approach where flow setup requests are considered at the individual flow level instead of aggregating flows based on destination. These works also did not model the response time of controllers since the proposed heuristics perform load balancing based on a cost function. Furthermore, they do not consider the resilience aspect, which is a key contribution of our work.

In this work, we develop a stable switch-controller mapping scheme that minimizes overall flow setup time in the network while providing fairness in terms of flow setup time to each individual switch and guaranteeing the resilience constraint. Our contributions are summarized below:

- We propose a fine-grained multiple mapping approach and formulate an optimization problem that minimizes the overall flow setup time in the network considering resilience constraint. The constraint guarantees that a certain percentage of new flows at every switch are not affected by single controller failure. Upon controller failure, the packet-in messages that are sent to the failed

The authors are with the Department of Electrical and Computer Engineering, National University of Singapore. Singapore 117583. E-mail: E0009097@u.nus.edu, elegm@nus.edu.sg, eletht@nus.edu.sg.

controller are not processed and there is a delay in the setup of those new flows until the controller recovery is completed. Along with the introduction of resilience, we mathematically model the response time of the controller using queuing theory, which has not been considered in previous studies in multiple mapping context.

- We also study the stability of the above approach when there is variation in packet-in message rates at the switches. The switch-controller mapping is determined on long term estimation of the packet-in rate at each switch. Frequent re-assignment of switches to other controllers due to short term variation in packet-in message rates is undesirable and it would lead to high overhead in the network as the controller previously monitoring the flows at the switch would need to transfer all necessary information to the new controller. The re-assignment process would also result in increase in controller response time during the handover procedure [9].

We expect that the MM approach will provide better fairness to the switches compared to the SM approach as it can flexibly distribute the load, i.e., the packet-in messages generated at a switch, among the controllers. The MM approach also makes it possible to guarantee resilience for each switch. In the event of a controller failure, the affected switches are still mapped to other controllers and the flow setup requests that have been sent to those controllers are not disrupted. Furthermore, only a portion of monitoring information of the existing flows will be lost. This is different from the SM approach, in which all the flow setup requests from the affected switches are disrupted.

In the remainder of this letter, we first present the proposed mapping scheme and problem formulation in Section II. We present the performance study and discuss the numerical results in Section III before we conclude the letter in Section IV.

II. Proposed Mapping Scheme and Problem Formulation

We assume that the packet-in message rate at each switch is estimated based on observation of the network over a suitable time interval. Based on this information, the MM scheme determines a long term switch-controller mapping and the fraction of packet-in messages to be distributed from each switch to their mapped controllers. We assume that the switch has a limited processing capability [10] and that the packet-in message distribution is done at the switch itself according to the fractional values set by the MM scheme. In case of variation in packet-in message rates, the MM scheme recomputes the switch-controller mapping accordingly. OpenFlow [11] defines controller roles with respect to a switch, which enables the realization of our scheme.

Table I presents all the mathematical notations used in this letter. The locations of the controllers are fixed and predefined. We assume that a path is available for a switch to send a packet-in message to a controller and the path can be computed by using any path selection method such as shortest path selection. We also assume that the propagation delay $D_{i,j}$ of the path between switch i and controller j is given. All the controllers are assumed to have equal processing capacity denoted as C, i.e., the maximum number of packet-in messages that can be processed per second. The packet-in message generation at switch i is assumed to be a Poisson process with rate T_i [4].

TABLE I: Mathematical notations

Notation	Description
N	Number of switches
K	Number of controllers
$D_{i,j}$	Propagation delay between switch i and controller j
C	Processing capacity of each controller
T_i	Packet-in message generation rate at switch i
$S_{i,j}$	Fraction of packet-in messages generated at switch i that are sent to controller j
δ	Resilience constraint
λ_j	Rate of packet-in messages sent to controller j
τ_j	Average response time of controller j

Let $S_{i,j}$ be the fraction of packet-in messages generated at switch i that are sent to controller j. The maximum fraction of packet-in messages that a switch can send to one controller is denoted by δ. Thus, in a scenario where at most one controller in the network fails, at least $(1-\delta)$ fraction of packet-in messages at each switch can be processed without disruption. Thus, the resilience of the mapping scheme is $(1-\delta)$. The maximum resilience achievable is $(1-1/K)$, where K is the number of controllers in the network. Let λ_j denote the arrival rate of packet-in messages that are sent by the switches to controller j. The load at controller j is calculated as $\lambda_j = \sum_{i=1}^{N} T_i S_{i,j}$.

We define τ_j as the average response time of controller j to a packet-in message. We assume that the controller is modeled as an M/M/1 queue [4], [5]. τ_j includes both the waiting time of flow setup requests in the queue and the service time of the controller to run the algorithm for generating forwarding rules. It is computed as $\tau_j = 1/(C-\lambda_j)$.

