Analysis of the Impact of Weight Parameter of Traffic Load Aware Replacement Algorithm of SDN Flow Table Cache

Tiep Vu Duc, Kyungbaek Kim

Department of Electronics and Computer Engineering
Chonnam National University
Gwangju, South Korea
ductiep91@gmail.com, kyungbaekkim@jnu.ac.kr

Abstract

Software defined networking (SDN) is one of the hottest topics in networking nowadays. In SDN, flow table size issue, which affects the scalability of a SDN system, is a challenging issue for researchers. One promising solution is to use a flow table cache with a smart flow replacement algorithm. Recently a traffic load aware flow table cache with a new ranking based replacement algorithm was proposed, in which traffic load of network flows are considered. The traffic load of a flow is represented through two factors: the average packet rate and the average bytes per packet. A packet rate weight and number of bytes per packet parameters are used to adjust to the influence of corresponding factors. Though the algorithm was evaluated to be more efficient than previous algorithms, the impact of these weight parameters to performance of traffic load aware SDN flow table cache have not been clearly analyzed. For different traffic scenarios, the average packet rate and the average number of bytes per packet might have different level of impact on the utilization of the network flow, and hence affect the performance of flow table cache. In this paper we investigated the relationship between these weight parameters and performance of the traffic load aware SDN flow table cache for different traffic scenarios. Through extensive experiments, we found the change tendency of performance of the traffic load aware flow table cache when the parameters are varying. For each traffic scenario, the optimal value of parameters and the tendency were well described and documented. The results indicated that our hypothesis was right.

SDN, flow table; flow table cache; traffic load aware replacement algorithm; packet rate weight; number of bytes per packet weight.

I. Introduction

Software defined networking (SDN) is an emerging technology that allows directly program the network control and the underlying infrastructure [1]. The decouple control plane and data plane enables many benefits such as centralized management, centralized provisioning, lower device cost, more flexibility, etc. However, SDN also contain many issues in which scalability is one of the most critical one. One cause for the scalability issue is the flow table size. Since the component is expensive and power hungry [2], flow table in SDN switch usually has small size, and cannot satisfy the requirement of large network system.

One way to mitigate the flow table size issue is to reduce the flow entries in flow table by improving SDN flows and rules management through the use of wildcard and tag as proposed in [3], [4] and [5]. However, the approach requires modifications of flow tables and SDN controller while the risk of fully filled flow table is still there. A more attractive approach is to use flow table cache to store flow entries when flow table is full as in [6], [7] and [8]. The flow table cache is a software module that sits between controllers and switches. It is inexpensive and big in size but it is slower than the physical flow table. Therefore, some flow entry should be store in flow table and the other stores in flow table cache in order to get the best performance of flow table cache. A smart flow replacement algorithm is required for that purpose. In [6], a flow table cache that adopted the popular Least Recently Used (LRU) algorithm was proposed. The LRU will select the flow entry that is unused for the longest time in flow table to be replaced by new flow entry. Another flow table cache was proposed in [7] which took advantage of the packet rate feature for flow replacement algorithm. First, flow entries are classified into groups. Each group contains only flow entries that are dependent on each other. Then, each flow entry is given a weight corresponding to its packet rate and a cost corresponding to the number of dependent flow entries. The flow entry which has smallest total weight will be replaced in flow table. However, only packet rate feature is not enough to represent the complexity of nowadays network traffics in which the traffic flows have very diverse packet rates and number of bytes per packet.

Recently a traffic load aware flow table cache with a ranking based replacement algorithm was proposed in [8]. The traffic load of a flow is represented by its average packet rate and its average bytes per packet. Based on the traffic low information, the utilization of each flow is calculated and the flows which have higher utilization will be keep in in flow table. The influence of packet rate and number of bytes per packet to the utilization of a flow can be adjusted through weight parameters. Although the algorithm was evaluated extensively and proved to be more efficient than previous algorithms in [8] and [9], the impact of these weight parameters to performance of traffic load aware flow table cache has not been clearly studied. We believe that the packet rate weight parameter plays an important role in the ranking based replacement algorithm. As the packet rate weight parameter is changing, there should be a changing tendency in performance of the traffic load aware flow table cache. Hence we can find an optimal value of the parameter which results in the best performance. Therefore, in this paper we studied carefully the impact of the packet rate weight parameter to the performance of traffic load ware flow table cache in different traffic scenarios.

Our contribution are:

- We emulated a SDN network using Opendaylight controller and Mininet. A flow table cache software module was implemented in Opendaylight controller with traffic load aware replacement algorithm.
- Through extensive experiment, we found the optimal packet rate weight of the traffic load aware algorithms which maximize the performance in some special network traffic scenarios.

