Loose Management for Multi-controller in SDN

Ligang Dong(✉), Jing Zhou, Tijie Xu, Dandan Yang, Ying Li, and Weiming Wang

School of Information and Electronic Engineering, Zhejiang Gongshang University, No. 18, Xuezheng Street, Xiasha University Town, Hangzhou 310018, China
donglg@zjgsu.edu.cn

Abstract. Centralized network control plane in SDN brings scalability and reliability problem to the network, therefore, the research of multi-controller is appeared. For improving the communication efficiency between the controller and the network device, this paper proposes a loose management strategy to dynamically adjust the frequency of interaction between controllers and network devices. Based on the above idea, firstly, this paper designed the scheme and algorithm of multi-controller loose management. Secondly, this paper quantitatively analyzed the advantages of multi-controller loose management algorithm by mathematically modeling the virtual network deployment success ratio and the management revenue between controllers and network devices. Finally, experiment results show that the multi-controller loose management idea can improve the communication efficiency between the controller and the network device and the controller management efficiency. Simulation results also show that mathematical model accurately predict the performance of loose management algorithm.

Keywords: Distributed control · Multi-controller · Loose management · SDN

1 Introduction

Software Defined Network (SDN) as a new network architecture [1, 2], realizes the centralized, dynamic, and programmable control of the entire network by the virtualization and the separation of application layer, control layer, and data layer.

Like other centralized systems, centralized control in SDN also causes problems of scalability and reliability. Therefore, it is necessary to establish a logical centralized control platform to management the entire network.

In the multi-controller structure of SDN, the controller may not know the status of the network device resources, so a heavy-load network device will probably repeatedly refuse requests from controllers. For improving the communication efficiency between the controller and the network device, this paper proposes a loose management strategy to dynamically adjust the frequency of interaction between controllers and network devices. We consider Virtual Networks (VNs) deployment in SDN as an example. When the number of VNs not deployed by a network device reaches a threshold, the

S. Guo et al. (Eds.): TridentCom 2016, LNICST 177, pp. 3–13, 2017.
DOI: 10.1007/978-3-319-49580-4_1

controller will temporarily stop the communication with the network device. After a period of time, the communication between the controller and the network device is resumed. It will improve the management and communication efficiency between controllers and network devices. That is the first contribution of this paper. The second contribution of paper is mathematically modeling of the Virtual Network (VN) deployment success ratio, and the communication benefits between controllers and network devices. Both of the model and simulation results confirm the advantages of loose management.

The remainder of the paper is organized as follows. Section 2 introduces the related work, including the classification of the multi-controller. Section 3 proposes the scheme and algorithm of loose management. Section 4 evaluates the model using simulations. Finally, Sect. 5 concludes the paper.

2 Related Work

Currently, the implementation for SDN [4] architecture is reliant upon a single controller to push flow rules to all SDN-enabled switches in the network, which creates a performance bottleneck and single point of failure in large networks [5]. Therefore, many scholars have attracted to the research of multi-controller. Multi-controller in SDN can be classified from four viewpoints.

(1) Whole network view controller and local network view controller. The former controllers have a complete information about the entire networks, e.g., HyperFlow [3] and D-ZENIC [7]. While the latter controller have not, e.g., Devolved [8].
(2) Multi-management controller and no multi-management controller. The former means that a single network device may be managed by more than one controller, e.g., Devolved [8], ElastiCon [9], and the literatures [10–12]. The latter refers to that every controller manages part of the network, and a single network device is managed only by one controller, e.g., HyperFlow [3].
(3) Single-level controller and multi-level controller. The latter controllers have a root controller as management operations coordinator of local controllers, e.g., Kandoo [13], D-ZENIC [7]. The former controllers locate on the same level of managing the network devices, e.g., Devolved [8], HyperFlow [3], ONOS [18], and the literature [10, 11].
(4) Static management controller and dynamic management controller. Their difference is whether or not the management relationship between network devices and controllers will change the controller with time on. In other words, a network device probably has different controllers in different situations. The typical examples of the former are Onix [14], HyperFlow [3], the literature [15], while the examples of the latter are literature [12, 16] and ElastiCon [9].

Based on the multi-controller multi-management, this paper proposes the loose management idea to improve communication efficiency between devices and controllers. There are some researches of improving communication efficiency between devices and controllers, e.g., the literature [17].

3 Scheme and Algorithm of Loose Management for Multi-controller

Control plane and data plane are physically separated in SDN network architecture, which makes centralized configuration and management of the network possible. Based on this, we propose a loose management scheme on network device for multi-controller multi-management, as shown in Fig. 1.

