CLOUD-ASSISTED ADAPTIVE VIDEO STREAMING AND SOCIAL-AWARE VIDEO PREFETCHING FOR MOBILE USERS

XIAOFEI WANG, TED "TAEKYOUNG" KWON, AND YANGHEE CHOI, SEOUL NATIONAL UNIVERSITY
HAIYANG WANG, UNIVERSITY OF MINNESOTA AT DULUTH
JIANGCHUAN LIU, SIMON FRASER UNIVERSITY

ABSTRACT

While the demands of video streaming services over the mobile networks have been souring over these years, the wireless link capacity cannot practically keep up with the growing traffic load. The gap between the traffic demand and the link capacity, along with time-varying link conditions, results in poor service quality of video streaming services over the mobile networks, such as intermittent disruptions and long buffering delays. Leveraging the current cloud computing technology, we propose and discuss a framework to improve the quality of video services for mobile users, which includes two parts: cloud-assisted adaptive video streaming, and social-aware video prefetching. For each active mobile user, a private agent is constructed in the cloud center to adaptively adjust the video quality (bit rate) by the scalable video coding technique based on the feedback of link condition. Meanwhile, the online social network interactions among mobile users are monitored by the cloud-based agents, so that the videos that are shared among users will be effectively prefetched to mobile users in advance. The adaptability of the video streaming and the effectiveness of the social-aware prefetching supported by the cloud computing are evaluated based on a prototype implementation of the framework.

INTRODUCTION

Recently, due to the rapid development of mobile communication technology, more and more people are attracted to enjoy video streaming services on phones and tablets while moving in the cars, buses and trains [1]. Despite the desperate efforts of network operators to enhance the wireless link bandwidth (e.g., 3G/4G), the soaring video traffic demands from mobile users are rapidly overwhelming the wireless link capacity.

While accessing video streams via 3G/4G mobile networks, users often wait for long buffering delays (dozens of seconds) but still suffer from intermittent interruptions [2] due to the bandwidth variation and the link fluctuation which are caused by multi-path fading and user mobility. It is crucial to improve the Quality of Service (QoS) of mobile video streaming while utilizing the networking and computing resources efficiently.

Therefore researchers are re-thinking what the core QoS factors for mobile streaming are, and how we can design new techniques for better performance. In this article, we mainly focus on the two following questions as our main concerns:

- Can mobile users enjoy stable and continuous video streaming without disruptions?
- Can mobile users enjoy click-to-play video streaming with less buffering delays?

ADAPTIVE STREAMING WITH LITTLE DISRUPTION

Regarding the first question, recently there have been many related research studies focusing on two aspects:

Adaptability: Traditional video streaming techniques are mostly designed under the stable Internet links between servers and users, and hence they may perform poorly in mobile environments [2]; with a particular bit rate, if the link bandwidth varies much, the video streaming can be frequently disrupted due to packet losses. For a better QoS experience, the fluctuating link condition should be properly handled for providing "stable" video streaming service by which the video quality can adapt to the environment. To address this issue, we need to tune the video bit rate adapting to the time-varying available link capacity of each mobile user, based on his/her feedback of the link quality.

Scalability: Mobile video streaming services should also support a wide spectrum of mobile devices, with different screen resolutions, different power supplies, and different wireless accesses (e.g., Wi-Fi, 3G and 4G). However, traditional method to store multiple versions (with different bit rates) of the same video content may incur tremendous storage overhead while the volume

1536-1284/13/$25.00 © 2013 IEEE

of video content is skyrocketing globally. The Scalable Video Coding (SVC) technique of the H.264 AVC video compression standard [3] defines a base layer (BL) with multiple enhancement layers (ELs). By utilizing the SVC, a video can be decoded and displayed at the lowest quality if only the BL is delivered, while the more ELs are delivered, the better quality of the video stream can be achieved. Therefore a high diversity of mobile devices and link conditions can be covered scalably.

Most of current research proposals that seek to combine the video scalability and adaptability highly rely on the active control from the server side. That is, all mobile users individually report the transmission status periodically to the server, and the server predicts the available bandwidth and allocates proper video streams for each user. In this manner, the server takes over the substantial processing overhead while the number of users increases. Therefore, the server has to be able to afford a huge burst of concurrent user demands at peak hours, but waste much of the resource at idle hours, which is practically inefficient.

Thanks to the invention of the elastic cloud computing technique, the cloud computing is poised to flexibly provide scalable resources to content and service providers depending on the current user demands [7, 8]. For instance, as investigated in [3] and [5], cloud data centers can not only easily provide quality-assured real-time video services for a huge amount of Internet users, but also save computing and storage resources as well as energy when there are not so many active users, due to cloud's auto-scaling ability.

However, extending the cloud-based services to mobile environments requires more factors to consider: wireless link dynamics, user mobility, the limited computation and storage capacity, as well as the restricted power supply of mobile devices [9]. So many studies on mobile cloud computing techniques have proposed to virtualise personalized agents for servicing active mobile users intelligently, e.g., Cloudlet [10]. In the cloud, multiple instances of user agents can be maintained dynamically and efficiently depending on the time-varying user demands. Therefore, we are motivated to design a new framework of mobile adaptive video streaming by using virtual agents in the cloud.

