Front. Comput. Sci., 2020, 14(4): 144501
https://doi.org/10.1007/s11704-019-8397-5

RESEARCH ARTICLE

An efficient multipath routing schema in multi-homing scenario based on protocol-oblivious forwarding

Pufang MA[1,2], Jiali YOU (✉)[1,2], Jinlin WANG[1,2]

1 National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China
2 University of Chinese Academy of Sciences, Beijing 100049, China

Abstract With the advent of 5G, multi-homing will be an increasingly common scenario, which is expected to increase transmission rates, improve transmission reliability, and reduce costs for users. However, the current routing methods are unable to fully utilize the resources of networks to achieve high-performance data transmission for multi-homed devices. In the current routing mechanism, there is only one destination address in the packet forwarded to the multi-homed host. Thus, the packet is difficult to adjust its path on the fly according to the status of the network to achieve better performance. In this paper, we present an efficient routing schema in multi-homing scenario based on protocol-oblivious forwarding (POF). In the proposed schema, the packet forwarded to the multi-homed host carries multiple destination addresses to obtain the ability of switching the transmission path; meanwhile, the router dynamically adjusts the path of the packet through the perception of the network status. Experimental results show that our schema could utilize the alternative paths properly and significantly improve the transmission efficiency.

Keywords multi-homing, routing, software-defined networking, protocol-oblivious forwarding

1 Introduction

With the explosive growth of network data, multi-homing is increasingly required to increase transmission efficiency. The statistics forecast that the global mobile data traffic will reach 49 exabytes per month in 2021 at a compound annual growth rate of 47 percent [1]. Mobile devices now usually connect the networks through various wireless access techniques, such as Wi-Fi, 3G and 4G. Through the multiple access technologies, users can enjoy better services, such as high transmission rate, high reliability and low cost. With the coming of 5G network, multi-homing will give users more choices to connect the network. Thus, multi-homing will be more necessary in future to help users achieve better quality of experience (QoE). Different access networks usually have different service capabilities. The authors in [2] point out that the concurrent use of multiple heterogeneous networks without proper control methods could even perform worse than one fast network. Thus, it is still important to design a method to provide desired services for users based on the dynamic status of the network.

The network is required to provide services with high-throughput and low-latency for some specified applications. Thus, the network should have the capability to dynamically adjust the flows of the applications when the current paths of flows cannot meet the demands. There are some techniques to implement dynamic selection of the paths, such as anycast and load balancer. Anycast [3] could enable the router to forward the packet to one of a group of hosts. This property of anycast enables it to discover the nearest resources. But it is difficult to deploy on a large scale and has poor scalability. Load balancer now is widely used, especially in cloud

Received November 21, 2018; accepted May 7, 2019

E-mail: youjl@dsp.ac.cn

computing [4]. Maglev [5] is a load balancer which has been used in Google's services to evenly distribute the packets to the service endpoints. To implement the function of load balancing, the network needs the extra deployment of the load balancers. When the size of the request is large, load balancer could become a bottleneck of the service, which will affect the quality of the service [4, 6]. Thus, elastic scalability is also problem for load balancers. Besides, as the authors in [4] say, the load balancers are usually dedicated for special server instances, and are not used for universal services.

It is difficult to manage today's network and hard to deploy new protocols without updating the network devices. To improve the agility of the network, Software-defined networking (SDN) [7] has been proposed. SDN enable the programmability of network through decoupling the control plane and forwarding plane of the network. OpenFlow [8] is mainstream south-bound protocol of SDN and could support various protocols. However, OpenFlow is still protocol-aware and has to be continuously updated to support new protocols. Programming protocol-independent packet processors (P4) [9] and protocol-oblivious forwarding (POF) [10] are the mainstream protocol-independent methods. P4 implements protocol-agnostic forwarding by designing a high-level language for programming protocol-independent packet processors. POF has been proposed to implement protocol oblivious forwarding and provide fully programmability of the network through defining a set of protocol-agnostic instructions. Because of the property of protocol oblivious, POF could support various new protocols, and enable the switch to flexibly modify and forward the packet according to the command of the controller. Therefore, POF makes it possible to design a new and efficient multi-homing routing schema to improve network flexibility.

In this paper, we design and implement an efficient routing schema in multi-homing scenario based on POF to enable agile routing and dynamic path selection. Our contributors are mainly twofold. First, we introduce the multiple destination addresses packet to allow the router to adjust the path of the packet in the process of forwarding, and the dynamic selection of path could improve the transmission efficiency and network resource utilization. Second, the maintaining of network dynamic status information supports the switch to dynamically select the forwarding port to bypass the links of poor ability. Experiments demonstrate that the proposed schema can improve the throughput by about 20% than baseline methods. Moreover, it can achieve stable performance even when the difference in path capabilities is very big. The completion time of the flow can also be reduced to improve the quality of user's experience.

The rest of the paper is arranged as below. In Section 2, we introduce the related work on the protocol-independent forwarding methods and multi-homing solutions. Section 3 will describe our multi-homing routing schema from different parts in detail. This is followed by experimental evaluations of the proposed routing schema in Section 4. Section 5 discusses the status computation method and status collection method in our future work. Finally, in Section 6, we summarize the work of this paper.

2 Related work

In this part, we will respectively introduce the protocol-independent forwarding methods and multi-homing solutions.

