# Joint Faces Scheduling and Bitrate Switching for Dynamic Adaptive Streaming over NDN Based on Stochastic Optimization

Xi Wei, Xiaobin Tan, Xiangyang Wu, Lei Xu
*Laboratory for Future Networks, University of Science and Technology of China, Hefei 230027, P.R. China*
E-mail: wx33921@mail.ustc.edu.cn, xbtan@ustc.edu.cn, wuxy13@mail.ustc.edu.cn, 4379906252@qq.com

***Abstract*—With the fast increase of video traffic transmitting over the Internet, dynamic adaptive streaming (DAS) has emerged as a conventional way for video streaming distribution. At the same time, Named Data Network (NDN) has been proposed to deal with various problems existed in the Internet. However, existing researches in DAS over NDN did not pay much attention to multihoming capability of NDN. Consumers who request data using multiple faces can aggregate bandwidth of different links which results in choosing higher layers of the video, increasing the quality of video requested. Multiple faces streaming can also smoothen the throughput seen by the client, leading to less fluctuation of the layer switching. Moreover, applying multiple faces can decrease the risk of port failure, increase reliability of the system. In this paper, we propose a dynamic adaptive algorithm of joint faces scheduling and bitrate switching for video streaming request over NDN aiming at optimizing long-term quality of Quality of Experience (QoE) and the cost of consumers under a constraint of playback smoothness and fluctuation. The proposed algorithm is based on Lyapunov Optimization which does not need priori knowledge of the dynamic network state information for each port. Both our theoretical analysis as well as experiments show that this algorithm is effective.**

***Index Terms*—Named Data Networking, Scalable Video Streaming, Stochastic Optimization, Multihoming**

## I. Introduction

There is no doubt that video streaming accounts the main part of traffic trends over the Internet. Applications offer the clients with video content, like YouTube, Youku and so on, have been popular all over the world. How to improving the utilization efficiency of the network while ensuring the video viewing experience naturally becomes a hot research topic. However, the design of the Internet was not proposed to serve such kind of streaming. In order to transmit data with large bitrates which the IP host-centric communication model could not deal with efficiently, the Information Centric Network (ICN) architecture has been proposed [1]. ICN pays attention to the data content rather than its position in the network, because the users just care about the content they get, ignoring where it comes from. The Named Data Network (NDN) is one of the future network architecture which obeys the philosophy of ICN. In addition, NDN provides services like in-network cache, multihoming and so on which can greatly improve the efficiency of video transmission.

At the same time, Dynamic Adaptive Streaming (DAS) [2] and H.264/SVC [3] has emerged as a standard method to transmit multimedia streaming. In DAS systems [4], video content is fragmented into segments with length of several seconds. Then every video fragment is layered into one basic layer and several enhanced layers. Meanwhile, the information about coding, layers, bitrates and so on is recorded in the Multimedia Presentation Description (MPD) which could act as resource list indexes for the client to reference. Unlike traditional streaming transmission, the video rate is not controlled by DAS directly [5]. DAS is designed client centric. First of all, the client shall request for MPD file to get information about the multimedia. Then the client dynamic adapts the video rate decisions, considering both the history or current state of the network (such as recently achieved throughput and delay) and the buffer state.

Recently, there is a study [6] showing that scalable video codes has a better performance than single layer codes. So our work adapts video encoded by H.264/SVC rather than H.264/AVC. The SVC is implemented by hierarchical coding, consisting of one Base Layer and several optional Enhancement Layers. Among them, the Base Layer is the basic for decoding. When the network is in good condition, in addition to the Base Layer, the client shall apply for Enhancement Layers as more as he can, expecting to get better quality of the video. In addition, there are two main reasons for us to adapt SVC. Firstly, the incremental decoding feature of SVC can cooperate well with the multihoming characteristics of NDN. Secondly, the layered coding of SVC agrees with the cache characteristics of NDN.

Of course, there have been a number of works carried out to achieve high video streaming quality. Jiang et al. [7] proposed FESTIVE to solve the problem of fairness, efficiency, and stability. ELASTIC [8] focus on the unfairness caused by the ON-OFF state of DAS. ELASTIC can avoid ON-OFF traffic pattern and take a good use of the bottleneck resource. Spiteri et al. [9] proposed BOLA as a solution to the utility maximization problem, using Lyapunov optimization techniques to minimize the risk of playback interruption and maximize video quality. This algorithm can provide a theoretical guarantee on the result, assuming that the buffer state can give all the infor-

This work is supported by The National Nature Science Foundation of China under Grant (61673360) and ZTE Industry-Academia Research Cooperation Funds.

