Joint Allocation and Scheduling of Network Resource for Multiple Control Applications in SDN

Tao Feng, Jun Bi, Ke Wang
Institute for Network Sciences and Cyberspace, Tsinghua University
Department of Computer Science and Technology, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology
fengt09@mails.tsinghua.edu.cn, junbi@tsinghua.edu.cn, wang-k10@mails.tsinghua.edu.cn

***Abstract*—The network resource allocation in SDN for control applications is becoming a key problem in the near future because of the conflict between the need of the flow-level flexibility control and the limited capacity of flow table. Based on the analysis of the difference of the definition of network resource between SDN and traditional IP network, the idea of the integrated allocation of link bandwidth and flow table for multiple control applications in SDN is proposed in this paper. Furthermore, a price-based joint allocation model of network resource in SDN is built by introducing the price for each of the resources, which can get the proportional fair allocation of link bandwidth and the minimum global delay at the same time. We have also designed a popular flow scheduling policy based on the proportional fair allocation of link bandwidth in order to achieve the minimum global delay. A flow scheduling module has been implemented and evaluated in Floodlight, named virtual forwarding space (VFS). VFS can not only implement the fair allocation of link bandwidth and minimum delay flow scheduling in data plane but also accelerate packet forwarding by looking up flow cache in control plane.**

Keywords—resource allocation; SDN; multiple applications

I. INTRODUCTION

The emergence of Software Defined Networking (SDN) enables continuous network innovation. SDN is a network architecture which decouples the control plane from the forwarding plane [1]. From the perspective of SDN, the forwarding plane consists of high-performance commodity switches while the control plane is refactored to an open and centralized control platform for the deployment and co-existence of multiple control protocols to control packets forwarding through a standard forwarding abstract interface (e.g. the OpenFlow protocol).

OpenFlow-based SDN have been applied to many scenarios, such as campus network [2], data center network (DCN) [3], and even inter-DCN WAN [4, 5]. SDN enable these abilities of load balance, flows filtering and traffic engineering into flow-level control leveraging by a global network view. From these cases, we observe that SDN provide more flexible and fine-grained control ability for network applications. SDN applications can signal control instructions to filter packets as a firewall and optimize forwarding path according to network traffic distribution.

In the current network, network ability mainly refers to how much a network transmits data packets at a period of time. This kind of network ability is explicitly expressed by forwarding one which is determined by link bandwidth. More bandwidth means higher transmission rate and thus lower transmission delay. Therefore, bandwidth become the most important network resource due to the limitation of link bandwidth in the consideration of multiple applications sharing and competition.

In a SDN-enable network, forwarding ability is not the only one but control ability can explicitly provide the other interesting one for network applications with a standard control protocol, such as OpenFlow. OpenFlow switches use *flow table* to cache network control rules. TCAM is currently a major implementation of flow table for hardware-based looking up and forwarding. However, this kind of special-purpose memory is costly and energy eager so that the capacity of memory cannot be very large. Current generation of OpenFlow switches can support almost 1K flow entries. Since the destination address based one-dimension looking up and forwarding in the traditional IP network faces the challenge of expandability, then a multi-dimension flow-level caching in SDN will surely cause "flow-explosion". Switches with limit forwarding capacity are far from enough to accommodate all of network control rules. Therefore, flow table, besides of bandwidth, will become the other key network resource to impact control abilities in consideration of the limit capacity of current OpenFlow switches.

In this paper, we firstly propose a different perspective of network resource allocation between SDN and IP network. Flow table and bandwidth constitute the essential network resource in SDN. Secondly, we design a price-based joint allocation model and a fair allocation algorithm of bandwidth and flow table for multiple control applications in SDN. With the evaluation of the resource allocation algorithm with the real traffic of CERNET and one of ISP in China, the algorithm can guarantee the fairness of network resource allocation and,

978-1-4799-0913-1/14/$31.00 ©2014 IEEE

furthermore, realize the maximization of mean forwarding rate as well as the minimization of mean delay time.

