

A Greedy-Based Video Bitrate Selection Algorithm with Consideration of QoE Fairness for Adaptive Streaming over Software Defined Network

Kuei-Hong Lin and Kuo-Huang Chung

Abstract—The video streaming may be the most important service of the internet nowadays, and it changes the viewing habits of the millions of people. The novel streaming technique – adaptive streaming helps the people to watch the videos smoother, that is, the video quality or bitrate would be changed with the bandwidth adaptively. However, the design of the traditional adaptive streaming does not take the entire internet into account, that is, each device may grab the bandwidth as much as possible, and the allocation of the bandwidth may not fairness which causes the unfairness of the quality of experience (QoE). In this study, the software defined network (SDN) is applied to be the centralized control for allocating the bandwidth. Besides, a greedy-based video bitrate selection algorithm is proposed for achieving the QoE fairness. By comparing the results to the brute-force method, the proposed method can achieve the QoE fairness as the optimum solution of the brute-force method.

Index Terms—Adaptive streaming, QoE fairness, SDN, greedy-based, video bitrate selection.

I. INTRODUCTION

The internet protocol television (IPTV) provides the video streaming to the clients everywhere, and the streaming service changes the viewing habits of the millions of people. For providing the users smoother environment while they are watching the video, the adaptive streaming is developed, and the related techniques are MPEG-dynamic adaptive streaming over HTTP (DASH), Adobe-dynamic streaming for flash, Apple - HTTP live streaming (HLS), and Microsoft - Smooth streaming. The concept of adaptive streaming is shown in Fig. 1 [1] and Fig. 2 [1]. In Fig. 1, the inputted video with high bitrate is encoded into multiple bitrates and separated into small segments at the server side, and the client would download the manifest or media presentation description (MPD) file from the web server which memorizes the information of the bitrates and segments. The streaming player of the client side would estimate the bandwidth continuously, and the bandwidth estimate would be applied to select the proper bitrate from the manifest as shown in Fig. 2. This design can help the users watch the videos smoother without interrupt.

As mentioned before, the bandwidth estimation and bitrate selection plays the most important roles in adaptive streaming, and it is the main difference between the

traditional streaming techniques. However, the design of the adaptive streaming may grab the bandwidth as much as possible, that is, the client would try to acquire the video with higher bitrate, and the problem of the QoE unfairness [2] may be happened, that is, the device's screen with higher resolution may be affected severely as shown in Fig. 3. For solving the problem, the OpenFlow[3]-assisted QoE fairness framework (QFF) is proposed in [2]. The QFF applied the SDN [4] technique to make the centralized control, and the bitrate selection algorithm is applied to help the clients to achieve the QoE fairness. Based on QFF [2], a greedy-based video bitrate selection algorithm is proposed in this study, and it can be applied to the campus networking environment with complex topology and heterogeneous devices.

This study is organized as follows. In Section II, background of QFF introduced. In Section III, the systems is modeled and formulated. In Section VI, the problem of the QoE fairness is formulated. In Section V, the proposed algorithm is described. In Section VI, the experiment environment is built for evaluating the proposed method, and the experiment results are shown in Section VII. Finally, concluding comments are made in Section VIII.

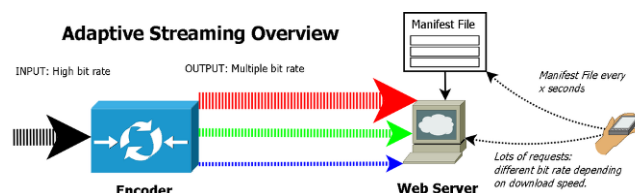


Fig. 1. Adaptive streaming overview.

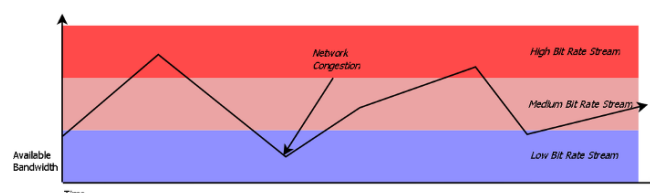


Fig. 2. Bitrate switching of the adaptive streaming.

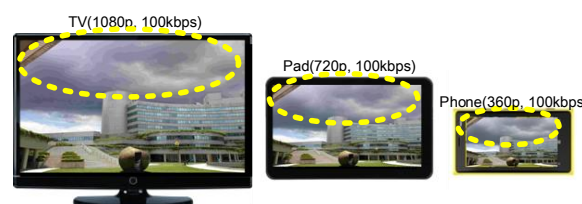


Fig. 3. Phenomena of the QoE unfairness.

II. BACKGROUNDS OF QFF

For achieving the QoE fairness over the heterogeneous devices, the video quality (VQ) of each video should be

Manuscript received August 20, 2014; revised December 12, 2014.