2.1 Protocol-independent forwarding

P4 and POF are the mainstream protocol-independent methods. P4 implements protocol-agnostic forwarding by designing a high-level language for programming protocol-independent packet processors. P4FPGA [11] extends the compiler of P4 and designs a P4-to-FPGA compiler to support different architecture configurations. The authors in [12] translate P4 to DPDK and utilize the DPDK to implement high speed packet forwarding. Network simulation is important to demonstrate the performance of the protocol designs of the researchers. NS4 [13] has been developed to support simulation of P4-enabled networks.

POF has been proposed to implement protocol oblivious forwarding and provide fully programmability of the network through defining a set of protocol-agnostic instructions. Because of the property of protocol oblivious, POF could support various new protocols, and enable the switch to flexibly modify and forward the packet according to the command of the controller. Therefore, POF makes it possible to design a new and efficient multi-homing routing schema to improve network flexibility. The authors in [14] have discussed the enhanced programmable property of POF and built a WAN-based POF network testbed to demonstrate the fully programmability of POF. Besides, some researches [15] use the source routing based on POF to improve the SDN scalability. The source routing based on POF could decrease the state explosion of flow tables in SDN switches. POFOX [16] is a SDN controller for POF that enables users to flexibly manage the forwarding of the flows in the POF network. The authors in [17] present PNPL, a control plane programming framework for POF that could automatically generate and maintain the forwarding pipelines for user-defined protocols. There are some popular applications that are designed and developed based on POF. FlowWatcher [18] utilizes the POF-based source routing to monitor the flow in the network. The authors in [19] has implemented a POF-based switch which could forward the packet at 10Gpbs and supports the vehicular networks. Because of the fully programmability, POF also is widely used in network slicing [20, 21]. The authors in [21] use the network virtualization based on POF to design a wireless network system. The system could orchestrate the network function virtualization to support mobile edge computing (MEC).

2.2 Multi-homing solutions

The network layer's solutions support multi-homing through changing the protocol stack or the network architecture. Locator/ID separation (LIS) and core/edge separation (CES) are the mainstream strategies to implement multi-homing [22]. IP address has been used as the routing identifier and end-point identifier. LIS suggests separating these functions into different parts. The two different parts are the node IDentity (NID) and location IDentity (LID). The NID, which is unique, is used to identify the host, while the LID is used for routing. The decoupling of IP addresses can provide mobility and multi-homing. The representative solutions of LIS are host identity protocol (HIP) [23] and Shim6 [24]. Different from LIS, the CES separates the network into edge network and core network. This architecture could solve the routing scalability problem. Locator/ID separation protocol (LISP) [25] is a presentative of CES. LISP enables separation of IP addresses into two new numbering spaces: end-point identifiers (EIDs) and routing locators (RLOCs). EIDs are used for identifying the devices and independent of the network topology. RLOCs are used for routing and topologically assigned to network points. The resolution system realizes the mapping between the EIDs and RLOCs. LISP offers traffic engineering, multi-homing, and mobility benefits.

The transport layer's proposals implement multi-homing through establishing multiple paths to improve the reliability and efficiency of the transmission. Streaming control transmission protocol (SCTP) [26] is a transport layer protocol, serving in a similar role as the popular protocols TCP and UDP. It establishes multiple paths: one primary path and other backup paths. SCTP uses the primary path for data transmission, and the flow will be switched to the available backup path when detecting the failure of the primary path. The authors of [27] argue that the primary path of SCTP will be switched only when the failure of primary path occurs. This will decrease the efficiency of the transmission. To efficiently use the paths, the authors suggest the SCTP chooses the primary path according to the detected round trip time. Nishida et al. [28] propose a quick failover algorithm for SCTP to avoid the performance degradation in the path switching process. Except for using the single primary path, there are also some researches to use the multiple paths of SCTP to improve the transmission rate. In [29, 30], the authors propose use concurrent multipath transfer (CMT) to reduce the delay and increase the throughput.

Multipath TCP (MPTCP) [31] is another transport layer multi-homing protocol. It aims at allowing to simultaneously use multiple paths to maximize resource usage and increase redundancy. Though the MPTCP could use the multiple paths to transmit the packets, it still has some problems. As [2, 32] point out, the performance of MPTCP will degrade dramatically when the paths are heterogeneous. This is because the default MPTCP scheduler prefers paths with lower RTT and ignores other properties of the paths, such as bandwidth. To efficiently utilize the paths, the authors propose to use all relevant information of the path to schedule the flow to the paths. With the development of SDN, the controller usually could obtain a lot of network information. Nam et al. [33] have proposed to use the network feedback information in real time to adjust the flow of different paths. Wang et al. [34] also propose to distribute the traffic among the multiple paths through monitoring the status of the paths.

The schemas mentioned above have some limitations. First, the adjustment of the path is inflexbile and the packet cannot dynamically change its path during the forwarding process. This is because the packet generally carries one destination address of the multi-homed host. When forwarding the packet, the router is not aware that the destination host of the packet is multi-homed. So the packet can only be forwarded following the fixed path, even when the path is unhealthy. Second, the network cannot timely adjust the packet of the packet according to its status. It at least takes one RTT time to obtain the status of the path and after that the end-point could change the path of the packet through changing the destination addresses. The ideal multi-homing routing schema should dynamically adjust the packet in the process of forwarding to improve the stability and efficiency of transmission. However, the current multi-homing methods cannot meet these needs. Our schema leverages the dynamic information of the network and the full programmability of the POF to facilitate flexible path selection during the forwarding process.

3 Efficient routing schema in multi-homing scenario based on POF

In this section, we will introduce the design principle of our proposed routing schema, including the network architecture, the packet format, and the structure and maintaining of network status information.