II. Related Works

The objective of forwarding resource allocation is different from the network and application's view. From the perspective of network, network throughput rate and link utility ratio are usually regarded as the optimization objective which is aimed to realize the maximization of network resources utilization. This can be regarded as network-centered resource allocation objective. From the perspective of control applications, the optimization objective is to maximize the utility of the satisfaction of every control application to accomplish specific packet process with allocated network resource. This can be regarded as application-centered forwarding resource allocation objective.

A. Network-centered Forwarding Resource Allocation

Network-centered forwarding resource allocation deems network resource utilization maximization as the objective, uniformly treats control rules generated by different control applications, amounting to that a control application processes different flow. Traffic-aware Flow Offloading, named TFO, selects heavy-hit flows in different time scales with Zipf distribution. These selected flows will be issued to the switch and others will be forwarded by the controller [6]. Decoupling control and global view, DevoFlow detects and controls significant flows in order to reduce control overhead [7]. It realizes control devolvement to switches with rule clone and local behavior.

B. Application-centered Forwarding Resource Allocation

Another important objective of network resource allocation is to achieve fair allocation of network resources for control applications. The fair criterion of resource allocation includes max-min fairness policy and proportional fairness policy etc. Max-min fairness policy can achieve better utility of resource than conserving equal sharing policy. Proportional fairness policy can allocate resource according to the same cost to each user. Fair allocation of resources is mainly applied in the spectrum management in wireless communication and cloud computing. The objective of resource allocation in wireless communication is usually to select better wireless channel to transmit more packets of a user. The objective of resource allocation in cloud computing is usually to achieve better SLA based on resource allocation of network, compute and store. According to "deadline flows" and "elastic flows" in a data center, the paper compares three types of resource allocation: Random (RND), or Oblivious, Greedy with local knowledge (GLK), Greedy with global knowledge (GGK). The result shows the fact that algorithm with local knowledge performs better than naive random algorithm [8]. To void the starvation of "deadline flows ", D^3 proposes a deadline-aware control protocol that is customized for the datacenter environment in order to make the network aware of flow deadlines by prioritize flows based on deadlines.

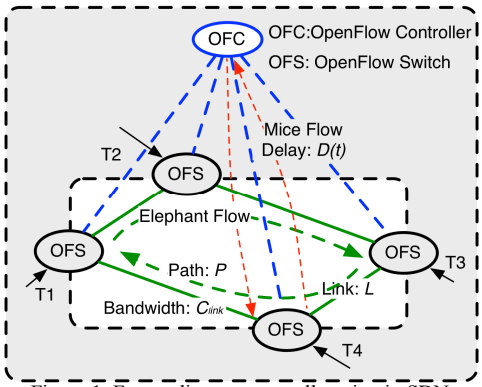

Figure 1. Forwarding resource allocation in SDN

III. Analyzing and Modeling Resource Allocation in SDN

A. Problem Statement

To effectively allocate network resource, MPLS TE is a current popular operation using many tunnels between ingress-egress router pairs. The tunnels of MPLS TE are marked with priorities and different types of services. Switches can map different DSCP bits in the IP header to different priorities queues. However, there are two problem of MPLS TE to allocate network capacity. One is poor efficiency caused by a local, greedy resource allocation model of MPLS TE. The other is poor sharing caused by the lack of network-wide fairness. Therefore, our goal of forwarding plane is to carry more traffic and support flexible network-wide sharing , as shown in Fig. 1. In Fig. 1, OpenFlow switches (OFS) are connected in the data plane of SDN while an OpenFlow controller (OFC) is running in the control plane of SDN. The centralized OFC estimates service demands and computes shortest forwarding paths for each service according to a global network view and services weight (e.g. the value or priorities of a service). When services weight are equal, the OFC will allocate bandwidth in the max-min manner.