The authors are with the Information and Communications Research Laboratories of Industrial Technology Research Institute of Taiwan, Taiwan R.O.C (e-mail: kueihong990316@itri.org.tw, khchung@itri.org.tw).

quantified first, that is, the VQ should be evaluated objectively. In QFF [2], the structural similarity (SSIM) [5] is adopted as the VQ metric. The range of SSIM is from 0 to 1 (the best). As shown in Fig. 4 (the figures are from <https://ece.uwaterloo.ca/~z70wang/research/ssim/>), the SSIM is more objective than the mean square error (MSE) for expressing the VQ, and it shows that that $SSIM \geq 0.98$ the VQ can be much closed to the original image. Besides, if the difference of SSIM in two images is more than 0.05, the human can recognize the blemishes in the image with lower SSIM.

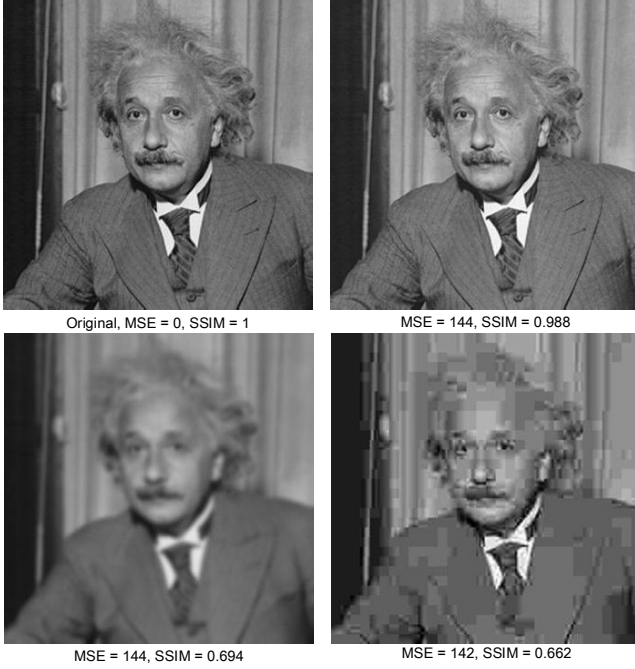


Fig. 4. SSIM vs MSE.

TABLE I: SET BITRATES FOR THREE VIDEO RESOLUTIONS

VID	Resolution	Video Bitrates (kbps)
1	1080p	100, 200, 600, 1000, 2000, 4000, 6000, 8000
2	720p	100, 200, 400, 600, 800, 1000, 1500, 2000
3	360p	100, 200, 400, 600, 800, 1000

TABLE II: MODEL AND COEFFICIENTS FOR UTILITY FUNCTION

VID	Resolution	Two-term Power Series Model: $f_i(b_i(l)) = Ab_i(l)^B + C$		
		A	B	C
1	1080p	-3.035	-0.5061	1.022
2	720p	-4.85	-0.647	1.011
3	360p	-17.53	-1.048	0.9912

After quantifying the VQ, the utility function (UF) is created for mapping the video bitrates to the VQ, and the two-term power series model is proven to be the most suitable function. The available bitrates of each video is expressed as

$$b_i = [b_i(1) b_i(2) \dots b_i(L_i)] \quad (1)$$

where b_i stands for the i^{th} video's available bitrates, i is the video identity (VID), $b_i(l)$ is the l^{th} available bitrate of b_i , and

L_i is the number of available bitrates of the i^{th} video. The corresponded VQs can be computed by the UF

$$f_i(b_i(l)) = Ab_i(l)^B + C \quad (2)$$

where the UF in (2) is the two-term power series model, and the coefficients of the model are A , B and C . For example, the Table I and Table II are the available bitrates of each video and the coefficients of the model. In Fig. 5, the mappings of the available bitrates (Table I) and the VQs from Table II are shown, and it shows that the relation between the bitrates and VQs is nonlinear.

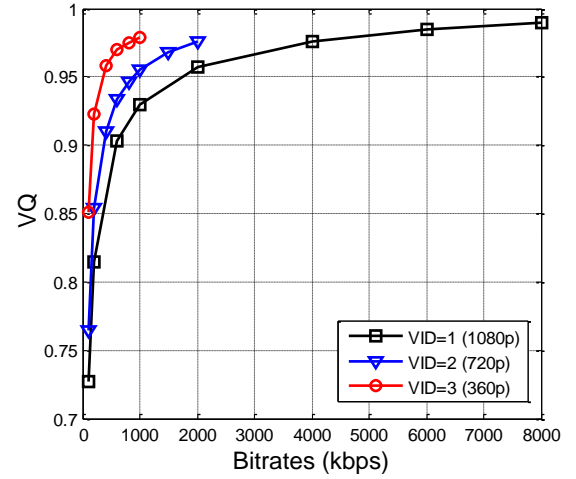


Fig. 5. Bitrate vs VQ.