Intelligent Streaming with Little Buffering Delay

Regarding the second question, in order to reduce the buffering delay, the intelligent prefetching, which pushes a part of (or the whole of) the video file into the mobile devices before user practically access, is highly needed. To this regard, a new trend to exploit the online Social Network Services (SNSs) for intelligent video prefetching is becoming more popular.

Because of the dramatic rise in the number of mobile users who participate in the SNSs, e.g., Facebook, Twitter, Sina Weibo and so on, a huge amount of video content is shared and spread rapidly and widely in the SNSs [1, 11]. There have been proposals to improve the quality of content delivery using SNSs, e.g., the study in [12]. So by investigating related studies on SNSs, we point out three following key points which can be utilized for intelligent pushing:

Social Impact: In the real world as well as the online SNSs, people mostly share interest content due to "word-of-mouth" propagation [13]. A user may probably watch a video that his/her friends have recommended [12]. Also he/she can follow famous accounts based on interests e.g., an official Facebook or Twitter account that shares the newest pop music videos, which are likely to be watched by the follower fans.

Locality: User relationships and interests in SNSs have significant homophily and locality properties [14]. Users are highly clustered by geographical regions and interests, which can be exploited for intelligent pushing as well as the optimal allocation of cloud resource.

Access Delay: Different mobile users have different patterns of accessing videos [11], which are per-user dependent mainly due to people's different life styles. Some users may access video streams frequently, while others may access with relatively longer intervals. The access delay provides us the hope to prefetch before the access of users.

From the above investigation, we explore the possibility to prefetch the video (fully or partially) to user devices in advance, based on their online interactions in SNSs while also considering their access delays; once the user clicks to watch the video, the video can instantly start playing without buffering. Therefore, the propose cloud agents have another role to track and learn the video-related social interactions among users in SNSs.

An Emerging Framework of Cloud-Assisted Mobile Video Services

In this article, we will investigate and address the emerging techniques for cloud-assisted mobile video services, which construct private agents for active mobile users in the cloud, in order to offer "non-terminating" and "non-buffering" mobile video streaming service. Private agents are elastically initiated and optimized in the cloud platform. Also the real-time SVC coding is done on the cloud side efficiently. The proposed framework leverages the SVC technique, and offers the scalable and adaptive streaming experiences by controlling the combination of video streams (layers) depending on the feedback of the fluctuating link quality from mobile users. Also based on the analysis of the SNS activities of mobile users, the proposed framework seeks to prefetch the video clips in advance from user's private agent to the local storage of the device. The strength of the social links between users and the history of various social activities can determine how much and which video will be prefetched.

We organize the article as follows: we first explain the cloud agent framework and give details of our proposal on adaptive video streaming and social-aware prefetching. Then the brief video delivery procedure is shown, and we discuss the performance evaluation of our proposal as well. The conclusion of the article is presented in the end.

Based on the analysis of the SNS activities of mobile users, the proposed framework seeks to prefetch the video clips in advance from user's private agent to the local storage of the device. The strength of the social links between users and the history of various social activities can determine how much and which video will be prefetched.

Traditional adaptive streaming frameworks have to maintain multiple copies of the video content with different bit rates, and thus bring huge burden of storage on the server. Therefore the recent H.264 Scalable Video Coding (SVC) technique has gained lots of attention.

