Hop-by-hop Adaptive Video Streaming in Content Centric Network

Zhengyang Liu
Beijing University of Posts and Telecommunications
Beijing, China
zhengyang-liu@hotmail.com

Yiran Wei
Beijing University of Posts and Telecommunications
Beijing, China
yiranwei314@gmail.com

Abstract—**To guarantee Quality of Experience (QoE) for video streaming services in a future Internet architecture, Content Centric Network (CCN), Dynamic Adaptive Streaming via HTTP (DASH) technology is used to deliver the proper video content according to the network situation. However, CCN enables a *host-to-content* communication model and has a universal caching design, which seriously decreases the performance of DASH over CCN. In this paper, we propose a hop-by-hop adaptive video streaming scheme (HAVS-CCN) to improve the performance of adaptive video streaming in CCN. HAVS-CCN is simple and applicable to be deployed on DASH over CCN. It directly adjusts video quality and solves network congestion at the bottleneck of transmission path when DASH inaccurately estimates the network throughout. Our scheme optimizes the hop-by-hop content transmission, which achieves video quality adaption and data packet flow control simultaneously. Simulation results, on small-scale networks and large-scale networks, reveal that DASH with HAVS-CCN scheme outperforms the original DASH over CCN, in terms of video playback quality and average delay.**

Keywords—**Content centric network, Adaptive video streaming, Flow control, Optimization.**

I. Introduction

Nowadays, the multimedia transmission occupies much of the Internet traffic. Internet video traffic will be 80 percent of all Internet traffic in 2019, up from 64 percent in 2014 [1]. That said, real-time video applications produce most of the Internet traffic. Today's Internet is based on host-to-host communication where IP network addresses are used to guide the location of desired content. However, consumers often care about the video content itself rather than building connections to the source location. To meet this growing trend of content delivery, a content oriented Internet architecture, Content Centric Networking (CCN, aka NDN) [2], has been proposed. Instead of referring to the physical location of video source, CCN identifies content by its name, which matches the multimedia data dissemination patterns. Besides, the universal caching design makes CCN more flexible and efficient for multimedia transmissions. Each CCN device caches pass-by data, which helps consumers access requested video content at a nearby position with less transmission delays. In this way, CCN has been an attractive solution for real-time applications like video streaming services.

To improve the QoE of video services, Dynamic Adaptive Streaming via HTTP (DASH) has become a hot issue both in academy and industry fields. For DASH, video services are provided as several bitrate versions. The basic operation is that a consumer requests a segment of video content using HTTP. Then the bitrate of requested video is adjusted according to current available bandwidth, which can be reflected by the transmission delay for last requested content. Deploying traditional DASH over CCN [3][4][5] is to use CCN for referencing and delivery. Due to in-network caching design, subsequent requests for same content can be served quicker in DASH over CCN. However, in this way the consumer side decisions of video quality face serious **bitrate oscillations** problem [6]. DASH decides the requested video bitrate based on the end-to-end throughput estimation. But in CCN, this consumer side adaptation algorithm may overestimate the available path bandwidth, which is due to the caches of requested segments along the path. This overestimation causes a repeated cycle that video quality decisions switch between high and low bitrate according to [6]. Moreover, it takes a delay for consumers in DASH to make quick response when network condition changes, especially for **network congestions** caused by the bursty traffic. If the network capacity cannot handle the transmission of high bitrate video flows, the consumer will resend interest requests for lower bitrate segments, which may cause playback pause happens at the consumer side.

Intuitively, the bitrate of a requested video segment should be decided at the consumer side considering end-to-end network capacity. However, our research reveals that this is not the case in CCN. Caches on a given path may have beneficial to other CCN routers in whole network. Thus, the evaluation and predication at the consumer side will be inaccurate. The existing researches [7][8] all focus on improving the accuracy of video quality prediction of end-to-end communication through HTTP, which cannot be applied to DASH over CCN. Therefore, a directly hop-by-hop **quality adjustment** scheme is needed to remedy the limitation of DASH over CCN. Moreover, to match the scalability feature of video streaming services [9], the Scalable Video Coding (SVC) extension of the H.264 video compression standard (H.264/SVC) [10] is used to encode video content into base and several enhancement layer content to support progressive transmission. Combined with the naming design of different layer data packets in CCN, the **flow control** and quality adjustment for video data transmission at the router level is feasible. The routers can proactively drop the less valuable SVC layer data packets when network congestion happens, instead of waiting for consumer decisions.

