

# Proactive Admission Control and Dynamic Resource Management in SDN-based Virtualized Networks

Sara Shakeri

Sharif University of  
Technology, Tehran, Iran

Email: sa.shakeri90@student.sharif.edu

Saeedeh Parsaeefard

Communication Technologies and  
Department, Iran Telecommunication

Research Center, Tehran, Iran

Email: s.parsaeefard@itrc.ac.ir

Mahsa Derakhshani

Wolfson School of Mechanical, Electrical  
and Manufacturing Engineering,

Loughborough University, Loughborough,

Leicestershire, U.K.

Email: m.derakhshani@lboro.ac.uk

**Abstract**—Network virtualization is a promising approach in which common physical resources are shared between service providers. Due to the substrate network limitations such as maximum available memory of each node of the substrate network as well as different service priorities and requirements, resource management in this setup is essential. On the other hand, SDN is bringing a considerable flexibility in resource management by introducing a centralized controller which can monitor all the substrate network states. In this paper, we propose a proactive admission control and dynamic resource management in SDN-based virtualized network in which the number of accepted high-priority virtual network (VN) requests is maximized, subject to both substrate limitations and memory requirement of each VN request. In the proposed formulation, based on the prediction of the substrate network utilization, we reserve resources for upcoming high-priority VN requests. Via simulation, we show that the algorithm can increase the acceptance ratio of the high-priority VN requests up to %100 where the substrate network is congested, i.e., arrival rates of both high-priority and low-priority VN requests are high.

## I. INTRODUCTION

Network virtualization has emerged as a promising approach enabling efficient network management among service providers (SPs). It brings an opportunity in which future innovation and deployment of new technologies are effectively affordable and manageable. In fact, in network virtualization, the substrate network provider offers a common infrastructure to support different service providers by slicing the physical network to virtual ones.

However, physical resource limitations raise a number of challenges in virtual network (VN) resource management [1]. One key challenge is related to serving a large number of virtual requests with different requirements and priorities. There are some VN requests which are more important than the others and serving them in the network is of an essential. For example, serving a VN request that is related to multimedia or financial process has higher priority than serving a non-profit one. Therefore, proposing an algorithm with high acceptance ratio of high-priority VN requests plays a significant role in virtualized networks efficiency [2], [3].

In parallel, SDN [4] can be used as a flexible and efficient way for managing the VN resources, with different requirements, by decoupling the control plane from the data plane. In a SDN-based virtualized network each virtual link is defined

as a flow and hence, a VN request is a set of flow rules in the substrate network switches. In fact, in these networks, one can propose a network-wide resource management and use variety of policies to dynamically provide the most efficient allocation of resources for different virtual requests considering their priorities.

There are some several works presenting efficient resource management methods in SDN-based virtualized networks [5]–[9]. For example, [5] introduces a static resource management approach by deploying policies in the SDN controller in order to minimize the delay between the controller and the substrate switches, and [6] proposes an optimal end-to-end virtual path using a linear algorithm in the centralized controller. In [10], a dynamic method is used to remap allocated VNs in order to increase acceptance ratio of VN requests and maximize utilization of the substrate network. However, the aforementioned studies did not consider the quality of service (QoS) requirements and priority levels of virtual requests. In this paper, to address this point, we propose a proactive admission control and dynamic resource management algorithm in a SDN-based virtualized network in which the priority of VN requests is taken into account.

The rationale behind the proposed algorithm is increasing the number of served high-priority virtual requests in the network, specifically when the substrate network is over utilized and rejecting high-priority VN requests is more probable. To achieve this goal, the network-wide state of the substrate network and traffic behavior of upcoming VN requests should be available. However, the latter is not a practical assumption. To tackle this implementation issue, we aim to extend the Floodlight controller [11] by adding an admission control module, which predicts the next state of the substrate network resources based on the arrival rate of VN requests after each successful mapping. Afterwards, this module calculates the substrate network utilization. If the network utilization is above a predefined threshold which could cause rejection of an upcoming high-priority VN request, this module proactively drops a low-priority VN request (which is being served) in order to avoid rejecting the high-priority ones. In this paper, we assume that the arrival rate of new VN requests is according to Poisson distribution. For evaluating the proposed method, we simulate the substrate network by Mininet [12] and deploy

Floodlight as the controller.

