International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 2, February - 2014

Empirical Analysis of User Based Cloud Mobile Video Streaming

M. Bhuvaneswari
Jayam College of Engineering and Technology,
Dharmapuri, Tamilnadu, India

R. Kavipriya
Jayam College of Engineering and Technology,
Dharmapuri, Tamilnadu, India

***Abstract*— A mobile network has unpleasant because of the demands on video traffic; the wireless link capacity cannot keep up with the traffic claim. The break between the traffic claim and the link capability, along with time-varying link conditions, produce the results in poor service quality of video streaming such as long buffering time and irregular disruptions. The cloud computing technology, propose a new mobile video streaming framework of cloud, which has two main parts: adaptive mobile video streaming and efficient social video sharing. It constructs a private agent to provide video streaming services efficiently for each mobile user. Adaptive mobile video streaming lets the private agent adaptively adjust the streaming with a scalable video coding technique based on the feedback of link quality. Efficient social video sharing monitors the social network interactions among mobile users and their private agents try to pref-etch video content in advance. Scalable video coding and adaptive streaming techniques can be jointly combined to accomplish effectively the best possible quality of video streaming services. To implement a prototype of the Cloud framework to demonstrate its concert. It is shown that the private agent in the clouds can effectively provide the adaptive streaming, and carry out video sharing (i.e., pref-etching) based on the social network analysis. Carry out large-scale implementation and with serious consideration on energy and price cost and ignored the cost of encoding workload in the cloud while implementing the prototype.**

Keywords: Adaptive video streaming, cloud computing, mobile networks, scalable video coding, social video sharing.

I. INTRODUCTION

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

A. Scalability:

Mobile video streaming services should support a wide spectrum of mobile devices; they have different video resolution, different computing power, different wireless links (like 3G and LTE). Also, the available link capacity of a mobile device may vary over time and space depending on its signal power, other user's traffic in the same mobile, and link condition deviation. Storing several 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 base layer (BL) with multiple enhances layers (ELs). These sub streams can be encoded by exploiting three scalability features: (i) spatial scalability by layering image resolution (screen pixels), (ii) sequential 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 deliver. However, the more ELs can be delivered, the enhanced value of the video stream is achieved.

B. Adaptability:

Traditional video streaming techniques designed by considering relatively stable traffic links between servers and users, perform poorly in portable environments. Thus the fluctuating wireless connection status should be properly dealt with to provide 'tolerable" video streaming services. To deal with this issue, we have to adjust the video bit rate adapting to the currently time varying available link bandwidth of each mobile consumer. Such adaptive streaming techniques can effectively reduce packet losses and bandwidth waste. Scalable video coding and adaptive streaming technique scan be jointly combined to accomplish effectively the best possible quality of video streaming services. With the intention of, we can dynamically adjust the number of SVC layers depending on the current link status. However most of the proposals seeking to jointly utilize the video scalability and adaptability rely on the active control on the server side. With the intention of, 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 accessible bandwidth for every user. Hence 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 center scan easily provision for large scale realtime video services as investigated in. 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 repost videos among friends and members in the same group, which imply a user may watch a video that her friends have recommended. Users in SNSs can also follow famous and popular users based on their interests (e.g., an official facebook or twitter account that shares the newest pop music videos), which is probable 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 so far. 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. This paper constructs a private agent for each mobile user in cloud computing environment, which is used by its two main parts:

- AMoV(adaptive mobile video streaming)
- ESoV (efficient social video sharing)

The offerings of this paper can be summarized as follows:

- Adaptive mobile video streaming offers the best possible streaming experiences by adaptively controlling the streaming bit rate depending on the fluctuation of the link quality. Adaptive mobile video streaming adjust 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 condition. Private agents of users are dynamically initiated and optimized in the cloud computing stage. 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 release network, and the second tier is a data interior. Among this structure, video sharing can be optimized inside the cloud. Preventable redundant downloads of popular videos can be prevented. Based on the analysis of the SNS activities of mobile users, Efficient social video sharing 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 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 prefetched.

