



A Stepwise Path Selection Scheme Based on Multiple QoS Parameters Evaluation in SDN

Lin Liu^{1,2}, Jian-Tao Zhou^{1(✉)}, Hai-Feng Xing³, and Xiao-Yong Guo¹

¹ College of Computer Science, Ecological Big Data Engineering Research Center of the Ministry of Education, Cloud Computing and Service Software Engineering Laboratory of Inner Mongolia Autonomous Region, National and Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian, Social Computing and Data Processing Key Laboratory of Inner Mongolia Autonomous Region, Big Data Analysis Technology Engineering Research Center of Inner Mongolia Autonomous Region, Inner Mongolia University, Hohhot, Inner Mongolia, China

² College of Computer Science and Technology, Inner Mongolia Normal University, Hohhot, Inner Mongolia, China

³ College of Computer Information and Management, Inner Mongolia University of Finance and Economics, Hohhot, Inner Mongolia, China

Abstract. Nowadays, the best-effort service can not guarantee the quality of service (QoS) for all kinds of services. QoS routing is an important method to guarantee QoS requirements. It involves path selection for flows based on the current network status and the performance criteria of the service requirements. However, it is difficult for proposed solutions to obtain all the available paths owing to not fully considering all the QoS parameters of paths. In this paper, we propose SWQoS, a novel, universal, and stepwise QoS guarantee scheme based on multiple QoS parameter evaluation for selecting the available paths including preferred paths, satisfied paths and reluctant paths in SDN. The experiments show that SWQoS can select all the available paths that meet the performance criteria of the service requirements and have better QoS parameter performance compared with other path selection methods.

Keywords: SDN · QoS routing · Path selection · AHP · Fuzzy synthetic evaluation

1 Introduction

Providing high-quality traffic delivery for various services is a hot topic of current QoS routing research. Currently, some researches including Integrated Services (IntServ)/Resource Reservation Protocol (RSVP) [1], Differentiated Services (Diffserv) [2] and Multi-Protocol Label Switching (MPLS) [3] exist management complicated and are challenging to deploy [4,5]. Path selection

is another common way used in QoS routing. Path selection is to find the paths that meet the service requirements based on the current network status and the performance criteria of the service requirements. However, that the current researches are based on the conventional network architecture will become extremely difficult owing to the lack of centralized functions and difficulty obtaining the global network status.

SDN [6–9], as a new type of network architecture, effectively makes up for the deficiencies of path selection research in the conventional network architecture. The centralized control and quickly obtaining the global network view function of SDN network provide path selection with the possibility to implement novel, more powerful strategies [5, 10]. However, the existing SDN path selection schemes do not fully consider multiple QoS parameters to determine the path selection, making it difficult to obtain optimal path and all the available paths. Furthermore, some method need dedicated computational resources and training data, yet. Aim to the above deficiencies, we propose SWQoS - a novel, universal and stepwise QoS routing evaluation scheme in SDN.

SWQoS scheme is pre-computing all the candidate paths between a source and a destination in advance. On this basis, the pros and cons of the candidate paths are evaluated by considering the performance criteria of the service requirements and the current network status. More specifically, SWQoS firstly looks for the paths from all the candidate paths where the network status fully meet the performance criteria of the service requirements, called preferred path set. Secondly, if no preferred path, then look for the paths where the network status basically meet the performance criteria of the service requirements, called satisfied path set. Finally, if none of the above paths exist, that is, there are no paths whose status fully meet the performance criteria of the service requirements, AHP (Analytic Hierarchy Process) [11] combined with the fuzzy synthetic evaluation method is used to look for the paths, namely reluctant path set. The contributions of our research are highlighted as follows:

- All the QoS parameters are fully considered and the network service preference is emphasized to ensure the QoS requirements of network services.
- As a universal path selection scheme, SWQoS scheme can be applied to the transmission of various types of service traffic by adjusting parameters and obtains all the available paths according to stepwise path selection.
- SWQoS scheme can obtain the optimal path.

The rest of the paper is organized as follows. In Sect. 2, we discuss the related work on QoS routing. In Sect. 3, we propose SWQoS scheme, including the scheme architecture, finding the candidate paths, determining the QoS weight values of the service requirements, and selecting the available paths. In Sect. 4, three group simulating experiments are implemented with VoIP as an example, and the results are analyzed. Finally, we conclude this work in Sect. 5.

2 Related Work

In the section we review existing researches including QoS routing in the conventional network architecture, QoS routing in SDN, and QoS routing adopting artificial intelligence in SDN.

The solutions of QoS routing, such as IntServ, Differv, and MPLS, mainly depend on more bandwidth capacity in the conventional network architecture. More bandwidth capacity would generate more cost. IntServ utilizes the resource reservation protocol (RSVP) [12] to guarantee the service requirements. The resources are reserved along the packet forwarding path in the IntServ model. Thereby, it has limited scalability. Differv classifies incoming the flows based on the different service types. It only supports per-hop QoS, which reduces the complexity of implementation, but weakens the guarantees. MPLS is another widely used technology that is explicit routed paths.

However, the above routed paths are static, lacking on-demand reconfigurability, difficultly deployed and management complicated.

SDN provides a new perspective for the solution of QoS routing. OpenQoS [13], a novel OpenFlow controller design for multimedia delivery with QoS guarantee, groups the multimedia and data flows. The multimedia flows are dynamically placed on QoS routing path and data flows remain on the conventional shortest-path. FlowQoS [14, 15] performs application identifications and flow rate shaping according to the service requirements. HiQoS [16], an SDN-based multipath solution to guarantee QoS, uses the queuing mechanisms of OpenFlow protocol to implement the bandwidth guarantees for different service traffic, and the multipath is generated using a modified Dijkstra algorithm with QoS constrained. ARVS [17] proposes an adaptive routing approach for video traffic with guaranteeing QoS requirements in SDN. It finds out a certain path that is not a good path for the service. Slightly different from the above method, Li et al. [18] propose the fuzzy synthetic evaluation mechanism (FSEM) for path load balancing based on SDN. The paths can be dynamically adjusted according to the global view of the network. This method first uses the Top-K paths selection algorithm to obtain k candidate paths based on the least hop count, and then selects the available paths from the k candidate paths using the fuzzy synthetic evaluation method.

The above schemes in SDN have not fully consider all the QoS parameters and the network service preference, and most of them only obtain an available path and the available path may not be the optimal path.

In recent years, artificial intelligence being trying to introduced into routing mechanisms based on SDN is used as a new potential research direction [19], while supervised learning and reinforcement learning is the most widely used. Bomin Mao et al. [20] propose a supervised deep learning system to directly calculate the paths using traffic patterns. Joao Reis et al. [21] propose a routing framework instead of heuristic algorithm to obtain the best routing decision by training deep neural networks. The characteristics of flows and the network status are taken as the inputs of the deep neural network, a value is an input for each link, and it is converted into a path at last. DROM [22], a deep rein-

forcement learning mechanism for SDN, achieves a universal and customizable routing optimization. DROM simplifies the network operation and maintenance, thus improving the network performance, such as delay and throughput.

