Path Selection with Joint Latency and Packet Loss for Edge Computing in SDN

Chin-Lin Hu*, Chao-Yu Hsu*, Sod-Erdene Khuukhenbaatar*, Yamkhin Dashdorj†, and Yongqiang Dong‡§
*Department of Communication Engineering, National Central University, Taoyuan 32001, Taiwan
†Department of Information Network and Security, Mongolian University of Science and Technology, Ulaanbaatar, Mongolia
‡School of Computer Science and Engineering, Southeast University, Nanjing, 211189, China
§ Key Lab. of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, China
chihlin.hu, chaoyu.hsu, sodkoh@g.ncu.edu.tw; dashdorj@must.edu.mn; dongyq@seu.edu.cn

Abstract—**This paper addresses QoS routing in an integrated network architecture based on the software-defined networking and edge computing technologies. Our study leverages the benefits of edge computing to differentiate data flows of different service types, as well as the programmable features of the software-defined networking to manipulate the forwarding behavior on data flows. Joint QoS-specific factors of latency and packet loss are used to design a novel path selection method for data delivery in such an integrated network. Simulation under heavy traffic patterns shows that our method performs better with lower latency and packet loss rate simultaneously as compared with typical OSPF and some greedy-like routing methods.**

Index Terms—**Path selection, QoS routing, edge computing, software defined networks.**

I. INTRODUCTION

Traditional traffic engineering and network resource management systems [1] employ dedicated protocols, such as Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP), Resource ReServation Protocol (RSVP), Next Steps in Signaling (NSIS), General Internet Signalling Transport (GISP) and Multi-Protocol Label Switching (MPLS), to reserve resources along the end-to-end data delivery in order to guarantee the quality of services (QoS) using the integrated service model. Those protocols are not routing but transport controlling protocols. Hosts and routers use those protocols to request resources in each node along a routing path to meet an acceptable level of QoS for data flows. Deploying those protocols in the network indeed complicates the router design. Significant cost by protocol processing and management can unfortunately deteriorate the performance of routers when an increasing number of data flows with different QoS levels come into the Internet.

To tailor traffic engineering against an exploding data volume in the Internet, the progress of Internet technologies account for software-defined networking (SDN) [2][3][4] and edge computing [5][6]. SDN enables the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. Edge computing is appealed to offload the cloud computing for leveraging distributed computation and storage resources of edge nodes used by hosts and applications in near fields. Hence, combining SDN and edge computing to establish an integrated network framework can make the network cost-effective, manageable and adaptable against network dynamics. Therefore, SDN can convey data flows from any ingress edge nodes to egress edge nodes across any specific QoS-specific routing paths over the network.

This paper designs a QoS-specific path selection method (PSM) with a joint effect of minimizing delay and packet loss for any end-to-end data delivery in an integrated Edge-SDN network. The PSM design contains four phases. First, PSM calculates request parameters in terms of bandwidth, delay and packet loss respectively. Second, PSM chooses an initial set of candidate paths for network services based on remaining bandwidth resource. Third, PSM calculates the parameter combination of delay and packet loss for all of satisfied paths. Finally, PSM selects the best path. To examine the efficiency of PSM, we conduct a synthetic simulation based on the Abilene Internet topology. Performance data about delay, packet load and accumulated number of received packets in sensitivity to the total of data flows and traffic workload are derived as compared with the Open Shortest Path First (OSPF) [7] and Greedy schemes. Results show that the OSPF obtains the best performance in case of lightweight traffic workload, while PSM remains performance close to OSPF. In case of heavy traffic workload, PSM is superior with lower latency and packet loss. There, our study investigates that PSM can sufficiently satisfy the QoS need.

The rest of the paper is organized below. Section II reviews several recent studies on QoS routing in SDNs. Section III describes the system model and specifies PSM with overload function formulations. Section IV shows the simulation and performance results. Section V concludes this paper.

