QLB: QoS Routing Algorithm for Software-Defined Networking

Piyawit Tantisarkhornkhet and Warodom Werapun
Department of Computer Engineering,
Faculty of Engineering, Prince of Songkla University
Phuket, Thailand
Email: {5810120060@email.psu.ac.th, warodom@coe.phuket.psu.ac.th}

***Abstract*—Software-Defined Networking (SDN) is a new efficiently idea of programmable networks that separates the control plane from data plane of all network devices. Internet service provider is responsible for all the control decisions and communication among the forwarding elements from centralized controller. SDN provides the various optimized services. Quality of service (QoS) routing is a path computation method that is suitable for the different traffics generated by several applications, while utilization of network resources has increased. This agreement of service is defined by QoS requirements such as throughput, delay, jitter and packet loss etc. Multimedia applications often require assured from multi QoS constrained, causing the NP-complete problem which cannot be simply solved in polynomial time and high management complexity in the transition network. SDN is able to reduce complexity and it is used to efficiently implement traffic, hence SDN significantly values to development QoS routing. In this paper, we propose QoS routing algorithm called Quantized Level Balance (QLB) for SDN that considers one or many QoS parameters relating to the network application. To satisfy the requirements, QLB selects QoS parameters depending to the level of appropriate application service quality. We have replicated our algorithm on simulate topology with Scalable Video-streaming Evaluation Framework (SVEF). We measure the Peak Signal-to-Noise Ratio (PSNR) and Mean Opinion Score (MOS) of Scalable Video Coding (SVC) at the receiver. Our propose algorithm is improved than single-metric approach that may choose poor QoS parameter paths.**

Keywords—Software-defined networking; multimedia; routing algorith; Dijkstra;

I. INTRODUCTION

Presently, there is a new efficiently network management called Software-defined networking (SDN) [1] which manages network by programmable concept. It offers important advantages over than traditional network by decoupling the control plane from data plane of all SDN network devices [2] to bring more flexible and manageable. Internet service provider is responsible for all the control decisions and communication among the forwarding elements from centralized controller. Consequently, it can apply SDN to lead control software relaying on network resources and routing.

Quality of Service (QoS) allows various applications or services operating as expected [3]. There are several problems of network performance such as the problems of throughput, delay, jitter and packet loss, which are called QoS parameter, in common network. The QoS constraints have three basic composition rules for complete path with respect to each QoS parameter such as Additive Metric (delay, hop count, cost, and jitter), Multiplicative metric (reliability and loss) and Concave metric (bandwidth) [4]. Each QoS parameter uses different rules. The evolution of QoS is more complex in routing as the QoS requirements specified by the various multimedia applications service. Moreover, QoS routing is suitable for multimedia applications such as voice over IP, video streaming and video conferencing etc. They are applications that require Multiple Constraint Path (MCP) [5]. On the other hand, some traditional routing protocols work on the network with a single mixed metric approach [6],[7] which is inadequate for multimedia applications. Additionally, they are increasingly used on the network.

In this paper, we propose QoS routing algorithm that is considered as one or many QoS parameters depending on the network application. To satisfy the requirements, our proposed algorithm takes QoS parameters by examining the level of appropriate quality for application services. We simulate our algorithm on simulate topology with SVEF which is an open-source SVC framework. We measure the PSNR and MOS of SVC at the receiver. It offers a better performance than only combination of equation that may choose some poor paths with unsatisfied QoS parameter.

The remainder of this paper is structured as follows. Literature review is discussed in Section II. Section III explains Quality of Service and SVC. Techniques of work are presented in Section IV. Finally, we summarize this work in Section V.