However, artificial intelligence be applied in routing mechanisms need the dedicated GPU devices, a lot of computational resources to train the model, and training is expensive.

3 Proposed Scheme: SWQoS

In this section, we design a path selection scheme adopting a stepwise strategy based on the multiple QoS parameter evaluation in SDN architecture. Next, we will introduce the architecture that we designed, and then elaborate all modules of the SWQoS scheme.

3.1 SWQoS Scheme Architecture

The reference architecture is shown in Fig. 1. SWQoS scheme architecture is built based on SDN architecture. It consists of data, control and application plane. SWQoS lies in the application plane and mainly includes three differentiated modules: path finding, QoS requirements of services, and path selection. The relationship between three modules is shown in Fig. 1, too.

- **Path Finding:** responsible for finding out all the candidate paths between a source and a destination according to the network topology and the hop count.

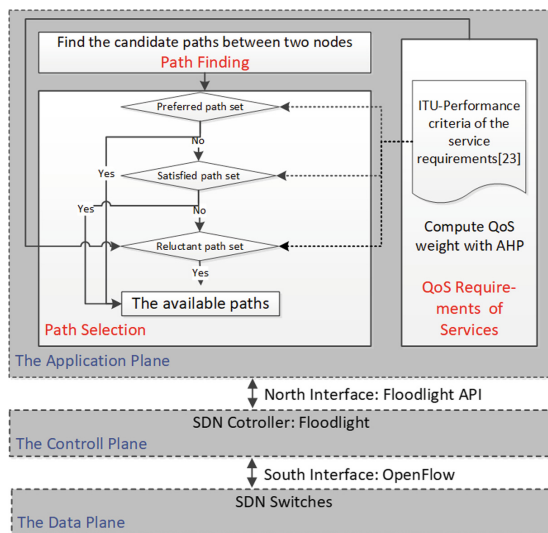


Fig. 1. SWQoS scheme architecture

- **QoS Requirements of Services:** responsible for quantifying the QoS requirement of services by AHP, namely computing the QoS weight values of the service requirements.
- **Path Selection:** responsible for selecting the available path set for the forwarded traffic. The available path set is divided into three classes: preferred path set, satisfied path set and reluctant path set. The preferred path set and satisfied path set are selected according to ITU-performance criteria of the service requirements standards [23]. When the preferred path set and the satisfactory path set do not exist, the reluctant path set is looked up by using the fuzzy synthetic evaluation method combined with the QoS weight values calculated by the QoS requirements of the service.

Path selection as an application is located on SDN application plane. The design does not impose additional functions in the control and data plane of SDN architecture, but makes full use of link discovery, network status statistics, and centralized control function obtained by SDN architecture.

3.2 Path Finding

In order to describe our algorithm clearly, several notions are introduced follows as. Consider an arbitrary network topology \mathcal{G} is composed of the set of nodes \mathcal{V} , the set of links \mathcal{E} . The nodes indicate all the switches in the network topology, $\mathcal{V} = \{s_1, s_2, \dots, s_n\}$, and n is the total number of nodes. \mathcal{E} indicates all the links in the network, $\mathcal{E} = \{e_1, e_2, \dots, e_m\}$, and m is the total number of links. In the network topology \mathcal{G} , node number starts from 1 to n , and each of the links is represented by (s_i, s_j) . We define a set that all the candidate paths are between a source and a destination $\mathcal{P} = \{p_1, p_2, \dots, p_i, \dots, p_k | 1 \leq i \leq k\}$. k is the total number of the candidate paths.

Two algorithms are designed to capture all the candidate paths. Algorithm 1 is the main frame. Line 2 deals with the situation that the source node is the same as the destination node. That is, node will send packets to itself. Line 4 gets the list of the next hop nodes adjacent to the source node from the adjacency table saving the network topology. Line 5 calls Algorithm 2 to capture all the candidate paths \mathcal{P} recursively.

In Algorithm 2, if the next hop is the destination node (Line 3), then a complete path from the source to the destination is captured; if the next hop is not contained in the incomplete path (Line 5), then the capture process continues by the way of recursion; otherwise, the new next hop is checked.

Algorithm 1: Path Finding

Input: sn , dn , adj_list // sn is source, dn is destination, and adj_list is the network topology.

Output: \mathcal{P} // \mathcal{P} is the set of all the candidate paths.

```

1 create_path(path, sn);
2 if sn = dn then
3   | P = P ∪ path;
4 else
5   | next_hop_list = adj_list[sn];
6   | get_all_paths(path, next_hop_list, P, dn)
7 end

```

Algorithm 2: get_all_paths

Input: $path$, $next_hop_list$, \mathcal{P} , dn

Output: \mathcal{P}

```

1 for each next_hop in next_hop_list do
2   | new_path = copy_path(path);
3   | if next_path = dn then
4     | add_tail(new_path, next_hop); P = P ∪ path;
5   | else if next_hop not in path then
6     | add_tail(new_path, next_hop);
7     | new_next_hop_list = adj_list[next_hop];
8     | get_all_paths(new_path, new_next_hop_list, P, dn);
9   | else
10  |   free_path(new_path);
11  |   end
12 end

```

Path finding module finds out all the paths between two nodes in the network topology, and takes them as the candidate paths, which is the basis for the next path selection.

3.3 QoS Requirements of Services

That different services on the Internet generate flows have the different QoS requirements, which are also called sensitive preferences for services. However, the QoS requirement of services given in Table 1 are qualitative, such as VoIP, video conferencing and online games, etc. [24]. They can not express the QoS requirement of services with accurate QoS weight value. In this subsection, we adopt AHP method to quantify the QoS requirement of services, and obtain accurate QoS weight value of services.

This paper takes VoIP as an example. Using AHP method to determine the QoS weight values of the VoIP service are commonly divided into the following three major steps:

Table 1. The QoS requirement of services [24]

QoS parameters	<i>VoIP</i>	<i>Video conferencing</i>	<i>Online gaming</i>
Bandwidth	Low	Mid	Low
Delay	High	High	High
Loss	Low	Low	Mid
Jitter	High	High	Low

Step 1: Construct a pairwise comparison matrix. In this step, using AHP will be asked to rate the relative importance of QoS weight values instead of the QoS qualitative values be directly used, so as to obtain accurate QoS weight values. We need to construct a pairwise comparison matrix of QoS parameters. We should provide a scale of numbers for the value judgments to construct this comparison at the beginning. That the scales are to judge the relative importance between the QoS parameters of services are shown in the Table 2. Therefore, a pairwise comparison matrix \mathcal{PM} can be obtained as follows:

$$\mathcal{PM} = \begin{matrix} \textit{Bandwidth} \\ \textit{Delay} \\ \textit{Loss} \\ \textit{Jitter} \end{matrix} \begin{pmatrix} \textit{Bandwidth} & \textit{Delay} & \textit{Loss} & \textit{Jitter} \\ 1 & 1/5 & 1 & 1/5 \\ 5 & 1 & 5 & 1 \\ 1 & 1/5 & 1 & 1/5 \\ 5 & 1 & 5 & 1 \end{pmatrix}$$

Each element of the matrix \mathcal{PM} represents the importance intensity of between the QoS parameters of VoIP service. The element value is supposed to be an approximation of the relative importance. For example, entering 5 in the (Delay, Bandwidth) position meaning that delay is strongly more important than bandwidth for VoIP.

