A Collaborated IPv6-Packets Matching Mechanism Base on Flow Label in OpenFlow

Weifeng Sun, Huangping Wei, Zhenxing Ji,
Qingqing Zhang, and Chi Lin(✉)

School of Software, Dalian University of Technology, Dalian, Liaoning, China
{wfsun, c.lin}@dlut.edu.cn,
{1299856803, jasonjee1990, 1165331393}@qq.com

Abstract. Software Defined Networks (SDN), the separation of a network device's control and data planes, do not need to rely on the underlying network equipment (routers, switches, firewall).It is a new network which collaborated IP and a lot of relevant technical content. The control of SDN is completely open, the user can customize any rule strategy to achieve network routing and transmission, which is more flexible and intelligent. Internet Protocol version 6 (IPv6), with the 128-bit address, is the next generation Internet. In this paper, we present a Flow Table structure by using Flow Label and a matching approach which use Flow Label within IPv6 protocol to decrease the size of Flow Table with OpenFlow and the time of forwarding IPv6 packets in SDN based on OpenFlow. The simulations and analyses show that this flow table mechanism performs better.

Keywords: IPv6 · OpenFlow · Flow label · Compression of flow table

1 Introduction

The Software Define Networks (SDN) [1] is an approach or architecture to not only simplify computer networks but also to make it more reactive to the requirements of workloads and services placed in the network. Originally, the control planes and data planes are consolidated into network equipments such as the IP routers and the Ethernet switches. SDN collaborates a lot of network transmission protocols and functions. It allows for a centrally managed and distributed control, management, and data plane, which policy that dictates the forwarding rules is centralized, but the actual forwarding rule processing are distributed among multiple devices.

Internet Protocol version 6 (IPv6) [2] is the latest version of the Internet Protocol. It uses a 128 bit address, allowing 2128 times as many as IPv4, which uses 32 bit addresses. IPv6 is bringing many unique benefits when it is combined with emerging technologies such as Clouds Virtualization, Internet of Things, etc. These benefits include host automation, scalability in addressing, route aggregation, forwarding and traffic steering functions [3]. IPv6 enables the transformation that occurs at the networking infrastructure level which can make SDN and network easily easy to scale.

There is no doubt that most of the SDN solutions are based on the OpenFlow protocol [4], which was defined by Open Networking Foundation (ONF) [5]. OpenFlow was

S. Guo et al. (Eds.): CollaborateCom 2015, LNICST 163, pp. 14–25, 2016.
DOI: 10.1007/978-3-319-28910-6_2

originally imagined and implemented as part of network research at Stanford University. And Nick McKeown firstly descripts detail the concept OpenFlow [6]. However, as the IPv6 support in OpenFlow, It is a problem that how to reduce the size of the flow table faced with OpenFlow. According to the OpenFlow standard, the flow table can be achieved with TCAM, Ternary Content Addressable Memory that is based on the contents of the address to find. In the traditional network equipment, TCAM mainly for FIB, MAC, MPLS Label and ACL table because of the length of the match fields for each table vary, so it can be separately designed, and has the maximum capacity limits in order to achieve a minimum overhead. Although OpenFlow designed multi-flow table, in order to reduce the overhead stream table, forming a handle in the form of pipeline, reducing the total number of records in the flow table. With the increasing of the matching fields, the space of flow table will be limited. This problem will become more and more prominent with the increasing of hosts. The size of flow table and the matching processing of OpenFlow that be more complex restrict the IPv6 development in SDN.

In this paper, we propose the flow table structure by using Flow Label within IPv6 protocol, in order to decrease the bite of flow table and the time of forwarding IPv6 packets in SDN based on OpenFlow. OpenFlow is designed forwarding of L1-L4 layer forwarding entries. We use Flow Label of L3 layer to match, instead of the address of IPv6 and the port of TCP/UDP, the OpenFlow match fields of L3-L4 Layer.