The rest of this paper is organized as follows: We present related works in Section II. System model and formulation of the problem are presented in Section III. Proposed solution and the algorithm of method is described in Section IV. Our proposal is evaluated in Section V, and the paper concluded in Section VI.

## II. RELATED WORKS

Virtualization is an important concept for managing enterprise networks. There have been intensive efforts to study VN embedding, considering link and node limitations. For example, [13] proposes an online heuristic embedding algorithm to maximize the total revenue of the accepted VN requests, subject to node and link resource constraints. Authors in [14] define node ranking which introduces the importance of each node and is computed using Markov random walk method based on nodes resources and their neighbors capacity. Two VN embedding algorithms are proposed in this paper. The first algorithm executes the node and link mapping in separate stages, while the second one performs it simultaneously. Both of the proposed algorithms could increase the long-term average revenue [13] as well as the acceptance ratio. In [15], a learning approach is proposed to increase network utilization and acceptance ratio. The authors introduce an automatic, dynamic and topology-aware solution to predict the next resource requirements based on the available resources of a node and its neighbors attributes. Although all the aforementioned works proposed some efficient resource management methods, none of them considers the priority of VN requests.

In addition, providing virtualization regarding SDN brings an appropriate platform for deploying VNs in which better management and flexibility can be offered. For example, FlowVisor [16] and FlowN [17] introduce efficient SDN-based virtualization by sharing the same hardware of the substrate network between multiple tenants. These two works bring the substrate network isolation as well as scalability. However, they do not consider the resource management of the substrate switches and links. [5] proposes a load balancing method in VN embedding to minimize the connection delay between the controller and the substrate network considering the limitations of nodes and links resources. In [6], by using an integer linear programming in the centralized controller, an optimal end-to-end virtual path is obtained. These two aforementioned studies propose static solutions supposing that the VN requests are known in advance. However, due to the dynamic behavior of the VN requests, proposing a dynamic resource management in the virtualized networks is of an essential. [10] addresses dynamic resource management in a SDN-based virtualized network considering the link and the substrate switches' constraints. The authors use a path migration method aiming to manage resources in an efficient way by modifying mapping after receiving a new virtual request or releasing an accepted one. The method presented in [18] compares the evolution of the network based on the admission decisions taken by different online algorithms, and

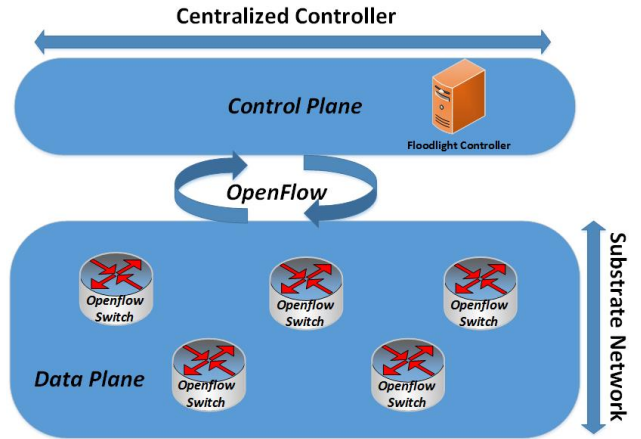


Fig. 1: SDN architecture

then, by simulating future condition intends to make a decision aiming to improve infrastructure.