On the other hand, because the flow table capacity of every OpenFlow switch is limited, flow table in OpenFlow switches cannot cache all of flows generated by control applications. Internet traffic feature with *power law* distribution means a

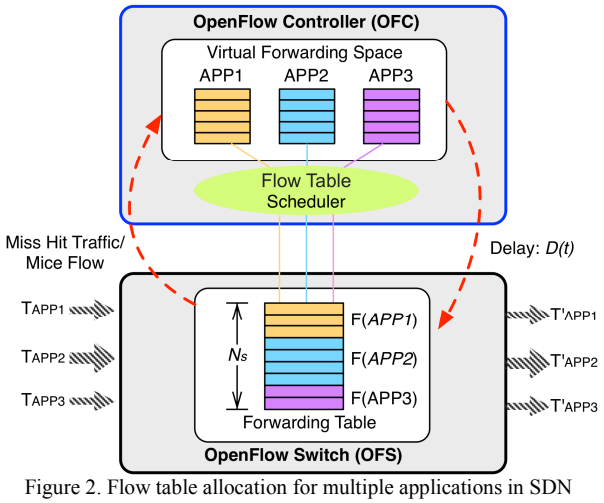

Figure 2. Flow table allocation for multiple applications in SDN

few popular prefixes will contribute most of traffic load and the rest prefixes have only trivial or even no traffic for a certain period of time[14-18]. Then, sacrificing some forwarding performance and loading some popular flows into switches is a trade-off in the condition of the limitation of flow table. Meanwhile, experimental research [9,10] show the popularity of a prefix only has short term stability. Because the traffic on a prefix can be very burst, and a popular prefix cannot have heavy traffic loads for all the time, a dynamic scheduling for flow table is required in an interval. During the interval, unmatched flows will be sent to the controller and redirected to output port through the control plane. The traffic volume of a flow across a SDN network is invariable as long as the path is fixed during a scheduling interval. This process of redirection will only introduce the delay of transmission from the switch to the controller compared to forwarding directly in the data plane. More detailedly, the delay of such process in SDN consists of the transmission time of unmatched packets to the controller, the computing time of control application for the packets and the transmission time back to switch. Keeping flow entries in the hardware flow table will therefore avoid such delay of packet forwarding, shown as in Fig. 2.

Generally speaking, not only link bandwidth but also flow table capacity should be taken into consideration in the network resource management of SDN. Both of these two resources should be allocated according to bandwidth cost and delay cost with the pay-for-use of different control applications. More bandwidth means higher transmission rate, while more flow table capacity means lower transmission delay.

B. Modeling Network Resouce Allocation in SDN

The fair criterion of resource allocation for multiple control applications in SDN is to guarantee the proportional allocation of link bandwidth and flow table capacity to the money which a user paid for bandwidth and flow table. The objective of efficiency is to guarantee the maximum traffic volume for each control application leveraging the proportional fairness and the minimum global delay through the selection of heavy-hit flows kept in hardware flow table.

We defined two prices ω_a, ν_a for link bandwidth and flow table respectively. The price ω_a will enable the proportional fairness of link bandwidth allocation while the price ν_a will adjust the delay of each flow. The notations of resource allocation model are shown in Table 1. The optimization objective of the bandwidth allocation in SDN is to maximize the sum of the logarithmic rate of each control application.

1) Bandwidth Allocation in SDN

The bandwidth allocation in SDN is aimed to maximize the rate of a control application based on fairness allocation of link bandwidth. The fairness criterion is the equation of the price per flow, which means the rate of a control application is proportional to the price of the bandwidth of the control application. Therefore, the optimization objective of the bandwidth allocation in SDN, as shown in the expression 3.1, is to maximize the sum of the logarithmic rate of each control application [11].

TABLE I. NOTATIONS OF RESOURCE ALLOCATION MODEL

Symbol	Meanings
A	Control applications set. Each element is a control application
L	Links set. Each element is a link
S	OpenFlow switches set. Each element is a switch
P	Path set. Each element is a path
ρ_{sp}	The indicator variable of a flow p whether it is in the hardware flow table
ω_a	The price of bandwidth which is paid by a control application a
υ_p	The price of flow table which is paid by a control application a
C_l	The bandwidth of a link l
N_s	The maximum capacity of a flow table.