**978-1-5386-4870-4/18/$31.00 ©2018 IEEE**

mation about the bandwidth. Considering the efficient use of feedback from the bandwidth and buffer state, Model Predict Control (MPC) based algorithm was proposed by Yin et al. [10], [11]. Experiments have showed that MPC can achieve a relative higher QoE for video streaming. However this method is sensitive to the accuracy of bandwidth prediction.

Nowadays, smart phones and laptops are becoming more and more popular in our daily life. Usually, they are able to access to multiple networks with 4G cellularthe Ethernet and WiFi. This feature drive us to explore approaches to make a good use of multihoming. However, the algorithms mentioned above did not focus on that. At the same time, multihoming is also a key advantage of NDN over traditional IP network. There have been many papers discussing dynamic adaptive streaming over multiple access networks, like [12], [13]. But most of these researches are based on the traditional IP network which can predict the bandwidth easier and more accurately. In this paper, we take notice that using multiple faces to access the client with the sources can both increase and smoothen the throughput seen by the client, enabling the adaptive algorithm to choose higher representations. Also, multiple faces make the system more stable, decreasing the risk of stalling. We take Lyapunov Optimization as the theory basic of our algorithm, because the NDN in-network cache usually results in the degradation of bandwidth prediction which is crucial for most of the adaptive approaches like MPC, FESTIVE and so on. Lyapunov Optimization theory can make decisions without priori knowledge of the network, making the bandwidth prediction just a factor for our choice. Essentially, the video streaming problem is a problem of queuing theory which can be well down by Lyapunov Optimization theory. Additionally, it provides a theoretical guarantee for the stability. The main contributions of this paper include the following aspects:

- We systematically model the problem of DAS in NDN when client applies for video streaming using multiple faces. Based on that, we regard the adaptive streaming as an optimization problem.
- We figure out that this optimization problem can be solved by Lyapunov Optimization theory, and proposed an excellent algorithm which can joint faces scheduling and bitrate switching, taking a full use of multiple faces and optimizing the QoE of dynamic adaptive scalable video streaming at the same time.
- Finally, we take experiments. The results show that our algorithm can achieve better performance than buffer-based and throughput-based algorithm in NDN. Additionally, it is worthy to notice that when one face is shut down, the dynamic adaptive using of multiple faces can avoid interruption effectively.

The rest of this paper is organized as follows. In section II, we discuss the model of this optimization problem, taking QoE of video combined with economy cost as the goal, constrained to buffer level and layer fluctuation. Then we propose the algorithm with details in section III. In section IV, the experiments are implemented to evaluate our solution, compared with some traditional approaches. Finally, we draw the conclusions in section V.

## II. System Model

In this section, we briefly show the overview of DAS in NDN. Then the model is introduced in details. At the last, we get the problem formulation.

### A. System Architecture

As show in the Fig. 1, each client and server both can connect to one or more networks. The NDN network is responsible for transmitting interest packets and data packets. When a node in the network receives an interest from the former node, it shall return the video segment, if there were

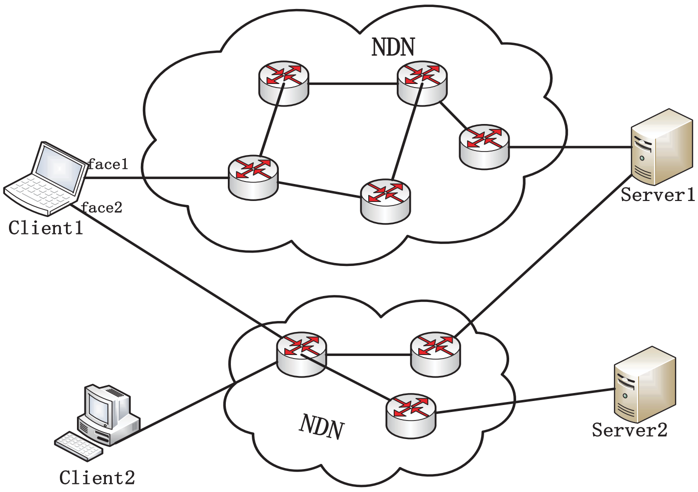


Fig. 1. System Architecture for DAS over NDN.

duplication in the Content Store, or the router should forward the interest to the next hop. On the one hand, servers just code the video data into the proper format which match the DAS and SVC standard. When servers receive the interests requesting for the stored multimedia, they need to hand out these data. On the other hand, the clients request for MPD firstly. According to the information contained in MPD and the state of the network, the clients can call the algorithm to decide which layer of video to request for and which face to transmit the interest packet and receive the data.

