A Segment List Management Algorithm Based on Segment Routing

Jianxin Zhou, Zhipeng Zhang, Ning Zhou*
School of Information Engineering, Wuhan University of Technology,
Wu Han 430070, China
e-mail: zjx@whut.edu.cn, 714139430@qq.com, zhouning@whut.edu.cn

***Abstract*—Segment routing is a new type of network architecture using the source routing paradigm. This new network technology can be directly applied to the MPLS architecture with no change to the forwarding plane. An ordered list of segments is encoded as a stack of labels. However, there is a physical limitation problem of Maximum Stack Depth (MSD). This paper proposed a segment list management algorithm of Longest Match Relay Push (LMRP), built a test system based on segment routing and Software-Defined Networking (SDN), and analyzed the number of labels and flow entries through mathematical modeling. Experiments comparing with other algorithms show that the LMRP algorithm can make multiple links share the same label according to the initial Forwarding Information Base (FIB). The simulation results demonstrate that LMRP can reduce the number of labels, flow entries and packet overhead, decouple routing calculation from segment list calculation, and can also realize packets forwarding of any long path through the source and relay nodes without MSD problem.**

Keywords-segment routing; MPLS, label stack; SDN; source routing paradigm

I. INTRODUCTION

Segment Routing (SR) is a new network architecture used to simplify the traditional network control plane. In RFC8402, Segment Routing Architecture was standardized by Internet Engineering Task Force (IETF) in July 2018 [1]. Segment routing originated from MPLS, but it achieved a revolutionary innovation. Due to its natural combination with Software-Defined Networking (SDN), it has gradually become the mainstream network architecture specification of SDN [2]. The centralized scheme of segment routing removes the complex LDP and RSVP-TE protocols of MPLS in the control plane. The controller is responsible for the distribution of Node-SID and Adj-SID. Segment routing adopts the source routing paradigm and maintains per-flow state only at the ingress node to the SR domain, and other SR routers only need to forward data packets according to SID without knowledge of the packets' other information. It realizes topology-independent and IP-optimized fast rerouting. These features of segment routing provide good support for SLA and improve the flexibility, availability and scalability of the network system [3]. In addition, segment routing can reuse the existing MPLS data planes, and the existing network devices needn't change or make small changes, which facilitates the progressive migration of existing networks [4].

SR-MPLS carries out path calculation for quality of service through traffic engineering. The ideal solution is to push enough labels at the source node, and then gradually pop labels from the stack at subsequent nodes until it reaches the target node. SR-MPLS uses the MPLS label header with a total of 32bits, of which the first 20bits are used to identify the SID. Since the current number of packet labels processed by ASIC on routers is limited to 5-10 [5], it is of great significance to use a more efficient SIDs label simplification method to reduce packet overhead, label resource cost and flow entry consumption. At the same time, due to physical limitation of Maximum Stack Depth (MSD), the relay nodes need to push additional labels to the packet header for the data stream with long path and more hops in the larger network, so that it can complete the packet transmission.

Based on simplified control plane of segment routing, centralized control and open programming of SDN, a Longest Match Relay Push (LMRP) algorithm for SIDs was proposed in this paper. Firstly, the segment routing forwarding tables are established through a routing algorithm (such as SPF: Shortest Path First) adapted to the characteristics of network data flow transmission. When a new data flow enters the network, the appropriate path is calculated, and the LMRP algorithm is used to convert the path into the simplified segment list. The source node and several relay nodes are calculated as routers which push labels to the packet, and the flow entries with the instructions of pushing the corresponding segment list are installed to these routers. The match fields are layer 2 to layer 4 information of this data packet. Labels of segment lists are pushed to the packet in the selected routers. Matching active segment and forwarding packet are performed on all routers of the path according to their segment routing forwarding tables independently. Experiments result on various network topologies show that the proposed algorithm LMRP can reduce consumption of label resource and flow entry resource and realize data stream forwarding in any long path without MSD problem.