II. RELATED WORK

A. *Adaptive Video Streaming Techniques*

In the adaptive streaming, the video traffic speed is attuned on the fly so that a user can experience the maximum possible video quality based on his or her link's time-varying bandwidth capacity. There are mainly two types of adaptive streaming technique, depending on whether the adaptively is controlled by the client or the server. The Microsoft's Smooth Streaming is alive adaptive streaming service which can switch among different bit rate segment encoded with configurable bit rates and video resolutions at servers, even as clients with passion request videos based on local monitoring of link value. Adobe and Apple also developed client side HTTP adaptive live streaming solutions operating in the related method. There are also some related adaptive streaming services where servers

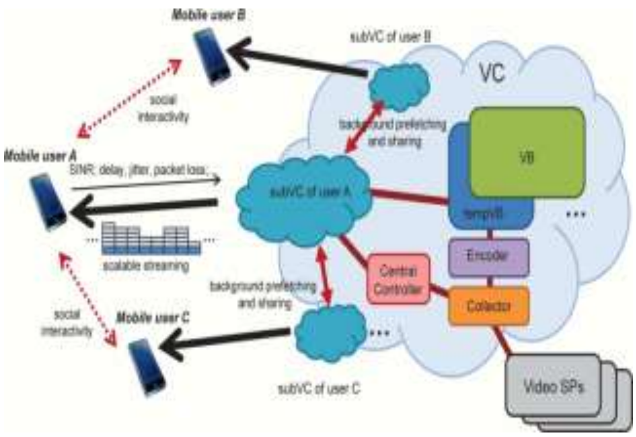


Fig. 1. Illustration of the AMES-Cloud framework with the Video Cloud (VC), subVCs for mobile users, the Video Base (VB), and the Video Service Providers (SPs).

controls the adaptive transmission of video segments, for example, the Quav live Adaptive Streaming. Conversely, most of these solutions maintain multiple copies of the video content with unusual bit rates, which brings enormous load of storage on the server. Regarding rate adaptation scheming techniques, TCP friendly rate control methods for streaming services over mobile networks are projected, where TCP throughput of a flow is predict as a function of packet loss rate, round trip time and packet size. In view of the predictable throughput, the bit rate of the streaming traffic be able to adjusted. A rate adaptation algorithm for conversational 3G video streaming is introduced. Then, a few cross layer adaptation techniques, which can acquire more accurate information of link quality so that the rate adaptation can be more exactly concluded. Conversely, the servers have to always control and thus suffer from large workload. Recently the H.264 Scalable Video Coding (SVC) technique has gained a momentum. An adaptive video streaming system based on SVC is deployed, which studies the real time SVC decoding and encoding at PC servers. The proposes a quality oriented scalable video releaseusing SVC, but it is only tested in a replicated LTE Network. Regarding the encoding presentation of SVC, Cloud Stream mainly proposes to deliver high-quality streaming videos through a cloud-based SVC proxy, which discovered that the cloud computing can significantly improve the performance of SVC coding. The beyond studies stimulate us to use SVC for video streaming on top of cloud computing.

B. Mobile Cloud Computing Techniques

The cloud computing has been well positioned to provide video streaming services, particularly in the wired Internet because of its scalability and capability. For example, the quality assured bandwidth auto scaling for VoD streaming based on the cloud computing is proposed and the CALMS framework is a cloud assisted live media streaming service for globally distributed users. Conversely, extending the cloud computing based services to mobile environments requires more factors to consider: wireless link dynamics, user mobility and the limited capability of mobile devices. More recently, new designs for users on top of mobile cloud computing environments are proposed, which virtualized private agents that are in charge of satisfying the requirements (e.g., QoS) of individual users such as Cloud-lets and Stratus. Thus, we are motivated to design the AMES-Cloud framework by using virtual a gents in the cloud to provide adaptive video streaming services.