3.1 Architecture of network in multi-homing scenario based on POF

Our routing schema needs the programmability technology to implement the dynamic path selection and flexible modification of the packet. POF has been proposed to implement protocol oblivious forwarding and enable the network fully programmable through defining a set of protocol-agnostic instructions.

Figure 1 shows the architecture of our proposed routing schema in multi-homing scenario based on POF. The network architecture contains the control plane and forwarding plane. The POF controller communicates with POF switch through POF channel. The controller maintains a mapping system to implement the mapping from endpoint identifiers (EIDs) to IPs. EID is the unique identifier of the device. The server requests the IPs of the multi-homed host with EID and the mapping system replies with the corresponding IPs. The packet forwarded to the multi-homed endpoint carries the multiple destination addresses. After matching each destination address of the packet, the router obtains multiple output ports to output the packet. Thus, the router also maintains a network status information base (NSIB) to select the output port at each hop. Different from FIB that is computed with static weights, the NSIB is calculated according to the dynamic information of the network. After the selection of the output port, the router deletes the addresses which are not corresponding to the output port to decrease the unnecessary hop. Therefore, the packet with multi-address could dynamically adjust its path during the forwarding process to pursue high-quality performance.

We use the example in Fig. 1 to illustrate the forwarding process of the multiple destination addresses' packet. The server receives 3 IPs through the mapping request of EID1 and inserts them in the packet. After matching FIB for the multi-address packet at R6, the router could achieve *port*1 and *port*2 to output the packet. Then the router compares the status value of *port*1 and *port*2 through NSIB and finds that the *port*2 has better status. So the router outputs the packet through *port*2. At the same time, the router deletes the destination *IP*1, which is corresponding to *port*1, and reserves *IP*2 and *IP*3. The Router R5 selects *port*1 to forward the packet and reserves *IP*2. The next router performs the same dynamic output port selection and address trimming until the packet reaches the host.

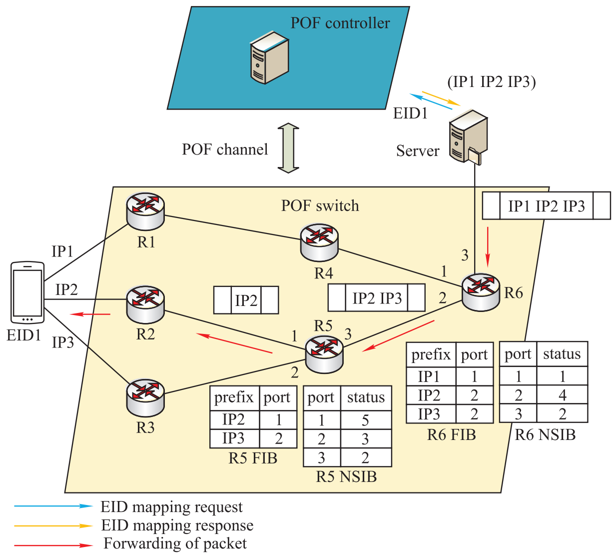

Fig. 1 Network routing architecture in multi-homing scenario

3.2 Packet format with multiple destination addresses

We insert the additional addresses in "option" fields of the IPv4 header as shown in Fig. 2. The "header length" field in the header indicates the length of the IPv4 header. Each destination address will match the forwarding table of the router. For a packet with n destination addresses, the set of destination IPs is described as:

$$IPs = \{IP_1, IP_2, \ldots, IP_n\}. \tag{1}$$

After matching the FIB, each destination address will obtain an output port. We assume that we obtain m forwarding ports and define the output ports and the mapping relationship as:

$$ports = \{port_1, port_2, \ldots, port_m\}, \tag{2}$$

$$mapping\,(IP_i) = port_j, 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m. \tag{3}$$

We classify the addresses according to the output port and the relationship between them is:

$$addresses\,(port_i) = \{IP_{i1}, IP_{i2}, \ldots, IP_{ik}\}, \tag{4}$$

where $mapping(IP_{ij}) = port_i$ and $IP_{ij} \in IPs$. From the above analysis, we can see that the packet with multiple destination addresses could obtain multiple output ports. Thus, the router could select the output port to meet the QoS requirement of the packet. The normal IP packet has only one destination IP address for the multi-homing network. When the packet matches the FIB, it will only have one forwarding port according to the longest prefix matching. The packet cannot adjust its path when confronting link congestion or failure.

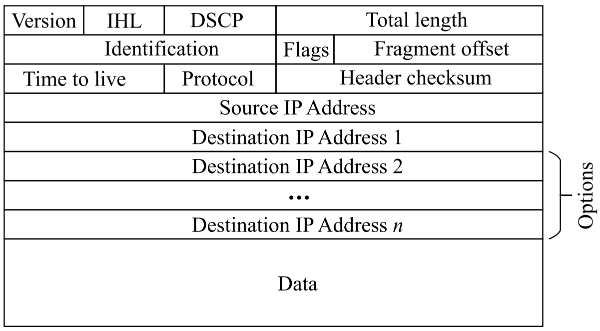

Fig. 2 Multiple destination addresses' packet format

3.3 Network status information maintenance

NSIB is used for directing the forwarding of the multiple destination addresses' packet with multiple output ports. It maintains the status of the surrounding links connected with the port. The structure of NSIB is shown in Fig. 3. The match field is the port identifier of the router. The calculation of status value of the port is defined as:

$$status\_value = S(X), X = (x_1, x_2, \ldots, x_k), \tag{5}$$

where X is the network information collected from that port, including bandwidth, load, and delay. $S(X)$ can be flexibly defined based on the optimization goal. After matching the FIB, there may be multiple output ports for the multiple destination addresses' packet. So the router could select among these ports according to the values of the ports. The packet usually chooses the output port with the best status value.