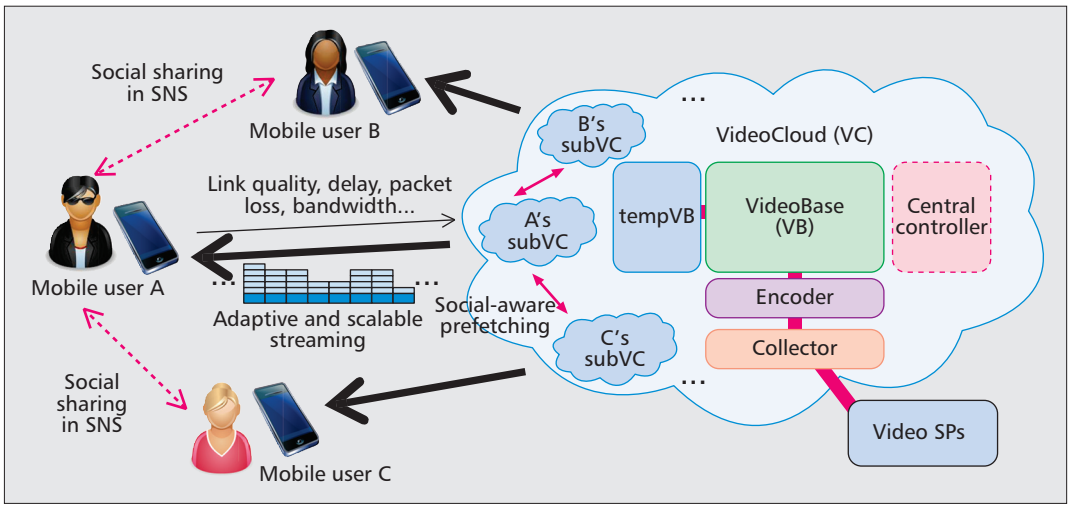

Figure 1. *An illustration of the proposed framework.*

Cloud Agent for Mobile Users

As shown in Fig. 1, the whole video storing and streaming system in the cloud is called the VideoCloud (VC). In the VC, there is a large-scale VideoBase (VB), which stores most of the popular video clips from the video service providers (VSPs). A temporary VideoBase (tempVB) is used to cache new candidates for the popular videos. The VC also keeps running a collector to seek popular videos from the VSPs, and re-encode the collected videos into SVC format and store into tempVB first.

Specialized for each active mobile user, a sub-VideoCloud (subVC) is created dynamically once there is any video streaming demand from the mobile user. Each sub-VC has a sub-VideoBase (subVB), which stores the recently fetched video segments. Note that the video deliveries among the subVCs and the VC in many cases are actually not "copy," but just "link" operations of the same file eternally within one cloud data center; even in other cases that videos are copied from one data center to another, it will be very fast [15]. There is also an encoding function in subVC (actually a smaller-scale encoder instance of the encoder in VC), and if the mobile user demands a new video that is not in the subVB or in the VB, the subVC will fetch, encode and transfer the video. During the video streaming, mobile users will always periodically report link conditions to their corresponding subVCs, and then the subVCs make prediction of the available bandwidth of next time window and adjust the combination of BL and ELs adaptively. Each mobile device also has a temporary caching storage, which is called local-VideoBase (localVB), and is used for buffering and prefetching.

If a video is accessed in the subVCs at a certain frequency threshold (e.g., 100 times per day), it will be uploaded to the tempVB; and if it is further accessed at a much higher frequency (e.g., 10,000 times per day), it will be stored with a longer lifetime in the VB. In such a 2-tier system, the subVB and VB can always store fresh and popular videos in order to increase the reusage probability. Note that management work will be handled by the controller in the VC.

Cloud-Assisted Adaptive Video Streaming

Traditional adaptive streaming frameworks, e.g., Microsoft's smooth streaming technique, Adobe's and Apple's HTTP adaptive live streaming solutions, have to maintain multiple copies of the video content with different bit rates, and thus bring huge burden of storage on the server. Therefore the recent H.264 Scalable Video Coding (SVC) technique has gained lots of attentions [3–6]. SVC defines diverse profiles of video streams with one base layer (BL) and multiple enhancement layers (ELs). These layers, or say substreams, can be encoded by exploiting three scalability features:

- Spatial scalability by layering image resolution (screen pixels)
- Temporal scalability by layering the frame rate
- Quality scalability by layering the image compression, and thus can offer videos for a high variety of quality with relatively less storage overhead

The real-time SVC decoding and encoding on PC servers is studied in [3]. Also the work in [5] has deployed the cloud-based SVC proxy and discovered that the cloud computing can significantly improve the performance of SVC coding. One more strength of cloud-based SVC encoding is that, once a user has requested to encode a video by a subVC, the encoded segments of layers will be able to be re-used among subVCs, and thus other users don't need to request to re-encode the video.

Once a mobile user actively initializes to stream a video, a cloud agent will be rapidly generalized for that user. The mobile client keeps tracking on metrics, including signal strength, packet round-trip-time (RTT), packet loss and bandwidth, under a certain duty cycle. And the client will periodically report to the subVC. Hereby we define the cycle period for the reporting as the "time window," denoted by T_{win}, Note that the video source inside subVC and VC is also segemented with interval T_{win} while being encoded by SVC.