III. SYSTEM MODEL DESCRIPTION

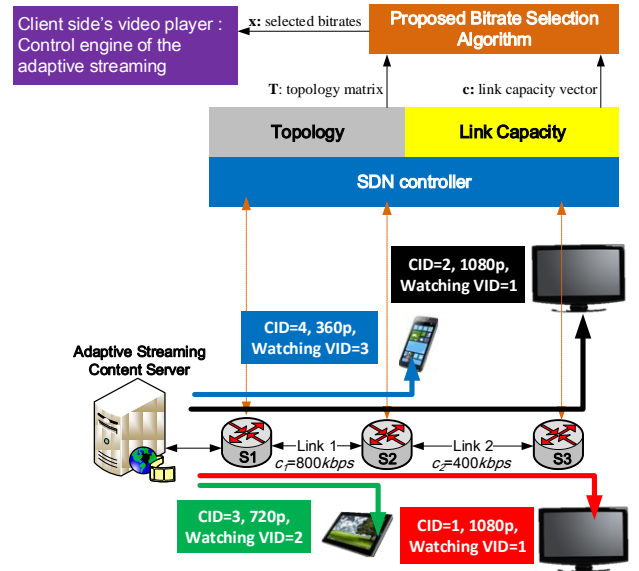


Fig. 6. QFF [1]-like system model.

The QFF-like system model is shown in Fig. 6. Note that the topology of Fig. 6 is just a simple sample for describing. The content server provides the adaptive streaming service for the heterogeneous client devices. Each client has a unique client identity (CID), and supports different resolution videos such as 360p, 720p, and 1080p. The clients would request the content with VID from the content server, and the content

server would provide the manifest to the client as mentioned in Fig. 1. The watching VID's of each client is denoted as

$$v = [v_1 \ v_2 \ \dots \ v_M] \quad (3)$$

where M is the number of clients, and $v(m)$ denotes the m^{th} element of v in (3). Note that $1 \leq m \leq M$ stands for the CID.

The topology and link capacity would be acquired by the northbound application programming interface (API) or application (APP) of the SDN controller from the open flow-enabled switches easily, and this is the advantage of SDN architecture. The topology of the network is represented as a matrix

$$T = [\delta_1 \ \delta_2 \ \dots \ \delta_M]^T$$

$$= \begin{bmatrix} \delta_{1,1} & \delta_{1,2} & \dots & \delta_{1,N} \\ \delta_{2,1} & \delta_{2,2} & \dots & \delta_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{M,1} & \delta_{M,2} & \dots & \delta_{M,N} \end{bmatrix} \quad (4)$$

where M and N are the number of clients and links, respectively, and $\delta_m = [\delta_{m,1} \ \delta_{m,2} \ \dots \ \delta_{m,N}]^T$ denotes as the m^{th} client's routing path. The m^{th} client's routing coefficient of the n^{th} link is defined as

$$\delta_{m,n} = \begin{cases} 1, & \text{if the } n^{th} \text{ link belongs to the } m^{th} \text{ clients routing path} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where m and n are the indexes of the clients and links, respectively. Note that the client index m corresponds to the CID. For example, the routing matrix T of Fig. 6 is represented as

$$T = [\delta_1 \ \delta_2 \ \delta_3 \ \delta_4]^T$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}^T \quad (6)$$

The capacity of each link c_n is represented as a vector

$$c = [c_1 \ c_2 \ \dots \ c_N] \quad (7)$$

where N is the number of links, and the elements of c in (7) stand for the available bandwidth of each link. The acquired topology matrix T in (4) and link capacity vector c in (7) would be provided to the proposed bitrate selection algorithm for maximizing the minimum QoE or VQ of the clients' watching videos, that is, it will try to find the bitrate set which can achieve the QoE fairness. The selected bitrates of each client is represented as

$$x = [x_1 \ x_2 \ \dots \ x_M] \quad (8)$$

where M is the number of clients. Finally, the selected bitrates x in (8) would be transmitted to the clients. In Fig. 6, the control engine receives x in (8), that is, the video player of

the client side is no need to operate available bandwidth estimation for making the video bitrate decision. The developed adaptive streaming over SDN application with the proposed bitrate selection algorithm would replace the video bitrate decision in traditional adaptive streaming player for achieving the QoE fairness. The proposed bitrate selection algorithm would be mentioned later.

IV. PROBLEM FORMULATION

The objective function for achieving the QoE fairness [1] is expressed as

$$p = \max[\min(q)] \quad (9)$$

where $q = [f_{v(1)}(x_1) \ f_{v(2)}(x_2) \ \dots \ f_{v(M)}(x_M)]$ are the video qualities or QoE values of each client, and $1 \leq i \leq I$ is the VID. Note that $f_i(x_m)$ in (2) is the video quality or QoE of the video with VID= i while the selected bitrate is x_m as mentioned in Section II, and I is the number of videos of the content server. For fitting the objective in (9), the proposed bitrate selection algorithm will find the bitrate set x , that is, a solution that can maximize the minimum QoE expressed as

$$x = \arg \max_x [\min(q)] \quad (10)$$

where x is the selected bitrates which are subjected to

$$c'_n = \sum_{m=1}^M x_m \delta_{m,n} \leq c_n \quad (11)$$

where c'_n is the demand of the n^{th} link. The constraints of each link in (11) can be rewritten as

$$c' = [c'_1 \ c'_2 \ \dots \ c'_N]^T$$

$$= xT$$

$$= [x_1 \ x_2 \ \dots \ x_M] \begin{bmatrix} \delta_{1,1} & \delta_{1,2} & \dots & \delta_{1,N} \\ \delta_{2,1} & \delta_{2,2} & \dots & \delta_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{M,1} & \delta_{M,2} & \dots & \delta_{M,N} \end{bmatrix} \quad (12)$$

$$\leq [c_1 \ c_2 \ \dots \ c_N]^T$$

where c' are the demands of the links.