port	f(X)
port 1	value 1
port 2	value 2
...	...
port k	value k

Fig. 3 NSIB structure

The status table directs the forwarding of the flows in the network. The objective of it is to properly distribute the traffic in the network to improve the efficiency of the transmission. Thus, the computation of status value could be described as an optimization problem with constraints. The dynamic network can be described as $G(V, E, S)$, where V and E are the nodes and edges of the network, S are the dynamic status values of the ports. We define the problem under constraints as follows:

$$\max \Phi(X), X = (x_1, x_2, \ldots, x_k), \tag{6}$$

where Φ is an objective function defined according to the need of service, and X is the dynamic information of the link, such as load, capacity and delay. For example, to balance the traffic in the network, Φ can be the link utilization $f_{u,v}/c_{u,v}$, where $f_{u,v}$ and $c_{u,v}$ are respectively the load and capacity of the link. The objective is to minimize $\max_{(u,v)\in V} \Phi(f_{u,v}, c_{u,v})$ and the constraint is $\sum_{d\in V} f^d_{u,v} \leqslant c_{u,v}$. We assume the multi-address flow has m available output ports $\{p_1, p_2, \ldots, p_m\}$ and the corresponding status values are $\{s_1, s_2, \ldots, s_m\}$. The router selects the port p_k $(1 \leqslant k \leqslant m)$ with the best status value, which satisfies $s_k \geqslant s_i$ $(i \neq k)$. Thus, the status values S of the router needs to be carefully calculated to meet the requirements of the objective function. The status values S could be defined according to the demand, such as load.

In this paper, to maximize the throughput of the transmission, we hope the router selects the output port with the largest available bandwidth for the multi-address packet. Thus, the computation of status value is defined as

$$S(X) = bandwidth - load. \tag{7}$$

That is to say, the router makes decisions with the greedy algorithm and selects the output port with the best available bandwidth.

We need a strategy to construct and maintain the status values of the different ports. The status value represents the status of the paths to the destination through the port. At each hop, the router may have multiple output ports to forward the multiple destination addresses' packet, but it is unaware of which port is better. Some researchers have used periodic probes in datacenter network to achieve in-network load balancing, such as CONGA [35] and HULA [36]. The periodic probing of the whole network will result in high overhead.

To reduce the overhead of the process, we use the passive network measurement method. The switch maintains a task to monitor the loads of the links of the switch. We divide the load of the link to N different levels. If the load of the link changes from one level to another, the switch will send a status change message to the controller. Then the controller will modify the status value of the port in the status table. The passive measurement method decreases the unnecessary overhead compared to active measurement method when the network is stable. Besides, we could also improve the accuracy of the measurement through increasing the number N, which could capture the small change of the network.

3.4 Output port selection

The port selection process selects the output port after sorting the status values of the matched ports. The mapping relationship between the matched ports and the corresponding status is defined as:

$$mapping\,(port_i) = value_i, 1 \leqslant i \leqslant m. \tag{8}$$

We define the key-value pairs of port and status value as:

$$\begin{aligned} map\,(port, value) = \{&port_1 : value_1, port_2 : value_2, \\ &\ldots, port_m : value_m\}. \end{aligned} \tag{9}$$

After obtaining these key-value pairs, we will sort the ports by the value and select the port with best value. The port selection function is defined as:

$$\begin{aligned} port_x &= select\_port\,(f(X), pack) \\ &= select\_port\,(map\,(port, value)), \end{aligned} \tag{10}$$

where *pack* is the packet carrying multiple destination addresses. The *select_port* function sorts the status values of the ports and selects the output port with best status.

3.5 Addresses trimming

At each hop, after the matching of FIB, the router may obtain multiple output ports to forward the multiple destination addresses' packet. Then, the router compares the status values of different output ports to determine which port to forward the packet. The router chooses the output port that has the best status value after matching the NSIB. It classifies the addresses based on the matched output port. The addresses that correspond the best output port are called "selected addresses", and other addresses are called "unselected addresses". The selected addresses are remained for routing and forwarding in the rest of the path. The unselected addresses should be deleted according to the need. The address trimming method is shown in Algorithm 1. The destination addresses of the packet will be dynamically trimmed during the forwarding process at each hop.

Algorithm 1 Delete the address according to the selected output port

function ADDRESS_DELETION_METHOD(*address*, *selected_port*)
  if *mapping*(*address*)! = *selected_port* **then**
    delete the address
  else
    reserve the address
  end if
end function

4 Performance evaluation

In this section, the performance of our routing schema in multi-homing scenario based on POF will be demonstrated.

4.1 Experiment setup

Mininet is chosen as the simulation platform, which has been widely used in SDN simulation. The switch we use is software-based POFSwitch [10], and the controller is the extended POX controller. We make use of the raw socket to enable the multi-homed host to receive packets at multiple interfaces. We choose three ISP topologies from Rocketfuel inter-domain topology data [37] to construct our topology. The topology information is descried in Table 1. Figure 4 shows the connection of the simulation topology. The reason why we use multiple ASes is to simulate the host connects two different ISPs. The whole topology could be seen as a router-level topology.