Our proposal is to provide flow control and quality adjustment for adaptive video streaming over CCN. There exists several principles that we should follow in our research. First, our scheme should be an add-on to traditional consumer decision based adaptive video streaming techniques in CCN. It can improve the consumer's QoE by adding hop-by-hop video quality adjustment and flow control. Second, our scheme should be simple and easy to be deployed for video services over CCN. Third, our scheme should achieve hop-by-hop QoE optimization, the principle of which can be applied to not only wired network, but also wireless environment.

With these principles, we propose HAVS-CCN, a proactive mechanism operated hop-by-hop at the router level, which can reduce the time delays for consumer decision process. The data packet transmission rate is directly controlled by setting a scheduling window based on network capacity. And HAVS-CCN can adaptively adjust video versions (different bitrates) with maximum utilization of link capacity by selecting the allowable number of enhancement layer data packets. The simulation results prove that HAVS-CCN can effectively improve the QoE for adaptive video streaming over CCN.

The **main contribution** for this paper is three-fold:

- We add a new component, Scheduling Data Queue (S-DQ), to the CCN router design, which provides directly controls on data packets for video transmission.
- We propose a novel scheme, HAVS-CCN, to adaptively adjust video quality and control traffic flows hop-by-hop. This scheme provides a new direction to improve the performance of DASH over CCN.
- We exhibits the effectiveness of our scheme based on the small scale and large scale simulations.

II. System model

In this section, we first introduce a new component of CCN router to help DASH with hop-by-hop optimization. Then, we show the naming design of SVC progressive transmission of video streaming services. This naming design makes it feasible to let CCN router schedule video content at a layer level.

A. Improved CCN router design

CCN uses unique content names for routing, instead of building end-to-end communication in IP network. The three main router components are Forwarding Information Base (FIB), Content Store (CS), and Pending Interest Table (PIT). The functionality of the original router components remains the same. Besides them, we introduce a novel *Scheduling Data Queue (SDQ)* to be an add-on to the original CCN content router design, as shown in Fig. 1.

The SDQ is a new component added to CCN router. It is used to schedule the outbound data flow for each interface. This component is designed for adaptive video streaming services. It records the information of all data packets that need to be transmitted through interfaces. The proposed scheduling scheme, HAVS-CCN, is operated on the SDQ component to control the data transmission at each router. For each interface, flow control can be achieved by limiting the transmission rate of the SDQ matched packets. And the quality adaption can be achieved by queueing up contents in SDQ based on their influences to the video playback. In the event that cached contents and the returned data packets want to transmit through a interface, they can be forwarded based on the SDQ entries towards downstream.

TABLE I
Scheduling Data Queue(SDQ)

Interface	ContentName	ContentPriority
0	/dst1/videos/baseball/s1/BL	0.89
0	/dst1/videos/baseball/s1/EL1	0.39
1	/dst1/videos/soccer/s1/BL	0.77
2	/dst1/videos/baseball/s1/BL	0.89

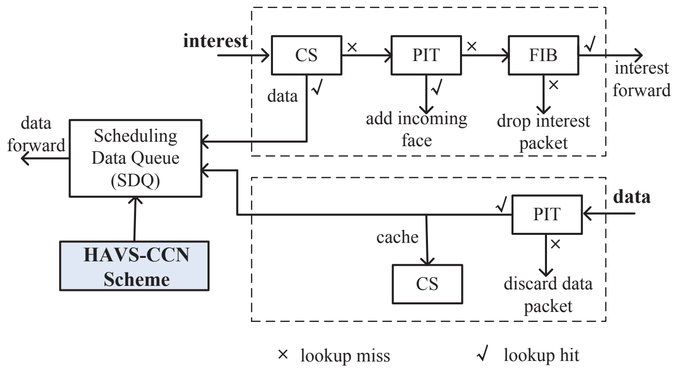

Fig. 1. Content router design with the new Scheduling Data Queue (SDQ).