Table 2. Lineal scale of preferences in the pair-wise comparison process [25]

Intensity of importance	<i>Definition</i>
1	Equal importance
3	Moderate importance
5	Strong importance
7	Very strong or demonstrated importance
9	Extreme importance
2, 4, 6, 8	The importance between 1, 3, 5, 7 and 9 scales

Step 2: Calculate the relative weights. We use the arithmetic mean method to calculate the QoS weight values of the service requirements. Firstly, the pairwise comparison matrix of QoS parameters are normalized by the column. Secondly, to sum the values by row to get the vector. Finally, to

normalize the vector obtained by the second step to get the QoS weight values of VoIP service. The symbol \mathcal{W} denotes the QoS weight value vector of the service requirements. After calculation, the QoS weight values vector of VoIP service is $\mathcal{W} = (0.08333333, 0.41666667, 0.08333333, 0.41666667)$.

Step 3: Check the consistency. This step is very important for the rationality of the judgment. A method is to calculate the Consistency Ratio(CR), namely $CR = CI/RI$. RI is the average random index, which is computed and tabulated as shown in Table 3. The Consistency Index (CI) is calculated as $CI = (\lambda_{max} - r)/(r - 1)$. r is the order of the pairwise comparison matrix \mathcal{PM} . λ_{max} represents the maximum eigenvalue of the matrix. If a value of CR is less than 0.1, the numerical judgments will be considered to be acceptable [25]. After the matrix \mathcal{PM} is calculated, $CR = -0.000000$, it meets the consistency requirements.

Table 3. Average random index values according to matrix size [25]

r	1	2	3	4	5	6	7	8	9	10	11
RI	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.52

QoS Requirements of Services can accurately calculate the QoS weight values of the service, which is the foundation for the application of fuzzy synthetic evaluation method in the reluctant paths.

3.4 Path Selection

In this subsection, we present a stepwise scheme to select the available path set for services among the candidate paths. To determine whether the path is available for services, we need to construct a matrix \mathcal{A} that presents the QoS parameter values of the candidate paths.

QoS parameters of links include delay, jitter, bandwidth, and packet loss rate, but they have different metric character. Bandwidth is a concave metric character, delay and jitter are additive metric characters, and the packet loss rate is a multiplicative metric character. Therefore, we represent bandwidth, delay, jitter, and packet loss rate of the path p_i with symbols bw_i , de_i , jt_i and lo_i , respectively. $(bw_{e_1}, bw_{e_2}, \dots)$, $(de_{e_1}, de_{e_2}, \dots)$, $(jt_{e_1}, jt_{e_2}, \dots)$ and $(lo_{e_1}, lo_{e_2}, \dots)$ represent the bandwidth, delay, jitter and packet loss rate on each link along the path, respectively. bw_i is minimum bandwidth constraints over the link of the i th path shown as Eq. (1). de_i and jt_i are the sum of each link delay and jitter over i th path shown as Eq. (2) and (3). lo_i is the multiplication of the loss of each link over i th path shown as Eq. (4).

$$bw_i = \min(bw_{e_1}, bw_{e_2}, \dots) \quad e_1, e_2, \dots \in p_i \tag{1}$$

$$de_i = \sum (de_{e_1}, de_{e_2}, \dots) \quad e_1, e_2, \dots \in p_i \tag{2}$$

$$jt_i = \sum(jt_{e_1}, jt_{e_2}, \dots) \quad e_1, e_2, \dots \in p_i \tag{3}$$

$$lo_i = \prod(lo_{e_1}, lo_{e_2}, \dots) \quad e_1, e_2, \dots \in p_i \tag{4}$$

The matrix \mathcal{A} is composed of four rows that represent four QoS parameters, and the k columns represent k candidate paths. Each element value in matrix \mathcal{A} is derived from Eqs. (1), (2), (3) and (4). The matrix \mathcal{A} is shown as follows.

$$\mathcal{A} = \begin{matrix} \text{Bandwidth} \\ \text{Delay} \\ \text{Jitter} \\ \text{Loss} \end{matrix} \begin{pmatrix} p_1 & p_2 & \dots & p_i & \dots & p_k \\ bw_1 & bw_2 & \dots & bw_i & \dots & bw_k \\ de_1 & de_2 & \dots & de_i & \dots & de_k \\ jt_1 & jt_2 & \dots & jt_i & \dots & jt_k \\ lo_1 & lo_2 & \dots & lo_i & \dots & lo_k \end{pmatrix}$$

We divided into three steps to find out preferred path set, satisfied path set and reluctant path set in order. The International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) G.1010 [23] provides an indication of suitable performance targets, as shown in Table 4.

Table 4. Performance criteria for VoIP services [23]

VoIP	Bandwidth (kbit/s)	Delay (ms)	Jitter (ms)	Loss (%)
Preferred	>64	<150	<1	<3
Limit	>4 & ≤64	≥150 & <400	<1	<3
Not proper	≤4	≥400	≥1	≥3

It is shown that the performance indicators required for VoIP traffic in Table 4. The performance is divided into Preferred, Limit, and Not proper. They correspond to the above mentioned preferred path set, satisfied path set, and reluctant path set, respectively. Next, we describe the details of selecting three path sets.

1) To select preferred path set: We find out preferred path set from the candidate paths, as shown in Eq. (5).

$$\begin{aligned} \text{preferred path set} = \{ & p_i(bw_i, de_i, jt_i, lo_i) \mid \max(bw_i) > 64 \\ & \wedge \min(de_i) < 150 \wedge \min(jt_i) < 1 \wedge \min(lo_i) < 3 \\ & \wedge (\max(bw_i), \min(de_i), \min(jt_i), \min(lo_i)) \in p_i \} \end{aligned} \tag{5}$$

If each of the QoS parameter values of the path meets the first line in Table 4, and all QoS parameter values are the preferred values of the current network status, then they belong to the preferred path set.

2) To select satisfied path set: If each of QoS parameter values of the paths meets the second line in Table 4. Namely, all QoS parameter values are the limit values of the current network status, then they belong to satisfied path set, as shown in Eq. (6).

$$\begin{aligned}
\text{satisfied path set} = \{ & p_i(bw_i, de_i, jt_i, lo_i) \mid \max(bw_i) > 4 \\
& \wedge \min(de_i) < 400 \wedge (\max(bw_i) < 64 \vee \min(de_i) \geq 150) \\
& \wedge \min(jt_i) < 1 \wedge \min(lo_i) < 3 \\
& \wedge (\max(bw_i), \min(de_i), \min(jt_i), \min(lo_i)) \in p_i \}
\end{aligned} \tag{6}$$

3) To select reluctant path set: If no path meets the above two cases, namely the paths that fully meet the service requirements do not exist, the fuzzy synthetic evaluation method is used to select the reluctant paths. The reluctant path refers to that some QoS parameters of the path meet the QoS requirements of the service, or the path is the best QoS parameter value in the current path set, but still can not fully meet the QoS requirements of the service. For example, the delay and jitter of a path meet Preferred, but the bandwidth and packet loss rate belong to Not proper. If the service is sensitive to delay and jitter, the path can meet the QoS requirements.