II. RELATED WORK

Traditional routing techniques, such as OSPF [7], and Dynamic Source Routing (DSR) [8], use the shortest-path rules to deliver data flows in the network. Because the network topology can vary and the traffic patterns of data flow are not uniform, a plain shortest-path routing technique may cause long latency, high packet loss rate or network congestion when delivering abundant data flows into one path. Recent studies [9][10] discusses QoS management with the advantages of integrating SDN in various network applications, e.g., IoT, mobile cellular, WiFi, Fiber networks. [9] focused on minimizing latency of critical traffic through SDNs. [10] designed a traffic-aware QoS routing scheme for incoming flows in software-defined IoT (SDIoT) networks. A greedy heuristic scheme based on Yen's k-shortest path algorithm was proposed to compute optimal routing paths, depending on delay or packet-loss measures of all flows in SDIoT networks. To maximize network performance in a heterogeneous network, [11] performed QoS routing by calculating the weighted Euclidean distance between the delay and packet-loss parameters and the performance of candidate nodes with respect to each data flow. Besides, [12] proposed an architecture model where federated SDN controllers collaborate synergically for QoS management in integrated Fiber-wireless (FiWi)-Edge IoT networks. This work considered not only the edge weights but also the node weights for a graph derived from the SDN topology. Then, an extended Dijkstra's algorithm was implemented to find the shortest path in terms of end-to-end latency under the Abilene network. Compared with prior studies, our study focuses on edge computing combined with SDN to choose suitable routing paths based on joint QoS parameters of latency and packet loss simultaneously.

III. SDN-EDGE FRAMEWORK MODELING

Let $G(V, L)$ represent the SDN system, where a node set $V = V_i \cup V_f$ consists of an ingress node set V_i and a forwarding node V_f. L is the set of links, and $\{i, j\}$ is a link from node n_i to node n_j and $n_i, n_j \in V$. Given with a set of distinct services, denoted as S, S_e represents the services available at any edge node $e \in E$ where E denotes the set of edge nodes. Then, we define $f^{T(k)}$ to describe the network flow of service k and $T(k) = (o_k, d_k, b_k, t_k, l_k)$, and $k \in S_e$ as a characteristic function of service k which refers to several attributes, including an edge node o_k, a destination node d_k, bandwidth, delay and packet loss parameters denoted as b_k, t_k, and l_k, respectively. In addition, a binary variable x_i^e describes the connection relationship between an edge node $e \in E$ and ingress node n_i in the SDN, as defined below.

$$x_i^e = \begin{cases} 1, & \text{if } e \text{ connects with ingress node } n_i, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Given that the system has P_k paths satisfying a service k. Let P_w define a path set: $P_w = \{n_a, n_b, \dots, n_c\}$ indicates a reachable path w from an ingress node n_a to a destination node n_c via some traversing nodes such as n_b for $n_a \in V_i$, $n_b, n_c \in V$, $w \in P_k$. Also, let $P_w^{M(w)}$ describe a path P_w in a characteristic form. Here, $M(w) = (n_a, n_c, b_r, t_r, l_r)$ is a characteristic function of P_w, where n_a and n_c are the first and the last nodes in P_w, as well as b_w, t_w and l_w denote bandwidth, delay and packet loss parameters of P_w, as follows.

$$\begin{aligned} b_w &= min\{b(n_a, n_b), \dots, b(n_x, n_c)\}, \\ t_w &= t(n_a, n_b) + \dots + d(n_x, n_c), \\ l_w &= 1 - \prod[\{1 - l(n_a, n_b)\}, \dots, \{1 - l(n_x, n_c)\}], \end{aligned} \tag{2}$$

where $b(i, j)$, $t(i, j)$ and $l(i, j)$ represent bandwidth, delay and packet loss parameter from a link from node n_i to node n_j.

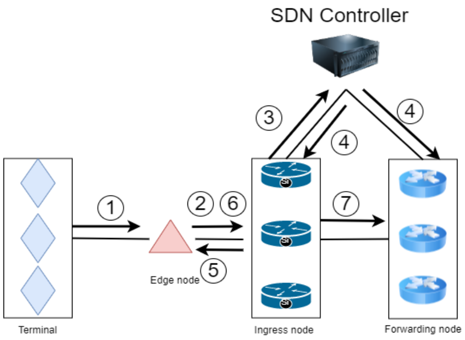


Fig. 1. Flow chart of finding the best ingress node.

Specifically, the SDN system conveys data flows from any ingress to egress edges nodes across a network. An SDN controller monitors network states, configures data flow tables on switches, and decides the routes of data flows over switches in a network. It collects the status information of nodes and link quality, such as remaining bandwidth capacity, link latency and packet loss parameters, and thus decides QoS routing paths. Fig. 1 shows the procedure in steps of handling data flows and forwarding in an integrated SDN-Edge network, as follows. (1) Terminal hosts in a certain network domain generate and send the data to an edge node in charge of this particular network domain. (2) When an edge node receives the data, it classifies the service types and service requests, and then announces this information to all of ingress nodes connected via the broadcast information. (3) After receiving this broadcast information, ingress nodes forward it to an SDN controller directly. (4) With this information, the SDN controller attempts to find the best path between the ingress node and the destination node via our proposed path selection method that will be described in Section IV. The SDN controller then notifies the first ingress node and updates the flow tables of which nodes are placed along the path. (5) When some ingress node receives the response message from the SDN controller, it asks the edge node to send the data packets. (6) The edge node receives this response and sends the data packets to ingress node which further forwards the packets to the destination node. (7) The ingress node or forwarding node can determine how to treat incoming packets according to its flow table. (8) Finally, all data packets are sent to the destination in light of the path decision by the SDN controller.