To manage the scheduling operation of each interface, the SDQ entries follow a form of *face-to-content*, aims to record the contents need to be transmitted through a specific interface. This form is different from *content-to-face* entries in PIT and FIB, which guides the directions (interfaces) of data and interest packets to be forwarded to. Moreover, SDQ adds priorities for each entries to decide which content to be transmitted first and which to be discarded. The SDQ entry can be denoted as a 3-tuple entry:

$$(Interface, ContentName, ContentPriority)$$

Table. I is an example of SDQ. The computation process of the *ContentPriority* is detailed described in SecIII.B. The entries in SDQ are sorted in descending order based on content priorities for each interface. And contents with high priority will be given precedence in the scheduling of data packets.

B. Scalable Video Segmentation and Naming

We choose the H.264/SVC video compression standard as an example to show how progressive transmission operates in CCN. H.264/SVC is to support scalability in video coding and transmission. Each video content can be SVC encoded into one base layer content with essential information and enhancement

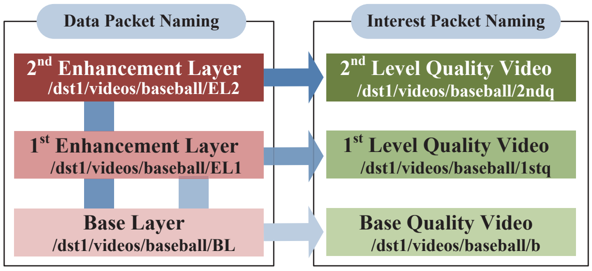

Fig. 2. H.264/SVC illustration.

layers content with information used for quality improvements. Since CCN identifies content by its name, a video segment can be separated to several layer packets to transmit. And at the consumer side, the related layer data packets are easy to be aggregated based on SVC standard, in order to play videos with specific quality defined by the enhancement layer packets. Fig. 2 shows data/interest packet naming design and decoding mechanism for the SVC data composed of one base layer and two enhancement layers. The naming of different level of data packet are classified as */BL* or */EL1*, */EL2*. The interest packet requests for different quality of video content are classified as */b*, */1stq* or */2ndq*.

To play an n^{th} level quality video, decoding is conducted using not only the base layer contents but also contents from the $\{1^{st}, 2^{nd}, \dots, n^{th}\}$ enhancement layers [11]. In CCN, each layer for a video segment is encapsulated independently for transmission. This SVC layer based segmentation design supports progressive transmission for video content, which will solve the overlapping transmission problem of contents with different bitrate versions. For example, when the bitrate of the requested segment is lower than the bitrate of a cached segment, the CCN cached node will send data packets for base layer and enhancement layers up to the requested level. And when facing network congestions, instead of requesting for a lower bitrate content, CCN nodes can directly drop the some enhancement layer packets and only transmit the base layer or low level enhancement layer packets to guarantee smooth playback.

III. BUILDING HAVS-CCN

In this section, we first state the hop-by-hop HAVS-CCN overview. Then we illustrate how HAVS-CCN guarantees the maximum QoE of video streaming service through optimizing the content values for limited transmission rate.

A. Design Overview

Let us illustrate a scenario to explain why and how HAVS-CCN operates on video streaming services in CCN. Suppose a system consists of N video segments, each is SVC encoded and cached with different bitrates among CCN. Live streaming services or video-on-demand services are provided to users by requesting video segments they want to play. And DASH over CCN system adaptively selects video qualities for each consumer to suit the network conditions. All video segments

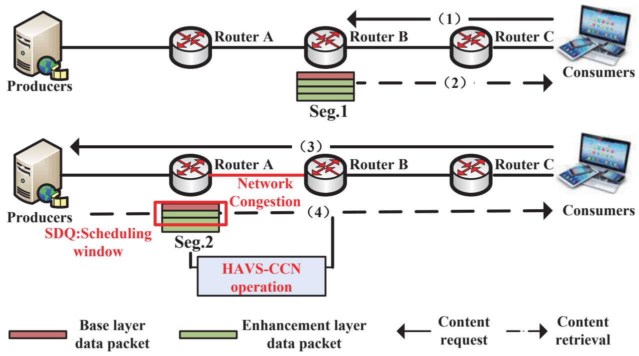

Fig. 3. HAVS-CCN overview.

are divided into one base layer packet and several enhancement layer packets to support progressive transmission.