II. LITERATURE REVIEW

This section shows some different routing algorithms for NP-complete solutions such as Heuristic, Approximation and Randomization [8] as follows:

A. Heuristics algorithms

Heuristics algorithms were proposed to solve a NP-complete problem by reducing the complexity of path computation. The algorithm is fast whereas, it is not efficient to implement a best solution with acceptable probability. Heuristics algorithms can be classified into three types as follows:

978-1-5090-0629-8/16/$31.00 ©2016 IEEE

1) Linear composition

Linear composition is an algorithm that combines additive metrics. Y. Cui [9] and J. M. Jaffe [10] proposed converting any two additive weights to a single metric. Z. Wang [6], [7] proposed a single mixed metric approach for bandwidth, delay or loss. Although it can be an indicator in path selection, it is not sufficient for reliable QoS routing.

2) Lagranges relaxation Linear compositions

Lagranges relaxation Linear compositions is an algorithm that calculates lower bound and finds good solutions. It combined the two weights in terms of α to linear equation forms as an aggregate weight $w=w_1+ \alpha w_2$. In SDN, the linear equations form was used in [11]. They had low time complexity. A. Juttner [12] proposed an aggregated concept that considers cost and delay. It was used with SDN for service Scalable Video Coding (SVC) transmission in [13].

3) Non-Linear

Non-Linear is an algorithm that combined multi-constrained to a single weight by using non-linear formation. It was suitable for the metrics that are not correlated. T. Korkmaz [14] proposed H_MCOP algorithm that is the current best Non-Linear Heuristics algorithm. It uses Dijkstra algorithm two times including reverse direction by a linear function and forward direction by a non-linear function. This algorithm has also been proposed in SDN [15].

B. Approximation algorithms

Approximation algorithms were those heuristic that also implement some error bounds. They are efficient in arbitrary specified precision. However, they have high time complexity. G. Xue [16] proposed an Approximation algorithm that approximates all k-constraints without enforcing any one constraint.

C. Randomization algorithms

Randomization algorithms have randomness concept that are used to avert sudden problems when routing for a feasible path. Although they can be executed for inaccurate or dynamic networks, they are unfriendly with small probability. T. Korkmaz [13] implemented Randomized Breadth First Search (BFS) invents nodes from a chance node to a final destination node.

III. QUALITY OF SERVICE

QoS is needed to deliver uninterrupted multimedia services. There are many researches on how to satisfy the QoS requirements [17] such as throughput, delay, jitter, and packet loss etc.

A. QoS Requirements

The multimedia applications offer many services [18] such as video streaming, video on demand, voice over IP and video streaming etc. Each multimedia application uses a mechanism to provide differentiated QoS shows in TABLE I.

Voice over IP application uses low bandwidth. However, it is a real-time service required low latency, jitter and loss. In a real-time video application such as video conference, they should be effective in all QoS constraints as well as being high bandwidth, low latency, low jitter and low loss. Therefore, this application strongly needs suitable QoS parameters regarding to its service quality.

TABLE I. QoS REQUIREMENTS FOR TYPICAL APPLICATIONS

Application Type	Throughput Demand	Latency Tolerance	Jitter Tolerance	Packet Loss Tolerance
Email	Low	High	High	High
Web browsing	Low	High	High	High
File transfer (FTP)	Low-High	High	High	High
Chart (IM)	Low	Medium	Medium	Medium
Video streaming	Medium-High	Medium	Medium	Medium
Video on demand	High	Medium	Medium	Low
Voice over IP / WiFi	Low	Low	Low	Low
Video conferencing	Medium-High	Low	Low	Low

B. Scalable Video coding

Scalable Video Coding (SVC) is a video compression standard that is extension of the Advanced Video Coding (AVC). It translates data stream and conversely translates video into a bit stream. The SVC offers many benefits such as more flexible, leading to higher storage and reduced redundancy. SVC was used in diverse video applications such as video on demand, video conferencing and video streaming etc.