The main contributions of this paper are:

(1) We propose the flow table structure by using Flow Label in IPv6 feature, instead of L3-L4 Layers elements of matching, in order to decrease the time of matching and the IPv6 flow table structure of OpenFlow.
(2) We analysis and compare the latency, the jitter and the size of flow table between flow table with flow label and flow table without flow label, and proved that the former can perform OpenFlow better in SDN.

2 Related Work

The first edition of OpenFlow focused on IPv4 and did not support IPv6 flow [7]. ONF started to consider how IPv6 flows could be accommodated. Then the ONF published their OpenFlow1.2 which is the first version that supports IPv6 packet matching [8]. The OpenFlow1.2 provides basic support for IPv6, OpenFlow1.2 compliant switch can match on IP protocol number (Ethernet type 0 × 86dd = IPv6), IPv6 source/destination addresses, traffic class, flow label, and ICMPv6 types. This is a start at allowing IPv6 unicast and multicast traffics to match and OpenFlow flow table in a switch. When the ONF published the OpenFlow1.3 [9]. There is a few more IPv6 functions added in OpenFlow1.3 which expands the IPv6 Extension Header handling support and essential features including Hop-by-hop IPv6 extension header, Router IPv6 extension header, Router IPv6 extension header, access control, quality of service and tunneling support and had the same ability to match on IPv6 header fields such as source/destination addresses, protocol number (next header, extension header), hop-limit, traffic class, flow label, and ICMPv6 type as in Fig. 1. When the ONF release OpenFlow1.4 nothing changed regarding IPv6 support [10].

Ingress Port	Ethernet (L2)				IP (L3)			Tranport (L4)	
	Src MAC	Dst MAC	Type	VLAN ID	Src IP	Dst IP	Protocol	Src Port	Dst Port

Fig. 1. Ten-tuple for matching in OpenFlow

With the development of OpenFlow and IPv6, Chia-Wei Tseng combines the existing IPv6 protocol and the SDN (SDNv6) for future network, to provide the smarter and reliable network communication architecture. The SDNv6 [11] motivates a network architecture composed of reliable virtual entities, plug-and-play access with auto-configures and flexible service clouds over a physical network. Wenfeng Xia proposed a Software Defined Approach to Unified IPv6 Transition which unifies the variety of IPv6 transition mechanisms [12]. Xiaohan Liu proposes an IPv6 Virtual Network Architecture (VNET6) to support flexible services in IPv6 network. IPv6 is a critical protocol in VNET6 [13]. The VNET6 is adaptive to video service with high bandwidth and low tendency and improves quality of experiences to users. Batalle [14] proposes the IPv4 and IPv6 routing separation, and provides a different OpenFlow controllers conducted by the inter-domain routing OpenFlow network management method. Rodrigo, Fernandes and Rothenberg in Brazil and Hungary started to leverage these new features in OpenFlow1.3 [15]. Ivan Pepelnjak also uses the new features in IPv6 [16] to describe how OpenFlow can be used to help secure the IPv6 Neighbor Discovery Protocol (NDP) because it suffers from many of the same vulnerabilities as IPv4 ARP. William Stallings present about the various elements of an OpenFlow table [17] that includes the IPv6 header fields that can be matched. Araji and Gurkan present ESPM, Embedding Switch ID, Port number and MAC Address within IPv6 protocol and SDN technology, to decrease CAM table entries on the switch by forwarding the packets [18]. David R. Newman goes through the installation steps required to set up an OpenFlow protocol network using Mininet with OpenFlow1.3 support for IPv6, to evaluate IPv6 unicast and IPv6 multicast [19].