Based on the feedback of link status of time

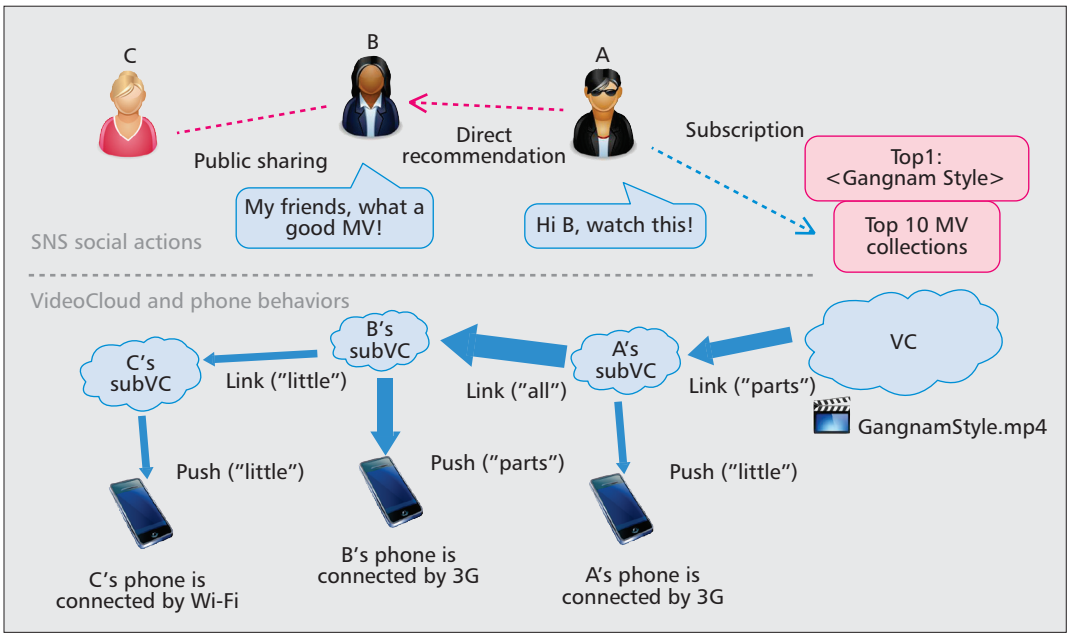

Figure 2. *Social-aware video prefetching.*

window T_i from clients, the cloud agent predicts the available link bandwidth of next time window T_{i+1} by a certain prediction algorithm. There are many accurate and sophisticated methods to predict the bandwidth, such as the work in [17]. However, we can just use a simple but still effective method to predict the bandwidth of T_{i+1} based on the practical bandwidth of T_i by multiplying with a adjusting factor, e.g., 0.9, that is, $BW_{i+1}^{pred} = BytesObtained_i^{prac}/T_{win} \cdot 0.9$ It actually has acceptable performance, because although the link quality is changing time by time, it is relatively stable if we zoom to small scale of time intervals, e.g., 5 seconds.

Once the bandwidth of next time window is predicted the subVC will match and decide how many ELs can be transmitted along with the BL. Note that different T_{win} and different number of encoded ELs can induce different encoding speed and overhead. Moreover, a fine-grained scheme with shorter Twin and larger number of ELs, can fit to the fluctuation better and waste less bandwidth, but it may bring more overhead; a coarse-grained scheme with longer T_{win} and smaller number of ELs has low complexity but cannot adapt to the link quality very well.

Social-Aware Video Prefetching

Social Content Sharing

In SNSs, users can subscribe to known friends, famous people, and particular content publishers; also there are various types of social activities among users in SNSs. So we need to define different strength levels for those social activities to indicate the different possibilities that the video shared by one user may be watched by the recipients of his/her sharing activities, so that subVCs can carry out effective background prefetching at subVB and even may push to user's localVB. Because after one shares a video, there may be a certain delay that the recipient gets to know the sharing, and initiates to watch [11, 16]. Therefore the prefetching in prior will not impact the users in most cases. Instead, a user can click to see without any buffering delay as the beginning part or even the whole video is already prefetched locally.

The amount of prefetched segments is mainly determined by the strength of the social activities and user's link status. And thus we classify the social activities in current popular SNSs into three kinds, regarding the impact of the activities and the potential reacting priority from the point of view of the recipients [12]:

Direct recommendation: In SNSs, a user can directly recommend a video to particular friend(s) by a short notice message. The recipients of the message may watch it with very high probability. This is considered as "strong".

Subscription: Like the popular RSS services, a user can subscribe to a particular video publisher based on interests. This interest-driven connectivity between the subscriber and the video publisher is considered as "median," because the subscriber may not always watch all subscribed videos.

Public sharing: The activity of watching or sharing a video by a user can be seen by his/her friends in their timeline of activity stream. We consider this public sharing as a "weak" connectivity among users, because many people may not watch the video that one has watched or shared with no specific recommendation.

Prefetching Levels

Different strengths of the social activities indicate different levels of probability that a video will be soon watched by the recipient. Correspondingly, we also define three prefetching levels regarding the social activities of mobile users:

"All": The video shared by the **direct recommendations** will be watched with a very high probability, so we propose to prefetch the BL and all ELs, in order to let the recipient(s)

In SNSs, users can subscribe to known friends, famous people, and particular content publishers; also there are various types of social activities among users in SNSs. So we need to define different strength levels for those social activities.

directly watch the video with a good quality, without any buffering.

"Parts": Because the videos that are published by **subscriptions** may be watched by the subscribers with a not so high probability, we propose to only prefetch parts of the BL and ELs segments, for example, the first 10 percent segments.

"Little": The **public sharing** has a weak impact, so the probability that a user's friends (followers) watch the video that the user has watched or shared is relatively low. We propose to only prefetch the first BL segment to those who have seen his/her activity in their timeline stream.

The prefetching happens among subVBs and the VB, and more importantly, will be performed from the subVB from the cloud to the localVB of the mobile device depending on the link quality. If a mobile user is covered by Wi-Fi access, due to Wi-Fi's capable link and low price (mostly for free), subVC can push as much as possible in most cases regarding the prefetching levels. However if he/she is with a 3G/4G connection, we propose to downgrade the prefetching level to save energy and money, but users can still benefit from the prefetching effectively for little buffering delay.