$$\max \sum_{a:a\in A} \ln \omega_a y_a(t), \forall a \in A \tag{3.1}$$

$$\text{subject to } a) \sum_{p:p\in P(a)} x_{ap}(t) = y_a(t), \forall a \in A \tag{3.2}$$

$$b) \sum_{p:p\in P(l)} x_{ap}(t) \le C_l, \forall l \in L \tag{3.3}$$

$$c) x_{ap}(t), y_a(t) \ge 0, a \in A, p \in P \tag{3.4}$$

$P(a)$ notes the path sets of a giving application. $P(l)$ notes the path sets of a giving link. The rate of a control application 'a' satisfies $y_a(t) = \sum_{p:p\in P(a)} x_{ap}(t)$. The sum of flows over link l should not exceed the capacity of link capacity $\sum_{p:p\in P(l)} x_{ap}(t) \le C_l$. There exists a unique optimum for the rate vector $y_a(t)$ for each control application because the objective function (3.1) is a strictly convex function of $y_a(t)$, but there may be many corresponding values of the flow rate $x_{ap}(t)$ satisfying the relations (3.2) and (3.3).

2) FlowTable Allocation in SDN

The transmission rate of each control application along each path is given by the link bandwidth allocation. Ideally, after the first step of optimization which determines the flow volume on each path, everything should be set and no more job is needed. However, since not all of the flows can be forwarded in the data plane due to the limitation of flow table capacity, some flow will have to be redirected to the controller which will then send packets to output port. Such process of redirection will increase packet delay. Therefore, the second step of optimization is required to achieve the minimal mean

delay. From the view of network, heavy-hit flows will be chosen to maximize the throughput of an OpenFlow switch. However, it is improper as some micro flows will then have little chance to be forwarded by the hardware flow table even though some of them may be "deadline flow", such as Map-Reduce packets. Therefore, a weight v is introduced to identify the importance of a certain flow to avoid micro flows starvation. This parameter v can be viewed as the cost of flow table capacity. Accordingly, the optimization model is modified as following, note there is a v_p associated with each flow.

$$min\, d_s(t) \tag{3.5}$$

$$a)\ d_s(t) = \frac{\sum_{p \in p(s)} \rho_p v x_p(t)}{\sum_{p \in p(s)} x_p(t)} \tag{3.6}$$

$$b)\ \sum_{p:p \in P(s)} \rho_p \leq N_s \tag{3.7}$$

$$c)\ \rho_p \in \{0,1\}, p \in P \tag{3.8}$$

Theorem I *The mean delay of SDN network is minimum if the mean delay of each OpenFlow switch is minimum.*

Proof. Considering a topology, the scheduling of a flow table cannot change the forwarding path after a path has be determined, but it will impact the packet delay. The global means delay is defined as

$$d_{global}(t) = \frac{\sum_{p:p \in P(s)} x_p d_p(t)}{\sum_{p:p \in P(s)} x_p} \tag{3.9}$$

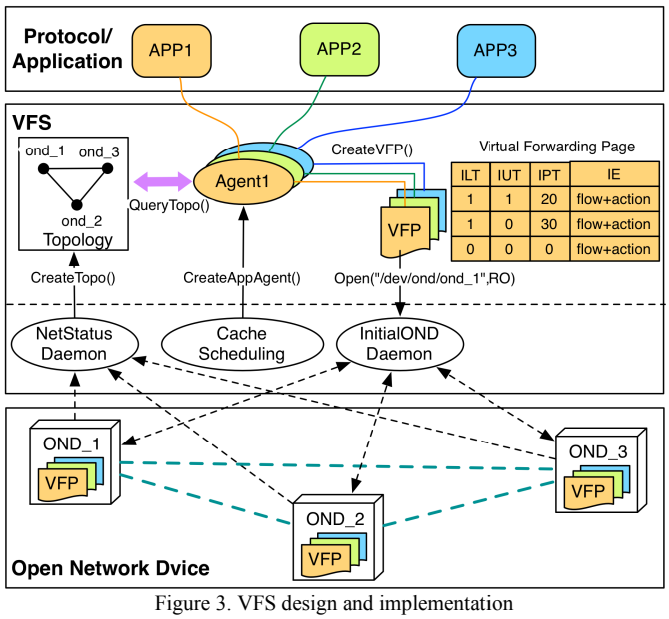

Figure 3. VFS design and implementation

The numerator can be also expressed as $\sum_{p:p \in P} x_p \sum_{s:s \in S(p)} d_{sp}(t)$. This indicates that the total delay of a single flow is the sum of delay introduced by each switch along the path of this flow. Changing the order of sum, the global mean delay can be rewritten as:

$$d_{global}(t) = \frac{\sum_s \sum_{p:p \in P(s)} x_p d_{sp}(t)}{\sum_{p:p \in P(s)} x_p} \tag{3.10}$$