II. RELATED WORKS

Giorgetti et al. [6] proposed the SR-D and SR-R algorithms based on unique shortest path matching and reverse matching, which reduced the average segment list depth (SLD) of data flow to a certain extent. Lazzeri et al. [7] proposed a simplified algorithm combining ECMP for weight restriction with segment list simplification, but the label calculation and routing calculation are highly coupled

978-1-7281-2184-0/19/$31.00 ©2019 IEEE

and cannot be separated for optimization. Guedrez et al. proposed the SR-LEA-A algorithm to simplify SIDs [8] and increase the resource cost of flow entries by adding forwarding entries about Adj-SID. And they proposed another TSIDs scheme which cannot fundamentally solve the limitation problem of MSD [5]. Cianfrani et al. [9] applied the MILP algorithm to SR domain to ensure the interaction between IP router and SR node, and discussed the flow table configuration method of ordered flow and unordered flow. Finally, the heuristic algorithm was used to determine the SR label required by the nodes' flow table. Huang et al. [10] proposed an improved segment routing structure for data plane based on the network programmability provided by OpenFlow, which reduced the overhead of additional flow entries and label space, and designed a path encoding scheme based on CSPF to minimize the segment list depth under a given maximum constraint.

These proposed label calculation methods can't address the problem of MSD physical limitation, and there is optimization space for the consumption of label resource. This paper proposed the LMRP algorithm which can further reduce the average segment list depth, the number of flow entries, decouple routing calculation and label calculation, without introducing additional forwarding table entries. It can calculate several corresponding segment lists for any long path without MSD problem, to complete segment routing forwarding work without network size limitation.

III. System Architecture

A. Three Fundamental Abstractions

The system structure based on SDN and segment routing is divided into three layers such as application layer, control layer and infrastructure layer as shown in Fig. 1. The proposed implementation of LMRP is mainly located in the application layer. The control layer parses and transmits the data packet, and the infrastructure layer is the underlying entity of the system function. The NBI (Northbound interface) API and SBI (Southbound Interface) serve as the interactive interfaces between three layers, and SBI uses the OpenFlow protocol. Detailed description of the three-tier implementation of the entire system is given as following.

In the application layer of the system, the segment routing application can be divided into topology discovery module, Forwarding Information Base (FIB) module, packet analysis module, route computation module, LMRP module and flow entry sending module, which are 6 modules in total. Topology discovery module for nodes, links, hosts discovery, stores the whole network topology. The FIB module builds forwarding tables for all nodes. Packet analysis module extracts packet header information of data link layer, network layer and transport layer. According to the extracted header information, the route computation module uses different routing algorithms to allocate path for the new data stream. The LMRP module will calculate the shortest segment list of the allocated path. The flow entry sending module install some kinds of flows entries to the specified node.

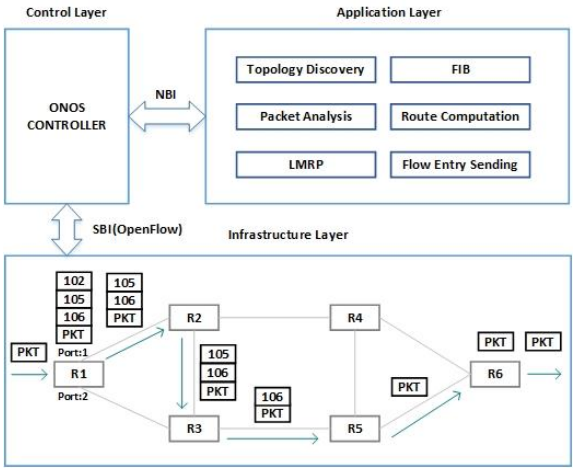

Figure 1. System architecture based on segment routing and SDN.

The control layer of the system, namely SDN controller, provides the abstract interfaces of services to the application layer. The segment routing application in the application layer returns the processing method to the control layer after making a business decision based on the service information. The control layer then sends the specific operation message to the equipment of the forwarding layer to complete the purpose function.