Our objective is to find the optimal mapping of the switches to the controllers such that the flow setup time in the network with resilience constraint is minimized. The problem formulation is presented as follows:

$$\text{Minimize:} \quad \sum_{j=1}^{K} \lambda_j \tau_j + 2\left(\sum_{j=1}^{K}\sum_{i=1}^{N} T_i S_{i,j} D_{i,j}\right) \tag{1}$$

$$\text{subject to:} \quad \lambda_j < C, \tag{2}$$

$$\sum_{j=1}^{K} S_{i,j} = 1, \quad i = 1 \dots N, \tag{3}$$

$$0 \leqslant S_{i,j} \leqslant \delta, \quad i = 1 \dots N, j = 1 \dots K. \tag{4}$$

Flow setup time for a flow setup request is defined as the sum of average response time at controller and the two-way propagation delay between the switch and controller. The objective function defined in Eq. (1) represents the total time taken to setup all the flow setup requests in the network. Constraint (2) ensures that the controller will not be overloaded. Constraint (3) ensures that all the packet-in messages at every switch are sent to the controllers. Constraint (4) is the resilience constraint that restricts the maximum fraction of packet-in messages that a switch can send to one controller.

III. Performance Study

We now describe the numerical analysis that we conduct to study the performance of the MM scheme. We compare the performance of the MM scheme against the SM scheme based on optimal results. We obtain two sets of numerical results to analyze the performance of the MM scheme. The first set of results provides insight on the performance of the MM scheme with regards to fairness and average flow setup time in the network. The second set of results provides the analysis on the stability of the MM scheme by observing the change in switch-controller mapping in dynamic traffic conditions. We plot the results with 95% confidence interval.

We consider a network with 12 switches and 3 controllers in our study. A random topology is generated with propagation delay between the switches and controllers chosen randomly in the range of $[0.1, 1]$ ms. The processing capacity of each controller is set to 1000 packet-in messages per second. Packet-in message rate at each switch is randomly chosen in the range of $[100, 400]$ packet-in messages per second. Set of packet-in message rates of all the switches defines the traffic matrix. Parameter δ, i.e., the maximum fraction of packet-in messages that a switch can send to a controller, takes a value in the range of $[0.5, 1]$. Hereafter, we use the term "MMδ" to denote the MM scheme with a specific value of δ.

A. *Performance of MM Scheme*

We generate one hundred random traffic matrices. For each traffic matrix, we find the mapping solutions for the SM scheme and the MMδ schemes. The performance metrics considered for comparison are the fairness and average flow setup time. The fairness is represented by the min-max ratio and coefficient of variation (COV). The min-max ratio of a mapping solution is the ratio of the switches with minimum flow setup time and maximum flow setup time. The COV of a mapping solution is the ratio of standard deviation to the mean flow setup time.

The average min-max ratio over the hundred traffic matrices for each mapping scheme is shown in Fig. 1. The MMδ schemes show an improvement of 27% over the SM scheme. This implies that the MMδ schemes perform better when it comes to preventing extreme variation in flow setup times for the individual switches. The average COV over hundred traffic matrices for each mapping scheme is shown in Fig. 2. The results show that deviation from the mean flow setup time in the SM scheme is more than four times that of the MMδ schemes. The results shown in Fig. 1 and Fig. 2 together show that the MMδ schemes achieve better fairness in terms of flow setup time for the switches since the MM approach allows more flexible distribution of load at the controller.

While achieving significant improvement in terms of fairness, we also observe that the MMδ schemes require lower average flow setup time (up to 5%) compared to the SM scheme. The comparable average flow setup time of the SM scheme is achieved at the cost of high imbalance of flow setup times (Fig. 1 and Fig. 2). Also, the SM scheme requires excessive re-mapping of switches to controllers in the event of traffic variations (as shown in the next section).

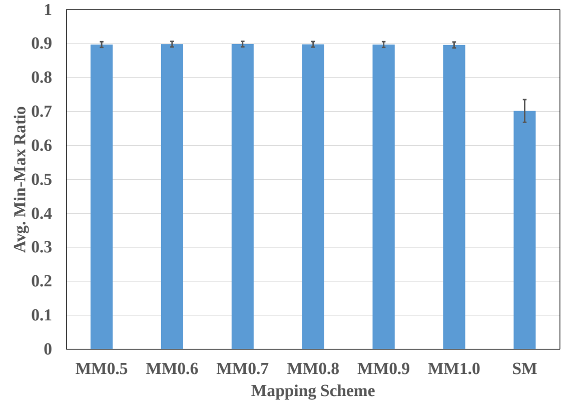

Fig. 1: Average Min-Max ratio.