HAVS-CCN is operated in DASH over CCN system as Fig. 3 shows. (1) A video segment request from the consumers will be broadcasted to other routers. This interest request includes the video quality requirement, which is predicated at the consumers side using DASH technology. (2) The $Router$ B has cached the requested video $Segment$ 1 and will transfer the content directly to consumers. The link capacity from $Router$ B to the consumers can support the transmission of this content. (3) Since this content request is satisfied quickly at a nearby cache, the consumers may be misled that the whole network condition is good. Therefore, consumers send requests for higher video quality of video $Segment$ 2 based on the high bandwidth estimated from the source of $Segment$ 1 to the consumers. (4) The link capacity between $Router$ A and $Router$ B can not hold the high quality of video content $Segment$ 2. Our scheme, HAVS-CCN, is operated to this situation, to provide a smooth but lower quality of video streaming services. The network congestion is alleviated since a scheduling window controls the data flow. And the QoE of consumers is guaranteed due to the base layer and low quality enhancement layers packets are transmitted successfully. Only the data packets for some enhancement layers video content, which the network capacity can not support, are dropped.

In original DASH over CCN, when facing network congestion in Fig. 3, the QoE of video streaming will decrease heavily. The reason is that traditional congestion control reacts to congestions only after data packets are randomly dropped without considering content meanings. This indifference congestion control with time delays do not take the advantages of H.264/SVC based progressive transmission for video streaming. For example, it may random drop the base layer data packets, which causes the related enhancement layer packets become nonsense at the consumers. Thus consumers need to resend interest requests for lower quality level video content, which causes long delays and poor user experience. To solve this problem, our scheme, HAVS-CCN, controls the data flows and adaptively adjusts video quality directly to the proper video quality according to network condition. To schedule the data transmission, new added router component

SDQ records the entries about all the data packets need to be transmitted. The scheduling window computed by Eq. (1) can control the transmission rate of the data packets by limiting the allowable sending quantity at any one time. At the same time, since SDQ entries are ordered based on their priorities, the allowable sending data packets match the SDQ entries from high priority to the low priority, which achieves the best quality adjustment of video services that the link capacity can support. Once a window size of data packets have been delivered, the SDQ entries with priorities are updated. Then a new round of data packets are delivered. More importantly, the scheduling window size can be dynamically adjusted based on the network capacity, which guarantees a HAVS-CCN can react to network changes quickly.

B. Hop-by-hop Adaptive Streaming in HAVS-CCN

HAVS-CCN optimizes the QoE of video streaming services hop-by-hop. To reduce content delivery delays and maintain low packet loss rate, it sets a **scheduling window** to limit the video transmission rate. To quickly react to dynamic network conditions, we directly control the data packets, instead of limiting the interest requests. The scheduling window of a interface k, which determines the allowable transmitting quantity of data packets for interface at any one time, is denoted as:

$$wnd_{scheduling}(k) = delay \cdot \alpha \cdot BW_{out}(k) \tag{1}$$

where $delay$ is the expected time for the data packet to be satisfied hop-by-hop. And $BW_{out}(k)$ is the available outbound bandwidth at interface k, which follows a well-known bandwidth model for video transport [12]. Since the flows of interest and data packets are in both directions of link, we denote $\alpha \cdot BW_{out}(k)$ as the available outbound bandwidth for data packets at interface k. α is the ratio that data packets flow to the whole outbound flow, which is an experience value related to the the average size ratio between interest and data packets in each direction. The scheduling window $wnd_{scheduling}(k)$ updates every time interval T to timely control traffic flow.

To adaptively adjust video quality with QoE optimization, it needs to push some selected video content to stuff the limited scheduling window. We set the concept of priority to measure the importance of each data packet, which is related to the type and the popularity of the content. Suppose $p_{i,seg,type}$ is the data packet for quality-defined $type$ of video segment seg from video content i.

Type-based priority. For H.264/SVC encoded adaptive video streaming, the low layer content has higher possibility to be referred compared with the high enhancement layer content. Owing to this subordinate features of the layers explained in Sec.II (B), the base layer content has higher priority than the enhancement layer. And the low quality enhancement layer content also has higher priority than high quality enhancement layer content. We denote

$$w_t(p_{i,seg,type}) = \frac{1}{\eta_{type}} \tag{2}$$

as the type-based priority for data packet $p_{i,seg,type}$, where η_{type} is

$$\eta_{type} = \begin{cases} 1, & type = base\ layer \\ i+1, & type = i^{th}\ enhancement\ layer \end{cases} \tag{3}$$