Although all mentioned works explore different methods for mapping virtual requests to the physical substrate network, to the best of our knowledge, there is no work in this context which takes into account the priority of each VN request in an SDN-based virtualized environment. To address this important practical issue, in this paper, we propose a proactive and dynamic resource management approach in an SDN-based virtualized network, considering different priorities of VN requests in resource management.

## III. SYSTEM MODEL

SDN network architecture is shown in Fig. 1. This figure demonstrates the separation of the control plane from the data plane using OpenFlow protocol. In an SDN-based virtualized network, all of the OpenFlow switches are considered as the substrate network and each virtual link is considered as a flow in the substrate network. Therefore, each VN request is a set of flow rules inserted by the controller in different Openflow switches. The controller is responsible for mapping VN requests to substrate switches and can develop different algorithms to reach the most effective resource management.

### A. Network Graph Representation

The SDN substrate network can be modeled as an undirected graph,  $\mathcal{G}(\mathcal{N}, \mathcal{L})$ , where  $\mathcal{N} = \{n_1, n_2, n_3, \dots, n_N\}$  represents a set of the substrate switches and  $\mathcal{L}$  represents a set of the links between these switches. For example,  $l_{n_1 n_3} \in \mathcal{L}$  connects the switch  $n_1$  to  $n_3$  in the substrate network. In addition, each OpenFlow switch is characterized by its available ternary content-addressable memory (TCAM) [19]  $m_{n_i}$  representing the number of the flow entries that switch  $n_i \in \mathcal{N}$  can support. Let denote the amount of available memory of the substrate switches by

$$M = [m_{n_1}, m_{n_2}, m_{n_3} \dots, m_{n_N}]. \quad (1)$$

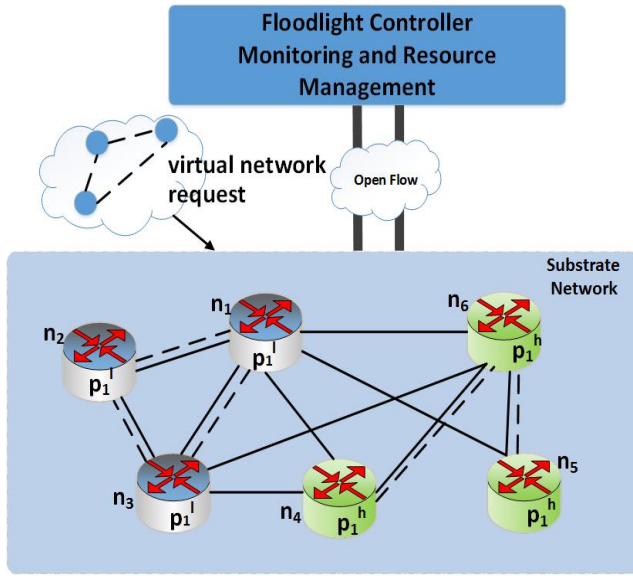


Fig. 2: SDN-based virtualized network

Furthermore, the allocated memory to VN requests of the switch  $n_i$ , is denoted by  $m'_{n_i} \leq m_{n_i}$ , and the allocated memories of all switches are considered as the network state and represented by

$$S = [m'_{n_1}, m'_{n_2}, m'_{n_3} \dots, m'_{n_N}], \quad (2)$$

which will be updated in case of allocating or releasing any of resources. In this paper, we aim to take into account the priorities of VN requests in dynamic resource allocation. Accordingly, we categorize allocated VN requests in two sets based on their priority:  $\mathcal{P}^L$  which is used for low-priority VN requests where  $p_i^l \in \mathcal{P}^L$  shows the  $i^{th}$  allocated low-priority VN request, and  $\mathcal{P}^H$  which is applied for high-priority ones in which  $p_i^h \in \mathcal{P}^H$  shows the  $i^{th}$  allocated high-priority VN request.  $\mathcal{P}^L$  and  $\mathcal{P}^H$  will be updated in the case of allocating or releasing any VN request.