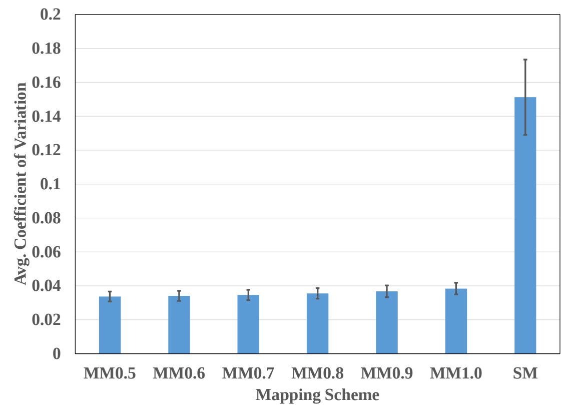

Fig. 2: Average Coefficient of Variation.

B. *Stability In Dynamic Traffic Conditions*

We generate a traffic matrix T that will be used as the long term traffic matrix. The mapping solution with T is computed and denoted as the long term mapping solution. We then randomly vary T to simulate short term variations in traffic and mapping solution is computed for the varied traffic matrix. The varied traffic matrix is obtained by multiplying individual elements of T with randomly sample values in $[-max, max]$ where max is the maximum allowed variation in the packet-in rate at a switch. The value of max is varied from 10% to 50%. For each value of max considered, one hundred randomly varied matrices based on T are generated and the new mapping solution for each traffic matrix is computed. We compare the new mapping solution with the long term mapping solution to observe the number of switches that were assigned to a new controller, i.e, the number of changes in mapping.

The average number of changes in mapping over the hundred varied traffic matrices for each range of variation is shown in Fig. 3. The result shows that SM scheme experiences more than three times the number of changes in mapping on average compared to the MMδ schemes in dynamic traffic conditions. The reason for the stability in the MMδ schemes is that the switches are mapped to multiple controllers based on long term traffic matrix. When there is short term variation in packet-in message rates at the switches, the fraction of packet-in messages that the switches send to the current controllers can be adjusted rather than re-assigning the switches to new controllers. Fig. 4 and Fig. 5 show the average min-max ratio and average COV for the new mapping solutions. The results

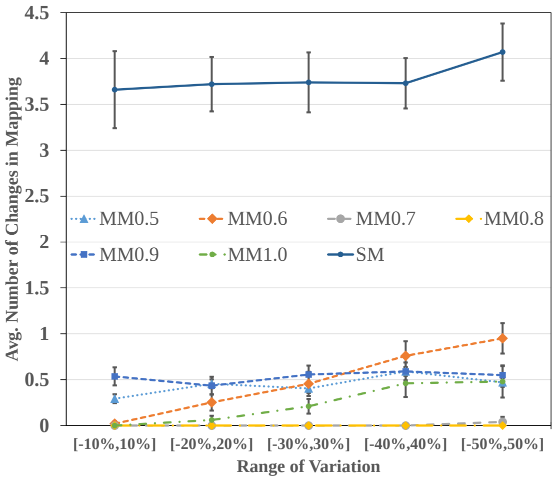

Fig. 3: Average number of changes in mapping.

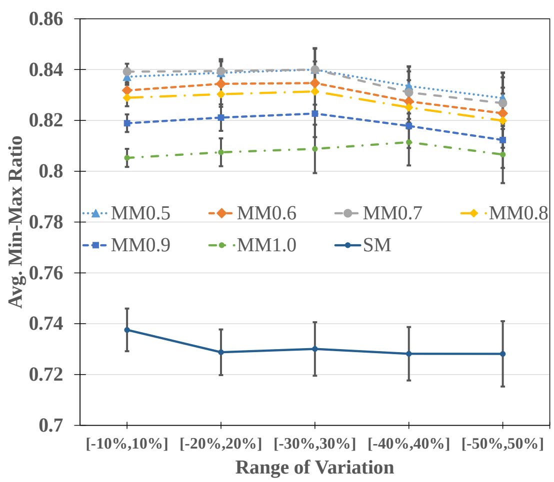

Fig. 4: Average Min-Max ratio.

show that MMδ schemes achieve better fairness compared to the SM scheme. Fig. 6 shows the probability of controller overloading due to variation in traffic matrix if the long term mapping solution is maintained, i.e., the long term mapping is unchanged even if the traffic matrix changes. We observe that the MMδ schemes with higher resilience ($\delta \leqslant 70\%$) have less than 5% chance of controller overloading whereas it is up to 28% for the SM scheme which is quite high.