II. AMES-CLOUD FRAMEWORK

In this section we explain the AMES-Cloud framework includes the Adaptive Mobile Video streaming (AMoV) and the Efficient Social Video sharing (ESoV). As shown in Fig. 1, the whole video storing and streaming system in the cloud is called the Video Cloud (VC). In the VC, there is a large-scale video base (VB), which stores the most of the popular video clips for the video service providers(VSPs). A temporal video base (tempVB) is used to cache new candidates for the popular videos, although tempVB counts the access frequency of every video. The VC keeps running a collector to seek videos which are already popular in VSPs and will re-encode the collected videos into SVC format and store into tempVB. By this 2-tier storage, the AMES-Cloud can keep serving most of popular videos everlastingly. Note that management work will be handled by the controller in the VC. Specialized for each mobile user, a sub-video cloud (subVC) is created dynamically if there is any video streaming demand from the user. The subVC has a sub video base (subVB), which stores the recently fetched video segments. Reminder that the video deliveries among the subVCs and the VC in most cases are actually not “copy”, but just “link” operations on the same file eternally within the cloud data center. There is also encoding function in subVC (actually a smaller scale encoder instance of the encoder in VC) and if the mobile user demands a new video which is not in the subVB or the VB in VC, the subVC will fetch, encode and transfer the video. All over video streaming, mobile users will always report link conditions to their corresponding subVCs, and then the subVCs offer adaptive video streams. Reminder that each mobile device also has a temporary caching storage, which is called local video base (localVB), and is used for buffering and prefetching. Note that as the cloud service may across different places or even continents so in the case of a video delivery and prefetching between different data centers and transmission will be carried out, which can be then called “copy”. And because of the optimal deployment of data centers as well as the capable links among the data centers, the “copy” of a large video file takes tiny delay.

IV. AMOV: ADAPTIVE MOBILE VIDEO STREAMING

A. SVC

As shown in Fig. 2, traditional video streams with fixed bit rates cannot adapt to the fluctuation of the link value. For a particular bit rate, if the sustainable link bandwidth varies much the video streaming can be frequently terminated due to the packet loss. In SVC a combination of the three lowest scalability is called the Base Layer (BL) while the enhanced combinations are called Enhancement Layers (ELs). To this regard, if BL is guaranteed to be delivered, although more ELs can be also obtained when the link can afford, a better video quality can be predictable. By using SVC indoctrination techniques, the server doesn’t need to

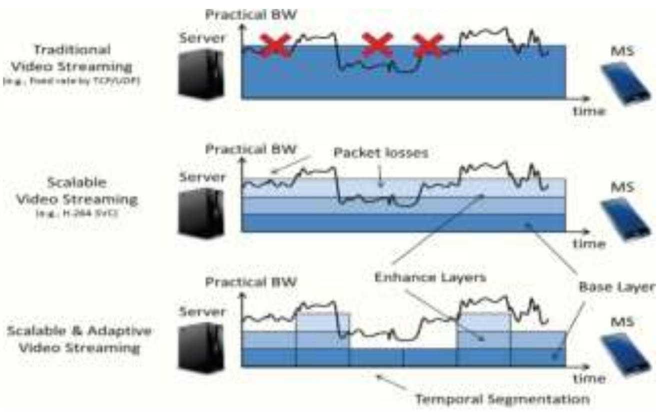


Fig. 2. A comparison of the traditional video streaming, the scalable video streaming and the streaming in the AMES-Cloud framework.

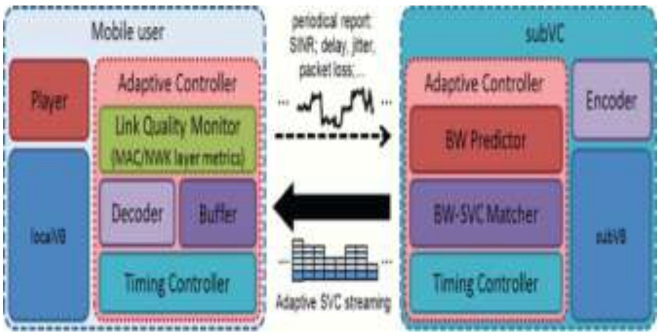


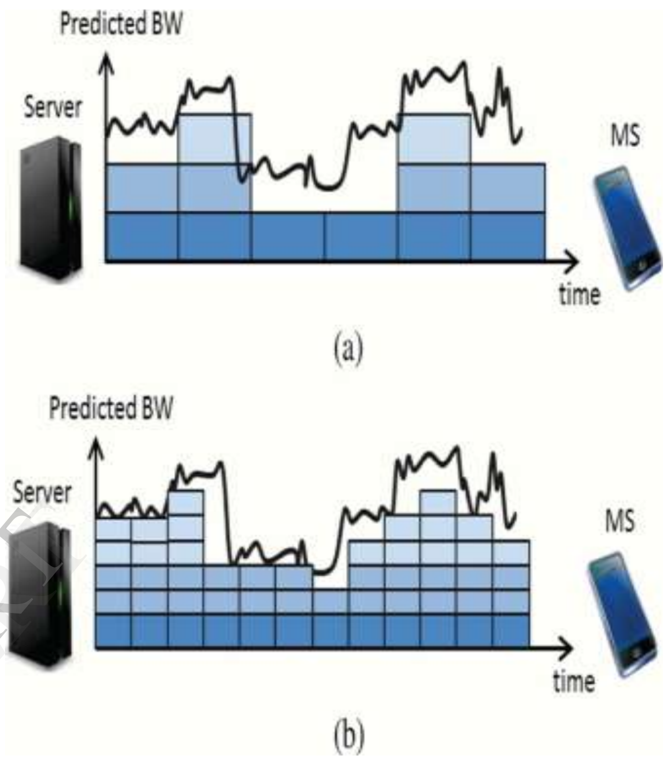
Fig. 3. Functional structure of the client and the subVC.