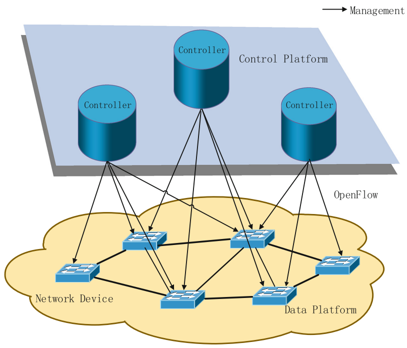

Fig. 1. Multi-controller multi-management

We assume that required resources of deploying a VN is R_{VN}. Here, the "resource" is a generic concept and can be referred to memory, bandwidth, CPU, etc., or the composite of various resource types, which depends on users' applications. We assume that the life cycle of a VN is T, the amount of resources in a network device is R_{sub}, the average time between two adjacent request of deploying a VN is Δ, the VN deployment requests arrive according to a Poisson process. When $(T/\Delta)R_{VN} \leq R_{sub}$, the amount of resources in a network device is adequate to deploy VN. When $(T/\Delta)R_{VN} > R_{sub}$, the amount of resources in a network device is insufficient to deploy VN. The later will cause that the network device is not able to participate in the deployment of VNs, and refuses requests from controllers, which wastes communication and management overheads (including receiving, handling, and replying the request, maintaining the communication state) in both controllers and network devices. Meanwhile, the success ratio of VN deployment is low since more requests are refused.

When the resources of network device are not enough to deploy VNs, the controller will suspend the communication with the network device for some time. When the resources in the network device are released, the controller will restore communication with the network device. Based on the above scheme, we propose an multi-controller loose management algorithm, shown as follows:

Step 1. Network devices send connection requests and establish connections with the controllers.
Step 2. A controller (i.e., any controller in the multi-controller structure) waits for network application requests.
Step 3. The controller selects appropriate network devices to deploy the VN, and sends the deployment message to related network devices.
Step 4. If each network device deploys the VN successfully, then go to Step 2, otherwise, the network device of deployment failure sends a failure message to the controller.
Step 5. The controller suspend to communicate with the network device, in which the number of deployment failure reach r (r is a positive integer). (After a preset period of time, the controller restores communication with the network device. The number of failures is reset to zero.) Go to Step 3.

Fig. 2. Multi-controller loose management algorithm.

4 Analysis of the Deployment Success Ratio and the Loose Management Revenue

We use two metrics to measure the improvement effects of the strategy of loose management. The first one is the deployment success ratio of VNs, which is defined as the ratio of the number of successful VNs deployment on a network device and the number of VNs deployment request on the network device. The second one is the net revenue of deploying a VN, which is defined as the difference between the revenue of a successful deployment and the cost of communication.

In this section, firstly, we conduct simulations to compare loose with non-loose management algorithms in terms of the above two metrics. Secondly, in order to better predict the performance of loose management algorithm, we establish the mathematical model and verified it by simulations.

The independent and dependent variables used in this section are defined in Tables 1 and 2 respectively.

Table 1. Independent variables

Parameters	Definition
R_{sub}	The resource capacity of a network device
R_{VN}	The resource requirement for deploying a VN
λ	The number of VNs deployment requests per unit time
r	The threshold number of VNs that the network device doesn't participate in before the communication is suspended
t_1	The duration of communication suspension
T	The lifecycle of VN
x	The communication cost of a VN deployment
s	The net income of deploying a VN
M	The total number of requests for deploying VNs

Table 2. Dependent Variables

Parameters	Definitions
m_0	The average number of VNs that one network device can participate in in unit time
y	The proportion of communication time in unit time
t_2	The average duration of a communication cycle
R_0	The net income of VN deployment in unit time
η	The success ratio of VN deployment requests
R_{ev}	The total net income of VN deployment

4.1 Comparison Between the Loose and Non-loose Management Algorithms

Based on the algorithm in Fig. 2, we use discrete event simulation to simulate multiple controllers communication with a single network device. It is worth explaining that our simulation scenario can represent the general case containing multi-controllers and multiple network devices, as every network device is independent. Our simulation platform is Eclipse IDE for C/C ++ Developers. The simulation of VNs request generated using a Poisson process.

By default, the number of VNs deployment requests per unit time is 0.04. The life cycle of each VN request is distributed with a mean of $T = 1000$ exponential distribution; the resource requirement for deploying a VN obeys [0, 25] uniform distribution; the resource capacity of a network device is 100; the total number of requests for deploying VNs is 2000.