IV. Conclusion

Switch-controller mapping is an important issue that arises in the study of distributed controller architecture in SDN. We proposed a multiple mapping approach that allows a switch to distribute flow setup requests to multiple controllers. We formulated our objective as an optimization problem that minimizes flow setup time in the network with resilience constraint. We presented numerical results comparing our approach with the single mapping approach. The results show that our approach achieves better fairness, incurs lower flow setup time and provides resilience in event of single controller failure. The results also demonstrate that in dynamic traffic conditions, our approach achieves at least three times greater stability compared to the single mapping approach.

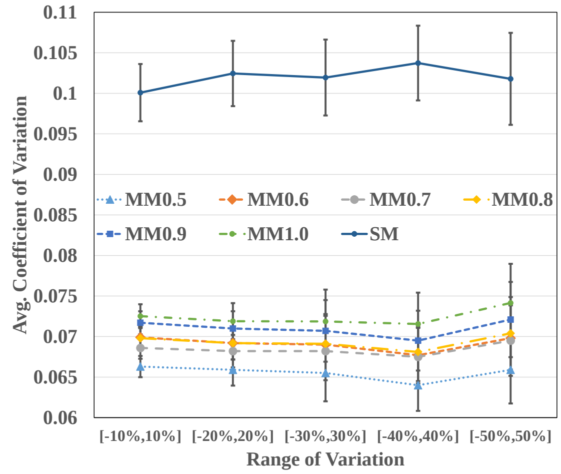

Fig. 5: Average Coefficient of Variation (COV).

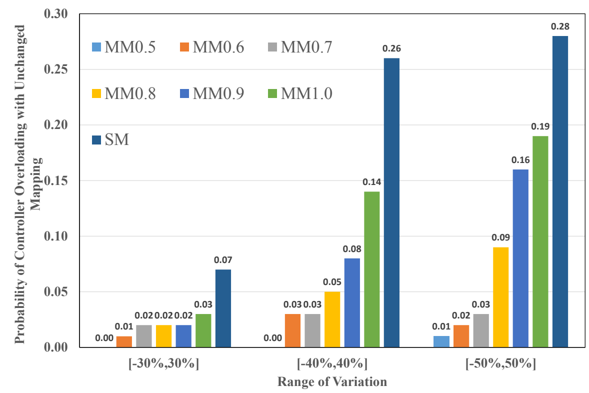

Fig. 6: Overloading probability with unchanged mapping.

Acknowledgment

This work was supported by Singapore MoE AcRF Tier-2 Grant MOE2015-T2-2-116, NUS WBS R-263-000-C17-112.

References

[1] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On Scalability of Software Defined Networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 136–141, 2013.

[2] A. Tootoonchian and Y. Ganjali, "Hyperflow: A Distributed Control Plane for Openflow," in *INM/WREN'10*, San Jose, USA, Apr. 2010.

[3] H. Yang, J. Zhang, Y. Zhao, J. Han, Y. Lin, and Y. Lee, "SUDOI: Software Defined Networking for Ubiquitous Data Center Optical Interconnection," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 86–95, 2016.

[4] T. Y. Cheng, M. Wang, and X. Jia, "QoS-guaranteed Controller Placement in SDN," in *IEEE GLOBECOM 2015*, San Diego, USA, 2015.

[5] T. Wang, F. Liu, J. Guo, and H. Xu, "Dynamic SDN Controller Assignment in Data Center Networks: Stable Matching with Transfers," in *IEEE INFOCOM 2016*, San Francisco, USA, Apr. 2016, pp. 1–9.

[6] X. Gao, L. Kong, W. Li, W. Liang, Y. Chen, and G. Chen, "Traffic Load Balancing Schemes for Devolved Controllers in Mega Data Centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 2, pp. 572–585, 2016.

[7] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, "Balanceflow: controller load balancing for openflow networks," in *IEEE CCIS 2012*, Hangzhou, China, Oct. 2012, pp. 780–785.

[8] H. Selvi, G. Gür, and F. Alagöz, "Cooperative Load Balancing for Hierarchical SDN Controllers," in *IEEE HPSR 2016*, Yokohama, Japan, June 2016, pp. 100–105.

[9] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. Kompella, "Towards an Elastic Distributed SDN Controller," *ACM SIGCOMM CCR*, vol. 43, no. 4, pp. 7–12, 2013.

[10] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "Devoflow: Scaling flow management for high-performance networks," *ACM SIGCOMM CCR*, vol. 41, no. 4, pp. 254–265, 2011.

[11] ONF, "OpenFlow Switch Specification," Open Networking Foundation, Tech. Rep., Oct. 2013.