It can be easily seen that the mean delay of each OpenFlow switch reaching the minimum equals to $\sum_{p:p \in P(s)} x_p d_{sp}(t)$ reaching the minimul. Sum up this expression over each switch s, we can see the $d_{global}(t)$ also reaches the minimul. ■

Therefore, According to **Theorem I**, The global optimal mean delay can be reached with the distributed scheduling policy of flow table.

IV. VFS Design and Implementation

To implement the allocation and scheduling of network resource for multiple applications in SDN, we have designed and implemented a functional module based on Floodlight [12], named Virtual Forwarding Space (VFS), as shown as Fig.3. VFS uses the idea of virtual memory management in computer science. VFS allocate a block memory in the controller server, named virtual forwarding page (VFP), to cache control instructions for each control application. VFS can schedule all of control instructions in VFPs into OpenFlow switches according to the hit-rate of a control instruction, the constraint of link bandwidth and the forwarding path of a flow.

A. Virtual Forwarding Page

VFP is an application-indexed table which is consisted of instructions loaded tag (ILT), instructions update tag (IUT), instructions popular tag (IPT), and instructions entry (IE).

ILT shows whether a control instruction has been loaded into a switch. If an instruction has been loaded, ILT will be set 1 by a cache scheduling module, otherwise ILT will be 0. IUT shows whether a control instruction has been updated by its control application. If an instruction has been updated and ILT is 1, IUT will be set 1 by the control application. IPT shows how much a control instruction is popular, which is determined by the hit-rate of a control instruction. Instructions entry field is a flow entry and its corresponding actions.

When a control application starts running in the controller, VFS will initial an agent daemon for the application using *CreateAppAgent* API and, meanwhile, allocate a block memory from the RAM of the server using *CreateVFP* API. When a control application generates a FLOW_MOD message to control packets forwarding, the message will be cached into VFP using *WriteControlRules* API. ILT, IUT and IPT are initialized to 0.