Table 1 Summary of ISP topologies

AS number	Nodes	Links
4006	7	18
4725	11	26
6461	19	68

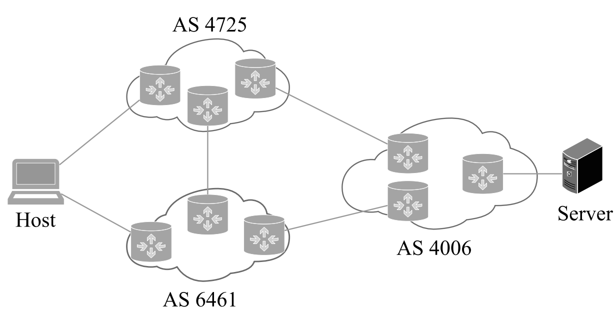

Fig. 4 Simulation topology

We will evaluate our routing schema from different parts. First, we demonstrate the performance of our routing schema by comparing it with HIP [23], TCP, ECMP [38], SCTP [26] and Multipath TCP [31] in terms of the transmission stability and flow completion time. Besides, the performances of different port selection methods are compared in Section 4.4 to prove the superiority of selecting the output port according to the status of the network. Our routing schema focuses on implementing and optimizing multi-homing at network layer. Although SCTP and MPTCP work at different layer with our schema, the goal is also to improve the performance of transmission. Thus, the results still can illustrate the advantages of our routing schema.

Our multi-homing routing method can be used for both TCP and UDP. The version of MPTCP we use is 0.90 implemented in the Linux Kernel by the IP Networking Lab [39]. The abbreviation of our multi-homing network routing schema is MHNR. We will evaluate the transmission efficiency of our method in heterogeneous paths and different network loads. We use Eq. (7) to calculate the status value in the simulation where the router selects the output port with the largest available bandwidth for the multi-address packet.

4.2 Transmission stability in heterogeneous paths

In the part, we will test our multi-homing routing schema in the heterogeneous conditions. The authors in [2] point out that the concurrent use of multiple heterogeneous networks without proper control algorithms could perform even worse than one fast network. In our simulation, the host, which is multi-homed, randomly connects one switch from AS 4006 and 4725, and the server randomly connects one switch from AS 6461. The bandwidth values of the links in AS 6461 and AS 4006 are respectively 30Mbps and 10Mbps. We set the bandwidth values of the links in AS 4725 to be 0.5Mbps, 1Mbps, 2Mbps, 4Mbps and 8Mbps to compare transmission stability in heterogeneous paths, which respectively correspond to scenarios 1, 2, 3, 4 and 5 in Fig. 5. IPerf3 is used to test the throughput from the server to the host. The results are averaged over 10 runs in each scenario.

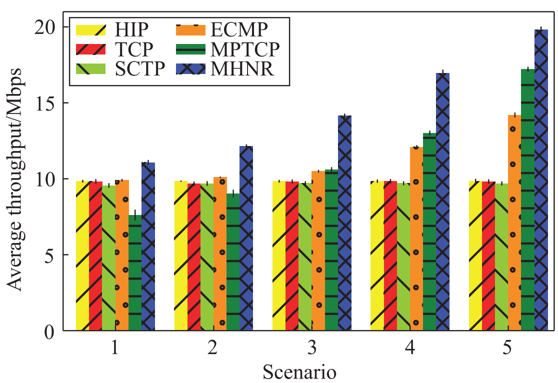

Fig. 5 Transmission performance in heterogeneous paths

Figure 5 describes the average throughput of different methods under the heterogeneous scenarios. The scenarios 1 and 2 in Fig. 5 show that when the paths are heterogeneous, the performance of MPTCP degrades severely and is even worse than TCP. At this case, it is better to abandon the path with poor transmission capacity. In scenarios 3, 4 and 5, with the increase of the link's bandwidth, the performance of MPTCP gradually improves. Since HIP, TCP and SCTP only use one path to transmit data, the performances of them are stable. ECMP performs a little better than TCP and SCTP. The throughput could improve about 20% compared with HIP and SCTP in scenarios 4 and 5. It also outperforms MPTCP when the transmission differences between the paths are large. But it does not perform better than MPTCP and our routing schema when the paths have small difference in transmission in scenarios 4 and 5. This is because ECMP could use multiple paths only when the costs of the multiple paths are equal, which is not usual for the flows. Our multi-homing routing schema behaves well when the bandwidth of the paths greatly differs. Because our routing schema could identify the heterogeneity of the paths and discard the path of poor performance. Besides, the average throughput of our method improves 20% compared with MPTCP. Due to carrying multiple destination addresses, our routing method could discover available links to improve the throughput.

4.3 Flow completion time in different network loads

In this part, we simulate the client requests contents from the server where the host randomly connects two switches from two different ISPs and the server randomly connects one switch in the third ISP. The bandwidth values of all links are 50Mbps. Ten groups of clients and servers are used. The client requests contents from server, and the requests follow a Poisson process. The flow sizes of requested contents have heavy-tailed characteristics. We use the web search workload in [40] to simulate the traffic patterns. Flow completion time (FCT) is used as the performance metric. We change the network load through changing the request rate of contents to test the performance of different methods under different network load.