The fuzzy synthetic evaluation method is based on fuzzy set theory developed by Zadeh [26] for capturing the uncertainties inherent in a system. The whole process is further divided into three steps.

Step 1: Determining the membership functions

The membership functions are established to calculate the membership degrees of each QoS parameter. A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value between 0 and 1. The symbol \mathcal{F} represents the membership function of the QoS parameters of services. The symbols \mathcal{B}_{bw_i} , \mathcal{B}_{de_i} , \mathcal{B}_{jt_i} and \mathcal{B}_{lo_i} denote the membership degrees of the service about bandwidth, delay, jitter and packet loss rate, and they are affiliated with Preferred, Limit, or Not proper, respectively. Equations (7), (8), (9), and (10) are shown as:

$$\mathcal{B}_{bw_i} = \mathcal{F}(bw_1, bw_2, \dots, bw_k) \tag{7}$$

$$\mathcal{B}_{de_i} = \mathcal{F}(de_1, de_2, \dots, de_k) \tag{8}$$

$$\mathcal{B}_{jt_i} = \mathcal{F}(jt_1, jt_2, \dots, jt_k) \tag{9}$$

$$\mathcal{B}_{lo_i} = \mathcal{F}(lo_1, lo_2, \dots, lo_k) \tag{10}$$

Step 2: Calculating the fuzzy synthetic evaluation matrix \mathcal{B}

The matrix \mathcal{B} is obtained from \mathcal{A} through Eq. (7), (8), (9), and (10) transformation. Each of the elements in matrix \mathcal{B} represents the membership degrees of QoS parameters of the paths.

$$\mathcal{B} = \begin{matrix} \text{Bandwidth} \\ \text{Delay} \\ \text{Jitter} \\ \text{Loss} \end{matrix} \begin{pmatrix} p_1 & p_2 & \cdots & p_i & \cdots & p_k \\ \mathcal{B}_{bw_1} & \mathcal{B}_{bw_2} & \cdots & \mathcal{B}_{bw_i} & \cdots & \mathcal{B}_{bw_k} \\ \mathcal{B}_{de_1} & \mathcal{B}_{de_2} & \cdots & \mathcal{B}_{de_i} & \cdots & \mathcal{B}_{de_k} \\ \mathcal{B}_{jt_1} & \mathcal{B}_{jt_2} & \cdots & \mathcal{B}_{jt_i} & \cdots & \mathcal{B}_{jt_k} \\ \mathcal{B}_{lo_1} & \mathcal{B}_{lo_2} & \cdots & \mathcal{B}_{lo_i} & \cdots & \mathcal{B}_{lo_k} \end{pmatrix}$$

Step 3: Obtaining the available paths

The evaluation results can be obtained by taking into account the QoS weight values of the service requirements, such that a vector \mathcal{R} can be calculated by:

$$\mathcal{R} = \mathcal{W} \bullet \mathcal{B} \tag{11}$$

Where \bullet is a multiplication operator, the evaluation results can be obtained by multiplying the QoS weight values of the service requirements \mathcal{W} and the fuzzy synthetic evaluation matrix \mathcal{B} . The conclusion of the fuzzy synthetic evaluation can be obtained by the maximum value principle. The key algorithm for computing \mathcal{R} is shown in Algorithm 3. Lines 1 to 5 are to calculate \mathcal{B} , and Line 6 gets \mathcal{R} .

Algorithm 3: Computing \mathcal{R}

Input: $\mathcal{A}[r][c]$, \mathcal{W} // \mathcal{A} is QoS parameter values of \mathcal{P} , \mathcal{W} is the weight vector of QoS parameters

Output: \mathcal{R} // the decision value vector of \mathcal{P}

```

1 for  $i \in [1, r]$  do
2   | for  $j \in [1, c]$  do
3   |   |  $\mathcal{B}[i][j] = \mathcal{F}(\mathcal{A}[i][j])$ ;
4   |   end
5 end
6  $\mathcal{R} = \mathcal{W} \cdot \mathcal{B}$ ;

```

Path Selection module orderly selects preferred path set, satisfied path set and reluctant path set. Even if the candidate paths do not fully meet the QoS requirements of services, a path approximating the QoS requirements of services will be recommended.

4 Experiments and Performance Evaluation

In this section, we conduct three experiments to evaluate the performance of SWQoS in all aspects of QoS parameters and compared with Widest-shortest path (WSP), Shortest-widest path (SWP), shortest-distance path (SDP), and FSEM [18] based on different link status.

4.1 The Experimental Environment and Topology

We have implemented the experiments with Mininet [27,28] on the virtual machine of personal computer. The Floodlight [29] is used as the SDN controller and iperf [30] is used to simulate VoIP flows. The testing results are obtained by iperf and ping tools.

All simulation experiments are performed on the asymmetric network topology including twelve OpenFlow switches and six hosts as shown in Fig. 2. There are fourteen paths from s1 to s12, which are obtained by Algorithms 1 and 2. Fourteen paths are shown in Table 5. The 3rd, 7th, and 14th paths are the shortest paths based on the hop count, which can be obtained by the Dijkstra algorithm usually.

We ran three groups of experiments to verify the effectiveness of SWQoS in the same topology mentioned above. Link status values assumed by three group

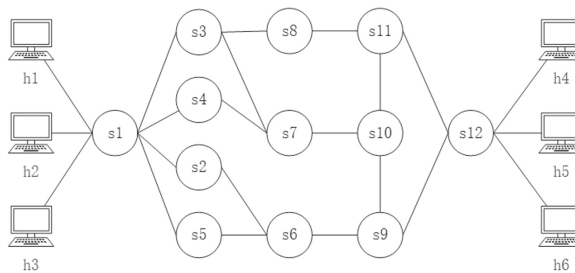


Fig. 2. The network topography

Table 5. All candidate paths from s1 to s12

Path numbers	Paths
1	s1, s2, s6, s9, s10, s7, s3, s8, s11, s12
2	s1, s2, s6, s9, s10, s11, s12
3	s1, s2, s6, s9, s12
4	s1, s3, s7, s10, s9, s12
5	s1, s3, s7, s10, s11, s12
6	s1, s3, s8, s11, s10, s9, s12
7	s1, s3, s8, s11, s12
8	s1, s4, s7, s3, s8, s11, s10, s9, s12
9	s1, s4, s7, s3, s8, s11, s12
10	s1, s4, s7, s10, s9, s12
11	s1, s4, s7, s10, s11, s12
12	s1, s5, s6, s9, s10, s7, s3, s8, s11, s12
13	s1, s5, s6, s9, s10, s11, s12
14	s1, s5, s6, s9, s12