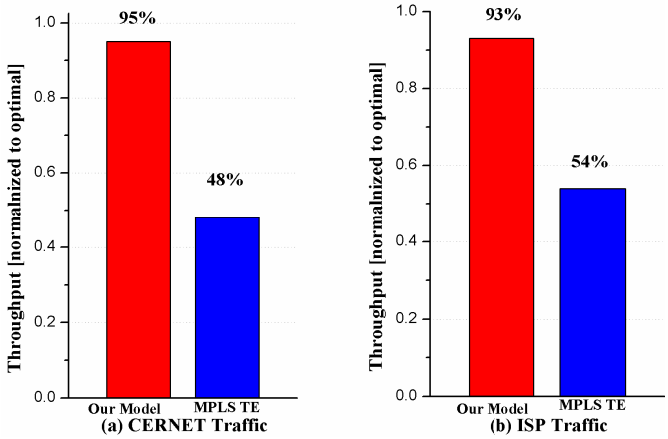

Figure 4. Different cache scheduling over two traces

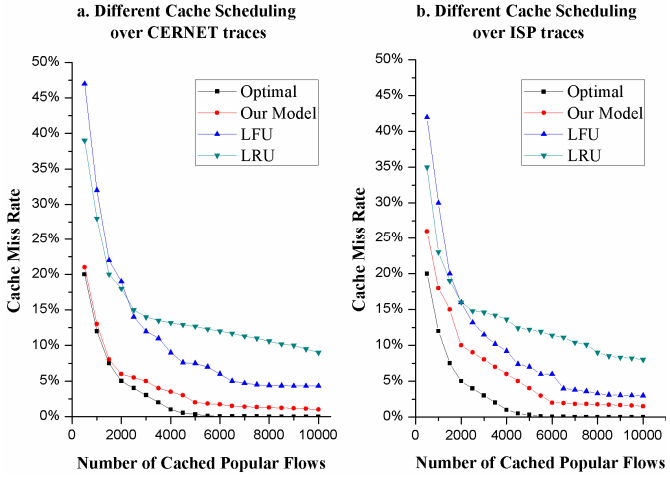

Figure 5. Different cache scheduling over two traces

B. *Bandwidth Allocation*

Bandwidth allocation in VFS use meter table to monitor and configure traffic rate per flow. In OpenFlow specification, a meter table consists of per-flow meter entries to implement rate-limiting or QoS. APP agents in VFS can assign the rate of flow according to the bandwidth of per-application.

C. *Cache Scheduling*

Cache scheduling module will compute flow allocation for each OpenFlow switch according to the volume of flows at the current time slot. The flows with large volume will be downloaded to switches until the flow table in switches is full.

D. *Unmatched Packets Forwarding*

Unmatched packets will be sent to the controller by an OpenFlow switch. VFS will receive the packet and conduct a looking-up in the VPF to find a match. Then VFS will redirect the packet into the output port of the first hop switch. Thus, the computing time for forwarding path for the unmatched packets is avoided.

I. EVALUATION

A. *Data-driven evaluation*

1) *Datasets*

Our evaluation uses two working traffic traces provided by CERNET and a top-level ISP of China. CERNET traces are sampled flow-level NetFlow traces in a period of two months at different backbone routers. Since the volume of the traffic transited by those routers is extremely large, the NetFlow sampling rate has been set as 1:5000 with the sampling techniques specified in [13]. In order to analyze the impact of sampling, we also collect flow-level traffic data of ISP.

2) *Network utilization*

To evaluate how much our method utilizes the network, we compare it to current practice, MPLS TE. VFS will search for better path periodically (5 minutes) and adjust bandwidth according to current traffic demand. We see that MPLS TE can only carry 50% of traffic. Our method can carry about 92% for both traces, as shown in Fig. 4. This difference means our method carries over 60% more traffic than MPLS TE thanks to a global computing of forwarding path based on a global network view in SDN.

3) *Cache replacement*

On evaluating the performance of cache scheduling of our method, we simulate all strategies at various cache size with CERNET t races and ISP traces. We plot average cache miss rate in Fig. 5 (a)-(b). From the simulation, LRU has better performance than LFU but some bursts in LRU will influence the flow cache efficiency from time to time and incur much higher overheads.

B. *Mininet-based evaluation*

We have simulated a network topology of Tsinghua campus, as shown in Fig. 6, with Mininet and generate the traffic with Iperf. We modified a forwarding application in Floodlight to generate control rules to process different traffic of control applications. In order to distinguish the flows of different control applications, we utilized VLAN tag as control application tag, and let different control applications be able to process respective campus network flows by adding different

