

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

FENG Tao<sup>1,2,3</sup>, BI Jun<sup>1,3</sup>, WANG Ke<sup>4</sup>

<sup>1</sup> Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China

<sup>2</sup> Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

<sup>3</sup> Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, China

<sup>4</sup> Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

**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 control rules in control plane.

**Keywords:** resource allocation; SDN; multiple applications

## I. INTRODUCTION

Software Defined Networking (SDN) is a network architecture which decouples the control plane from the forwarding plane by abstracting network forwarding actions and defining standard network control protocols [1]. SDN has become such an important network architecture, and it enables the innovation of networking functions more agile. SDN has been applied to campus network [2], data center network (DCN) [3] and even inter-DCN WAN [4, 5]. All these cases of practical deployment and operation are demanding SDN to support fast and reliable communications.

To build a high-performance SDN, it is a key challenge to allocate and schedule the network resources of SDN dynamically and efficiently. Resource management of SDN is the same as traditional networks from the view of the goal of optimization which is to improve network utilization and to guarantee quality of service by increasing network throughput and declining transport delay. However, compared to traditional networks, this allocation and scheduling in SDN is much harder because of the reasons listed below.

1) *The granularity of resource allocation and scheduling.* SDN provides standard control interfaces and dynamic global views for

more flexible and fine-grained control abilities. SDN enable these abilities of load balance [20, 21], flows filtering [22] and traffic engineering [4, 5] into flow-level network control. Applications can signal control instructions to filter packets as a firewall and optimize forwarding path according to network traffic distribution. Therefore, the allocation and scheduling of network resource in SDN is based on some single flows or aggregate flows.

2) *The view of network resource allocation.* SDN architecture is divided with three layers: an infrastructure plane with open network elements, a control plane with a logically centralized controller and an application plane with network control programs [23]. Control plane and application plane in SDN can deployed and operated independently outside network devices rather than locking in a box in traditional networks. Control applications manipulate network packet forwarding by means of flow tables. The hardware implementaton of a flow table in a network device has limited capacity while these applications share a common flow table space. Therefore, flow table is a unique network resource to reflect control abilities for control applications in SDN. SDN can allocate and schedule not only fowarding related network resource (bandwidth) but also control related network resource (flow tables).

3) *The mechanism of network resource scheduling.* The control mechanism of SDN is that the packet or its header which is not matched in the forwarding plane will be sent to a controller while a control applicaton running upon the controller will gernerate forwarding rules according the packet and issue them into a flow table of the forwarding plane. However, when a flow table is full, new flow entries are not inserted in the flow table and this may cause a disruption of service. Therefore, flows caching and sheduling for all of control applications are required as basic characteristics of a control plane in SDN.

In this paper, we firstly analyze a different perspective of network resource allocation and scheduling between SDN and traditional networks. Flow table and bandwidth consti-

tute 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, 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.

### 2.1 Network-centered resource allocation and scheduling

Network-centered forwarding resource allocation is regarded 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

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.

to switches with rule clone and local behavior. Hedera delivers bisection bandwidth that is 96% of optimal and up to 113% better than static load-balancing methods by dynamic flow scheduling in data center networks [24]. B4, a centralized traffic engineering service, drives links to near 100% utilization in inter-DCN WAN.

## 2.2 Application-centered resource allocation and scheduling

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. FlowVisor can allocate statically bandwidth and flow table for multiple control applications with separated slices [25]. Fallocc propose a distributed algorithm for datacenter bandwidth allocation via a bargaining game approach to achieve the asymmetric Nash bargaining solution [27].