3 IPv6-Packets Flow Label Matching Mechanism

In this section, the matching approach by using Flow Label within IPv6 and the structure of flow table by using Flow Label within IPv6 is proposed and defined. Flow Label is a new field in the IPv6 protocol and it can be used for flow classification. And then, we shows the matching approach by using Flow Label in OpenFlow instead of L3-L4 Layer elements, the IPv6 source address, the IPv6 destination address, the protocol version, the TCP/UDP source port, the TCP/UDP destination port and Flow Label within IPv6 is introduced in Flow Table as the IPv6 required match fields to reduce the size of flow table. This paper considers the matching approach which flow label is used for match process and flow table can reduce the rate of IPv6 flow matching and the size of IPv6 Flow table, to provide the faster and more reliable network communication in OpenFlow.

3.1 Flow Label

Flow Label is a new field in the IPv6 protocol proposed and defined as the length of 20 bits. RFC defines the flow label which the source node can use the flow label of IPv6 header marked packets, and the source node request for specially treatment, such as, QoS services and other real-time transactions. Flow label can be resolved to meet the needs and rules by checking the flow label to determine which stream it belongs to. According to the forwarding rules, the routers and hosts which do not support the flow label need to flow label field to all zeros, and the receiving packets do not modified the value of flow label.

The traditional traffic classification process is as follows:

(1) Find a destination IP address;
(2) To compare the protocol number;
(3) Compare the destination port number (transport layer performed);
(4) Comparing the source IP address and filter address (here mainly is to determine whether the packet matches the filtering rules);
(5) Comparing the source port number.

However, a uniqueness flow must have the same attributes, including the source address, destination address, flow label. Therefore, we propose a matching method of using Flow label within the IPv6 protocol for OpenFlow which supports IPv6.

3.2 The Matching Approach by Using Flow Label Within IPv6 Based on OpenFlow

Flow is an important concept not only in the data flow communication network but also in OpenFlow. The Open Flow Switcher consists of one or more flow tables, which perform packet lookups and forwarding. The switcher communicates with the controller and the controller manages the switcher by the OpenFlow switch protocol. Using the OpenFlow protocol, the controller can add, update, and delete flow entries in flow tables. Each flow table in the switcher contains a set of flow entries; each flow entry consists of match fields, counters, and a set of instructions to apply to matching packets. A flow table entry consists of a set of L2/L3/L4 match conditions.

In the matching process of L3-L4 Layer which don't use the flow label, the OpenFlow agreement will check the packets which are the IPv4 packets or IPv6 packets, and compare the destination IP address. After the comparison of TCP/UDP source and destination port numbers, using multi-stream pipeline processing table can effectively enhance the flow table processing efficiency, especially in matching IP layer, IPv4 needed source IP address, destination IP address, and IPv6 IP address will be added to 128 bits, greatly increasing the flow table match delay. Before flow label field is introduced, the common matching process of L3/L4 layer is as follows:

(1) Compare the protocol number;
(2) Compare the source IP address and destination IP address which has 128 bits in IPv6;
(3) Compare the source port number and the destination port number (transport layer performed);

In the pipeline of multi-flow table, the matching process of flow table match is very complex as it is in Fig. 2, so that the delay of flow table match of OpenFlow Switcher will increases obviously, especially in the match of IPv6 address, which has 128 bits.