### B. System Model

As mentioned above, the features of SVC, like incremental decoding and layered encoding, can cooperate well with the multihoming and in-network cache characteristics of NDN. So we take the video encoded by H.264/SVC rather than AVC. It is assumed that there are L layers in total for each video segment, including one base layer and $\mathrm{L}-1$ optional enhancement layers. Let $R_1(\mathrm{t})$ and $R_2(\mathrm{t})$ represents the network bandwidth of $face1$ and $face2$ respectively at the moment $t$. They are estimated by former received data size and the waiting time. We suppose that the bandwidth of each face obeys the independent identically distributed model. The client requests for video segment $\mathrm{k}$ at time $\mathrm{t}_k$. It is supposed that the video segment with the same layer is relatively uniform and the network condition does not fluctuate wildly. So we can take the dynamic adaptive streaming problem as a queuing problem with fixed time slot T. Then we set $\mathrm{Q}(\mathrm{t})$ to be the buffer level of the player at moment $t$, with length of time as the unit. Certainly there is a boundary for buffer. That is $0 \leq Q(t) \leq Q_{\max}$. As the MPD contains all the information about the video and the playback is continuous, the client can know frame rate f for each segment. It is reasonable to say that Tf is the amount of video content consumed in every time slot T. Next, we let $\mathrm{F}(\mathrm{t_k})$ indicates the amount of content that client requested and received form the network in each time slot. According to the characteristic of SVC, we can estimate $\mathrm{F}(\mathrm{t_k})$ as follows:

$$\mathrm{E}\{\mathrm{F}(\mathrm{t_k})\} = \frac{\mathrm{R_x}(\mathrm{t_k})\,\mathrm{T}}{\sum_{\mathrm{i}=1}^{\mathrm{l}(\mathrm{t_k})} \mathrm{d_{ik}}}\mathrm{f} \tag{1}$$

where $\mathrm{l}(\mathrm{t_k})$ is the selected layer of segment k , and $\mathrm{d_{ik}}$ is the size of each layer. The selected face x to transmit the coming interest and data packet also can also influence the throughput $\mathrm{R_x}(\mathrm{t_k})$ seen by the client, causing the decision module to request different layers of the segment. Moreover the relationship between $Q(\mathrm{t})$ and $F(\mathrm{t})$ can be expressed as:

$$\mathrm{Q}(\mathrm{t_{k+1}}) = (\mathrm{Q}(\mathrm{t_k}) - \mathrm{Tf})^{+} + \mathrm{F}(\mathrm{t_k}) \tag{2}$$

where $(\mathrm{a})^{+}$ means $\max(\mathrm{a}, 0)$.

In this paper, we take QoE combined with the economy cost as the goal. Generally, average quality and quality jitter can have major impact on QoE. Therefore based on Lyapunov Optimization theory, we constrain the fluctuation of layers and buffer level. We measure the layer fluctuation in a slot with $\mathrm{S}(\mathrm{t_k})$ which is the difference between the number of requested layers in two consecutive time slots, $\mathrm{S}(\mathrm{t_k}) = |\mathrm{l}(\mathrm{t_k}) - \mathrm{l}(\mathrm{t_{k-1}})|$. And we use $\mathrm{N}(\mathrm{t_k})$ to measure the quality of video segment.

Considering the long term average of quality, we define the average of $\mathrm{Q}(\mathrm{t})$, $\mathrm{S}(\mathrm{t})$, $\mathrm{N}(\mathrm{t})$as :

$$\begin{aligned}
\overline{Q} &\triangleq \lim_{\mathrm{t}\to\infty} \tfrac{1}{t} \sum_{\tau=0}^{t-T} E\{Q(\tau)\} \\
\overline{S} &\triangleq \lim_{\mathrm{t}\to\infty} \tfrac{1}{t} \sum_{\tau=0}^{t-T} E\{S(\tau)\} \\
\overline{N} &\triangleq \lim_{\mathrm{t}\to\infty} \tfrac{1}{t} \sum_{\tau=0}^{t-T} E\{N(\mathrm{l}(\tau))\}
\end{aligned} \tag{3}$$

where $\mathrm{N}(\mathrm{l}(\tau))$ indicates the instantaneous video quality, in moment $\tau$ when the $\mathrm{l}(\tau)$ layers are received by the client.