Presently, there are many SVC frameworks such as SVEF [19], myEvalSVC-Mininet [20] and DASH-SVC-Toolchain [21]etc. In this paper, we use SVEF for video streaming application since SVEF has been used in several frameworks. Therefore, we measure average PSNR and MOS from 4CIF YUV video at 30 fps that was encoded in SVC format with cross-layer scheduling by Joint Scalable Video Model (JSVM) in four constraints such as bandwidth, delay, jitter and loss. Since delay and jitter have low effect on PSNR, we show only MOS results that use in *Quantized_to_3levels()* as follows:

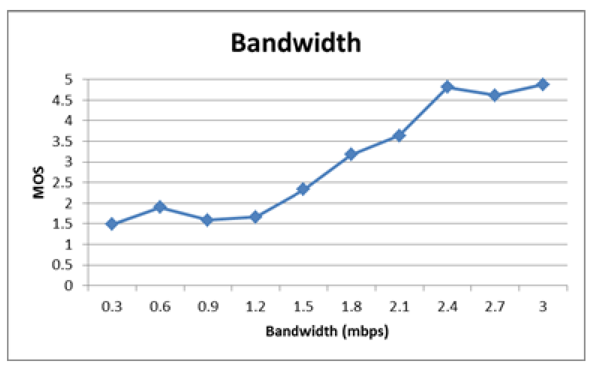

Fig. 1. MOS of received SVC stream from bandwidth constraint

From Fig. 1, in too low bandwidth, the traffic will be congested, lead to have high jitter and loss. Thus, users feel annoyed when they watch the video stream at low bandwidth. In Fig. 2 shows loss rate which is important effect on PSNR and MOS down. It makes distortion in video frame.

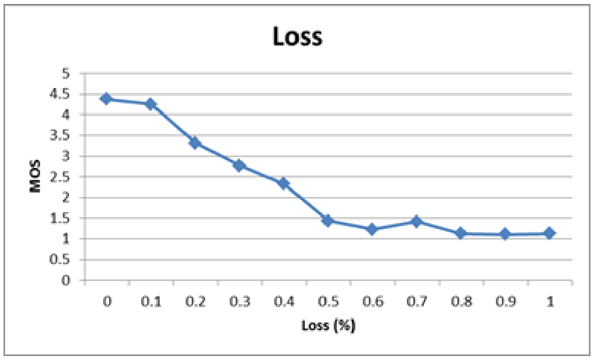

Fig. 2. MOS of received SVC stream from loss constraint

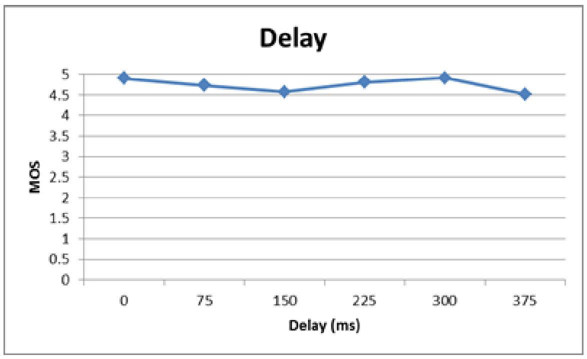

Fig. 3. MOS of received SVC stream from delay constraint

Delay has low effect on MOS as show in Fig. 3. We set the delay at 100 ms (fair rating) because there is nothing changed for MOS results of a received video for various fixed delay values [22] as show in Fig. 4.

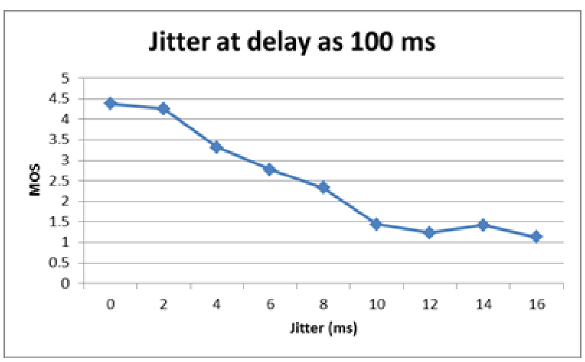

Fig. 4. MOS of received SVC stream from jitter constraint

These graphs are examining by calculating the average MOS values for each constraint. The user's perceived quality of watching a video can be analyzed. We use the MOS for group levels. The MOS level is normally used to rate the quality of videos from 1 to 5. However, our experiment currently supports 3 levels. Thus, we define lower 3 in MOS is 3 in QLB, between 3 and 4 in MOS is 2 in QLB and upper 4 in MOS is 1 in QLB.