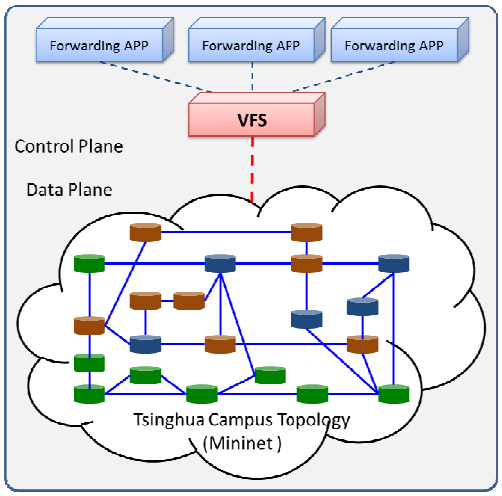

Figure 6. Hit rate comparison under two policies

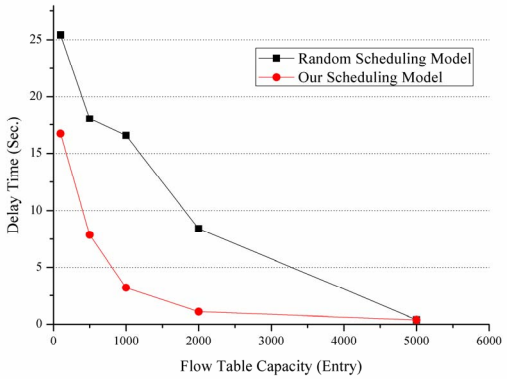

Figure 7. Delay time comparison under two policies

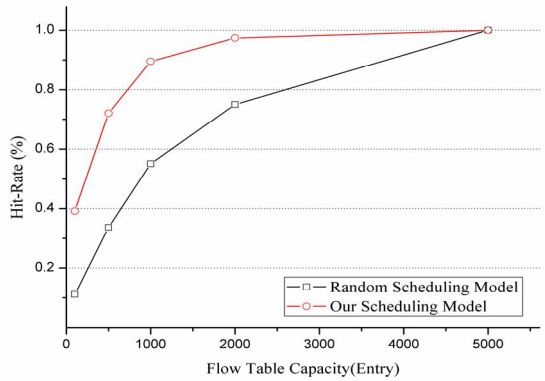

Figure 8. Hit rate comparison under two policies

VLAN tag in packets and allocating tag. Moreover, we defined the capacity parameter of the flow table size in a switch in Floodlight to limit the maximum capacity of the flow table in a switch so that it is easy to estimate the effects of resource allocation in different scenarios of capacity of the forwarding table in a switch.

The network topology is consisted of twenty-one OpenFlow switches. The link bandwidth of switches is different. Every edge switch is connected to four hosts which will generate packets using Iperf. Three control applications will process different flows in three VLANs respectively. First, source hosts select some specific target hosts to send packets. At that time, OpenFlow switch will receive these initial packets and send them to Floodlight. The control applications will compute forwarding path according to the destination address in the packet. VFS will then compute the bandwidth allocation for every control applications according to the forwarding path, the limitation of the link bandwidth and the relative weight of each application. The objective of resource allocation is that the bandwidth is proportional to the price of every application. After the bandwidth optimization, each path of a control application can be allocated with a fair link bandwidth as follows.

1) Comparison of Delay

VFS will decide each flow whether to be forwarded by hardware flow table or not according to the hit rate of a flow. If a flow is popular, it will be forwarded by hardware flow table directly. If a flow is not popular, it will be redirected to the controller. Through the measurement of the packets, we achieve the mean delay of every flow and the global mean delay of SDN network. To compare the global mean delay with deferent scheduling policy, we also implement a random scheduling policy, as shown in Fig. 7. The mean packet delay of the network with our scheduling model is lower than random scheduling model. When the size of flow table is very limited, this delay is large, as we increase the flow table size, both algorithm can produce less delay time. As we reach another extreme, that is, the flow table size is unlimited, both algorithm generate the same result since now there is no flow selection involved.

2) Comparison of Hit-Rate

Fig. 8 depicted that our scheduling model can achieve higher hit rate than random scheduling model. When flow table capacity is 2000 entries, our scheduling model can achieve nearly 100% hit rate rather than 70% of random scheduling model. Meanwhile, the scheduling policy based on the hit rate will result in the starvation of micro flows so a weight parameter will be introduced in (3.6) to avoid the starvation of "deadline flow".

II. CONCLUSION AND FUTURE WORK

A. Conclusions

Due to the characteristic of centralized control in SDN, we analyze the difference of the definition of network resource between SDN and traditional IP network. In traditional IP network, link bandwidth is the key resource as there are many bandwidth-consuming applications, for example, video on-demand, such as Netflix, Hulu. Nevertheless, in SDN, not only link bandwidth but also flow table capacity should be taken into consideration as valuable resource because the ability of specific and fine grain flow control is indispensable. To implement this broader view of resource management, a price-based joint allocation of network resource in SDN is proposed.