During the experiment we generate VN deployment requests in accordance with the above parameters configuration. We conducted simulation experiments to compare the non-loose and the loose management algorithm. The simulation process of non-loose management algorithm is shown in Fig. 3 below. The simulation process of loose management algorithm is shown in Fig. 4 below.

In the simulation, default parameters are: $\mathrm{r} = 3$, $t_1 = 300$, $\mathrm{T} = 1000$, $\lambda = 0.04$, $\mathrm{M} = 2000$.

Figures 5 and 6 show the performance of η and R_{ev} with the change of λ (from 0.04 to 0.08), respectively.

Figures 7 and 8 show the performance of η and R_{ev} with the change of T (from 500 to 2000), respectively.

Figures 9 and 10 show the performance of η and R_{ev} with the change of M (from 500 to 2500), respectively.

From the simulation results, we concluded that:

(a) Compared with the non-loose management algorithm, the loose management algorithm has higher success ratio of deployment of VN requests and higher net income of VN deployment. The simulation result is consistent with the analysis in Sect. 3.

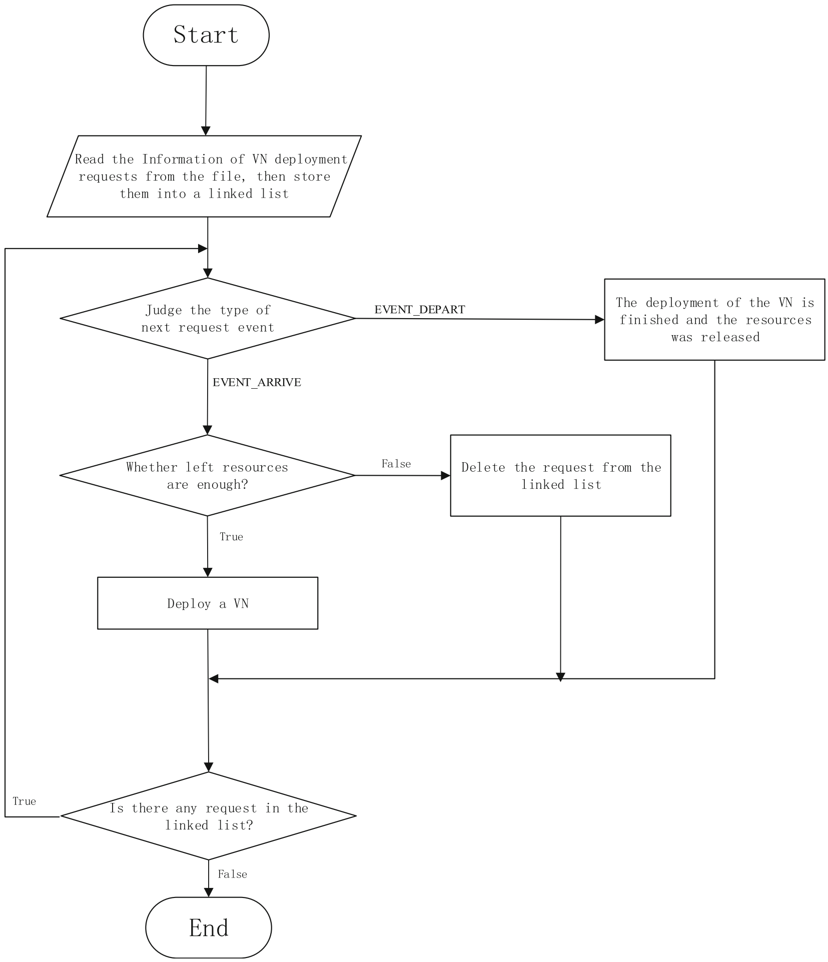

Fig. 3. Non-loose management on network devices

(b) The more number of VNs deployment requests per unit time causes the more number of VNs deployment that the network device doesn't participate in because of limited network device resources, so that the net income is lower.
(c) The longer life cycle of VNs means the longer occupation of network device resources by the VN. It causes the network device participate in a less number of VNs deployment, so that the net income of VN deployment is lower.
(d) The more number of VNs deployment requests causes the more net income of VN deployment. The success ratio of deployment of VN requests have little change vary with the number of VNs deployment requests.