## III. ANALYZING AND MODELING RESOURCE ALLOCATION IN SDN

### 3.1 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

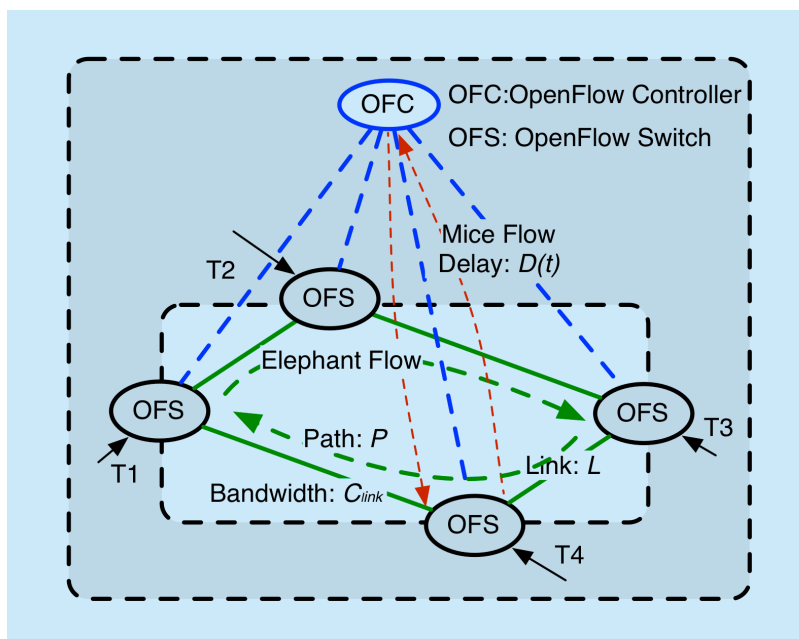


Fig.1 Forwarding resource allocation in SDN

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 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. 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.

### 3.2 Modeling network resource 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  $\omega_a$ ,  $\nu_p$  for link bandwidth and flow table respectively. The price  $\omega_a$  will enable the proportional fairness of link bandwidth allocation while the price  $\nu_p$  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.

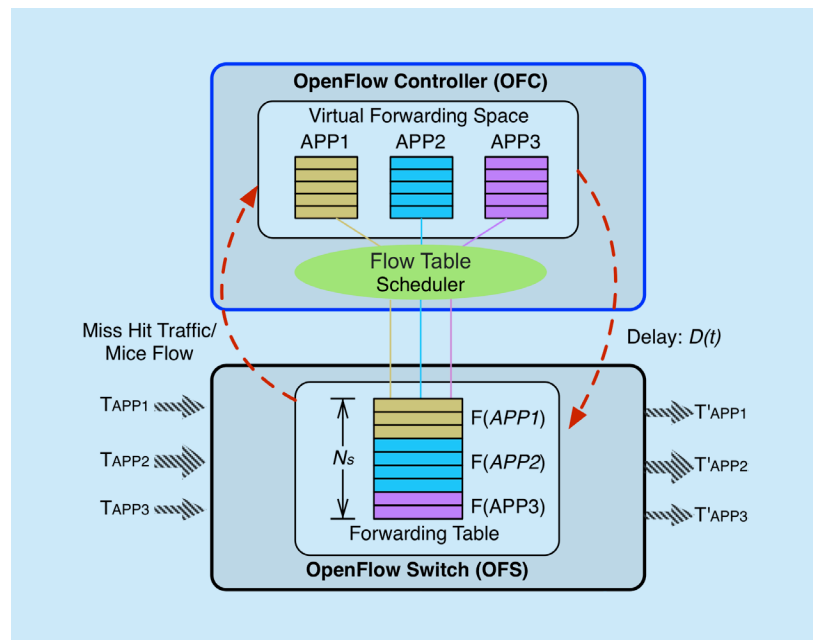


Fig.2 Flow table allocation for multiple applications in SDN

**Table 1** 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
$\rho_{sp}$	The indicator variable of a flow $p$ whether it is in the hardware flow table
$\omega_a$	The price of bandwidth which is paid by a control application $a$
$v_p$	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.

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].

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

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

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

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

$P(a)$  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  $\sum_{p: p \in P(l)} x_{ap}(t) \leq C_l$ . 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.

$$\min d_s(t) \quad (3.5)$$

$$a) d_s(t) = \frac{\sum_{p \in P(s)} \rho_p v_p x_p(t)}{\sum_{p \in P(s)} x_p(t)} \quad (3.6)$$

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

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

Some flows within a control application may be redirected to a controller and the controller will forward them under the condition of the satisfaction of the maximum of the throughput of a control application. At this time, if the mean delay of all of flows in a switch is minimum, then the global mean delay of all of the flows of SDN is minimum.