### B. Problem Description

Fig. 2 shows an example of virtual mapping in a SDN-based virtualized network. In this example, a low-priority VN request  $p_1^l$  is mapped onto the nodes  $n_1$ ,  $n_2$  and  $n_3$ , and a high-priority one  $p_1^h$  is mapped onto the nodes  $n_4$ ,  $n_5$  and  $n_6$ . Suppose that all available memories of these switches are allocated, and the substrate network cannot accept any other new VN request. In this situation, in case of arriving a new high-priority VN request as shown in Fig. 2, it will be rejected due to the lack of resources. However, this high-priority VN request could be accepted if the low-priority VN request was not allocated. Since admission of a high-priority VN request is of great importance, it is preferable to accept a high-priority VN request rather than keeping a low-priority one, where the substrate network is over utilized. Accordingly, the main

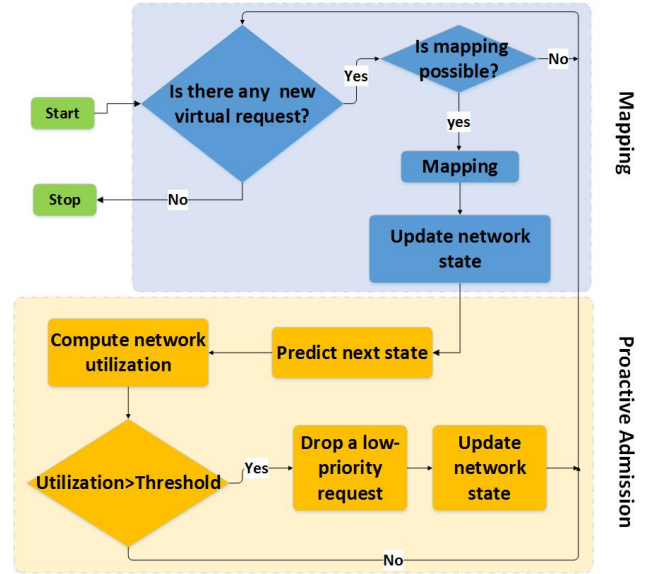


Fig. 3: Proactive admission control algorithm

objective of this paper is to increase the acceptance ratio of high-priority VN requests applying the proactive admission control algorithm discussed in the next section.

## IV. THE PROPOSED ALGORITHM

In this section, we propose a proactive admission control algorithm whose flow chart is depicted in Fig. 3. The proposed algorithm (written as a module in Floodlight controller), monitors the state of the network resources and decides whether or not mapping, rejecting or dropping VN requests. In addition, the proposed algorithm collects the information of the available resources in the substrate network for resource management purposes. Indeed, the stored data in this module will be updated in case of any successful mapping or releasing of any resources. The proposed algorithm consists of the following two important steps.

### A. Mapping

In order to map a VN request to the substrate network, each virtual node should be mapped on to a unique substrate network switch, and only one request is served at any given time. Moreover, each virtual link should be mapped on one or more substrate network links to bring the connection between virtual nodes. The main focus of this paper is on proposing a proactive admission method that decides about the substrate network resources after mapping. Therefore, this step is out of the scope of this paper, and for the mapping step we use a modified version of one of the state-of-the-art mapping algorithms proposed in [20]. At this step, we use proposed method in [13] for VN mapping with some modifications as follows.

For mapping virtual nodes, a greedy algorithm, as presented in the algorithm (1) is deployed. At the first step, the virtual nodes based on the amount of their requirements are sorted in