Overestimation of the bandwidth makes the client to choose very high layers, leading to interruption which is quite harmful for users experience. So we give out a limitation on the buffer level, $\overline{Q} \geq \theta$. Similarly, the constraint for the layer switching is defined as $\overline{S} \leq \eta$.

In order to weaken the bandwidth fluctuation of a single face, we adapt a face scheduling strategy at the same time. We take the economy index into consideration. Since the cost for a unit flow can be variable for different access to the network. Our goal is to maximize the QoE while paying for lowest cost. Its supposed face x needs to pay $\omega(x)$ for a unit flow.

### C. Problem Formulation

We consider the QoE combined with the economy cost as the optimization objective, limited to the buffer constraint and fluctuation constraint. Based on the system model presented previously, we get the problem formulation:

$$\begin{aligned}
Maximize \quad \overline{G} = &\left(\alpha\overline{N} - \beta\left|Q_{\max} - \overline{Q}\right| - \gamma\overline{S}\right) \\
&- \mu C(l(t))\,\omega(x)
\end{aligned} \tag{4}$$

$$Subject\ \ to \qquad \begin{aligned} \overline{Q} \geq \theta \\ \overline{S} \leq \eta \end{aligned} \tag{5}$$

$$Variable \qquad \begin{aligned} l(t_k) &\in \{1, 2, ..., L\} \qquad \forall t_k \\ x(t_k) &\in \{1, 2\} \qquad \forall t_k \end{aligned} \tag{6}$$

where $\alpha, \beta, \gamma, \mu$ are parameters for normalization, and $\overline{G}$ measures the global optimization goal from the start to current decision moment. From the objective function we can see that higher average requested layers, lower video switching, lower risk of stalling and less cost on economy lead to better performance. It is worth to note that the switch between faces can both influence the throughput and the economy cost. At last we suppose that if every step we take is the most optimal, we can get the most optimal result.

## III. Analysis and Algorithm

In this section, we present our strategy and algorithm. Besides, the analysis is give out to show why the approach is effective.

### A. Face Schedule and Layer Switchig Strategy

First of all, we construct two virtual queues $H(t)$ and $Z(t)$. The update equation of the virtual queue can be written as:

$$\mathrm{H}(\mathrm{t}+1) = (\mathrm{H}(\mathrm{t}) - \mathrm{Q}(\mathrm{t}+1))^{+} + \theta \tag{7}$$

$$\mathrm{Z}(\mathrm{t}+1) = (\mathrm{Z}(\mathrm{t}) - \eta)^{+} + \mathrm{S}(\mathrm{t}) \tag{8}$$

The virtual queues $H(t)$ and $Z(t)$ are constructed to facilitate the use of Lyapunov drift and Lyapunov Optimization theory. According to the powerful tool of Lyapunov drift and Lyapunov Optimization, the time average queue constraint and layer switching constraint could be transferred to the stability problem of the virtual queues. That is to say, mean rate stability of $H(t)$ and $Z(t)$ indicates that time average constraints for buffer level and layer switching are satisfied. Next, we get the general queue length vector $\Theta(t)$:

$$\Theta(t) \triangleq \{H(t), Z(t)\} \tag{9}$$

Then we define a quadratic Lyapunov function as:

$$L(\Theta(t)) = \frac{1}{2}\left[H(t)^2 + Z(t)^2\right] \tag{10}$$

So, we get the conditional Lyapunov drift for the slot $t$:

$$\Delta\left(\Theta\left(t\right)\right)\triangleq E\left\{L\left(\Theta\left(t{+}1\right)\right)-L\left(\Theta\left(t\right)\right)|\Theta\left(t\right)\right\} \tag{11}$$

where the expectation depends on the layer switching and face scheduling policy, with respect to the random channel throughput (the choice of face affects the throughput, which indirectly influence the selection of the layers). As Lyapunov drift theory suggests, when we minimize a bound on $\Delta\left(\Theta\left(t\right)\right)$, virtual queues $H\left(t\right)$ and $Z\left(t\right)$ would be stable, leading to the constraints for buffer level and layer switching satisfied. However, we consider the optimization goal at the same time. So instead of making a switching decision on minimize a bound on (11), we aim at minimizing a bound on drift-plus penalty as follows:

$$\Delta\left(\Theta\left(t\right)\right)-VE\left\{G\left(l\left(t\right),x\right)|\Theta\left(t\right)\right\} \tag{12}$$

where $V\geq 0$ is an importance weight on the playback quality maximization, denotes the weight of average mass in the whole QoE. And the larger the value for $V$ is, the higher the video quality is in QoE compared with the constraints mentioned above. The relation is $O\left(V,\frac{1}{V}\right)$. Following, we shall show a bound on this formula. According to the update equations of the virtual queues, we have:

$$\begin{aligned} H(t+1)^2 &\leq \left(H\left(t\right)-Q\left(t+1\right)\right)^2+\theta^2+2H\left(t\right)\theta \\ &= H(t)^2+Q\left(t+1\right)+\theta^2+2H\left(t\right)\left(\theta-Q\left(t+1\right)\right) \end{aligned} \tag{13}$$

$$Z(t+1)^2\leq Z(t)^2+\eta^2+S(t)^2+2Z\left(t\right)\left(S\left(t\right)-\eta\right) \tag{14}$$

Plugging (13), (14) and (1) into (12) yields:

$$\begin{aligned} \Delta\left(\Theta\left(t\right)\right)-VE\left\{G\left(l\left(t\right),x\right)|\Theta\left(t\right)\right\} &\leq K+K_1\left(t\right) \\ &+E\left\{\left(Q\left(t\right)-H\left(t\right)\right)\left(F\left(t\right)-f\right)|\Theta\left(t\right)\right\} \\ &+E\left\{Z\left(t\right)\left(S\left(t\right)-\eta\right)|\Theta\left(t\right)\right\} \\ &-VE\left\{G\left(l\left(t\right),x\left(t\right)\right)|\Theta\left(t\right)\right\} \end{aligned} \tag{15}$$

where the positive constant $K$ and $K_1$ are defined as:

$$K=\frac{1}{2}\left[f^2+{F_{\max}}^2+\theta^2+\eta^2+{S_{\max}}^2\right] \tag{16}$$

$$K_1\left(t\right)=\frac{1}{2}Q(t)^2+H\left(t\right)\left(\theta-Q\left(t\right)\right) \tag{17}$$

Both of them are known as constant at the time slot $t$ , since the values of $H\left(t\right)$ and $Q\left(t\right)$ could be calculated. So the problem could be simplified as minimizing the bound given in the right-hand-side of (15). According to the drift-plus-penalty framework in the context of Lyapunov Optimization, the algorithm for face scheduling and layer switching can be derived as below.

At the beginning of each time slot, the player obtains the buffer level $Q\left(\mathrm{t}_k\right)$. Then it can calculate the value of the virtual queues $H\left(\mathrm{t}_k\right)$ and $Z\left(\mathrm{t}_k\right)$. Also it is easy to get the layer fluctuation $S\left(\mathrm{t}_k\right)$. Next we can choose the layer $l\left(\mathrm{t}_k\right)$ and face $x\left(\mathrm{t}_k\right)$ to minimize the following formula:

$$\begin{aligned} \left(Q\left(t_k\right)-H\left(t_k\right)\right)\left(F\left(t_k\right)-f\right)&+Z\left(t_k\right)\left(S\left(t_k\right)-\eta\right) \\ &-VG\left(l\left(t_k\right),x\left(t_k\right)\right) \end{aligned} \tag{18}$$

Similarly $G\left(l\left(t_k\right),x\left(t_k\right)\right)$ is defined as:

$$\begin{aligned} \mathrm{G}\left(l\left(t_k\right),x\left(t_k\right)\right) &\triangleq \left(\alpha N-\beta\left|Q_{\max}-Q\right|-\gamma S\right) \\ &-\mu C\left(l\left(t_k\right)\right)\left(\omega_1+\omega_2\left(1-x\left(t_k\right)\right)\right) \end{aligned} \tag{19}$$

The term $\left(Q\left(t_k\right)-H\left(t_k\right)\right)\left(F\left(t_k\right)-f\right)$ may be viewed as a queuing-weight expected mismatch between the expected frame arrival rate $F\left(t_k\right)$ and the playback rate $f$. Similarly, the term $Z\left(t_k\right)\left(S\left(t_k\right)-\eta\right)$ is a virtual queuing weight difference between the current layer switch and its tolerance $\eta$.