**Theorem 1** 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

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

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:

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

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 minimal. Sum up this expression over each switch  $s$ , we can see the  $d_{global}(t)$  also reaches the minimal.

Therefore, According to **Theorem 1**, The global optimal mean delay can be reached with the distributed scheduling policy of flow table.

#### IV. VFS DESIGN AND IMPLEMENTATION

To implement the fair allocation of link bandwidth and effective scheduling of flow table for multi control applications, we propose the concept of Virtual Forwarding Space (VFS) to manage network resource in SDN using the idea of virtual memory management in computer science. The components of VFS, the flow initialization and scheduling in VFS will be depicted in more detail below.

##### 4.1 The components of VFS

The components of VFS includes virtual forwarding page, flow performance table and flow status table. The design of VFS is shown as Fig. 3.

1) Virtual Forwarding Page (VFP): VFP stores flow entries generated by control applications. VFS will allocate a VFP for every control application. VFP consists of flow identification (FID), datapath identification (DID) and flow entries (FE). FID is a hash code of the control application identification which generates the flow and the flow match field. DID is the MAC address of a network device which a flow issue to. A flow entry is a flow discription and its corresponding actions.

2) Flow Performance Table (FPT): FPT stores flows statistics for very control applications. FPT consists of FID and flow statistics

(FS). FID is consistent with flow identification in VFP. FS is a packet counter which updates packets are processed by a flow.

3) Flow Status Table (FST): FST stores the scheduling status of a flow set from the view