---

**Algorithm 1: Mapping**

---

**Data:** Network State, VN Request

```
1 Function NODE MAPPING
2 Sort the VN request nodes based on the amount of their
  memory requirements descendingly in list  $R$ ;
3 foreach node  $k$  from the top of list  $R$  do
4   Find the set  $Q$  of the substrate network nodes that
  satisfy the virtual node  $k$  requirements;
5   Delete virtual node  $k$  from list  $R$ ;
6   if  $Q == null$  then
7     Flag=0;
8     Reject the current VN request;
9     Return Flag;
10  else
11    Select the substrate node from  $Q$  with the most
  available memory for allocating to node  $k$ ;
12    Delete the selected substrate node from the
  available nodes for this VN request;
13 Update the network state according to Equation (2);
14 Call Function LINK MAPPING;
15 Flag=1;
16 Return Flag;
17 End Function
18 Function LINK MAPPING
19 foreach Virtual Link do
20   Find the first shortest path in the substrate network.
21 End Function
```

---

---

**Algorithm 2: Proactive Admission Control**

---

**Data:** Network State

```
1 Function PROACTIVE ADMISSION CONTROL
2 Call Function NODE MAPPING from Algorithm (1);
3 if Flag==1 then
4   Predict the next state of the network based on
  Equation (3);
5   Calculate utilization ( $U$ ) based on Equation (4);
6   if  $U \geq Threshold$  then
7      $D_i^L = 1$ , i.e., drop the low-priority allocated VN
  request which requires the longest time to be
  released;
8 End Function
```

---

the list  $R$  descendingly. Then, each virtual node from top of the sorted list will be selected, and the sets of the substrate nodes which can support its requirements are determined and stored in  $Q$ . Afterwards, for each virtual node, a node with the most available memory will be selected and then omitted from the available substrate nodes for this VN request. The current VN request will be accepted if the algorithm can find a substrate node for all of the virtual nodes. Otherwise, it will be rejected, and the next request is considered. For mapping

virtual links, since we do not have any restriction on the substrate network links, the first shortest path algorithm is selected. It is worth mentioning that although the proposed algorithm is designed for one type of resources, i.e., memory, the extension to different types of resources is straightforward.

### B. Proactive Admission

The admission control is the main focus of the proposed algorithm in which the proactive decision is made to allocate resources. Assuming that the knowledge of average arrival rates of VN requests is perfectly known, the next state of the substrate network can be predicted by considering the arrival probability of the next VN request in the next time unit. Afterwards, based on the forecast of the next state, an estimate of resource utilization in the next time unit will be calculated. Then, the algorithm decides whether to drop a low-priority VN request or not. Therefore, the proposed proactive admission algorithm is comprised of two phases including *Prediction* and *Utilization-based Admission*.

1) *Prediction*: This phase is responsible for providing an estimation of the next state of the substrate network based on the arrival probability of the upcoming VN request in the next time unit. In particular, the next state of network is predicted as

$$\hat{S}_{T+1} = \alpha S_{T+1} + (1 - \alpha) S_T \quad (3)$$

where  $\alpha$  represents the probability of arriving a new request in the next time unit. The first term of the right-hand side of (3) represents the case that the new VN request is arrived, and the needed resources are allocated based on the average amount of requested memory. Moreover, the second term of the right-hand side of (3) represents the case that the state of the substrate network does not change when there is no request in the next time unit with probability of  $(1 - \alpha)$ . Finally,  $\hat{S}_{T+1}$  represents the predicted next state of the substrate network for the next time unit.

2) *Utilization-based Admission*: Based on the predicted next state,  $\hat{S}_{T+1}$ , the network resource utilization is computed as

$$U = \frac{\sum_{n_i \in \mathcal{N}} \hat{m}_{n_i}}{\sum_{n_i \in \mathcal{N}} m_{n_i}} * 100 \quad (4)$$

where  $U$  is the ratio of the allocated resources to the total resources of the substrate network, and  $\hat{m}_{n_i}$  represents the  $i^{th}$  element of  $\hat{S}_{T+1}$ . Also,  $m_{n_i}$  is the  $i^{th}$  element of  $M$ . If the percentage of network utilization is larger than a preselected threshold, there is a chance that in the next state, upcoming high-priority VN requests would be rejected. Therefore, in this situation, the proposed admission control algorithm proactively drops one of the low-priority allocated VN requests from  $\mathcal{P}^L$ . In order to increase the number of completed low-priority VN requests, the VN request which requires the longest time to be released will be dropped. For this dropping criteria, we introduce  $D_i^L$  as