Figure 6 shows the FCT performance of different methods under the loads from 0.1 to 0.9. Figs. 6(a), 6(b) and 6(c) respectively indicate the average flow completion times of all flows, small flows and large flows under different network load. The researchers in [36] point out that the sizes of the flows in the network are heavy tailed. That is to say, most of the flows in the network are small, but the small number of large flows contribute to about 80% of the traffic in the network. The average performance of FCT may mainly reflect the performance of large flows. It is possible that the performance of small flows is bad while the average performance is pretty good. Thus, we particularly show the performance of small flows with the method in Fig. 6(b). We can see that MPTCP and MHNR performance better than HIP and SCTP. Because they can use the idle resources of the network to improve transmission efficiency. ECMP performs better than

TCP and SCTP and decreases the FCT by about 5%. But its performance is worse than MPTCP with an increase of 10.1% in FCT when the load of the network is high. This is because ECMP could use multiple paths only when the costs of the multiple paths are equal. MHNR and MPTCP have almost the same performance when the network loads are low. However, when the loads of the network increase, MHNR has better performance than MPTCP. When the load of network is up to 70%, the FCT of MHNR decreases about 18% than MPTCP. This is because MHNR has dynamic state information of the network and can adjust the paths of flows when congestion occurs.

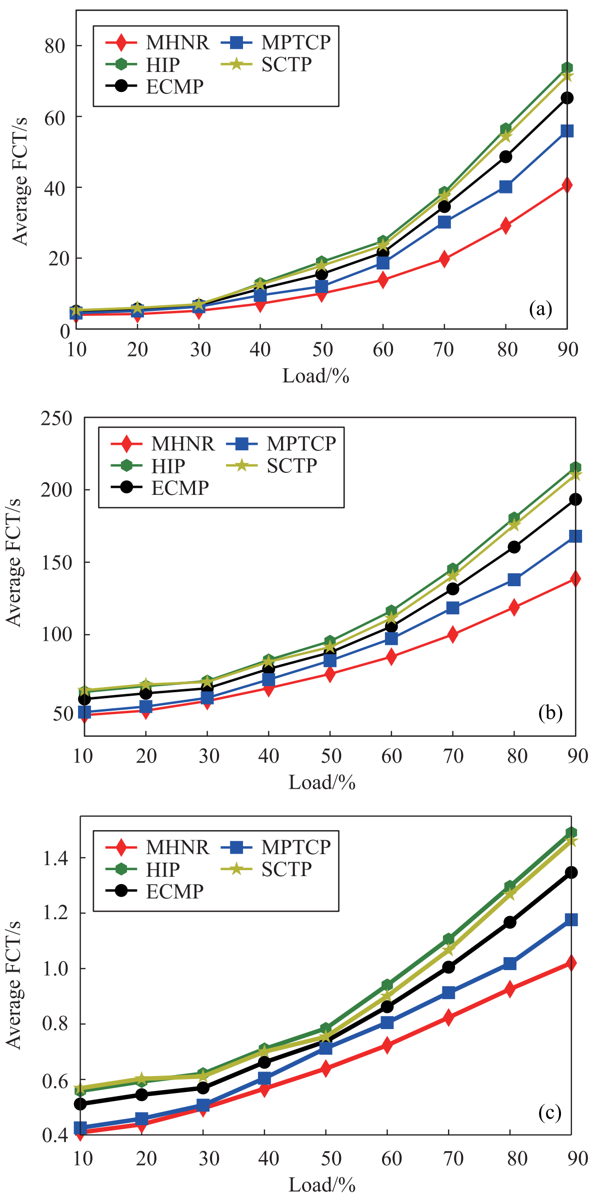

Fig. 6 FCT under different network loads. (a) Overall average FCT; (b) small flows (<100KB); (c) large flows (>10MB)

4.4 Transmission performances of different port selection methods

In the above experiment, we select the output port based on the status value of the port which is calculated according to Eq. (7). The router could also select the available output port using other methods, such as random selection and polling selection, which may be simpler than status selection. Random selection randomly selects the output port from the available ports, while polling selection in turn uses the ports. They do not need the dynamic information of the network. In the following, we compare the performances of different output port selection methods to discuss whether the status selection can obtain the satisfactory performance.

We compare the transmission stability of three different port selection methods in Scenarios 1 and 5 which are described in Section 4.2. Figure 7 shows that the status selection could achieve higher throughput and more stable than random selection and polling selection. The transmission jitter of random selection is very severe. This is because random selection is not aware of the status of the network, which will cause the out-of-order of the packets. The performance of polling selection is better than random selection, but its

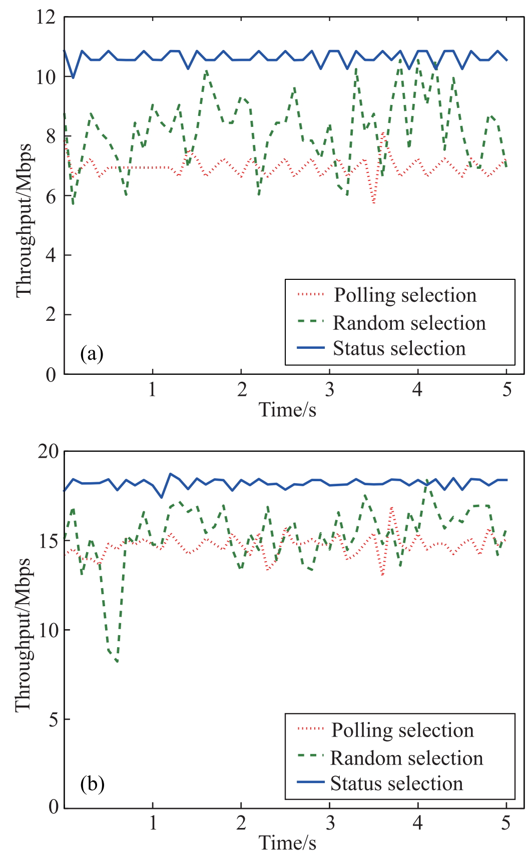

Fig. 7 Transmission stability of different port selections. (a) Scenario 1; (b) Scenario 5

throughput is still lower than status selection. Thus, we conclude that it is necessary to select the output port according to the status of the network.