In Table II, we provide summarizing results of constrained level such as bandwidth, loss and jitter. Constraint Level (Table II) is used in *Quantized_to_3levels()*. However, all constrained varies with the types of codec used and video motion.

TABLE II. CATEGORY REFERENCES OF QUALITY PARAMETER

QoS Constraints	QLB Levels		
	1 Good (MOS < 3)	2 Medium (4 > MOS > 3)	3 Poor (MOS > 4)
Bandwidth level	> 2.2	1.6-2.2	< 1.6
Loss level	< 0.2	0.2-0.3	> 0.3
Jitter level	< 4	4-6	> 6

IV. TECHNIQUES

This section shows the single mixed metric [6], [7] problem and proposes our algorithm called QLB that improved the single mixed metric. The mixed QoS parameter method is a tempting Heuristic. It consists of bandwidth, delay and loss rate as expressed in Equation (1).

$$f(p) = \frac{bandwidth(p)}{delay(p)*loss(p)} \quad (1)$$

In order to compare the single mixed metric to our approach, the equation (1) must be inverted into weight from (2). This equation will be applied in our simulation.

$$Weight(p) = \frac{\text{delay(p)*loss(p)}}{\text{bandwidth(p)}} \quad (2)$$

From the above equation (1), we simply define 2 mbps bandwidth with delay and packet loss following in TABLE III, and then an example of single mixed metric result can be calculated as follows:

For example: $f(p) = \frac{2000}{100\times25} = \frac{2000}{200\times12.5} = \frac{2000}{500\times5} = 0.8$

TABLE III. QOS PARAMETER VALUE EXAMPLES

No.	Delay	Packet Loss
1	Good (100)	Poor (25)
2	Medium (200)	Good (12.5)
3	Poor (500)	Good (5)

The single mixed metric combines QoS parameters. Although different QoS parameters are used, the single mixed metric may produce the same result. Nevertheless, a good routing algorithm must select a path that has balance-QoS parameter values as second row of TABLE III for multimedia applications which should not have any poor QoS parameter.

A. The algorithm description

Our proposed algorithm takes QoS parameters suitable for application services. Selection of QoS parameters depends on TABLE I and level for QoS parameters regarding on TABLE II e.g., video streaming should analyze four parameters such as bandwidth, delay, jitter and loss since TABLE I illustrate that video streaming is able to work on medium throughput, delay, jitter and loss. However, from constraint measure results in Fig. 1-4 show that delay does not has effect on the video quality too much. Thus, we use throughput, jitter and loss for this work.

Steps:

1. Use *Dijkstra's algorithm* for QLB
2. Quantized all QoS parameters to a desired level and use the mixed metric weight.
3. In this step, there are two choices for classify each QoS parameters to level QoS parameters as follows:
   a. Total Weight
   b. Balance Weight

B. Dijkstra's algorithm for QLB

Dijkstra's algorithm is serviced to search the optimum shortest paths between nodes in a directed graph G (V, E), denote: set of nodes (V) and set of edges (E). We modify Dijkstra's algorithm to consider three constraints and consider weights as follows: ew[] is node, ew[][] are raw QoS parameters, ew[][][] are quantized QoS parameters, level weight (ew[][][P]) and the mixed single metric weight (ew[][][P+1]). Denote: Number of Level of category (L) and Number of QoS parameter (P).