## *B. Algorithm*

The architecture of the system is described in Fig. 2. When the player need to make a decision of which layer to request and which face to transfer the interest, it shall look up the buffer state and estimated throughput for different faces firstly. Then the player can calculate the middle variables and transfer them to the decision maker. Finally the decision module can iterate through all the available layers and faces, calculates the value of (19), and returns the combination of the layer and face which has the minimum value. The algorithm is shown

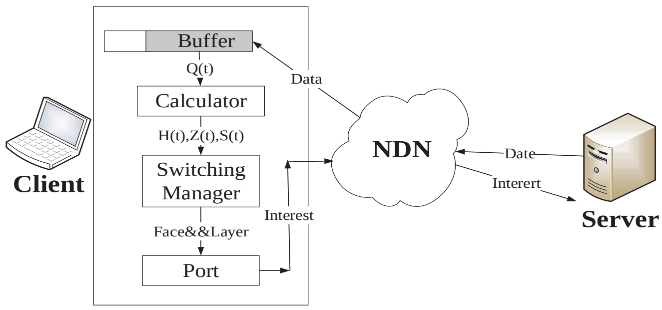


Fig. 2. System Architecture.

in Algorithm 1. The core part of our algorithm is to obtain and calculate all parameters (19) needed accurate online and make decision by finding the face and layer number, making the value of (19) minimum through iteration. Moreover, we also consider the extreme situations when the cache become empty for some reasons such as link disconnection, a surge in network traffic and so on. When it happens, we make an immediate end to the unfinished requirements for the current segment and make a decision to request the lowest layer of the next segment and the face with larger bandwidth immediately instead of using our layer switching strategy, improving the performance of our algorithm in extreme situations.

## IV. Evaluation

Our experimental system is built on the NDN DAS Platform (NDP). The topology is shown in Fig. 3. The client connects to the server with two independent faces. It is worth to note that in the actual scenario, the client can probably connect to several servers distributed in the network. This can be both beneficial to increasing the bandwidth seen by the client and reducing the load on each server. In the experiment, we adapt MPEG-DASH-compliant SVC-encoded multimedia Big Buck Bunny as the content. It is taken from the SVC dataset [14],

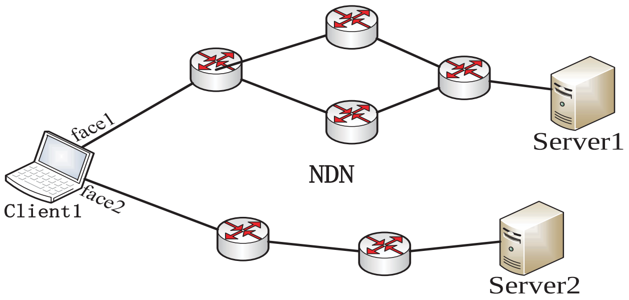


Fig. 3. Topology for Evaluation.

**Algorithm 1** dynamic face scheduling and layer switching strategy

**Input:** $IRep, segmentnumber$
**Output:** $l, x$
update $R_1, R_2$
**if** buffer is empty **then**
  set $l$ minimize, select $x$ with larger bandwidth
**end if**
**if** $representation$ for current segment is empty **then**
  obtain $Q, S$
  calculate $H, Z$
  initialize $layernum, layerForRep, minValue$
**end if**
**while** the layer number existed **do**
  iterate all the faces available
  find out the scheduled face $x$ and layer number $l$ maximizing the Lyapunov drift
**end while**
**return** $l, x$

divided into segments with size of two seconds [15]. Further, we apply tc-tool to control the bandwidth of different links.

### A. Results and Analysis

As we can see from the Fig. 4 and Fig. 5, it is obviously that the Lyapunov based adaptive video streaming switching strategy performs better than the traditional state-of-art algorithm (buffer based and rate based), achieving higher QoE and lower fluctuations between segments. The buffer based algorithm is designed to reduce the impact of inaccuracy estimation of the throughput in NDN, ignoring the state of the network. The rate based method focuses on the measured real-time bandwidth. However, it is usually difficult to estimate the bandwidth accurately, causing frequent switching of layers between consecutive segments. More importantly, the one which adapts dynamic schedule for multiple faces can get better result than the one using only one face. Because this method can aggregate the bandwidth of the two faces.

Form the Fig. 6, we can see that adaptive schedule for multiple faces can smoothen the throughput seen by the client. Applying the same adaptive video streaming switching strategy (the Lyapunov based), the client which dynamically schedules two faces with lower bandwidth can receive relatively higher layers of the video, performing better than the one only using a single face. And the switching between segments also decreased. The reason is that when two faces are used, the random fluctuation of the two ports are offset by each other.