5 Discussion

In our current work, there are still some points to be further improved, including the status computing function. The focus of this paper is to introduce the principle of our proposed efficient multipath routing schema in multi-homing scenario. We want to highlight the advantages of the routing schema. Thus, the computation method of status value is not complex. We are now trying to optimize the computation of status value with improved maximum flow method and convex optimization method. Heuristic methods are also a good way to optimize the computation. The classical ant colony optimization is used in AntNet [41] to implement adaptive routing, which optimizes the selection of the path of the packet through exchanging collected information. Therefore, in our future work, we will try to use the appropriate heuristic method to improve the accuracy of the status value.

Additional delay is introduced in the process of status information collection, which will affect the performance of the transmission. When the topology of the network is large, the delay will increase. Thus, it is also difficult to maintain the global status of the network for a single controller in real-time. To solve these problems, some researchers [42] has proposed the distributed architecture of SDN control plane. Each controller maintains an area of switches and can quickly respond to the local events. The distributed controllers exchange the network status information through the east-west interface. Thus, the control plane can obtain the global status of the network in relatively real-time. The communication and synchronization between the multiple controllers could also be a problem. Therefore, we will try to use the distributed multiple controllers to solve these problems in our future work.

6 Conclusion

In this letter, we propose and implement an efficient routing schema in multi-homing scenario based on POF to enable the dynamic path adjustment of the packet with multiple destination addresses according to the status of the network. Experimental results show that our routing schema could effectively use the idle resources of the network to increase the throughput by more than 20% and achieve more stable transmission performance in heterogeneous paths and heavy network loads compared with HIP, SCTP and MPTCP. Besides, when the load of network is up to 70%, the schema can reduce the completion time of the flow by about 18% compared with MPTCP.

Acknowledgements This work was supported by the SEANET Technology Standardization Research and System Development (XDC02010701) and Youth Innovation Promotion Association CAS (Y529111601).

References

1. Cisco visual networking index. Global mobile data traffic forecast update, 2016-2021. Cisco: San Jose, 2017
2. Lim Y, Nahum E M, Towsley D, Towsley D F, Gibbens R J. ECF: an MPTCP path scheduler to manage heterogeneous paths. Measurement and Modeling of Computer Systems, 2017, 44(1): 33–34
3. Ballani H, Francis P. Towards a global IP anycast service. ACM SIGCOMM Computer Communication Review, 2005, 35(4): 301–312
4. Rahman M, Iqbal S, Gao J. Load balancer as a service in cloud computing. In: Proceedings of the 8th International Symposium on Service Oriented Software Engineering. 2014, 204–211
5. Eisenbud D E, Yi C, Contavalli C, Contavalli C, Smith C, Kononov R, Mannhielscher E, Cilingiroglu A, Cheyney B, Shang W, Hosein J D. Maglev: a fast and reliable software network load balancer. In: Proceedings of the 13th USEHIX Symposium on Networked Systems Design and Implementation. 2016, 523–535
6. Xu M, Tian W, Buyya R. A survey on load balancing algorithms for virtual machines placement in cloud computing. Concurrency and Computation: Practice and Experience, 2017, 29(12): e4123
7. Farhady H, Lee H, Nakao A. Software-defined networking. Computer Networks, 2015, 81: 79–95
8. McKeown N, Anderson T, Balakrishnan H, Parulkar G M, Peterson L L, Rexford J, Shenker S, Turner J S. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69–74
9. Bosshart P, Daly D P, Gibb G, Izzard M J, Mckeown N, Rexford J, Schlesinger C, Talayco D, Vahdat A, Varghese G. P4: programming protocol-independent packet processors. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 87–95
10. Song H. Protocol-oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. 2013, 127–132
11. Wang H, Soule R, Dang H T, Lee K S, Shrivastav V, Foster N, Weatherspoon H. P4FPGA: a rapid prototyping framework for P4. In: Proceedings of Symposium on SDN Research. 2017, 122–135
12. Laki S, Horpácsi D, Vörös P, Kitlei R, Leskó D, Tejfel M. High-speed forwarding: a P4 compiler with a hardware abstraction library for Intel DPDK. In: Proceedings of P4 Workshop. 2016
13. Bai J, Bi J, Kuang P, Fan C, Zhou Y, Zhang C. NS4: enabling programmable data plane simulation. In: Proceedings of the Symposium on SDN Research. 2018
14. Li S, Hu D, Fang W, Ma S, Chen C, Huang H, Zhu Z. Protocol obliv-

ious forwarding (POF): software-defined networking with enhanced programmability. IEEE Network, 2017, 31(2): 58–66

