International Journal of Advanced Technology and Innovative Research

ISSN 2348–2370
Vol.06,Issue.12,
December-2014,
Pages:1429-1436

www.ijatir.org

Mobile Video Streaming and Efficient Social Video Sharing in the Clouds Based on Adaptive Framework

AKEPOGU JOYPRANAHITHA[1], K. SAMSON PAUL[2]

[1]PG Scholar, Dept of CSE, Stanley Stephen College of Engineering and Technology, Panchalingala, Kurnool, AP, India, Email: joysofttech@gmail.com.

[2]Asst Prof, Dept of CSE, Stanley Stephen College of Engineering and Technology, Panchalingala, Kurnool, AP, India.

Abstract: While demands on video traffic over mobile networks have been souring, the wireless link capacity cannot keep up with the traffic demand. The gap between the traffic demand and the link capacity, along with time-varying link conditions, results in poor service quality of video streaming over mobile networks such as long buffering time and intermittent disruptions. Leveraging the cloud computing technology, we propose a new mobile video streaming framework, dubbed AMES-Cloud, which has two main parts: AMoV (adaptive mobile video streaming) and ESoV (efficient social video sharing). AMoV and ESoV construct a private agent to provide video streaming services efficiently for each mobile user. For a given user, AMoV lets her private agent adaptively adjust her streaming flow with a scalable video coding technique based on the feedback of link quality. Likewise, ESoV monitors the social network interactions among mobile users, and their private agents try to prefetch video content in advance. We implement a prototype of the AMES-Cloud framework to demonstrate its performance. It is shown that the private agents in the clouds can effectively provide the adaptive streaming, and perform video sharing (i.e., Perfecting) based on the social network analysis.

Keywords: Scalable Video Coding, Adaptive Video Streaming, Mobile Networks, Social Video Sharing, Cloud Computing.

I. INTRODUCTION

Over the past decade, increasingly more traffic is accounted by video streaming and downloading. In particular, video streaming services over mobile networks have become prevalent over the past few years [2]. While the video streaming is not so challenging in wired networks, mobile networks have been suffering from video traffic transmissions over scarce bandwidth of wireless links. Despite network operators' desperate efforts to enhance the wireless link bandwidth (e.g., 3G and LTE), soaring video traffic demands from mobile users are rapidly overwhelming the wireless link capacity. While receiving video streaming traffic via 3G/4G mobile networks, mobile users often suffer from long buffering time and intermittent disruptions due to the limited bandwidth and link condition fluctuation caused by multi-path fading and user mobility [3] [4] [5]. Thus, it is crucial to improve the service quality of mobile video streaming while using the networking and computing resources efficiently [6] [7] [8] [9]. Recently there have been many studies on how to improve the service quality of mobile video streaming on two aspects:

Scalability: Mobile video streaming services should support a wide spectrum of mobile devices; they have different video resolutions, different computing powers, different wireless links (like 3G and LTE) and so on. Also, the available link capacity of a mobile device may vary over time and space depending on its signal strength, other user's traffic in the same cell, and link condition variation. Storing multiple versions (with different bit rates) of the same video content may incur high overhead in terms of storage and communication. To address this issue, the Scalable Video Coding (SVC) technique (Annex G extension) of the H.264 AVC video compression standard [10] [11] [12] defines a base layer (BL) with multiple enhance layers (ELs). These sub streams can be encoded by exploiting three scalability features: (i) spatial scalability by layering image resolution (screen pixels), (ii) temporal scalability by layering the frame rate, and (iii) quality scalability by layering the image compression. By the SVC, a video can be decoded/played at the lowest quality if only the BL is delivered. However, the more ELs can be delivered, the better quality of the video stream is achieved.

Adaptability: Traditional video streaming techniques designed by considering relatively stable traffic links between servers and users perform poorly in mobile environments [3]. Thus the fluctuating wireless link status should be properly dealt with to provide 'tolerable" video streaming services. To address this issue, we have to adjust the video bit rate adapting to the currently time-varying available link bandwidth of each mobile user. Such adaptive streaming techniques can effectively reduce packet losses and bandwidth waste. Scalable video coding and adaptive streaming techniques can be jointly combined to accomplish effectively the best possible quality of video streaming services. That is, we can dynamically adjust the number of SVC layers depending on the current link status [10].