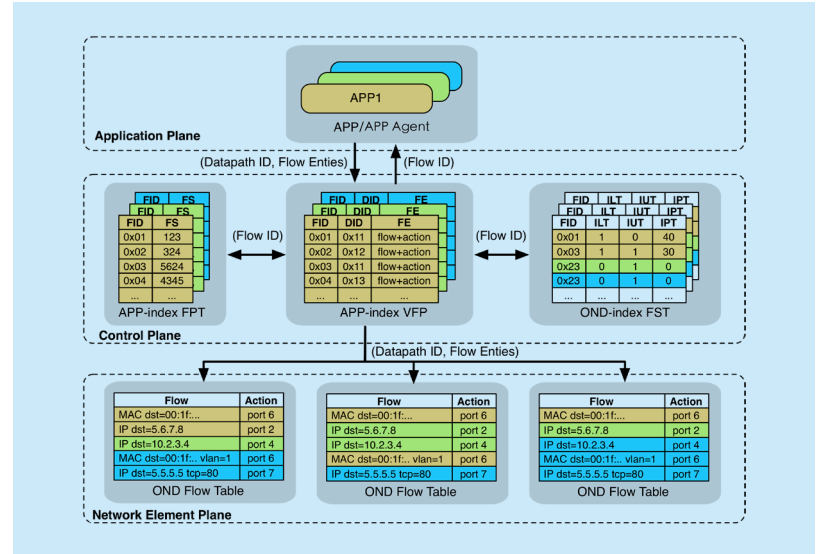


Fig.3 VFS design and implementation

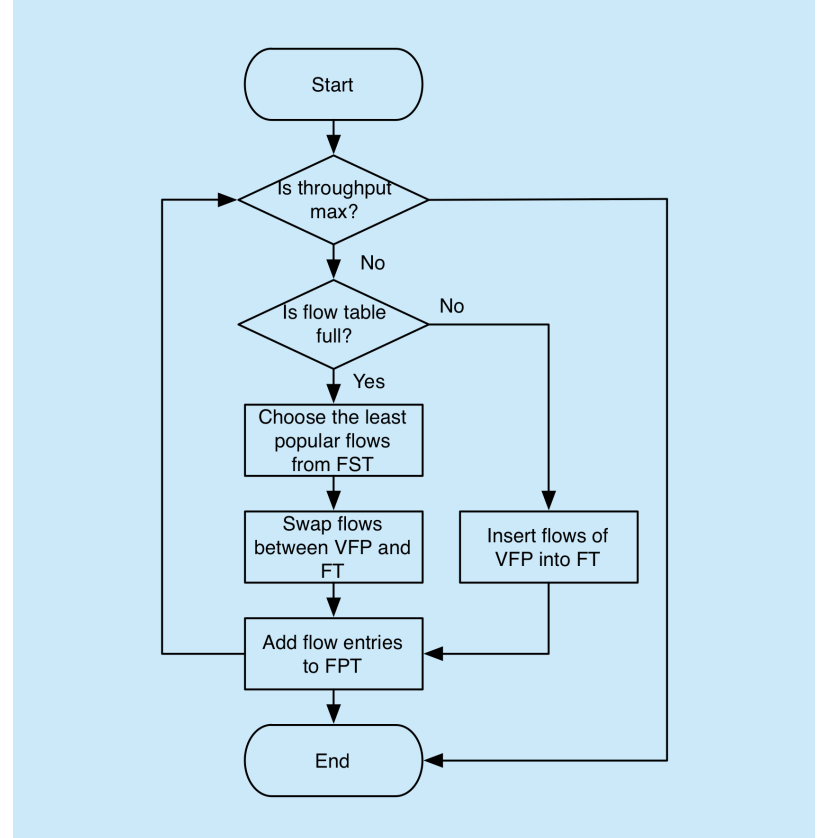


Fig.4 A flow chart of flow scheduling with VFS

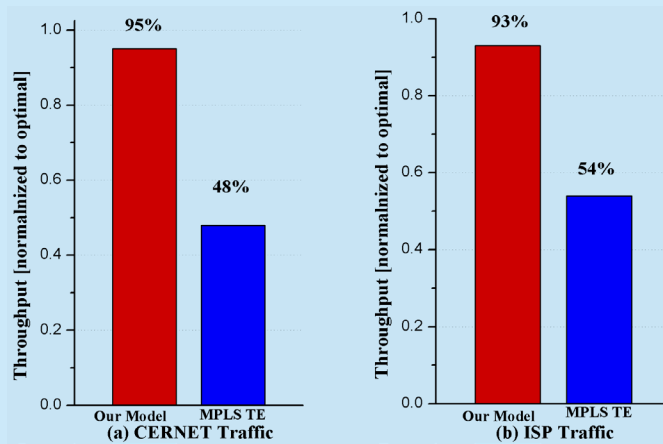


Fig.5 Different network utilization over two traces

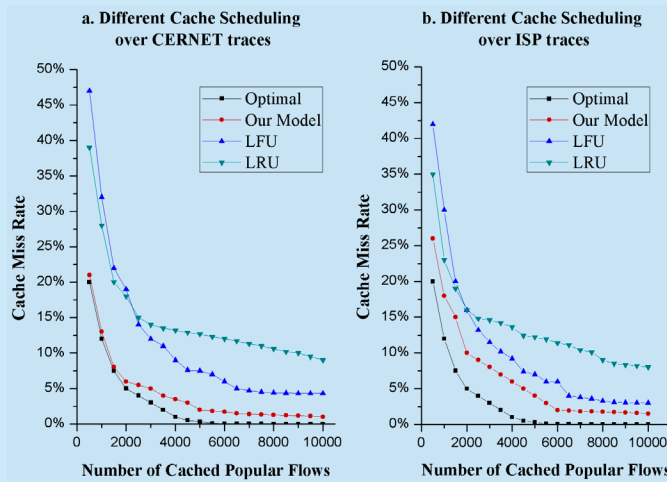


Fig.6 Different cache scheduling over two traces

of forwarding plane. FST is a network device indexed tables which consists of FID, instructions loaded tag (ILT), instructions update tag (IUT), instructions popular tag (IPT). FID is consistent with flow identification in VFP. 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.