concern the client side or the link quality. Constant some packets are lost, the client still can decode the video and display.

B. Adaptability With Monitoring on Link Quality

We design the mobile client and the subVC with the structure as shown in Fig. 3. The link quality monitor at mobile client keeps tracking on metrics including signal strength, packet round-trip-time (RTT), jitter and packet loss with a certain task 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 , Note that the video is also split by temporal segmentation by interval . Once the subVC gets the information of the link quality, it will perform a calculation and predict the potential bandwidth in the next time window. Note that we will use "predicted bandwidth" and "predicted goodput" interchangeably in following parts.
Suppose sequence number of current time window is, the predicted bandwidth can be estimated by: where, indicating the importance of each factor, is for packet loss rate, is for RTT, is for the signal to interference and noise ratio and are three functions reflecting the value change of each factor compared with that of last time window. Actually in this paper we deploy a measurement based prediction, that is we directly use of last time window as the of subsequently window which is proved with already high accuracy.

C. Matching Between Bandwidth Prediction and SVC Segments



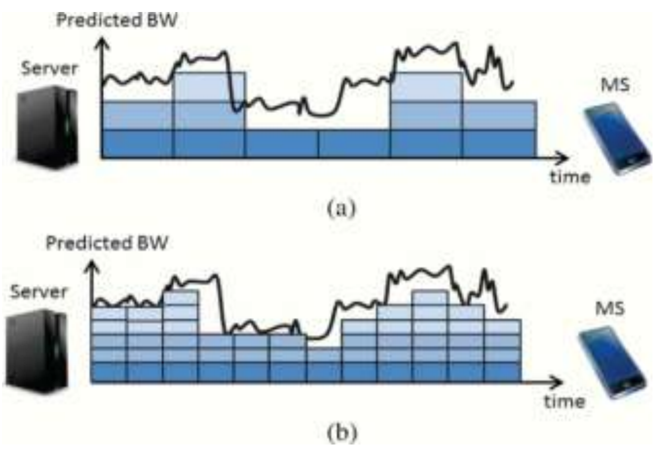


Fig. 4. Matching between predicted bandwidth and SVC-segments with different resolutions. (a) Fine-grained (high resolution). (b) Coarse-grained (low resolution).

After obtaining the predicted bandwidth or say goodput of subsequently window, subVC will match and decide how many video segments of BL and ELs can be transmitted approximately. We hereby define the term “resolution” to indicate the level of temporal segmentation and the number of ELs. If is small and there are more ELs, the SVC-based video source is with a higher resolution. We illustrate two cases of coarse-grained (low resolution) and a relatively fine grained (high resolution) for matching between the SVC segments and the predicted good put in Fig. 4. The resolution with two ELs and a larger can hardly fit to the signal fluctuation, and thus there are some bandwidth wasted or packets lost.

In contrast a higher resolution with more ELs and a smaller can always fit the fluctuation of the bandwidth. Conversely a higher resolution also induces more encoding workload to the servers. Suppose there are totally ELs, and the bit rate of the EL is denoted as while the bit rate of the BL is). We let indicate the SVC segment of BL with temporal sequence , and let indicate the SVC segment of the EL with temporal sequence. So the algorithm of matching between predicted bandwidth and SVC segments is shown in Algorithm 1 as following:

Algorithm 1 Matching Algorithm between BW and Segments

```
i=0
BW0=RBL
Transmit BL0
Monitor BW0practical
repeat
sleep for Twin
Obtain pi,RTTi, SINRi etc., from client's report
Prdict BWi+1estimate (or BWi+1estimate= BWipractical)
K=0
BWEL=0
repeat
k++
if k>=j break
BWEL=BWEL+ RELk
Until BWEL>= BWi+1estimate –RBL
Transmit BLi+1 and ELi+11,ELi+12,…,ELi+1k-1
Monitor BWi+1practical
i++
Until all video segments are transmitted
```