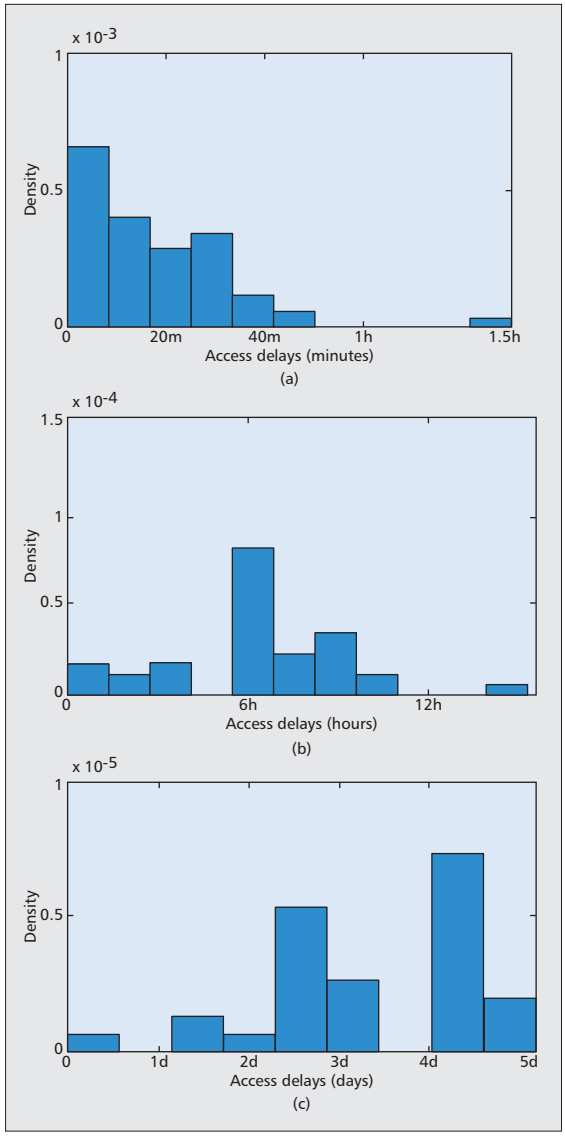

Figure 3. *Access delays of mobile users to video contents.*

For example, as shown in Fig. 2, when user A gets an interesting music video (MV) from his/her subscription, prefetching level "parts" should be chosen; however because A is connected by 3G, the prefetching will be downgraded to "little". User *B* gets direct recommendation of the MV from *A*, and thus *B*'s subVC will prefetch the video at the level of "all"; but *B* is also connected by 3G, so only "parts" of the video segments will be pushed. User *C* sees *B*'s sharing activity of the MV while *C* is connected by Wi-Fi, and thus *C*'s subVC will push "little" of the segments to *C*. Note that users themselves can also configure the prefetching conditions on demand.

Different mobile users have different patterns of accessing videos [11, 16] that are per-user dependent mainly due to people's different life styles. We capture a realistic trace from the Chinese biggest SNS site, Sina Weibo, and check the access delay of about 2 million people during July 2012. We draw the Probability Distribution Functions of access delays from three randomly selected users in Fig. 3. We find that although user X has very short delays for checking updates in SNSs, user Y and user Z often have access delays for hours, and even up to one or two days. Based on our analysis on the whole user base, the average access delay is about 6.5 hours, and even more than one third of the users have access delay larger than 1 day. These large access delays offer us huge potential to prefetch the videos before users access them.

Video Storage and Streaming Flow

The two parts, cloud-assisted adaptive video streaming and social-aware video prefetching in the framework, have tight connections and will together service the video streaming and sharing: they both rely on the cloud computing platform and are carried out by the private agencies of users; while prefetching, the streaming part will still monitor and improve the transmission considering the link status.

Once a mobile user starts to watch a video via a link, the localVB will first be checked whether there are any prefetched segments of the video. If there is none or just some parts, the client will report to its subVC, and if the subVC has the video, the subVC will initiate for transmission the remaining segments. But once there is no prefetched parts of the video in the subVB, the tempVB and VB in the central VC will be checked. In the case that there is no such a video in tempVB or VB, the collector in VC will immediately fetch the video from external video providers via the link, and the subVC will re-encode in SVC format, taking a bit large delay, and then stream to the mobile user.

Performance Evaluation

We evaluate the performance of the framework by a prototype implementation. We choose the U-cloud server (premium) in the cloud computing service offered by KT, and utilize the virtual

server with 6 virtual CPU cores (2.66GHz) and 32GB memory, which is fast enough for encoding 480P (480 by 720) video with H.264 SVC format in 30fps near real-time [3]. In the cloud, we deploy our server application based on Java, including one main program handling all tasks of the whole VC, while the program dynamically initializes, maintains and terminates instances of another small Java application as private agents for all active users. We implement the mobile client at a mobile phone, Samsung Galaxy II, with android system version 4.0. Note that the mobile data service is offered by LG U+ LTE network.