The infrastructure layer, also called as forwarding layer, composed by SDN routers which can support the mainstream SBI protocol OpenFlow, can establish normal connection with SDN controller. It can process the packet transmission among host, router and controller. By information interaction, the SDN controller configures various flow entries on the nodes, such as FIB flow entries, stack flow entries, link discovery and ARP flow entries, and the default IP flow entries. The routers can forward packets based on the configured flow entries.

B. Working Flow

When the system is initialized, the SDN controller performs node, link and host discovery, collect and stores the network topology information. The application specifies that the unique Segment Routing Global Block (SRGB) of the network is 100-200, then the Node-SID of R1 is 101, the Node-SID of R2 is 102, and so on. The target paths between nodes is calculated by Shortest Path First (SPF) algorithm. If there are multiple shortest paths with equal number of hops, one of them is selected randomly. According to the shortest path, the SDN controller sends the source node a forwarding table flow entry, of which the match field is the Node-SID of the destination node, and the instructions are the shortest path's corresponding output port. Partial flow entry tables of router R1 in Fig.1 is shown in Table 1 as example. If the destination node is the adjacent node, popping the top label is required. At the same time, MAC flow entries are installed to all edge nodes connected to hosts, and when the packet is received by edge node at the end of the path, it will be sent to the corresponding host according to the MAC address.

TABLE I. FLOW TABLES OF R1 AFTER INITIALIZATION

R1 Flow Tables			
Stage	***MatchFields***	***Priority***	***Instructions***
0	dl_type=IP	5	Output:Controller
1	102	10	Pop;Output Port:1
1	103	10	Pop;Output Port:2
1	104	10	Output Port:1
1	105	10	Output Port:2
1	106	10	Output Port:1

When a source host connected to R1 sends a data stream to the destination host connected to R6, several packets are packaged in OpenFlow header and uploaded to the SDN controller according to the default IP flow entry with priority 5.

The SDN controller then uploads the message information to the application layer's segment routing APP according to the northbound interface. After parsing the header domain information of the packet, the optimal path for the data flow is calculated on the basis of network topology, the QoS information and the specified routing algorithm. Then LMRP is used to calculate the simplified segment list of the allocated path, and SDN controller send the flow entries of the corresponding segment lists to selected nodes in the path for pushing labels.

Because the data message cannot be matched twice with the flow entry in the same stage flow table, the pipeline of two stages of flow tables is used to build the same forwarding tables. In each node for pushing labels, it is required to push segment list at stage 0 in the flow tables, and to match the forwarding tables at stage 0 in the flow tables. But in other nodes, it can match the forwarding table at stage 0 in the flow tables.

The path allocated to the new data stream is {R1,R2,R3,R5,R6} , as shown in Fig. 1. According to LMRP calculation, the Node SID to be pushed in turn is 106,105,102. When MSD $\geqslant$ 3, only the segment list {106,105,102} pushed in the source node R1 is needed. The SDN controller sends a new flow entry of stage 0 to R1, as shown in below Table II.

TABLE II. NEW FLOW ENTRY OF R1 INSTALLED BY SDN CONTROLLER.

Stage	***MatchFields***	***Priority***	***Instructions***
0	Packet Five-Tuple	20	Push:106,105,102;resubmit(1)

After new flow entry installed on the R1, when the source host sends new data packet to R1, according to the flow entry of priority (value of 20, higher than the default IP flow entry) and match fields (Five-Tuple: source IP address, destination IP address, source port, destination port and protocol), R1 executes the instructions, which pushes 106,105,102 in turn into the packet header, 102 for the current top label. And then the packet is sent to stage 1 flow table. The stage 1 flow table matches the Node-SID of 102, so R1 pops the top label 102 and sends the packet to port 1 for R2. R2 sends the packet directly to R3 without popping labels on its forwarding entry. R3 pops the top label 105 and sends it to R5. R5 pops the top label 106 and sends it to R6, which then sends the packet to an adjacent host based on the MAC flow entries. The transmission process of a new data stream is illustrated in Fig. 1.