Table 6. Hypothetical link status of three group experiments

LINKS	The first group experiment				The second group experiment				The third group experiment			
	Bandwidth (kb/s)	Delay (ms)	Loss (%)	Jitter (ms)	Bandwidth (kb/s)	Delay (ms)	Loss (%)	Jitter (ms)	Bandwidth (kb/s)	Delay (ms)	Loss (%)	Jitter (ms)
s1, s3	80	3	0.1	0.09	60	37	7	9	65	37	3	0.9
s1, s4	68	9	1	0.3	60	9	0.1	0.3	68	9	1	0.63
s1, s2	70	1	0.6	0.1	63	25	5	1	50	0	1	0.001
s1, s5	80	30	2	1	63	12	0.2	0.38	30	30	2	0.01
s3, s8	52	31	3	0.13	52	31	3	0.13	32	31	3	0.99
s3, s7	80	1	3	0.06	63	1	0.4	0.06	50	32	3	0.632
s4, s7	70	9	1	0.03	63	9	0.1	0.03	48	9	1	0.476
s2, s6	80	5	1	0.4	63	5	1	0.4	80	5	1	0.001
s5, s6	90	3	2	0.3	63	3	0.2	0.3	90	3	2	0.003
s8, s11	60	35	3	0.93	60	35	3	0.93	60	35	3	0.935
s7, s10	90	1	1	0.06	60	1	0.1	0.06	49	10	1	0.594
s6, s9	80	0	1	0.01	60	5	0.1	0.01	80	0	0.1	0.687
s10, s11	80	1	1	1	60	1	1	1	41	11	1	0.1
s10, s9	95	1	2	0.3	60	1	0.2	0.3	50	50	2	0.001
s11, s12	70	40	8	0.3	60	40	8	0.3	47	40	0.8	0.1
s9, s12	70	0	1	0.1	60	0	0.1	0.1	70	1	1	3

experiments can simulate the scenario of selecting different path sets shown as Table 6. (*i*) Hypothetical link status of the first group experiment aims to verify performances of the preferred path set; (*ii*) hypothetical link status of the second group experiment aims to verify performances of the satisfied path set; (*iii*) hypothetical link status of the third group experiment aims to verify performances of the reluctant path set. In the experiments, VoIP simulation traffic of G.711 encoding is first sent along with every path selected respectively from host 1 to host 4 shown as Fig. 2. Then, we capture the throughput, delay, jitter and packet loss rate of the paths selected under the same VoIP traffic load. Last, we implement and analyze three group experiments respectively.

4.2 The First Group Experiment: Simulating the Network Status of Selecting the Preferred Paths

In the first group experiment, we simulate the network status scenario that can obtain the preferred path set. After calculation by Eqs. (1), (2), (3) and (4), the path status is obtained as shown in Table 7.

Table 7. QoS parameters of paths in the first group experiment

Paths	Bandwidth (kb/s)	Delay (ms)	Loss (%)	Jitter (ms)
1st	52	115	17.77	2.29
2nd	70	48	13.04	2.11
3rd	70	6	2.68	0.61
4th	70	6	2.68	0.61
5th	70	46	9.47	1.51
6th	52	71	8.9	2.55
7th	52	109	13.52	1.45
8th	52	87	10.98	2.85
9th	52	125	15.5	1.75
10th	68	20	4.14	0.79
11th	68	60	10.82	1.69
12th	52	142	19.75	3.09
13th	70	75	15.13	2.91
14th	70	33	5.02	1.41

It can be concluded from Table 7 that the status values of the 3rd and 4th paths meet the condition of Eq. (5), so they are the preferred paths. The paths calculated by other comparison algorithms are the 3rd, the 4th and the 14th paths. WSP and SDP select the 14th path. SWP selects the 4th path. FSEM selects the 3rd path. We respectively inject the simulated flow into three paths and monitors the status of the paths every 5s. The experimental results are shown in Fig. 3 and Fig. 4. Figure 3 shows the results for throughput, delay, jitter and packet loss rate of the 3rd, the 4th and the 14th paths under the same VoIP traffic load. Figure 4 shows the results for the average status values of three paths under the same load.

We observe that the QoS parameter performance of the 3rd path and the 4th path are better than the 14th path, thereby indicating that SWQoS, SWP and FSEM are more efficient compared with WSP and SDP. However, SWP and FSEM schemes can only select the 3rd path or the 4th path, they can not select all paths that meet the requirements of the network services. Therefore, SWQoS scheme is more better compared with the others because it can not only select the best paths but also select all the paths that meet the network service requirements.

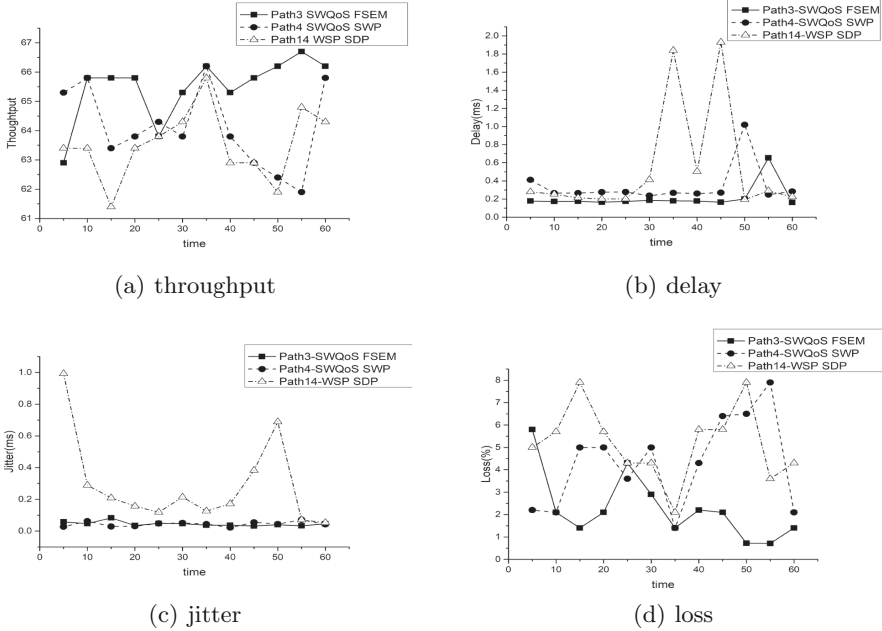


Fig. 3. Comparison of QoS parameters in the first group experiment

4.3 The Second Group Experiment: Simulating the Network Status of Obtaining Satisfied Paths

Similar to the first group experiment above, the purpose of the second group experiment is to select a satisfied path set and compare with the QoS parameter performance of the paths selected by the others schemes. It can be concluded from Table 8 that the status values of the 10th and the 14th paths meet the condition of Eq. (6), thus they are the satisfied paths.

Same as in the first group experiment, the 10th path and the 14th path obtained by SWQoS are better than the 3rd path in QoS parameter performance as shown in Fig. 5 and Fig. 6. Similarly, SWP and FSEM schemes can also obtain the 10th path or the 14th path, both the 10th path and the 14th path cannot be obtained at the same time by two schemes. However, SWQoS can select both the optimal paths of performance and all the paths that meet the requirements of services. Therefore, SWQoS is more effective.