The test video is the Tomb Raider 2012 Trailer in H.264 format with 480P resolution downloaded from YouTube. Its size is 13.849 Mbytes and with a duration of 180 seconds. First, we decode it by the x264 decoder into the YUV format, and re-encode it by the H.264 SVC encoder, the Joint Scalable Video Model (JSVM) software of version 9.1. We just use default settings for the decoding and encoding in the virtual server in the cloud. We split the video into segments by varied T_{win} of 1s, 2s, 3s, 4s and 5s. By JSVM, besides the base layer, we further generate 5 temporal layers (1.875, 3.75, 7.5, 15, and 15 fps), 2 spatial layers (240 by 360 and 120 by 180) and 2 more quality layer (low and high). Thus we define the best resolution configuration as "1:5:2:2," denoted by the sequence of "BL : temporal : spatial : quality". Due to the limited space, we cannot test all the combinations of the layers, so we choose five resolution configurations, including "1:1:1:1," "1:2:2:2," "1:3:2:2," "1:4:2:2," and "1:5:2:2".

Adaptive Video Streaming

Firstly we examine whether there is a deep relationship between the measured bandwidth of last time window and the practical bandwidth of next time window. We test the video streaming service via 3G/4G link, and move the device around in the building to try to change the signal quality. Note that all tests are carried out for five times. We collect the relative errors of the predicted bandwidth to the practical bandwidth for every time window, calculated by

$$\frac{\left|BW^{pred} - BW^{prac}\right|}{BW_{prac}},$$

and find that when T_{win} is 2 seconds, the predicted bandwidth is near to the practical bandwidth by around 10 percent relative error, but large values of T_{win} have relatively poor prediction accuracy. So a short T_{win} for the accurate prediction is suggested in practical implementation.

Video Streaming in subVC and VC by SVC

We evaluate how H.264 SVC works in the cloud regarding the above mentioned SVC configurations. As shown in Fig. 4, because of the strong computational capacity of the cloud computing, the encoding speed is dramatically fast. The best resolution configuration "1:5:2:2" with 5-second temporal segmentation scheme requires about 560 ms for encoding. For those very short intervals of Twin, the encoding delay is small under 50 ms.

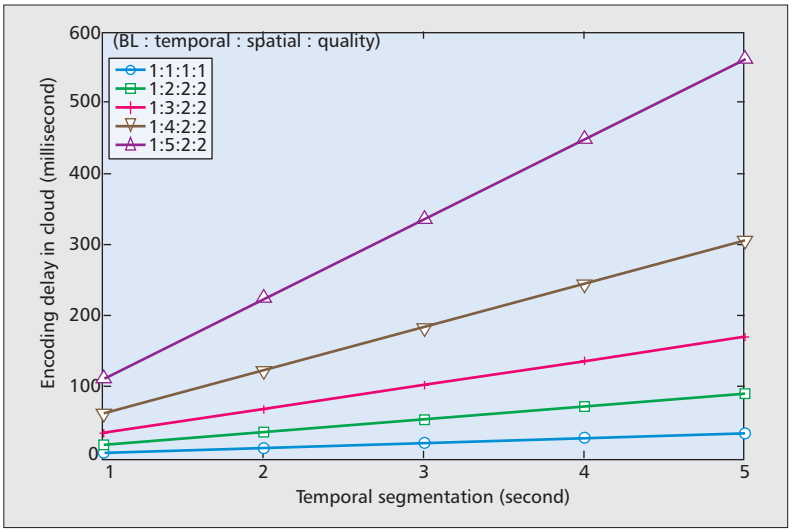

Figure 4. *SVC encoding delays in the cloud.*

More ELs induce higher overhead due to the duplicated I-frames, and thus we test the storage overhead, which is calculated by the ratio of the extra size of the video segments during SVC encoding to the size of only the low-quality BL. As shown in Fig. 5, the resolution scheme of "1:1:1:1" has a low overhead around below 10 percent, and "1:2:2:2" with two ELs for each scalability feature has about 17 percent overhead, which is acceptable. However higher resolution like "1:4:2:2" has 61 percent overhead, and "1:5:2:2" has even 120 percent overhead, which is not efficient. Overall, an SVC stream should not contain too many ELs, that is, a high scalability practically brings high overhead.

Click-to-Play Delay

It is obvious that the video prefetching among localVB, subVB and VB can be significantly fast due to the high link capacity of cloud data centres. Then we only show our test results on how long one user has to wait from the moment that he/she clicks the video in the mobile device to the moment that the first segment is displayed, which is defined as "click-to-play" delay. As shown in Fig. 6, if the video has been cached in localVB, the video can be displayed nearly immediately with ignorable delay. When we watch videos which are fetched from the subVC or the VC, it generally takes no more than 1 second to start. Even if the user accesses the content by 3G/4G link, he/she will suffer a very short delay (around 1s).

For the cases to fetch videos which are not in the cloud (but in our server at lab), denoted by "outside," the delay is a bit higher. This is mainly due to the fetching delay via the link from our server at lab to the cloud data center, as well as the encoding delay in the cloud. However this won't happen frequently, since most of the popular videos will be cached in the VC. Furthermore the access delays that analyzed earlier are much larger than those click-to-play delays, which means this social aware-prefetching can perform perfectly to match the user demands. More measurement and tests will be carried out as future work.