The contributions of this paper are as follows:

- The idea of the integral management of link bandwidth and flow table for multiple control applications in SDN. The resource allocation in SDN for control applications will become a crucial problem in the near future because of the conflict between the flow-level flexibility control and the limited capacity of flow table. Therefore, this new perspective of integral network resource in SDN is essential under the new environment.
- A price-based joint allocation model in SDN. By introducing the price of both the link bandwidth and flow table, we can achieve a proportional fair allocation of link bandwidth and the minimal global delay at the same time.
- Popular flow scheduling policy based on the proportional fair allocation of link bandwidth. According to the hit rate on a packet basis, the popularity of each flow can be determined. Through our

distributed policy, each switch can acquire a minimal delay, and according to Theorem I, the minimal global delay is then guaranteed. A flow scheduling module based on this popular flow policy is implemented in TUNOS [19], which has been demonstrated in ONS 2013.

B. Future Work

In our current analysis, we assumed that the bandwidth between an OpenFlow switch and a controller is unlimited. Under this assumption, we eliminate packet loss at the expense of increasing packet delay. However, in a more real environment, this bandwidth is finite which will cause possible packet loss. The larger this bandwidth is, the less the packet loss rate will be. In other words, the bandwidth between OpenFlow switches and the controller is also a valuable resource. In the next step, this kind of resource will be taken into consideration as well as the former two we have analyzed above in order to make the model more complete.

Besides, the price of link bandwidth and flow table is now charged respectively. In the future work, we will find a reasonable link between this two and establish a proper relationship, which can also be adjusted according to the actual situation of a network operator.

ACKNOWLEDGMENT

This work is supported by the National High-tech R&D Program ("863" Program) of China (No.2013AA010605), the National Science Foundation of China (No.61161140454), and National Science & Technology Pillar Program of China (No.2012BAH01B01). Jun Bi is the corresponding author.

REFERENCES

[1] Software-Defined Networking: The New Norm for Networks. ONF white paper. 2012.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev., 38(2): 69–74, 2008.

[3] M. Jarschel, R. Pries. An OpenFlow-Based Energy-Efficient Data Center Approach. SIGCOMM 2012.

[4] C. Hong , S. Kandula and R. Mahajan. Achieving High Utilization with Software-Driven WAN. SIGCOMM 2013

[5] S.Jain, A. Kumar, S. Mandal. B4: Experience with a Globally-Deployed Software Defined WAN. SIGCOMM 2013

[6] D Kuptsov. Demand-aware flow allocation in data center networks. CSWS'12

[7] J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. sharma, A. R. Curtis, and S. Banerjee. Devoflow: cost-effective flow management for high performance enterprise networks. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Hotnets-IX, p. 1:1-1:6, USA, 2010

[8] C Wilson, H Ballani, T Karagiannis. Demand-Aware Flow Allocation in Data Center Networks ACM SIGCOMM Computer Communcation Review, 2011.

[9] C. Park et al., Long-range dependence in a changing Internet traffic mix, Computer Networks 48 (3) (2005) 401–422.

[10] E. Kohler et al., Transactions on Networks 14 (6) (2006) 1207–1218.

[11] F. Kelly. Charging and rate control for elastic traffic. European Transactions on Telecommunications. 8(1): 33-37, 1997.

[12] FloodLight Web Site. http://www.projectfloodlight.org/floodlight/

[13] Sampled NetFlow, Cisco Systems. http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/12s_sanf.html

[14] C. Williamson, Internet traffic measurement, IEEE Internet Computing 5 (6) (2001) 70–74.

[15] M. Arlitt, C. Williamson, Internet Web servers: workload characterization and performance implications, IEEE/ACM Transactions on Networking 5 (5) (1997) 815–826

[16] L. Breslau et al., Web caching and Zipf-like distributions: evidence and implications, in: Proceedings of the IEEE INFOCOM, 1999.

[17] M. Crovella, A. Bestavros, Self-Similarity in World Wide Web traffic: evidence and possible causes, IEEE/ACM Transactions on Networking 5 (6) (1997) 835–846.

[18] W. Fang, L. Peterson, Inter-AS traffic patterns and their implications, in: Proceedings of the Global Internet, 1999.

[19] T. Feng, J. Bi, H. Hu. TUNOS: A Novel SDN-oriented Networking Operating System. ICNP Poster, 2012