15. Li S, Han K, Ansari N, Bao Q, Hu D, Liu J, Yu S, Zhu Z. Improving SDN scalability with protocol-oblivious source routing: a system-level study. IEEE Transactions on Network and Service Management, 2018, 15(1): 275–288
16. Tan X, Zou S, Guo H, Tian Y. POFOX: towards controlling the protocol oblivious forwarding network. In: Park J, Yi G, Jeong Y S, Shen H, eds. Advances in Parallel and Distributed Computing and Ubiquitous Services. Singapore: Springer, 2016
17. Wang X, Tian Y, Zhao M, Li M, Mei L, Zhang X. PNPL: simplifying programming for protocol-oblivious SDN networks. Computer Networks, 2018, 147: 64–80
18. Zhao M, Li M, Mei L, Tian Y. FlowWatcher: adaptive flow counting for source routing over protocol independent SDN networks. In: Proceedings of the 8th International Conference on Electronics Information and Emergency Communication. 2018, 237–242
19. Sun Q, Xue Y, Li S, Zhu Z. Design and demonstration of high-throughput protocol oblivious packet forwarding to support software-defined vehicular networks. IEEE Access, 2017, 5: 24004–24011
20. Huang H, Niu B, Tang S, Li S, Zhao S, Han K, Zhu Z. Realizing highly-available, scalable, and protocol-independent vSDN slicing with a distributed network hypervisor system. IEEE Access, 2018, 6: 13513–13522
21. Han K, Li S, Tang S, Huang H, Zhao S, Fu G, Zhu Z. Application-driven end-to-end slicing: when wireless network virtualization orchestrates with NFV-based mobile edge computing. IEEE Access, 2018, 6: 26567–26577
22. Gladisch A, Daher R, Tavangarian D. Survey on mobility and multihoming in future internet. Wireless Personal Communications, 2014, 74(1): 45–81
23. Moskowitz R R, Nikander P, Jokela P. Host identity protocol. RFC 5201, 2008
24. Nordmark E, Bagnulo M. Shim6: level 3 multihoming shim protocol for IPv6. RFC 5533, 2009
25. Farinacci D, Lewis D, Meyer D, Fuller V. The locator/ID separation protocol (LISP). RFC 6830, 2013
26. Stewart R, Metz C. SCTP: new transport protocol for TCP/IP. IEEE Internet Computing, 2001, 5(6): 64–69
27. Katsaros K, Dianati M, Tafazolli R. Analytical model of RTT-aware SCTP. In: Proceedings of International Conference on Connected Vehicles and Expo. 2014, 439–443
28. Nishida Y, Natarajan P, Caro A. SCTP-PF: a quick failover algorithm for the stream control transmission protocol. IETF, 2016
29. Iyengar J R, Amer P D, Stewart R R. Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths. IEEE/ACM Transactions on Networking, 2006, 14(5): 951–964
30. Shailendra S, Bhattacharjee R, Bose S K. MPSCTP: a simple and efficient multipath algorithm for SCTP. IEEE Communications Letters, 2011, 15(10): 1139–1141
31. Ford A, Raiciu C, Handley M, Bonaventure O. TCP extensions for multipath operation with multiple addresses. RFC 6824, 2013
32. Deng S, Netravali R, Sivaraman A, Balakrishnan H. WiFi, LTE, or both?: measuring multi-homed wireless internet performance. In: Proceedings of Internet Measurement Conference. 2014, 181–194
33. Nam H, Calin D, Schulzrinne H. Towards dynamic MPTCP path control using SDN. In: Proceedings of NetSoft Conference and Workshops. 2016, 286–294
34. Wang W, He W, Su J. M2SDN: achieving multipath and multihoming in data centers with software defined networking. In: Proceedings of International Workshop on Quality of Service. 2015, 11–20
35. Alizadeh M, Edsall T, Dharmapurikar S, Vaidyanathan R, Chu K, Fingerhut A, Matus F, Pan R, Yadav N, Varghese G. CONGA: distributed congestion-aware load balancing for datacenters. ACM SIGCOMM Computer Communication Review, 2014, 44(4): 503–514
36. Katta N, Hira M, Kim C, Sivaraman A, Rexford J. Hula: scalable load balancing using programmable data planes. In: Proceedings of the Symposium on SDN Research. 2016
37. Spring N, Mahajan R, Wetherall D, Anderson T E. Measuring ISP topologies with Rocketfuel. IEEE ACM Transactions on Networking, 2004, 12(1): 2–16
38. HOPPS C. Analysis of an equal-cost multipath algorithm. RFC 2992, 2000
39. Paasch C, Barré S. Multipath TCP in the Linux kernel. See Multipath-tcp.org Website. 2013
40. Alizadeh M, Greenberg A G, Maltz D A, Padhye J, Patel P, Prabhakar B, Sengupta S, Sridharan M. Data center TCP (DCTCP). ACM SIGCOMM Computer Communication Review, 2010, 40(4): 63–74
41. Dorigo M, Stützle T. Ant Colony Optimization: Overview and Recent Advances. Handbook of Metaheuristics, Springer, Cham, 2019, 311–351
42. Asten B J, van Adrichem N L M, Kuipers F A. Scalability and resilience of software-defined networking: an overview. 2014, arXiv preprint arXiv: 1408.6760

Pufang Ma received the BS degree from the School of Information Science and Technology, University of Science and Technology of China, China in 2014. He is currently pursuing the PhD degree with National Network New Media Engineering Research Center, Institute of Acoustic, Chinese Academy of Sciences, China. His research interests include software-defined networking, information-centric networking, multi-homing network, routing protocols, and routing algorithms.

Jiali You is an associate professor of the National Network New Media Engineering Research Center, Institute of Acoustics (IOA), Chinese Academy of Sciences (CAS), China. She received her PhD degree in Signal and Information Processing from the Institute of Acoustics, Chinese

Academy of Sciences, China in 2008. Between January 2015 and January 2016, she was a visiting scholar at University of Massachusetts Amherst, USA. Her research interests include distributed network and cloud computing.

Jinlin Wang is a professor of the National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, China. He received the BS degree in 1986 from the Department of Mathematics, University of Science and Technology of China, and the MS degree in 1989 from Graduate University of Chinese Academy of Sciences, China. His research interests include digital signal processing, new media technology, and future networks.