V. ESOV: EFFICIENT SOCIAL VIDEO SHARING

A. *Social Content Sharing*

In SNSs, users subscribe to recognized friends, famed people and particular interested content publishers as well; also there are various types of social activities among users in SNSs such as direct message and public redistribution. Intended for spreading videos in SNSs one can post a video in the public and his/her subscribers can quickly see it; one can also directly recommend a video to specified friend(s); further more one can periodically get noticed by subscribed content publisher for new or popular videos. We describe different strength levels for those social activities to indicate the probability that the video shared by one user may be watched by the receivers of the one's sharing activities, which is called a “hitting probability” so that subVCs can carry out effective background prefetching at subVB and even localVB. Since after a video sharing activity there may be a certain delay that the recipient gets to know the sharing and initiates to watch. Hence the prefetching in prior will not impact the users at most cases. As an alternative a user can click to see without any buffering delay as the beginning part or even the whole video is already prefetched at the localVB. The quantity of prefetched segments is mainly determined by the strength of the social actions. As well as the prefetching from VC to subVC only refers to the “linking” action so there is only file locating and linking operations with tiny delays; the prefetching from subVC to localVB also depends on the strength of the social activities but will also consider the wireless link status. 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 recipient:

- Subscription: Like the popular RSS services an user can subscribe to a particular video publisher or a special video collection service based on his/her welfare. 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.
- Direct recommendation: InSNSs, an user directly recommend a video to particular friend(s) with a short message. The recipients of the message may watch it with very high probability. This is considered as “strong”.

Table I Social Activities And Background Pushing Strategies

	Direct recommendation	Subscription	Public sharing
VB→subVB	All	Parts	Little
subVB→locVB (via Wi-Fi)	All	Parts	Little
subVB→locVB (via 3G/4G)	Parts	Little	None

- Public sharing:

Each user in SNSs has a timeline-based of activity stream which shows his/her recent activities. The activity of a user watching or sharing a video can be seen by his/her friends (or followers). We consider this public sharing with the “weak” connectivity among users because not many people may watch the video that one has seen without direct recommendation.

B. Prefetching Levels

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

• "Parts": Because the videos that published by subscriptions may be watched by the subscribers with a not high probability we propose to only push a part of BL and ELs segments for example, the first 10% segments.

• "All": The video shared by the direct recommendations will be watched with a high probability, so we propose to prefetch the BL and all ELs, in order to let the recipient(s) directly watch the video with a good quality without any buffering.

• "Little": The public sharing has a weak connectivity among users so the probability that a user's friends (followers) watch the video that the user has watched or shared is low. We intend to only prefetch the BL segment of the first time window in the beginning to those who have seen his/her activity in the stream. The prefetching happens among subVBs and the VB, also more importantly will be performed from the subVB to localVB of the mobile device depending on the link value. If a mobile user is covered by Wi-Fi access due to Wi-Fi's capable link and low price (or mostly for free) subVC can push as much as possible in most cases. Conversely if it is with a 3G/4G connection which charges a lot and suffers limited bandwidth we propose to downgrade the prefetching level to save energy and cost as listed in Table I but users can still benefit from the prefetching effectively. Note that some energy prediction methods can be deployed in order to actively decide whether current battery status is suitable for "parts" or "little". If a user, A, gets the direct recommendation of a video from another user B. User A's subVC will immediately prefetch the video either from B's subVB, or from the VB (or tempVB) at the level of "all", if A is with Wi-Fi access. However if user A is connected to 3G/4G link we will selectively prefetch a part of the video segment to A's local storage at the level of "parts". Note that the subscribed videos will be not prefetched when user A is at 3G/4G connection, as it is downgraded from "little" to none. A better extension of the prefetching strategy by social activities can be designed by an self updating mechanism from the user's hitting history in an evolutionary method. This learning based prefetching is out of the scope of this paper and will be explored as our future work.