However most of the proposals seeking to jointly utilize the video scalability and adaptability rely on the active control on the server side. That is, every mobile user needs to individually report the transmission status (e.g., packet loss, delay and signal quality) periodically to the server, which predicts the available bandwidth for each user. Thus the problem is that the server should take over the substantial processing overhead, as the number of users increases. Cloud computing techniques are poised to flexibly provide scalable resources to content/service providers, and process offloading to mobile users. Thus, cloud data centers can easily provision for large-scale real-time video services as investigated in [10]. Several studies on mobile cloud computing technologies have proposed to generate personalized intelligent agents for servicing mobile users, e.g., Cloudlet and Stratus. This is because, in the cloud, multiple agent instances (or threads) can be maintained dynamically and efficiently depending on the time-varying user demands.

Recently social network services (SNSs) have been increasingly popular. There have been proposals to improve the quality of content delivery using SNSs. In SNSs, users may share, comment or re-post videos among friends and members in the same group, which implies a user, may watch a video that her friends have recommended (e.g). Users in SNSs can also follow famous and popular users based on their interests (e.g., an official face book or twitter account that shares the newest pop music videos), which is likely to be watched by its followers. In this regard, we are further motivated to exploit the relationship among mobile users from their SNS activities in order to prefetch in advance the beginning part of the video or even the whole video to the members of a group who have not seen the video yet. It can be done by a background job supported by the agent (of a member) in the cloud; once the user clicks to watch the video, it can instantly start playing.

In this paper, we design a adaptive video streaming and prefetching framework for mobile users with the above objectives in mind, dubbed AMES-Cloud. AMES-Cloud constructs a private agent for each mobile user in cloud computing environments, which is used by its two main parts: (i) AMoV (adaptive mobile video streaming), and ESoV (efficient social video sharing). The contributions of this paper can be summarized as follows: AMoV offers the best possible streaming experiences by adaptively controlling the streaming bit rate depending on the fluctuation of the link quality. AMoV adjusts the bit rate for each user leveraging the scalable video coding. The private agent of a user keeps track of the feedback information on the link status. Private agents of users are dynamically initiated and optimized in the cloud computing platform. Also the real-time SVC coding is done on the cloud computing side efficiently.

AMES-Cloud supports distributing video streams efficiently by facilitating a 2-tier structure: the first tier is a content delivery network, and the second tier is a data center. With this structure, video sharing can be optimized within the cloud. Unnecessary redundant downloads of popular videos can be prevented. Based on the analysis of the SNS activities of mobile users, ESoV seeks to provide a user with instant playing of video clips by prefetching the video clips in advance from her private agent to the local storage of her device. The strength of the social links between users and the history of various social activities can probabilistically determine how much and which video will be pre fetched. The rest of the paper is organized as follows. Comparative Study on Video Sharing and Streaming Methods in Section II, and Streaming and Sharing of Videos in Mobile Network in Section III. Finally, we evaluate the prototype implementation in Section IV, and conclude the paper in Section V.

II. COMPARATIVE STUDY ON VIDEO SHARING AND STREAMING METHODS

There are number of studies show the video sharing and rendering in wireless devices and mobiles has been carried over the last decade. Juan Carlos Fernandez et al has proposed idea of negotiation the bandwidth with service provider dynamically so to provide the QoS to the customer. The service agreement can also be dynamically as the negotiation of the service bandwidth changes dynamically and have proposed novel method for dynamically managing the wireless network by observing the usage logs of the smart phone users and usage patterns of the customer under a particular service provider. This helps to understand and allocates reliable resource for the customer as per their requested service. Guenther Liebl et al used TFRC – TCP friendly rate control for adaptively streaming videos over the wireless and mobile network. Which provides the analysis of data transfer over the devices in the network and load of the service is dynamically balanced as per the video service requests from the user.

Prasad Calyam et al have constructed Future Internet Performance Architecture (FIPA), which provides new scheme for providing service over the internet to the customer based on their request. The architecture provide stable based for application oriented service over the internet. The AMES cloud was built specifically to provide service of video sharing and streaming over the cloud. The user of the video service in cloud would be mobile users most of the time. The data rate and the quality of service should not be affected in any way such as data disruption or low bandwidth etc. AMES provides protocol to be serviced to client and service provided to monitor and give the reliable service.