Conclusions and Future Remarks

In this article, we discussed our proposal of the cloud-assisted adaptive mobile video streaming and social-aware prefetching, which efficiently stores videos in the clouds and elastically constructs private agent (subVC) for active mobile user to try to offer "non-terminating" video streaming by adapting to the fluctuation of link quality based on SVC technique, and to try to provide "non-buffering" video streaming experience by background prefetching based on the tracking of the interactions of mobile users in their SNSs. We evaluated the framework by prototype implementation, and showed that the cloud computing technique brings significant improvement to the adaptability and scalability of the mobile streaming, and the efficiency of intelligent prefetching. Regarding the future work, we will carry out large-scale implementation tests with the consideration on energy and price cost regarding the usage patterns of mobile users [18]. Also we will try to extend our framework with more concerns of security and privacy.

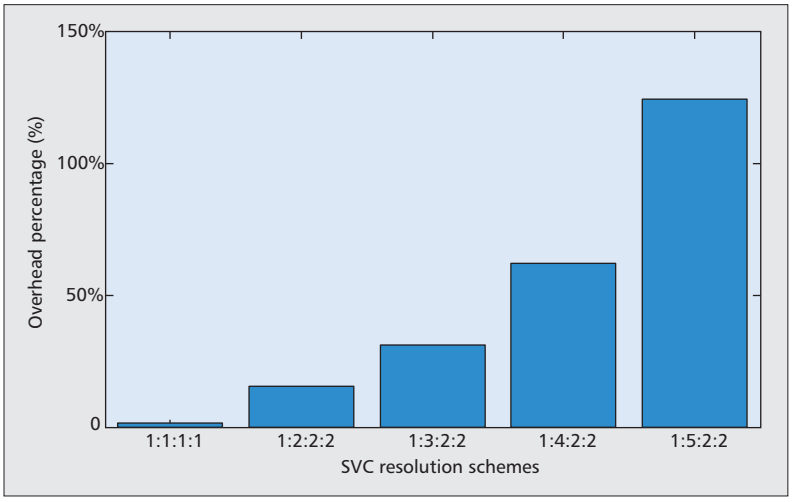

Figure 5. *SVC encoding overhead in the cloud.*

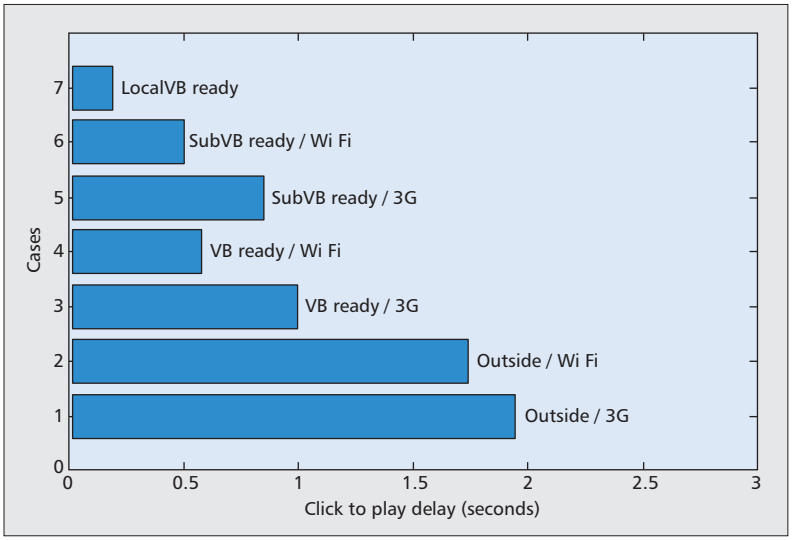

Figure 6. *Click-to-play delays for various cases.*

Acknowledgement

This research was partially supported by the KCC (Korea Communications Commission), Korea, under the R&D program supervised by the KCA (Korea Communications Agency) (KCA-2013-11-911-05-002), and also supported by International Collaborative R&D Program of the Ministry of Trade, Industry & Energy (MOTIE), Korea/Korea Institute for Advancement of Technology (KIAT). Prof. Ted "Taekyoung" Kwon is the corresponding author of this article.

References

[1] CISCO, "Cisco Visual Networking Index : Global Mobile Data Traffic Forecast Update , 2011–2016," Technical Report, 2012.

[2] Y. Li, Y. Zhang, and R. Yuan, "Measurement and Analysis of a Large Scale Commercial Mobile Internet TV System," *Proc. ACM IMC*, 2011, pp. 209–24.

[3] M. Wien *et al.*, "Real-Time System for Adaptive Video Streaming Based on SVC," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 17, no. 9, 2007, pp. 1227–37.

[4] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Trans. Circuits and Systems for Video Tech.*, Vol. 17, No. 9, pp. 1103–1120, 2007.