Function of Dijkstra's algorithm consists of two functions as follows:

```
Modify_Dijkstra_Algorithm function
Input: G, s
Output: dis[V], pre[V]
for each v in V
    dis[v] ← ∞
    pre[v] ← null
Q ← set(node.keys)
while (Q!=null)
    u← Minimum_distance(d,Q)
    for each v adjacent to u do
        adj[v][][0] ← Min (dis[u][0], ew[u,v][][0])
        adj[v][][1] ← dis[u][1]+ew[u,v][][1]
        adj[v][][2] ← dis[u][2]+ew[u,v][][2]
        adj[] ← Quantized_to_3levels (adj[])
        adj[] ← Total_Weight (adj[]) or
                          Balance_Weight(adj[])
     if dis[v][0] = adj[v][][P] then
         if dis[v][1] > adj[v][][P+1] then
             dis[v][0]← adj[v][][P]
             dis[v][1]← adj[v][][P+1]
             pre[v][0]←dis[v][0]
             pre[v][1]←dis[v][1]
      else if dis[v][0] > adj[v][][P] then
          dis[v][0]← adj[v][][P]
          dis[v][1]← adj[v][][P+1]
          pre[v][0]←dis[v][0]
          pre[v][1]←dis[v][1]
```

Fig. 5. Modify_Dijkstra's Algorithm for QLB

Our *Modify_Dijkstra_Algorithm()* (Fig. 5) is enhanced from an original algorithm that considers QoS parameters while collect weight from source to node v in adj[v][][0] (bandwidth), adj[v][][1] (loss) and adj[v][][2] (jitter). It uses *Quantized_to_3levels()* for calculating a mixed single metric (adj[][][P+1]) and quantizing each QoS parameter and uses *Total_Weight ()* or *Balance_Weight ()* for calculating weight level (adj[][][P]). Additional consideration is to focus on the level weight (adj[][][P]) and a mixed single metric (adj[][][P+1]), respectively. Its time complexity is O(E) as same as an original Dijkstra's algorithm.

```
Modify_Minimum_distance function
Input: distance, Q
Output: node, min
1:  min_level ← ∞ ;
2:  min_mix ← ∞ ;
3:  node ← 0
4:  for v in Q do
5:      adj_node[] ← Quantized_to_3levels (distance[])
6:      adj_node[] ← Total_Weight (distance []) or
                         Balance_Weight(distance [])
7:      if adj_node [v][][P] = min_level then
8:          if adj_node [v][][P+1] < min_mix then
9:              min_level ← adj_node [v][][P]
10:             min_mix ← adj_node [v][][P+1]
11:             node ← v
12:     else if adj_node [v][][P] < min[0] then
13:         min_level ← adj_node [v][][P]
14:         min_mix ← adj_node [v][][P+1]
15:         node ← v
```

Fig. 6. Modify_Minimum_distance function

Modify_Minimum_distance() is a component of Dijkstra's algorithm used to retrieve a node that has the first minimum level weight [line11]. If the minimum level weight is equal to the weight being compared [line 6], the next weight has to be checked [line 7].

C. The Quantized of QoS parameters

This section presents the conversion of the QoS parameter to a desired level by the quality of threshold that is defined in TABLE II., Definition: Good=1, Medium=2 and Poor=3. *Quantized_to_3levels()* (Fig. 7) is used to calculate a mixed metric equation and divide three threshold levels.

In calculate mixed metric equation, we adapt (2) that consist of bandwidth, delay and loss without jitter. Jitter is added into (3) in order to calculate mixed metric.

$$Weight_{\textit{mixed metric}} = \frac{\text{delay}*\text{jitter}*\text{loss}}{bandwidth} \quad (3)$$

From (3), we calculate each QoS parameter for weight of path by different rules. Delay and jitter are additive metric and loss is multiplicative metric. However, loss constraint of path in Mininet simulator, calculate from additive metric during links (4).

$$Weight_{\textit{mixed metric path}} = \frac{\sum \text{delay}*\sum \text{jitter}*\sum loss}{bandwidth} \quad (4)$$

In addition, it is serviced for QoS parameters which are stored at edge weights (ew[][]). It begins from Bandwidth [line 2-4], Jitter [line 5-7] and Loss. [line 8-10]. Thresholds in Fig. 7, they are defined from TABLE II which we can take them to a multimedia service appropriately.