A. Adaptive and Efficient Video Streaming and Sharing In Cloud

The fig.1 shows the architecture of the adaptive and efficient way of enhancing the video streaming and sharing of video to the mobile users. The architecture was constructed based on the video service provided in cloud called as “AMES”. The architecture contains

Video service provider (VSP): the originated place of actual video data. It used the traditional video service provider. VSP can handle multiple requests at the same time, while coming to the QoS with the mobile users, the VSP does not provide service up to the mark.

Video cloud (VC): the cloud step up has been established with many components working together, virtually to get the original video data from the VSP and provide the reliable service to the mobile user and it also provides availability of video and makes the sharing of those videos among the users much easier.

Video base (VB): Video base consists of the video data that are provided as the service to the mobile users in cloud.

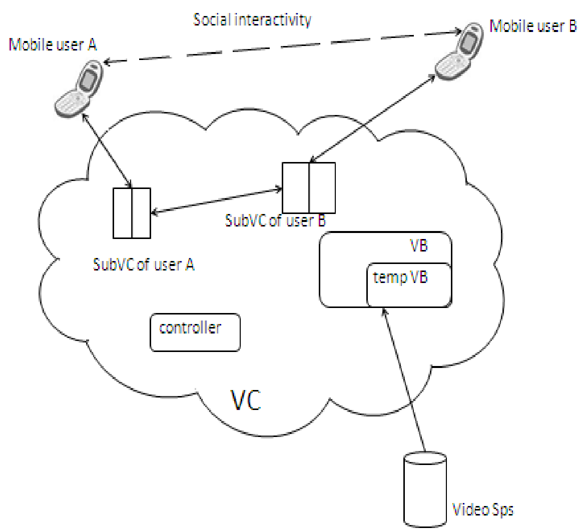


Fig.1. VC architecture.

Temp video base (TVB): it contains the most recently accessed video data and it also contains most frequently accessed video data.

Vagent: it is an agent created for every mobile user who requests for the video service to the video cloud.

Mobile users: the users who are mobile and providing the availability of the service to their location is difficult.

The video cloud provides services under two main methodologies adaptive mobile video streaming and efficient mobile video sharing. The video streaming and video sharing plays the vital role in providing the reliable service to the customers. The rate in which frames of the videos are streams determines the quality and availability of the video service. Video data are most commonly shared among the users in the network. Mobile users are most commonly found to use social networking sites more offently. The mobile device and mobile computing provides them space to be connected on the social network. Multimedia data such as images and videos are shared among the friend and users of the social media. The request of the video and sharing of video are two main actions requested from customer. Video cloud provides platform to provide these two services in better way.

The video service provider (VSP) contains the raw video data; the videos available in VSP can be used to service the customer's request. But VSP does not have sufficient resource to provide QoS and better video sharing among mobile devices and users. The Video cloud (VC) contain video base (VB) which collect the requested videos from the VSP and keeps the copy of the video, so as the request for the videos can be services. The Temporary video base (Temp VB) stores the link of the videos that are accessed more recently and frequently, the links provides faster access to the videos on the VB. The controller plays the important role of managing the working and coordination of all the components on the video cloud and mobile users. For every mobile user who comes for the service in cloud, one agent is created "Vagent". This video agent is responsible for processing the user's request and delivery the servers' response to the user. The requested videos link will be saved in Vagent for retransmission and for services if the same videos are requested again by the client. The Vagent can communicate among them for providing adaptive streaming of services. The video source or link available to one Vagent can be accessed and used by another Vagent. The mobile user can also communicate among them. The social interaction are carried out, the sharing of videos are also tracked and carried out through the Vagent of each user. Hence tracking of the video source availability and provides video to the requested user becomes easier. The video sharing in social media becomes efficient for video streaming.

III. STREAMING AND SHARING OF VIDEOS IN MOBILE NETWORK

A. Cloud Computing Technique

Cloud computing techniques are used to flexibly provide scalable resources to content, service providers, and process offloading to mobile users. Thus, cloud data centers can easily provision for large-scale real-time video services as. Several studies on mobile cloud computing technologies have proposed to generate personalized intelligent agents for servicing mobile users, hence, in the cloud, multiple agent instances or multithreads can be maintained dynamically and efficiently depending on the time-varying user demands.