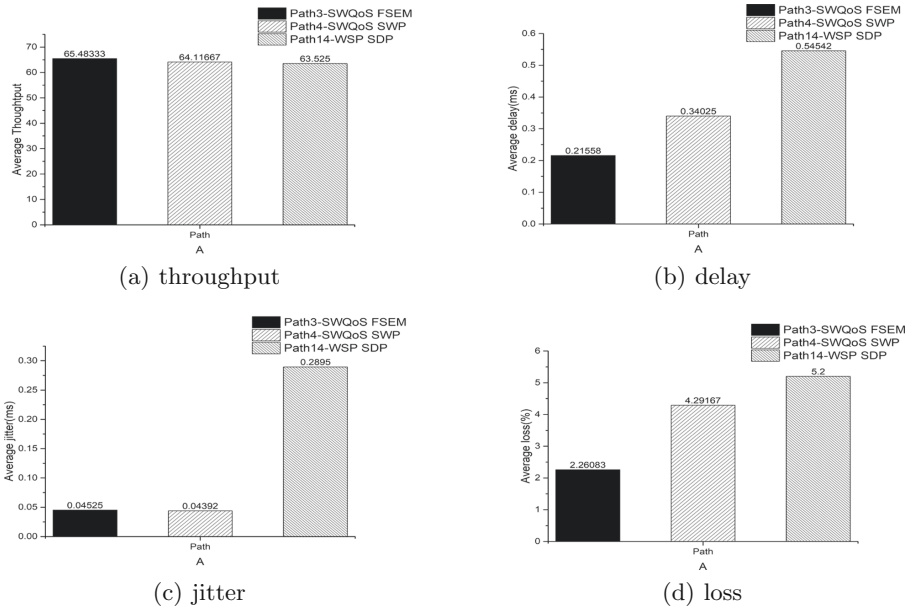


Fig. 4. The average values of QoS parameters in the first experiment

Table 8. QoS parameters of paths in the second group experiment

Paths	Bandwidth (kb/s)	Delay (ms)	Loss (%)	Jitter (ms)
1st	52	144	19.24	3.19
2nd	60	77	14.6	3.01
3rd	60	35	6.14	1.51
4th	60	40	7.74	9.52
5th	60	80	15.72	10.42
6th	52	105	13.63	11.46
7th	52	143	19.5	10.36
8th	52	87	7.69	2.85
9th	52	125	13.96	1.75
10th	60	20	0.6	0.79
11th	60	60	9.19	1.69
12th	52	129	14.47	2.47
13th	60	62	9.56	2.29
14th	60	20	0.6	0.79

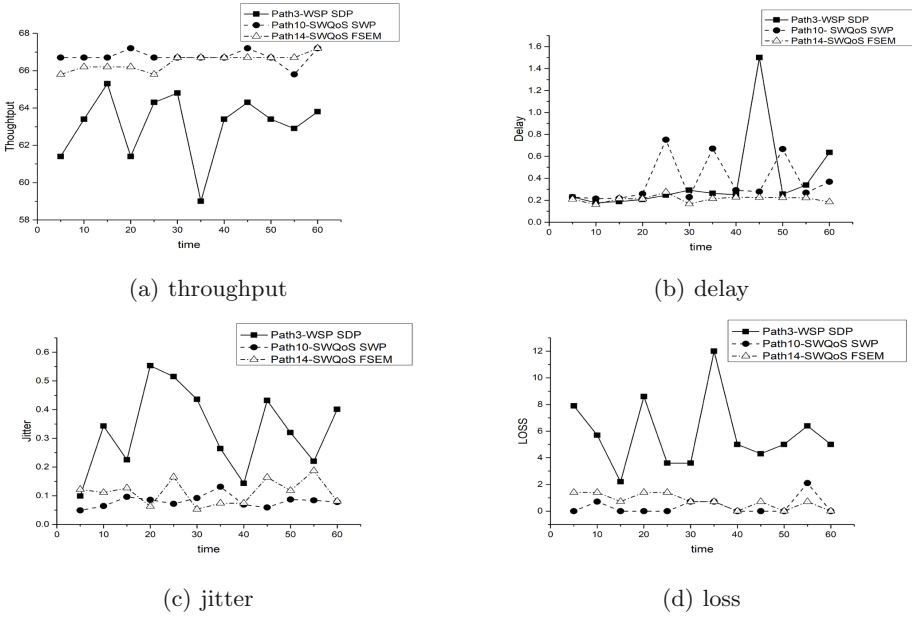


Fig. 5. Comparison of QoS parameters in the second group experiment

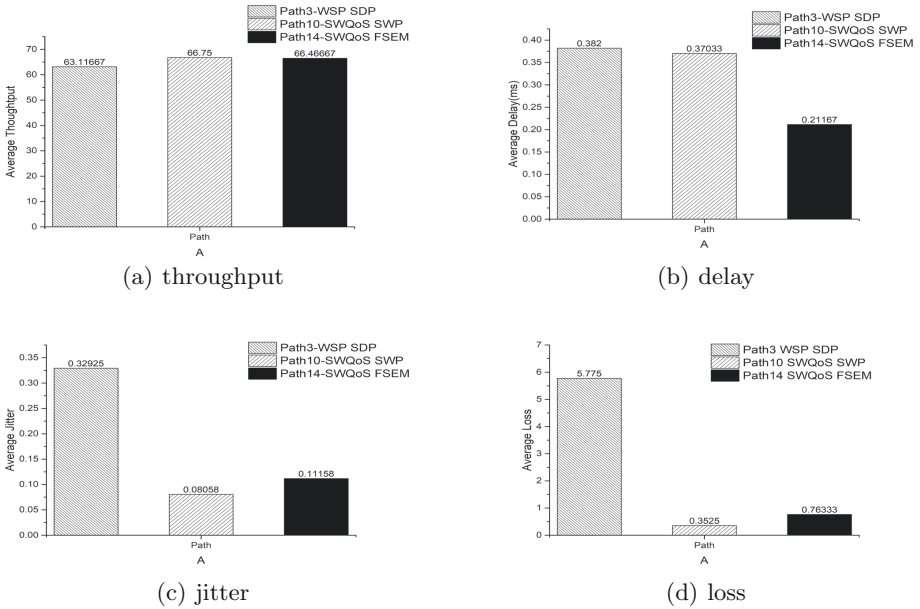


Fig. 6. The average values of QoS parameters in the second experiment

4.4 The Third Group Experiment: Simulating the Network Status of Obtaining Reluctant Paths

In the third group experiment, we first obtain the status of the paths, as shown in Table 9. However, none of the path parameters in the Table 9 match Eqs. (5) or (6). Thus, we implement the step to select the reluctant paths.