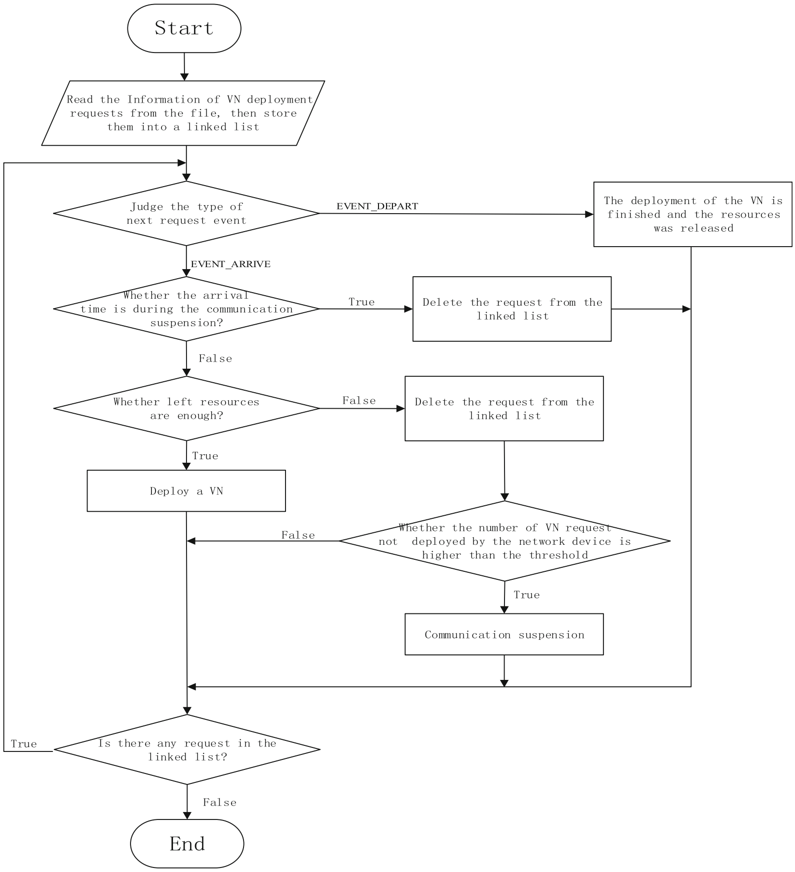

Fig. 4. Loose management on network devices

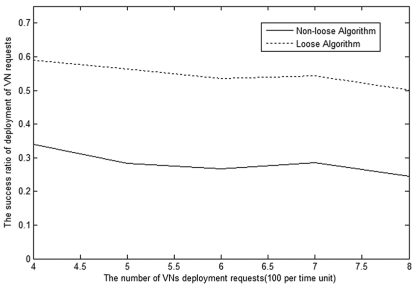

Fig. 5. Relationship between λ and η

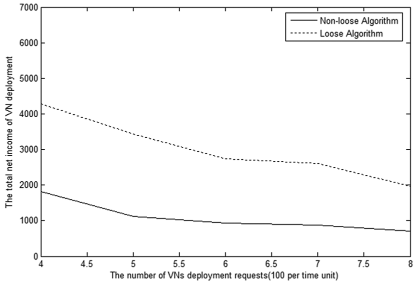

Fig. 6. Relationship between λ and R_{ev}

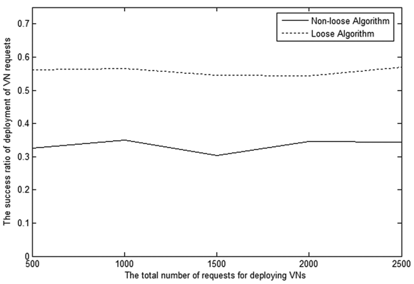

Fig. 7. Relationship between T and η

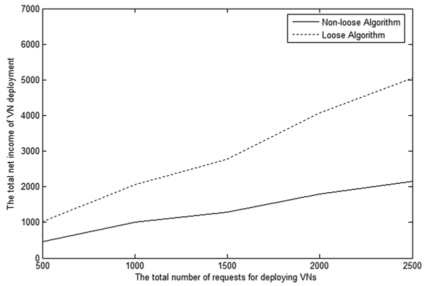

Fig. 8. Relationship between T and R_{ev}

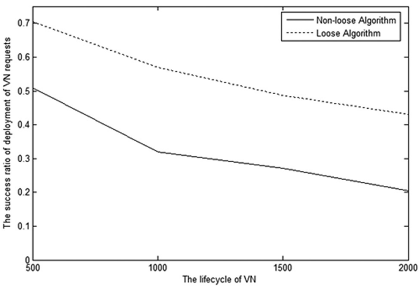

Fig. 9. Relationship between **M** and **η**

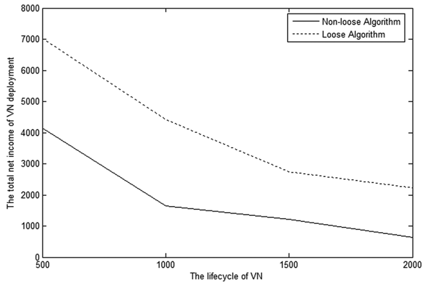

Fig. 10. Relationship between **M** and $\boldsymbol{R_{ev}}$

4.2 Mathematical Model of Loose Management Algorithm

To simplify the derivation, we assume that the request of VNs are uniform arrived in our mathematical modeling.

The maximum number of virtual nodes that a single network device can support at the same time is defined as R_{sub}/R_{VN}, $(R_{sub}/R_{VN})+r$ is the number of requests for deploying VNs from the beginning to the suspension of communication. $((R_{sub}/R_{VN})+r)/\lambda$ is the average duration of a communication cycle. Next, we will discuss two cases.

(1) $t_1 < (T-((R_{sub}/R_{VN})+r)/\lambda)$ means the duration of communication suspension is shorter. Assume the proportion of communication time in unit time is y, the duration of communication time during the lifecycle of a VN is yT. So the average number of VNs that a network device can participate in unit time is

$$m_0 = (R_{sub}/R_{VN})/(yT). \tag{1}$$

During a period of communication between network devices and controllers, when the number of failed VN deployment reaches k, the network device will suspend the communication with the controller, therefore the average duration of a communication cycle is

$$t_2 = r/(\lambda - m_0). \tag{2}$$

Since y is the proportion of communication time in unit time, then,