The rest of the paper is organized as follows: in section 2, we present the previous work about Traffic Load Aware Flow Table Cache as well as our observations and motivation for the evaluation. The evaluation is described in section 3. In section 4, we present the obtained results and our comments. Finally, we conclude this paper in section 5.

II. Traffic Load Aware Replacement Algorithm

A flow table cache is a software module which sits between SDN controller and network switches, and it supports switch to handle information of network flow when the flow table is already full. When a packet comes into network switch, the switch will check whether the packet matches any of its flow table entry. If it does, the switch just simply forward the packet to the next switch. Otherwise, the packet will be forwarded to the corresponding flow table cache. Similarly, the packet will also be checked if it matches any flow entry in flow table cache. If there is no flow matched the packet, the flow table cache will forward the packet to controller and the controller will generate a new flow entry for it. Since a software flow table cache has slower speed than the physical switch's flow table, the performance can be improved if more useful flow entries are put in the hardware flow table. For that reason, the traffic load aware replacement algorithm was proposed. The traffic load of a flow is characterized by its average packet rate and average bytes per packet. These information of a network flow are collected by the flow table cache and they are used to calculate the utilization of the network flow as shown in equation 1.

$$U_f = \frac{W_p \times p_f}{P_{max}} + \frac{W_b \times b_f}{B_{max}} \qquad (1)$$

In the equation 1, p_f is the average packet rate of a network flow f, and b_f is the average packet rate number of bytes per packet of a flow f. P_{max} and B_{max} are the maximum packet rate and the maximum number of bytes per packet of network flows collected by the flow table cache. W_p and W_b are weight values of packet rate and number of bytes per packet respectively and $W_b = 1 - W_p$. The range of the utilization is from 0 to 1, and 1 means that the corresponding network flow entry has the highest utilization. When a flow table needs an empty flow entry, the flow entry which has the lowest utilization will be evicted.

The forwarding time by using flow table cache is higher than flow table, hence packets must wait longer in switch queue to be forwarded. If too many packets come in and the replacement takes too long, the queue will be full and cause some packets to be dropped. The situation could be worse if there are more highload network flows. If the replacement can quickly put high load flow in flow table and evict the small load flow, the performance can be increased. The weight parameters have important roles in this process. In order to improve the performance of the algorithm, finding the optimal value for these parameters is an important issue.

To find the optimal value, we defined four special network traffic scenarios. First, we consider the uniform load traffic where every flow has the same average packet rate and number of bytes per packet. In this case, the utilization of each network flow is the same in every way. Therefore, changing the weight parameter might only have a small effect to performance of the traffic load aware flow table cache. The parameter would be more useful when network traffics are diverse such as in the "1 low : 1 high" network traffic, which has a half number of low packet rate flows and the other half is high packet rate flows. This kind of traffic requires more flow replacement time, and cause more packet dropped. We also consider two other scenarios: the "2 low : 1 high", and the "1 low : 2 high" traffics. The "2 low : 1 high" network traffic has two-third number of low packet rate flows and one-third number of high packet rate flow. The last case "1 low : 2 high" network traffic has one third of low packet rate flows and the other two third portion of high packet rate flows.

III. Evaluations

We emulated a SDN network by using OpenDayLight [10] controller and Mininet [11]. They are installed in two different machines to avoid any possible interference of computing resources. OpenDaylight is run on computer which has 2 3.4GHz CPUs and 4GB memory. Mininet runs on a different computer with 2 3.3GHz CPUs and 4GB memory. A traffic load aware SDN Flow Table Cache module is implemented and installed in OpenDaylight Controller.

The emulated network is depicted in figure 1. The network

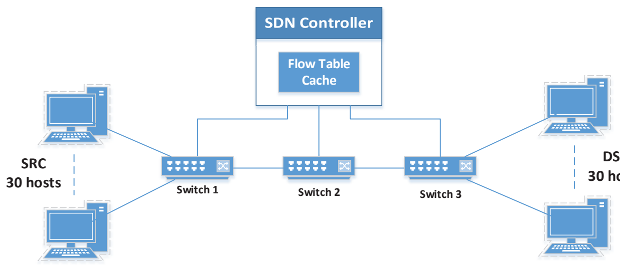

Fig. 1 Topology of the experiment.

consists of 3 network switches and 60 hosts. The switches are connected together in a chain, and they are all connected to controller. Hosts are divided into two same size groups: 30 source hosts in SRC group connect to switch 1 and other 30 destination hosts in the DST group connect to switch 3. That makes 30 pairs of hosts of different domains. Each pair generates a UDP network flow so that there are 30 flows on the emulated network. In order to create the situation in which the flow table of network switches are overflowed, we reduced the size of flow table of every switch to support only 11 flows. The other flows will be handled by the flow table cache.