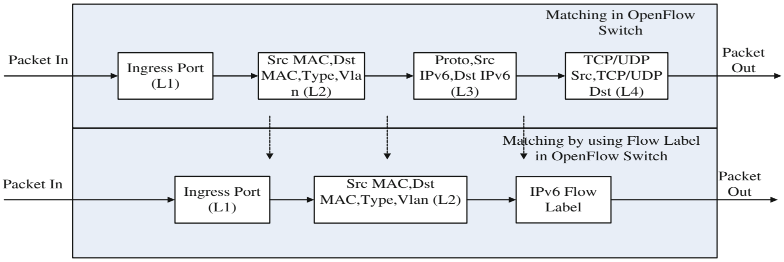

Fig. 2. IPv6 packets are matched against flow table in OpenFlow switcher

We propose a matching approach by using Flow Label within IPv6 feature in IPv6 based on OpenFlow instead of L3-L4 matching fields, such as IPv6 source address, IPv6 destination address, TCP/UDP source port, TCP/UDP destination port. In the IPv6 with flow label mechanism, we can use this field, flow label (20 bits), quintuple information will be combined to generate a random flow label serial number for each quintuple information, and generates the corresponding entries in the flow state, it can find the entry for traffic classification according to the value of each stream flow label. The different source node may randomly send the same flow label value, but the probability is very small (about 106 level), which can be considered that the source node will correspond the only flow label value in simple OpenFlow network. The possible Hash algorithm can be used in the matching of flow label. Thus, the process based on traffic classification IPv6 flow label can be simplified to a one-step process: Find the value of the stream flow state table based on the Flow Label. Obviously, in such a mechanism, matching process of match fields is simplified a lot when the IPv6 packets are matched against flow table.

3.3 Flow Table with Flow Label Within IPv6 Based on OpenFlow

Flow table is an important concept in OpenFlow, Flow Table are a pieces of the forwarding tables, as MAC table, IP table, ACL in tradition network. In OpenFlow agreement, each flow table composed by many Flow entries. Flow table entry is the smallest unit of flow table matched to each flow in network transmission. According to OpenFlow standard, A flow table consists of many flow entries, and each flow entry include: Match Fields which match packets, Priority matching precedence of flow entry, Counters which update when packets are matched, Instructions which can modify the action set or pipeline processing, timeouts that can maximum the amount of time or idle time before flow is expired by the switch, cookie of which the data value chosen by the controller. And a flow table can support L1-L4 matching, so the match fields can consists of L1-L4 matching element, such as L1 (Ingress port, physical port), L2 (VLAN ID, VLAN PCP, Ethernet source address. Ethernet destination address, Ethernet type), L3 (IP protocol number, IPv4 or IPv6 source address, IPv4 or IPv6 destination address, IP DSCP, IP ECN, Flow Label), L4 (TCP or UDP source port, TCP or UDP destination port), just as it is in Fig. 3.

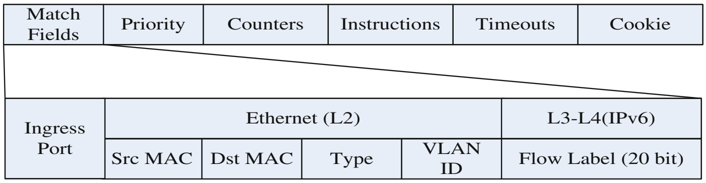

Fig. 3. Main components of a flow entry in a flow table with flow label

In this paper, we propose to use Flow Label fully within IPv6 protocol in place of the L3-4 match fields in IPv6 network match fields based on OpenFlow. For example, in IPv6 flow table based on OpenFlow, when a IPv6 host1 connect to a IPv6 host2, the Flow Labe will be assigned according to the Packet-in packet, not the match fields of L3-4 Layer, such as IPV6 source address, IPv6 destination address and TCP/UDP source port, TCP/UDP destination port. Especially in the match fields of IPv6 address, which has 128 bits, will increase the size of flow table obviously. Thus the controller automatically and naturally determines the flow label number of the flow based on its location in the IPv6 network.

3.4 The Analyses of Flow Label Matching Mechanism Based on OpenFlow

According to OpenFlow standard, OpenFlow defines the flow table entries of matching L1-L4 Layers match fields to lookup the match fields and forward packets. A flow table consists of flow table entries is designed to match against packets, which consists of the L1-L4 Layer metadata of matching flow table entry. When the packets are matched against multiple tables in the pipeline, the process of flow table match is very complex, the complexity of match is expressed:

$$T(n) = O(n! \times a(2^l)) \quad (1)$$

And n is the number of flow tables, a is the number of actions and l is the length of match fields. As the number of flow tables, the number of actions and the length of match fields is increasing, it will lead to the increasing of complexity in the pipeline process. So we analyze the flow label matching mechanism from theory and implementation in IPv6. The latency of packet forwarding is an important indicator of network performance. For the network performance of forwarding, we can use the arithmetic average to evaluate, and is shown in Eq. (2).