VI. VIDEO STORAGE AND STREAMING FLOW BY AMOV AND EMOS

The two parts, AMoV and EMoS, in AMES-Cloud 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 in EMoS the AMoV will still monitor and improve the transmission considering the link status; with a certain amount of prefetched segments by EMoS, AMoV can offer better video quality. With the efforts of AMoV and EMoS, we illustrate the flow chart of how a video will be streamed in Fig.5. Communication that in order to exchange the videos among the localVBs, subVBs, tempVB and the VB, a videomap (VMap) is used to indicate the required segments. Once a mobile user starts to watch a video by a link, the local VB will first be checked whether there are any prefetched segments of the video so that it can directly start. Stipulation there is none or just some parts the client will report a corresponding VMap to its subVC. Stipulation the subVC has

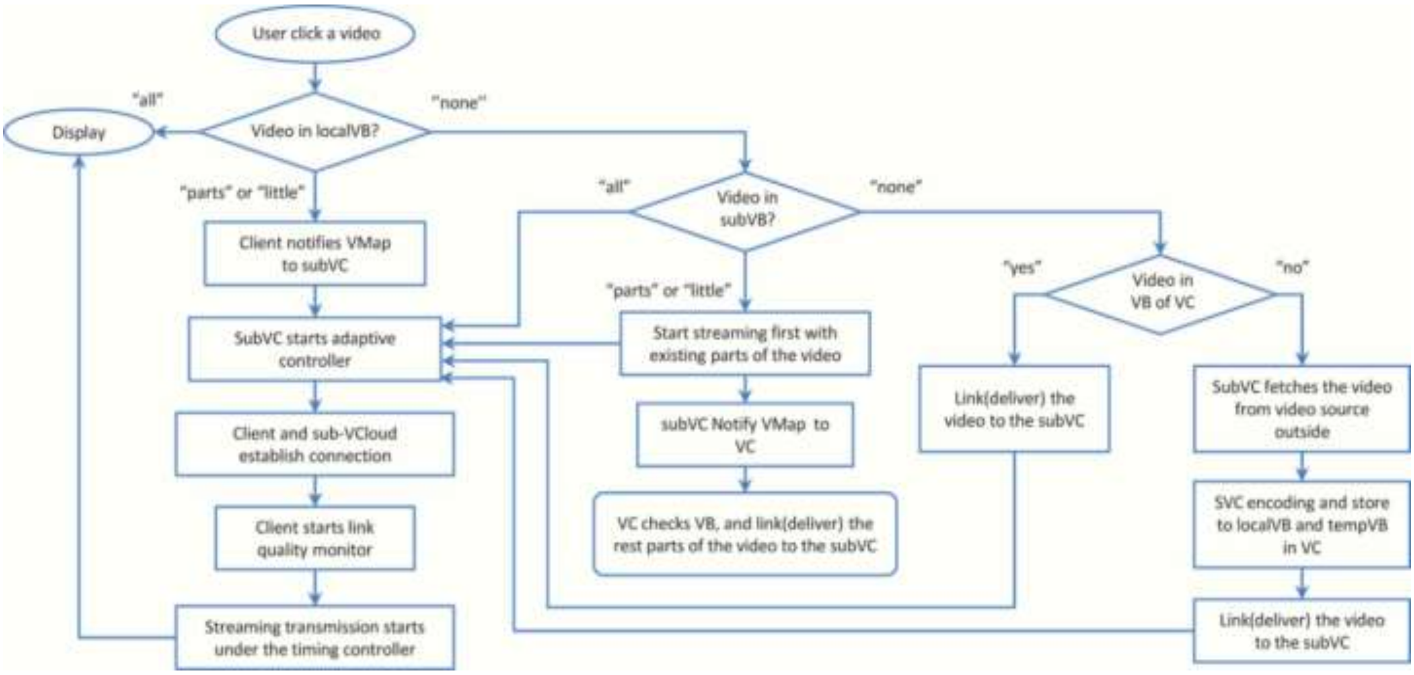


Fig. 5. Working flow of video streaming in the subVC and VC of AMES-Cloud framework.

refetched parts in subVB the subVC will initiate the segment transmission. Although if there is also none in the subVB the tempVB and VB in the center VC will be checked. Intended for a non-existing video in AMES-Cloud the collector in VC will immediately fetch it from external video providers via the link; after re-encoding the video into SVC format taking a bit longer delay the subVC will transfer to the mobile user. Also in AMESCloud if a video is shared among the subVCs at a certain frequency threshold (e.g., 10 times per day), it will be uploaded to the tempVB of the VC; and if it is further will be stored with a longer life time in the VB. In such a manner, which is quite similar to the leveled CPU cache, the subVB and VB can always store fresh and popular videos in order to increase the probability of reusage.