The performance of the traffic load aware replacement algorithm is characterized by two factors: the average packet drop ratio and the average packet delay time. We are going to measure these two factors in our experiment. In order to see changing tendency of the performance, we let W_p vary from 0.1 to 0.9, hence W_b varies from 0.9 to 0.1 respectively.

We consider four type of traffics in our experiments. They

Table I. The four traffic profiles for the experiment

Traffic profile	Description
Uniform	All flows are has the same average packet rate
1 low : 1 high	A half low packet rate flows and a half high packet rate flows
2 low : 1 high	Two portions of low packet rate flows and one portion of high packet rate flows
1 low : 2 high	One portion of low packet rate flows and two portion of high packet rate flows

are summarized in table 1 and are graphically depicted in figure 2. The packet size for every packet is 1024KB. The total bandwidth of all flows in each traffic profiles are the same but the network traffic portion occupied by each type of flow in each profiles are different.

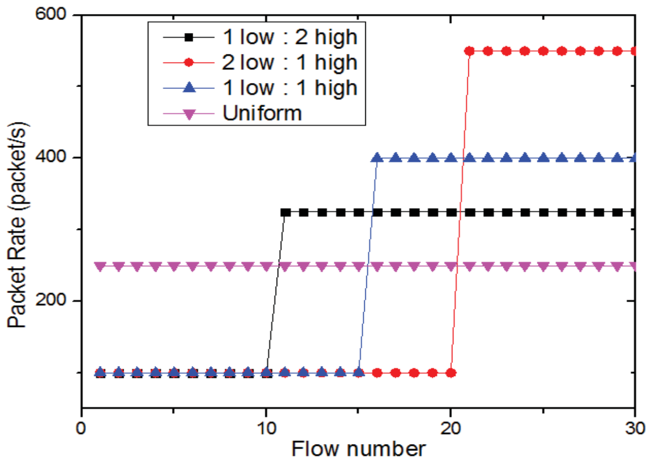

Fig. 2 Graphical Representation of four traffic profiles.

IV. Evaluation Results

The experimental average packet drop ratio and average packet delay time are shown in figure 3 and figure 4 respectively.

First the uniform traffic, it can be seen that the optimal value for packet rate weight is 0.5 with average packet drop ratio 0%. At $W_p = 0.6$ the average packet drop is as low as the 0.5 case. However, the average packet delay time of 0.6 case is higher then 0.5 case around 17ms. At other values of packet rate weight, the average packet drop ratio and the average packet delay time appear to have no significant difference.

In the 2 low : 1 high traffic profile, there are 20 low packet

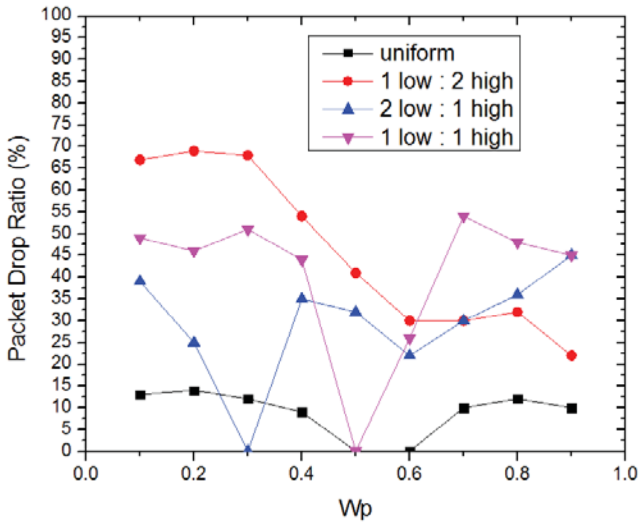

Fig. 3 The average packet drop ratio.