$$t_1 + t_2 = t_2/y. \tag{3}$$

According to formula (1), (2), and (3), we can obtain

$$y = (r + \frac{R_{Sub}/R_{VN}}{T} t_1)/(r + \lambda t_1). \tag{4}$$

(2) $t_1 \geq (\mathrm{T} - ((R_{sub}/R_{VN}) + r)/\lambda)$ means the duration of communication suspension is longer, so that the network device restores communication with the controller after the VNs are already finished. Therefore, the average duration of a communication cycle is

$$t_2 = (R_{Sub}/R_{VN} + r)/\lambda. \tag{5}$$

So the average number of VNs that a network device can participate in in unit time is

$$m_0 = (R_{Sub}/R_{VN})/((R_{Sub}/R_{VN} + r)/\lambda). \tag{6}$$

According to formula (2) and (6), we can obtain

$$y = r/(r + (\lambda - m_0)t_1) \tag{7}$$

For both cases, the net income of VN deployment in unit time is,

$$R_0 = (m_0 \cdot s - \lambda \cdot x)y \tag{8}$$

The success ratio of VN deployment requests is:

$$\eta = m_0/\lambda \tag{9}$$

Next we will contrast mathematical models and simulation of the loose management.

In the simulation, default parameters are: r = 3, T = 1000, $\lambda = 0.04$, M = 2000, $R_{sub} = 100$, $R_{VN} = 12.5$. The simulation results are shown in Figs. 11 and 12 below.

From Figs. 11 and 12 we can see that the mathematical model can accurately reflect the performance of the loose management.

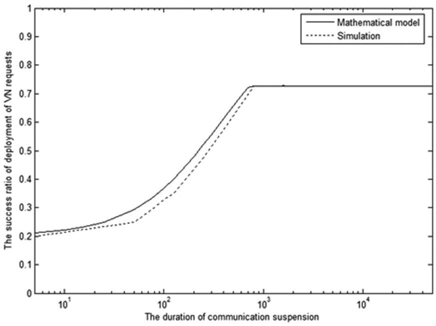

Fig. 11. Relationship between t_1 and η

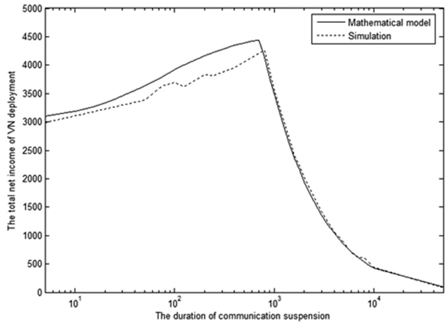

Fig. 12. Relationship between t_1 and R_{ev}

When the duration of communication suspension is much shorter. The number of communication suspension will decrease with the increasing of the duration of the communication suspension, therefore the number of VNs that the network device doesn't participate is fewer, so the success ratio of VN deployment requests and the total net income of VN deployment will increase.