## 4.2 The flow initialization in VFS

In SDN, when a packet is not matched with flow entries in the flow table of a network device, this packet or packet header will be sent to a controller. A control application running upon the controller will generate a series of flow entries to control packet forwarding. These new flow entries will firstly be stored into VFP and simultaneously a FID will be created and returned to the control application in VFS. After that, VFS uses the new FID to add a new flow status entry in FST and a new flow statistics entry in FPT.

## 4.3 The bandwidth allocation in VFS

The throughput of a control application in a forwarding plane is calculated by associating with FPT and FST in VFS. The real throughput of a control application in a forwarding plane is a summary of the flow statistics value in FPT of a set of FID whose ILT value equals 1 in FST. According to expression (3.1), we can get an allocated throughput of every control application. Then, a set of flows which are satisfied with the allocated throughput of the control application can be got through a greed algorithm. The traffic rate per flow can be monitored and configured using meter table in OpenFlow [26]. A meter table consists of per-flow meter entries to implement rate-limiting or QoS. VFS can assign the traffic rate of flow according to the throughput of per-application.

## 4.4 The flow scheduling in VFS

The flow scheduling in VFS is a series of procedures of flows swap from VFS to open network devices. Flow scheduling module will compute flow allocation for each OpenFlow switch according to the volume of a flow and the weight of a flow at the current time slot according to expression (3.5). The flows with large volume and high popular rate will be downloaded to switches until the flow table in switches is full. Fig. 4 shows a flow chart of

flow scheduling with VFS.

## V. EVALUATION

### 5.1 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. 5. 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 traces and ISP traces. We plot average cache miss rate in Fig. 6 (a)-(b). From the simulation, LRU has better performance than LFU but some bursts in LRU will influence the flow scheduling efficiency from time to time and incur much higher overheads.

### 5.2 Mininet-based evaluation

We have simulated a network topology of Tsinghua campus, as shown in Fig. 7, with Mininet and generate the traffic with Iperf. We modified a forwarding application in Floodlight to generate control rules to process dif-

ferent 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 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 consists 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

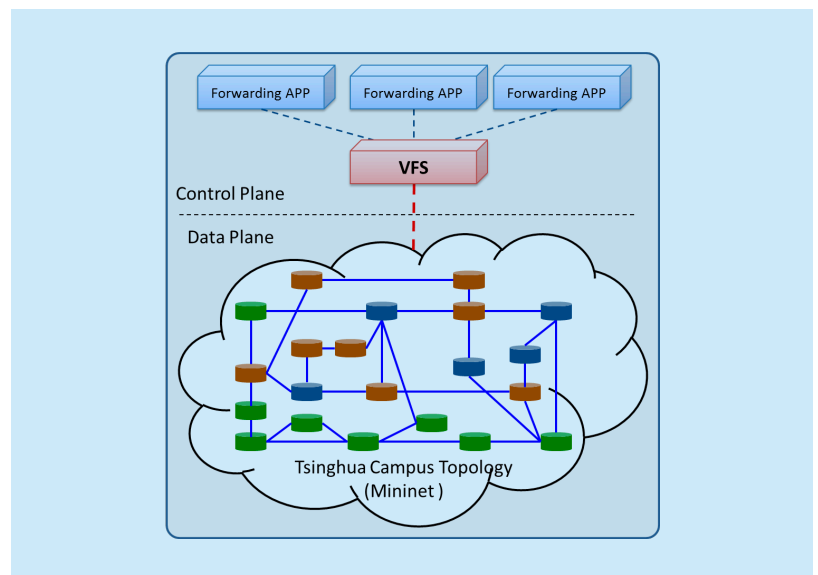
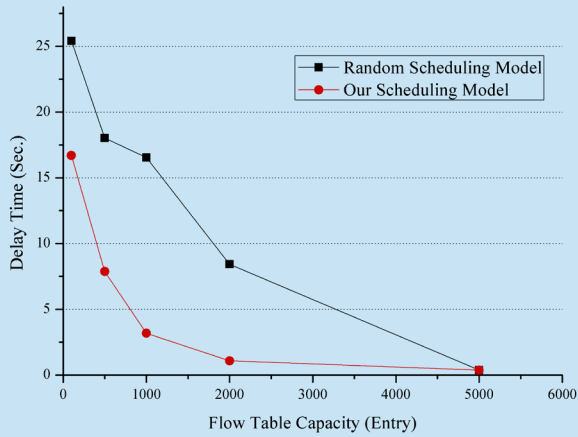
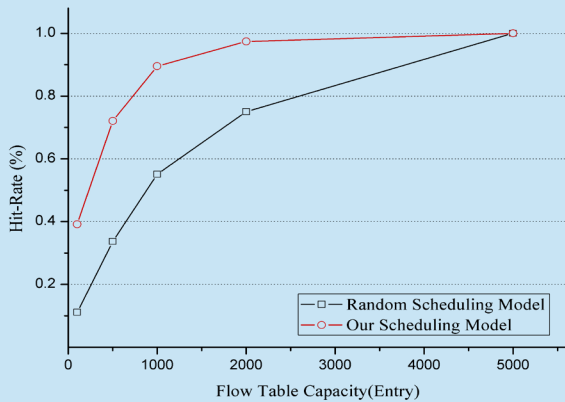