B. Scalable Video Coding (SVC)

SVC is an extension to the H.264/AVC standard. It is classified as a layered video codec which can encode a video stream in several types and numbers of enhancement layers on top of the H.264/AVC compatible base layer. These enhancement layers can be added or removed from the bit stream during streaming without re-encoding of the media. The transmission rate of scalable video streams in the mobile network can be controlled by using TCP friendly rate control. The streams are encoded using the

SVC extension of the H.264/AVC standard. Adding or removing the layers is decided based on the TFRC during varying channel conditions of the mobile network. SVC provides a high quality multimedia communication services in heterogeneous network environment, especially when the client processing power, system resources, and network state unknown. The SVC video streams have flexible, Scalability and high quality coding efficiency.

1. TFRC (TCP-friendly rate control)

The bit rate of the stream can be dynamically adapted to the changing channel conditions which greatly improve all performance indicators such as interruption time, loss rate, and delay and buffer requirements. This also implies that more users could be admitted to the cell and it would still be able to guarantee certain service qualities. This is especially true in loaded situation where there are not enough radio resources to combat bad reception quality to some users. However, since the TFRC was not designed for a mobile environment, we expect that it can be further optimized.

2. H.264/SVC

In the scalable video coding extension of the H.264/AVC standard, an exhaustive search technique is used to select the best coding mode for each macro block. This technique achieves the highest possible coding efficiency, but it demands a higher video encoding computational complexity which constrains its use in many practical applications. This proposes combined fast sub-pixel motion estimation and a fast mode decision algorithm for inter-frame coding for temporal, spatial, and coarse grain signal-to-noise ratio scalability. The correlation is used between the macro block and its enclosed partitions at different layers. It has been observed that there is a high correlation between the MB and its enclosed partitions when estimating the motion at different resolutions. Therefore a two step fast sub –pixel motion estimation scheme based on this observation has been developed. a) In the first step, if the 16×16 MB finds a best match in the full-pixel motion search that does not change after performing the sub-pixel motion search (cond_1), then the sub-pixel motion search for all the enclosed 16×8 and 8×16 blocks is disabled. b) Similarly, if in the second step the 8×8 block partitions of the 16×16 MB find the same best match in the full and sub-pixel motion searches (cond_2), the sub-pixel motion search for all the enclosed 8×4, 4×8 and 4×4 sub-blocks is disabled. We motivate the use of non reference video quality evaluation metrics which can be deployed in future for on-the-fly video evaluation which can used as part of a video quality assessment system in commercial deployments.

3. Quality for video services

Any issues that degrade a network’s ability to deliver packets will, as a consequence, degrade the quality of any real-time services of customers currently connected to the network. In the case of video services this degradation is likely to take on the following forms: pausing of playback due to buffer starvation, macro blocking in the case of lost (bi-) predictive frames or full loss of picture in the case of lost Intra-frames.

C. Video Streaming

In streaming procedure, it clip data file is sent to the end individual in a (more or less) continuous flow. It is simply a strategy for shifting information such that it can be prepared as a stable and ongoing flow and it is known as Streaming or encoded movie that is sent across information system is known as Streaming. Streaming movie is a series of "moving images" that are sent in compacted form over the Internet and shown by the audience as they appear. If a web individual is getting the information as sources then he/she does not have to wait around to obtain a large data file before viewing it clip or enjoying the sound.

1. Streaming Principle

Real-time video applications require media packets to arrive in a timely manner; excessively delayed packets are useless and are treated as lost. In streaming programs it is necessary for the information packets to reach their location in regular basis because the wait can cause the network blockage, and can result in the decrease in all those packets suffering from extreme wait. This causes decrease in quality of information, the synchronization between customer and hosting server to be damaged and mistakes to distribute in the provided movie. There are two types of steaming, one is real-time and other is prerecorded streaming. The protocol used for streaming purpose is UDP (User Datagram Protocol), which delivers the multimedia flow as a sequence of small packets. The majority of transport protocols perform over an RTP stack, which is implemented transport for video streaming.

2. Video Streaming Architecture

A cloud based source implements a streaming hosting server which is responsible for retrieving, sending and adapting it clip flow. Depending on the application, it clip may be protected on-line for a real-time broadcasting or pre-encoded and stored for broadcasting an on demand. Programs such as interactive movie, live broadcast, mobile movie streaming or interactive online games require real time encoding. However, applications such as movie on-demand require pre-encoded movie. When the multicast session is initialized, the streaming hosting server retrieves the compressed movie and begins the loading with the adequate bit rate stream.