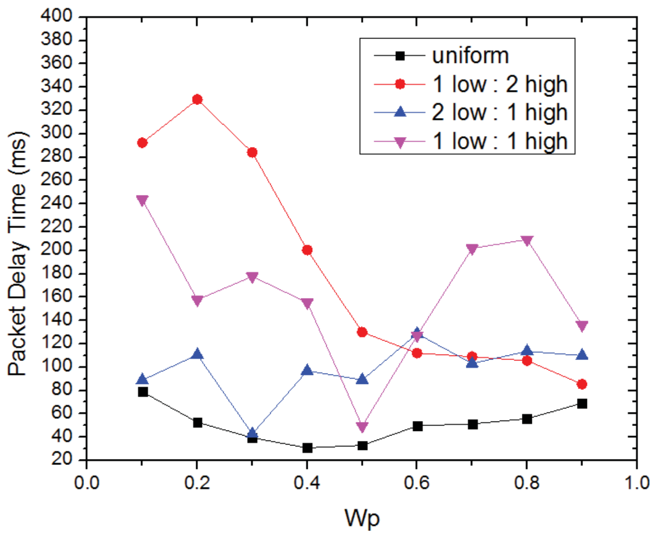

Fig. 4 The average packet delay time.

rate flows and 10 high packet rate flows. The optimal value for packet rate weight is 0.3 with the lowest average packet drop ratio 0% and the lowest average packet delay time 42ms. It can be seen that the active high packet rate flow is 10 which is smaller than flow table size. Therefore, the number of byte per packet seems to be more important in this case.

As the packet rates of traffic flows get higher and more diverse, the performance of the algorithm starts to degrade with more packet drop and higher packet delay time. The impact of the packet rate weight parameter also become more significant.

In case of 1 low : 1 high traffic profile, there are 15 low packet rate flows and 15 high packet rate flows. The optimal value for packet rate weight is clearly 0.5. As can be seen in figure 3 and figure 4, at W_p=0.5, the average packet drop ratio and the average packet delay time has the lowest values of 0% and 49ms respectively.

In the last case, the 1 low : 2 high traffic profile has 10 low packet rate flows and 20 high packet rate flows, hence makes the impact of packet rate feature really significant. It can be seen from figure 3 and figure 4, as the packet rate weight increases to the highest value, the average packet drop ratio and the average packet delay rate decrease.

The optimal values are summarized in table 2.

Table II. The optimal value for each traffic profiles

Traffic profile	Optimal value of W_p
Uniform	0.5
1 low : 1 high	0.5
2 low : 1 high	0.3
1 low : 2 high	0.9

V. Conclusion

In this paper, we studied and analyzed the weight parameter of traffic load aware replacement algorithm for SDN flow table cache in different network traffic scenarios. To this end, we presented our experiment and obtained results in detail as well as our comment and discussion. The results indicate that our hypothesis is correct. Choosing the optimal value of packet rate weight for each network traffic types is important in traffic load aware replacement algorithm. Especially in the case of heavy network traffics with diverse packet rate, the optimal value of packet rate weight can really improve the performance of the algorithm.

Acknowledgment

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2014R1A1A1007734).

References

[1] "Software-Defined Networking (SDN) Definition", Available at: https://www.opennetworking.org.

[2] "TCAMs and OpenFlow – What Every SDN Practitioner Must Know", Available at: https://www.sdxcentral.com, 2012.

[3] S. Banerjee and K. Kannan, "Tag-in-tag: Efficient flow table management in sdn switches.," CNSM, ed. D. Raz, M. Nogueira, E.R.M. Madeira, B. Jennings, L.Z. Granville, and L.P. Gaspary, pp.109–117, *IEEE*, 2014.

[4] M. Yu, J. Rexford, M.J. Freedman, and J. Wang, "Scalable Flow-Based Networking with DIFANE", *SIGCOMM Comput. Commun. Rev.*, vol.40, no.4, pp.351–362, Aug. 2010.

[5] A.R. Curtis, J.C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "Devoflow: Scaling flow management for high performance networks", Proceedings of the *ACM SIGCOMM 2011 Conference*, SIGCOMM '11, New York, NY, USA, pp.254–265, ACM, 2011.

[6] B.S. Lee, R. Kanagavelu, and K. Aung, "An efficient flow cache algorithm with improved fairness in software-defined data center networks", *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on*, pp.18–24, Nov 2013.

[7] N. Katta, O. Alipourfard, J. Rexford, and D. Walker, "Infinite cacheflow in software-defined networks", *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, New York, NY, USA, pp.175–180, ACM, 2014.

[8] N.T.T Hiep, "Traffic Load Aware Flow Table Cache for a Scalable SDN", unpublished, September 2015.

[9] V.D Tiep, N.T.T Hiep, K. Kyungbaek, "Evaluation of Replacement Algorithms of SDN Flow Table Cache for Heavy Network Traffics", *KISM Fall 2015 Conference*, Chosun University.

[10] OpenDaylight, "https://www.opendaylight.org", Accessed: 2015-09-01.

[11] Mininet, "http://mininet.org", Accessed: 2015-09-01.