IV. PATH SELECTION METHOD (PSM) DESIGN

This section describes the proposed path selection method in use for determining the path based on the similarity measure between the service characteristics of network flows. Candidate paths for each network flow can be updated based on available bandwidth resource.

The selection of the path is based on the fittingness function (FF) which has two basic features: (a) if the value between 0 and 1, the slope is in a sharply ascending level, and (b) if the value bigger than 1, the slope is in a slowly descending level. Thus, by means of a fittingness function, we can dynamically adjust network resource to satisfy the user service. That is, edge nodes on the border networks can fine tune network resource allocation to data flows to/from the SDN. Particularly, if the ratio value of the offered parameter on a path to the requested parameter by the edge node is smaller than 1, its weighted value will be controlled to fall quickly. Conversely, if the value is bigger than 1, its weighted value should fall slowly.

The following specifies the PSM. Note that three types of user service, including bandwidth-sensitive, delay-sensitive and packet-loss-sensitive services are considered. So, edge nodes in the boundary of an SDN will assist in classifying these service types from/to the SDN.

1) Calculate the request parameter: Given with specific service types, b_r, t_r and l_r represent bandwidth, delay and packet-loss parameters by service requests belonging to an edge node e, respectively, and are defined below:

$$\begin{aligned} b_r &= b_1 + \cdots + b_{S_e}, \\ t_r &= min\{t_1, \ldots, t_{S_e}\}, \\ l_r &= l - \prod[\{1 - l_1\}, \ldots, \{1 - l_{S_e}\}]. \end{aligned} \tag{3}$$

2) Calculate a candidate path set for network services: For each network flow, it is to select the paths whose remaining available bandwidth resource can fulfill the flow request, i.e., $b_w \geq b_r$ where b_w refers to (2). Here, suppose that P_k paths can satisfy this condition, which will be used in the next step.

3) Calculate FF factors for all of satisfied paths: Let $U^w_{t,t}$ and $U^w_{t,l}$ denote the delay and packet loss factors for a delay-sensitive service on a path w and $w \in P_k$. And, let $U^w_{l,t}$ and $U^w_{l,l}$ denote the delay and packet loss factors for a packet-loss-sensitive service on a path w and $w \in P_k$. The measures of $U^w_{t,t}$, $U^w_{t,l}$, $U^w_{l,t}$ and $U^w_{l,l}$ are given below.

$$U^w_{t,t} = \frac{[\alpha * (t_w/t_r)]^\beta}{1 + [\alpha * (t_w/t_r)]^\beta}, \tag{4}$$

$$U^w_{t,l} = \frac{[\alpha * (l_w/l_r)]^\beta}{1 + [\alpha * (l_w/l_r)]^\beta}, \tag{5}$$

$$U^w_{l,t} = \frac{[\alpha * (t_w/t_r)]^\beta}{1 + [\alpha * (t_w/t_r)]^\beta}, \tag{6}$$

$$U^w_{l,l} = \frac{[\alpha * (l_w/l_r)]^\beta}{1 + [\alpha * (l_w/l_r)]^\beta}. \tag{7}$$

Then, τ is a normalization factor used to ensure that the FF metric does not exceed 1, and is given below:

$$\tau = 1 - e^{\frac{1}{(\beta-1)^{\frac{1}{\beta}} + (\beta-1)^{\frac{1-\beta}{\beta}}}}. \tag{8}$$

Finally, let $v_{d,w}$ and $v_{l,w}$ represent respective performance results of a delay-sensitive service and a packet-loss-sensitive service on a path w for $w = 1, \ldots, P_k$, and are given below:

$$v_{d,w} = c_1 \frac{1 - e^{-\frac{U^w_{t,t}}{\alpha * (t_w/t_r)}}}{2\tau} + (1 - c_1) \frac{1 - e^{-\frac{U^w_{t,l}}{\alpha * (l_w/l_r)}}}{2\tau}, \tag{9}$$