D. Video Streaming Techniques

There are various streaming techniques for different mobiles, Smartphone describe below:

1. Progressive Download

The mobile customer has the option to use HTTP or HTTPS to gradually download a pre-created press data file partitioned in the appropriate codec’s for the product to play. As the data file starts to gradually download, playback is started enabling an almost immediate watching of the material. In the qualifications, the press gamer is constantly on the download the rest of the material. By comparison, without modern download the user would have to wait for the whole data file to obtain to the product before watching would start. During the play-back process, audiences are able to seek back and forth through the whole press data file. If the audience looks for forward to a point in the schedule that has not yet downloadable, the press gamer stop playback until the data comes.

2. HTTP Live Streaming

HTTP Live streaming (also known as HLS) is an HTTP based media streaming communications protocol implemented by Apple Inc. as part of their QuickTime X and phone. Apple's HTTP Live Streaming protocol (HLS), is an adaptive streaming video delivery protocol for ions devices. It utilizes the H.264 video codec, which is segmented and encapsulated in MPEG2 transport streams, and .M3U8 index files to deliver live and on-demand video. The device automatically selects the most appropriate stream given available bandwidth, CPU and platform constraints, downloads a manifest for that stream, and then downloads segmented chunks to the buffer for the playback.

3. Peer streaming

Peer streaming general architecture follows a client/server scheme and the P2P network helps the server in distributing the media content. In addition, when a peer has viewed the media, it gets at the same time a copy on its local hard-drive. Now it can provide the media to other requesting peers, lightening the load on the server. Any peer in the network could then provide the whole or part of the media to a client. It is important to design lightweight peers, which are not so dependent on each other. A peer helping the server delivering the data should perform simple operation with low CPU load since the peer might perform some other tasks in parallel. The client has more responsibility and should perform more complex tasks: co-coordinating the peers, retrieving the media from multiple peers, performing load balancing, handling peers online/offline status and displaying the media in real time. It is primordial to understand that servers, serving peers and clients are all nodes in the overlay network. A server is a peer, which has the data and sends it to the client, a serving peer is a machine in the P2P overlay, which has also the data or part of it and sends it to the client. A client is a peer, which requests data from the network. Peer-To-Peer (P2P) network is at the moment one of the most effective solution to improve the bandwidth and distribute streaming data within a large scale overlay, involving potentially thousands of nodes.

4. Cool streaming

Cool Streaming is a data-driven overlay network. This application coded in Python language creates its own overlay P2P network following a mesh topology. Its architecture is divided into three layers: network layer, streaming layer and display layer. Using an efficient scheduling algorithm to fetch video segments from each peer and a strong buffering system, Cool Streaming achieves a smooth video playback and a very good scalability as well as performance. The system has been extensively tested over the Planet Latest-bed. Their technical report shows that the overall streaming rate and playback continuity of Cool Streaming system is proportional to the amount of peers online at any given time. The Peer Streaming application has been tested in two ways: with and without embedded coded media. Embedded coding is a special feature of the Microsoft DirectShow framework in both tests; the streaming rate is between 16 and 128 kbps.

E. Metrics Affecting Streaming: Quality

1. Attributes of Streaming Quality

The streaming quality is a prerequisite for users to watch videos smoothly without interruptions, and thus directly impacts the human subjective perception. The access time that a user experiences before the start of an on-demand video playback represents the overall responsiveness of the video proxy. The latencies incurred at both trans coding and streaming components can contribute to the access time. Video freezes are caused by the unavailability of new video data at their scheduled playback time due to the combined contribution of trans-coding and streaming jitters. The user side buffering time should be large enough to accommodate the maximum streaming jitter in order to avoid video freezes. The video decoding time is negligible at both the trans coding component and the user.

IV. IMPLEMENTATION AND EVALUATION

We evaluate the performance of the AMES-Cloud framework by a prototype implementation. We choose the U-cloud server (premium) in the cloud computing service offered by Korean Telecom, 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 30 fps at real time [10]. 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. The mobile data service is offered by LG U+ LTE network, while in some uncovered area the 3G network is used. Note that we still use "3G" to indicate the general cellular network. We test in the downtown area, so the practical bandwidth of the mobile link is not as high as we expected, but this won't impact our experiment results.