Fig.7 A network topology of Tsinghua campus with Mininet





**Fig.8** Delay time comparison under two policies



**Fig.9** Hit rate comparison under two policies

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. 8. 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. 9 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 expression (3.6) to avoid the starvation of “deadline flow”.

## VI. CONCLUSION AND FUTURE WORK

### 6.1 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 fol-

lows:

- 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.

## 6.2 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 rela-

tionship, which can also be adjusted according to the actual situation of a network operator.

## ACKNOWLEDGMENT

Supported by the National High-tech R&D Program (“863” Program) of China (No.2013AA013505), the National Science Foundation of China (No.61472213) and National Research Foundation of Korea (NRF 2014K1A1A2064649).

## References

- [1] Software-Defined Networking: The New Norm for Networks. Open Networking Foundation 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 Communication 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](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
- [20] N.I. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, R. Johari, Plug-n-Serve: Load-balancing Web Traffic using OpenFlow, in: *Proceedings of the ACM Sigcomm Demo*, 2009
- [21] R. Wang, D. Butnariu, J. Rexford, OpenFlow-based Server Load Balancing Gone Wild, in: *Proceedings of Hot-ICE*, 2011
- [22] A. Nayak, A. Reimers, N. Feamster, R. Clark, Resonance: Dynamic Access Control in Enterprise Networks, in: *Proceedings of WREN*, 2009
- [23] SDN Architecture Overview. V1.0. Open Networking Foundation. December, 2013
- [24] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, A. Vahdat, Hedera: Dynamic Flow Scheduling for Data Center Networks, *NSDI*, 2010
- [25] R. Sherwood, G. Gibb, K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, “Can the Production Network Be the Testbed?” in *Proc. of In Operating Systems Design and Implementation - OSDI 2010*, Vancouver, BC, Canada, October 2010.
- [26] OpenFlow Switch Specification, version 1.4.0. Open Networking Foundation, October, 2013
- [27] J. Guo, F. Liu, H. Tang, Y. Lian, H. Jin, J. C.S. Lui, “Falloc: Fair Network Bandwidth Allocation in IaaS Datacenters Via a Bargaining Game Approach”, in *Proc. of IEEE ICNP*, October, Goettingen, Germany, 2013.

## Biographies

**FENG Tao**, Ph.D. candidate with Department of Computer Science, Tsinghua University, China. His main research interests include network architecture, Software Defined Networking and network management. Email: fengt09@mails.tsinghua.edu.cn

**BI Jun**, The corresponding author, email: junbi@tsinghua.edu.cn. He is a professor and Ph.D. supervisor with Institute for Network Sciences and Cyberspace, Tsinghua University, China. His main research interests include Internet architecture and protocols, future Internet (SDN and NDN), Internet routing, and source address validation and traceback.

**WANG Ke**, Undergraduate student with Department of Electronic Engineering, Tsinghua University, China. His main research interests include network architecture and software defined networking. Email: wang-k10@mails.tsinghua.edu.cn.