Popularity-based priority. A video content is requested more, this video is more popular and the priority will be higher. The reason is that transmitting popular contents can satisfy more users' demands, which improves the overall QoE experience. For an interface, the sum of the segments of one video content in SDQ entries can reflect the popularity of the content through this interface. We denote $\eta_{i,k}$ as the sum of SDQ entries for video segments belongs to video content i at the interface k, and SDQ_k as the SDQ queue size of the interface k. The popularity-based priority of data packet $p_{i,seg,type}$ for this hop can be expressed as:

$$w_p(p_{i,seg,type}, k) = \frac{\eta_{i,k}}{SDQ_k} \tag{4}$$

A data packet with higher type-based priority and higher popularity-based priority will have preference to be transmitted. We define the overall priority which jointly considers the data packet type and its popularity level as:

$$w(p_{i,seg,type}, k) = w_t(p_{i,seg,type}) \cdot w_p(p_{i,seg,type}, k) \tag{5}$$

The aim of the HAVS-CCN is to transmit high priority data packets first when the link capacity is limited. Let $c_{i,seg,type}$ decide to place or not to place the data packet $p_{i,seg,type}$ into the scheduling window. And $p_{size}(type)$ denotes the data packet size of different layer of video content based on H.264/SVC standard. Note that the scheduling strategy should avoid the starvation of base layer packets, since base layer content is essential to guarantee the smooth playback. We can formulate the data flow control and QoE oriented adaptive video streaming problem for a specific interface k under steady state as:

Objective:

$$maximize f = \sum_{i,seg,type} c_{i,seg,type} \cdot e^{w(p_{i,seg,type},k)} \tag{6}$$

Subject to:

$$\begin{aligned} \sum c_{i,seg,type} \cdot p_{size}(type) &= wnd_{scheduling}(k) \\ c_{i,seg,type} &\in \{0,1\} \\ c_{i,seg,type} &\neq 0, type = base \end{aligned} \tag{7}$$

The objective of this optimization is to maximize the priority of transmitted data packets , subject to limited link capacity and avoid the starvation of base layer content. Obviously, the content value equation $e^{w(p_{i,seg,type},k)}$ is a convex function and the value of weight $c_{i,seg,type}$ is either 1 or 0. The objective function Eq. (6) and the constraint function Eq. (7) are all convex functions. Therefore, the formulation is a convex optimization problem and is mathematically tractable. The detailed priority transmission algorithm of HAVS-CCN is presented in **Algorithm 1**.

Algorithm 1 HAVS-CCN Priority Transmission Algorithm

procedure OnWindowEmpty($interface$)
  $TxQueue \leftarrow SDQ[interface]$.
  if $sizeof(TxQueue) > wnd_{scheduing}$ **then**
    $c_{i,seg,type} \leftarrow$ Solver($p_{i,seg,type}$, $w_{i,seg,type,k}$)
    $TxQueue \leftarrow \{p_{i,seg,type} | c_{i,seg,type} \neq 0\}$
  end if
  Send($TxQueue$).
end procedure

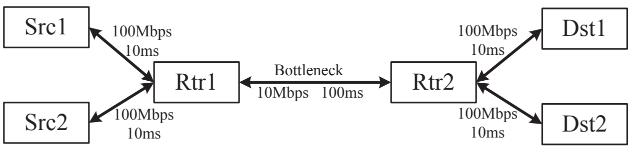

Fig. 4. Topology setup in small scale simulation

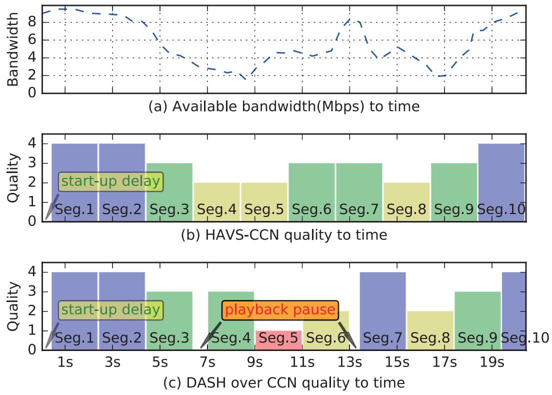

Fig. 5. The playback quality in HAVS-CCN and DASH over CCN to time.