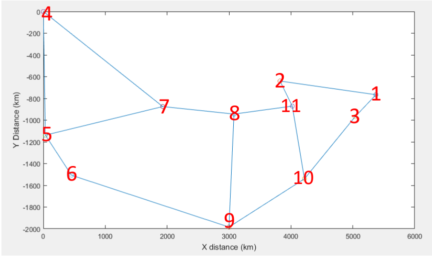


Fig. 2. Abilene topology.

$$v_{l,w} = c_2 \frac{1 - e^{-\frac{U^w_{l,t}}{\alpha * (t_w/t_r)}}}{2\tau} + (1 - c_2) \frac{1 - e^{-\frac{U^w_{l,l}}{\alpha * (l_w/l_r)}}}{2\tau}. \tag{10}$$

4) Average the delay and packet-loss-sensitive values: Let v_i represent the weighted value of a path w.

$$v_w = v_{d,w} + v_{l,w}, \quad w = 1 \ldots P_k. \tag{11}$$

5) Select the best path: After processing the FF, an SDN controller can calculate and determine whether there are overloaded nodes on P_k selected paths. First, we sort the performance result upon a candidate set as

$$v_1 > v_2 \ldots > v_{P_k}. \tag{12}$$

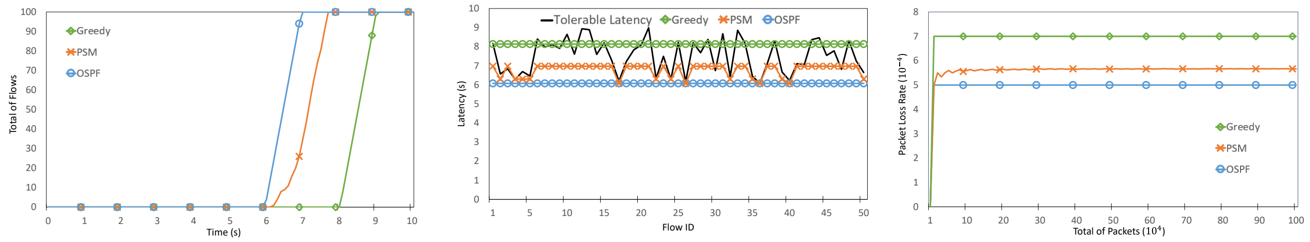
If a path with the maximum weighted v_1 has any overloaded node as its forwarding node, then we select v_2 and so on. If the last path v_{P_k} still has any overload node, we will have the above flow in waiting for the next run of the path selection.

6) Finally, according to v, the SDN controller updates all of nodes which are in the selected path and the first node, i.e., finding an ingress node in link with the edge node.

V. PERFORMANCE RESULTS

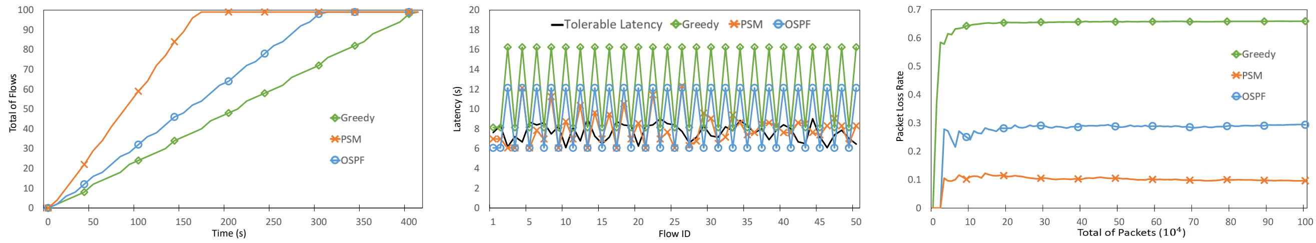
Simulation results in terms of latency, packet loss and the total of received packets by target nodes are derived on the Abilene topology as Fig. 2 where packets are transmitted from node 4 to node 1 while a new flow is created per 10 ms. In simulation settings, the transmission time required is the sum of the ideal shortest path plus a random value of [0,3000] ms, which is used as the delay tolerance of the packet. The packet loss rate is the sum of the packet loss rate of the shortest path plus a random value of [0,3] multiplied by 10^{-4}, which is used as the packet loss rate tolerance of the packet. The packet loss rate of each link is fixed at 10^{-4}. The delay of each link is converted by the length of the path in units of 1 Km/ms. A flow outputs 100 packets per 10 ms. Each packet is 5 KB. In the case of latency setting, 100 flows are transmitted to observe the transmission completion time. In the case of packet loss setting, 10,000 flows are sent to observe the packet loss rate.