Table 9. QoS parameters of paths in the third group experiment

Paths	Bandwidth (kb/s)	Delay (ms)	Loss (%)	Jitter (ms)
1st	32	203	4.32	3.941
2nd	41	106	0.16	0.89
3rd	50	6	0.1	3.689
4th	49	130	18	5.127
5th	41	130	7.2	2.326
6th	32	165	54	5.926
7th	32	143	21.6	2.925
8th	32	178	54	6.764
9th	32	156	21.6	3.763
10th	48	79	2	4.701
11th	41	79	0.8	1.9
12th	30	231	17.28	3.952
13th	30	134	0.64	0.901
14th	30	34	0.4	3.7

1) Settings of Membership Functions. Several membership functions need to be proposed, such as bandwidth, delay, jitter, and packet loss rate. The symbols $bw_{threshold}$, $de_{threshold}$, $jt_{threshold}$ and $lo_{threshold}$ represent the thresholds of the bandwidth, delay, jitter and packet loss rate, respectively. According to the habit of selecting membership functions, three commonly used membership functions are selected, which conform to the characteristics of VoIP protocol. They include the Cauchy function, piecewise function, and normal distribution function. After repeating experiments, the membership functions adopt improved normal distribution functions. The fuzzy membership functions could be defined separately as below:

$$\mathcal{F}_{bw}(x) = \begin{cases} e^{-\left(\frac{x - bw_{threshold}}{e}\right)^2} & x < bw_{threshold} \\ 1 & \text{otherwise} \end{cases} \quad (12)$$

$$\mathcal{F}_{de}(x) = \begin{cases} e^{-\left(\frac{x - de_{threshold}}{e}\right)^2} & x \geq de_{threshold} \\ 1 & \text{otherwise} \end{cases} \quad (13)$$

$$\mathcal{F}_{jt}(x) = \begin{cases} 1 & x < jt_{threshold} \\ e^{-e*(x - jt_{threshold})} & \text{otherwise} \end{cases} \quad (14)$$

$$\mathcal{F}_{lo}(x) = \begin{cases} 1 & x < lo_{threshold} \\ e^{-e^{*(x-lo_{threshold})}} & \text{otherwise} \end{cases} \quad (15)$$

Adjusting the thresholds of the membership functions are very important to experimental results. The range of threshold values adopts a principle, which is as small as possible. The range of threshold values should be less than the range given in Table 4. For instance, $bw_{threshold}$ value is equal to the preferred bandwidth value in Table 4 plus half of the absolute value of the difference between the preferred bandwidth value and the current maximum bandwidth value in Table 4. In this way, the selected paths can meet the service requirements to the maximum extent. Other thresholds are also determined by the same method. However, the monotonicity of the membership functions needs attention.

2) Result. According to the membership functions, we can compute the fuzzy synthetic evaluation matrix \mathcal{B} . Then, the QoS weight values of the service requirements \mathcal{W} are multiplied by the fuzzy synthetic evaluation matrix \mathcal{B} to get the reluctant path according to Eq. (10). $\mathcal{R} = \mathcal{W} \bullet \mathcal{B} = \{0.00012136, 0.41671141, 0.48580011, 0.00000478, 0.00965493, 0.00000054, 0.00194193, 0.00000006, 0.00019796, 0.36397667, 0.39579707, 0.00011811, 0.41666693, 0.41724337\}$. The max value of \mathcal{R} is the 3rd path. Therefore, the reluctant path is the 3rd path in the candidate paths by implementing the SWQoS scheme.

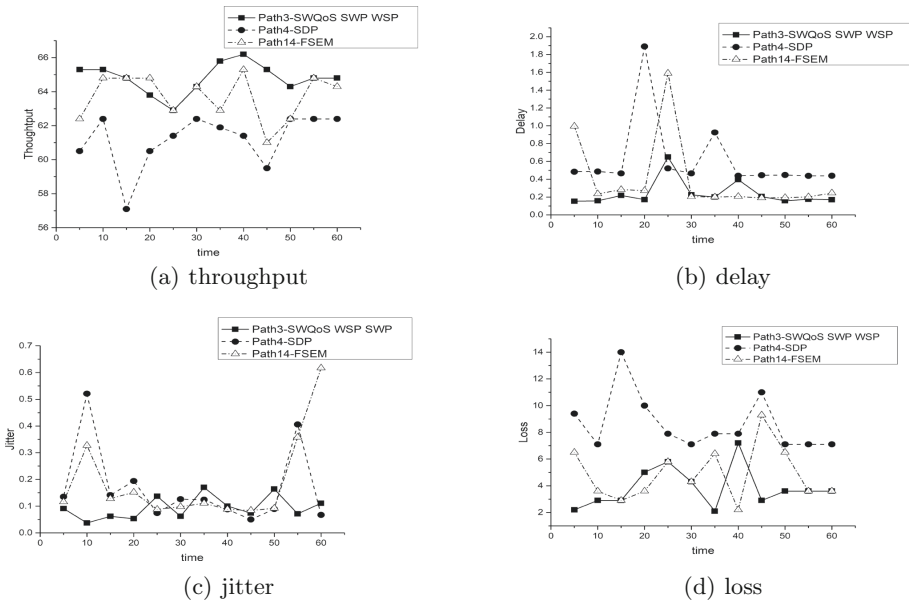


Fig. 7. Comparison of QoS parameters in the third experiment

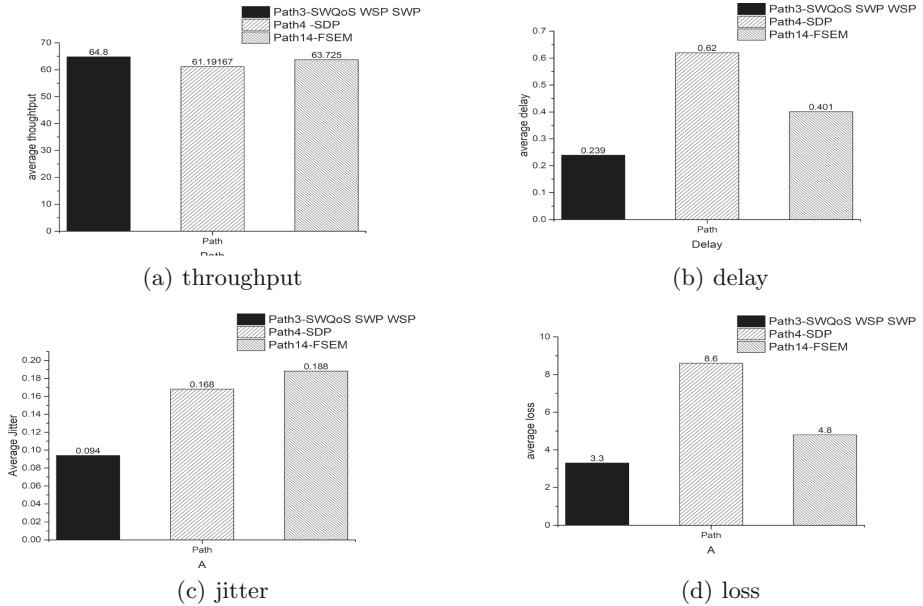


Fig. 8. The average values of QoS parameters in the third experiment

Figure 7 and Fig. 8 show the simulation results for network status under VoIP traffic load in the third group experiment. It can be seen that the QoS parameter performance of the 3rd path is better compare with the 4th path and the 14th path most of the time. Although the 3rd path has lower QoS parameter performance than other paths some time, they do not affect the whole QoS guarantee. That's because VoIP protocol is more sensitive to delays and jitters, and the effect of two parameters is more obvious. The results of the average value of QoS parameters shown in Fig. 8 more obviously prove the above situation. Thus the 3rd path can guarantee the QoS requirements of VoIP as much as possible.