In HAVS-CCN, each router does bandwidth estimation and maintains a scheduling window to control the data packet flows. Compared with the flow control achieved by limiting the rate of interest packets, HAVS-CCN directly works on data packets delivery, which reduces the delays and can make response to the bandwidth variation timely. Moreover, instead of selecting appropriate video bitrate according to the inaccurate estimation of network bandwidth from consumer to content provider, HAVS-CCN achieves adaptive video quality adjustment hop-by-hop based on the priority set for each data packet. Since the optimization process happens hop-by-hop, the previous hop output shall not affect the operation at current hop. Therefore, the amount of exchange information should not be large and the satisfied QoE for video streaming service can be guaranteed.

IV. Performance Evaluation

In this section, we present our evaluation results. To build our simulation platform, we adopt the CCN simulator MiniCCNx [13]. We use GLPK [14] as our linear programming solver. A H.264/SVC encoded video from University Klagenfurt [15] is used in all of our experiments. The demo video has 500 frames which are encoded into 11 segments. Each segments have a base layer and three enhancement layers.

We run our simulations on two different scales – a smaller 6-node topology and a much-larger realistic topology. We use the 6-node topology since it can clearly demonstrate the real-time response of HAVS-CCN when the available bandwidth changes. The larger ISP-like topology reflects how HAVS-CCN tackles congestions on a much more complex real-world network. Our simulation methodology resembles the fact that the effective bandwidth is limited and dynamically changed.

A. Small scale simulation

We build our initial experiment topology as Fig. 4 shows. Video consumers $Src1$ and $Src2$ request video content from the producers $Dst1$ and $Dst2$. They exchange interest and data packets through a shared link connection between $Rtr1$ and $Rtr2$. $Src1$ requests video by sending interest packets for each video segment to the upstream. And $Src2$ randomly requests contents to create a varying pressure on the link to simulate the unexpected bursty traffic. $Dst1$ and $Dst2$ provide all requested data packets when receiving interest requests. We precache $Segment$ 6 on $Rtr1$, to simulate CCN in-network caching operation. The available bandwidth of $Src1$ changes during the time is shown in Fig. 5(a). We compare the performance of HAVS-CCN with original DASH over CCN under this scenario.

The metric we choose to quantify a segment's playback quality is the number of playable layer packets $Src1$ receives. Fig. 5(b) and Fig. 5(c) show the comparison of playback qualities for two schemes. DASH over CCN and HAVS-CCN have similar performance when available bandwidth is constant (from 0.2s to 4s). However, when the link encounters a varying pressure from $Src2$, HAVS-CCN performs better than DASH over CCN. When the available bandwidth descends (from 4s to 8.5s), DASH over CCN has a playback pause of 0.6s (from 6.4s to 7s). This playback pause happens if the requested video quality is beyond the link capacity. The reason is that original DASH over CCN decides video quality at the consumer side, which causes the consumer lacks the timely reactions to the suddenly decrease in available bandwidth. We observe that HAVS-CCN is able to avoid this playback pause by adaptively adjusting the video quality level to $Quality$ 2 when faces the network bottleneck.

The playback pause from 13s to 13.2s in Fig. 5(c) is due to a different reason. Since the interest request for $segment$ 6 hits the cache at $Rtr1$, which misleads $Src1$ that the available bandwidth is sufficient for better quality playback. However, $Segment$ 7 is not available in $Rtr1$, and needs to be fetched from upstream. Thus requesting better quality ($Quality$ 4 for $Segment$ 7) on the link with limited available bandwidth causes the playback pause (from 13s to 13.4s). With $Segment$ 6 precached at $Rtr1$, HAVS-CCN can correctly adjusts the respond content data from $Quality$ 4 to $Quality$ 3 on $Rtr2$ through directly dropping some enhancement layer

TABLE II
LARGE SCALE SIMULATION LINK SETUP

LinkType	Delay	Bandwidth
Backbone-Backbone	10ms	100 Mbps
Gateway-BlackBone	10ms	40 Mbps
Gateway-Gateway	20ms	20 Mbps
Client-Gateway	50ms	5Mbps

data packets, which prevents the playback pause. It shows our HAVS-CCN can avoid the playback pause caused by bandwidth overestimation with the help of the HAVS-CCN's instant video quality adjustment on the intermediate routers.