VII. IMPLEMENTATION AND EVALUATION

We evaluate the performance of the AMES-Cloud framework by a prototype implementation. We choose the Ucloud server (premium) in the cloud computing service offered by Korean Telecom, and utilize the virtual server with 6 virtual CPU cores (2.66 GHz) and 32 GB memory, which is fast enough for en- coding 480P (480 by 720) video with H.264 SVC format in 30 fps at real time. 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 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 down town 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

formatwith480PresolutiondownloadedfromYouTube.Its size is 13.849 Mbytes and with a duration of 180 seconds. 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 version 9.1. 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 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). Thus we define the best resolution configuration as "1+5+2+2". As well as also test different resolution configurations, including "1+1+1+1", "1+2+2+2 ", "1+3+2+2 " and "1+4+2+2 ".

A. *Adaptive Video Streaming Based on SVC*

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 value. Note that all tests are run 5 times. The collected the relative errors for the predicted bandwidth to the practical bandwidth for every time window, calculated by |BWestimate-BWpractical| / {BWpractical} , are shown in Fig. 6, where the bar indicates the 25% and 75% quartiles, and the whiskers indicate the 5% and 95% percentiles. When 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 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.

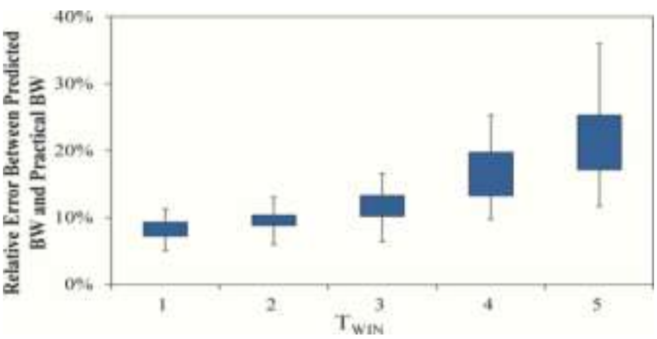


Fig. 6. Relative errors between predicted bandwidth and practical bandwidth (percentage).

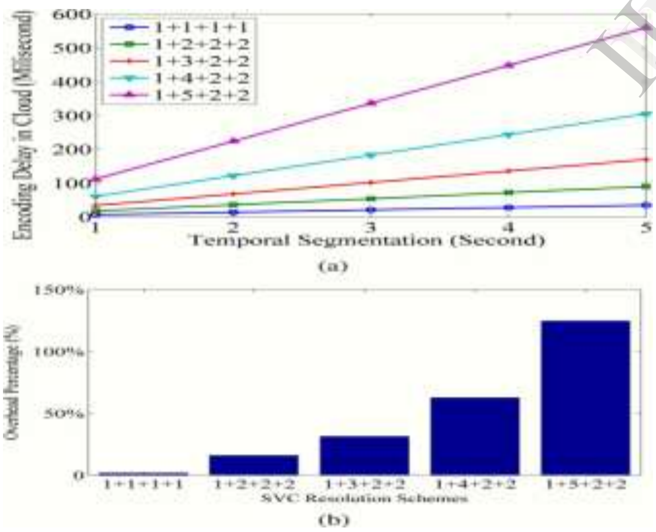


Fig. 7. Evaluation of SVC resolution schemes. (a) Delay of difference SVC resolution schemes in the Cloud (b) Overhead of different SVC resolutions schemes in the Cloud.

TABLE II 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)

B. *Video Streaming in SubVC and VC*

We evaluate working of H.264 SVC in AMES- Cloud framework regarding the above mentioned SVC resolution configurations. As exposed in Fig. 7(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.

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 exposed in Fig. 7(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 " has 61% overhead, and " 1+5+2+2" has even 120% overhead which is not efficient.