Figs. 3(a)-3(c) show that OSPF achieves the best performance in terms of the number of traffic flows, latency, and packet loss rate, respectively, under light traffic workload. That is to say, with the condition of sufficient bandwidth and buffer in each node, OSPF chooses the shortest path to transmit packets and gets the benefits from the shortest path



(a) Total of data flows (b) Latency (c) Packet loss rate

Fig. 3. Results under light traffic workload.



(a) Total of data flows (b) Latency (c) Packet loss rate

Fig. 4. Results under heavy traffic workload.

transmission. In contrast, our PSM only achieves performance between OSPF and Greedy. But, it is worth to note that though PSM still perfectly meets the requirements by considering both network conditions and user requirements.

On the other hand, our PSM exhibits excellent performance under heavy traffic workload. As Figs. 4(a)-4(c) show, PSM achieves the best performance in terms of injecting number of flows, latency, and packet loss rate, respectively. Because both OSPF and Greedy choose fixed paths to transmit packets, under the condition of limited bandwidth and buffer size in each node, network congestion makes the performance become worse. In contrast, PSM can choose transmission path in a flexible manner. Thus, different transmission paths can be chosen to lower the burden of a specific transmission path to achieve the better performance than OSPF and Greedy.

VI. CONCLUSION

This paper develops a QoS routing scheme with a specific fittingness function in an integrated SDN-Edge network. This scheme can appropriately adjust network resource to maintain transmission latency and packet loss rate against various service requirements by user demands. Compared with OSPF and Greedy schemes, our PSM performs better under heavy traffic conditions, while PSM still meets the user requirements under light traffic conditions.

ACKNOWLEDGMENT

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contracts MOST-105-2221-E-008-029-MY3 and MOST-108-2221-E-008-021.

REFERENCES

[1] E. Mirzamany, A. Laseba, and O. Gemikonakli, “Comparison between resource reservation protocol and next steps in signalling in mobile ip networks,” *IET Networks*, vol. 4, no. 2, p. 83–92, March 2015.

[2] Open Networking Foundation (ONF), “Software-defined networking (SDN) definition,” https://www.opennetworking.org/sdn-definition/.

[3] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, “Traffic engineering in sdn/ospf hybrid network,” in *Proc. of IEEE 22nd International Conference on Network Protocols*, Oct. 2014, pp. 563–568.

[4] R. Muñoz, et al., “Integration of iot, transport sdn, and edge/cloud computing for dynamic distribution of IoT analytics and efficient use of network resources,” *Journal of Lightwave Technology*, vol. 36, no. 7, pp. 1420–1428, April 2018.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, October 2016.

[6] W. Yu, et al., “A survey on the edge computing for the internet of things,” *Journal of Parallel and Distributed Computing*, vol. 6, pp. 6900–6919, 2016.

[7] J. McEachen and R. Chesser, “Vulnerabilities in the open shortest path first interior gateway protocol,” in *Proc. of IEEE MILCOM 2000*, vol. 2, Oct. 2000, pp. 1224–1228.

[8] Z. G. Al-Mekhlafi and R. Hassan, “Evaluation study on routing information protocol and dynamic source routing in ad-hoc network,” in *Proc. of the 7th International Conference on Information Technology in Asia*, Jul. 2011, pp. 1–4.

[9] J. M. Llopis, J. Pieczerak, and T. Janaszka, “Minimizing latency of critical traffic through sdn,” in *Proc. of IEEE International Conference on Networking, Architecture and Storage*, Aug. 2016, pp. 1–6.

[10] N. Saha, S. Bera, and S. Misra, “Sway: Traffic-aware qos routing in software-defined iot,” *IEEE Transactions on Emerging Topics in Computing*, p. DOI: 10.1109/TETC.2018.2847296, Jun. 2019.

[11] W. Zhu, X. Liu, X. Zhu, T. Jiang, and X. Zhao, “Joint optimization of heterogeneous cellular and backhaul networks for services awareness in sdn,” in *Proc. of 2018 IEEE/CIC International Conference on Communications in China*, Aug. 2018, pp. 353–357.

[12] P. Bellavista, C. Giannelli, T. Lagkas, and P. Sarigiannidis, “Quality management of surveillance multimedia streams via federated sdn controllers in fiwi-iot integrated deployment environments,” *IEEE Access*, vol. 6, pp. 21 324–21 341, Apr. 2018.