IV. MATHEMATICAL MODEL

A. Longest Path Matching Analysis

Because the number of labels in the segment list is an important part of the network resource overhead, optimizing the number of labels is important to improve system performance. Using the simplified segment list algorithm with the method of the longest path matching can maximize the reduction the labels of a single flow. Along the allocated path, the longest path matching algorithm is matching the label of current node with forwarding tables of all the nodes in the sub-path, which is from node with last SID label to current node. If all nodes of sub-path is matched, the data flow can be forwarded to the current node, and continue matching. Otherwise, it indicates that a single label can reach the last node farthest, push the label of last node to segment list. Then a new label needs to matching subsequent nodes in the path. If the number of labels pushed by a single node reaches the upper limit (MSD), the farthest matched node is replaced as a new relay node for pushing labels. The mathematical proof of the longest path matching method is given below.

$$P = \bigcup_{v_i \in V_P} P_{v_i} = \bigcup_{l_j \in L_P} P_{l_j} = \bigcup_{e_k \in E_P} P_{e_k} \qquad (1)$$

P represents the path allocated to a flow in (1). P_{v_i}, P_{l_j} and P_{e_k} respectively represent subpaths of path P decomposed according to nodes for pushing label, labels of segment list and unidirectional side links as shown in the Fig. 2, in which the MSD is 2. V_P, L_P and E_P respectively represent the sets of nodes for pushing label, labels of segment list and unidirectional side links according to the LMRP algorithm. V_P^*, L_P^*, E_P^* respectively are used to represent the same sets calculated by other methods.

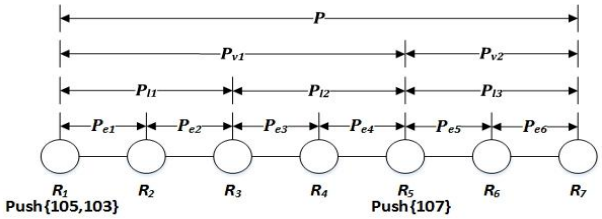

Figure 2. Path division which decomposes P into P_{v_i}, P_{l_j} and P_{e_k}

When $P_{l_j} = P_{l_j^*}(1 \leq j < q)$, there is $l_q \neq l_q^*$. Because the path of new label l_q^* can not be longer than the the path with the farthest node for matching forwarding tables, we can conclude $P_{l_q} \supseteq P_{l_q^*}$. Then for the path of the next

allocated label l_{q+1} and l_{q+1}^*, there will be two kinds of circumstances.

(1) $P_{l_q} \supseteq P_{l_q^*} \cup P_{l_{q+1}^*}$. It's obviously that $P_{l_q} \cup P_{l_{q+1}} \supseteq P_{l_q^*} \cup P_{l_{q+1}^*}$ because the set of paths on the left is larger.

(2) $P_{l_q} \subset P_{l_q^*} \cup P_{l_{q+1}^*}$. As long as $l_{q+1} = l_{q+1}^*$, there must be $P_{l_q} \cup P_{l_{q+1}} = P_{l_q^*} \cup P_{l_{q+1}^*}$, and when $l_{q+1} \neq l_{q+1}^*$, there may be $P_{l_q} \cup P_{l_{q+1}} \supset P_{l_q^*} \cup P_{l_{q+1}^*}$, so it also true that $P_{l_q} \cup P_{l_{q+1}} \supseteq P_{l_q^*} \cup P_{l_{q+1}^*}$.

Above (1) (2) shows that $P_{l_q} \cup P_{l_{q+1}} \supseteq P_{l_q^*} \cup P_{l_{q+1}^*}$. By mathematical induction, we can derive

$$\bigcup_{1 \le j \le p} P_{l_j} \supseteq \bigcup_{1 \le j \le p} P_{l_j^*} \tag{2}$$

And we also have

$$P = \bigcup_{l_j \in L_P} P_{l_j} = \bigcup_{l_j^* \in L_P^*} P_{l_j^*} \tag{3}$$

So there is $card(L_P) \le card(L_P^*)$, which means adopting the longest path matching method, the labels will get the minimum number. The same result can be derived if the longest path matching is done from the reverse direction. The results of the mathematical proof are applicable to the lable allocation scheme through initial forwarding table matching.