Through the above three group experimental results, we can observe that SWQoS can select all the available paths and the performance of the paths are better than other paths compared with other routing methods.

5 Conclusions

In this paper, we propose an stepwise, universal SWQoS scheme to select the available paths from all the candidate paths in SDN. To obtain all the available paths, we fully consider all the performance criteria of the service requirements and current network status. In the absence of preferred path set and satisfied path set, we select the reluctant path set using AHP combined with the fuzzy synthetic evaluation method. Thus, SWQoS can meet various QoS requirements of services and obtain the optimal path. Experimental results show that the QoS

parameter performance of paths selected by the SWQoS scheme are best and can obtain more available paths compare with other methods.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (61662054), the Major Project of Inner Mongolia Natural Science Foundation (2019ZD15), Research and Application of Key Technology of Big Data for Discipline Inspection and Supervision (No. 2019GG372), Inner Mongolia Colleges and Universities of Young Technology Talent Support Program under Grant No. NJYT-19-A02, and Inner Mongolia application research and development project (201702168). It was also sponsored by the Ecological Big Data Engineering Research Center of the Ministry of Education, Cloud Computing and Service Software Engineering Laboratory of Inner Mongolia Autonomous Region, National Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian, Social Computing and Data Processing Key Laboratory of Inner Mongolia Autonomous Region, Big Data Analysis Technology Engineering Research Center of Inner Mongolia Autonomous Region, Natural Science Foundation of Inner Mongolia under Grand No. 2020MS06030 and Digital Engineering Demonstration of Mongolian Music Resources and Key Technology Research, Key Technology Research Project in Inner Mongolia Autonomous Region No. 2019GG147, Ministry of Education Industry-University Cooperation Collaborative Education Project No. 201902035056, 202002215071, 202002142055.

References

1. Braden, R., Clark, D., Shenker, S.: RFC 1633: integrated services in the internet architecture: an overview (1994)
2. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: RFC2475: an architecture for differentiated service (1998)
3. Rosen, E., Viswanathan, A., Callon, R.: RFC3031: multiprotocol label switching architecture (2001)
4. Egilmez, H.E., Civanlar, S., Tekalp, A.M.: An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks. *IEEE Trans. Multimedia* **15**(3), 710–715 (2012)
5. Karakus, M., Durrresi, A.: Quality of service (QoS) in software defined networking (SDN): a survey. *J. Netw. Comput. Appl.* **80**, 200–218 (2017)
6. Alto, P.: Software-defined networking: the new norm for networks [white paper]. ONF White Paper (2012)
7. Kreutz, D., Ramos, F.M., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. *Proc. IEEE* **103**(1), 14–76 (2014)
8. Feamster, N., Rexford, J., Zegura, E.: The road to SDN: an intellectual history of programmable networks. *ACM SIGCOMM Comput. Commun. Rev.* **44**(2), 87–98 (2014)
9. Nunes, B.A.A., Mendonca, M., Nguyen, X.N., Obraczka, K., Turletti, T.: A survey of software-defined networking: past, present, and future of programmable networks. *IEEE Commun. Surv. Tutor.* **16**(3), 1617–1634 (2014)
10. Guck, J.W., Bemten, A.V., Reisslein, M., Kellerer, W.: Unicast QoS routing algorithms for SDN: a comprehensive survey and performance evaluation. *IEEE Commun. Surv. Tutor.* **20**(1), 388–415 (2018)

11. Satty, T.: How to make a decision: the analytic hierarchy process. *Eur. J. Oper. Res.* **48**(1), 9–26 (1990)
12. Zhang, L., Deering, S., Estrin, D., Shenker, S., Zappala, D.: RSVP: a new resource reservation protocol. *IEEE Commun. Mag.* **40**(5), 116–127 (2002)
13. Egilmez, H.E., Dane, S.T., Bagci, K.T., Tekalp, A.M.: OpenQoS an OpenFlow controller design for multimedia delivery with end-to-end quality of service over software-defined networks. In: 2012 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (2012)
14. Seddiki, M.S., et al.: FlowQoS: QoS for the rest of us. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, pp. 207–208 (2014)
15. Seddiki, M.S., Shahbaz, M., et al.: FlowQoS: per-flow quality of service for broadband access networks. Technical report, Georgia Institute of Technology (2015)
16. Yan, J., Zhang, H., Shuai, Q., Liu, B., Guo, X.: HiQoS: an SDN-based multipath QoS solution. *China Commun.* **12**(5), 123–133 (2015)
17. Yu, T.F., Wang, K., Hsu, Y.H.: Adaptive routing for video streaming with QoS support over SDN networks. In: International Conference on Information Networking 2015, pp. 318–323, March 2015. <https://doi.org/10.1109/ICOIN.2015.7057904>
18. Li, J., Chang, X., Ren, Y., Zhang, Z., Wang, G.: An effective path load balancing mechanism based on SDN. In: 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications (2014)
19. Gelenbe, E.: Machine learning for network routing. In: 2020 9th Mediterranean Conference on Embedded Computing (MECO), p. 1 (2020). <https://doi.org/10.1109/MECO49872.2020.9134073>
20. Mao, B., et al.: Routing or computing? The paradigm shift towards intelligent computer network packet transmission based on deep learning. *IEEE Trans. Comput.* **66**(11), 1946–1960 (2017). <https://doi.org/10.1109/TC.2017.2709742>
21. Reis, J., Rocha, M., Phan, T.K., Griffin, D., Le, F., Rio, M.: Deep neural networks for network routing. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2019). <https://doi.org/10.1109/IJCNN.2019.8851733>
22. Yu, C., Lan, J., Guo, Z., Hu, Y.: DROM: optimizing the routing in software-defined networks with deep reinforcement learning. *IEEE Access* **6**, 64533–64539 (2018). <https://doi.org/10.1109/ACCESS.2018.2877686>
23. ITU-T: ITU-T recommendation G.1010 end-user multimedia QoS categories. ITU-T (2001)
24. Yi, Z.: Research on QoS routing of SDN based on OpenFlow. Master’s thesis, University of Electronic Science and Technology of China (2018)
25. Saaty, T.L.: Decision making with the analytic hierarchy process. *Int. J. Serv. Sci.* **1**(1), 83–98 (2008)
26. Zadeh, L.A.: Fuzzy sets. In: *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A Zadeh*, pp. 394–432. World Scientific (1996)
27. Octopress: mininet.org. <http://mininet.org/>. Accessed 12 Feb 2020
28. Fernando, O.A., Xiao, H., Che, X.: Evaluation of underlying switching mechanism for future networks with P4 and SDN (workshop paper). In: Wang, X., Gao, H., Iqbal, M., Min, G. (eds.) CollaborateCom 2019. LNICST, vol. 292, pp. 549–568. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30146-0_38
29. projectfloodlight.org. <http://www.projectfloodlight.org/floodlight/>. Accessed 11 Aug 2020
30. iperf.fr. <http://iperf.fr/>. Accessed 23 Nov 2020