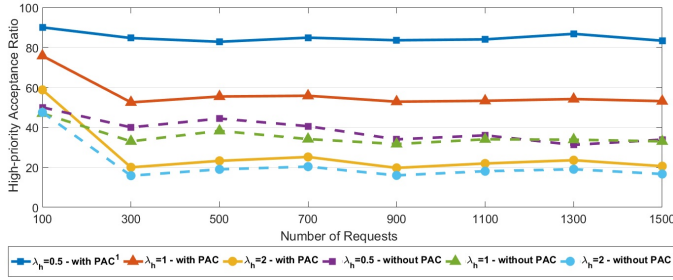


Fig. 4: High-priority acceptance ratio with  $\lambda_l = 0.5$  requests/time unit

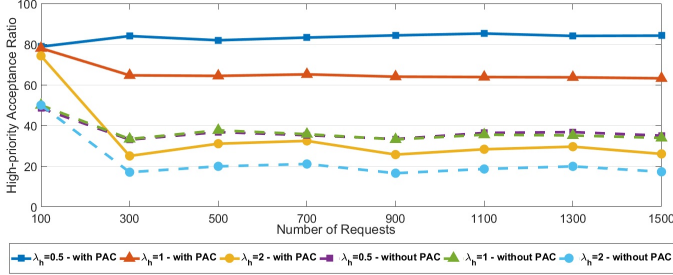


Fig. 5: High-priority acceptance ratio with  $\lambda_l = 1$  requests/time unit

$$D_i^L = \begin{cases} 1 & \text{if } U \geq \text{Threshold} \\ 0 & \text{if } U < \text{Threshold} \end{cases}$$

where  $i$  is the index of the low-priority accepted VN request in  $\mathcal{P}^L$ , which has the longest time to be released. Note that when  $D_i^L$  is 1,  $i^{\text{th}}$  low-priority request will be dropped.

## V. SIMULATION RESULTS

In order to evaluate the proposed method in this paper, we use the Floodlight controller and Mininet for simulating SDN, where they both run on the same Ubuntu virtual machine with 4.0GB RAM and 2 core processors. All topologies in this paper are created by Brite [21], and the number of nodes in the substrate network is 16. In addition, the number of nodes required for each VN request's topology is uniformly distributed between 5 and 10. The memory capacity of the substrate switches is uniformly distributed between 200 and 400 units, and the memory demand for each node in VN requests is uniformly distributed between 3 and 10 units. Here we assume that each flow rule in the substrate switches needs one memory unit, and the service time of each virtual request is set to 150 time units. For this simulation setup, VN request arrivals follow Poisson distribution with the average arrival rate of  $\lambda_h$  for high-priority VN requests and  $\lambda_l$  for low-priority ones. We estimate the next state of the network based on the average arrival rate of high-priority VN requests according to their Poisson distribution.

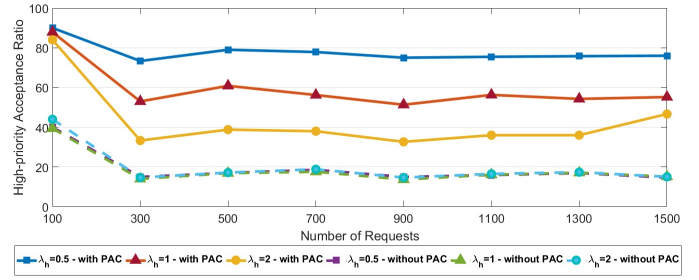


Fig. 6: High-priority acceptance ratio with  $\lambda_l = 2$  requests/time unit

## A. Performance Metrics

To evaluate performance of the proposed proactive admission control algorithm, we consider the following two performance metrics.