$$Average_{Latency} = \frac{1}{N}\sum_{i=1}^{n} \text{Latency}_i \tag{2}$$

For the fluctuation of delay, we use the standard deviation to evaluate, and it is shown in Eq. (3).

$$SD_{latency} = \sqrt{\frac{1}{N}\sum_{i=1}^{n} (Latency_i - Average_{Latency})^2} \tag{3}$$

And N is the number of all packets, $lantency_i$ is the latency of i packet. $Average_{latency}$ is the average of latency. If the SD is smaller, the network performance of fluctuation is better. The jitter of transmitting can react the stability of transmitting performance in network. The jitter of packet forwarding is shown in Eq. (4).

$$Jitter = \frac{1}{N}\sum_{i=1}^{n} (L_i - \bar{L})^2 \tag{4}$$

And N is the number of packets, L_i is the latency of the packet i. $\bar{L}$ is the average of the latency of n packets. According to OpenFlow1.3, OpenFlow defines the size of the match fields in IPv6 network based on OpenFlow, such as the size of IPv6 source address and IPv6 destination address are 128 bits, the size of IP Protocols is 8 bits and the size of TCP/UDP source port and TCP/UDP destination port are 16 bits, the size of IPv6 table Label is 20 bits. The size of flow table is shown in Eq. (5).

$$FL = Sum(match + prio + coun + inst + tout + cookie) \tag{5}$$

The compression of a flow table i of an OpenFlow switch at some moment is shown in Eq. (6).

$$Compression = (1 - \frac{FL'}{FL}) \times 100\% \tag{6}$$

And FL' is the size of flow table by using Flow Label, FL is the size of flow table by using IPv6 address and TCP/UDP port. In this paper, we mainly focus on the match fields in IPv6 based on OpenFlow, which uses flow label to replace the elements of L3-4 Layer, that uses IPv6 flow label (20 bits) instead of the IPv6 source address, IPv6 destination address, IP protocol number and TCP/UDP source port and TCP/UDP destination port. Therefore, The flow table i of the OpenFlow switcher at some moment is shown: $FL_i = match + otherelement$. The $match$ is the size of match fields of flow table i by using Flow Label or IPv6 address and the other element is the size of Priority, counters, instructions, timeouts and the cookie of flow table. The match fields of matching packets support the element of L1-4 Layer; the size of match fields of flow table i is $match = m_{l1} + m_{l2} + m_{l3} + m_{l4}$ and $m_{l1} - m_{l4}$ is the size of L1-4 Layer's match fields. The compression of a flow table i of an OpenFlow switcher is shown in Eq. (6),

$m_{l3} - m_{l4}$ is the size of L3-4 Layer's match fields of flow table i and e is the size of the L1-L2 Layer's match fields and others of flow table:

$$Comprsssion_i = (1 - \frac{m\prime_{l1-l2} + m\prime_{l3-l4} + otherele'}{m_{l1-l2} + m_{l3-l4} + otherele}) \times 100\% \quad (7)$$

For every table flow, m and e will be different. The compression of every flow table of an OpenFlow switcher at some moment is shown in Eq. (7), $Compression_i$ is the compress of a flow table i of an OpenFlow Switcher and n is the number of flow table of an OpenFlow switch: $Compression = \frac{1}{N}(\sum_{j=i}^{i+n} Compression_j)$. Analysis with Eq. (7), the gaps of m'_{l3-l4} and m_{l3-l4} is changeless to flow table because of L3-L4 Layer. So the sizes of m_{l1-l2} and e will be determined to the compress of flow table. In the common conditions of OpenFlow networks, the state of flow table accords with the complexity of Instruction. So there will be more space of flow table when there are more instructions in relative simple network, this mechanism of flow table which uses flow label will perform better.