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. We first 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 version9.1. We just use default settings for the decoding and encoding, and do the H.264 SVC encoding at the virtual server in the cloud. We split the video into segments by 1 second to 5 seconds that is to vary T_{win} with values 1s, 2s, 3s, 4s and 5s. By JSVM, besides the base layer, we further make five temporal layers (1.875, 3.75, 7.5, 15, and 15 fps), two spatial layers (240 by 360 and 120 by 180) and two more quality layer (low and high), referring. Thus we define the best resolution configuration as “1+5+2+2”. And we also test different resolution configurations, including “1+1+1+1”, “1+2+2+2”, “1+3+2+2” and “1+4+2+2”.

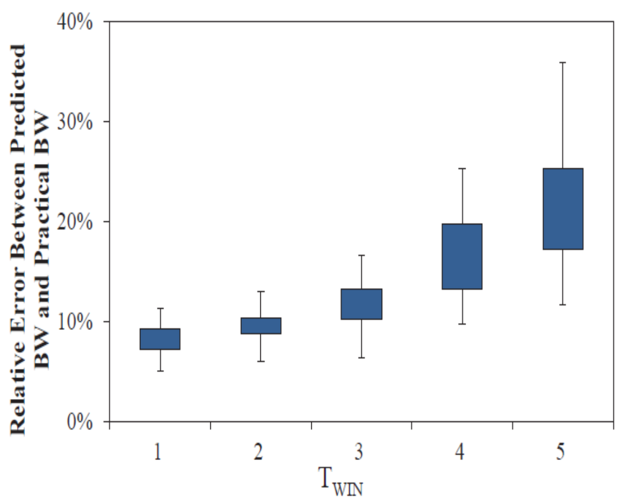
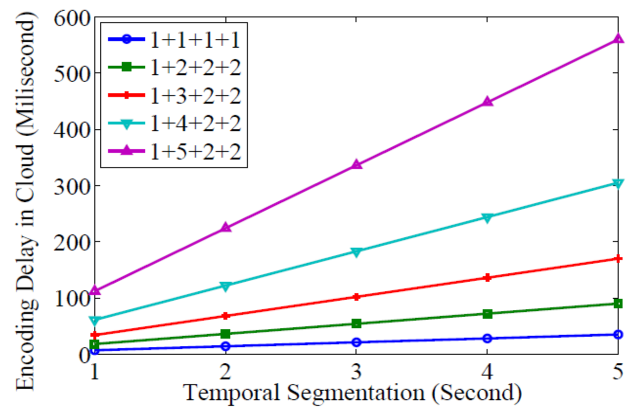


Fig.2. Relative errors between predicted bandwidth and practical bandwidth (percentage).

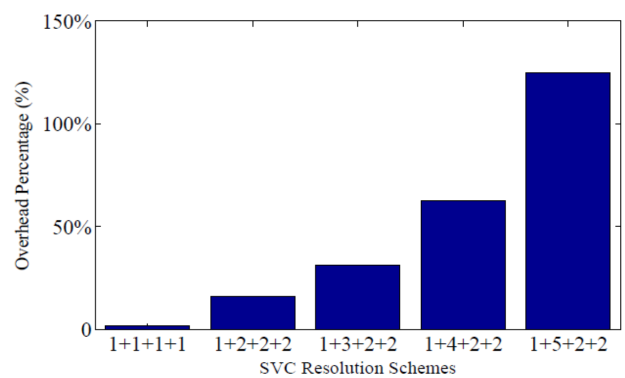
Firstly we examine whether there is a deep relationship between the measured bandwidth of last time window and the practical bandwidth of next time window (good put by Kbps). We test the video streaming service via cellular link, and move the device around in the building to try to change the signal quality. Note that all tests are ran five times. The collected the relative errors for the predicted bandwidth to the practical bandwidth for every time window, calculated by $\frac{|BW^{estimate} - BW^{practical}|}{BW^{practical}}$ are shown in Fig.2, where the bar indicates the 25% and 75% quartiles, and the whiskers indicate the 5% and 95% percentiles. When T_{win} is 1 second or 2 seconds, the predicted bandwidth is very near to the practical one with around 10% relative error, but large values of T_{win} have relatively poor prediction accuracy, which reflects the similar results. So we suggest a short T_{win} of 2 or 3 seconds for accurate prediction in practical designs.