Denote: ew[][0] is edge weight of delay level (unused in this work), ew[][1] is edge weight of bandwidth level, ew[][2] is edge weight of loss level and ew[][3] is edge weight of jitter level.

```
Quantized_to_3levels function
Input: ew
Output: ew
ew[][][P+1]=mixed metric equation (4)
if ew[][1] > 2.2 then ew[][][1] ← 1
else if ew[][1] ≥ 1.6 then ew[][][1] ← 2
else if ew[][1] < 1.6 then ew[][][1] ← 3
if ew[][2] < 0.2 then ew[][][2] ← 1
else if ew[][2] ≤ 0.3 then ew[][][2] ← 2
else if ew[][2] > 0.3 then ew[][][2] ← 3
if ew[][3] < 4 then ew[][][3] ← 1
else if ew[][3] ≤ 6 then ew[][][3] ← 2
else if ew[][3] > 6 then ew[][][3] ← 3
```

Fig. 7. Quantized_to_3levels function

D. Total Weight

This method is to sum the levels of each QoS parameters. It requires time complexity lower than Balance Weight.

Fig.8 shows *Total_Weight()* function. It is used for grouped of addition level of each QoS parameter (ew[][][P+1]) [line 1] and subtract the summation of edge weight from (L-1) for weight level [line 2].

```
Total_Weight function
Input: ew
Output: ew
1: ew[][][P+1] ← sum(each ew[0])
2: ew[][][P+1] ← (ew[][][P+1] – (L-1)) / MG
```

Fig. 8. Total_Weight function

Total Weight has a maximum number of groups (MG) that depends on Number of Level of category (L) and Number of QoS Parameter (P). We can predict a maximum number of groups from equation (5), (6).

$$MG_{Total\ Weight} = 1, \quad where\ L=1 \tag{5}$$

$$MG_{Total\ Weight} = L + [(P-1)*2^{(L-2)}],\ where\ L > 1 \tag{6}$$

E. Balance Weight

This method classifies by examining each QoS category level and sorts desired levels from low to high. It is used to avoid some poor QoS parameters.

Fig. 9 presents *Check_level_parameter()* that supports *Balance_Weight()* for counting the numbers in each level of QoS parameters. Its time complexity is O(P*L).

```
Check_level_parameter function
Input: ew, P, L
Output: count
for p=1 to P do
    for l=1 to L do
        if ew[l] = p
            count[p]+1
```

Fig. 9. Check_level_parameter function

```
Balance_Weight function
Input: ew
Output: ew
1:  count[] ← Check_level_parameter(ew, P, L)
2:  if count[1] = L
3:      ew[][][P+1] = 1/MG
4:  if count[2] > 0 and count[3] = 0
5:      for n=1 to P do
6:          if count[1] = P-n and count[2] = n
7:              ew[][][P+1] = n+1/MG
8:  if count[3] > 0
9:  for n=1 to P do
10:         if count[3] = n
11:             for m= 0 to P-1
12:                 if count[1] = P-n-m and count[2] = m
13:                 ew[][][P+1] = P+m+n+1/MG
```

Fig. 10. Balance_Weight

Fig. 10 illustrates *Balance_Weight()* that uses *Check_level_parameter()* for counting number of each QoS parameter level in link. This function supports the maximum level is 3. If QoS parameter level has only 1, it is found in [line 2-3]. If QoS parameter level has 1 and 2, it is found in [line 4-7]. If QoS parameter level has 1, 2 and 3, it is found in [line 9-13]. The time complexity *Balance_Weight()* function is depended on maximum metric level.