4 Evaluation of IPv6-Packets Matching Approach

In this paper, we implement the evaluation to verify the feasibility and performance of our matching approach and Flow Table. We also implement the function of matching in Mininet 2.0. The controller is Ryu [20] which supports OpenFlow1.3 in Ubuntu 14.04. In the simulation, we used the 1-switch-2 hosts topology in Fig. 4. We did simulation with the matching approach by utilizing the Flow Label based on Openlow1.3 match fields in order to compare the difference performance between the flow table with IPv6 flow label and flow table including IPv6 addresses. The IPv6 matching approach implementation extends the OpenFlow1.3, we used OpenFlow1.3 because it is available at the time when the work was done, Through the evaluation metrics of the performance of network, The aim of the flow label matching approach obtained the best performance of the OpenFlow network system.

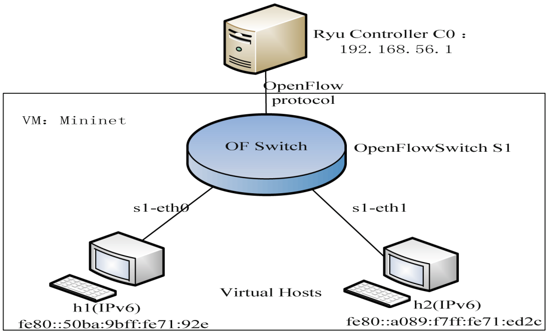

Fig. 4. Topology of 1-switch-n hosts simple network

We used three parameters, the latency of forwarding, the jitter of transmitting and the compression of flow tables, to evaluate the performance of network. We respectively added the flow table based on flow label and the flow table based on IPv6 address to the OpenSwitch in Mininet and we sent the 64 bytes packet per one second and send the 128 bytes UDP packets per one second in which the rate is 10 M. we compared the latency of 120 packets (2 min), the jitter of 30 s and the size of flow table in IPv6 based on the OpenFlow1.3 match fields and the matching by the model we proposed in Sects. 3.2 and 3.3. We assigned each one host an IPv6 address. Host 1 pings Host 2 and the IPv6 packet was analyzed by the Ryu, OpenFlow Protocol matched the flow table to make a decision on the forwarding of the flows. We got the statistical results of delay, jitter and flow table size of two matching mechanism during the simulation.

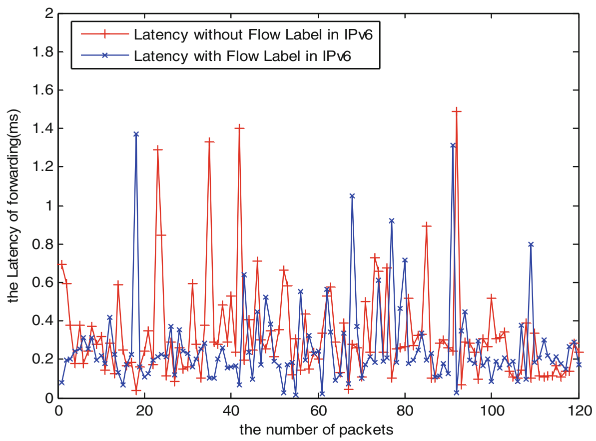

Fig. 5. The comparison of the latency between two different matching mechanisms

In Fig. 5, we use the arithmetic average and standard deviation of the delay to evaluate the performance of the network, according to the Eq. (2),

$$Average_{latency\ without\ flow\ label} = 0.3284\,\text{ms}$$

$$Average_{latency\ with\ flow\ label} = 0.2604\,\text{ms}$$

The average latency with IPv6 address is 0.3284 ms, but the average with flow label is 0.2604 ms in the forwarding of 120 IPv6 packets. From this value, we can see that the Latency of OpenFlow by using flow label in IPv6 network is smaller than the latency without flow label. And according to the Eq. (3),