B. Large scale simulation

In this section, the realistic large-scale topology we used is based on a modified version of Rockfuel's AT&T topology [16]. We extract one of the connected components containing 192 nodes from the original topology. All the nodes are classified into clients, gateways and backbones according to the degree. Nodes having degree less than four are classified as clients (95 nodes), nodes directly connected to the clients are classified as gateways (58 nodes), and remaining nodes are classified as backbones (39 nodes). We assign bandwidth and delay to the links based on their type (Table. II). Each backbone node and gateway node are installed with a video streaming server application which responds to the incoming interests with the corresponding data packets. The consumers are placed only at clients nodes. Each consumer is persistently requesting video segments by sending interest packets of the testing video to upstream.

1) Average Quality: We define a consumer's average quality as the sum of each received segment's playable bitrate divided by the total number of segments the consumer requests. Fig. 6 shows the average quality against the throughput in 20s' duration. The consumers with high throughput means their link to the upstream is sufficient for data delivery, while the consumers with low throughput means their link to upstream is congested. Both schemes perform well on the high throughput consumers, indicating that video segments with high bitrate are mostly displayed. This is because high video quality packets can be successfully transmitted in unobstructed links. Consumer nodes with throughput less than 2.5Mbps have an average quality around 50kbps to 1000kbps under DASH over CCN. While HAVS-CCN outperforms the original DASH over CCN with a average quality around 50kbps to 2400kbps under the same throughput. Due to the timely quality adjustment, HAVS-CCN can provide better consumer video playback quality and higher link utilization than original DASH over CCN under crowded networks.

2) Average Delay: Since a base layer is enough to provide smooth video playback, the delay that a consumer experienced can be defined as the time difference from sending interest packet request to receiving the base layer packet. The average delay for a consumer is the total delays divided by its total

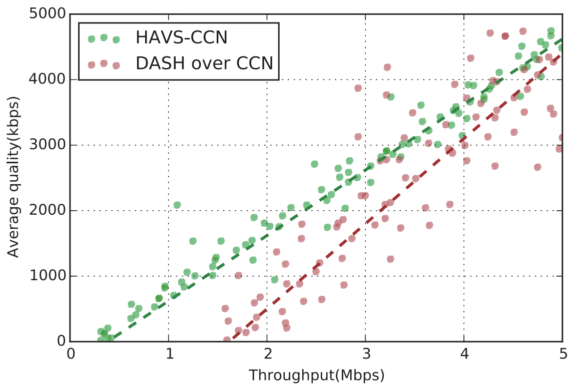

Fig. 6. A scatter figure of average quality to consumer throughput

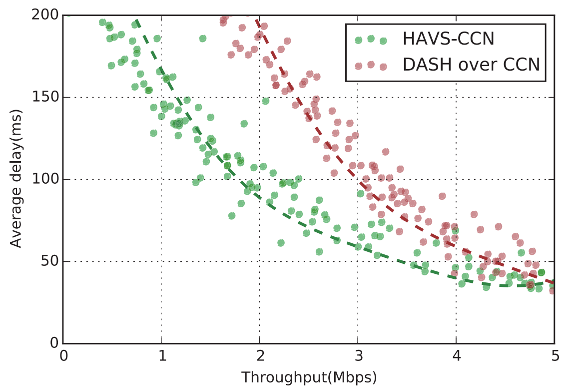

Fig. 7. A scatter figure of average delay to consumer throughput

number of interest packets. Consumers with higher average delays are easier to suffer from playback pause. Fig. 7 illustrates the average delay against consumer throughput in 20s' duration. We observe that a consumer's average delay increases as the throughput decreases. The reason is that the consumer with lower throughput means its link to upstream is more crowded. The crowded link leads to high queueing delay, frequent packet loss and retransmission, which results in high average delays. Both schemes can guarantee the average delay is less than 80ms when the consumer's throughput is higher than 4Mbps. Considering the low throughput situation, original DASH over CCN cannot provide a tolerable average delay(under 200ms) for the consumer whose throughput is lower than 2Mbps. While HAVS-CCN keeps the average delay under 200ms for the consumer with throughput between 1Mbps - 2Mbps. It reflects that by deploying our HAVS-CCN scheme, consumers in low throughput can still control the delay at a acceptable range. Compared with DASH over CCN, HAVS-CCN reduces much more delays at the low throughput situation, thanks to its instant quality adjustment.