B. Flow Entry Analysis

The number of flow entries of all nodes is analyzed and calculated below. It is assumed that the network has *M* data streams, n is the number of nodes, and l is the number of links.

$$N_{FIB} = 2n(n-1) + 4l \tag{4}$$

In Equation (4), N_{FIB} represents the sum of flow entries of forwarding tables by all nodes. It distinguishes top and bottom of the label stack to pop label and set MPLS or IP for the type of packets. It uses two stages of flow tables.

$$N_F = \sum_{1 \le m \le M} \left\lceil \frac{N_f^L m}{N_{MSD}} \right\rceil \tag{5}$$

In Equation (5), N_F represents the number of flow entries for pushing labels into *M* flows. $\lceil . \rceil$ represents the integer function for rounding up. $N_f^L m$ is the total number of labels for the mth flow and N_{MSD} is the value of MSD.

$$N_{MAC} = N_{Host} \tag{6}$$

In Equation (6), N_{MAC} represents the number of flow entries of which the matching fields are MAC addresses for packets sent by the edge nodes to the adjacent hosts, and N_{Host} represents the number of hosts in the edge of current network.

$$N = N_{FIB} + N_F + N_{MAC} + N_{other} \tag{7}$$

In Equation (7), *N* represents the number of flow entries for all nodes when there is M flows. N_{other} represents other flow entries, the default IP flow entries for uploading new packets to SDN controller and flow entries for link discovery, and so on.

V. Algorithm Design

The LMRP algorithm generates the simplified segment lists of the allocated path P, which correspond to each node for pushing labels. The first loop traverses nodes in turn from P to determine whether to replace the current node with a new one to push the label. The second loop traverses the farthest nodes. The third loop traverses the path from the node for pushing labels to the farthest node to determine whether the last SID can be used to go through the path. If not, the third loop will exit and it will push the new label. When the segment list reaches MSD, the specified flow entry is sent to the target node v_p. The time complexity of algorithm 1 is $O(m^2)$, where m is number of network links.

Algorithm 1 Longest Match Relay Push

Input: The allocated path P, v_i is the ith node in P, l_k is the kth link in P, n is number of last node for the P, global forwarding table T, maximum stack depth (MSD).
Output: Flow entries of nodes for pushing labels

```
for i ← 0 to n − 1 do
    if nextDevice == true then
        nextDevice ← false
        v_p ← v_i
    end if
    for j ← i to n − 1 do
        for k ← i to j do
            port ← getLinkPort(v_k, l_k)
            if (FIB (T, v_k, v_{j+1}) == port) then
                continue;
            else
                segment[n_L] ← getSID(v_j)
                nextSegment ← true
                n_L++
                break
            end if
        end for
        if(nextSegment == true) then
            nextSegment ← false
            break
        end if
    end for
    i ← j
    if(n_L == MSD) then
        nextDevice ← true;
        n_L ← 0
        installSREntry(v_p, segment)
    end if
    if(k == n − 1) then
        segment[n_L] ← getSID(v_n)
        installSREntry(v_p, segment)
        break
    end if
end for
```

VI. EXPERIMENTAL RESULT ANALYSIS

In the experimental experiment, the operating system is Ubuntu 14.04 of x64 and the hardware equipments are CPU of 2.80 GHz and RAM of 8.00GB. ONOS 1.14.0 is used as SDN controller, Mininet 2.3.0d4 as network simulator, Open vSwitch 2.5.7 as virtual switch, and OpenFlow 1.3 as SBI protocol. The label algorithms include Strict Encoding (SE) [8], Multi-Protocol Label Switching (MPLS), SR-D [6], and LMRP. The routing algorithm is Random K Shortest Path(RKSP). NSFNet, Fat-Tree (k=4), COST266 and Matrix(k=6) are selected as the experimental topologies. The following table describes the topological parameters. The hosts are connected to the peripheral nodes of each network topology. The uplink bandwidth and downlink bandwidth of all links are set to 10Mbps, and the link delay is 3ms. We randomly select the source and destination hosts to send multiple data streams, and take the average value as the experimental result including number of labels, packet overhead and number of entries.