$$SD_{latency\ without\ flow\ label\ in\ IPv6} = 0.0334\,\text{ms}$$

$$SD_{latency\ with\ flow\ label\ in\ IPv6} = 0.0167\,\text{ms}$$

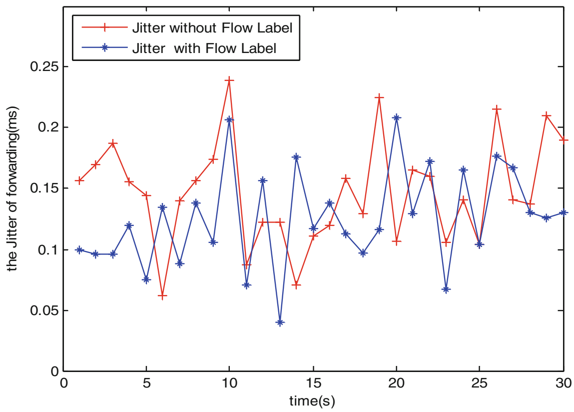

Fig. 6. The comparison of the jitter between two different matching mechanisms

From this value, we can see that the fluctuation of Latency by using flow label in IPv6 network is smaller than the fluctuation of latency without flow label. Thus, Fig. 5 shows that the latency without flow label is in general higher than the latency with flow label obviously.

And in Fig. 6, it shows that the jitter without flow label is higher than the jitter with flow label. The jitter time which is based on flow label is lower about 70 % than the matching mechanism without flow label in transmitting.

$$Average_{jitter\ without\,flow\,label} = 0.1469\,\mathrm{ms}$$

$$Average_{jitter\,with\,flow\,label} = 0.1254\,\mathrm{ms}$$

From these values, we can see that the fluctuation of Latency by using flow label in IPv6 network is litter smaller than the fluctuation of latency without flow label.

In Fig. 7, it shows the comparison of the size of flow table in two matching mechanism when host1 pings to h2 which send IPv6 packets in 1-switch-2 hosts topology. The size of flow table with flow label is less than the size of flow table without flow label obviously and the size of flow table with flow label is decreased about 20 % which compared with the size of flow table without flow label. So the flow table with flow label is better than the flow table without flow label in IPv6 based on OpenFlow1.3 in this scene.

For comparing the performance of different matching approach, which are between the flow table with flow label and the flow table without flow label, we recorded the latency of transmitting and the jitter between flow table with flow label and flow table without flow label when h1 pings to h2 which send IPv6 packets, in order to compare the performance of different matching approach in both network. In Figs. 5, 6, and 7, the performance of the latency, jitter and flow tables between flow table with flow label and flow table without flow label is obvious. From the result of simulation, we find that the network performance by using the Flow label could make OpenFlow perform better in 1-switch-2-IPv6 hosts topology.

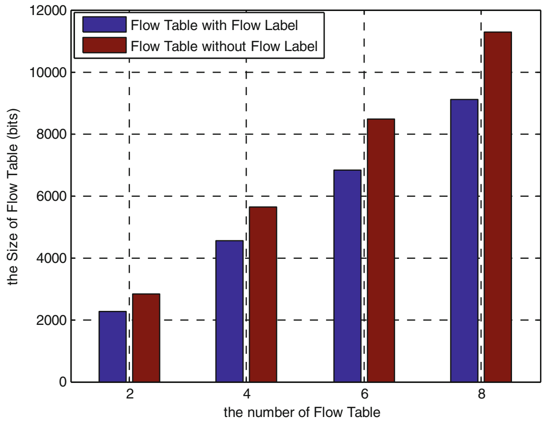

Fig. 7. The comparison of the size of flow table between flow table with flow label and flow table without flow label

5 Conclusion and Future Work

In this paper, we proposed a matching approach by using Flow Label within IPv6 protocol and a flow table structure by using Flow Label within IPv6 protocol in SDN based on OpenFlow, and compared the size of difference flow table between flow table with flow label and flow table without flow label in simple 1-switch-2-hosts topology based on OpenFlow1.3. Through the analysis and the simulation of the difference between the two flow tables in this paper, we concluded that the flow table by using Flow Label within IPv6 protocol can lead to the decrease of the latency of forwarding, the jitter of transmitting and the size of flow table space. The simulations compared the size of flow table between flow table with using flow label and flow table without using flow label and the result showed that the flow table by using Flow Label within IPv6 protocol can reduce the size of flow table effectively in simple network scenarios.