The optimization of bandwidth allocation in (9) and (10) is not similar to the traditional maximum-minimum fairness (MMF) [6] problem, that is, the optimization of the bitrate selection for achieving QoE fairness is not a straightforward problem [2]. Since the available bitrates of each video is discrete. Moreover, it is impossible to find the solution via brute-force method from all the available bitrate sets immediately. Note that the total number of the available bitrate sets is

$$N_T = \prod_{m=1}^M L_{v(m)} \quad (13)$$

where m is the CID, M is the number of clients, and $L_{v(m)}$ stands for the number of available bitrates of the $v(m)^{th}$ video in (1) and (3).

V. PROPOSED BITRATE SELECTION ALGORITHM

For solving the optimization of bandwidth allocation in (9) and (10), the proposed bitrate selection algorithm applied the concept of greedy method [7] to find the sub-optimum solution rapidly. As shown in (10), the objective of the proposed algorithm is to find the bitrate set which can maximize the minimum VQ. Hence, the strategy of the proposed algorithm is to maximize the client with minimum VQ interactively.

The operation of the proposed algorithm could be divided into several steps as shown in Fig. 7. The details of the steps would be discussed in the following subsections, and the operating example is given in Table III which follows the example shown in Fig. 6.

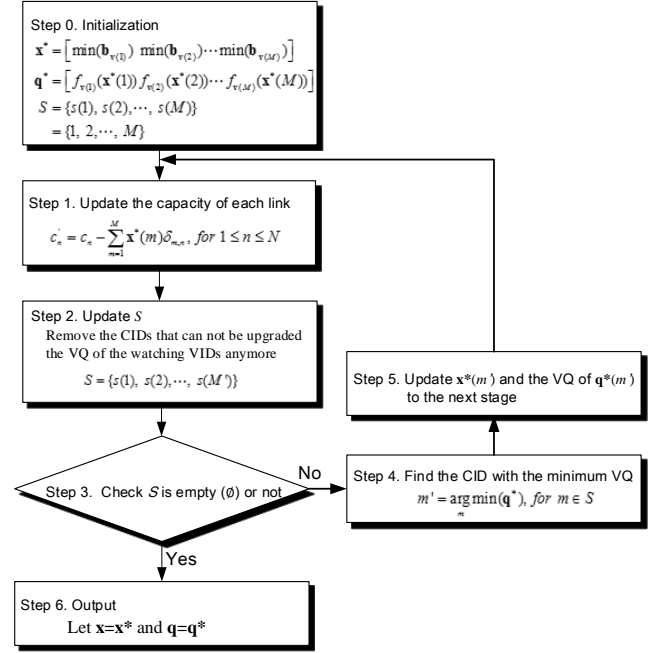


Fig. 7. Proposed greedy-based video bitrate selection algorithm.

TABLE III: EXAMPLE OF THE OPERATING PROCEDURE

t	Inputted and Updated S	Inputted x^* and q^*	m'	Updated x^* and q^*
0	$\{1, 2, 3, 4\}$ $\{1, 2, 3, 4\}$	$x^* = [100 \ 100 \ 100 \ 100]$ $q^* = [0.73 \ 0.73 \ 0.77 \ 0.85]$	1	$x^* = [200 \ 100 \ 100 \ 100]$ $q^* = [0.80 \ 0.73 \ 0.77 \ 0.85]$
1	$\{1, 2, 3, 4\}$ $\{2, 3, 4\}$	$x^* = [200 \ 100 \ 100 \ 100]$ $q^* = [0.80 \ 0.73 \ 0.77 \ 0.85]$	2	$x^* = [200 \ 200 \ 100 \ 100]$ $q^* = [0.80 \ 0.80 \ 0.77 \ 0.85]$
2	$\{2, 3, 4\}$ $\{3, 4\}$	$x^* = [200 \ 200 \ 100 \ 100]$ $q^* = [0.80 \ 0.80 \ 0.77 \ 0.85]$	3	$x^* = [200 \ 200 \ 200 \ 100]$ $q^* = [0.80 \ 0.80 \ 0.86 \ 0.85]$
3	$\{3, 4\}$ $\{4\}$	$x^* = [200 \ 200 \ 200 \ 100]$ $q^* = [0.80 \ 0.80 \ 0.86 \ 0.85]$	4	$x^* = [200 \ 200 \ 200 \ 200]$ $q^* = [0.80 \ 0.80 \ 0.86 \ 0.94]$
4	$\{4\}$ \emptyset	$x^* = [200 \ 200 \ 200 \ 200]$ $q^* = [0.80 \ 0.80 \ 0.86 \ 0.94]$	null	null