When the communication suspension is much greater. Therefore, each communication cycle has almost the same number of VN deployment requests and the same number of successful VN deployment. Consequently, the success ratio of VN deployment requests will remain unchanged. However the total number of requests for deploying VNs will decrease with the increasing of the duration of the communication suspension, so the number of successfully deployment VNs will decrease, therefore the net income of VN deployment will decrease.

5 Conclusion

This paper proposes a novel loose management strategy to dynamically adjust the frequency of interaction between controllers and network devices. In detail, When the number of not deploy VNs in a network device reaches a threshold, the controller will temporarily stop the communication with the network device. After a period of time, the communication between the controller and the network device is resumed. It will improve the management and communication efficiency between controllers and network devices.

Based on the above idea, firstly, we designed the scheme and algorithm of controller loose management. Secondly, we quantitatively analyzed the advantages of controller loose management algorithm by mathematically modeling the VN deployment success rate and the communication revenue between controllers and network devices. Finally, simulation results show that the controller loose management idea can improve the communication efficiency between the controller and the network device and the controller management efficiency. Simulation results also show that mathematical model accurately predict the performance of loose management algorithm.

References

1. O.M.E. Committee.: Software-defined networking: the new norm for networks. In: ONF White Paper, Palo Alto, US. Open Networking Foundation (2012)
2. Agarwal, S., Kodialam, M., Lakshman, T.V.: Traffic engineering in software defined networks. In: Proceedings of IEEE INFOCOM, pp. 2211–2219 (2013)
3. Tootoonchian, A., Ganjali, Y.: Hyperflow: a distributed control plane for OpenFlow. In: Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking USENIX Association, p. 3 (2010)
4. OpenFlow Swtich Specification v1.5.0. Open Networking Foundation (2014)
5. Dan Marconett, S., Yoo, J.B.: FlowBroker: a software-defined network controller architecture for multi-domain brokering and reputation, pp. 328–359 (2014)
6. Casado, M., Freedman, M.J., Pettit, J.: Ethane: taking control of the enterprise. In: Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. ACM, pp. 1–12, New York (2007)
7. Hu, Y., Tian, T., Wang, J.: D-ZENIC: a scalable distributed SDN controller architecture. ZTE Technology, pp. 23–27 (2014)
8. Tam, A.S.W., Xi, K., Chao, H.J.: Use of devolved controllers in data center networks. In: Proceedings of 2011 IEEE Conference on Computer Communications Workshops, pp. 596–601 (2011)
9. Dixit, A., Hao, F., Mukherjee, S.: Towards an elastic distributed sdn controller. In: Proceedings of The Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. ACM, pp. 7–12 (2013)
10. Canini, M., Kuznetsov, P., Levin, D.: A Distributed SDN Control Plane for Consistent Policy Updates. arXiv preprint arXiv (2014)
11. Canini, M., Kuznetsov, P., Levin, D.D.: Software transactional networking: concurrent and consistent policy composition. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. ACM, pp. 1–6 (2013)
12. Fu, Y.H., Bi, J., Wu, J.P.: A dormant multi-controller model for software defined networking. In: Proceedings of Communications System Design, pp. 45–55, China (2014)
13. Yeganeh, S.H., Ganjali, Y.: Kandoo: a framework for efficient and scalable offloading of control applications. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks. ACM, pp. 19–24 (2012)
14. Koponen, T., Casado, M., Gude, N.: Onix:. A distributed control platform for large-scale production networks. In: Proceedings of the 9th USENIX OSDI Conference, pp. 1–6 (2010)
15. Macapuna, C.A.B., Rothenberg, C.E., Magalh, F.: In-Packet bloom filter based data center networking with distributed openflow controllers. In: Proceedings of 2010 IEEE International Workshop on Management of Emerging Networks and Services. IEEE, pp. 584–588 (2010)
16. Bari, M.F., Roy, A.R., Chowdhury, S.R.: Dynamic controller provisioning in software defined networks. In: Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013), pp. 18–25 (2013)
17. Xin, J., Jennifer, R., David, W.: Incremental update for a compositional SDN hypervisor. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, HotSDN 2014, pp. 187–192, New York (2014)
18. ONOS Team. Open network operating system (2012). http://onosproject.org/