TABLE III. PARAMETERS FOR EACH TOPOLOGY.

Topology	Vertices	Edges	Hosts
NSFNet	14	20	9
Fat-Tree(k=4)	20	32	16
COST266	28	41	21
Matrix(k=6)	36	60	20

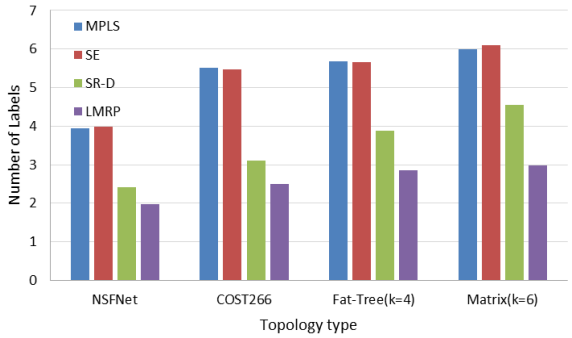

Figure 3. Number of labels in test topologies using different algorithms.

Fig. 3 shows the average number of labels for four algorithms using the random data flow scheme under four network topologies. Because SE and MPLS need to use new labels in each hop of the path, the average numbers of labels are larger in the four network topologies. The two algorithms have the same number of labels in the same path, and the slight difference is caused by the randomization of data flows. Although SR-D has significantly reduced the number of labels so that multiple links share the same label, for some shortest paths with the same hops, the flow entries must be increased, resulting in a higher number of labels than LMRP. LMRP has the best optimization effect on the number of labels. Compared with SE and MPLS, it reduces labels by 51.4% on average, while LMRP reduces by 24.9% on average in contrast to SR-D, so LMRP can maximize the reduction of label resource.

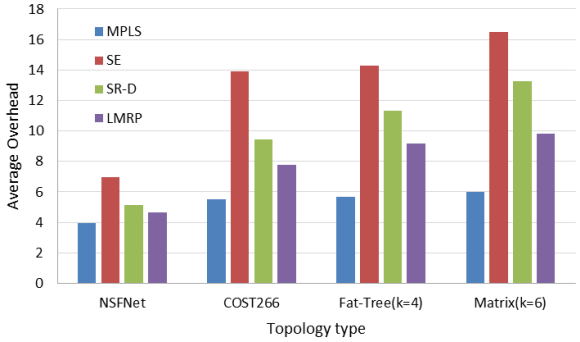

Figure 4. Average overhead in test topologies using different algorithms.

Fig. 4 shows the average packet overhead for four algorithms using the random sending data flow scheme under four network topologies. Here the packet overhead of each data stream is the sum of SLD on each link of the allocated path and the MSD is set to 10. Because MPLS adopts label switching technology, the packet header only has a unique MPLS label on every hop, so its average overhead is minimal. SE, SR-D and LMRP adopt source routing paradigm accounting for the larger number of labels in each link, so their average overhead is larger than MPLS. The three algorithms for the same link of the same path reduce the number of labels in turn, so the average overhead decrease in turn. Compared with the SE, the average overhead in LMRP fall by 38.4%, and compared to SR-D it decreases by 17.9% in LMRP, but compared with the MPLS it increased by 46.0% for LMRP. Therefore, LMRP can reduce the data streams' packet overhead to some extent.

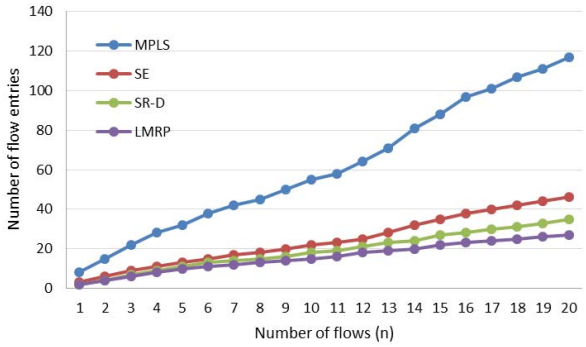

Figure 5. Relationship between the number of flow entries and the number of data streams using different algorithms in Matrix(k=6).