In general, our algorithm superior to the traditional ones in two aspects. First of all, the Lyapunov Optimization gives a guarantee on math that this method can achieve near optimal solution. And the theory is effective in solving problem of queuing theory. Synthetically, the method considers the bandwidth, the buffer state and the QoE constraint through Lyapunov drift. Secondly, the proposed method takes advantage of multihoming which is supported by NDN naturally. Dynamic adaptive scheduling multiple faces on the client can aggregate bandwidth of different links, and smoothen the throughput seen by the client. This can be efficient for increasing the quality of multimedia and decreasing the fluctuations. What is more, our algorithm is computation friendly. It can work online while methods like Q-learning can not.

## V. Conclusion

In this paper, we propose an algorithm which joints faces scheduling and bitrate switching for dynamic adaptive streaming in NDN. Based on Lyapunov Optimization, this method does not need a priori knowledge of the network. Instead it takes use of the information contained in the buffer level and the estimated bandwidth. Taking advantage of multihoming supported by the NDN, our method dynamic schedules multiple faces, increasing the throughput seen by the client and decreasing the fluctuations. To evaluate the proposed algorithm in NDN, we compared it with some traditional state-of-art algorithms, the buffer based and rate based. The experiments are carried out on a real NDN plant. And the results show that the proposed algorithm performs well, improving the quality of the video and decreasing the fluctuations.

## References

[1] Sun, Yan Bin, Y. Zhang, and H. L. Zhang. "Survey of Research on Information-Centric Networking Architecture." Acta Electronica Sinica (2016).

[2] Sodagar, Iraj. "The MPEG-DASH Standard for Multimedia Streaming Over the Internet." IEEE Multimedia 18.4(2011):62-67.

[3] Schwarz H, Marpe D, Wiegand T. "Overview of the scalable video coding extension of the H. 264/AVC standard[J]." IEEE Transactions on circuits and systems for video technology, 2007, 17(9): 1103-1120.

[4] T. Stockhammer, Dynamic Adaptive Streaming over HTTP: Standards and Design Principles, in Proceedings of the Second Annual ACM Conference on Multimedia Systems, ser. MMSys 11. New York, NY, USA: ACM, 2011, pp. 133144.

[5] Kua, Jonathan, G. Armitage, and P. Branch. "A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP." IEEE Communications Surveys Tutorials 19.3(2017):1842-1866.

[6] Snchez, Yago, et al. "Improved caching for HTTP-based Video on Demand using Scalable Video Coding." Consumer Communications and NETWORKING Conference IEEE, 2011:595-599.

[7] J. Jiang, V. Sekar, and H. Zhang, Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE, in Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies, ser. CoNEXT 12. New York, NY, USA: ACM, 2012, pp. 97108.

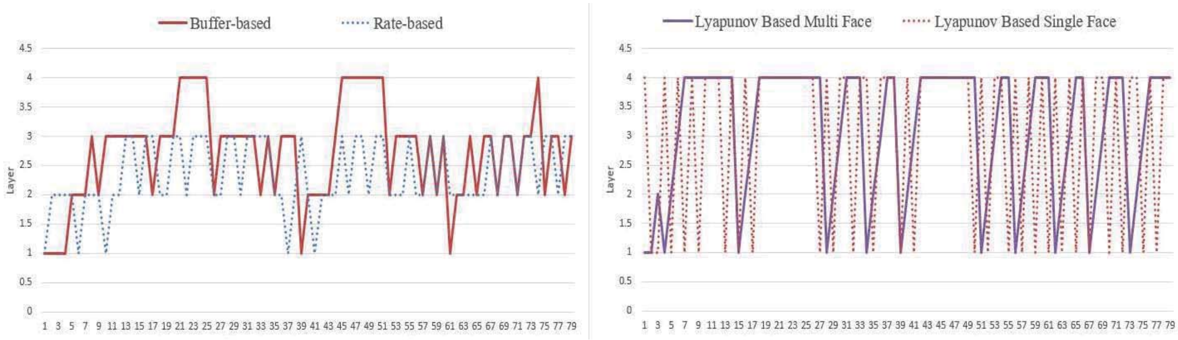


Fig. 4. Lyapunov Based V.S Traditional State-of-Art (QoE).