Step 0. Initialization

For providing the promised service to the clients, the initial bitrate of each client is the expressed as

$$\begin{aligned} x^* &= [b_{v(1)}(\theta_1) \ b_{v(2)}(\theta_2) \ \dots \ b_{v(M)}(\theta_M)] \\ &= [b_{v(1)}(1) \ b_{v(2)}(1) \ \dots \ b_{v(M)}(1)] \end{aligned} \quad (14)$$

where x^* stands for the allocated bitrates, $x^*(m) = b_{v(m)}(\theta_m)$ is the allocated bitrate of the m^{th} client, and $b_{v(m)}(\theta_m)$ is the θ_m^{th} element of the available bitrates $b_{v(m)}$ in (1). Note that $v(m)$ is the m^{th} element of the watching VIDs v in (3). The vector for saving the bitrate index θ_m of the m^{th} client with the corresponded watching VID $v(m)$ in (3) is

$$\theta = [\theta_1 \ \theta_2 \ \dots \ \theta_M] \quad (15)$$

Hence, all the initial values of θ are “1” which means the minimum values of $b_{v(m)}$ in (1), and the initial state of x^* in (14) can be simply expressed as

$$x^* = [\min(b_{v(1)}) \ \min(b_{v(2)}) \ \dots \ \min(b_{v(M)})] \quad (16)$$

The corresponded VQs of x^* in (16) is

$$q^* = [f_{v(1)}(x^*(1)) \ f_{v(2)}(x^*(2)) \ \dots \ f_{v(M)}(x^*(M))] \quad (17)$$

where the elements of q^* are computed by the UF in (2). Besides, the initial candidates for upgrading the VQs later are represented as

$$\begin{aligned} S &= \{s(1), s(2), \dots, s(M)\} \\ &= \{1, 2, \dots, M\} \end{aligned} \quad (18)$$

where the elements of the set S in (18) stand for the CIDs which have the opportunity to be upgraded the VQs later. Initially, the elements are the entire CIDs from 1 to M , and the updating procedure would be processed interactively until $S = \emptyset$.

Step 1. Update the Capacity of Each Link

In each iteration, the capacity of each link is updated by

$$c'_n = c_n - \sum_{m=1}^M x^*(m) \delta_{m,n}, \text{ for } 1 \leq n \leq N \quad (19)$$

where c'_n is the updated link capacity, c_n is the link capacity in (7). $x^*(m)$ is the m^{th} element of x^* in (14), and $\delta_{m,n}$ in (5) is the m^{th} client's routing coefficient of the n^{th} link.

Step 2. Update S

After updating the capacity of each link in Step 2, the updated available bandwidth of the m^{th} path is defined as w_m . If the bitrate gap Δ_m is smaller than w_m , the $VID=m$ would be removed from S , and the updated S is

$$S = \{s(1), s(2), \dots, s(M')\} \quad (20)$$

where M' is the number of element of S . Note that the bitrate gap Δ_m is defined as

$$\Delta_m = b_{v(m)}(\theta_m + 1) - b_{v(m)}(\theta_m), \text{ for } \theta_m \leq L_{v(m)} - 1 \quad (21)$$

where $L_{v(m)}$ is the number of available bitrates of the $v(m)^{th}$ video defined in (1) and (3).

Step 3. Check S is Empty (\emptyset) or Not

After processing the Step 2, if S becomes is not an empty (\emptyset) set. The procedure of the proposed algorithm would be processed interactively. If $S=\emptyset$, the interactive procedure would be stopped, and jumps to step 6 for outputting the selected bitrates $\mathbf{x}=\mathbf{x}^*$ and the corresponded $\mathbf{q}=\mathbf{q}^*$.

Step 4. Find the CID with the Minimum VQ

In Step 4, the VQ of the client would be updated if it has the minimum value. The CID for being updated is

$$m' = \arg \min_m (q^*), \text{ for } m \in S \quad (22)$$

where m' is the CID for being update the VQ of $v(m)$ later. Note that if the clients have the same VQ, and the VQ is the minimum value of \mathbf{q}^* in (17). Then, m' in (22) would be selected randomly from those clients.

Step 5. Update $x^*(m')$ and the VQ of $q^*(m')$ to the Next Stage

After processing Step 4, the m^{th} element of θ would be set as

$$\theta_{m'} = \theta_{m'} + 1 \quad (23)$$

Hence, the m^{th} element of \mathbf{x}^* in (14) and \mathbf{q}^* in (17) would be updated concurrently.