B. Video Streaming in sub VC and VC

We evaluate how H.264 SVC works in AMES-Cloud framework regarding the above mentioned SVC resolution configurations. As shown in Fig. 3(a), because of the strong computational capacity by the cloud computing, the encoding speed is fast. The best resolution configuration “1+5+2+2” with 5 second temporal segmentation scheme requires about 560 ms for encoding. For shorter intervals of *Twin*, the encoding delay is very small under 50 ms because more ELs induce higher overhead due to the duplicated I-frames; we test the overhead, which is calculated by the ratio of the total size of the video segments after SVC encoding to the size of only the BL. As shown in Fig. 3(b), the resolution scheme of “1+1+1+1” has a low overhead around below 10%, and “1+2+2+2” with two ELs for each scalability feature has about 17% overhead, which is acceptable. However higher resolution like “1+4+2+2”



(a) Delay of difference SVC resolution schemes in the Cloud



(b) Overhead of different SVC resolutions schemes in the Cloud

Fig.3. Evaluation of SVC Resolution Schemes.

has 61% overhead, and “1+5+2+2” has even 120% overhead, which is not efficient. Overall, an SVC stream should not contain too many enhance layers for extremely high scalability, which may practically bring too much overhead.

C. Prefetching Delays

In ESoV, video segments can be prefetched among VB, temp VB, and local VBs of the mobile users, based on their activities in SNSs. we evaluate the required delays for different levels of prefetching as shown in Table.1. We here use the normal resolution configuration of “1+2+2+2” with 2 second temporal segmentation by default (the same in following tests). We also set the sharing length of “little” as only the first 5 seconds of the BL and ELs, that of “parts” as the first 15 seconds of the BL and ELs, and that of “all” as all BL and ELs segments.

TABLE I: DELAYS OF PREFETCHING SHARING FOR VARIOUS LEVELS

	Little	Parts	All
subVBs↔VB	0.011 s	0.023 s	0.098 s
subVB→locVB via Wi-Fi	2.421 s	4.359 s	23.221 s
subVB→locVB via 3G	N&A	18.430 s (little)	37.308 s (parts)

We can see that prefetching supported by the cloud computing is significantly fast. When prefetching via wireless links, it takes several seconds. However it is obvious that in most cases a recipient of the video sharing may not watch immediately after the original sharing behavior, that is normal users have significant access delay gaps, so this prefetching transmission delay won’t impact user’s experience at all, but will bring “non-buffering” experience in fact when the user clicks to watch at a later time.

D. Watching Delay

We test how long one user has to wait from the moment that one clicks the video in the mobile device to the moment that the first streaming segment arrives, which is called as “click-to-play” delay. As shown in Fig.4, if the video has been cached in local VB, the video can be displayed nearly immediately with ignorable delay. When we watch video which is fetched from the sub VC or the VC, it generally takes no more than 1 second to start.

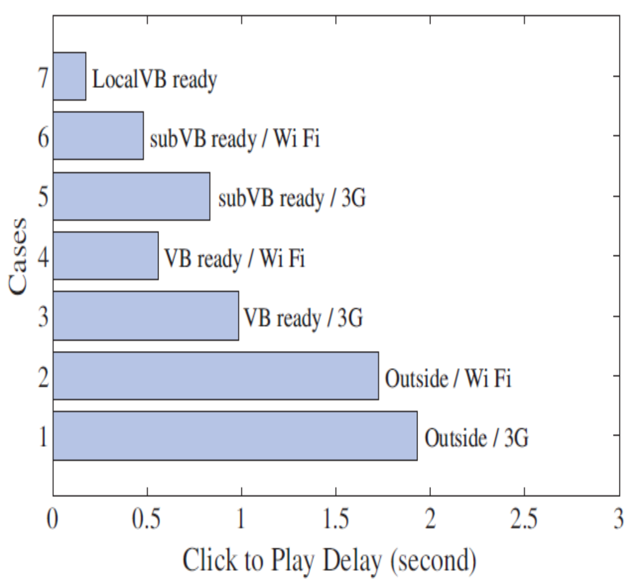


Fig.4. Average Click-to-Play delay for Various Cases.