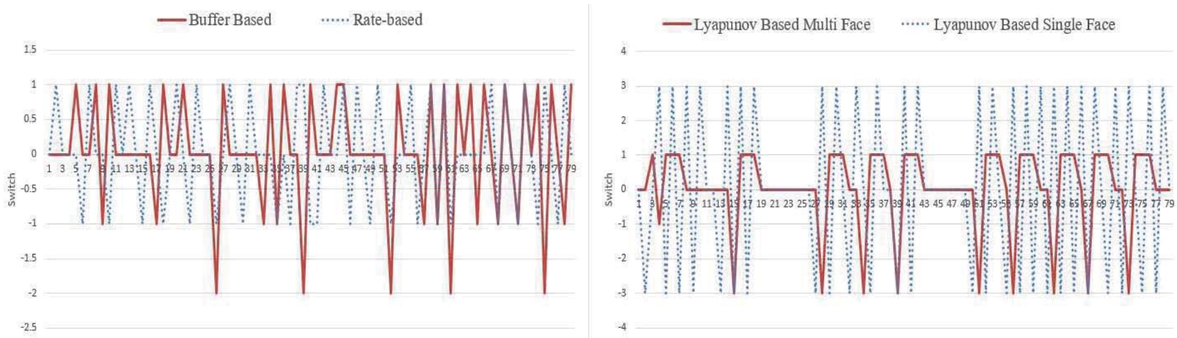


Fig. 5. Lyapunov Based V.S Traditional State-of-Art (Fluctuation).

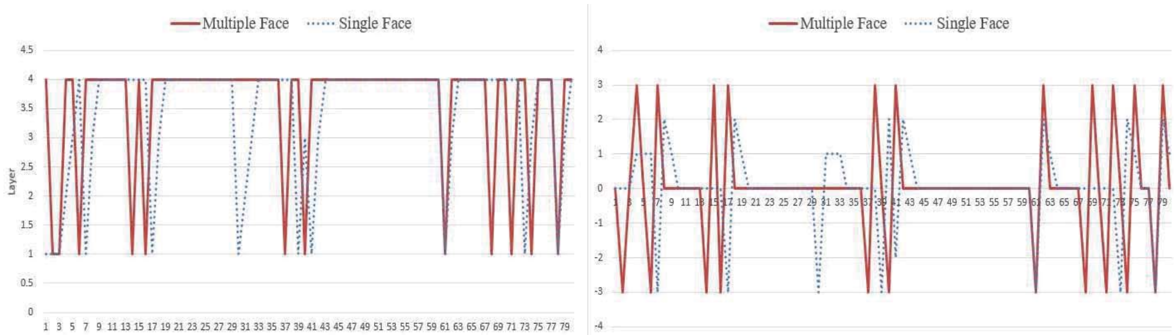


Fig. 6. Multiple Face V.S Single Face (Lyapunov Based).


[8] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, What Happens when HTTP Adaptive Streaming Players Compete for Bandwidth? in Proceedings of the 22Nd International Workshop on Network and Operating System Support for Digital Audio and Video, ser. NOSSDAV 12. New York, NY, USA: ACM, 2012, pp. 914.

[9] Spiteri, Kevin, R. Urgaonkar, and R. K. Sitaraman. "BOLA: Near-optimal bitrate adaptation for online videos." INFOCOM 2016 - the, IEEE International Conference on Computer Communications, IEEE IEEE, 2016:1-9.

[10] Yin, Xiaoqi, et al. "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP." ACM Conference on Special Interest Group on Data Communication ACM, 2015:325-338.

[11] Yin, Xiaoqi, V. Sekar, and B. Sinopoli. "Toward a Principled Framework to Design Dynamic Adaptive Streaming Algorithms over HTTP." ACM Workshop on Hot Topics in Networks ACM, 2014:9.

[12] Hsu, Cheng Hsin, et al. "Rate control and stream adaptation for scalable video streaming over multiple access networks." Packet Video Workshop IEEE, 2011:33-40.

[13] Schierl, Thomas, et al. "Using H.264/AVC-based Scalable Video Coding (SVC) for real time streaming in wireless IP networks." (2007):3455-3458.

[14] Kreuzberger, Christian, D. Posch, and H. Hellwagner. "A scalable video coding dataset and toolchain for dynamic adaptive streaming over HTTP." (2015):213-218.

[15] Lederer, Stefan, C. Mller, and C. Timmerer. "Dynamic adaptive streaming over HTTP dataset." (2012):89-94.