Step 6. Output

As mentioned in Step 3, Step 6 is the final procedure while $S=\emptyset$. The selected bitrates are defined as

$$\mathbf{x} = \mathbf{x}^* \quad (24)$$

An Operating Example

In Fig. 2, there are 4 clients in the network, and the CIDs and VIDs are shown in the figure. Thus, the CID is from 1 to 4, and the corresponded watching VIDs are $v=[1 \ 1 \ 2 \ 3]$. The link capacities are $\mathbf{c}=[800 \ 400]$ (kbps). The routing matrix \mathbf{T} is expressed in (6). In Table III, the index t is the interaction times.

At $t=0$, the set S , the allocated bitrate of each of each client \mathbf{x}^* , and the corresponded VQs of \mathbf{x}^* of the Step 0 are

$$S = \{1, 2, 3, 4\} \quad (25)$$

and

$$\mathbf{x}^* = [100 \ 100 \ 100 \ 100] \quad (26)$$

$$\mathbf{q}^* = [0.73 \ 0.73 \ 0.77 \ 0.85] \quad (27)$$

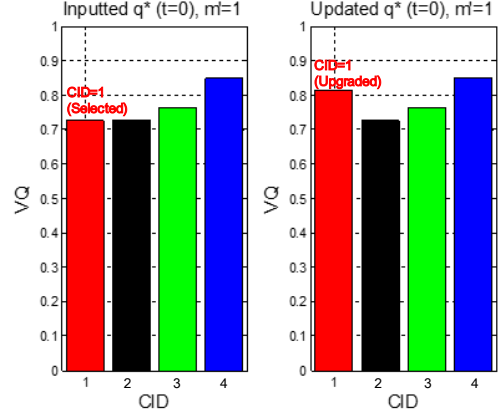


Fig. 8. (a) $t=0$, minimum VQ=0.73, and $m'=1$.

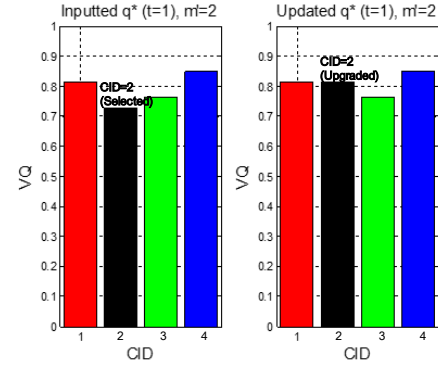


Fig. 8. (b) $t=1$, minimum VQ=0.73, and $m'=2$.

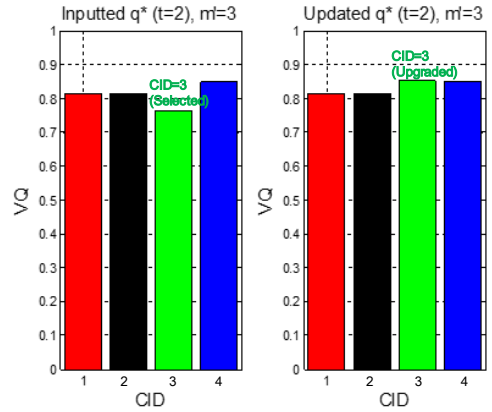


Fig. 8. (c) $t=2$, minimum VQ=0.77, and $m'=3$.

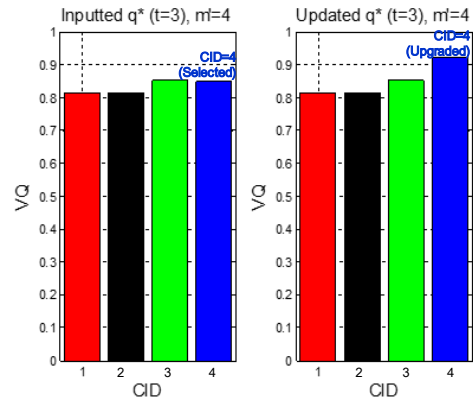


Fig. 8. (d) $t=3$, minimum VQ=0.85, and $m'=4$.

In Step 1 (Fig. 8(a)), the updated capacities of link 1 and 2 are $c_1=400$ (kbps) and $c_2=200$ (kbps). Thus, the updated S is same as (25) in Step 2, that is, $S \neq \emptyset$ in Step 3. In Step 4, the CID with the minimum VQ is $m'=1$, and

$$\mathbf{x}^* = [200 \ 100 \ 100 \ 100] \quad (28)$$

and

$$\mathbf{q}^* = [0.80 \ 0.73 \ 0.77 \ 0.85] \quad (29)$$

Are the updated version of \mathbf{x}^* (25) and \mathbf{q}^* (27).

At $t=1$, the updated capacities of link 1 and 2 are $c_1=300$ (kbps) and $c_2=100$ (kbps) in Step 1. Since the bandwidth is not enough for upgrading the VQ of CID=1, the updated $S=\{2, 3, 4\}$ in Step 2. In Step 4, the CID with the minimum VQ is $m'=2$, and

$$\mathbf{x}^* = [200 \ 200 \ 100 \ 100] \quad (30)$$

and

$$\mathbf{q}^* = [0.80 \ 0.80 \ 0.77 \ 0.85] \quad (31)$$

are the updated version of \mathbf{x}^* (27) and \mathbf{q}^* (29).