We can predict each maximum number of groups from equations (7) - (10).

$$MG_{Balance} = 1, \quad where\ L=1 \tag{7}$$

$$MG_{Balance} = 1+P, \quad where\ L=2 \tag{8}$$

$$MG_{Balance} = 1 + P + \sum_{k=1}^{P} k,\ where\ L=3 \tag{9}$$

$$MG_{Balance} = 1 + P + \sum_{k=1}^{P} k + MG_{Balance\ (P-1)},\ where\ L=4 \tag{10}$$

F. Simulate

In our proposed algorithm, we simulate on the topology and information regarding to Bandwidth (B), jitter (J) and loss (L) by using three threshold levels in Mininet. We measure PSNR from 4CIF YUV video at 30 fps that is encoded in SVC format by Joint Scalable Video Model (JSVM).

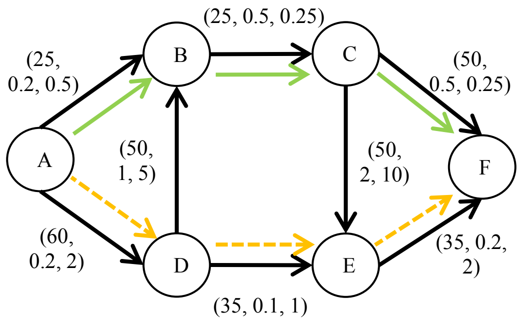

Fig. 11. Simple topology

Fig. 11 shows simple topology with three constrained such as bandwidth (mbps), loss rate (percent) and jitter (ms) at delay as 100 ms and results from calculating links and paths by Total Weight and Balance Weight, respectively. We show different cases for path choosing between Total Weight and Balance Weight. Total Weight function selects the top path (A-B-C-F) since total level weight is lowest (PSNR is 17 dB and MOS is 1.7). Balance Weight function takes the bottom path (A-D-E-F) which is the lowest balance level weight (PSNR is 20 dB and MOS is 2.3). The comparison results between total weight and balance weight reveal that balance weight has about 3 dB PSNR and 0.6 MOS which are greater than total weight.

V. Conclusion

Multimedia application has been continuously increased in the world network. QoS routing is popular requested for multimedia application. Furthermore, the most QoS transmissions have the NP-complete problem since they consider multi-constrained. SDN is another network management concept which is developed continuously for the reason that it has advantages more than traditional network. Therefore, it is suitable for various services, particularly multimedia services.

In this paper, we propose an algorithm called QLB with SDN-aware. It is used to consider MCP which uses quantized and classifies into groups. We arrange into two types: 1) Total Weight and 2) Balance Weight. Total Weight is summarized QoS category levels, whereas Balance Weight is sorted each QoS category level in ascending order. Although time complexity of Balance Weight is more than Total Weight, Balance Weight is suitable for generally multimedia applications since the majority of multimedia services required each suitable QoS category with well path selection. Future work, we will quantize five levels from Mean Opinion Score (MOS). Moreover, we are going to simulate various videos with frame sizes on different large topologies that have dynamic update of QoS parameters

Acknowledgment

This research has been supported by the Engineering Post-Graduate Scholarship, Faculty of Engineering, Prince of Songkla University, Thailand.

References

[1] H. Farhady, H. Lee, and A. Nakao, "Software-defined networking: A survey," *Comput. Netw.*, vol. 81, pp. 79–95, 2015.

[2] B. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *Commun. Surv. Tutor. IEEE*, vol. 16, no. 3, pp. 1617–1634, 2014.

[3] M. Curado and E. Monteiro, "A survey of QoS routing algorithms," in *Proceedings of the International Conference on Information Technology (ICIT 2004), Istanbul, Turkey*, 2004.

[4] P. Karkazis, P. Trakadas, H. C. Leligou, L. Sarakis, I. Papaefstathiou, and T. Zahariadis, "Evaluating routing metric composition approaches for QoS differentiation in low power and lossy networks," *Wirel. Netw.*, vol. 19, no. 6, pp. 1269–1284, 2013.