[5] Z. Huang *et al.*, "CloudStream: Delivering High-Quality Streaming Videos through A Cloud-based SVC Proxy," *Proc. IEEE INFOCOM*, 2011, pp. 201–205.

[6] P. McDonagh, C. Vallati, A. Pande, and P. Mohapatra, "Quality-Oriented Scalable Video Delivery using H. 264 SVC on an LTE Network," *Proc. WPMC*, 2011.

[7] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud Computing: State-of-the-art and Research Challenges," *J. Internet Services and Applications*, vol. 1, no. 1, 2010, pp. 7–18.

[8] D. Niu *et al.*, "Quality-Assured Cloud Bandwidth Auto-Scaling for Video-on-Demand Applications," *Proc. IEEE INFOCOM*, 2012.

[9] H. T. Dinh *et al.*, "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches," *Wiley J. Wireless Commun. and Mobile Computing*, 2012.

[10] N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, no. 4, 2009, pp. 14–23.

[11] H. Kwak *et al.*, "What is Twitter, a Social Network or a News Media?" *Proc. WWW*, 2010.

[12] Z. Wang *et al.*, "Guiding Internet-Scale Video Service Deployment Using Microblog-Based Prediction," *Proc. IEEE INFOCOM*, 2012.

[13] T. Rodrigues *et al.*, "On Word-of-Mouth Based Discovery of the Web," *Proc. ACM IMC*, 2011.

[14] M. P. Wittie *et al.*, "Exploiting Locality of Interest in Online Social Networks," *Proc. ACM CoNEXT*, 2010.

[15] G. Wang and T. E. Ng, "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center," *Proc. IEEE INFOCOM*, 2010.

[16] F. Benevenuto *et al.*, "Video Interactions in Online Social Networks," *ACM Trans. Multimedia Computing, Commun. and Applications*, vol. 5, no.4, 2009, pp. 30–44.

[17] M. Mirza *et al.*, "A Machine Learning Approach to TCP Throughput Prediction," *Proc. ACM SIGMETRICS*, 2007.

[18] J. Kang, S. Seo, and J. Hong "Personalized Battery Lifetime Prediction for Mobile Devices based on Usage Patterns," *J. Computer Science and Engineering*, vol. 5, no. 4, 2011, pp. 338–45.

Biographies

Xiaofei Wang (xiaofeiwang@ieee.org) is a Ph.D candidate in the Multimedia & Mobile Communication Laboratory (MMLAB), School of Computer Science and Engineering (CSE), Seoul National University (SNU), Korea. He received the B.S. degree in the Department of Computer Science and Technology of Huazhong University of Science and Technology (HUST) in 2005, and M.S. degree from the School of CSE at SNU in 2008. He is a recipient of the Chinese Scholarship for Outstanding Self-financed Students Studying Aboard 2012. His current research interests are multimedia transmission by mobile cloud computing and traffic offloading in mobile content-centric networks.

TED "TAEKYOUNG" KWON (tkkwon@snu.ac.kr) is a professor in the School of CSE, SNU, Korea. Before joining SNU, he was a postdoctoral research associate at UCLA and CUNY. He obtained B.S., M.S., and Ph.D. degrees from the School of CSE, SNU, in 1993, 1995, and 2000, respectively. During his graduate program, he was a visiting student at IBM T. J. Watson Research Center and at University of North Texas. His research interest lies in sensor networks, wireless networks, IP mobility, and ubiquitous computing.

YANGHEE CHOI (yhchoi@snu.ac.kr) is a professor in the School of CSE, SNU, Korea. Prof. Choi received the B.S. degree in electronics engineering from SNU, Korea, in 1975, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1977, and the Ph.D. degree of engineering in computer science from the Ecole Nationale Superieure des Telecommunications, Paris, France, in 1984. From 1988 to 1989, he was a visiting scientist with the IBM T. J. Watson Research Center, Yorktown Heights, NY. Since 1991, he has been with the School of CSE, SNU, where he is currently leading the MMLAB. Prof. Choi is the President of the Korean Institute of Information Scientists and Engineers, and also the Chair of the Future Internet Forum. His research interests mainly include the Future Internet and Content-Centric Networks.

HAIYANG WANG (hwang@d.umn.com) received the Ph.D. degree in the department of Computing Science from Simon Fraser University, Burnaby, British Columbia, Canada in 2013. He is currently an assistant professor in the Department of Computer Science, University of Minnesota Duluth. His research interests include cloud computing, bigdata, peer-to-peer networks and multimedia systems/networks.

JIANGCHUAN LIU (jcliu@cs.sfu.ca) is currently an associate professor in the School of Computing Science, Simon Fraser University, British Columbia, Canada, and was an assistant professor in the Department of Computer Science and Engineering at The Chinese University of Hong Kong from 2003 to 2004. He received the B.S. degree from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from The Hong Kong University of Science and Technology in 2003, both in computer science. He is a recipient of Microsoft Research Fellowship (2000), Hong Kong Young Scientist Award (2003), and Canada NSERC DAS Award (2009). His research interests include multimedia systems and networks, wireless ad hoc and sensor networks, and peer-to-peer and overlay networks.