The interactive procedure operates from $t=0$ (Fig. 8(a)) to $t=3$ (Fig. 8(d)). At $t=4$ (Fig. 8(e)), the updated capacities of link 1 and 2 are $c_1=0$ (kbps) and $c_2=0$ (kbps) in Step 1. The updated S is \emptyset in Step 2. Thus, the interactive procedure would be stopped, and jumps to step 6 for outputting the selected bitrates

$$\mathbf{x} = \mathbf{x}^* = [200 \ 200 \ 200 \ 200] \quad (32)$$

and

$$\mathbf{q} = \mathbf{q}^* = [0.80 \ 0.80 \ 0.86 \ 0.94] \quad (33)$$

VI. EXPERIMENT SETUP

The bitrates and VQs for testing the proposed method are from [2] which are given in Table I and Table II. There are 3 videos with different resolutions such as 360p, 720p, and 1080p. The client number is from 2 to 200, and the link number is 20. The link capacity \mathbf{c} in (7), watching VID of each client \mathbf{v} in (3), and routing matrix \mathbf{T} in (4) are generated randomly. The number of simulating (N_s) times is 1000.

For evaluating the performance of the proposed method, there are 3 methods for being compared, where the compared methods are the brute-force, the equal bandwidth (EQ-BW), and the traditional adaptive streaming. The brute-force method would list all available bitrate sets, and choose the set with the maximum-minimum VQ, that is, the brute-force method provides the global solution that can achieve the objective in (9) and (10), and maximize the utilization of bandwidth. The equal-bandwidth method, would compute the averaged capacity of each link

$$\bar{\mathbf{c}} = [\bar{c}_1 \ \bar{c}_2 \ \cdots \ \bar{c}_N] \quad (34)$$

where \bar{c}_n is the n^{th} link's averaged capacity which is acquired by

$$\bar{c}_n = c_n \left(\sum_{m=1}^M \delta_{m,n} \right)^{-1}, \text{ for } 1 \leq n \leq N \quad (35)$$

where M is the number of clients, and $\delta_{m,n}$ in (5) is the m th client's routing coefficient on the n th link. And the allocated bandwidth of each client is

$$x_{EQ} = \min(\bar{\mathbf{c}}) \quad (36)$$

Note that the bandwidth allocation of the traditional adaptive streaming method is a random number, that is, the QoE is not fairness.

The evaluating metrics are the correcting probability (P_c) defined as the average of the minimum VQ (\bar{q}_{\min}) of the N_s simulations, the average of the VQ (\bar{q}) of the N_s simulations, and the averaged bandwidth utilization defined as

$$P_c = \frac{\left(\begin{array}{l} \text{Number of times} \\ \text{minimum VQ of the compared method} \\ = \text{the minimum VQ of the brute-force method} \end{array} \right)}{N_s} \quad (37)$$

$$u = \frac{\sum_{n=1}^N \left(\frac{c_n - \sum_{m=1}^M \tilde{\mathbf{x}}(m) \delta_{m,n}}{c_n} \times 100\% \right)}{N}, \text{ for } 1 \leq n \leq N \quad (38)$$

where c_n in (7) is the capacity of each link, $\tilde{\mathbf{x}}(m)$ is the selected bitrate of the client with CID= m via the proposed method, the equal bandwidth, or the traditional adaptive streaming.

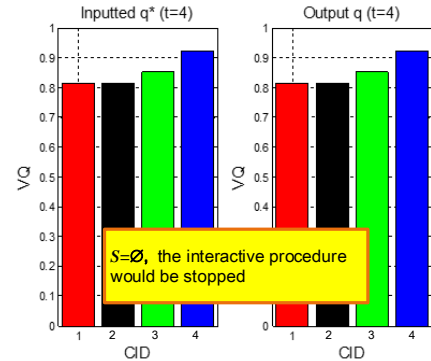


Fig. 8. (e) $t=4$, $S=\emptyset$, $\mathbf{x}=\mathbf{x}^*$ and $\mathbf{q}=\mathbf{q}^*$.

VII. EXPERIMENT RESULTS

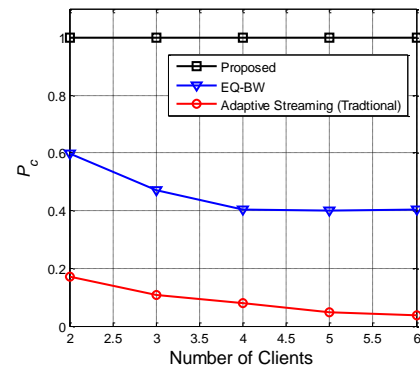


Fig. 9. Correcting probability (P_c).