- 1) Acceptance ratio of high-priority VN requests: According to the goal of this paper in managing resources in order to increase the number of accepted high-priority VN requests in the substrate network, we calculate the ratio of the accepted high-priority VN requests to the total number of high-priority VN requests.
- 2) Utilization: We also measure the average ratio of sum of utilized resources to the total available resources of the substrate network at any time unit.

## B. Results Evaluation

In Figs. 4-6, high-priority acceptance ratio of the substrate network with proactive admission control (which is shown by PAC<sup>1</sup>) at different average arrival rates versus the number of requests is demonstrated. Figs. 4 and 5 show that the proposed admission control approach increases the average high-priority acceptance ratio up to %60 compared to the case that no admission control is deployed for all average arrival rates.

In addition, in a congested scenario in Fig. 6, the acceptance ratio of high-priority VN requests will be increased up to %100 using the proactive admission control as compared to the case where the proposed method is not applied. This significant increase confirms that the proactive admission control method based on (3) performs effectively and improves the network high-priority acceptance ratio considerably.

Comparing Figs. 7 and 8, it is clear that the network resource utilization of proactive admission control algorithm is close to the case that admission control is not applied for all values of  $\lambda_h$  and  $\lambda_l$ . In other words, the proposed algorithm can effectively utilize the network resources and additionally, increase the high-priority acceptance ratio in comparison to where no admission control scheme is deployed.

## VI. CONCLUSION

This paper proposes a dynamic resource management method in a SDN-based virtualized network in which the

<sup>1</sup>Proactive Admission Control

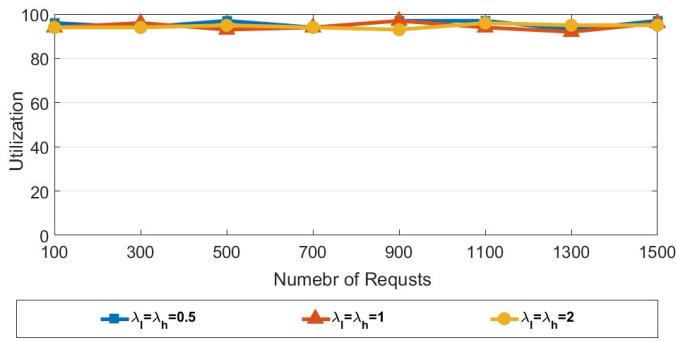


Fig. 7: Average utilization with different  $\lambda$  applying proactive admission control

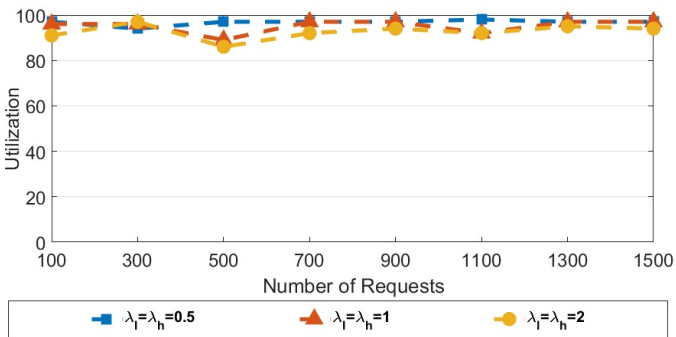


Fig. 8: Average utilization with different  $\lambda$  without applying proactive admission control

importance of high-priority VN requests is considered. In a virtualized network, there are variety of requests with different importance and criticality levels where ensuring enough resources for critical requests is necessary. This paper suggests a method that proactively drops a low-priority VN request to avoid rejecting a high-priority one, considering the network state and resource utilization. Applying this approach, the acceptance ratio of high-priority VN requests can increase up to %100 in the highly congested network scenario. Also, the simulation results confirm that the proposed method does not decrement the network utilization. For the future work, we intend to propose a more precise prediction method applying learning algorithms and wider traffic patterns aiming to increase the overall acceptance ratio of network.