However, we will have a lot of work to do, we don't optimize the suitable flow label hash algorithm about IPv6 address and don't analyze the QoS and the security of OpenFlow which uses the flow label within IPv6 protocol. In addition, we experiment on simple network. We do not take these into consideration in this paper. In future work, we will talk about the analysis about the QoS and the security of OpenFlow which uses the flow label within IPv6 protocol, and our future work will involve more experiments and analyses of flow label scenarios.

Acknowledgement. This work is supported by the National Science Foundation, under grant No. 61471083 and grant No. 61300017. The Fundamental Research Funds for Central Universities (DUT15QY28, jg2015036).

References

1. Zhang, C.K., Cui, Y., Tang, H.Y., Wu, J.P.: State-of-the-art survey on software-defined networking (SDN). J. Softw. **26**(1), 62–81 (2015)
2. Deering, S.E.: Internet protocol, version 6 (IPv6) specification. RFC 2460. https://tools.ietf.org/html/rfc2460
3. Gu, D., Liu, X., Qin, G., et al.: VNET6: SOA based on IPv6 virtual network. In: Proceedings of the 2012 International Conference on Cloud and Service Computing, pp. 32–39. IEEE Computer Society (2012)
4. OpenFlow. http://www.OpenFlow.org/
5. Open Networking Foundation (2014). https://www.opennetworking.org
6. McKeown, N., Anderson, T., Balakrishnan, H., et al.: OpenFlow: enabling innovation in campus networks. J. ACM SIGCOMM Comput. Commun. Rev. **38**, 69–74 (2008)
7. Heller, B.: OpenFlow switch specification, version 1.0. 0 (2009)
8. O N F. OpenFlow switch specification version 1.2. 0 (2011)
9. OpenFlow Switch Specification 1.3.0. http://www.opennetworking.org
10. Specification-Version, OpenFlow Switch. "1.4. 0." http://www.opennetworking.org
11. Tseng, C.W., Yang, Y.T., Chou, L.D.: An IPv6-enabled software-defined networking architecture. In: 2013 15th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 1–3. IEEE (2013)
12. Xia, W., Tsou, T., Lopez, D.: A software defined approach to unified IPv6 transition. In: 2013 International Conference Special Interest Group on Data Communication (SIGCOMM). Hong Kong (2013)
13. Liu, X., Qin, G., Yan, S., Luo, Z., Yan, B.: VNET6: SOA based on IPv6 virtual network. In: 2012 International Conference Cloud and Service Computing (CSC), pp.32–39 (2012)
14. Batalle, J: On the implementation of NFV over an OpenFlow infrastructure: routing function virtualization. In: Future Networks and Services (SDN4FNS), pp 1–6 (2013)
15. Denicol, R.R., Fernandes, E.L., Rothenberg, C.E., et al.: On IPv6 support in OpenFlow via flexible match structures. J. OFELIA/CHANGE Summer School (2011)
16. IPv6 First-Hop Security: Ideal OpenFlow Use Case. http://blog.ipspace.net/2012/10/ipv6-first-hop-security-ideal-OpenFlow.html
17. Stallings, W.: Software-defined networks and OpenFlow. Internet Protoc. J. **16**, 2–14 (2013)
18. Araji, B., Gurkan, D.: Embedding switch number, port number, and MAC address (ESPM) within the IPv6 address. In: 2014 Third GENI Research and Educational Experiment Workshop (GREE), pp. 69–70. IEEE (2014)
19. Newman, D.R.: Technology Validation Experiment: IPv6 and Multicast Support on OpenFlow (2014)
20. Ryu. http://osrg.github.com/ryu/