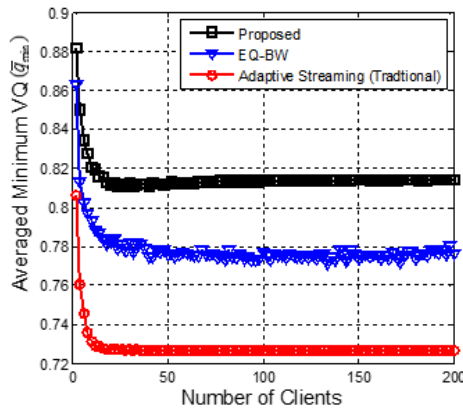


Fig. 10. Averaged minimum VQ.

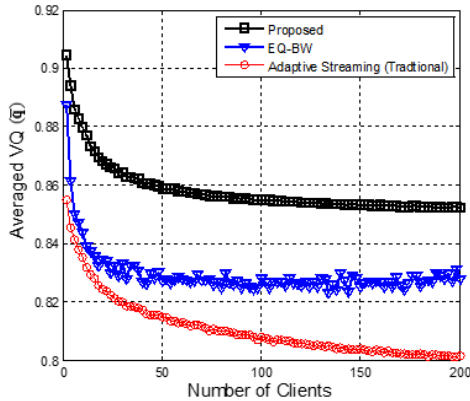


Fig. 11 Averaged VQ

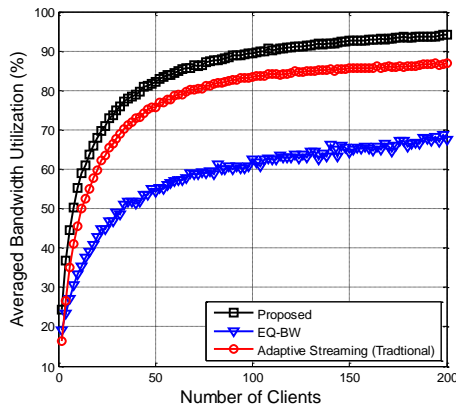


Fig. 12. Averaged bandwidth utilization.

In Fig. 9, the proposed method, equal-bandwidth, traditional adaptive streaming are compared to the brute-force method, and the correcting probability (P_c) is computed by (37). In the experiment, the result shows that the proposed method can achieve the same minimum VQ as the brute-force method with $P_c=1$, and it is well above other methods. The result shows the proposed method is an optimum solution.

In Fig. 10 and Fig. 11, the difference between the proposed method and other methods are more than 0.05. As mentioned in Fig. 4, if the difference of SSIM in two images is more than 0.05, the eyes can recognize the blemishes in the image with lower SSIM. Without a doubt, the proposed method can help the adaptive streaming over SDN to achieve the QoE fairness.

In Fig. 12, the proposed method can make the bandwidth utilization better than the compared methods. Therefore, the

proposed method can achieve QoE fairness and the better bandwidth utilization simultaneously.

VIII. CONCLUSION

A greedy-based video bitrate selection algorithm with consideration of QoE Fairness for adaptive streaming over SDN is proposed in this study. The proposed method could maximize the minimum VQ, and achieve the QoE fairness. Besides, the complexity of the proposed method is lower than the brute-force method. The refinements of the proposed method would be studied continuously, and we hope the proposed method could be a SDN's northbound application for providing better streaming service in the future.

ACKNOWLEDGMENT

This research is supported by the Information and Communications Research Laboratories (ICL), Industrial Technology Research Institute (ITRI), Taiwan, Republic of China (ITRI Grant Project Code D301AR2R30)

REFERENCES

- [1] T. D. Monchamp, "Adaptive bit-rate streaming-minimizing end-user buffer times in real-time video delivery," *Sight: Rivier Academic Journal*, vol. 9, no. 2, pp. 1-12, 2013.
- [2] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using open flowassisted adaptive video streaming," in *Proc. the 2013 ACM SIGCOMM Workshop on Future Humancentric Multimedia Networking (FhMN '13)*, ACM, August 2013, pp. 15-20.
- [3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69-74, 2008.
- [4] A. Valdivieso, L. Barona, and L. Villalba, "Evolution and Challenges of Software Defined Networking," in *Proc. the 2013 Workshop on Software Defined Networks for Future Networks and Services*, November 2013, pp. 61-67.
- [5] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121-132, 2004.
- [6] M. Kaufmann, M. Pioro, and D. Medhi, *Flow, and Capacity Design in Communication and Computer Networks*, 1st ed., 2004, pp. 307-349.
- [7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed., the MIT Press, 2009, pp. 414-450.



Kuei-Hong Lin is currently pursuing the master degree at National Chiao Tung University in communications engineering. Besides, he is also a software engineer of the Industrial Technology Research Institute of Taiwan, R.O.C. His research interests include communications signal processing, wireless channel modeling, FPGA-based software defined radio implementation, IPTV, adaptive streaming, SDN, and data mining.



Kuo-Huang Chung received his master degree in computer science from National Sun-Yat Sen University, in 2002. He is currently pursuing the PhD degree at National Chiao Tung University in Computer Science. Besides, he is also a software engineer of the Industrial Technology Research Institute of Taiwan, R.O.C. His research interests include voice signal processing, video compression, SDN, and IPTV.