However if the user accesses to AMES-Cloud service via the cellular link, he will still suffer a bit longer delay (around 1s) due to the larger RTT of transmission via the cellular link. For the cases to fetch videos which are not in the AMES-Cloud (but in our server at lab), 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 practical, there are be optimized links in the Internet backbone among video providers and cloud providers, and even recent video providers are just using cloud storage and computing service. Therefore this delay can be significantly reduced in practice. Also this won’t happen frequently, since most of the popular videos will be already prepared in the AMES-Cloud.

V. CONCLUSION

In this paper, we discussed our proposal of an adaptive mobile video streaming and sharing framework, called AMES-Cloud, which efficiently stores videos in the clouds (VC), and utilizes cloud computing to construct private agent (sub VC) for each mobile user to try to offer “non-terminating” video streaming adapting to the fluctuation of link quality based on the Scalable Video Coding technique. Also AMES-Cloud can further seek to provide “non buffering” experience of video streaming by background pushing functions among the VB, sub VBs and local VB of mobile users. We evaluated the AMES-Cloud by prototype implementation and shows that the cloud computing technique brings significant improvement on the adaptively of the mobile streaming. The focus of this paper is to verify how cloud computing can improve the transmission adaptability and prefetching for mobile users. We ignored the cost of encoding workload in the cloud while implementing the prototype. As one important future work, we will carry out large-scale implementation and with serious consideration on energy and price cost. In the future, we will also try to improve the SNS-based prefetching, and security issues in the AMES-Cloud.

VI. REFERENCES

[1] Xiaofei Wang, Student Member, IEEE, Min Chen, Senior Member, IEEE, Ted “Taekyoung” Kwon, Senior Member, IEEE, Laurence T. Yang, Senior Member, IEEE, Victor C.M. Leung, Fellow, IEEE, “AMES-Cloud: A Framework of Adaptive Mobile Video Streaming and Efficient Social Video Sharing in the Clouds”, IEEE Transactions on Cloud Computing Vol: 15 No: 4 Year 2013.

[2] CISCO, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016,” Tech. Rep., 2012.

[3] Y. Li, Y. Zhang, and R. Yuan, “Measurement and Analysis of a Large Scale Commercial Mobile Internet TV System,” in ACM IMC, pp. 209–224, 2011.

[4] T. Taleb and K. Hashimoto, “MS2: A Novel Multi-Source Mobile-Streaming Architecture,” in IEEE Transaction on Broadcasting, vol. 57, no. 3, pp. 662–673, 2011.

[5] X. Wang, S. Kim, T. Kwon, H. Kim, Y. Choi, “Unveiling the Bit Torrent Performance in Mobile Wi-MAX Networks,” in Passive and Active Measurement Conference, 2011.

[6] A. Nafaa, T. Taleb, and L. Murphy, “Forward Error Correction Adaptation Strategies for Media Streaming over Wireless Networks,” in IEEE Communications Magazine, vol. 46, no. 1, pp. 72–79, 2008.

[7] J. Fernandez, T. Taleb, M. Guizani, and N. Kato, “Bandwidth Aggregation-aware Dynamic QoS Negotiation for Real-Time Video Applications in Next-Generation Wireless Networks,” in IEEE Transaction on Multimedia, vol. 11, no. 6, pp. 1082–1093, 2009.

[8] T. Taleb, K. Kashibuchi, A. Leonardi, S. Palazzo, K. Hashimoto, N. Kato, and Y. Nemoto, “A Cross-layer Approach for An Efficient Delivery of TCP/RTP-based Multimedia Applications in Heterogeneous Wireless Networks,” in IEEE Transaction on Vehicular Technology, vol. 57, no. 6, pp. 3801–3814, 2008.

[9] K. Zhang, J. Kong, M. Qiu, and G.L Song, “Multimedia Layout Adaptation Through Grammatical Specifications,” in ACM/Springer Multimedia Systems, vol. 10, no. 3, pp.245–260, 2005.

[10] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon, “Real-Time System for Adaptive Video Streaming Based on SVC,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1227–1237, Sep. 2007.

[11] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[12] H. Schwarz and M. Wien, “The Scalable Video Coding Extension of the H. 264/AVC Standard,” in IEEE Signal Processing Magazine, vol. 25, no. 2, pp.135–141, 2008.