V. CONCLUSION

This paper aims to improve the performance for DASH over CCN. The challenges in this setting are: 1) reducing the negative effect of inaccurate bitrate predication for DASH over CCN. 2) providing flow controls at the router level. We propose a hop-by-hop solution, HAVS-CCN, to achieve adaptively adjustment of video quality and flow control at the bottleneck of transmission path. A novel SDQ component is added to CCN router which records and queues data packets for each interface. HAVS-CCN also values the playback importance and popularity of each data packet to choose proper SVC layer content. Therefore, the video quality has got a second chance to be adjusted during the transmission path, instead of resending interest request from consumer side with long playback delays. Simulation results demonstrate how DASH with HAVS-CCN scheme outperforms the original DASH over CCN on playback quality and average delay. In the future, we will mathematically analyse the priority weights setting for different H.264/SVC layer data packets. The weights we use in this paper mostly come from the experience. The optimization of priority weights definition can surely improve the overall QoE of video services.

REFERENCES

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2014-2019," CISCO, Tech. Rep., May 2015.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. P. ans N. Briggs, and R. Braynard, "Networking named content," in *Proc. ACM International conference on emergine networking experiments and technologies (CoNEXT)*, Rome, Italy, Feb. 2009, pp. 1–12.

[3] S. Lederer, C. Mueller, B. Rainer, C. Timmerer, and H. Hellwagner, "Adaptive streaming over content centric networks in mobile networks using multiple links," in *Proc. IEEE International Conference on Communications (ICC) workshop on Communications*, Budapest, Hungary, Jun. 2013, pp. 677–681.

[4] Y. Liu, J. Geurts, J. C. Point, S. Lederer, B. Rainer, and C. Muller, "Dynamic adaptive streaming over ccn: A caching and overhead analysis," in *Proc. IEEE International Conference on Communications (ICC)*, Budapest, Hungary, Jun. 2013, pp. 3629–3633.

[5] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *Proc. ACM SIGCOMM on Special Interest Group on Data Communication*, London, British, Aug. 2015, pp. 325–338.

[6] D. H. Lee, C. Dovrolis, and A. C. Begen, "Caching in http adaptive streaming: Friend or foe?" in *Proc. ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV 2014*, Singapore, Singapore, Mar. 2014, pp. 31–36.

[7] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in http adaptive streaming," *IEEE/ACM Transactions on Networking*, vol. 99, p. 1, Sep. 2015.

[8] M. Claeys, S. Latre, J. Famaey, and F. D. Turck, "Design and evaluation of a self-learning http adaptive video streaming client," *IEEE Communications Letters*, vol. 18, pp. 716–719, Feb. 2014.

[9] X. Wang, T. T. Kwon, Y. Choi, H. Wang, and J. Liu, "Cloud-assisted adaptive video streaming and social-aware video prefetching for mobile users," *IEEE Wireless Communication*, vol. 20, pp. 72–7920, Jun. 2013.

[10] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h. 264/avc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[11] J. Lee, K. Lim, and C. Yoo, "Cache replacement strategies for scalable video streaming in ccn," in *Proc. Asia-Pacific Conference on Communications (APCC)*, Aug. 2013, pp. 184–189.

[12] A. Razzaq and A. Mehaoua, "Layered video transmission using wireless path diversity based on grey relational analysis," in *Proc. IEEE International Conference on Communications (ICC)*, Kyoto, Japan, Jun. 2011, pp. 1–6.

[13] C. Cabral, C. E. Rothenberg, and M. F. Magalhaes, "Mini-ccnx: fast prototyping for named data networking," in *Proc. ACM SIGCOMM workshop on Information-centric networking*, Hong Kong, China, Aug. 2013, pp. 33–34.

[14] Glpk (gnu linear programming kit). [Online]. Available: https://www.gnu.org/software/glpk/

[15] C. Kreuzberger, D. Posch, and H. Hellwagner, "A scalable video coding dataset and toolchain for dynamic adaptive streaming over http," in *Proc. ACM Multimedia Systems Conference (MMSys)*, Oregon, USA, Mar. 2015, pp. 213–218.

[16] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," in *ACM SIGCOMM Computer Communication Review*, Pittsburgh, PA, United states, Aug. 2002, pp. 133–145.