[5] P. Van Mieghem, F. A. Kuipers, T. Korkmaz, M. Krunz, M. Curado, E. Monteiro, X. Masip-Bruin, J. Solé-Pareta, and S. Sánchez-López, "Quality of service routing," in *Quality of Future Internet Services*, 2003, pp. 80–117.

[6] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *Sel. Areas Commun. IEEE J. On*, vol. 14, no. 7, pp. 1228–1234, 1996.

[7] Z. Wang and J. Crowcroft, "Bandwidth-delay based routing algorithms," in *Global Telecommunications Conference, 1995. GLOBECOM'95., IEEE*, 1995, vol. 3, pp. 2129–2133.

[8] S. Upadhaya and G. Devi, "Characterization of QoS based routing algorithms," *Int. J. Comput. Sci. Emerg. Technol.*, vol. 133, 2010.

[9] Y. Cui, K. Xu, and J. Wu, "Precomputation for multiconstrained QoS routing in high-speed networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, 2003, vol. 2, pp. 1414–1424.

[10] J. M. Jaffe, "Algorithms for finding paths with multiple constraints," *Networks*, vol. 14, no. 1, pp. 95–116, 1984.

[11] S. Laga, T. Van Cleemput, F. Van Raemdonck, F. Vanhoutte, N. Bouten, M. Claeys, and F. De Turck, "Optimizing scalable video delivery through OpenFlow layer-based routing," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*, 2014, pp. 1–4.

[12] A. Jüttner, B. Szviatovski, I. Mécs, and Z. Rajkó, "Lagrange relaxation based method for the QoS routing problem," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2001, vol. 2, pp. 859–868.

[13] T. Korkmaz and M. Krunz, "Multi-constrained optimal path selection," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2001, vol. 2, pp. 834–843.

[14] Y. Tsung-Feng, K. Wang, and Y.-H. Hsu, "Adaptive routing for video streaming with QoS support over SDN networks," in *Information Networking (ICOIN), 2015 International Conference on*, 2015, pp. 318–323.

[15] L. Sheng, Z. Song, and J. Yang, "A multi-constrained routing algorithm for software defined network based on nonlinear annealing," *J. Netw.*, vol. 10, no. 6, pp. 376–384, 2015.

[16] G. Xue, A. Sen, W. Zhang, J. Tang, and K. Thulasiraman, "Finding a path subject to many additive QoS constraints," *IEEEACM Trans. Netw. TON*, vol. 15, no. 1, pp. 201–211, 2007.

[17] W. Sugeng, J. E. Istiyanto, K. Mustofa, and A. Ashari, "The Impact of QoS Changes towards Network Performance," *Int. J. Comput. Netw. Commun. Secur.*, vol. 3, no. 2, pp. 48–53, 2015.

[18] Y. Chen, T. Farley, and N. Ye, "QoS requirements of network applications on the Internet," *Inf. Knowl. Syst. Manag.*, vol. 4, no. 1, pp. 55–76, 2004.

[19] A. Detti, G. Bianchi, W. Kellerer, and others, "SVEF: an Open-Source Experimental Evaluation Framework," in *In Proc. of IEEE MediaWIN 2009, Sousse, Tunisia*, 2009.

[20] C.-H. Ke, "myEvalSVC: an Integrated Simulation Framework for Evaluation of H. 264/SVC Transmission.," *TIIS*, vol. 6, no. 1, pp. 379–394, 2012.

[21] C. Kreuzberger, D. Posch, and H. Hellwagner, "A scalable video coding dataset and toolchain for dynamic adaptive streaming over HTTP," in *Proceedings of the 6th ACM Multimedia Systems Conference*, 2015, pp. 213–218.

[22] V. D. Bhamidipati and S. Kilari, "Effect of delay/delay variable on QoE in video streaming," *M Thesis Sch. Comput. Blekinge Inst. Technol. Swed.*, 2010.