## REFERENCES

- [1] N. M. Mosharaf Kabir Chowdhury and R. Boutaba, "A survey of network virtualization," *Comput. Netw.*, vol. 54, no. 5, pp. 862–876, Apr. 2010.
- [2] J. Xiao and Z. Wang, "A priority based scheduling strategy for virtual machine allocations in cloud computing environment," in *Cloud and Service Computing (CSC), 2012 International Conference on*. IEEE, 2012, pp. 50–55.
- [3] J. Cai, X. Nian, H. Gu, and L. Zhang, "A user priority-based virtual network embedding model and its implementation," in *2013 IEEE 4th International Conference on Electronics Information and Emergency Communication*, Nov 2013, pp. 33–36.

- [4] B. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.
- [5] D. Mehmet and A. Mostafa, "Design and analysis of techniques for mapping virtual networks to software-defined network substrate," *computer communication*, vol. 45, no. 0, pp. 1–10, 2014.
- [6] R. Trivisonno, I. Vaishnavi, R. Guerzoni, Z. Despotovic, A. Hecker, S. Beker, and D. Soldani, "Virtual links mapping in future SDN-Enabled networks," in *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, Nov 2013, pp. 1–5.
- [7] <https://www.necam.com/PFlow/doc.cfm?t=PFlowController>.
- [8] M. F. Ramdhani, S. N. Hertiana, and B. Dirgantara, "Multipath routing with load balancing and admission control in software-defined networking (SDN)," in *2016 4th International Conference on Information and Communication Technology (ICOICT)*, May 2016, pp. 1–6.
- [9] K. Zhu, Y. Ran, E. Yang, and J. Yang, "Joint admission control and routing via neuro-dynamic programming for streaming video over SDN," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, June 2017, pp. 20–25.
- [10] R. Mijumbi, J. Serrat, J. Rubio-Loyola, N. Bouten, F. D. Turck, and S. Latre, "Dynamic resource management in SDN-based virtualized networks," in *10th International Conference on Network and Service Management (CNSM) and Workshop*, Nov 2014, pp. 412–417.
- [11] S. A. Shah, J. Faiz, M. Farooq, A. Shafi, and S. A. Mehdi, "An architectural evaluation of SDN controllers," in *2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 3504–3508.
- [12] R. L. S. de Oliveira, C. M. Schweitzer, A. A. Shinoda, and L. R. Prete, "Using Mininet for emulation and prototyping software-defined networks," in *2014 IEEE Colombian Conference on Communications and Computing (COLCOM)*, June 2014, pp. 1–6.
- [13] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Mar. 2008.
- [14] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 38–47, Apr. 2011.
- [15] R. Mijumbi, S. Hasija, S. Davy, A. Davy, B. Jennings, and R. Boutaba, "A connectionist approach to dynamic resource management for virtualised network functions," in *2016 12th International Conference on Network and Service Management (CNSM)*, Oct 2016, pp. 1–9.
- [16] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKown, and G. Parulkar, "Can the production network be the testbed?" in *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, ser. Operating System Desing and Implementation'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 365–378.
- [17] D. Druts koy, E. Keller, and J. Rexford, "Scalable network virtualization in software-defined networks," *IEEE Internet Computing*, vol. 17, no. 2, pp. 20–27, March 2013.
- [18] S. Paris, J. Leguay, L. Maggi, M. Draief, and S. Chouvardas, "Online experts for admission control in SDN," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 1003–1004.
- [19] K. Kannan and S. Banerjee, "Compact TCAM: Flow entry compaction in TCAM for power aware SDN," in *International Conference on Distributed Computing and Networking*. Springer, 2013, pp. 439–444.
- [20] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, April 2006, pp. 1–12.
- [21] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *Proceedings of the Ninth International Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, ser. MASCOTS '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 346–353.