Fig. 5 shows the relationship between the number of flow entries and the number of data flows for four algorithms under the Matrix(k=6) topology. In this figure, only flow entries for pushing labels are calculated and other flow entries are ignored because they are installed in system initialization phase. The three algorithms of source routing adopt relay mode. Here the MSD is set to 3. Because MPLS is based on tunnel technology, each node in the path needs flow entries to exchange labels, so the number of flow entries is relatively large. However, SE, SR-D and LMRP all

adopt source routing paradigm. SDN controller only needs to send flow entries of pushing labels to the source node and relay nodes of the data flow. Therefore, the number of flow entries required by a single data flow is small and equal to the number of nodes which pushing labels. For each link of the same path, the number of labels used by the three algorithms, SE, SR-D, LMRP, decreases in turn, so the number of nodes pushing labels decreases in turn, so the number of flow entries also decreases in turn. For 20 data streams, the number of flow entries used in LMRP is 23.1% of MPLS, 58.7% of SE and 77.1% of SR-D. Therefore, LMRP can optimize the consumption of flow table resource.

VII. CONCLUSIONS

In this work, we propose an LMRP algorithm to solve the problems of physical limitation of MSD and insufficient label resource using segment routing network and SDN architecture. A test system combining segment routing and SDN is built for experimental simulation. We analyze the initialization phase and data packet processing phase of the system in detail. We also describe the segment routing forwarding process of a data stream along an allocated path. Through mathematical derivation, it is proved that the shortest segment list can be obtained by the longest path matching according to the initial forwarding table. Experiments comparing with other algorithms show that LMRP algorithm can make multiple links share the same label, can reduce the number of labels, flow entries and packet overhead, and can also realize the stream data transmission with arbitrary long path by relay nodes without MSD problem.

REFERENCES

[1] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018.

[2] Z. Li, L. Huang, H. Xu and G. Zhao, "Segment routing in hybrid software-defined networking," 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, 2017, pp. 160-165.

[3] A. Cianfrani, M. Listanti and M. Polverini, "Incremental Deployment of Segment Routing Into an ISP Network: a Traffic Engineering Perspective," in IEEE/ACM Transactions on Networking, vol. 25, no. 5, pp. 3146-3160, Oct. 2017.

[4] A. Giorgetti, A. Sgambelluri, F. Paolucci, F. Cugini and P. Castoldi, "Segment routing for effective recovery and multi-domain traffic engineering," in IEEE/OSA Journal of Optical Communications and Networking, vol. 9, no. 2, pp. A223-A232, Feb. 2017.

[5] R. Guedrez, O. Dugeon, S. Lahoud and G. Texier, "A new method for encoding MPLS segment routing TE paths," 2017 8th International Conference on the Network of the Future (NOF), London, 2017, pp. 58-65.

[6] A. Giorgetti, P. Castoldi, F. Cugini, J. Nijhof, F. Lazzeri and G. Bruno, "Path Encoding in Segment Routing," 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, 2015, pp. 1-6.

[7] F. Lazzeri, G. Bruno, J. Nijhof, A. Giorgetti and P. Castoldi, "Efficient label encoding in segment-routing enabled optical networks," 2015 International Conference on Optical Network Design and Modeling (ONDM), Pisa, 2015, pp. 34-38.

[8] R. Guedrez, O. Dugeon, S. Lahoud and G. Texier, "Label encoding algorithm for MPLS Segment Routing," 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, 2016, pp. 113-117.

[9] A. Cianfrani, M. Listanti and M. Polverini, "Incremental Deployment of Segment Routing Into an ISP Network: a Traffic Engineering Perspective," in IEEE/ACM Transactions on Networking, vol. 25, no. 5, pp. 3146-3160, Oct. 2017.

[10] L. Huang, Q. Shen, W. Shao and C. Xiaoyu, "Optimizing Segment Routing With the Maximum SLD Constraint Using Openflow," in IEEE Access, vol. 6, pp. 30874-30891, 2018.