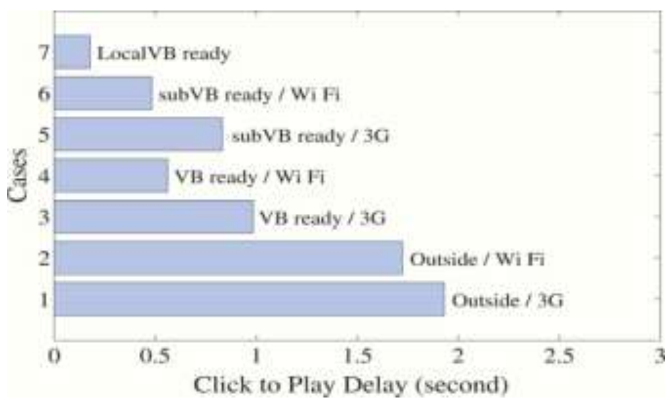


Fig. 8. Average click-to-play delay for various cases.

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 and localVBs of the mobile users based on their activities in SNSs. Evaluate the required delays for different levels of prefetching as exposed in Table. 2. 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. We can see that prefetching supported by the cloud computing is significantly fast. As soon as prefetching via wireless links it takes several seconds. Conversely 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 exposed in Fig. 8 if the video has been cached in localVB the video can be displayed nearly immediately with ignorable delay. As soon as we watch video which is fetched from the subVC or the VC it generally takes no more than 1 second to start. Conversely 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 privileged. 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 realistic 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 observe. Moreover this won't happen frequently since most of the popular videos will be already prepared in the AMES-Cloud.

VIII. 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 (subVC) 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. Furthermore AMES-Cloud can further seek to provide "non-buffering" experience of video streaming by background pushing functions among the VB, subVBs and localVB of mobile users. Evaluated the AMES-Cloud by prototype implementation and shows that the cloud computing technique brings significant improvement on the adaptivity of the mobile streaming. The spotlight of this paper is to verify how cloud computing can improve the transmission adaptability and prefetching for mobile users. We unobserved the cost of encoding workload in the cloud while implementing the prototype. Since 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.

REFERENCES

[1]. Almeida.J, Almeida.V, Benevenuto.F and Redrigues.T (2009), "Video interactions in online social networks," ACM Trans. Multimedia Computing, Communication Application., vol. 5, no. 4, pp.

[2]. Akshabi.S, Begen.A.C and Dovrolis.C ," Evaluation of rate adaptation algorithms in adaptive streaming over HTTP", IEEE Trans.Mobisystem, vol 76.

[3]. Amon.P , Cazoulat.R, Graffunder.A, Hutter.A and Wien.M, (sep 2007), "Real-time system for adaptive video streaming based on SVC," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 9, pp. 1227–1237.

[4]. Antonio Ortega , Jose I.Ronda, Julian Cabrera, "Stochastic rate control of video coders for wireless channels", IEEE Trans, vol 65.

[5]. Athina Markopolou, John Apostolopolos, Nicholas Bambos, "Joint power playout control for media streaming over wireless links", IEEE Wireless. Vol 34, no. 44 .

[6]. Balasubramanian.A, Mahajan.R, and Venkataramani.A (2010), "Augmenting mobile 3G uses Wi-Fi," in Proc. ACM MobiSystem, pp. 209–222.

[7]. Chetan.S, Kumar.G, Dinesh.K and Mathew.K(2011),"Cloud computing for mobile world ",IEEE Trans. Multimedia, vol. 17, no. 6, pp.1221-123

[8]. Fellow, Philp.A.Chou, Zhourong Miao, "Rate Distoration optimized streaming of packetized media",2010.

[9]. Gu.Z, Li.J. Ming, Qiu.G.M, Quan.Z and Qin.X (2012) "Online optimization for scheduling preempt able tasks on IaaS cloud systems," J. Parallel Distribution. Computing, vol. 72, no. 5, pp. 666–677.

[10]. Hashimoto.K and T. Taleb(2011), "MS2: A novel multi-source mo- bile-streaming architecture," IEEE Trans. Broadcasting, vol. 57, no. 3, pp. 662–673.

[11]. Haiyang wang, Kwon .T.T, Xiaofei wang , Yanghee Choi, "Cloud assisted adaptive video streaming and social aware video prefetching for mobile users" ,2010.

[12]. Hsinchu, Taiwan, Yi-Hsing Tsai," The cloud streaming service migration in cloud video storage system", IEEE Tans.Broadcasting vol 90.

M.Bhuvaneswari , Student of Master of Computer Science and Engineering at Jayam College of Engineering and Technology under Anna University Chennai-India.(bhuvimanoharan@gmail.com)

R.Kavipriya, Assistant Professor at Jayam College of Engineering and Technology under Anna University Chennai-India.